

NCT[®] 2xxT

NCT[®] 3xxT

Eszterga vezérlő

Programozási leírás

Az n.15.2 szoftver változattól

Gyártó és fejlesztő: NCT Ipari Elektronikai kft.

H1148 Budapest Fogarasi út 7

✉ Levélcím: 1631 Bp. pf.26

☎ Telefon: (+36 1) 467 63 00

☎ Telefax:(+36 1) 467 63 09

Villanyposta: nct@nct.hu

Honlap: www.nct.hu

Tartalomjegyzék

1 Bevezetés	<u>10</u>
1.1 Az alkatrészprogram	<u>10</u>
1.2 A csatorna	<u>11</u>
1.3 Alapfogalmak	<u>12</u>
2 Vezérelt tengelyek	<u>16</u>
2.1 A tengelyek neve és száma	<u>16</u>
2.2 Kiterjesztett tengelynevek	<u>17</u>
2.3 Tengelyek csatornához rendelése	<u>17</u>
2.4 A tengelyek mértékrendszere, és a pozíciókijelzés pontossága	<u>18</u>
3 Előkészítő funkciók (G kódok)	<u>19</u>
4 Az interpoláció	<u>24</u>
4.1 A pozicionálás (G0)	<u>24</u>
4.1.1 Pozicionálás lineáris interpolációval	<u>24</u>
4.1.2 Pozicionálás a gyorsmeneti mozgások átlapolásával	<u>25</u>
4.2 Az egyenes interpoláció (G1)	<u>26</u>
4.3 A körinterpoláció (G2, G3)	<u>28</u>
4.3.1 A síkbeli spirálinterpoláció (G2, G3)	<u>33</u>
4.3.2 A hengeres spirálinterpoláció (G2, G3)	<u>35</u>
4.3.3 A kúpos spirálinterpoláció (G2, G3)	<u>38</u>
4.4 Egyenletes emelkedésű menet vágása (G33)	<u>40</u>
4.5 A polárkoordináta interpoláció (G12.1, G13.1)	<u>42</u>
4.6 A hengerinterpoláció (G7.1)	<u>46</u>
5 A koordinátaadatok	<u>49</u>
5.1 Abszolút és inkrementális programozás (G90, G91), az I operátor, U, V, W cím	<u>49</u>
5.2 Inch/Metrikus átalakítás (G20, G21)	<u>50</u>
5.3 Átmérőben, vagy sugárban történő programozás	<u>50</u>
5.3.1 Sugárban/átmérőben való programozás átkapcsolása (G10.9)	<u>51</u>
5.4 Adatmegadás polárkoordinátákkal (G15, G16)	<u>52</u>
5.5 Koordinátaadat megadása és pontossága	<u>54</u>
5.6 Forgó tengelyek átfordulás kezelése	<u>55</u>
6 Az előtolás	<u>61</u>
6.1 A gyorsmeneti előtolás	<u>61</u>
6.2 A munkaelőtolás	<u>61</u>
6.2.1 Percenkénti (G94) és fordulatonkénti (G95) előtolás	<u>62</u>
6.3 Az előtolásvezérlő funkciók	<u>62</u>
6.3.1 Pontos megállás a mondat végén (G9)	<u>62</u>
6.3.2 Pontos megállás üzemmód (G61)	<u>63</u>
6.3.3 Folyamatos forgácsolás (G64)	<u>63</u>
6.3.4 Override és stop tiltás (G63)	<u>63</u>
6.3.5 Automatikus előtoláscsökkentés belső sarkoknál (G62)	<u>63</u>

6.4 Automatikus előtöláscsökkentés belső köríveknél..	64
7 A gyorsulás.	65
7.1 Automatikus lassítás sarkoknál G64 állapotban..	68
7.2 A pálya mentén normális irányban fellépő gyorsulások korlátozása.	71
7.3 A gyorsulásugrás korlátozása.	73
8 A várakozás (G4).	75
9 A referenciapont.	76
9.1 Automatikus referenciapont felvétel (G28).	77
9.2 A 2., 3., 4. referenciapontra állás (G30).	78
10 Koordinátarendszerek, síkválasztás.	79
10.1 A gépi koordinátarendszer.	79
10.1.1 Pozicionálás a gépi koordinátarendszerben (G53).	80
10.2 A munkadarab koordinátarendszerek.	80
10.2.1 A munkadarab koordinátarendszer kiválasztása (G54...G59).	81
10.2.2 A munkadarab koordinátarendszerek eltolásának beállítása (G10 L2).	83
10.2.3 A bővített munkadarab koordinátarendszerek kiválasztása (G54.1 P).	84
10.2.4 A bővített munkadarab koordinátarendszerek eltolásának beállítása (G10 L20)	84
10.2.5 Új munkadarab koordinátarendszer létrehozása (G92).	85
10.3 A lokális koordinátarendszer (G52).	86
10.4 Síkválasztás (G17, G18, G19).	87
11 Az orsófunkciók.	89
11.1 Az orsó fordulatszám parancs (S kód).	89
11.1.1 Hivatkozás több orsóra. Az S cím kiterjesztése	89
11.1.2 Orsók csatornákhöz rendelése.	90
11.2 Az orsóvezérlő M funkciók.	90
11.3 A fordulatszám tartományok kezelése..	91
11.4 A főorsó. A főorsó kiválasztása.	92
11.5 A konstans vágósebességszámítás.	92
11.5.1 A konstans vágósebesség számítás megadása (G96 S, G97)..	93
11.5.2 A fordulatszám korlátozása konstans vágósebességszámításkor (G92 S)..	94
11.5.3 Tengely kijelölése konstans vágósebesség számításához (G96 P).	94
11.6 Az orsók fordulatszám ingadozás figyelése.	94
11.7 Az orsók pozicionálása.	95
11.7.1 Az orsók orientálása.	95
11.7.2 Az orsók megállítása és a pozíciószabályzó hurok zárása..	95
11.7.3 Az orsók pozicionálásának programozása..	96
11.7.4 Két orsó pozícióhelyes szinkronizálása.	97
11.7.5 Az orsók pozíciószabályozott üzemmódjának kikapcsolása.	99
11.8 Orsó tengellyé, tengely orsóvá alakítása.	99
12 A T funkció.	101
12.1 A szerszámváltás programozására.	101

13 Vegyes és segédfunkciók	103
13.1 Vegyes funkciók: M kódok.....	103
13.2 Segédfunkciók (A, B, C, U, V, vagy W).....	105
13.3 Pufferürítő funkciók.	105
14 Az alkatrészprogram szervezése	107
14.1 A mondatszám (N cím).....	107
14.2 Feltételes mondatkihagyás (/ cím).	107
14.3 Megjegyzések írása az alkatrészprogramba: (komment).	108
14.4 Főprogram és alprogram.....	108
14.4.1 Programok azonosítása a tárban. A programszám (O).....	108
14.4.2 Alprogram hívása (M98).	109
14.4.3 Visszatérés alprogramból (M99).	111
14.4.4 Ugrás a főprogramon belül.....	113
14.5 A csatornák közötti összevárás M funkciói.	113
15 A szerszámkorrekció	115
15.1 A korrekciós tár. Hivatkozás szerszámkorrekcióra (T, vagy D).....	115
15.2 A 2. geometriai korrekciós tár.	118
15.3 Szerszámkorrekciós értékek módosítása programból (G10).	120
15.4 A szerszámhossz-korrekció figyelembe vétele T kódra.	120
15.5 A szerszámhossz-korrekció figyelembe vétele G kódra (G43.7, G49).....	123
15.6 Maró korrekciók hívása (G43, G44).....	124
15.7 A szerszámsugár korrekció (G40, G41, G42).....	128
15.7.1 A sugárkorrekció számítás bekapcsolása. Ráállítás a kontúrra.....	131
15.7.2 A sugárkorrekció számítás bekapcsolt állapota. Haladás a kontúron.....	137
15.7.3 A szerszámsugár korrekciószámítás kikapcsolása. Leállítás a kontúrról.....	144
15.7.4 Irányváltás a sugárkorrekció számításban.....	149
15.7.5 A vektor megtartás programozása (G38).	152
15.7.6 Sarokív programozása (G39).	153
15.7.7 A kontúrkövetés zavarproblémái. Interferenciavizsgálat.....	155
16 Különleges transzformációk	163
16.1 Alakzat elforgatása adott pont körül (G68, G69).	163
16.2 Alakzat léptékezése adott ponthoz viszonyítva (G50, G51).....	165
16.3 Alakzat tükrözése egy, vagy több egyenesre (G50.1, G51.1).....	168
16.4 A különleges transzformációk programozási szabályai.	170
17 Automatikus geometriai számítások	172
17.1 Letörés és lekerekítés programozása.....	172
17.2 Egyenes megadása irányszögével.	173
17.3 Síkbeli metszéspontszámítások.	175
17.3.1 Két egyenes metszéspontja.....	175
17.3.2 Egyenes és kör metszéspontja.	177
17.3.3 Kör és egyenes metszéspontja.	179
17.3.4 Két kör metszéspontja.	181
17.3.5 A metszéspontszámítások láncolása.	183
18 Esztergáló ciklusok	184

18.1 Egyszerű ciklusok.....	184
18.1.1 A hosszesztergáló ciklus (G77).....	184
18.1.2 Az egyszerű menetvágó ciklus (G78).....	186
18.1.3 Az oldalazó ciklus (G79).....	188
18.1.4 Az egyszerű ciklusok használata.....	190
18.2 Összetett ciklusok.....	191
18.2.1 Nagyoló ciklus (G71).....	191
18.2.2 Homlok nagyoló ciklus (G72).....	200
18.2.3 Minta ismétlő ciklus (G73).....	204
18.2.4 Símitó ciklus (G70).....	208
18.2.5 Homlok beszűrő ciklus (G74).....	209
18.2.6 Beszűrő ciklus (G75).....	211
18.2.7 A menetvágó ciklus (G76).....	213
19 Fúróciklusok.....	219
19.1 A fúróciklusok részletes leírása.....	225
19.1.1 Nagysebességű mélyfúróciklus (G83.1).....	225
19.1.2 Balmenet fúrása kiegyenlítő betéttel (G84.1).....	226
19.1.3 Kiesztergálás automatikus szerszámelhúzással (G86.1).....	227
19.1.4 A ciklusállapot kikapcsolása (G80).....	228
19.1.5 Fúróciklus, kiemelés gyorsmenettel (G81).....	228
19.1.6 Fúróciklus várakozással, kiemelés gyorsmenettel (G82).....	229
19.1.7 Mélyfúróciklus (G83).....	230
19.1.8 Jobbmenet fúrása kiegyenlítő betéttel (G84).....	231
19.1.9 Menetfúrás kiegyenlítő betét nélkül (G84.2, G84.3).....	232
19.1.10 Menetfúrás kiegyenlítő betét nélkül, forgácstöréssel (G84.2, G84.3).....	234
19.1.11 Fúróciklus, kiemelés előtolással (G85).....	237
19.1.12 Fúróciklus, gyorsmeneti kiemelés álló főorsóval (G86).....	238
19.1.13 Kézi működtetés a talpponton/ Kiesztergálás visszafelé (G87).....	239
19.1.14 Fúróciklus, várakozás után kézi működtetés a talpponton (G88).....	241
19.1.15 Fúróciklus várakozással, kiemelés előtolással (G89).....	242
19.2 Megjegyzések a fúróciklusok használatához.....	243
20 Tengelyvezérlő funkciók.....	244
20.1 A sokszögesztergálás.....	244
20.1.1 A sokszögesztergálás működési elve.....	244
20.1.2 A sokszögesztergálás programozása (G51.2, G50.2).....	246
20.2 Fogaskerekek lefejtőmarása (G81.8).....	248
20.3 Tengelyek szinkron mozgása.....	249
20.4 Tengelycsere.....	252
20.5 Tengelyek szuperponált mozgása.....	257
20.6 Tengelyirány váltás.....	259
21 Mérőfunkciók.....	261
21.1 Mérés a maradék út törlésével (G31).....	261
21.2 Maradék út törlése nyomatékhatárra (G31).....	262
21.3 Automatikus szerszámhossz mérés (G36, G37).....	264
22 Biztonsági funkciók.....	266

22.1 Végállás..	266
22.2 Paraméterről állítható/programozható munkatér határolás (G22, G23).	267
22.3 Belülről tiltott terület.	270
22.4 Tiltott tartomány figyelés mozgásindítás előtt.	270
23 A makróprogramozás.	272
23.1 A programnyelv változói.	273
23.1.1 Hivatkozás változókra.	273
23.1.2 A makróváltozók számábrázolása.	274
23.1.3 A lokális változók: #1 – #33.	274
23.1.4 Globális változók.	275
23.1.5 Rendszerváltozók leírásánál használt jelölések.	275
23.1.6 Az üres változó. Konstansok.	276
23.1.7 Az alakrészprogram és a PLC program közötti változók.	277
23.1.8 Az akatrészprogram üzenetei.	278
23.1.9 Idők, munkadarab számlálók.	280
23.1.10 Az automata üzem működését befolyásoló változók.	282
23.1.11 A mondatkeresés és teszt állapotok lekérdezése.	284
23.1.12 Tükrözési állapot.	284
23.1.13 A főprogram száma.	285
23.1.14 Öröklődő információk.	286
23.1.15 Pozícióinformációk.	291
23.1.16 Az aktuális hosszkorrekció értéke.	293
23.1.17 Egyéb pozícióinformációk.	293
23.1.18 Szerszámkorrekciós tár értékei.	294
23.1.19 Munkadarab nullponteltolások.	296
23.1.20 Az orsó és készenléti magazinokban lévő szerszám adatainak kiolvasása.	299
23.1.21 A munkahelyen és a szerelőhelyen lévő paletta adatainak kiolvasása.	301
23.2 A programnyelv utasításai.	302
23.2.1 Az értékadó utasítás.	302
23.2.2 Aritmetikai műveletek.	302
23.2.3 Logikai műveletek.	303
23.2.4 Függvények.	305
23.2.5 Konverziós utasítások.	306
23.2.6 Összetett aritmetikai műveletek végrehajtási sorrendje.	306
23.2.7 Feltételes kifejezések.	307
23.2.8 Feltétel nélküli elágazás.	307
23.2.9 Feltételes elágazás.	308
23.2.10 Feltételes utasítás.	308
23.2.11 Ciklusszervezés.	308
23.2.12 Indirekt tengelyhivatkozások.	312
23.2.13 Adatkiadási parancsok.	312
23.3 Makrók, rendszermakrók, rendszerprogramok hívása.	319
23.3.1 Az egyszerű makróhívás (G65).	322
23.3.2 Öröklődő makróhívás minden mozgásparancs után: (G66).	324
23.3.3 Öröklődő makróhívás minden mondatból: (G66.1).	325
23.3.4 Rendszermakró hívás paraméteren megadott G kódra.	327
23.3.5 Rendszermakró hívás paraméteren megadott M kódra.	329
23.3.6 Rendszerprogram hívás paraméteren megadott M kódra.	330

23.3.7 Rendszerprogram hívás paraméteren engedélyezett A, B, C, S, T kódra...	332
23.3.8 Rendszerprogram hívás paraméteren megadott ASCII kódra..	333
23.3.9 Makrók és alprogramok mondatainak kijelzése automata üzemmódban....	334
23.4 A megszakítási makró.	335
23.5 NC és makró utasítások. A makromondatok végrehajtása..	339
23.6 Üregmaró makróciklus..	341
24 Paraméterek írása és olvasása.	345
24.1 Paraméterek írása alkatrészprogramból (G10 L52).	346
24.2 Paraméterek olvasása alkatrészprogramból (PRM).	347
Jegyzetek ..	349

19.11.14

© Copyright NCT November 14, 2019

E leírás tartalmára minden kiadói jog fenntartva. Utánnymáshoz – kivonatosan is – engedélyünk megszerzése szükséges.

A leírást a legnagyobb körültekintéssel állítottuk össze és gondosan ellenőriztük, azonban az esetleges hibákért vagy téves adatokért *és az ebből eredő károkért felelősséget nem vállalunk.*

1 Bevezetés

1.1 Az alkatrészprogram

Az alkatrészprogram olyan utasítások halmaza, amelyeket a vezérlés értelmezni képes és amelyek alapján a gép működését irányítja.

Az alkatrészprogram mondatokból áll. A mondatokat szavak alkotják.

Szó: Cím és Adat

A szó két részből tevődik össze: címből és adatból. A cím egy vagy több karakter, az adat pedig numerikus érték, amelynek lehet egész és tizedes értéke is. Bizonyos címek kaphatnak előjelet, illetve I operátort. A címeket megadhatjuk a programban kis- és nagybetűvel is, mindkét változatot elfogadja a vezérlő.

Címek:

Címek	Jelentés	Értékhatár
O	programszám	0001 - 9999
/	opcionális mondat	1 - 9
N	mondatszám	1 - 99999999
G	előkészítő funkció	*
X, Y, Z, U, V, W	hosszkoordináták	I, -, *
A, B, C,	szögkoordináta	I, -, *
R	körsugár, segédadat	I, -, *
I, J, K	kör középpont koordináta, segédkoordináta	-, *
E	segédkoordináta	-, *
F	előtolás	*
S	orsó fordulatszám	*
M	vegyes funkció	1 - 99999999
T	szerszámszám/korrekció száma	1 - 99999999
D	korrekció száma	1 - 999
L	ismétlési szám	1 - 99999999
P	segédadat, várakozási idő	*
Q	segédadat	*
,C	letörés szárhossza	-, *
,R	lekerekítés sugara	-, *
,A	egyenes irányszöge	-, *

Azoknál a címeknél, amelyeknél a * jel látható az értékhatár oszlopban, az adat tizedes értéket is felvehet.

Azoknál a címeknél, ahol az I jel és a - jel látható, a címre adható inkrementális operátor illetve

előjel.

Nem jelezzük ki, és nem tároljuk a + jelet.

Mondat

A mondat szavakból tevődik össze.

A mondatokat a tárban LF (Line Feed) karakter választja el egymástól. A mondatokban nem kötelező a mondatszám használata.

Főprogram és alprogram

Az alkatrészprogramokat két fő részre lehet osztani:

főprogramokra, és
alprogramokra.

Az alkatrész megmunkálását a főprogram írja le. Ha a megmunkálás során ismétlődő mintákat kell a főprogramban különböző helyeken megmunkálni, akkor ezeket a programszakaszokat nem kell ismét leírni a főprogramban, hanem alprogramot (szubrutint) lehet rá szervezni, amely tetszőleges helyről hívható, akár egy másik alprogramból is. Az alprogramból vissza lehet térni a hívó programba.

Programformátum a tárban

A tárban elhelyezkedő *alkatrészprogramok ASCII kódolású szövegfájlok.*

1.2 A csatorna

Általában egy gépen egyidőben egy szerszám dolgozik, amelynek a mozgását a vezérlőn keresztül az alkatrészprogram vezérli.

Ha ugyanaz a vezérlő kettő, vagy több, egymás mozgásától független szerszám pályáját vezérli egyidőben két, vagy több alkatrészprogram végrehajtásával, többcsatornás működésről beszélünk.

Minden csatornához ki lehet jelölni egy-egy alkatrészprogramot végrehajtásra.

Ha egy gépen ugyan csak egy szerszám dolgozik egyidőben, de például ki van építve a gépre egy munkadarabadagoló, ahol a munkadarab mozgásokat programozni kell, miközben a szerszám forgácsol, akkor is többcsatornás működés szükséges.

A programok azon pontjaira, ahol a ***szerszám pályák mozgásainak össze kell egymást várniuk, szinkronizációs pontokat*** lehet programozni.

Minden csatornához külön munkadarab nullponttáblázat, szerszámkorrekciós táblázat és makrováltozók tartoznak. A szerszámkorrekciós táblázat és a makrováltozók egy paraméteren kijelölhető része közössé tehetők az összes csatorna részére.

Az NCT2xx vezérlők alap kivitelben egycsatornásak, opcionálisan, többcsatornás működésre is képesek. Maximum 8 csatorna építhető ki vezérlőnként.

A vezérlőben minden csatornában paraméteren meg lehet határozni a csatorna működési módját.

Ez az NCT2xx vezérlőkben ***kétféle lehet: eszterga, vagy maró csatorna.*** Egy vezérlőn belül a csatornák vegyesek is lehetnek. Például:

1. csatorna: eszterga
2. csatorna: eszterga
3. csatorna: maró

A továbbiakban ez a leírás az eszterga csatorna programozási leírását tartalmazza.

Ha az adott vezérlőben maró csatorna is kiépítésre került, annak a programozási leírását az NCT2xxM Marógép és megmunkáló központ vezérlő programozási leírás című könyv tartalmazza.

1.3 Alapfogalmak

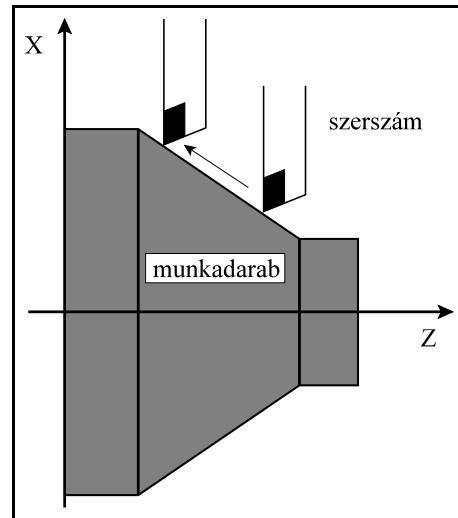
Az interpoláció

A vezérlés a megmunkálás során a szerszámot egyenes- és körpálya mentén képes mozgatni. Ezt a tevékenységet a továbbiakban interpolációnak nevezzük.

Szerszámmozgás egyenes mentén:

program:

```
G01 X__ Z__
```

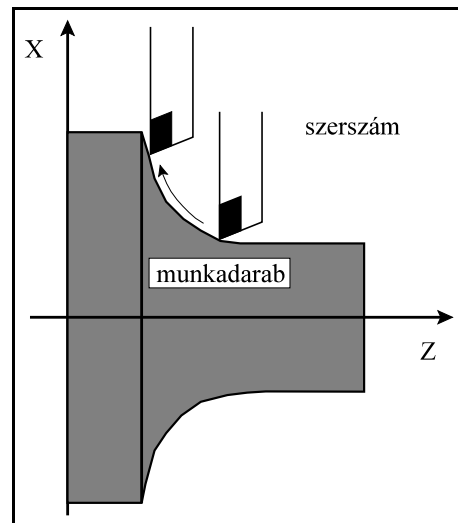


1.3-1 ábra

Szerszámmozgás körív mentén:

program:

```
G02 X__ Z__ R__
```



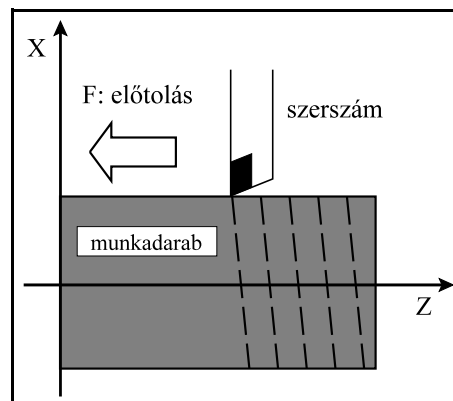
1.3-2 ábra

Előkészítő funkciók (G kódok)

Egy adott mondat által végrehajtandó tevékenység típusát az előkészítő funkciók, vagy más néven G kódok segítségével írjuk le. Például: a G01 kód egyenes interpolációt vezet be.

Előtolás

Előtolásnak nevezzük a szerszámnak a munkadarabhoz viszonyított sebességét forgácsolás közben. Programban F címen és egy számértékkel adhatjuk meg a kívánt előtolást. Például: F2 jelentése 2 mm/fordulat.



1.3-3 ábra

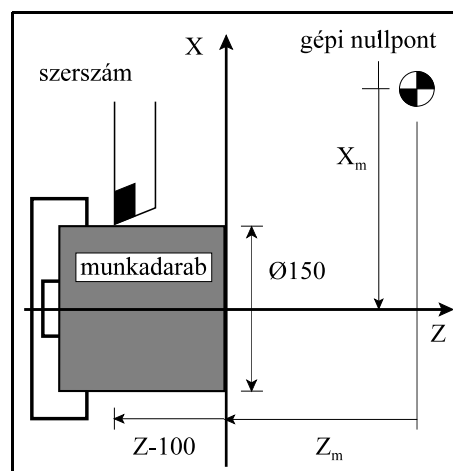
Referenciapont

A referenciapont a szerszámgépen egy a gyártó által kijelölt, a gépre jellemző fix pont. A gép bekapcsolása után a szánokat referenciapontra kell küldeni, ha nincs a gépen abszolút mérőeszköz. A referenciapont felvételére külön üzemmód van a vezérlőbe beépítve. Ezután a vezérlés már abszolút pozícióparancsokat is végre tud hajtani.

Koordinátarendszer

Az alkatrész rajzán feltüntetett méretek az alkatrész egy adott pontjához képest értendők. Ez a pont a munkadarabkoordinátarendszer nullpontja. Az alkatrészprogramba ezeket a méretadatokat kell beírni a koordinátacímekre. Például: X150 Z-100 jelentése: a munkadarab koordináta-rendszer 150; és -100 mm koordinátájú pontja X és Z irányban.

Ahhoz, hogy a vezérlés a programozott koordinátaadatokat értelmezni tudja meg kell adni a gépi nullpont és a munkadarab nullpont közti távolságot. Ez a munkadarab nullpont bemérésével történik.



1.3-4 ábra

Abszolút koordinátamegadás

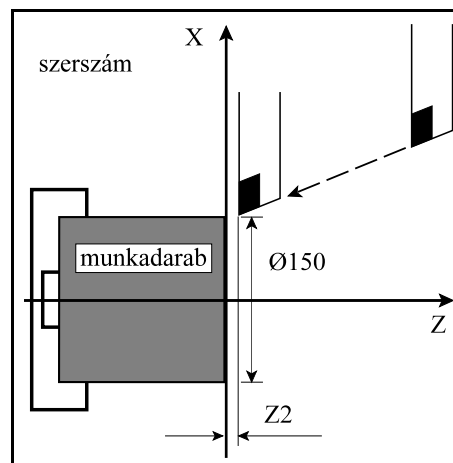
Abszolút koordinátamegadás esetén a szerszám a koordinátarendszer kezdőpontjától számított távolságra mozog, azaz a koordinátán megadott pozíciójú pontra.

Az abszolút adatmegadás kódja: G90.

A

```
G90 X150 Z2
```

utasítással a fenti pozíciójú pontra mozgatja a szerszámot, bárhol is állt a parancskiadás előtt.



1.3-5 ábra

Növekményes (inkrementális) koordinátamegadás

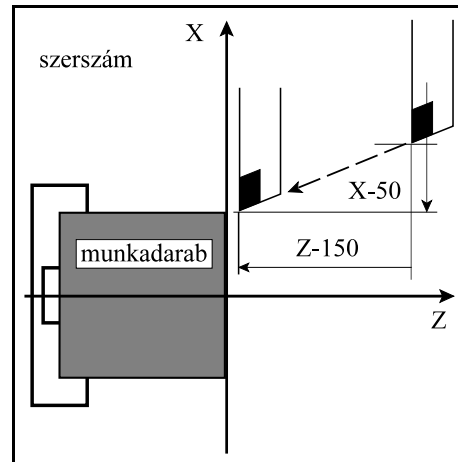
Növekményes koordinátamegadás esetén a vezérlés a koordinátaadatokat úgy értelmezi, hogy a szerszám a pillanatnyi pozíciótól számított távolságra mozogjon.

Az inkrementális adatmegadás kódja: G91. A G91 kód az összes koordinátaértékre vonatkozik.

A

```
G91 X-50 Z-125
```

utasítással az előző pozíciótól a fenti távolságra mozgatja el a szerszámot.



1.3-6 ábra

Átmérőben történő programozás

Az X irányú méretet paraméter beállítás alapján lehet átmérőben is programozni.

Öröklődő funkciók

A programnyelvben bizonyos utasítások hatása, vagy értékük nagysága öröklődik mindaddig, amíg ellenkező értelmű parancsot nem adunk ki, vagy más értéket nem adunk a megfelelő funkciónak. Például: Az

```
N15 G90 G1 X20 Z30 F0.2
N16 X30
N17 Z100
```

programrészletben az N15 mondatban felvett G90 (abszolút adatmegadás) és a G1 (lineáris interpoláció) állapota, illetve F (előtolás) értéke öröklődik az N16-os és N17-es mondatokban. Így nincs szükség ezeket a kódokat mondatról-mondatra megadni.

Nem öröklődő (egylövetű) funkciók

Bizonyos funkciók hatása, vagy adatok értéke csak az adott mondatban érvényes. Ezeket a funkciókat nem öröklődő, vagy egylövetű funkcióknak nevezzük.

Főorsó fordulatszám parancs

A főorsó fordulatszámot S címen lehet megadni. Ezt szokás még S funkciónak is nevezni. Az S1500 utasítás azt mondja meg, hogy a főorsó 1500 ford/perces fordulatszámmal forogjon.

Konstans vágósebesség számítás

A vezérlés automatikusan úgy változtatja a főorsó fordulatszámát az átmérő függvényében, hogy a szerszám hegyének a munkadarab felületéhez képesti sebessége állandó legyen. Ezt nevezzük konstans vágósebesség számításnak.

Szerszámszám

A megmunkálás során különböző szerszámokkal kell a különböző forgácsolási műveleteket elvégezni. A szerszámokat számokkal különböztetjük meg egymástól. A szerszámokra T kóddal hivatkozhatunk. A T kód első két számjegye a szerszám kódja (vagyis hányadik pozícióban található a revolverfejben), a T kód második két számjegye pedig a kiválasztott szerszámhoz tartozó korrekciós csoport száma. A programban a

```
T0212
```

utasítás azt jelenti, hogy a 02-es szerszámot választottuk és a 12-es korrekciós csoportot rendeltük hozzá.

Vegyes funkciók

A megmunkálás során számos ki-, bekapcsolási műveletet kell elvégezni. Például: elindítani a főorsót, bekapcsolni a hűtővizet. Ezeket a műveleteket a vegyes vagy M funkciók segítségével lehet elvégezni. Például: az

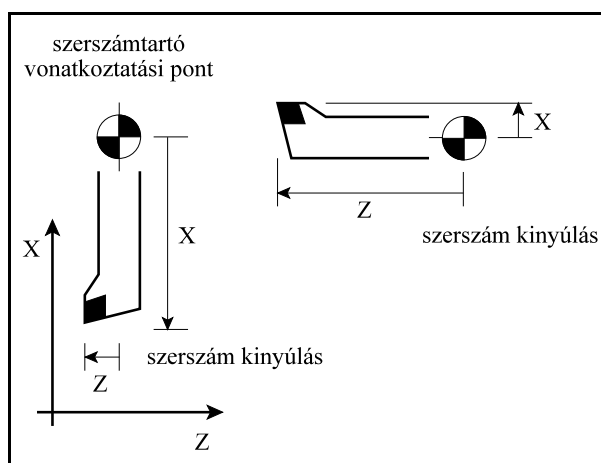
M3 M8

utasításokban M3 jelentése: főorsó forgás óramutatóval megegyező irányba, M8 jelentése pedig: kapcsold be a hűtővizet.

Hosszkorrekció

A megmunkálás során különböző hosszúságú szerszámokkal végezzük a különböző műveleteket. Ugyanazt a műveletet viszont egy nagyobb széria gyártása esetén, például a szerszám törése miatt, szintén különböző hosszúságú szerszámmal kell végezni. Annak érdekében, hogy az alkatrészprogramban leírt mozgások függetlenek legyenek a szerszám hosszától, azaz kinyúlásától, a vezérléssel közölni kell a különböző szerszámhosszakat. Ehhez a szerszámok hosszát be kell mérni. Ha

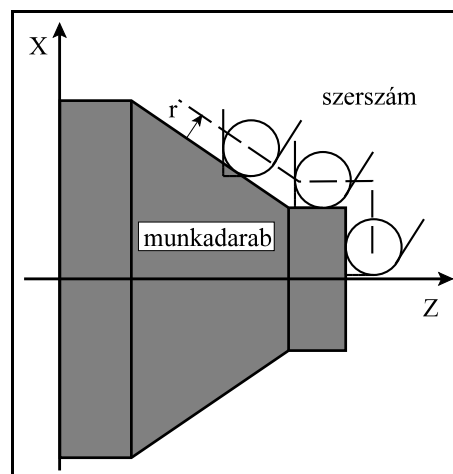
a programban azt akarjuk, hogy a szerszám csúcsa mozogjon a megadott pontra, le kell hívni annak a hosszadatnak az értékét, amelyet bemérés során megadtunk. Ez a T kód második két helyiértékén történik. Ettől kezdve a szerszám csúcsát mozgatja a vezérlő a megadott pontra.



1.3-7 ábra

Sugárkorrekció

Egy kontúr esztergálásánál, ha nem a tengelyekkel párhuzamos mozgást végez a szerszám, akkor kapunk pontos méretet, ha nem a szerszám hegyét vezetjük a kontúron, hanem a szerszámsugar közép pontját vezetjük a kontúrra merőlegesen, attól sugárnyi (r) távolságra. Annak érdekében, hogy a programban ne a szerszámközpont pályáját kelljen leírni, figyelembe véve a szerszámok sugarát, hanem a darab tényleges kontúradatait, be kell vezetni a sugárkorrekciót. A programban a T címen lehívott korrekciós csoportban kell megadni a szerszám sugarát.



1.3-8 ábra

2 Vezérelt tengelyek

Tengelyek száma alapkiépítésben	2 tengely
Bővítőtengelyek száma	max. 14 további tengely ugyanabban a csatornában
Tengelyek maximális száma	32 tengely összesen, több csatornában

2.1 A tengelyek neve és száma

A vezérelt **tengelyek elnevezését** a paramétertárban lehet definiálni. Itt ki lehet jelölni, hogy melyik fizikai tengely milyen címre mozogjon.

Alapkiépítésben egy esztergavezérlőben a tengelyek nevei: X és Z.

Ezek a tengelyek az N0103 Axis to Plane paraméteren **főtengelyek**ként kerülnek beállításra.

A bővítőtengelyek elnevezése a tengely típusától függ.

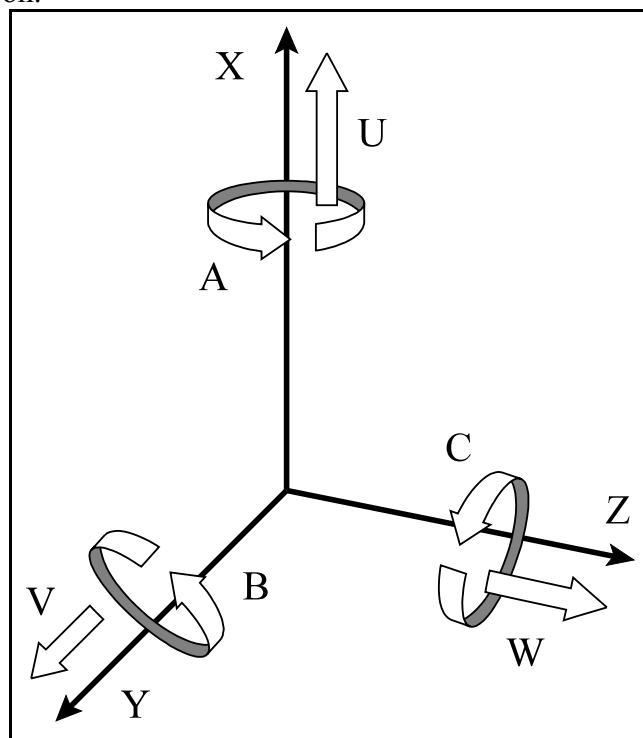
Ha bővítőtengelyként az esztergán az Y tengely is ki van építve az Y tengelyt is fő-tengelyként kell beállítani az N0103 Axis to Plane paraméteren.

A további, lineáris mozgást végző bővítőtengelyek lehetséges elnevezése: U, V és W. Ha ezek a tengelyek valamelyik főiránnyal párhuzamosak, akkor az X tengellyel **párhuzamos bővítőtengely** neve U, az Y-nal párhuzamos neve V, és a Z-vel párhuzamos neve W.

Az, hogy egy lineáris bővítőtengely párhuzamos-e egy alaptengellyel, az N0103 Axis to Plane paraméteren kerül beállításra.

A forgómozgást végző tengelyek nevei: A,

B, és C. Az X iránnyal párhuzamos tengelyű forgótengely neve A, az Y-nal párhuzamos neve B, és a Z-vel párhuzamos pedig C. Azt, hogy egy tengely forgó tengely-e, az N0106 Axis Properties paraméter ROT=1 beállításával adható meg.



2.1-1 ábra

Az egyes **tengelyeket** a vezérlő belül a **számuk alapján** tartja nyilván. A tengelyneveket is a tengelyszámhoz kell rendelni. A szokványos beállítás: X az 1. tengely, Y a 2. tengely, Z a 3. tengely. Egyes programutasításokban, pl. tengelypozíciót lekérdező makrováltozóra való hivatkozás esetén, nem a tengelynevet, hanem a tengelyszámot kell megadni.

A tengelyek számozását kérje a gép építőjétől.

2.2 Kiterjesztett tengelynevek

Bonyolult, soktengelyes gépeken nem elég a fenti max. 9 betű az összes tengely elnevezésére. Ezért be lettek vezetve a kiterjesztett tengelynevek, amikor nem egy betűn, hanem **max. 3 karakteren adható meg egy tengely elnevezése.**

Az első karakter kötelezően X, Y, Z, U, V, W, A, B, vagy C kell, hogy legyen, amit az N0100 Axis Name1 paraméteren adhatunk meg. Ezt a paramétert akkor is ki kell tölteni, ha csak egykarakteres tengelyneveket használunk.

A 2., illetve a 3. karakternevet az N0101 Axis Name2 és az N0102 Axis Name3 paraméteren adhatjuk meg. Ezek lehetnek az angol ABC betűi: A, B, C, D, ... Y, Z, illetve számok: 0, 1, 2, ..., 9. Ha a 2., vagy a 3. tengelynevet nem használjuk a paraméterek értéke 0.

Így megadhatunk XDE tengelynevet is például, de használhatjuk az Z1 és Z2 tengelynevet is. Ha a tengelynév betűre végződik, a hozzá tartozó értéket mellé írhatjuk. Az

XDE127.81

jelentése: az XDE tengely menjen a 127.81 pozícióra.

Ha a tengelynév számra végződik a tengelynév után mindig = jelet kell írni. A

Z1=87.257

jelentése: a Z1 tengely menjen a 87.257 pozícióra.

Természetesen a kiterjesztett tengelynevek használatakor is meg kell adni az N0103 Axis to Plane paraméteren, hogy melyek a fő-, vagy alaptengelyek, illetve melyek a párhuzamos tengelyek.

A továbbiakban a leírásban általában az egykarakteres tengelyneveket használjuk.

2.3 Tengelyek csatornákhöz rendelése

A gép építője az egyes tengelyeket, paraméterek segítségével, hozzárendeli a különböző csatornákhöz. Ezek a paraméterbeállítások a bekapcsolás utáni helyzetet jelentik.

A **tengelyeket** mindig a **számuk alapján** rendeljük az egyes csatornákhöz. A **tengelyszám** tehát a vezérlőn belül **globális**, míg a **tengelynevek lokálisak**, az egyes csatornákhöz vannak rendelve. Egy csatornán belül, természetesen, nem lehetnek azonos nevű tengelyek, viszont különböző csatornáknál igen. Egy csatornán belül maximum 16 tengelyt kezel a vezérlő.

Például:

1. csatorna:

- 1. tengely: X
- 2. tengely: Y
- 3. tengely: Z
- 4. tengely: C

2. csatorna:

- 5. tengely: X
- 6. tengely: Y
- 7. tengely: Z
- 8. tengely: C

A megmunkálás során szükség lehet arra, hogy egy vagy több tengelyt egy másik csatornában használjunk. Ilyenkor két tengelyt két csatorna között fel lehet cserélni, vagy egy tengelyt át lehet adni egy másik csatornának. A csere a tengelyszám alapján történik. A csere után a tengely neve maradhat, de meg is változhat.

A tengelyek cseréjét a gép építője a PLC programon keresztül valósítja meg, pl. M funkció segítségével. Ezek leírását mindig az adott gép gépkönyve tartalmazza.

2.4 A tengelyek mértékrendszere, és a pozíciókijelzés pontossága

Koordinátaadatokat maximum **15 számjegy pontossággal** lehet megadni. A tizedespontot csak akkor kell kitenni, ha nem egész koordinátájú pontra kell pozicionálni. A koordinátaadatoknak lehet előjele is. A “+” előjelet nem kell kitenni a szám elé.

A programban a hosszkoordináták adatait meg lehet adni **mm**-ben és **inch**-ben. Ez a **bemeneti mértékrendszer**. A bemeneti mértékrendszert a programból lehet G kóddal (G21/G20) kiválasztani.

A gépre felszerelt útmérő eszköz mérheti a pozíciót mm-ben és inchben is. Az útmérő eszköz határozza meg a **kimeneti mértékrendszert**, amit a vezérlőnek az N0104 Unit of Measure paraméter IND bitjén kell megadni. Egy gépen belül nem lehet a kimeneti mértékrendszereket a tengelyek között keverni.

Amennyiben a be- és kimeneti mértékrendszer különböző, az átváltást a vezérlő automatikusan végzi.

A forgó tengelyek mértékrendszere mindig **fok**. **A forgó tengelyeket az N0106 Axis Properties paraméter ROT=1 beállításával jelölhetjük ki. Ennek a paraméternek a helyes beállítása azért fontos, mert az így kijelölt tengelyekre a vezérlő nem végez inch/metrikus átváltást.**

Azt, hogy a **pozíciókijelzés** hány tizedesjegy **pontossággal** történjék az N0105 Increment System paraméteren állíthatjuk be.

A rendszer belső pozícióábrázolása az Increment System paraméter értékétől független:

Hossztengelyeknél metrikus mérés esetén: 10^{-6} mm,

Hossztengelyeknél inches mérés esetén: 10^{-7} inch,

Körtengelyek esetén: 10^{-6} fok

pontosságú.

Jel	tengely	Kijelzés pontossága a mértékrendszer függvényében	
		G21 metrikus	G20 inch
ISA	hossz	0.01 mm	0.001 in
	forgó	0.01 fok	0.01 fok
ISB	hossz	0.001 mm	0.0001 in
	forgó	0.001 fok	0.001 fok
ISC	hossz	0.0001 mm	0.00001 in
	forgó	0.0001 fok	0.0001 fok
ISD	hossz	0.00001 mm	0.000001 in
	forgó	0.00001 fok	0.00001 fok
ISE	hossz	0.000001 mm	0.0000001 in
	forgó	0.000001 fok	0.000001 fok

Tetszőleges tengelyt kijelölhetünk az N0106 Axis Properties paraméter DIA=1 beállításával **útmérőben** történő adatbevitelre és pozíciókijelzésre.

3 Előkészítő funkciók (G kódok)

A G cím és az azt követő szám határozza meg az adott mondatban a parancs jellegét.

Vannak úgynevezett **egylövetű G kódok**, amelyek hatása csak az adott mondatban érvényes, és vannak úgy nevezett modális, vagy **öröklődő G funkciók**, amelyek hatása addig tart, amíg egy másik G kóddal ki nem kapcsoljuk, vagy meg nem változtatjuk ezt a hatást.

Az egylövetű G kódok a 0-ás csoportba tartoznak. Az öröklődő G kódok közül, amelyek egymásra hatnak egy 0-tól különböző csoportszámot kapnak.

Egy mondatba több G kód is írható, azzal a megkötéssel, hogy azonos csoportba tartozó funkciók közül csoportonként csak egy szerepelhet.

A **vezető nullát nem kell beírni a kódba**, de a vezérlő elfogadja azt. Pl: írhatunk a programba G01-et és G1-et is.

A következő táblázat tartalmazza a vezérlés által értelmezett G kódokat, csoportszámukat és funkciójukat.

G kód	Csoport	Funkció	Oldal
G0*	01	pozicionálás	24
G1*		egyenes interpoláció	26
G2		kör és spirális (síkbeli, hengeres, kúpos) interpoláció óramutató járásával egyező irányba	28, 33, 35, 38
G3		kör és spirális (síkbeli, hengeres, kúpos) interpoláció óramutató járásával ellentétes irányba	28, 33, 35, 38
G4	00	várakozás	75
G9		pontos megállás az adott mondatban	62
G7.1		hengerinterpoláció	46
G10		programozott adatbeadás	83, 84, 120,
G11		programozott adatbeadás kikapcsolása	
G10.9		sugárban/átmérőben programozás átváltása programból	51
G12.1	21	polárkoordináta interpoláció be	42
G13.1*		polárkoordináta interpoláció ki	42
G15*	17	adatmegadás polárkoordinátákkal ki	52
G16		adatmegadás polárkoordinátákkal be	52

G kód	Csoport	Funkció	Oldal
G17*	02	$X_p Y_p$ sík választása	87
G18*		$Z_p X_p$ sík választása	87
G19*		$Y_p Z_p$ sík választása	87
G20*	06	inches adatmegadás	50
G21*		metrikus adatmegadás	50
G22*	04	munkatér határolás be	267
G23*		munkatér határolás ki	267
G28	00	programozott referenciapont felvétel	77
G30		második, harmadik és negyedik referenciapontra állás	78
G31		mérés a maradék út törlésével	261
G33	01	menetvágás	40
G36	00	automatikus szerszámhosszmérés X irányban	264
G37		automatikus szerszámhosszmérés Z irányban	264
G38		sugárkorrekciós vektor megtartása	152
G39		sarokív sugárkorrekcióval	153
G40*	07	szerszámsugár-korrekció számítás kikapcsolása	128
G41		szerszámsugár-korrekció számítás balról	128
G42		szerszámsugár-korrekció számítás jobbról	128
G43	08	maró hosszkorrekció be +	124
G44		maró hosszkorrekció be -	124
G43.7*		szerszámhossz-korrekció figyelembe vétele G kódra	123
G49*		hosszkorrekció ki	123
G50*	11	léptékezés ki	165
G51		léptékezés be	165
G50.1*	22	tükrözés kikapcsolása	168
G51.1		tükrözés bekapcsolása	168
G51.2	31	sokszögesztergálás be	246
G50.2*		sokszögesztergálás ki	246
G52	00	koordinátaeltolás	86
G53		pozicionálás a gép koordinátarendszerében	80

G kód	Csoport	Funkció	Oldal
G54*	14	első munkadarab koordinátarendszer választása	82
G55		második munkadarab koordinátarendszer választása	82
G56		harmadik munkadarab koordinátarendszer választása	82
G57		negyedik munkadarab koordinátarendszer választása	82
G58		ötödik munkadarab koordinátarendszer választása	82
G59		hatodik munkadarab koordinátarendszer választása	82
G54.1		kiterjesztett munkadarab koordinátarendszer választása	84
G61	15	pontos megállás üzemmód	63
G62		előtoláscsökkentés sarkoknál	63
G63		override tiltás	63
G64*		folyamatos forgácsolás üzemmód	63
G65	00	egyszerű makrohívás	322
G66	12	öröklődő makrohívás minden mozgásparancs után	324
G66.1		öröklődő makrohívás minden mondatból	325
G67*		öröklődő makrohívás törlése	324
G68	16	alakzat síkbeli forgatása adott pont körül	163
G69*		síkbeli forgatás ki	163
G70	00	símító ciklus	208
G71		nagyoló ciklus	191
G72		homlok nagyoló ciklus	200
G73		minta ismétlő ciklus	204
G74		homlok beszúró ciklus	209
G75		beszúró ciklus	211
G76		menetvágó ciklus	213
G77		01	hosszesztergáló ciklus
G78	egyszerű menetvágó ciklus		186
G79	oldalazó ciklus		188

G kód	Csoport	Funkció	Oldal	
G83.1	09	nagysebességű mélyfűrőciklus	225	
G84.1		balmenet fúrása kiegyenlítő betéttel	226	
G86.1		kiesztergálás automatikus szerszámelhúzással	227	
G80*		ciklusállapot kikapcsolása	228	
G81		fűrőciklus, kiemelés gyorsmenettel	228	
G82		fűrőciklus várakozással, kiemelés gyorsmenettel	229	
G83		mélyfűrőciklus	230	
G84		jobbmenet fúrása kiegyenlítő betéttel	231	
G84.2		jobbmenet fúrása kiegyenlítőbetét nélkül jobbmenet fúrása kiegyenlítőbetét nélkül forgácstöréssel	232	
G84.3		balmenet fúrása kiegyenlítőbetét nélkül balmenet fúrása kiegyenlítőbetét nélkül forgácstöréssel	232	
G85		fűrőciklus, kiemelés előtolással	237	
G86		fűrőciklus, gyorsmeneti kiemelés álló főorsónál	238	
G87		kiesztergálás visszafelé, automatikus/kézi szerszámelhúzással	239	
G88		fűrőciklus, kézi működtetés a talpponton	241	
G89		fűrőciklus, talpponton várakozás, kiemelés előtolással	242	
G80.8		34	elektronikus hajtómű ki	248
G81.8			elektronikus hajtómű be	248
G90*	03	abszolút méretmegadás	49	
G91*		növekményes méretmegadás	49	
G92	00	új munkadarab koordinátarendszer létrehozása	85	
G94*	05	percenkénti előtolás	62	
G95*		fordulatonkénti előtolás	62	
G96	13	konstans vágósebesség számítás be	93	
G97*		konstans vágósebesség számítás ki	93	
G98*	10	visszatérés fűrőciklusból a kiindulási pontra	220	
G99		visszatérés fűrőciklusból az R (megközelítési) pontra	220	

Bekapcsolás utáni alaphelyzet

Egy csoporton belül a *-gal jelölt G kódok azt az állapotot jelentik, amit a vezérlés **bekapcsolás után** felvesz.

Ahol egy csoporton belül több G kód után is * jel található ott az N1300 DefaultG1 és az N1301 DefaultG2 paraméter alapján lehet kiválasztani, melyik legyen érvényes **bekapcsolás után**. Ezek a következők:

G00, G01;
G17, G18, G19;
G20, G21;
G22, G23;
G90, G91;
G94, G95.

Reset gomb nyomása, illetve program vége utáni alaphelyzet

A **reset** gomb megnyomásának, vagy a **program végének** (M2, M30) hatására a vezérlő, ha az N1301 DefaultG2 paraméter CLR bitje

=0: feltétel nélkül, a G kód táblázatban *-gal jelölt állapotot veszi fel, illetve az N1300 DefaultG1 és az N1301 DefaultG2 paraméteren beállított, bekapcsolás utáni alaphelyzetre állítja vissza a G kód értékeket,

=1: a CLR G Table1, 2, 3, 4, 5 paraméteren megadott értékek alapján G kód csoportonként alaphelyzetbe áll, vagy változatlanul hagyja az öröklött értéket.

Ha a CLR G Table1, 2, 3, 4, 5 paraméteren a G kód csoporthoz tartozó Cnn bit (ahol nn a G kód csoportszáma):

=0: a megfelelő G kód csoportot alaphelyzetbe hozza,

=1: a megfelelő G kód csoportot hagyja a kialakult, megöröklött állapotban.

4 Az interpoláció

4.1 A pozícionálás (G0)

A G0 pozícionálás parancs a szerszámot az összes, a mondatban programozott tengely mentén a megadott pontra mozgatja.

A mozgás gyorsmenettel történik. A gyorsmenet nagyságát a gép építője határozza meg tengelyenként, paraméteren, azt programból nem lehet állítani.

Abszolút adatmegadás esetén az aktuális munkadarab koordinátarendszerben mozog a megadott pozíciójú pontra.

Inkrementális adatmegadás esetén a szerszám pillanatnyi pozíciójától számítva lépi le a megadott távolságot.

A mondat formátuma:

G0 v

ahol v a mondatban megadott koordináták. A v jelölés itt (és a továbbiakban) az adott csatornában használt összes vezérelt tengelyre vonatkozik. A csatorna összes tengelye mentén pozícionálhatunk egyidőben.

A G0 helyett megadható G00 is.

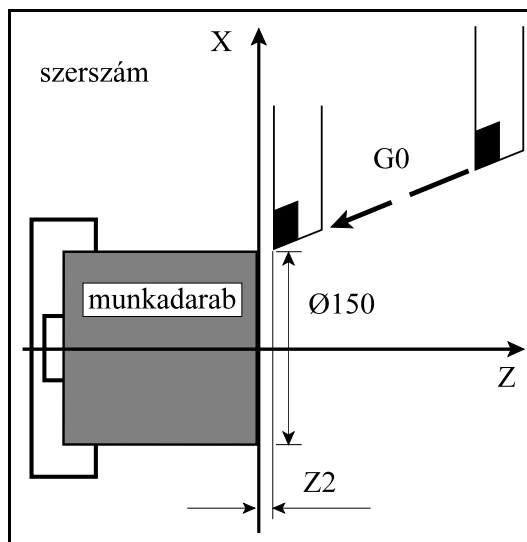
Példa:

G0 X150 Z2

A mondatban megadható más G kód és funkció is.

Pl:

G0 G90 X150 Z2 S2000 M3



4.1-1 ábra

A G0 öröklődő kód, addig érvényes, amíg egy másik, interpolációs parancs át nem írja.

Például:

G0 X150 Z2

X20 Z1 (pozícionáló mondat, mert G0 öröklődik)

Bekapcsolás után az N1300 DefaultG1 paraméter G01=0 bitállása esetén G0 kód van érvényben.

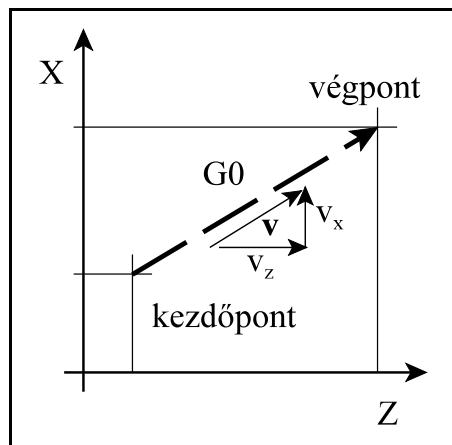
4.1.1 Pozícionálás lineáris interpolációval

Több tengely egyidejű mozgatása esetén a vezérlő pozícionáláskor a szerszámot a kezdőpontot a végponttal összekötő egyenes mentén mozgatja, az N0421 Acc Contr paraméter ROL=0 bitállása esetén.

A sebességvektor eredőjét (v) úgy számítja ki a vezérlő, hogy a pozícionálás minimális idő alatt történjék, és a sebesség egyik tengelyen se lépje túl az arra a tengelyre beállított gyorsmeneti értéket.

A mozgás befejezése után a vezérlés ellenőrzi a "pozícióban" jelet, az N1337 Execution Config paraméter PCH=1 bitállása esetén.

A "pozícióban" jelre az N1340 Inpos Timeout paraméte-



4.1.1-1 ábra

ren beállított ideig vár, ha ezután sem érkezik meg a jel 2501 Pozícióhiba jelzést ad.

A pozíciótól mért legnagyobb, még elfogadható eltérést az N0516 Inpos paraméteren lehet tengelyenként megadni.

A pozícióellenőrzést csak indokolt esetben állítsuk be, különben a végrehajtási idő indokolatlanul megnőhet. Például a

```
G0 X60
Z1
X56
```

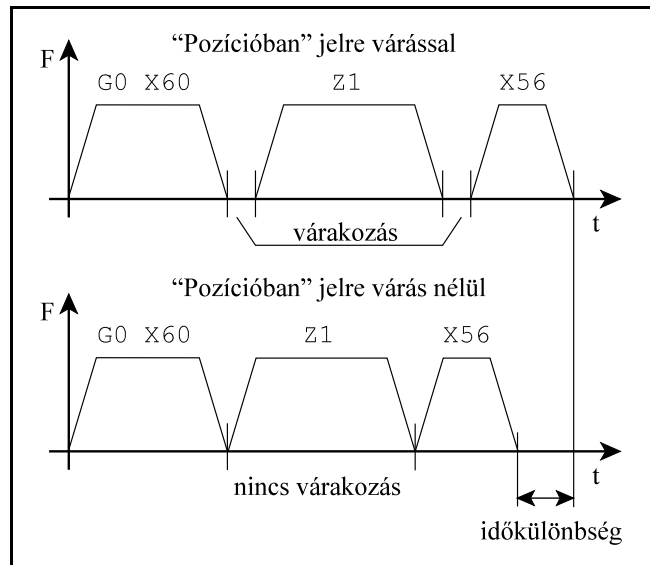
programrészletet végrehajtva a mellékelt ábrán látható a két időtartam közötti különbség.

A vezérlő **mindig végez pozícióellenőrzést** a mondat végpontján

G61 (pontos megállás üzemmód) állapotban, vagy abban a pozicionáló mondatban ahova

G9 (pontos megállás)

kódot írtunk, még a PCH=0 (nincs pozícióellenőrzés) paraméterállásnál is.



4.1.1-2 ábra

4.1.2 Pozicionálás a gyorsmeneti mozgások átlapolásával

Az egymást követő, különböző tengelyeken végzett pozicionálások tovább gyorsíthatók a pozicionáló mondatok mozgásának átlapolásával. Ez azt jelenti, hogy miközben egy pozicionáló mondatban az egyik tengely a végponti pozícióra lassít, a következő pozicionáló mondatban részt vevő másik tengely már elkezd gyorsítani.

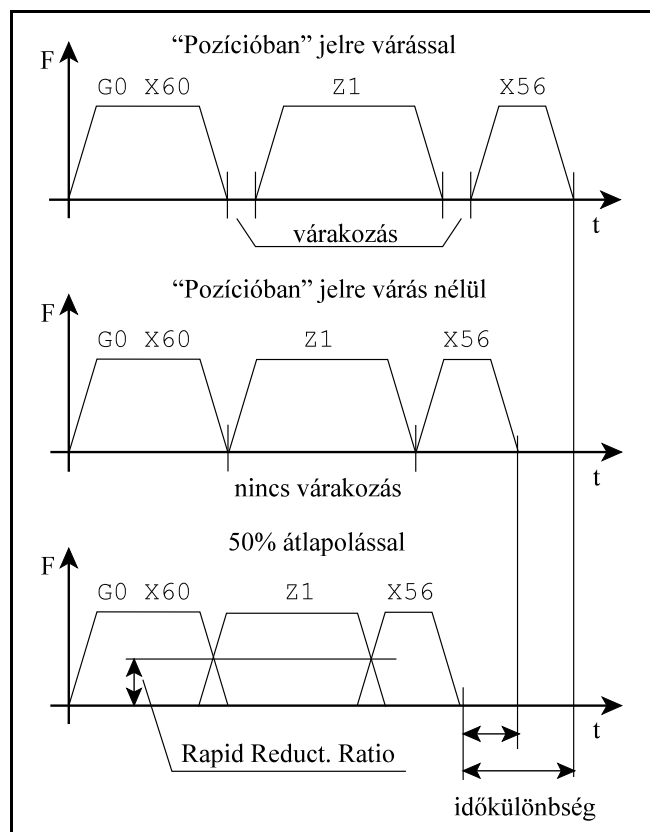
Az átlapolást az N0421 Acc Contr paraméter ROL=1 bitállása kapcsolja be.

Azt, hogy az előző mondat lassítási szakaszában a sebesség hány százalékára való lassítás után indul meg a következő mondat, az N0422 Rapid Reduct. Ratio paraméteren lehet beállítani százalékban.

Az előző fejezetben említett példánál maradván a

```
G0 X60
Z1
X56
```

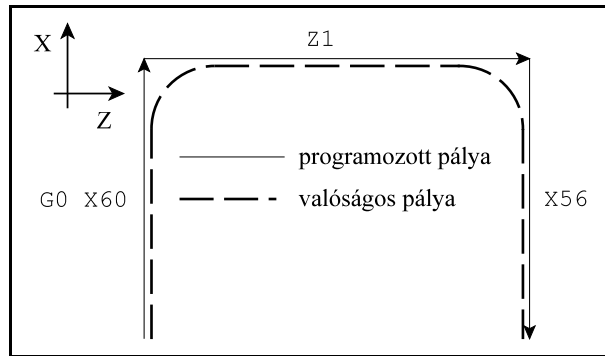
programrészletet végrehajtva a mellékelt ábrán látható a végrehajtási idők különbsége.



4.1.2-1 ábra

Az átlapolással történő pozicionálás esetén a szerszám pályája nem sarkos, hanem lekerekített. Emiatt a programban vigyázni kell, hogy a megszokottnál esetleg jobban elemeljük a darabtól a szerszámot.

Több tengely egy mondatban programozott pozicionálásakor a szerszám **csak közelítőleg mozog egyenes pálya mentén**, és a pozícióba is időeltéréssel érnek be a különböző tengelyek.



4.1.2-2 ábra

A vezérlő **felfüggeszti** a pozicionáló mondatok közötti **átlapolást** és **mindig végez pozícióellenőrzést** a mondat végpontján

G61 (pontos megállás üzemmód)

állapotban, vagy abban a pozicionáló mondatban ahova

G9 (pontos megállás)

kódot írunk, még a PCH=0 (nincs pozícióellenőrzés) paraméterállásnál is.

4.2 Az egyenes interpoláció (G1)

A G1 egyenes interpoláció parancs a szerszámot az összes, a mondatban programozott tengely mentén a megadott pontra mozgatja egyenes pálya mentén.

A mozgás a mondatban programozott, vagy a megörökölt F előtolással történik.

Abszolút adatmegadás esetén az aktuális munkadarab koordináta-rendszerben mozog a megadott pozíciójú pontra.

Inkrementális adatmegadás esetén a szerszám pillanatnyi pozíciójától számítva lépi le a megadott távolságot.

A mondat formátuma:

G1 v F

ahol v a mondatban megadott koordináták, F az előtolás értéke. A csatorna összes tengelye mentén mozoghatunk egy időben.

G1 helyett megadható G01 is.

Példa:

G1 X136 Z26 F0.5

A mondatban megadható más G kód és funkció is.

Példa:

G1 G91 X5 Z10 S2000 M3

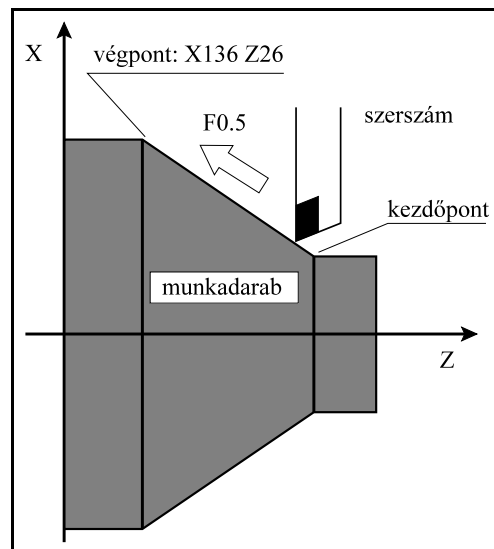
A G1 öröklődő kód, addig érvényes, amíg egy másik, interpolációs parancs át nem írja.

Például:

G1 X136 Z26 F0.5

Z0 (egyenes interpoláció F.5-tel, mert G1 és F öröklődik)

Bekapcsolás után az N1300 DefaultG1 paraméter G01=1 bitállása esetén G1 kód van érvényben.



4.2-1 ábra

Az F címen programozott előtolás mindig a programozott pálya mentén érvényesül. Tengelymenti komponensei:

$$\text{Előtolás az X tengely mentén: } F_x = \frac{\Delta x}{L} F$$

$$\text{Előtolás az Z tengely mentén: } F_z = \frac{\Delta z}{L} F$$

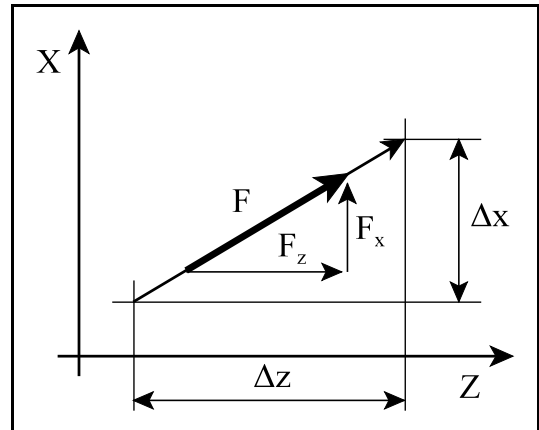
.....

A képlet folytatódik a mondatban programozott összes tengelyre.

Ahol: Δx , Δz , ...: a megfelelő tengelyek mentén számított elmozdulásértékek,

L: a programozott elmozdulás hossza:

$$L = \sqrt{\Delta x^2 + \Delta z^2 + \dots}$$



4.2-2 ábra

Forgó tengely mentén az előtolás °/perc dimenzióban értelmezett. A

G1 C270 F120

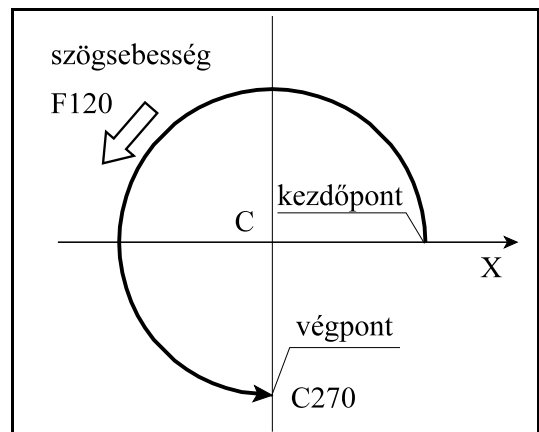
mondatban F120 jelentése: 120 °/perc.

Abban az esetben, ha egy lineáris és egy forgó tengely mozgását kapcsoljuk össze lineáris interpolációval az előtoláskomponensek szétosztása a fenti képletek alapján megy végbe.

Például:

G91 G01 Z100 C45 F120

mondatban a Z illetve B irányú előtoláskomponensek:



4.2-3 ábra

$$\text{Előtolás a Z tengely mentén: } F_z = \frac{100}{\sqrt{100^2 + 45^2}} 120 = 109.4 \quad \text{mm/perc}$$

$$\text{Előtolás a C tengely mentén: } F_c = \frac{45}{\sqrt{100^2 + 45^2}} = 49.2 \quad \text{°/perc}$$

4.3 A körinterpoláció (G2, G3)

A G2, vagy a G3 parancs a szerszámot a kiválasztott síkban a mondatban megadott pontra mozgatja körív mentén. A mozgás a mondatban programozott, vagy a megörökölt F előtolással történik.

A mondat formátuma:

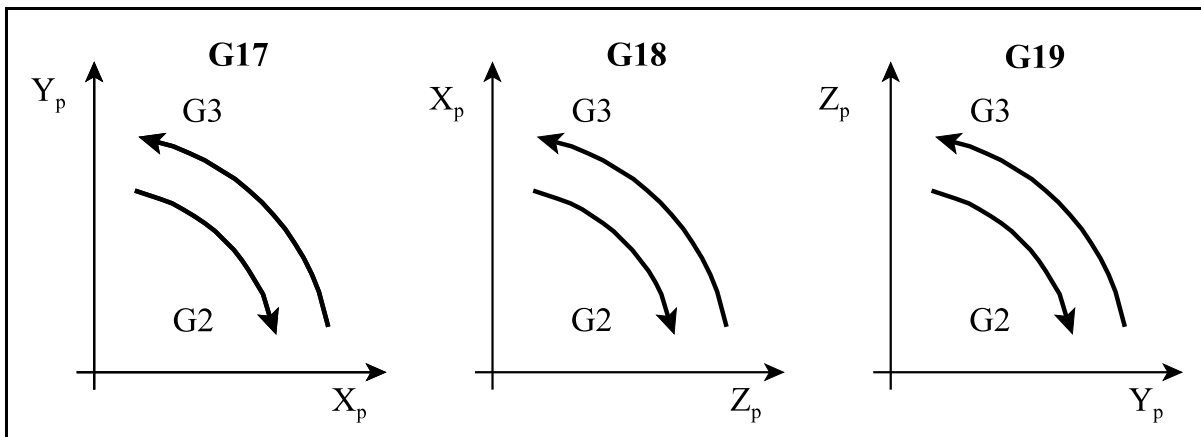
$$G17 \left\{ \begin{array}{l} G2 \\ G3 \end{array} \right\} X_{p-} Y_{p-} \left\{ \begin{array}{l} R_{-} \\ I_{-} J_{-} \end{array} \right\} F_{-}$$

$$G18 \left\{ \begin{array}{l} G2 \\ G3 \end{array} \right\} X_{p-} Z_{p-} \left\{ \begin{array}{l} R_{-} \\ I_{-} K_{-} \end{array} \right\} F_{-}$$

$$G19 \left\{ \begin{array}{l} G2 \\ G3 \end{array} \right\} Y_{p-} Z_{p-} \left\{ \begin{array}{l} R_{-} \\ J_{-} K_{-} \end{array} \right\} F_{-}$$

A körinterpoláció a G17, G18, G19 parancs által kiválasztott síkban megy végbe.

G2 esetén az *óramutató járásával megegyező*, **G3** esetén az *óramutató járásával ellentétes* irányban:



4.3-1 ábra

G2 és G3 helyett G02 és G03 kód is írható a programba.

G2 és G3 öröklődő kódok, addig érvényesek, amíg egy másik, interpolációs parancs át nem írja.

X_p , Y_p , Z_p jelentése itt, és a továbbiakban:

X_p : X tengely, vagy azzal párhuzamos tengely,

Y_p : Y tengely, vagy azzal párhuzamos tengely,

Z_p : Z tengely, vagy azzal párhuzamos tengely.

X_p , Y_p , Z_p értéke az adott koordinátarendszerben a *kör végpontjának koordinátája* abszolút, vagy a kezdőponttól mért inkrementális adatként megadva.

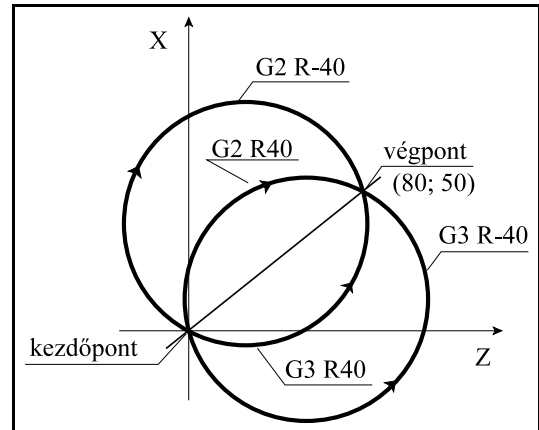
A kör további adatainak megadása kétféleképp történhet:

1. eset: kör megadása a sugarával R címen

Ekkor a vezérlő a kezdőpont koordinátáiból (az a pont ahol a vezérlés a körmondattal beolvasásának pillanatában tartózkodik), a végpont koordinátáiból (X_p , Y_p , Z_p címen definiált érték), valamint a programozott R körsugárból automatikusan kiszámítja a kör középpont koordinátáit. Egy adott körüljárási irány esetén ($G2$, vagy $G3$) a kezdő és végpont között két különböző, R sugarú kör húzható.

Ha a kör sugarát, azaz R értékét **pozitív** számmal adjuk meg 180° -nál kisebb **negatív** számmal adjuk meg 180° -nál nagyobb ívet interpolál. Például:

1. ívszakasz: G2 X80 Z50 R40
2. ívszakasz: G2 X80 Z50 R-40
3. ívszakasz: G3 X80 Z50 R40
4. ívszakasz: G3 X80 Z50 R-40



4.3-2 ábra

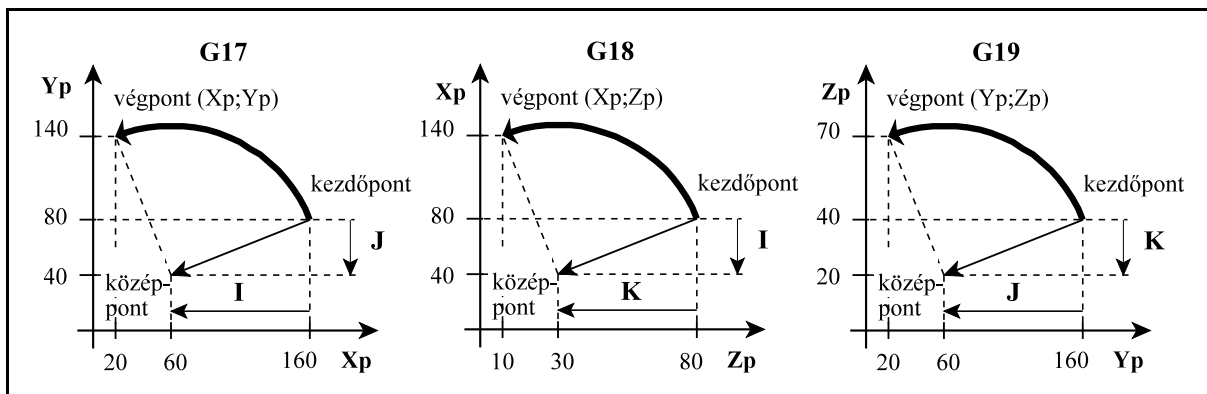
2. eset: kör megadása a középpontjával I, J, K címen

Az I, J, K címen megadott értékeket **inkrementálisan** értelmezi a vezérlő, úgy, hogy az I, J, K értékek által definiált vektor a kör **kezdőpontjától** a kör **középpontjára** mutat.

I, J, K értékeket mindig **sugárban** kell megadni, még akkor is, ha a hozzájuk tartozó tengelyek átmérőben történő programozásra vannak is állítva.

Például:

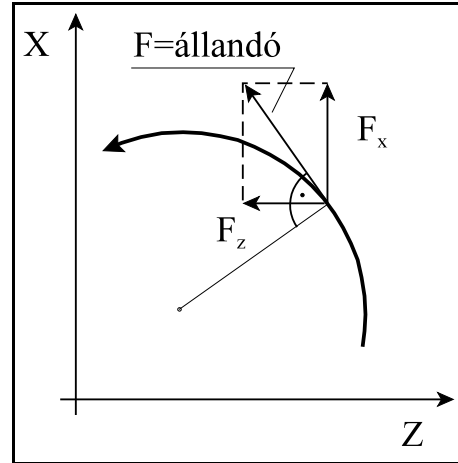
- G17 esetén: G3 X20 Y140 I-50 J-20 (X, Y átmérőben megadva)
 G18 esetén: G3 X140 Z10 I-20 K-50 (X átmérőben megadva)
 G19 esetén: G3 Y20 Z70 J-50 K-20 (Y átmérőben megadva)



4.3-3 ábra

4 Az interpoláció

F címen a pályamenti előtolást programozhatjuk, amely mindig a körérintő irányába mutat és állandó az egész pályán mentén.



4.3-4 ábra

Példa:

A mellékelt ábrán látható pályát programozzuk.

Kör programozása abszolút koordinátákkal és R megadásával:

```
G90 G18
G0 X0 Z130 M3 S1000
G1 X40 F500
G3 X180 Z60 R70
G2 X100 Z40 R50
G1 Z0
...
```

Kör programozása abszolút koordinátákkal és I, K kör középpont megadásával

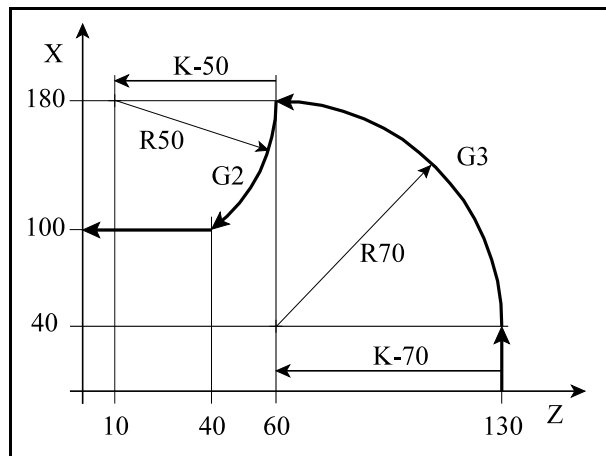
```
G90 G18
G0 X0 Z130 M3 S1000
G1 X40 F500
G3 X180 Z60 K-70
G2 X100 Z40 K-50
G1 Z0
...
```

Kör programozása inkrementális koordinátákkal és R megadásával:

```
G90 G18 G0 X0 Z130 M3 S1000
G91
G1 X40 F500
G3 X140 Z-70 R70
G2 X-80 Z-20 R50
G1 Z-40
...
```

Kör programozása inkrementális koordinátákkal és I, K kör középpont megadásával

```
G90 G18 G0 X0 Z130 M3 S1000
G91
G1 X40 F500
G3 X140 Z-70 K-70
G2 X-80 Z-20 K-50
G1 Z-40
...
```



4.3-5 ábra

I0, J0, K0 megadása elhagyható. Például:

```
G0 X0 Z100 F500
G18 G03 X200 Z0 K-100
```

100 mm sugarú, origó középpontú negyedkör programozása esetében, mivel a kör középpontja X irányban 0 távolságra van az X0 Z100 ponttól I0-t nem kell kiírni.

Ha X_p, Y_p, Z_p , mind elhagyásra kerül:

– ha kör középponti koordinátákat adunk meg I, J, K címen: 360°-os ívű, teljes kört interpolál a vezérlő. Például:

```
G0 X400 Y0 F500
G17 G03 I-100
```

esetben 100 mm sugarú, X200 Y0 középpontú, teljes kört interpolál a vezérlő

– ha R sugarat adunk meg, például

```
G0 X0 Z200 F500
G18 G03 R100
```

a vezérlő nem mozog és nem jelez hibát.

Ha a körmondattal **sem sugarat R, sem I, J, K-t** nem tartalmaz, a vezérlő 2015 Körmegadás hibás üzenetet küld.

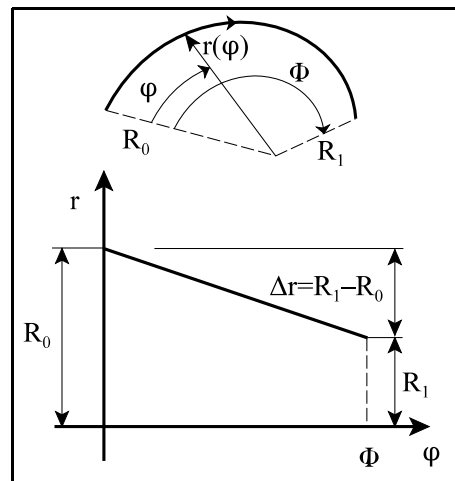
Ha a **kiválasztott síkon kívüli I, J, K** címre történik hivatkozás a vezérlő 2015 Körmegadás hibás üzenetet küld.

Ha a G2, G3 mondatban meghatározott **kör kezdőponti és végponti sugarának különbsége** nagyobb, mint az N1339 Radius Diff paraméteren meghatározott érték a vezérlés 2012 Sugárkülönbséghiba körben jelzést ad.

Ha a sugarak különbsége kisebb a fenti paraméteren megadott értéknél a vezérlés a szerszámot olyan síkbeli spirális pálya mentén mozgatja, amelynél a sugár a központi szög függvényében lineárisan változik.

Változó sugarú körív interpolációjánál nem a pályamenti sebesség, hanem a szögsebesség lesz állandó.

N1339 Radius Diff paraméter értéke legyen nagyobb, mint 0, pl. 0.01, különben szükségtelen hibajelzéseket küld a vezérlő.

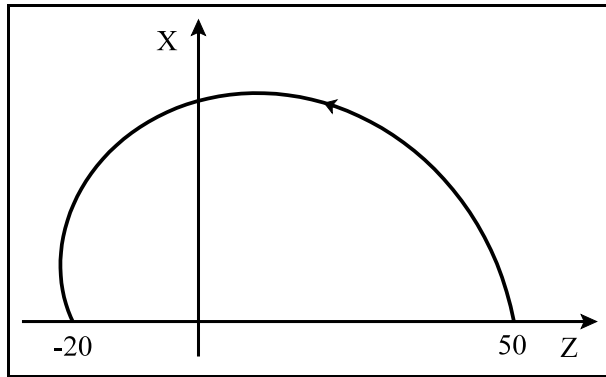


4.3-6 ábra

Sugárkülönbség hibát, vagy változó sugarú kör interpolációját az alábbi esetek okozhatnak:

Ha az I, J, K címen megadott kör középponti pozíció nem jó. Például:

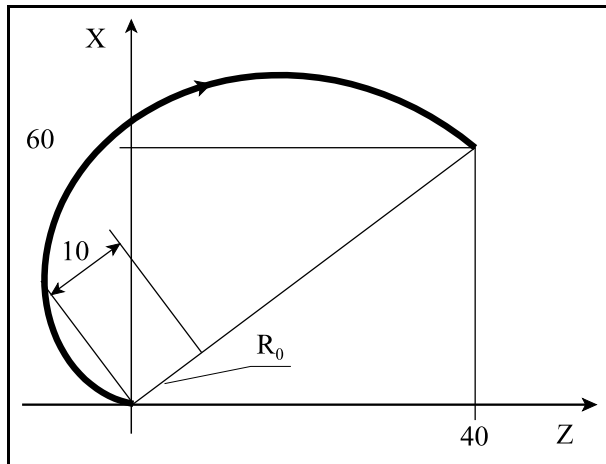
```
G18 G90 G0 X0 Z50
G3 Z-20 K-50
```



4.3-7 ábra

Ha a megadott körsugár kisebb, mint a kezdőpontot a végponttal összekötő egyenes távolságának a fele, a vezérlő a megadott körsugarat tekinti a kör kezdőpontjára, és olyan változó sugarú kört interpolál, amelyik középpontja a kezdőpontot a végponttal összekötő egyenesen van, a kezdőponttól R távolságra:

```
G18 G0 G90 X0 Z0
G2 X60 Z40 R10
```



4.3-8 ábra

A kör középponti pozícióját, I, J, K-t, meg lehet adni abszolút, a munkadarab nullpottól számított értékkel is. Ehhez az N1337 Execution Config paraméter #2 CCA=1 bitbeállítás szükséges. Ez az eset kerülendő.

4.3.1 A síkbeli spirálinterpoláció (G2, G3)

A G2, vagy a G3 paranccsal síkbeli spirálinterpolációt programozhatunk úgy, hogy L címen megadjuk a spirál fordulatainak a számát is. A *kör középpontját és végpontját* úgy adjuk meg, hogy a *kezdőponti és a végponti körsugár különböző* legyen. A mozgás a mondatban programozott, vagy a megörökölt F előtolással történik.

A mondat formátuma:

$$G17 \left\{ \begin{array}{l} G2 \\ G3 \end{array} \right\} X_p _ Y_p _ I _ J _ L _ F _$$

$$G18 \left\{ \begin{array}{l} G2 \\ G3 \end{array} \right\} X_p _ Z_p _ I _ K _ L _ F _$$

$$G19 \left\{ \begin{array}{l} G2 \\ G3 \end{array} \right\} Y_p _ Z_p _ J _ K _ L _ F _$$

X_p , Y_p , Z_p értéke az adott koordinátarendszerben a *spirál végpontjának koordinátája* abszolút, vagy a kezdőponttól mért inkrementális adatként megadva.

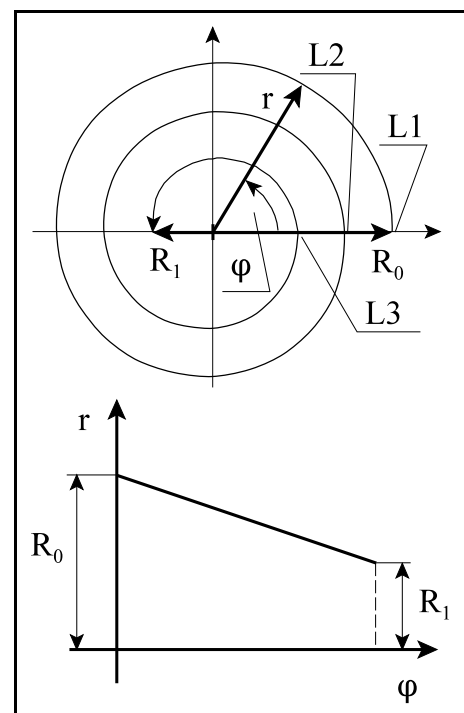
Az I , J , K címeiken a spirál *középpontjának koordinátáit* adjuk meg a kezdőponttól mért *inkrementális* adatként, úgy, hogy az I , J , K értékek által definiált vektor a spirál *kezdőpontjától* a spirál *középpontjára* mutat.

I , J , K értékeket mindig *sugárban* kell megadni, még akkor is, ha a hozzájuk tartozó tengelyek átmérőben történő programozásra vannak is állítva.

L címen a *spirál fordulatainak számát* adjuk meg. Minden bekezdés egy új fordulatot jelent, még akkor is, ha nem teljes fordulat következik utána. L cím *pozitív egész szám*.

A spirális interpoláció során a kezdőponti sugarat (R_0) a forgásszög (φ) függvényében lineárisan úgy változtatja, hogy mire az L címen megadott fordulatot megtette és a végponti pozícióra ér, a végponti sugár a programozott adatoknak feleljen meg.

A fentiekből következik, hogy spirálinterpoláció során, amit az L cím kitöltése jelez a vezérlőnek, a kezdőponti sugár eltér a végponti sugártól. Ha L cím ki van töltve a vezérlő sohasem vizsgálja a N1339 Radius Diff paraméteren meghatározott maximális sugárkülönbség értéket. A spirálinterpolációban megadott F *előtolás a spirál teljes hosszában állandó* marad.



4.3.1-1 ábra

Példa:

A mellékelt ábrán látható spirált programozzuk.

A spirál kezdőpontja X180 Y0, középpontja a kezdőponttól mérve I-90, J0, fordulatonkénti sugárváltozása 24, és 2,5 fordulatot tesz meg. Így a spirál végponti sugara:

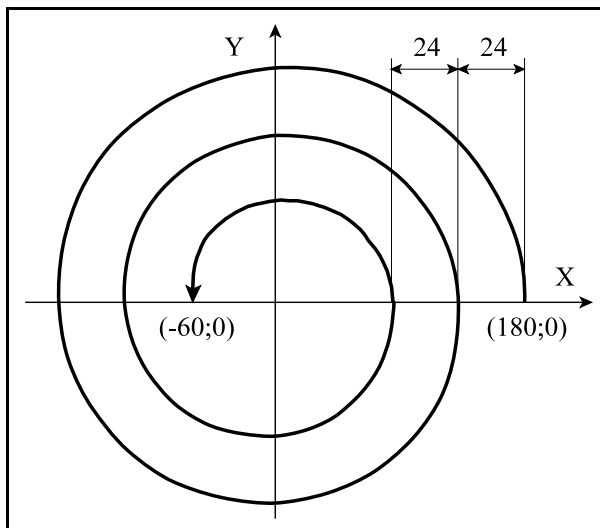
$$R=90-2*24-24/2=30$$

A spirál megkezdett fordulatainak száma 3.

A program:

```
G17 G90 G94 M3 S1000
G0 X180 Y0 F1000
G3 X-60 I-90 L3
```

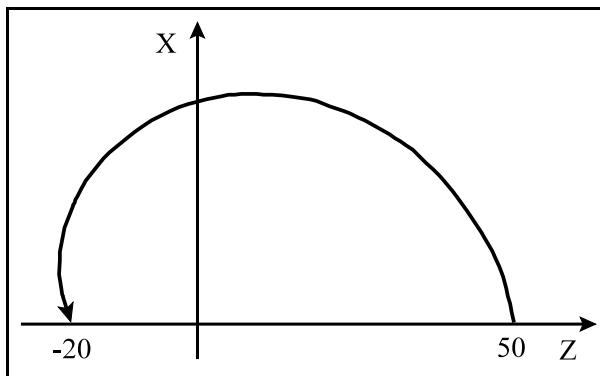
A fenti példában *X* és *Y* koordináta **átmérőben** van megadva.



4.3.1-2 ábra

Ha azt akarjuk, hogy a 4.3 A körinterpoláció fejezetben (28 oldal) megadott példára ne jelezzon a vezérlő 2012 Sugárkülönbséghiba körben hibát így módosítsuk a programot:

```
G18 G90 G0 X0 Z50
G3 Z-20 K-50 L1
```



4.3.1-3 ábra

4.3.2 A hengeres spirálinterpoláció (G2, G3)

A G2, vagy a G3 paranccsal hengeres spirálinterpolációt programozhatunk úgy, hogy **a kör síkjára merőleges tengelyen mozgást programozunk**. **L címen** adhatjuk meg a **spirál fordulatainak a számát**. A mozgás a mondatban programozott, vagy a megörökölt F előtolással történik.

A mondat formátuma:

$$G17 \left\{ \begin{array}{l} G2 \\ G3 \end{array} \right\} X_{P_} Y_{P_} \left\{ \begin{array}{l} R \\ I_ J_ \end{array} \right\} Z_{P_} q \dots L_ F_$$

$$G18 \left\{ \begin{array}{l} G2 \\ G3 \end{array} \right\} X_{P_} Z_{P_} \left\{ \begin{array}{l} R_ \\ I_ K_ \end{array} \right\} Y_{P_} q \dots L_ F_$$

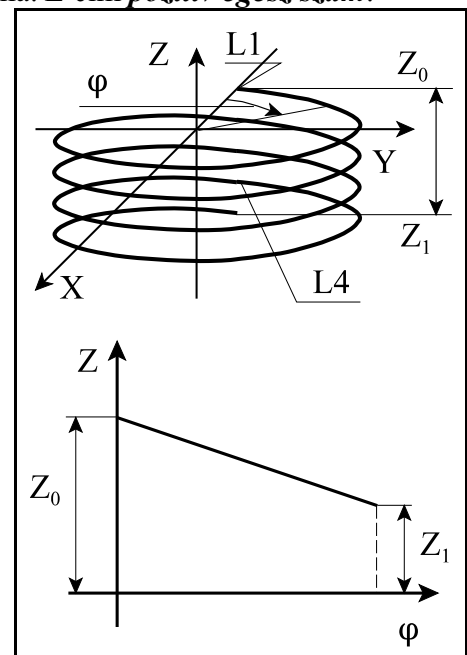
$$G19 \left\{ \begin{array}{l} G2 \\ G3 \end{array} \right\} Y_{P_} Z_{P_} \left\{ \begin{array}{l} R_ \\ J_ K_ \end{array} \right\} X_{P_} q \dots L_ F_$$

A kör megadása a körinterpolációnál megadott szabályok szerint történik. A körív síkjára merőleges tengelyen az elmozdulás a körív mentén végzett elmozdulással arányos.

A körre merőleges tengely mellé tetszőleges számú, a fenti képletben q...-val jelzett, a csatornában programozható tengelyre adhatunk meg elmozdulást.

L címen a hengeres spirál fordulatainak számát adjuk meg. Minden bekezdés egy új fordulatot jelent, még akkor is, ha nem teljes fordulat következik utána. **L cím pozitív egész szám**.

A hengeres spirálinterpoláció során a vezérlő a menetemelkedést a forgásszög (φ) függvényében lineárisan úgy változtatja, hogy mire az L címen megadott fordulat megtette és a végponti pozícióra ér, a kör síkján kívül eső tengelyeken a végponti pozíció a programozott adatoknak feleljen meg.



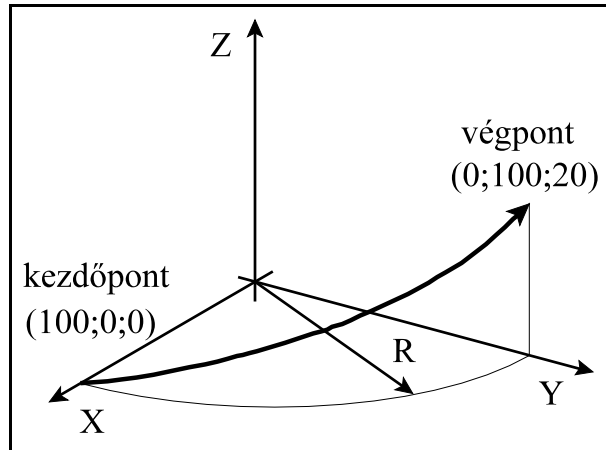
4.3.2-1 ábra

Példák:

A mellékelt ábrán látható hengeres spirálinterpolációt a következőképp adhatjuk meg:

```
G17 G90 G0 X100 Y0 Z0
G03 X0 Y100 Z20 R50 F150
```

A fenti példában X és Y koordináta *átmérőben* van megadva.

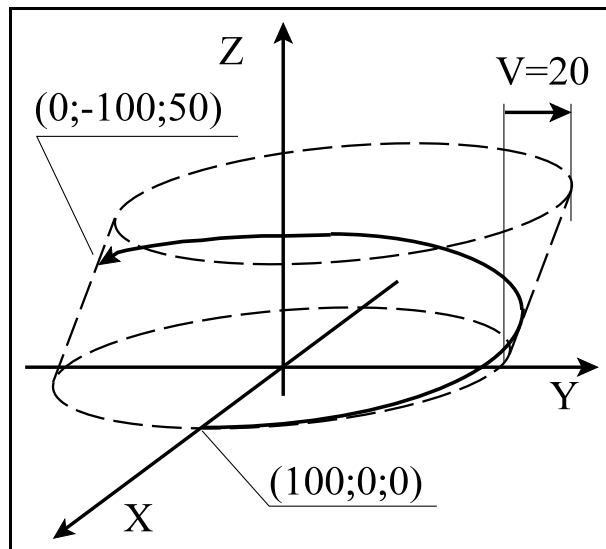


4.3.2-2 ábra

A mellékelt ábrán látható ferde henger palástjára marunk körívet. A V tengely az Y tengellyel párhuzamos, amit a Z tengellyel együtt mozgat a vezérlő:

```
G17 G90 G0 X100 Y0 Z0 V0
G03 X0 Y-100 Z50 V20 I-50
```

A fenti példában X és Y koordináta *átmérőben* van megadva.



4.3.2-3 ábra

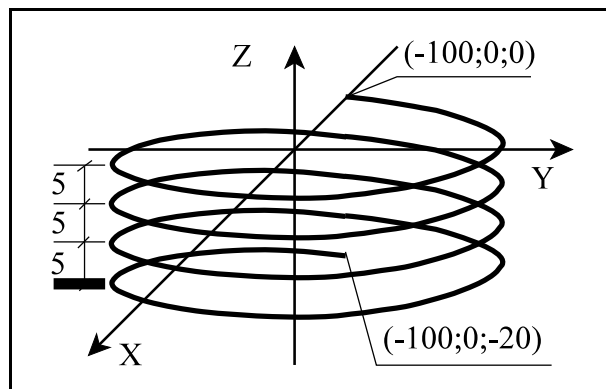
A mellékelt ábrán látható 4 fordulatú, hengeres spirált programozzuk.

A spirál kezdőpontja $X-100 Y0, Z0$, középpontja a kezdőponttól mérve $I-50, J0$, fordulatokénti menetemelkedése 5, és 4 egész fordulatot tesz meg.

A program:

```
G54 G17 G90 ...
G0 X-100 Y0
Z0
G2 I50 Z-20 L4 F100
...
```

A fenti példában X és Y koordináta *átmérőben* van megadva.



4.3.2-4 ábra

Az F címen megadott előtolás alapesetben az N1337 Execution Config #3 $HEF=0$ bitállásánál a **körpálya mentén** érvényesül. Ekkor a tengelyekre az előtolás az alábbi képletek szerint adódnak:

$$F_{iv} = F$$

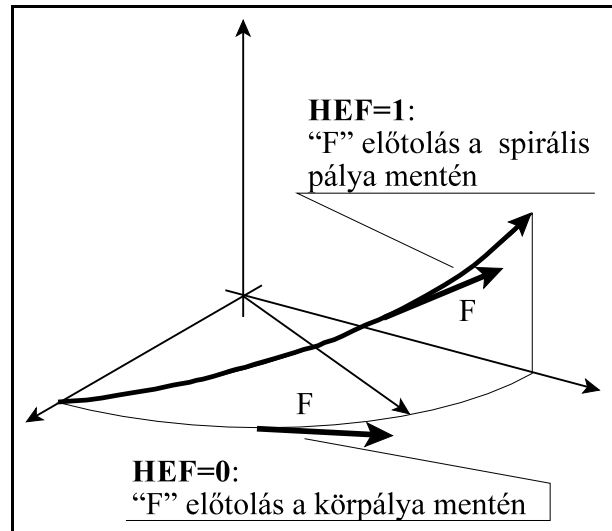
$$F_q = \frac{L_q}{L_{iv}} F$$

A $HEF=1$ bitállásnál az előtolást a **spirális pálya mentén** számolja. Ekkor a tengelyekre az előtolás az alábbi képletek szerint adódnak:

$$F_{iv} = \frac{L_{iv}}{\sqrt{L_{iv}^2 + L_q^2}} F$$

$$F_q = \frac{L_q}{\sqrt{L_{iv}^2 + L_q^2}} F$$

ahol L_q : elmozdulás a q tengely mentén,
 L_{iv} : a körív hossza,
 F : a programozott előtolás,
 F_q : előtolás a q tengely mentén
 F_{iv} : előtolás a körív mentén



4.3.2-5 ábra

A megadott **szerszámsugár korrekció** mindig a **kör** síkjában a körpálya mentén érvényesül. Abban az esetben, ha a kiválasztott síkban megadott **kör sugara változó**, az interpoláció a megadott kúp palástja mentén történik.

4.3.3 A kúpos spirálinterpoláció (G2, G3)

A G2, vagy a G3 paranccsal kúpos spirálinterpolációt programozhatunk úgy, hogy **a kör síkjára merőleges tengelyen mozgást programozunk**. A **kör középpontját és végpontját** úgy adjuk meg, hogy a **kezdőponti és a végponti körsugár különböző** legyen. **L címen** adhatjuk meg a **kúpos spirál fordulatainak a számát**. A mozgás a mondatban programozott, vagy a megörökölt F előtolással történik.

A mondat formátuma:

$$G17 \left\{ \begin{array}{l} G2 \\ G3 \end{array} \right\} X_P_Y_P_I_J_Z_P_q\dots L_F_$$

$$G18 \left\{ \begin{array}{l} G2 \\ G3 \end{array} \right\} X_P_Z_P_I_K_Y_P_q\dots L_F_$$

$$G19 \left\{ \begin{array}{l} G2 \\ G3 \end{array} \right\} Y_P_Z_P_J_K_X_P_q\dots L_F_$$

X_p , Y_p , Z_p értéke az adott koordináta-rendszerben a **spirál végpontjának koordinátája** abszolút, vagy a kezdőponttól mért inkrementális adatként megadva.

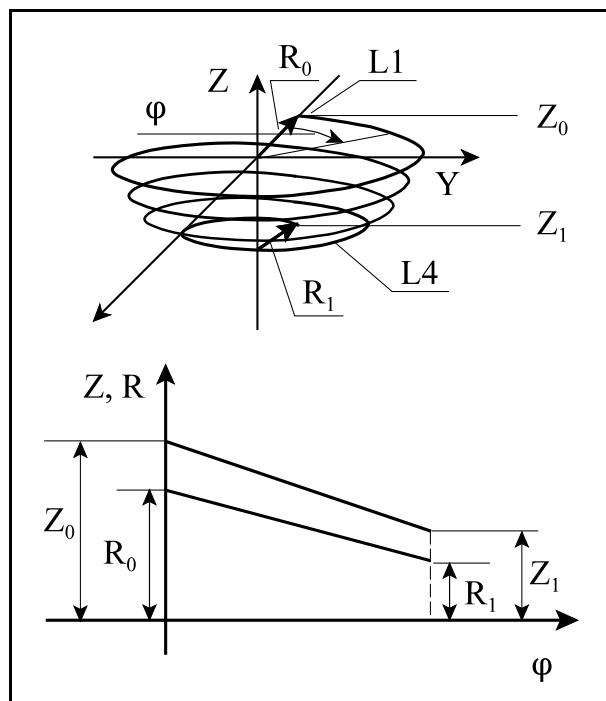
A körre merőleges tengely mellé tetszőleges számú, a fenti képletben $q\dots$ -val jelzett, a csatornában programozható tengelyre adhatunk meg elmozdulást.

Az **I, J, K** címeken a kúpos spirál **középpontjának koordinátáit** adjuk meg a kiválasztott síkban, a kezdőponttól mért **inkrementális** adatként, úgy, hogy az I, J, K értékek által definiált vektor a spirál **kezdőpontjától** a spirál **középpontjára** mutat.

I, J, K értékeket mindig **sugárban** kell megadni, még akkor is, ha a hozzájuk tartozó tengelyek átmérőben történő programozásra vannak is állítva.

L címen a spirál fordulatainak számát adjuk meg. Minden bekezdés egy új fordulatot jelent, még akkor is, ha nem teljes fordulat következik utána. **L cím pozitív egész szám**.

A kúpos spirálinterpoláció során a menetemelkedést a forgásszög (φ) függvényében lineárisan úgy változtatja, hogy mire az L címen megadott fordulatot megtette és a végponti pozícióra ér, a kör síkján kívül eső tengelyeken a végponti pozíció a programozott adatoknak feleljen meg. Ezzel együtt a kör su-



4.3.3-1 ábra

garát is lineárisan változtatja a forgászög függvényében.

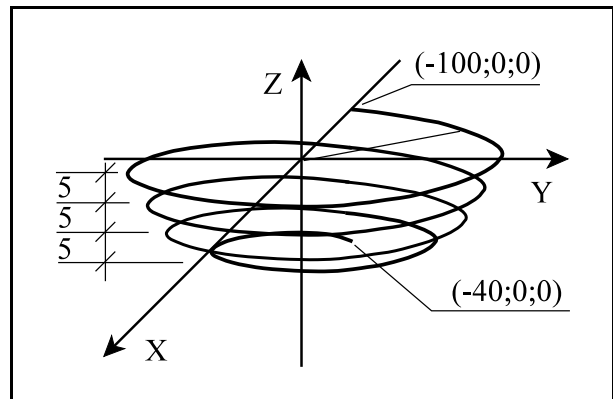
Példa:

Programozzuk 4 fordulatú kúpos spirált, aminek a kezdőponti sugara 50, végponti sugara 20 és a fordulatonkénti menetemelkedése 5.

A program:

```
G17 G90 ...
G0 X-100 Y0
Z0
G2 X-40 I50 Z-20 L4 F100
...
```

A fenti példában X és Y koordináta *átmérőben* van megadva.



4.3.3-2 ábra

4.4 Egyenletes emelkedésű menet vágása (G33)

A G33 utasítással hengeres, vagy kúpos, egyenletes emelkedésű menet vágását lehet programozni. A mondat formátuma:

G33 v F Q

vagy

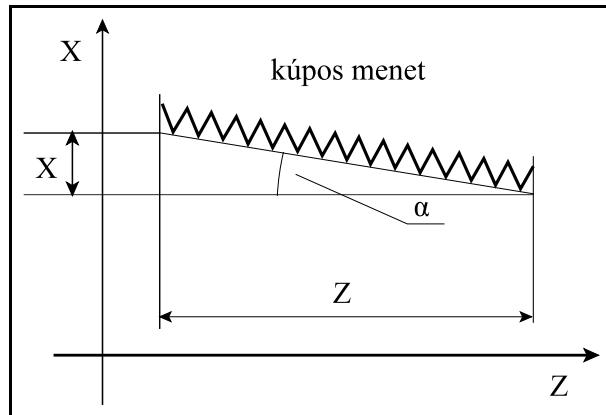
G33 v E Q

A "v" vektorra maximum két tengely koordinátaadatát lehet beírni. Ha a v vektoron két koordináta adata van feltüntetve a vezérlés kúpos menetet vág. A menetemelkedést azon tengely mentén veszi figyelembe a vezérlő, amelyiken hosszabb elmozdulás adódik.

ha $\alpha < 45^\circ$, azaz $Z > X$ a programozott menetemelkedést a **Z tengely mentén**,

ha $\alpha > 45^\circ$, azaz $X > Z$ a programozott menetemelkedést az **X tengely mentén** veszi figyelembe.

A menetemelkedést kétféleképp lehet definiálni:



4.4-1 ábra

Menetemelkedés megadása F címen

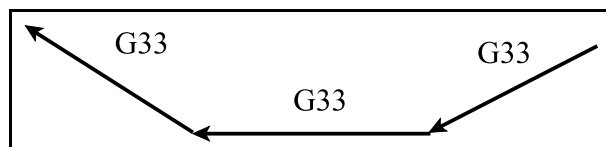
Ha a menetemelkedést **F címen** adjuk meg, akkor az adat értelmezése **mm/ford**, vagy **inch/ford**. Ha tehát egy 2.5 mm emelkedésű menetet akarunk vágni F2.5-öt kell programozni.

Menetemelkedés megadása E címen

Ha a menetemelkedést **E címen** adjuk meg a vezérlés inches menetet vág. **E cím** értelmezése **inchenkénti menetszám**. Ha például E8-at programozunk, akkor a vezérlő $1/8" = 25.4/8 = 3.175$ mm emelkedésű menetet vág.

Q címen adjuk meg azt a szögértéket, hogy az **orsó jeladó nullimpulzusától számítva hány fokot forduljon** el az orsó, mielőtt elkezd a menetet vágni. **Több-bekezdésű menetet** a **Q** érték megfelelő programozásával lehet vágni, vagyis itt lehet programozni, hogy a különböző bekezdéseket milyen orsó szögelfordulás alatt kezdje el vágni a vezérlő. Például, ha egy kétbekezdésű menetet akarunk vágni, az első bekezdést Q0-ról indítjuk (külön megadni nem kell), a második bekezdést pedig Q180-ról.

A G33 öröklődő funkció. Ha egymás után több menetvágó mondatot programozunk, tetszőleges, egyenes szakaszokkal határolt felületre vághatunk menetet.



A vezérlés a főorsó jeladó nullimpulzusára az **4.4-2 ábra**

első mondatban szinkronozódik rá, és a további mondatoknál már nem végez szinkronizációt, következésképp a menet emelkedése folyamatos lesz az összes szakaszon. Ebből adódóan a programozott Q főorsó szögelfordulást is csak az első mondatban veszi figyelembe.

Példa:

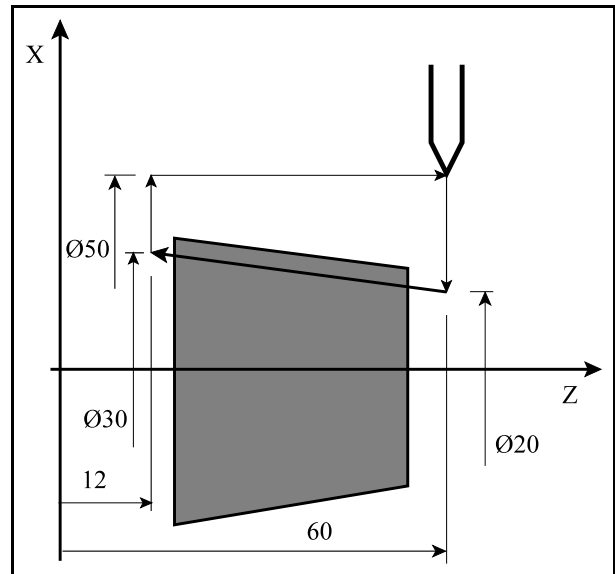
Vágjunk 2 mm emelkedésű menetet:

```

...
G0 G90 X50 Z60
X20
G33 X30 Z12 F2
G0 X50
Z60
...

```

A példában X-ben átmérő programozást vetünk figyelembe.



4.4-3 ábra

☞ *Megjegyzések:*

- Ha a menetvágó mondatban F és E cím is ki van töltve, F0-t, vagy E0-t adunk meg a vezérlő 2021 Menetvágás megadása hibás hibajelzést ad.
- G33 parancs végrehajtása során az előtolás és főorsó override értékeket automatikusan 100%-nak veszi a vezérlés és a Stop állj billentyű hatástalan.
- A szervorendszer követési hibája miatt a menet elején és végén ráfutási és kifutási távolságot kell hagyni az anyagon kívül a szerszámnak, hogy a menetemelkedés állandó legyen a teljes szakaszon.
- Menetvágás parancs kiadása előtt ki kell kapcsolni a konstans vágósebességszámítás (G96) állapotot.

4.5 A polárkoordináta interpoláció (G12.1, G13.1)

A polárkoordináta interpoláció a vezérlésnek egy olyan működési módja, amelyben a derékszögű (Descartes) koordinátarendszerben leírt munkadarab kontúr pályáját egy **lineáris** és egy **forgó tengely** mozgásával járja le, vagyis a derékszögű koordinátákkal megadott pályát a mozgás során pillanatról pillanatra átszámolja polárkoordináta adatokkal ábrázolt pályára.

A

G12.1 polárkoordináta interpoláció beutasítás bekapcsolja a polárkoordináta üzemet. Az ezután következő programrészben a marószerszám pályáját **derékszögű koordinátarendszerben** egy lineáris és egy **forgó tengely** (**virtuális lineáris** tengely) programozásával írhatjuk le. *Az utasítást mindig külön mondatban adjuk meg, és más utasítást nem programozhatunk mellé.*

A

G13.1 polárkoordináta interpoláció kiutasítás kikapcsolja a polárkoordináta üzemet. *Az utasítást mindig külön mondatban adjuk meg, és más utasítást nem programozhatunk mellé.* A vezérlés bekapcsolás, vagy a reset után mindig G13.1 állapotot vesz fel.

A lineáris és a forgó tengely kiválasztása

A polárkoordináta interpoláció bekapcsolása előtt ki kell választani egy lineáris és egy forgó (virtuális) tengelyt, amelyek a polárkoordináta interpolációban részt vevő tengelyek.

A tengelyek kiválasztása a G17, G18, G19 síkválasztó utasítással történik.

G17 X_p_ egy forgó (virtuális) tengely címe_

G18 Z_p_ egy forgó (virtuális) tengely címe_

G19 Y_p_ egy forgó (virtuális) tengely címe_

Mindig a **kiválasztott sík első tengelye** lesz a **lineáris tengely**. Párhuzamos tengely is kiválasztható.

A síkválasztó utasításban megadott **forgó tengely** lesz a polárkoordináta interpoláció forgó tengelye. Ezek után a G12.1 utasítás az így megadott lineáris és forgó tengelycímeiken megadott adatokkal számol.

Például a

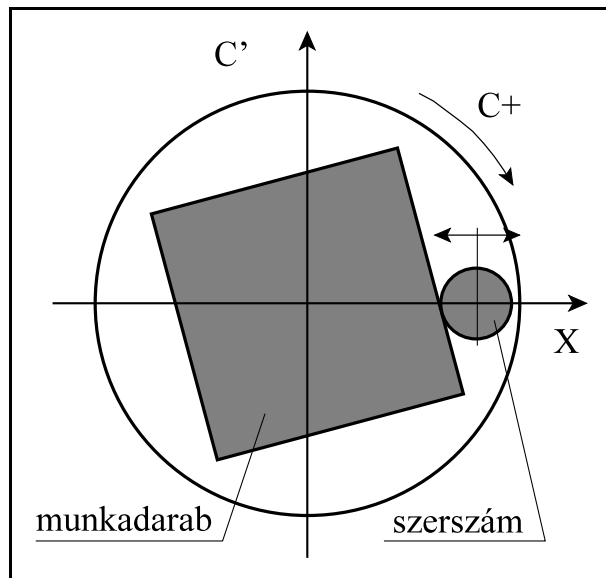
G17 X_ C_

utasítás az X tengelyt jelöli ki lineáris, a C-t forgó tengelynek.

A

G19 Y_ C_

utasítás viszont az Y tengelyt jelöli ki lineáris, a C-t forgó tengelynek.



4.5-1 ábra

A munkadarab nullpont helyzete a polárkoordináta interpoláció során

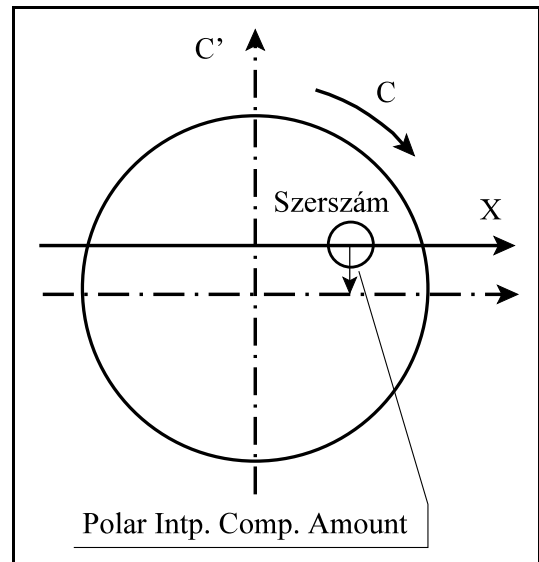
A polárkoordináta interpoláció bekapcsolása előtt olyan munkadarab koordinátarendszert kell kiválasztani, amelyben a **forgó tengely forgásközéppontja egybeesik** a polárinterpoláció **lineáris tengelyének origójával**.

Például, ha C a forgó és X a lineáris tengely, a koordinátarendszer nullpontját az X tengelyen úgy kell megválasztani, hogy a szerszám X=0 pozíciója egybeesék a körtengely (C) forgástengelyével.

Ha a forgó tengely forgásközéppontja nem esik egybe a kiválasztott sík első tengelyével (lineáris tengely), más szavakkal a forgó tengely forgásközéppontja nem esik egybe a szerszám forgástengelyével a lineáris tengelyre merőleges irányban, a megmunkált darab torzulni fog. A torzulás mértéke annál nagyobb lesz, minél közelebb megy a szerszám az origóhoz.

Mivel ebben az irányban általában nincs tengely ezért nullponteltolással ez az eltérés nem kompenzálható.

Az N0217 Polar Intp. Comp. Amount paraméteren ezt az eltérést kompenzálni lehet. A kompenzáció értékét **mindig a forgó tengely címére** kell írni mm-ben, vagy inch-ben.



4.5-2 ábra

A tengelyek helyzete a polárkoordináta interpoláció bekapcsolása pillanatában

A polárkoordináta interpoláció bekapcsolása előtt (G12.1 utasítás) gondoskodni kell arról, hogy a **körtengely a 0 pozíciój**ú pontban legyen. A **lineáris tengely pozíciója** lehet **negatív** és **pozitív** is, de **nem lehet 0**.

A hosszadatok programozása a polárkoordináta interpoláció során

A polárkoordináta interpoláció bekapcsolt állapotában a kiválasztott síkhoz tartozó mindkét tengelyen hosszadatokat programozunk: a kiválasztott síkban szereplő forgó tengely lesz a második (virtuális) tengely. Ha pl. a G17 X_ C_ utasítással az X, C tengelyt választottuk ki, a C címet úgy programozhatjuk, mint a G17 X_ Y_ síkválasztás esetén az Y-t.

Polárkoordináta interpolációban alkalmazhatjuk a G16 programozás polárkoordinátában funkciót is. Ekkor, értelemszerűen, a kiválasztott sík első tengelycímén adjuk meg a polársugarat, a forgó tengelyen a polárszöveget.

Alapesetben a virtuális tengely programozását nem befolyásolja, hogy az első tengely programozása átmérőben történik-e, a virtuális tengelyen mindig sugárban kell megadni a koordinátaadatokat. Ha pl. a polárkoordináta interpoláció az X C síkban történik, függetlenül attól, hogy az X címet átmérőben, vagy sugárban adjuk meg, a C címre írt értéket sugárban kell megadni.

A polárkoordináta interpolációban részt nem vevő tengelyek mozgása

A szerszám ezeken a tengelyeken, a polárkoordináta interpoláció bekapcsolt állapotától függetlenül, úgy mozog, mint normális esetben.

Körinterpoláció programozása a polárkoordináta interpoláció során

A polárkoordináta interpoláció bekapcsolt állapotában kör megadása a már ismert módon, sugárral, vagy kör középponti koordináta programozásával lehetséges. Ha ez utóbbit választjuk I, J, K címeiket a kiválasztott síknak megfelelően, az alábbiak szerint kell használni:

- G17** X_p egy forgó (virtuális) tengely címe I J
- G18** Z_p egy forgó (virtuális) tengely címe I K
- G19** Y_p egy forgó (virtuális) tengely címe J K

Szerszámsugár korrekció használata polárkoordináta interpoláció esetén

G41, G42 utasítás a megszokott módon használható a polárkoordináta interpoláció bekapcsolt állapotában. Ügyeljünk arra, hogy a forgó szerszám korrekciós csoportjában a szerszámállítás kódja Q=0 kell legyen. Használatára az alábbi megszorítások vonatkoznak:

A polárkoordináta interpoláció bekapcsolása (G12.1 utasítás) csak G40 állapotban lehetséges, Ha G12.1 állapotban bekapcsoltuk G41, vagy G42-t, a polárkoordináta interpoláció kikapcsolása (G13.1 utasítás) előtt G40-et kell programozni.

Programozási megszorítások a polárkoordináta interpoláció során

A polárkoordináta interpoláció bekapcsolt állapotában az alábbi utasítások nem használhatók:

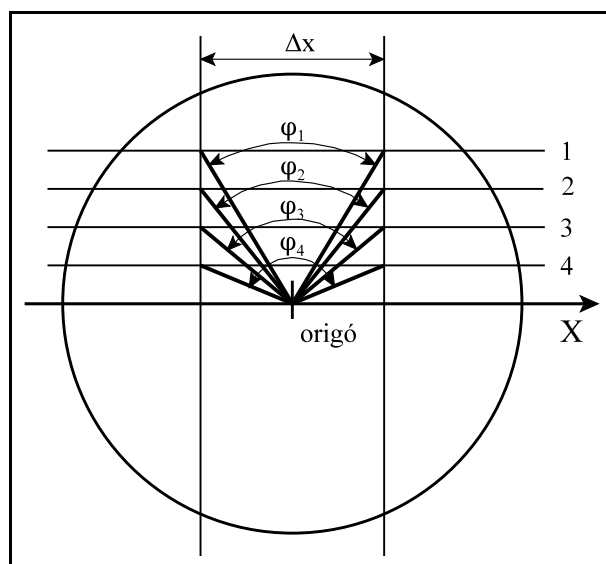
- síkváltás: G17, G18, G19,
- koordinátatranszformációk: G52, G92,
- munkadarab koordinátarendszer váltás: G54, ..., G59,
- pozicionálás a gépi koordinátarendszerben: G53
- G28 referenciapont felvétel, G30 Pp: 2. 3. 4. referenciapontra futás,
- G31 mérés a maradékút törlésével.

Az előtolás a polárkoordináta interpoláció során

Az előtolás értelmezése a polárkoordináta interpoláció bekapcsolt állapotában a derékszögű interpolációnál megszokott módon, pályamenti sebességként történik: a munkadarab és a szerszám relatív sebességét adja meg.

A polárkoordináta interpoláció során egy derékszögű koordinátarendszerben megadott pályát egy lineáris és egy forgó tengely mozgásával járja le. Ahogy a szerszám középpontja közeledik a körkoordináta forgástengelyéhez, úgy kellene a forgó tengelynek időegység alatt mind nagyobb és nagyobbat lépnie ahhoz, hogy a pályamenti sebesség állandó legyen. A körtengely sebességnek viszont határt szab a forgó tengelyre megengedhető maximális sebesség, amit paraméter határoz meg. Ezért az origó közelében a vezérlő fokozatosan csökkenti a pályamenti előtolást, annak érdekében, hogy a forgó tengely sebessége ne növekedjék minden határon túl.

A mellékelt ábra azt az esetet mutatja, amikor az X tengellyel párhuzamos egyeneseket (1, 2, 3, 4) programozunk. A programozott előtóláshoz időegység alatt Δx elmozdulás tartozik. A Δx elmozduláshoz a különböző egyenesek (1,



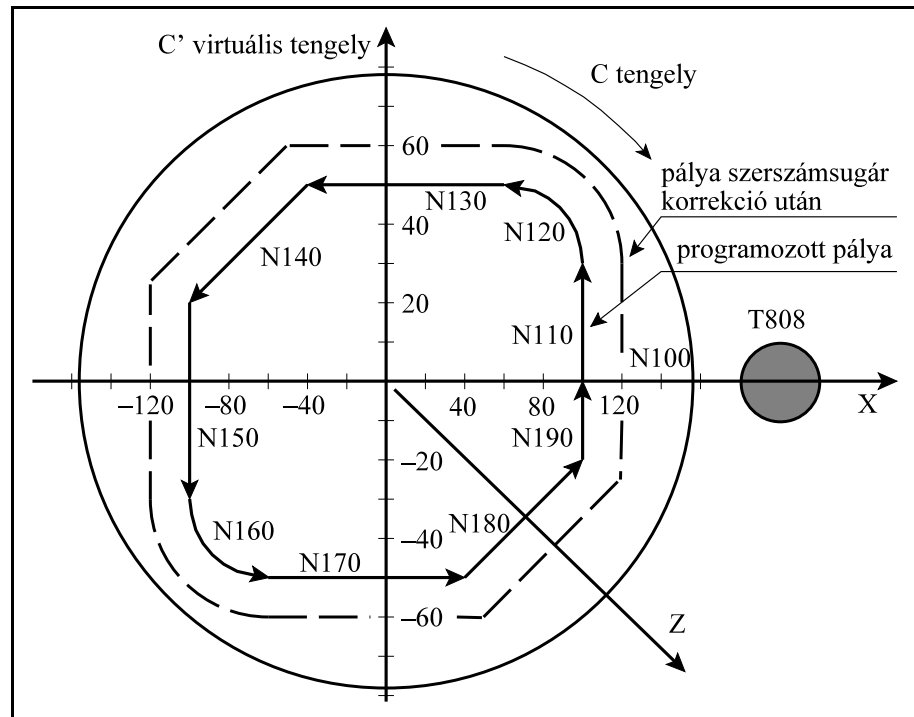
4.5-3 ábra

2, 3, 4) esetén más és más szögelfordulás ($\varphi_1, \varphi_2, \varphi_3, \varphi_4$) tartozik. Látszik, hogy minél közelebb jár a megmunkálás az origóhoz, annál nagyobb szögelfordulást kell a forgó tengelynek időegység alatt megtennie, hogy a programozott előtolást tartani tudja.

Ha ilyen esetben a szögsebesség meghaladja a forgó tengelyre az N0305 Max Feed paraméteren beállított értéket a vezérlő fokozatosan csökkenti a pályamenti előtolást.

Mintapélda

Az alábbiakban közlünk a polárkoordináta interpoláció használatára egy mintapéldát. Az interpolációban részt vevő tengelyek: X (lineáris tengely) és C (forgó tengely). **X tengely** programozása **átmérőben**, **C tengelyé sugárban** történik.



4.5-4 ábra

...

N050 T808

N060 G59

(G59 koordinátarendszer kezdőpontja X irányban C forgástengelye)
(síkválasztás X, C; pozicionálás X≠0, C=0 koordinátára)

N070 G17 G0 X200 C0

N080 G94 Z-3 S1000 M3

N090 G12.1

(polárkoordináta interpoláció be)

N100 G42 G1 X100 F1000

N110 C30

N120 G3 X60 C50 I-20 J0

N130 G1 X-40

N140 X-100 C20

N150 C-30

N160 G3 X-60 C-50 R20

N170 G1 X40

N180 X100 C-20

N190 C0

N200 G40 G0 X150

N210 G13.1

(polárkoordináta interpoláció ki)

N220 G0 G18 Z100

(szerszám visszahúzás X, Z síkválasztás)

...

4.6 A hengerinterpoláció (G7.1)

Ha egy henger palástjára vezérpályát kell marni hengerinterpolációt alkalmazunk. Ilyenkor a henger középvonalának és egy forgó tengely forgástengelyének egybe kell esnie.

A programot a **henger középvonalával párhuzamos lineáris tengely** és a **hengert forgató körtengely** programozásával írjuk le. A programban a **forgó tengely** elmozdulását szögben, **fok** egységben adjuk meg, amit a vezérlő átszámít a palást mentén a henger sugarának függvényében ívhosszá, mm-re, vagy inch-re. Az interpoláció után kiadódó elmozdulást visszaalakítja a forgó tengely számára szögel-fordulássá.

A hengerinterpoláció közben megadott **F előtölést** mindig a **henger palástja mentén** veszi figyelembe.

A

G7.1 Qr hengerinterpoláció be-
utasítás bekapcsolja a hengerinterpolációt, ahol

Q: a hengerinterpolációban részt vevő forgó tengely címe

r: a henger sugara.

Ha például a hengerinterpolációban részt vevő forgó tengely a C tengely, és a henger sugara 50 mm, akkor a hengerinterpolációt a G7.1 C50 utasítással kapcsolhatjuk be.

Az ezután következő programrészben egyenes és körinterpoláció megadásával írhatjuk le a henger palástjára marandó pályát. A hossz tengelyen a koordinátát mindig mm-ben, vagy inch-ben, a forgó tengelyen pedig °-ban adjuk meg.

A

G7.1 Q0 hengerinterpoláció ki-
utasítás kikapcsolja a hengerinterpolációt, vagyis a G kód ugyanaz, mint a bekapcsolásé, csak a forgó tengely címére 0-t kell írni.

A fenti példával (G7.1 C50) bekapcsolt hengerinterpolációt a G7.1 C0 utasítással kapcsolhatjuk ki.

A vezérlés bekapcsolás, program vége, vagy reset után mindig hengerinterpoláció ki állapotot vesz fel.

A G7.1 utasítást külön mondatban kell megadni.

A lineáris és a forgó tengely kiválasztása

A hengerinterpoláció bekapcsolása előtt ki kell választani egy lineáris és egy forgó tengelyt, amelyek a hengerinterpolációban részt vevő tengelyek.

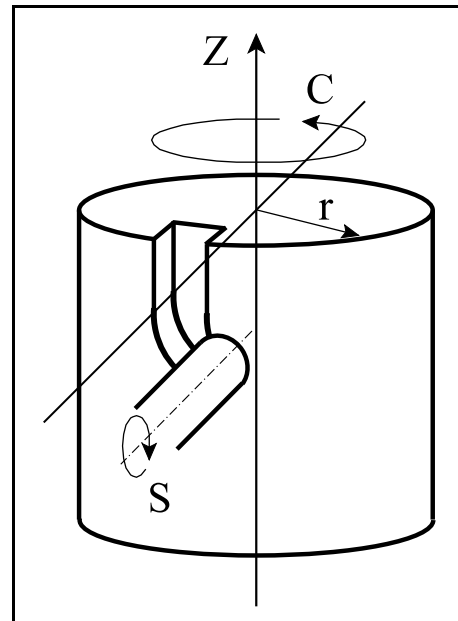
A tengelyek kiválasztása a G17, G18, G19 síkválasztó utasítással történik.

G17 X_{p_}, vagy Y_{p_} és egy forgó tengely címe

G18 X_{p_}, vagy Z_{p_} és egy forgó tengely címe

G19 Y_{p_}, vagy Z_{p_} és egy forgó tengely címe

A hengerinterpoláció a kiválasztott sík és a síkhoz tartozó lineáris tengely alapján értelmezi a G2, G3 körinterpolációs irányokat és a szerszámsugár korrekció (G41, G42) irányát. A körtengely lesz a sík másik tengelye. Párhuzamos tengely is kiválasztható.



4.6-1 ábra

Például, legyen a Z tengely párhuzamos a henger középvonalával és a C tengely legyen a henger forgástengelye. Ekkor a hengerinterpoláció bekapcsolása előtt vagy a

G18 Z_ C_

vagy a

G19 Z_ C_

utasítással jelölhető ki a lineáris és a forgó tengely.

Körinterpoláció

Hengerinterpolációs üzemmódban körinterpoláció megadása lehetséges, azonban csak a sugár R megadásával.

Körinterpoláció a kör középpontjának megadásával (I, J, K) nem lehetséges hengerinterpoláció esetén.

A kör sugara mindig mm-ben, vagy inch-ben kerül értelmezésre, soha nem fokban.

Körinterpolációt például a Z és a C tengely között kétféleképp adhatunk meg:

G18 Z_ C_
G2 Z_ C_ R_
G3 Z_ C_ R_

G19 C_ Z_
G3 C_ Z_ R_
G2 C_ Z_ R_

Ha ugyanazt a pályát akarjuk leírni G18 és G19 síkválasztással a körirányok egymáshoz képest felcserélődnek.

Szerszámsugár korrekció használata hengerinterpoláció esetén

G41, G42 utasítás a megszokott módon használható a hengerinterpoláció bekapcsolt állapotában. Használatára az alábbi megszorítások vonatkoznak:

- A hengerinterpoláció bekapcsolása (G7.1 Qr utasítás) csak G40 állapotban lehetséges,
- Ha a hengerinterpoláció állapotában bekapcsoltuk a G41, vagy G42-t, a hengerinterpoláció kikapcsolása (G7.1 Q0 utasítás) előtt G40-et kell programozni.

Programozási megszorítások a hengerinterpoláció során

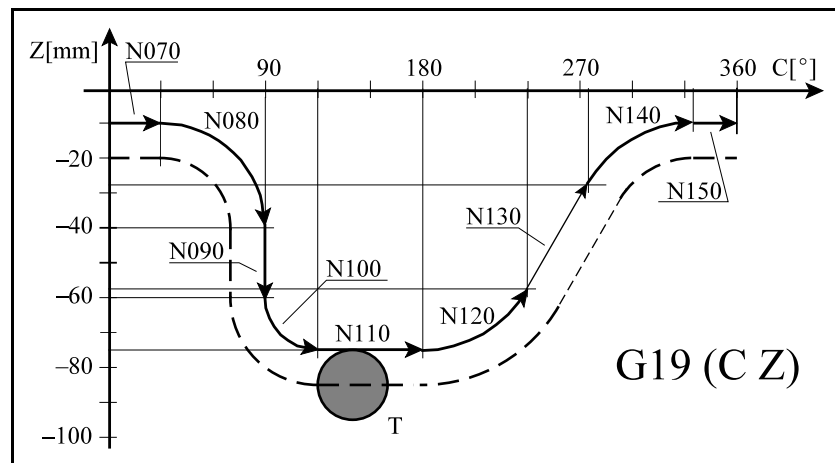
A hengerinterpoláció bekapcsolt állapotában az alábbi utasítások nem használhatók:

- síkváltás: G17, G18, G19,
- koordinátatranszformációk: G52, G92,
- munkadarab koordinátarendszer váltás: G54, ..., G59,
- pozicionálás a gépi koordinátarendszerben: G53,
- körinterpoláció a kör középpontjának (I, J, K) megadásával,
- fűróciklusok.

4.6 A hengerinterpoláció (G7.1)

Mintapélda

Marjunk egy $R=28.65$ mm sugarú henger palástjára 3 mm mélyen, egy, a mellékelt ábrán látható pályát. A T marószerszám párhuzamos az X tengellyel.



4.6-2 ábra

A henger palástján az egy fokra (1°) eső elmozdulás:

$$28.65\text{mm} \cdot \frac{1^\circ}{180^\circ} \cdot \pi = 0.5\text{mm}$$

Az ábrán látható tengelyrendezés G19 síkválasztásnak felel meg.

(HENGERINTERPOLACIO)

...

N030 G19 Z-20 C0

(G19: C-Z sík válsztása)

N040 G1 X51.3 F100

(3 mm mély fogás)

N050 G7.1 C28.65

(hengerinterpoláció bekapcsolása, a forgó tengely: C, a henger sugara 28.65mm)

N060 G1 G42 Z-10 F250

N070 C30

N080 G2 Z-40 C90 R30

N090 G1 Z-60

N100 G3 Z-75 C120 R15

N110 G1 C180

N120 G3 Z-57.5 C240 R35

N130 G1 Z-27.5 C275

N140 G2 Z-10 C335 R35

N150 G1 C360

N160 G40 Z-20

N170 G7.1 C0

(hengerinterpoláció kikapcsolása)

N180 G0 X100

...

5 A koordinátaadatok

5.1 Abszolút és inkrementális programozás (G90, G91), az I operátor, U, V, W cím

A bemenő koordinátaadatok megadhatók abszolút és növekményes értéként is. Abszolút adatmegadásnál a végpont koordinátáit kell a vezérlésnek megadni, míg növekményes adatnál a mondatban megteendő távolságot.

G90: Abszolút adatmegadás programozása

G91: Növekményes adatmegadás programozása

A G90, G91 öröklődő funkciók. Bekapcsolásra az N1300 DefaultG1 paraméter #7 G91 bitje alapján dönti el a vezérlő, hogy melyik állapotot vegye. Program végén, vagy reset hatására szintén a paraméteren beállított kód érvényesül.

Abszolút pozícióra való mozgás csak referenciapontfelvétel után lehetséges.

Példa:

Az ábra alapján kétféleképp lehet a mozgást programozni.

```
G90 G01 X100 Z20
```

```
G91 G01 X60 Z-40
```

Az **I operátor** G90 abszolút adatmegadási állapotban hatásos. **Csak arra a koordinátára vonatkozik, amelyik címe után áll.** Jelentése: inkrementális adat.

A fenti példa a következőképp is megoldható:

```
G90 G01 XI60 ZI-40
```

```
G01 XI60 Z20
```

```
G01 X100 ZI-40
```

Többkarakteres tengelynév használata esetén, ha a név **számra végződik**, például X2 tengely esetén, az I operátort a = jel után kell tenni:

```
X2=I100
```

Ha az **U, V, W címek** nincsenek tengelynek kijelölve akkor felhasználhatók az X, Y, Z irányú inkrementális mozgások jelölésére:

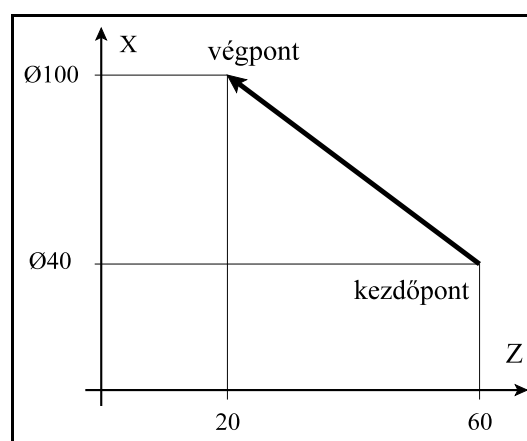
	abszolút parancs címe	inkrementális parancs címe
mozgás parancs X irányban	X	U
mozgás parancs Y irányban	Y	V
mozgás parancs Z irányban	Z	W

A fentiek figyelembe vételével a mintapélda:

```
G90 G01 U60 W-40
```

```
G01 U60 Z20
```

```
G01 X100 W-40
```



5.1-1 ábra

5.2 Inch/Metrikus átalakítás (G20, G21)

A bemenő adatokat megadhatjuk akár metrikus akár inches mértékrendszerben, a megfelelő G kód programozásával.

G20: Inches mértékrendszer választása.

G21: Metrikus mértékrendszer választása.

A kiválasztott mértékrendszer mindaddig érvényben marad amíg ellenkező értelmű parancsot nem adunk ki, tehát G20, G21 öröklődő kódok.

Bekapcsolásra az N1300 DefaultG1 paraméter #3 G20 bitje alapján dönti el a vezérlő, hogy melyik állapotot vegye. Program végén, vagy reset hatására szintén a paraméteren beállított kód érvényesül.

Például:

```
G21 G0 G54 X200 Z50 (pozícionálás X=200, Z=50 mm-re)
```

```
G20 X2 Z1 (pozícionálás X=2, Z=1 inch-re)
```

A következő tételekre van hatással a mértékrendszer megváltoztatása:

- Koordináta- és korrekcióadatok, (mm/inch)
- Előtolás (mm/min, inch/min, mm/ford, inch/ford),
- Konstans vágósebesség (m/min, feet/min),
- A pozíció-, korrekció-, nullponteltolás és előtolásértékek mindig a kiválasztott mértékrendszerben jelennek meg a kijelzőn,
- A makrováltozók (eltolás, pozícióadatok stb.) olvasásakor az adatok a kiválasztott mértékegységben kerülnek kiolvasásra,
- Inkrementális jog és kézikerek lépésmagysága,
- Kézi mozgatás (jog) előtolása.

5.3 Átmérőben, vagy sugárban történő programozás

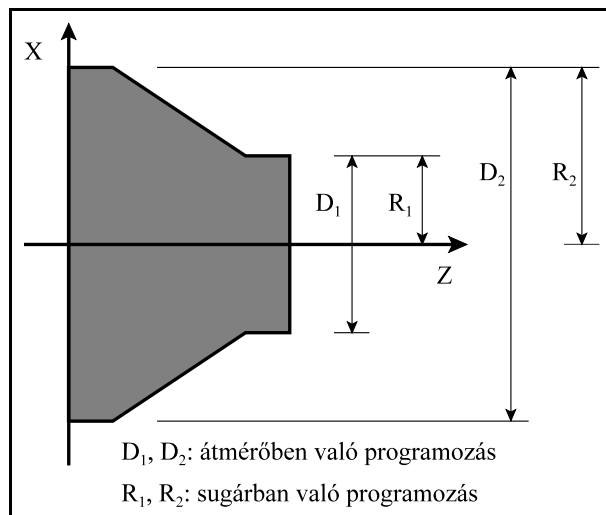
Mivel esztergákon a megmunkált munkadarabok metszete általában kör, a darab átmérő mérésével ellenőrizhető, az X tengely irányú méreteket célszerű átmérőben megadni. Ha Y tengely is van a gépen, általában az Y irányú méretet is célszerű átmérőben kezelni. Azt, hogy egy adott tengelyen a méretet átmérőben, vagy rádiuszban értelmezze a vezérlő az N0106 Axis Properties paraméter #0 DIA bitjén adhatjuk meg, tengelyenként:

rádiuszban történő programozás esetén:

#0 DIA=0

átmérőben történő programozás esetén:

#0 DIA=1



5.3-1 ábra

Ha a paramétert átmérő programozására állítottuk a következőket vegyük figyelembe:

eset	megjegyzés
abszolút mozgásparancs	átmérőben adjuk meg
inkrementális mozgásparancs	átmérőben adjuk meg (ábránkon $D_2 - D_1$)
koordináta és nullpont eltolás	átmérőben adjuk meg
szerszámhossz korrekció	átmérőben adjuk meg
Átmérőben kezelt tengelyre vonatkozó paraméterek ciklusokban, mint pl. fogásmélység	mindig sugárban adjuk meg
körinterpoláció megadásánál R és I, J, K értéke	mindig sugárban adjuk meg
tengely pozíció kijelzése	átmérőben történik
Átmérőben kezelt tengely irányú eltolás	mindig sugár/ford, vagy sugár/perc
lépésméret léptetés és kézikerek üzemmódban	1 inkremens választása esetén $1\mu\text{m}$ -t lép átmérőben

5.3.1 Sugárban/átmérőben való programozás átkapcsolása (G10.9)

Az egyes tengelyeken a sugárban, vagy átmérőben való programozás módját az

N0106 Axis Properties paraméter #0 DIA

állása határozza meg. Ha a DIA bit értéke:

=0: sugárban,

=1: átmérőben

történik a tengely programozása.

Alkatrészprogramban tetszőleges helyen át lehet kapcsolni a programozás módját. Azt, hogy az átkapcsolás PLC-ből, vagy G kóddal történik az

N0106 Axis Properties paraméter #5 MGD

bitje határozza meg. Ha a bit értéke

=0: a sugárban/átmérőben való programozás átkapcsolása PLC jelzón keresztül,

=1: a sugárban/átmérőben való programozás átkapcsolása G10.9 kóddal történik.

A PLC-ből való átkapcsolás mikéntjét a gép építője határozza meg.

A továbbiakban a G10.9 kóddal való átkapcsolást írjuk le.

A

G10.9 v

utasítás kapcsolja át a sugárban, vagy átmérőben való programozást, ahol

v: tetszőleges lineáris tengelyek címe: X, Y, ... stb.

Ha a tengelycímre írt érték:

=0: a tengelyt sugárban, ha

=1: a tengelyt átmérőben programozzuk.

Például marógépen az X és Y tengelyt alapesetben sugárban programozzuk. Ha az alkatrészprogram egy adott szakaszán átmérőben célszerű megadni X és Y értékét, az alkatrészprogramba ezt kell írni:

... (X, Y sugárban megadva)

...

G10.9 X1 Y1 (átmérő programozása)

```

... (X, Y átmérőben megadva
...
G10.9 X0 Y0 (sugár programozása)
... (X, Y sugárban megadva)
...

```

Ha esztergán X tengelyt átmérőben programozzuk, de polárinterpolációval X, C tengellyel végzett marásnál X-et sugárban akarjuk programozni:

```

... (X átmérőben megadva, esztergálás)
...
G10.9 X0 (sugár programozása)
G12.1 (polárinterpoláció be)
... (X, sugárban megadva, marás)
...
G13.1 (polárinterpoláció ki)
G10.9 X1 (átmérő programozása)
... (X átmérőben megadva, esztergálás)
...

```

A G10.9 kódot mindig külön mondatban kell megadni!

Program végén, vagy reset hatására a G10.9 által beállított programozási mód törlődik és a vezérlő az N0106 Axis Properties paraméter #0 DIA bitjén beállított alaphelyzetet veszi fel.

5.4 Adatmegadás polárkoordinátákkal (G15, G16)

A programozott koordináták értékei polárkoordinátákkal, azaz mm-ben, vagy inch-ben megadott sugárral és fokban megadott szöggel is bevíhetők.

G16: Adatmegadás polárkoordinátákkal

G15: Adatmegadás polárkoordinátákkal kikapcsolása

G15, G16 öröklődő funkciók. Bekapcsolás után, program végén és resetre a vezérlő G15 állapotba kerül.

A polárkoordinátában megadott adatok a G17, G18, G19 által meghatározott síkban kerülnek értelmezésre. Adatmegadáskor a sík első tengelyének címét tekinti a sugárnak, a második tengelyét pedig a szögnek.

G17 X_p (sugár) Y_p (szög)

G18 Z_p (sugár) X_p (szög)

G19 Y_p (sugár) Z_p (szög)

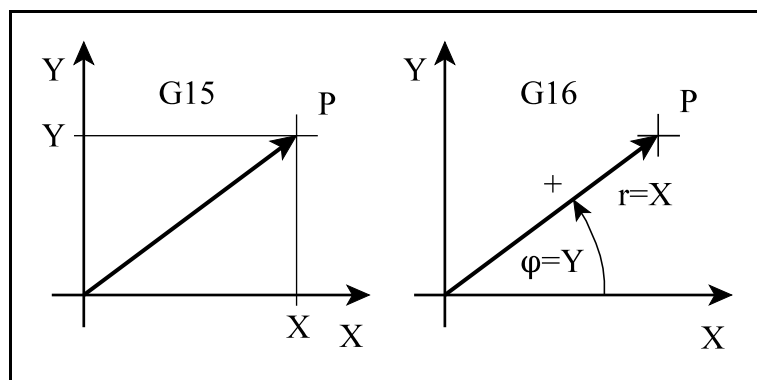
X_p , Y_p , Z_p lehet az X, Y, Z alap-tengely, vagy azzal párhuzamos tengely.

A szög adatmegadása esetén az *óramutató járásával ellentétes irány* a szög *pozitív* iránya, az *óramutató járásával megegyező irány* a szög *negatív* iránya.

A többi, a kiválasztott síkon kívül

első tengely adatait Descartes (derékszögű) koordinátában megadott adatnak veszi.

A sugarat és a szöget is meg lehet adni abszolút és növekményes adatként is.



5.4-1 ábra

Amikor a **sugarat abszolút** adatként adjuk meg, az aktuális koordinátarendszer origója lesz a polárkoordinátarendszer kezdőpontja:

```
G90 G16 X100 Y60
```

A szög is és a sugár is abszolút adat, a szerszám a 100 mm sugarú 60°-os pontra fut.

```
G90 G16 X100 YI40
```

A szög növekményes adat. A szerszám a 100mm sugarú pontra fut, az előző szöghelyzethez képest 40°-kal arrébb.

Amikor a **sugarat növekményes** értéként definiáljuk, a tengelyek mondat eleji pozíciójától számítva mozogja le a megadott sugarat a megadott szög irányában:

```
G90 G16 XI50 Y60
```

A kezdőponttól számítva méri az 50 mm sugarat és a 60°-os abszolút szögre mozog.

```
G91 G16 X50 Y40
```

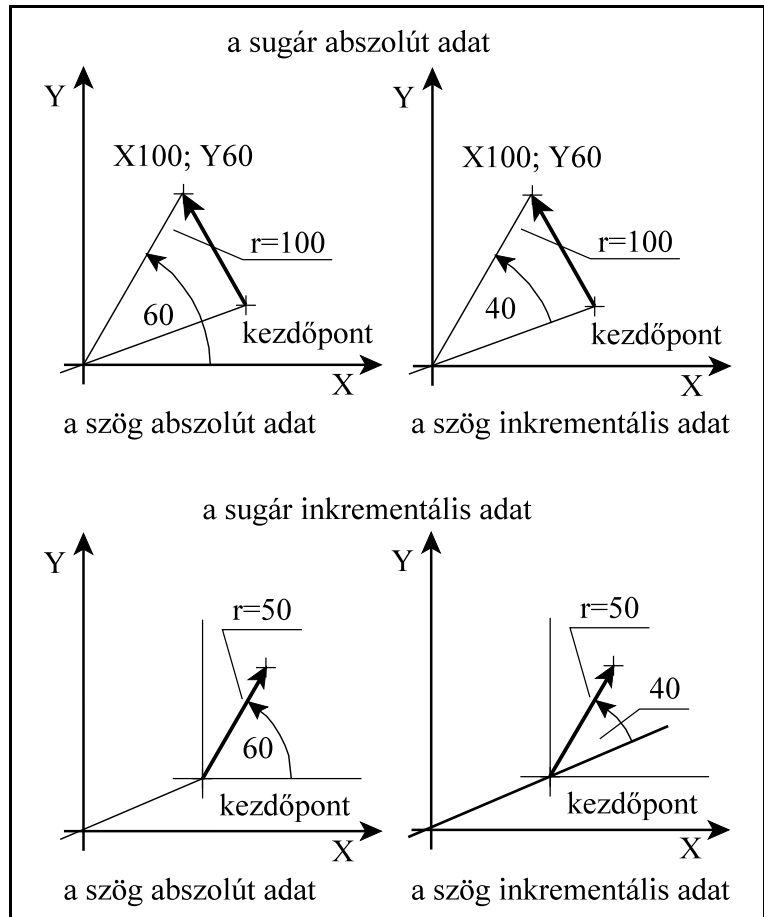
A kezdőponttól számítva méri az 50 mm sugarat és a kezdőszögtől mért 40°-os szögre mozog.

G16 polárkoordinátás adatmegadás bekapcsolt állapotában is lehet kört programozni. A kört meg lehet adni sugárral és I, J, K-val is. Az utóbbi esetben azonban **I, J, K** címet **mindig derékszögű adatnak** tekinti a vezérlés.

Ha az aktuális koordinátarendszer középpontja egybeesik a kör középpontjával polárkoordinátás adatmegadással többfordulatú kör, vagy spirális is programozható:

```
G17 G16 G90 G02 X100 Y-990 Z50 R-100
```

A fenti mondatban egy 2 egész $\frac{3}{4}$ fordulatú spirált adtunk meg, az óramutató járásával megegyező forgásiránnyal. Többfordulatú kör programozásakor ügyeljünk, hogy G2 irány esetén negatív polárszöget, G3 irány esetén pedig pozitív polárszöget programozzunk.



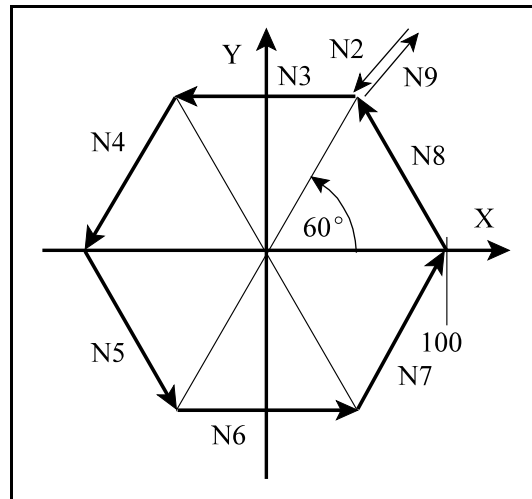
5.4-2 ábra

A következő utasításokban előforduló címeket a vezérlő nem tekinti polárkoordináta adatoknak, még ha a G16 állapot be is van kapcsolva:

- G10 beállító utasításban előforduló koordináták,
- G52 koordinátaeltolás,
- G92 koordinátabeállítás,
- G28, G30 közbülső pont koordinátái,
- G53 pozicionálás a gépi koordinátarendszerben,
- G68 koordinátarendszer elforgatás középpontja,
- G51 léptékezés középpontja,
- G50.1 tükrözés középpontja.

Mintapélda: hatszög marása

```
N1 G90 G17 G0 X120 Y120 F120  
N2 G16 G1 X100 Y60  
N3 Y120  
N4 Y180  
N5 Y240  
N6 Y300  
N7 Y0  
N8 Y60  
N9 G15 G0 X120 Y120
```



5.4-3 ábra

5.5 Koordinátaadatok megadása és pontossága

Az alkalmazott mértékrendszer függvényében értelmezi a tizedespontot:

- X2.134 jelentése 2.134 mm, vagy 2.134 inch,
- B24.236 jelentése 24.236 fok, ha B címen szögadatot adunk meg.

A tizedespont használata nem kötelező:

- X325 jelentése például 325 mm.

A vezető nullák elhagyhatók:

- .032=0.032

Tizedespont után a követő nullák elhagyhatók:

- 0.320=.32

Koordinátaadatokat maximum 15 decimális számjegy pontossággal lehet megadni.

5.6 Forgó tengelyek átfordulás kezelése

Ez a funkció forgó tengelyek esetén használható, vagyis, ha egy tengelycím (pl. C) forgó tengelynek van kijelölve.

Átfordulás kezelésem azt értjük, hogy az adott tengelyen a pozíciót nem plusz mínusz végtelen között tartja nyilván a vezérlő, hanem a tengely periódikusságát figyelembe véve, pl.: 0° és 360° között.

Tengely kijelölése forgó tengelynek és ennek a hatása

Ezt a kijelölést az N0106 Axis Properties paraméter #1 ROT=1 bitállítással kell végezni. Forgó tengelyre

- a vezérlő **nem végzi el az inch/metrikus** konverziót,
- a forgó tengelyre **engedélyezhető az átforduláskezelés**.

Az átforduláskezelés engedélyezése

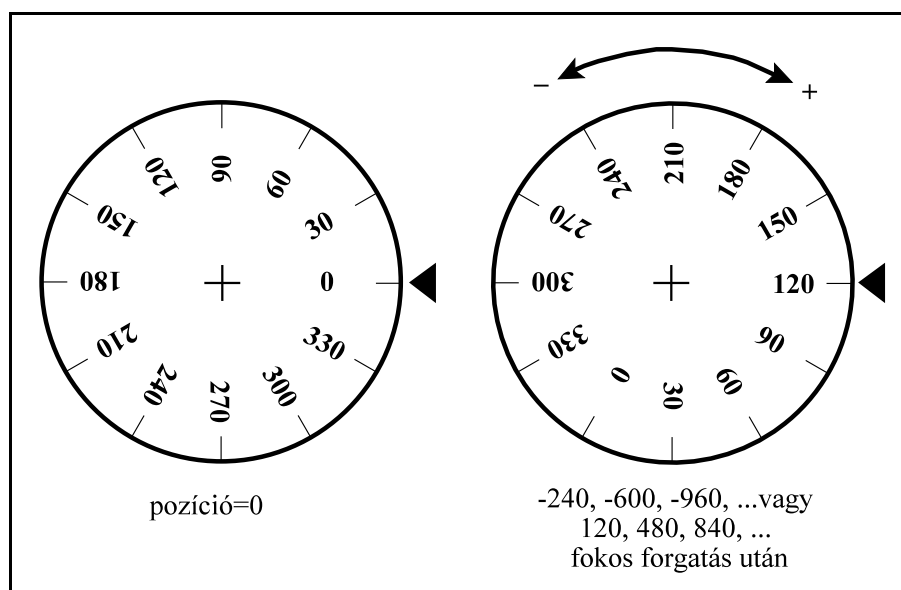
A funkciót az N0107 RollOver Control paraméter #0 REN=1 bitállítással élesíti. Ha a REN bit értéke:

- =0: a forgó tengelyt úgy kezeli, mint a lineáris tengelyeket, és a további paraméterek kitöltése hatástalan,
- =1: a forgó tengelyre alkalmazza az átfordulás kezelést, amelynek lényegét az alábbiakban leírtak határozzák meg.

Az egy fordulatra eső út megadása

Az N0108 RollAmount paraméteren adjuk meg a tengely egy körülfordulására eső utat. Tehát, ha a tengely egy körülfordulására 360° -ot forog, a megfelelő paraméterre írandó érték: 360. A rendszer tetszőleges periodicitást tud kezelni, nem csak 360 fokosat.

A fenti paraméterbeállításokkal a forgó tengely pozícióját mindig a 0° - $+359.999^\circ$ közötti tartományban jelzi ki a vezérlő, függetlenül attól, hogy melyik irányban forgott, és hány fordulatot tett meg a forgó tengely.



5.6-1 ábra

Az ezután következő beállítási lehetőségek felsorolása feltételezi az N0107 RollOver Control paraméter #0 REN=1 bitállást és az N0108 RollAmount paraméter beállítását.

Forgatás a megadott pozícióra: a forgatás alapesete

Az N0107 RollOver Control paraméter

#2 ABS=0 és #1 ASH=0

beállítása esetén, abszolút programozáskor (G90) a forgatás a megadott pozícióra történik. Az elmozdulás nagysága, mindig a RollAmount paraméteren megadott értéknél kisebb lesz, tehát az egész fordulatokat mindig levágja a mozgás során.

Példa:

Legyen a C tengely kiinduló pozíciója

C=0

A

G90 C210

G90 C570

G90 C930

...

utasításokra mindig pozitív rányban mozog 210 fokot ($\Delta C=210^\circ$), az egész fordulatokat levágva, és a végponti pozíció 210° lesz.

Legyen a C tengely kiinduló pozíciója

C=0

A

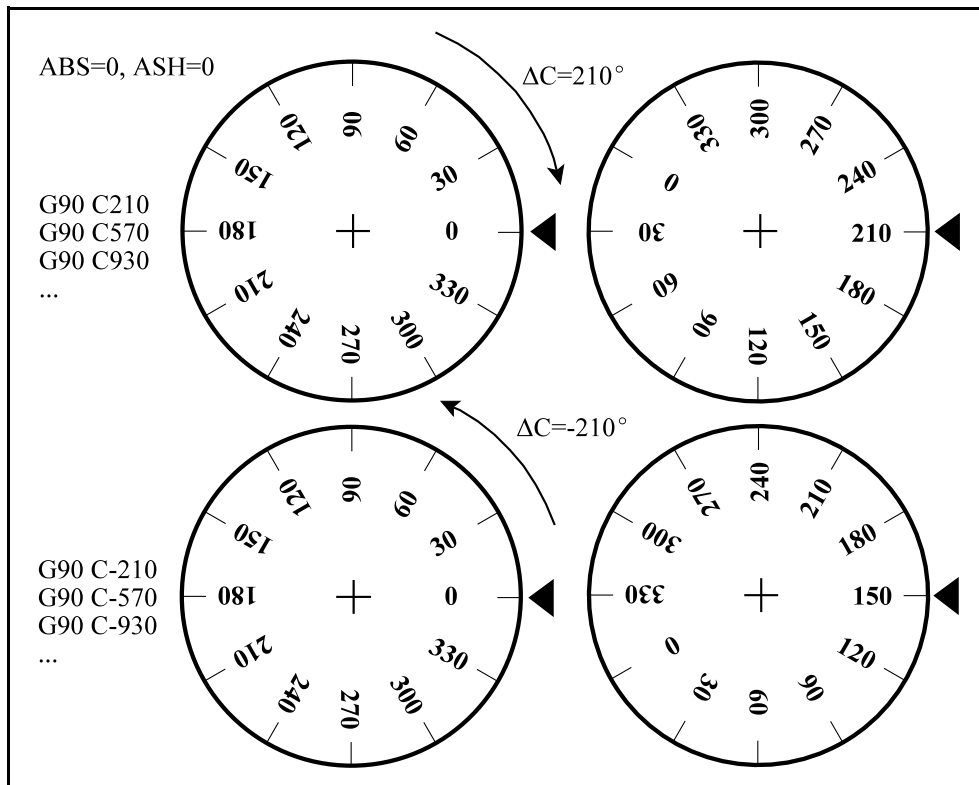
G90 C-210

G90 C-570

G90 C-930

...

utasításokra mindig negatív rányban mozog 210 fokot ($\Delta C=-210^\circ$), az egész fordulatokat levágva, és a végponti pozíció $150^\circ (= -210^\circ)$ lesz.



5.6-2 ábra

Forgatás a megadott pozícióra a rövidebb úton

Az N0107 RollOver Control paraméter

#2 ABS=0 és #1 ASH=1

beállítás esetén, abszolút programozáskor (G90) a forgatás a megadott pozícióra történik és a rövidebb úthoz tartozó irányban forogva. Az elmozdulás nagysága, mindig a RollAmount paraméteren megadott értéknél kisebb lesz, tehát az egész fordulatokat mindig levágja a mozgás során.

Példa:

Legyen a C tengely kiinduló pozíciója

C=0

A

```
G90 C210
G90 C570
G90 C930
```

...

utasításokra negatív irányban (a rövidebb úton) mozog 150 fokot ($\Delta C = -150^\circ$), az egész fordulatokat levágva, és a végponti pozíció 210° lesz.

Legyen a C tengely kiinduló pozíciója

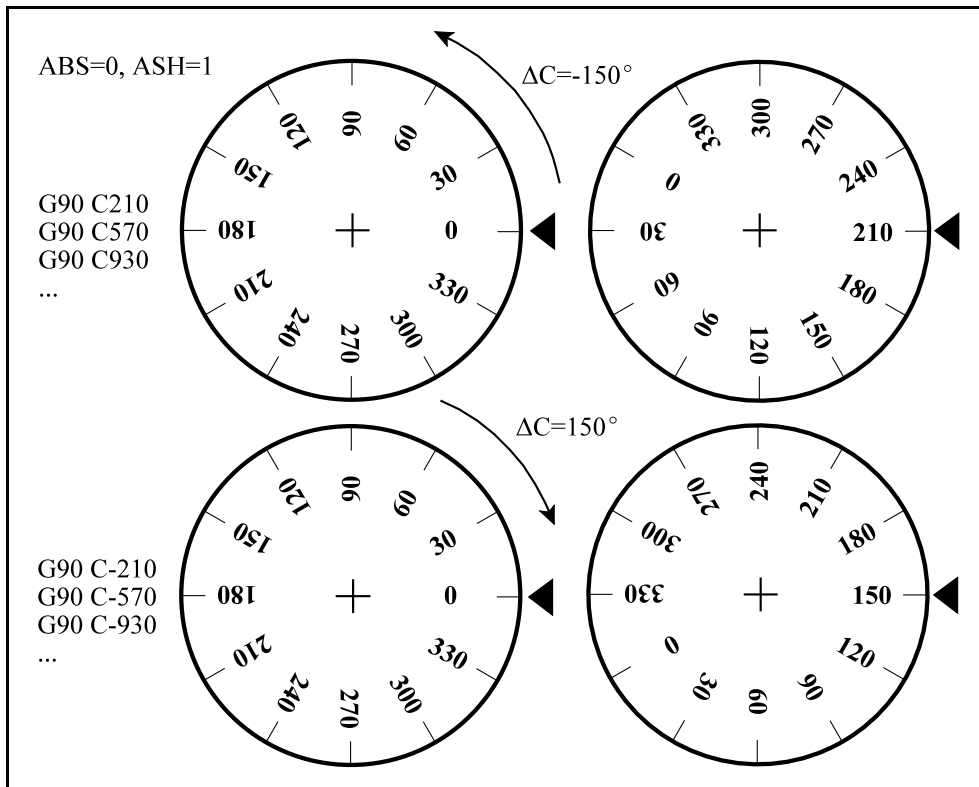
C=0

A

```
G90 C-210
G90 C-570
G90 C-930
```

...

utasításokra pozitív irányban (a rövidebb úton) mozog 150 fokot ($\Delta C=150^\circ$), az egész fordulatokat levágva, és a végponti pozíció $150^\circ (= -210^\circ)$ lesz.



5.6-3 ábra

Forgatás a megadott pozíció abszolút értékére, az előjel szerinti irányban

Az N0107 RollOver Control paraméter

#2 ABS=1 és #1 ASH=0

beállítása esetén, abszolút programozáskor (G90) a forgatás a megadott pozíció abszolút értékére, az előjel szerinti irányban történik. Az elmozdulás nagysága, mindig a RollAmount paraméteren megadott értéknél kisebb lesz, tehát az egész fordulatokat mindig levágja a mozgás során.

Példa:

Legyen a C tengely kiinduló pozíciója

C=0

A

G90 C30
G90 C390
G90 C750
...

utasításokra mindig pozitív irányban mozog 30 fokot ($\Delta C=30^\circ$), az egész fordulatokat levágva, és a végponti pozíció 30° lesz.

Legyen a C tengely kiinduló pozíciója

C=0

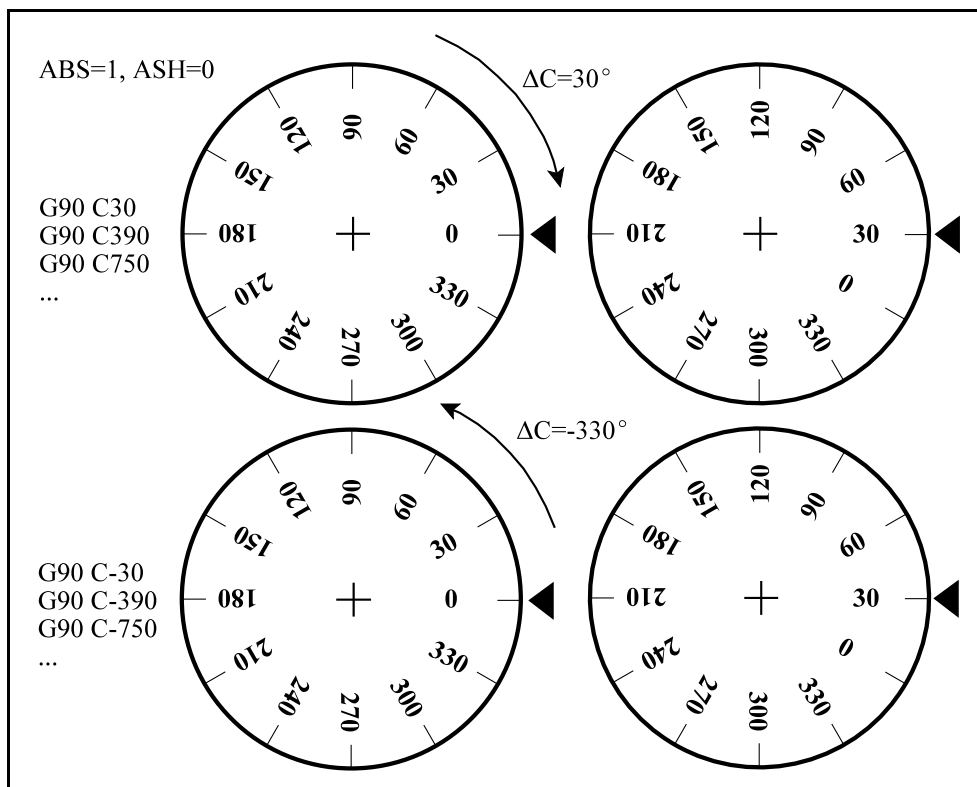
A

G90 C-30
G90 C-390

G90 C-750

...

utasításokra mindig negatív rányban mozog 330 fokot ($\Delta C = -330^\circ$), az egész fordulatokat levágva, és a végponti pozíció 30° lesz.



5.6-4 ábra

☞ **Megjegyzés:** A 0 (nulla) pozitív szám! Tehát 0 pozíciót programozva pozitív irányban fog mozogni.

Legyen a C tengely kiinduló pozíciója

C=30

A

G90 C0

utasításra pozitív irányban forog 330 fokot ($\Delta C = 330^\circ$), és a végponti pozíció 0 fok lesz.

Ha negatív irányban akarunk 30 fokról 0 fokra mozogni programozzuk a

G90 C-360

mondatot. Ekkor negatív irányban mozog 30 fokot ($\Delta C = -30^\circ$), és a végponti pozíció 0 fok lesz.

Forgó tengely mozgása inkrementális programozás esetén

Inkrementális adatmegadás programozása esetén az elmozdulás iránya mindig a programozott előjel szerint történik.

Az N0107 RollOver Control paraméter

REN=0

állása esetén az inkrementálisan programozott útra nem alkalmazza a Roll Amount paramétert, tehát inkrementális megadással több fordulatnyi elmozdulás is programozható.

Legyen a C tengely kiinduló pozíciója

C=0

A

G91 C30 (elmozdulás 30 fok)
G91 C390 (elmozdulás 390 fok)
G91 C750 (elmozdulás 750 fok)

...

mondatok végrehajtásakor nem vágja le az egész fordulatokat. A pozíciókijelzőn C tengely pozíciója viszont mindig 30 fok lesz a mozgás végén, mert a pozíciókijelzésre mindig érvényes a RollAmount paraméteren beállított érték.

Az N0107 RollOver Control paraméter

REN=1

állása esetén az inkrementálisan programozott útra is alkalmazza a Roll Amount paramétert, tehát inkrementális megadással sem programozható egy fordulatnál nagyobb elmozdulás.

Legyen a C tengely kiinduló pozíciója

C=0

A

G91 C30 (elmozdulás 30 fok)
G91 C390 (elmozdulás 30 fok)
G91 C750 (elmozdulás 30 fok)

...

mondatok végrehajtásakor levágja az egész fordulatokat. A pozíciókijelzőn C tengely pozíciója viszont mindig 30 fok lesz.

6 Az előtolás

6.1 A gyorsmeneti előtolás

A pozícionálás gyorsmenettel történik G0 parancs hatására. A G0 pozícionáláson kívül gyorsmeneti előtolással mozognak a G53, G28, G30 paramcsok, illetve a ciklusok pozícionáló fázisai is. A pozícionálási gyorsmenet tengelyenkénti értékét **a gép építője paraméteren állítja be mm/min, inch/min, vagy fok/min egységben**. A gyorsmenet nagysága tengelyenként különböző lehet.

Ha több tengely végez egyidejűleg gyorsmeneti mozgást, az eredő előtolás értékét a vezérlő úgy számítja ki, hogy a tengelyekre vetített sebességkomponens egyik tengelyen se haladja meg az arra a tengelyre érvényes, paraméterben megadott gyorsmeneti értéket, és a pozícionálás minimális idő alatt menjen végbe.

A gyorsmeneti előtolás nagyságát módosítja a gyorsmeneti százalék (override) kapcsoló.

A gyorsmeneti override működését a gép építője határozza meg a PLC programban. Leírását keresse a gép építője által kiadott kézikönyvben.

100% fölé nem megy a gyorsmeneti override értéke.

A gyorsmeneti előtolást mindig leállítja az előtolás százalék kapcsoló 0% állása.

Érvényes referenciapont híján a paramétermezőben a gép építője által definiált csökkentett gyorsmeneti értékek lesznek érvényben tengelyenként, mindaddig, amíg a referenciapontfelvétel meg nem történt.

A tengelymozgató (jog) gombokkal végzett szánmozgatáskor a gyorsjárat sebesség a pozícionálási gyorsmenettől különböző, szintén paraméteren beállított, tengelyenként különböző érték. Értelemszerűen a pozícionálási sebességnél kisebb érték, hogy a megálláshoz az emberi reakcióidőt is be lehessen kalkulálni.

6.2 A munkaelőtolás

Az előtolást

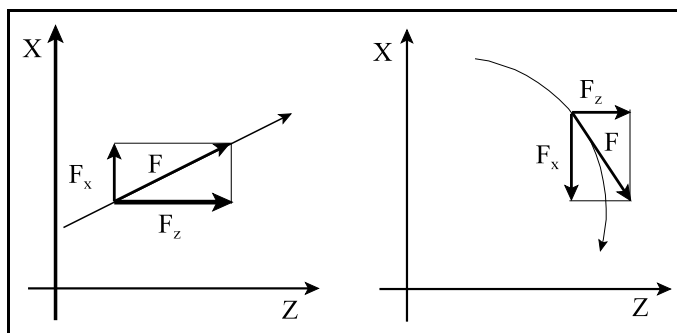
F címen

programozzuk.

A programozott előtolás lineáris- (G01) és körinterpolációs (G02, G03) mondatokban érvényesül.

Az **előtolás mértékegységét a G94, G95** kódok határozzák meg.

Az előtolást a programozott pálya mentén tangenciálisan számítja a vezérlő.



6.2-1 ábra

F : előtolás érintő irányú nagysága (programozott érték)

F_x : előtoláskomponens az X irányban

F_z : előtoláskomponens a Z irányban

$$F = \sqrt{F_x^2 + F_z^2}$$

A programozott előtolást az előtolásszázalék (override) kapcsolóval lehet módosítani kivéve a G63, százalékkapcsoló és stop tiltás, állapotot.

Az előtolás override működését a gép építője határozza meg a PLC programban. Leírását és érték határát keresse a gép építője által kiadott kézikönyvben.

Az előtolás öröklődő érték.

Bekapcsolás után a vezérlő paraméteren meghatározott értéket vesz fel. G94 állapotban az N0300 Default F G94, G95 állapotban az N0301 Default F G95 paraméterről veszi az előtolás kezdeti értékét.

Az adott gépen programozható maximális előtolást a gép gyártója a paramétermezőben tengelyenként bekorlátozza. Az itt beállított érték mindig percenkénti dimenzióban értelmezett. Ez az érték egyben a SZÁRAZ FUTÁS kapcsoló bekapcsolt állapotában az előtoló mozgások sebessége. Az előtolás értékét maximum 15 decimális számjegyre pontossággal lehet megadni. Tizedespont használható. Mindig pozitív szám.

6.2.1 Percenkénti (G94) és fordulatonkénti (G95) előtolás

Az előtolás mértékegységét a G94 és G95 kódokkal lehet megadni a programban:

G94: percenkénti előtolás

G95: fordulatonkénti előtolás

Percenkénti előtoláson a mm/perc, inch/perc, vagy fok/perc dimenzióban megadott előtolást értjük.

Fordulatonkénti előtoláson az egy főorsó fordulatra végzett előtolást értjük mm/ford, inch/ford, vagy fok/ford dimenzióban.

Öröklődő értékek.

Bekapcsolás, reset, vagy program vége után az N1301 DefaultG2 paraméter #0 G95 bitje dönti el, hogy a vezérlő G94, vagy G95 állapotot vesz fel.

A G94/G95 állapot nem befolyásolja a gyorsmeneti előtolást, az mindig percenkénti dimenzióban értendő.

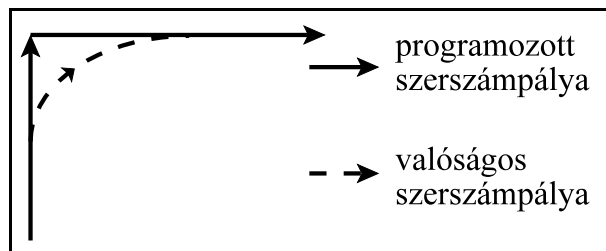
6.3 Az előtolásvezérlő funkciók

Az előtolásvezérlő funkciókra

- *sarkok megmunkálásakor* van szükség,
- illetve olyan esetben, amikor a technológia azt kívánja meg, hogy az *override* illetve *stop* kapcsolók *tiltva* legyenek.

Sarkok megmunkálásakor, ha folyamatos forgácsolás üzemmódot használunk, a szánok tehetetlenségük folytán nem képesek követni a vezérlő által kiadott útparancsokat. Ekkor a szerszám az előtolás függvényében kisebb, vagy nagyobb mértékben lekerekíti a sarkot.

Ha a munkadarabnál éles sarkokra van szükség, akkor a vezérlővel közölni kell, hogy a mozgás végén lassítson le, várja meg amíg a tengelyek megállnak, és a következő mozgást csak ezután indítsa.



6.3-1 ábra

6.3.1 Pontos megállás a mondat végén (G9)

Ez a funkció *nem öröklődik*, csak abban a mondatban érvényes amelyikben programozták.

Annak *a mondatnak a végén*, ahol megadásra került, a vezérlő az interpoláció végrehajtása után lelassít, *megáll és megvárja a mérőrendszer pozícióban jelét*.

Ha a jel egy paraméteren beállított idő eltelte után sem jön be, *2501 Pozícióhiba* üzenetet ad a vezérlő.

Ez a funkció éles sarkok pontos kerülésére szolgál.

6.3.2 Pontos megállás üzemmód (G61)

Öröklődő funkció. Törlésre kerül G62, G63, G64 paranccsal.

A vezérlő *minden mondat végén* lelassít, *megáll és megvárja a mérőrendszer pozícióban jelét*, és csak ezután indítja a következő interpolációs ciklust.

Ha a jel egy paraméteren beállított idő eltelte után sem jön be, *2501 Pozícióhiba* üzenetet ad a vezérlő.

Ez a funkció éles sarkok pontos kerülésére szolgál.

6.3.3 Folyamatos forgácsolás (G64)

Öröklődő funkció. *Bekapcsolás, reset, program vége után a vezérlő ezt az állapotot veszi fel.* A következő kódok szüntetik meg ezt az állapotot: G61, G62, G63.

Ebben az üzemmódban az interpoláció végrehajtása után nem áll meg a mozgás, nem várják meg a mérőrendszer pozícióban jelét a tengelyek, hanem azonnal elkezdődik a következő mondat interpolációja.

Ebben az üzemmódban éles sarkokat nem lehet megmunkálni, mert az átmenetknél lekerekíti azokat.

6.3.4 Override és stop tiltás (G63)

Öröklődő funkció. A G61, G62, G64 kódok megszüntetik ezt az állapotot.

Ebben az üzemmódban *az előtolás- és főorsó százalékkapcsoló (override), valamint az előtolás stop hatástalan.* A százalék értékeket függetlenül azok állásától 100%-nak veszi. Az interpoláció végrehajtása után nem lassít le, hanem azonnal indítja a következő interpolációs ciklust.

Ezt az üzemmódot különböző menetmegmunkálások esetén lehet használni.

6.3.5 Automatikus előtolásnövelés belső sarkoknál (G62)

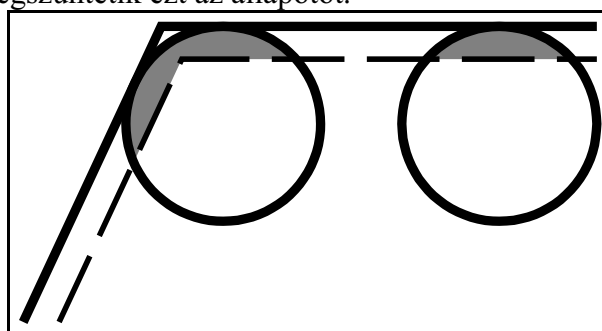
Öröklődő funkció. A G61, G63, G64 kódok megszüntetik ezt az állapotot.

Belső sarkok megmunkálása esetén a szerszámra ható erő megnövekszik a sarok előtti és utáni szakaszon. Annak érdekében, hogy a szerszám ne rezegjen be, és a felület megfelelő maradjon, a vezérlő G62 bekapcsolt állapotában a belső sarkok előtti és utáni szakaszon az előtolást automatikusan csökkenti.

Az előtolásnövelés a következő feltételek mellett hatásos:

- 1. a síkbeli szerszámsugár korrekció bekapcsolt állapotában (G41, G42),
- 2. G1, G2, G3 mondatok között,
- 3. a kiválasztott síkban végzett mozgásoknál,
- 4. ha a sarkot belülről kerüli a szerszám,
- 5. ha a sarok szöge kisebb, mint egy paraméteren meghatározott szög,
- 6. a sarok előtt, és után paraméteren meghatározott távolságra.

Az előtolásnövelés funkció mind a négy lehetséges átmenetre: egyenes-egyenes, egyenes-kör,



6.3.5-1 ábra

kör-egyenes, kör-kör működik.

A φ belső szög értékét az N1409 CornAngle paraméteren lehet beállítani az 1–180° szögtartományban.

A sarok előtt L_d távolságra kezd lassítani, a sarok után pedig L_a távolságra gyorsítani. Körívek esetén L_d és L_a távolságot az ív mentén veszi figyelembe a vezérlő.

L_d távolság megadása a N1407 DecDist paraméteren, L_a távolságé pedig az N1408 AccDist paraméteren történik.

Az N1410 CornOver paraméteren 0 és 1 közötti viszonyzámmal meg lehet adni, hogy milyen mértékig csökkentse a [6.3.5-2](#) ábra vezérlő az előtolást belső sarkoknál t. Az előtolás

$$F * \text{CornOver}$$

lesz, ahol F a programozott előtolás. Az így kapott előtolásra még az override kapcsoló is hatással van.

Ha G62 állapotban pontos megállást akarunk programozni az adott mondatba G09-et kell írni.

6.4 Automatikus előtoláscsökkentés belső körívknél

A síkbeli szerszámsugár korrekció bekapcsolt állapotában (G41, G42), körinterpoláció során a programozott előtolás a szerszám középpontja mentén határos. Körívek belső megmunkálásakor a vezérlő automatikusan csökkenti az előtolás értékét, hogy a forgácsolási sugáron legyen határos a programozott előtolás. Az előtolás nagysága a szerszám-sugár középpontján:

$$F_c = \frac{R_c}{R} F$$

ahol F_c : a szerszámsugár középpont előtolása (korrigált előtolás)

R: a programozott körsugár

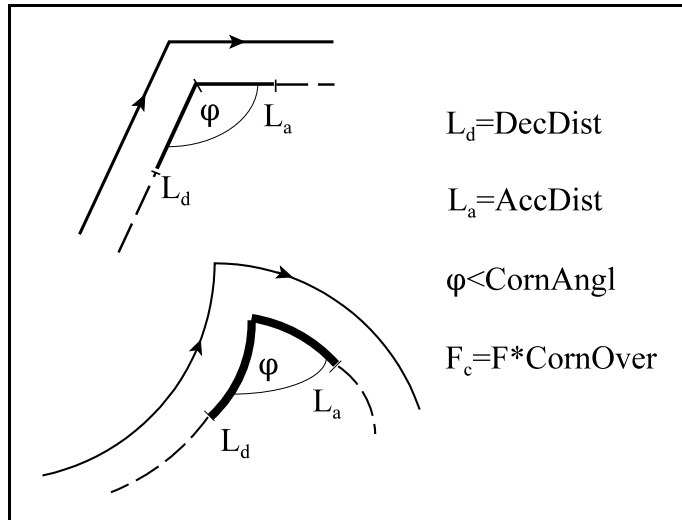
R_c : a korrigált körsugár

F: a programozott előtolás.

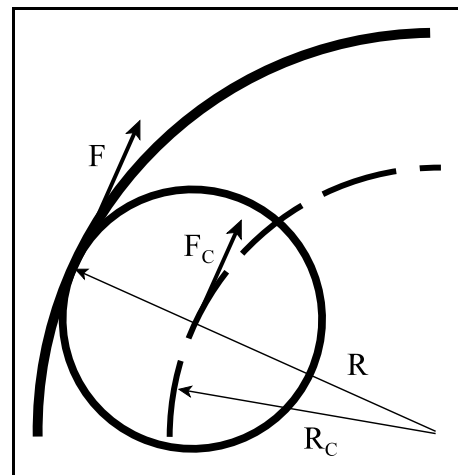
Az automatikus előtoláscsökkentésnek alsó határt szab az N1406 CircOver paraméter, ahol 0 és 1 közötti viszonyzámmal meg lehet adni az előtolás csökkentés minimumát, vagyis az aktuális előtolásra az

$$F_c \geq F * \text{CircOver}$$

feltétel teljesül. A körsugár miatti override összeszorozódik az előtolás és sarokoverride értékekkel, és így kerül kiadásra.



[6.3.5-2](#) ábra



[6.4-1](#) ábra

7 A gyorsulás

A **gyorsulás a sebesség időbeli változásának mértéke**. Minél rövidebb idő alatt érünk el egy adott sebességet annál nagyobb a gyorsulás.

Minél nagyobb gyorsulást akarunk elérni, annál nagyobb teljesítményű motorokra és hajtásokra van szükség.

Mozgás közben a gépre ható erők nagysága, végső soron a gép igénybevétele, a gépen fellépő gyorsulással egyenesen arányos.

A gyorsítás paramétereit a gép építője a fenti két szempont alapján állítja be tengelyenként.

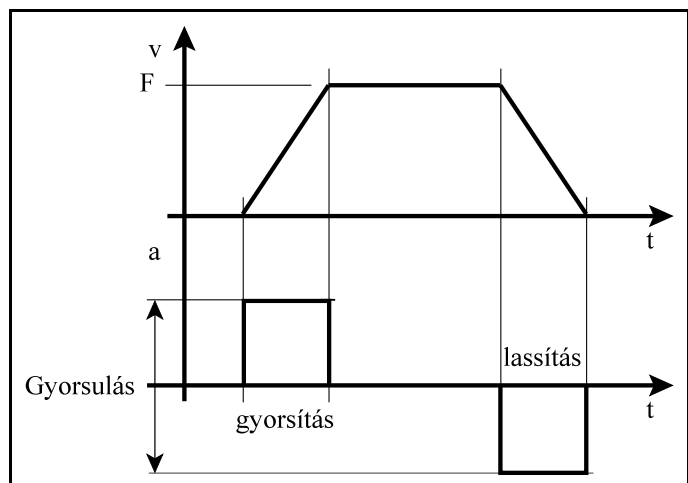
A vezérlő mindig a pálya menti (vektoriális) előtolás értékét gyorsítja. A gyorsítás értékét úgy számítja ki, hogy a tengelyekre eső gyorsuláskomponensek nagysága egyik tengelyen se haladja meg a tengelyre beállított értéket.

Kétféle gyorsítás állítható be:

- lineáris és
- haranggörbe alakú.

Lineáris gyorsítás esetén a gyorsítás, illetve a lassítás alatt a gyorsulás értéke állandó, a vezérlő az előtolást lineáris függvény szerint növeli induláskor, illetve csökkenti megálláskor.

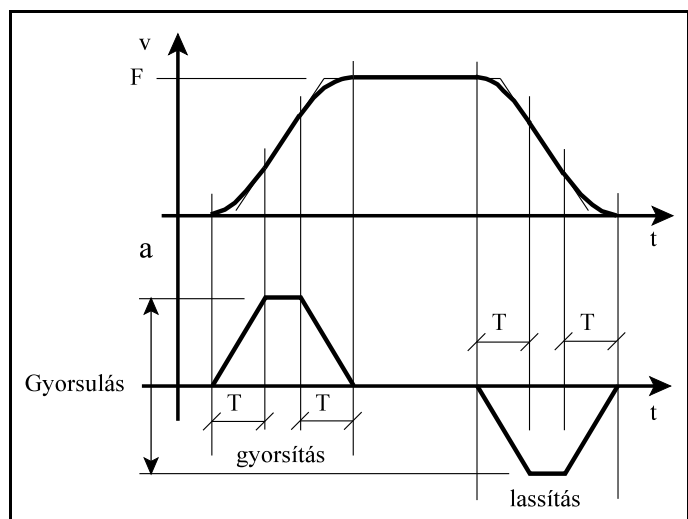
A gyorsulás nagysága tengelyenként, paraméteren állítható be.



7-1 ábra

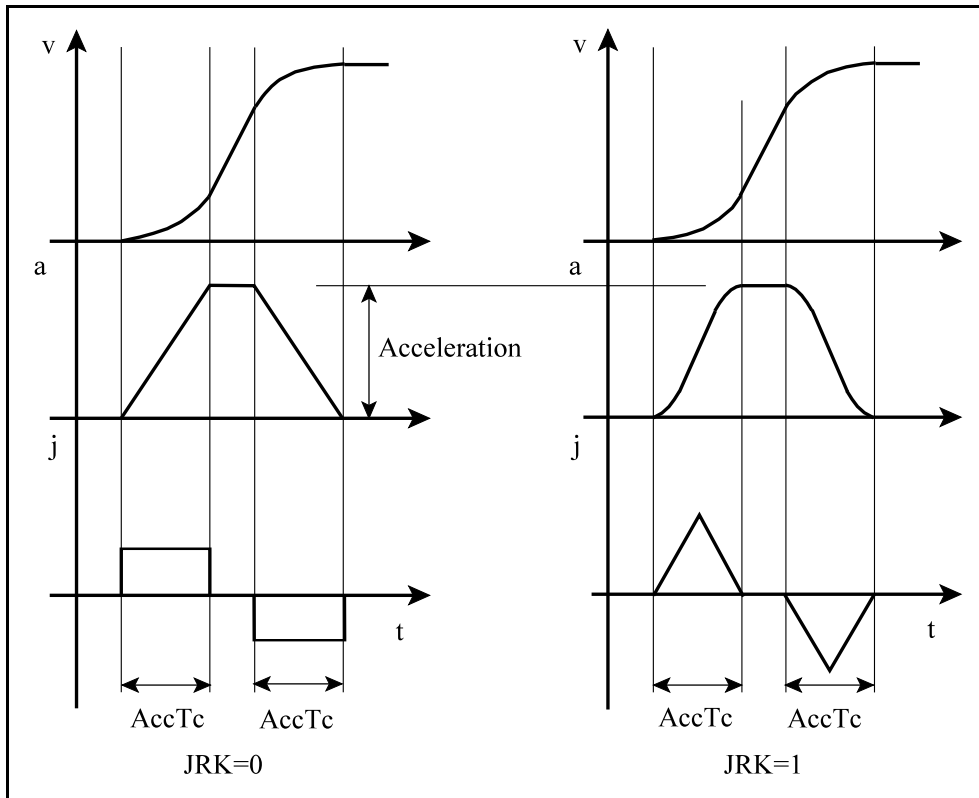
Haranggörbe alakú gyorsítás esetén a gyorsítás, illetve a lassítás alatt a gyorsulás értéke is változik, lineárisan nő, amíg el nem éri a beállított gyorsulási értéket illetve lineárisan csökken, mielőtt eléri a célsebességet. Ennek következtében az előtolás fel-, lefutásának alakja az idő függvényében haranggörbe, ezért nevezzük haranggörbe alakú gyorsításnak.

Az a T időt, amely alatt a beállított gyorsulási értéket eléri a vezérlő, tengelyenként, paraméteren állítható.



7-2 ábra

Rántásmentes gyorsítás beállításával tovább finomítható a haranggörbe alakú gyorsítás. Az N0421 Acc Contr paraméter #1 JRK=1 bitállásával lehet a rántásmentes gyorsítást bekapcsolni. Ilyenkor már a gyorsulásfüggvény fel-, lefutása is haranggörbe alakú lesz, másképp mondva a gyorsulás (a) első deriváltjában (j) sem lesz ugrás.



7-3 ábra

A JRK=1 rántásmentes gyorsítás beállításával nagyobb gyorsulás állítható be a gépen, ugyanakkor az indulás és megállás finomabb lesz.

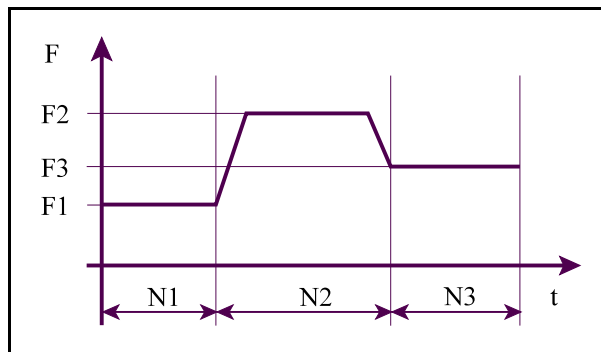
Nagysebességű megmunkálások esetén a megkívánt pontosság eléréséhez sebességelőreccsatolást kell alkalmazni. Ebben az esetben mindig haranggörbe alakú gyorsítást kell beállítani!

Normál körülmények között a vezérlő a következő esetekben gyorsít, illetve lassít:

- kézi mozdítások esetén,
- gyorsmeneti pozicionálás (G0) során a mondat elején a mozgás mindig 0 sebességről indul, és a pozicionálás végén mindig 0 sebességre lassít,
- előtoló mozgások (G1, G2, G3) esetén G9, vagy G61 állapotban a mondat végén mindig 0 sebességre lassít,
- lassít, ha az előtolást STOP gombbal megállítjuk, illetve gyorsít, ha az előtolást START-tal elindítjuk a mozgást,
- lassítással áll meg, ha a mozgás után funkció végrehajtása következik és a mondat végén, ha a MONDATONKÉNTI üzemben.

Új, az előzőnél nagyobb előtolásértékre való gyorsítást a vezérlő mindig annak a mondatnak a végrehajtása során kezdi el, amelyikben az új előtolást megadták. Ez a folyamat szükség esetén több mondaton is átnyúlhat.

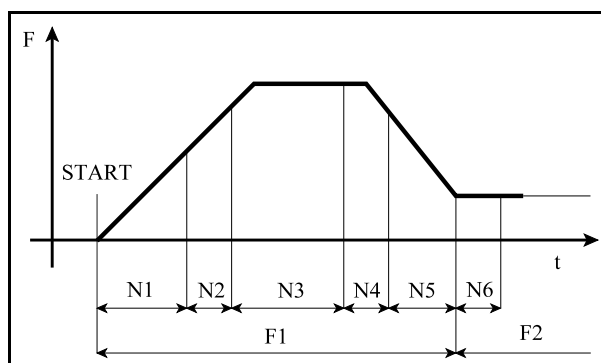
Új, az előzőnél kisebb, előtolásértékre való lassítást a vezérlő egy olyan megfelelő megelőző mondatban kezdi el, hogy abban a mondatban, ahol az új előtolást megadták, már az abban a mondatban programozott sebességgel kezdje a megmunkálást.



7-4 ábra

A vezérlés a tangenciális sebességváltozásokat előre figyeli, és nyilvántartja. Erre azért van szükség, hogy a kívánt célsebességet akár több mondat végrehajtásán átnyúló folyamatos gyorsítással érje el.

Ugyanúgy induláskor vagy megálláskor a kívánt célsebesség elérése több mondaton is átnyúlhat.

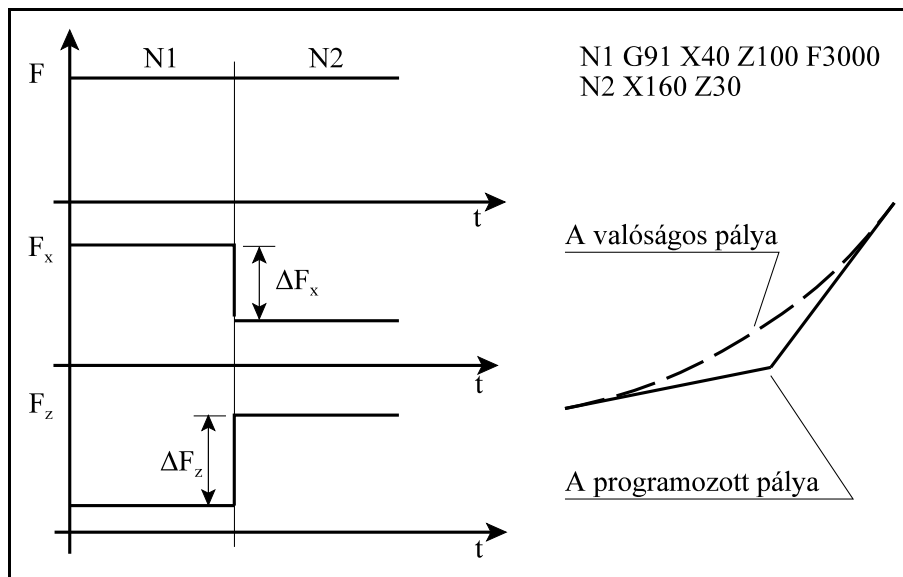


7-5 ábra

7.1 Automatikus lassítás sarkoknál G64 állapotban

Folyamatos forgácsolás esetén, G64 állapotban, a vezérlő igyekszik a programozott előtolással követni a pályát.

Ha két mondat között sarkot talál, le kell lassítania a pályamenti előtolást.



7.1-1 ábra

Ha két, egymást követő N1, N2 mondatban a saroknál nem lassít, akkor az egyes tengelyek mentén az ábrán látható előtoláskülönbségek (ΔF_x , ΔF_z) lépnek fel

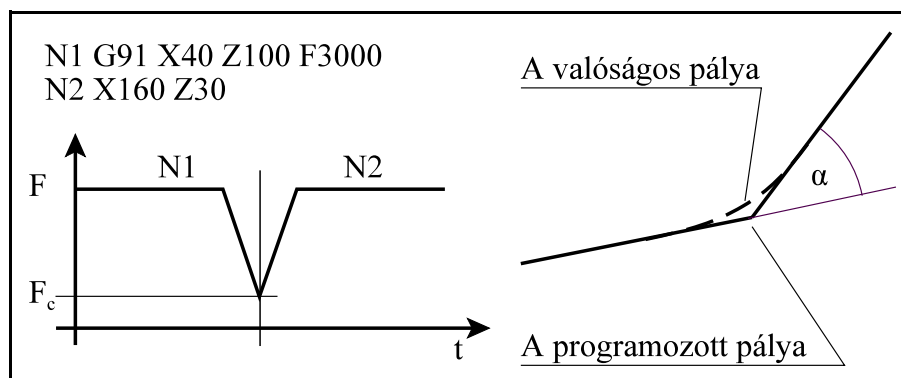
Két okból van szükség az előtolásváltozások (sarkok) detektálására, és egyúttal az előtolás lassítására:

- A pálya hirtelen irányváltozásából adódó, tengelyenkénti előtolásváltozások (gyorsulások) olyan nagyok lehetnek, hogy lassítás nélkül a hajtások nem tudják lengés nélkül követni azt, és ez a pontosság rovására megy, illetve mechanikusan túlzottan igénybe veszi a szerzőgépet.
- Ha a sarokot “élessé” kell tenni a forgácsolás során, de nem akarunk megállni és ezzel növelni a forgácsolási időt (G61 pontos megállást programozni) szintén le kell lassítani. Minél jobban lelassítjuk az előtolást, a sarok annál élesebb lesz.

Sarkok detektálását a vezérlő kétféleképp végezheti: a pálya irányszögének változását, illetve a tengelyenkénti előtoláskomponensek változását figyelve. Paraméter alapján ki lehet választani, hogy melyik módszer alapján működjön a lassítás.

Lassítás sarkoknál a pálya irányszögének változását figyelve.

Az N0306 Feed Control paraméter #0 FDF=0 bitállásánál, a lassítás a pálya irányszögének változását figyelve történik.



7.1-2 ábra

Ha a mellékelt ábrán látható N1, N2 mondat találkozásánál **az α szög értéke túllépi** a paraméteren engedélyezett értéket a vezérlő lelassítja az előtolást F_c értékre.

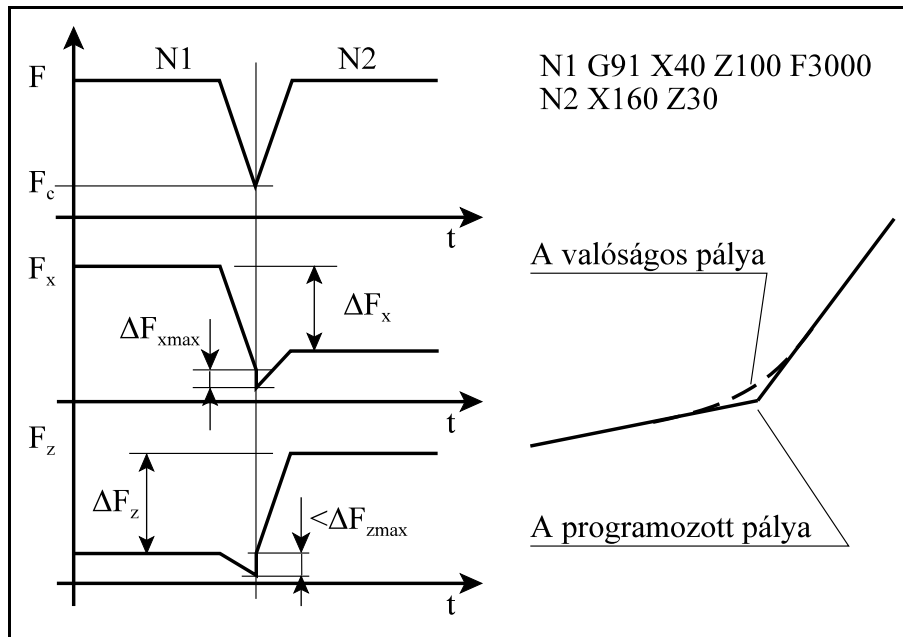
Az N0307 Crit A Diff paraméteren állítható be a kritikus szög értéke fokban: $\alpha = \text{Crit A Diff}$. Az N0308 Feed Corn paraméter értéke adja meg, hogy a kritikus szöget túllépve mekkora sarokelőtolásra lassítson le a vezérlő: $F_c = \text{Feed Corn}$.

Minél nagyobb a kritikus szög és a sarokelőtolás értéke annál gyorsabb a megmunkálás, viszont annál jobban igénybe veszi a gépet és annál jobban lekerekíti a sarkot.

Ez a beállítás nem felel meg nagysebességű megmunkálásoknál!

Lassítás sarkoknál a tengelyenkénti előtoláskomponensek változását figyelve.

Az N0306 Feed Control paraméter #0 FDF=1 bitállásánál, a lassítás az előtoláskomponensek változását figyelve történik.



7.1-3 ábra

Ha a mellékelt ábrán látható N1, N2 mondat találkozásánál **az előtoláskomponensek változása** ΔF_x , ΔF_z értéke túllépi a paraméteren engedélyezett maximumot, a vezérlő lelassítja a pályamenti előtolást F-et F_c értékre.

Az előtolást úgy lassítja le, hogy az előtolásváltozás mértéke egyik tengelyen se lépje túl az arra a tengelyre paraméteren engedélyezett (ΔF_{xmax} , ΔF_{zmax}) kritikus előtoláskülönbséget, melyet az N0309 Crit F Diff paraméteren adhatunk meg tengelyenként: $\Delta F_{xmax} = \text{Crit F Diff}_x$, $\Delta F_{zmax} = \text{Crit F Diff}_z$.

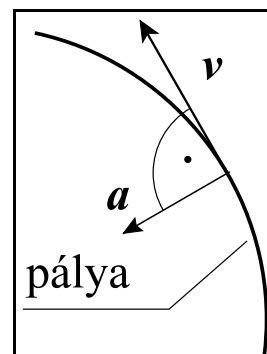
Minél nagyobb a kritikus előtoláskülönbség értéke annál gyorsabb a megmunkálás, viszont annál jobban igénybe veszi a gépet és annál jobban lekerekíti a sarkot.

A vezérlő a gyorsulásbeállítási alapján felülről korlátozza a beállított kritikus előtoláskülönbség értékét.

Ezt a beállítást kell alkalmazni nagysebességű megmunkálásoknál!

7.2 A pálya mentén normális irányban fellépő gyorsulások korlátozása

A vezérlő a megmunkálás során az előtolást a pálya érintője mentén (tangenciális irányban) állandó értéken tartja. Ennek az a következménye, hogy tangenciális irányban nem lépnek fel gyorsuláskomponensek. Nem úgy normális (a pályára, illetve a sebességre merőleges) irányban. A normális irányú gyorsulás tengelyekre eső komponensei az egyes tengelyeken túlléphetik az adott tengelyre megengedett értéket. Ezt elkerülendő a pálya menti sebességet a pálya görbületének mértékében korlátozni kell. A megengedhető normális irányú gyorsulás maximumát az N0402 Normal Acc paraméteren állíthatjuk be.



7.2-1 ábra

A normális irányú gyorsulás korlátozása köríveknél

Körívek megmunkálása során az előtolás F nagyságát az

$$F = \sqrt{a \cdot r}$$

összefüggés alapján korlátozza be, ahol:

- a: a körinterpolációban részt vevő tengelyekre beállított gyorsulásértékek közül a kisebb,
- r: a kör sugara.

A körinterpolációt már az így kiszámított sebességgel kezdi el.

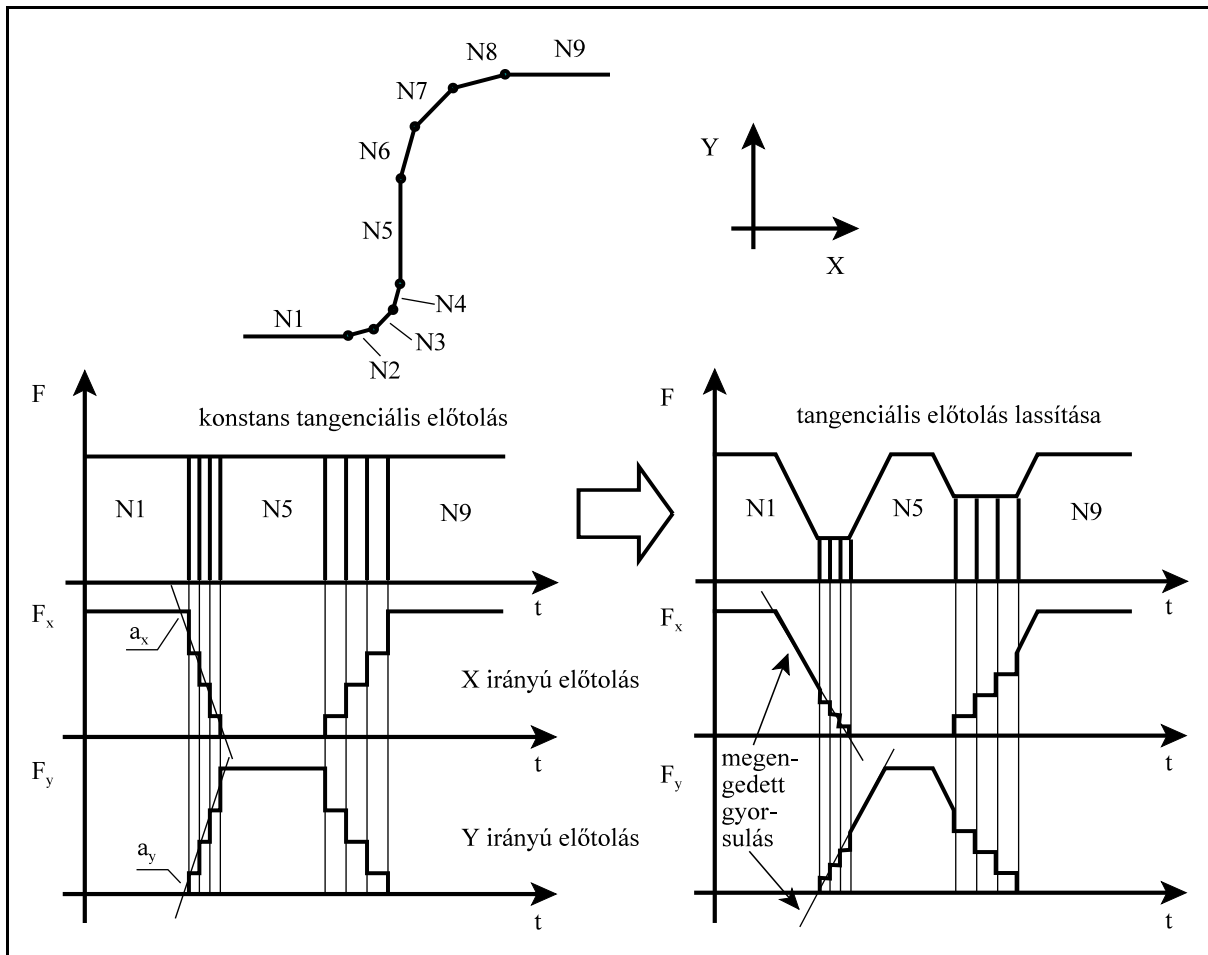
Az N0310 Circ F Min paraméteren megadott előtolásnál kisebbre nem csökkenti a sebességet, függetlenül a fenti összefüggéstől.

☞ **Figyelem:**

Ez a funkció nem tévesztendő össze a körívek belső megmunkálásánál G41, G42 állapotban történő automatikus előtoláscsökkentéssel!

A normális irányú gyorsulás korlátozása egyéb interpolációnál

Egyenes interpoláció esetén az adott szakasznak (pályának) nincs ugyan görbülete, ezért normális irányú gyorsuláskomponens sincs, ez azonban csak hosszú egyenes szakaszokra igaz. Ha egy **pálya apró, egyenes szakaszokból épül fel**, mint pl. az a szerszámgyártásban bevett, akkor az így kiadódó pálya görbülete számottevő lehet, és az előtolást lassítani kell, mint az alábbi példából látszik:



7.2-2 ábra

Az N2, N3, N4, illetve az N6, N7, és N8 mondatokban a pálya rövid, egyenes szakaszokból épül fel. Ha a pályamenti előtolást állandó értéken tartjuk (az ábrán a bal oldali diagram) a pálya geometriájából (irányváltozásából) adódóan az X, illetve az Y tengelyen a sebességváltozás meredeksége (a normális irányú gyorsulás) meghaladhatja az arra a tengelyre engedélyezett értéket.

Ezért a pályát mondatról mondatra végigvizsgálja a vezérlő, hogy a normális irányú gyorsulásokat korlátozni tudja. Ahol a geometriából adódóan az egyes tengelyek mentén a megengedettnél nagyobbak a gyorsuláskomponensek ott a pályamenti sebességet lassítani kell. A jobb oldali diagram azt mutatja, hogyan csökken a sebességváltozás mértéke (a normális irányú gyorsulás) az egyes tengelyek mentén a pályamenti előtolás lassításának hatására.

A G5.1 Q2 **símitó interpoláció esetén** a vezérlő szintén vizsgálja a pálya görbületéből adódóan fellépő normális irányú gyorsulásokat, és szükség esetén csökkenti az előtolást.

7.3 A gyorsulásugrás korlátozása

A pálya egyes szakaszain hirtelen gyorsulásugrás, rántás alakulhat ki, amely lengéseket okoz, mechanikusan igénybe veszi a gépet és a forgácsolt felületen meglátszik. Ilyen eset például, amikor egy egyenes szakasz után egy érintőkörrel folytatódik a megmunkálás, vagy amikor egy körívet egy érintő egyenes követ.

Ennek a funkciónak az a célja, hogy az átmeneti ponton az előtolás csökkentésével a vezérlő korlátozza a gyorsulásugrás mértékét.

A gyorsulásugrás korlátozása körmondatok elején és végén

Egyenes szakaszból érintő körívbe F előtolással belépésnél a gyorsulásugrás mértéke az

$$a = \frac{F^2}{r}$$

összefüggésből számítható, ahol:

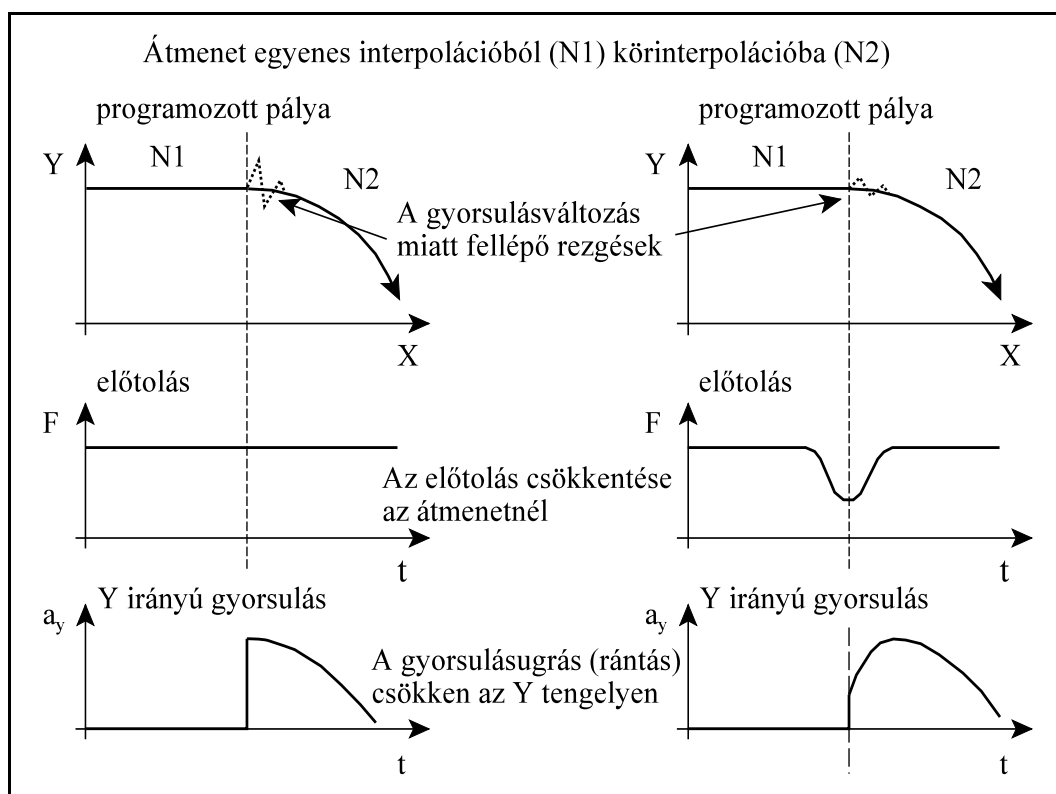
a: a gyorsulásugrás nagysága,

r: a kör sugara.

Ha például F6000-es előtolással egy 10 mm sugarú körívbe lép a gép az alábbi ábrán látható módon, az Y tengelyen kialakuló gyorsulásugrás számszerű értéke:

$$a = \frac{\left(\frac{6000 \text{ mm}}{60 \text{ sec}}\right)^2}{10 \text{ mm}} = 1000 \frac{\text{mm}}{\text{sec}^2}$$

Ha azt akarjuk, hogy a gyorsulásugrás ne legyen nagyobb, mint 250 mm/sec^2 , a fenti egyenlet alapján az előtolást $F=50 \cdot 60=3000 \text{ mm/min}$ -re kell csökkenteni.



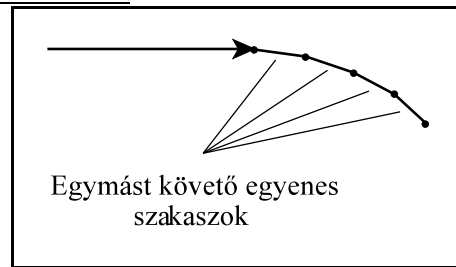
7.3-1 ábra

A gyorsulásugrás körívek esetén az N0404 Acc Diff Circ paraméteren korlátozható.

A gyorsulásugrás korlátozása egymást követő egyenes mondatokban

Ha a pálya hosszú egyenes szakaszból áll, a gyorsulás-változás elhanyagolható mértékű. Ebben az esetben a tengelyekre eső előtoláskomponensek változása korlátozhatja az előtolást.

Más a helyzet, ha a pálya nagyon rövid egyenes szakaszból áll. Ebben az esetben előállhat az az eset, hogy két egyenes szakasz között az egyes tengelyekre eső előtolás-változás kicsi, emiatt nem korlátozza az előtolást az interpolátor, ám az egyes tengelyekre eső gyorsulásugrás magas. Ilyen esetekben is korlátozni kell az előtolást a megengedhető gyorsulásugrás függvényében. Egemást követő egyenes szakaszok esetén a gyorsulásugrás az N0403 Acc Diff paraméteren korlátozható.



7.3-2 ábra

8 A várakozás (G4)

A

G94 G4 P....

parancsal várakozási időt programozhatunk másodpercben.

A

G95 G4 P....

parancsal várakozási időt programozhatunk főorsó fordulaton.

P pontossága 15 decimális számjegy.

Az N1337 Execution Config paraméter #1 SEC=1 bitállásánál a várakozás mindig másodpercben történik, még G95 állapotban is.

A várakozás mindig a következő mondat végrehajtásának programozott késleltetését jelenti. Nem öröklődő funkció.

9 A referenciapont

A referenciapont a tengely azon pontja, ahol a mérőrendszer 0 pozíciót ad fel.

Inkrementális mérőeszközökkel szerelt tengelyeken ezt a pontot meg kell keresni. Ezt a folyamatot nevezzük referenciapont felvételnek. A referenciapont megléte után lehet bemérni a munkadarab koordinátarendszereket, és abszolút pozícióra állni. Csak referenciapont felvétel után hatásosak a paraméteres végállások és a programozott munkatérbehatárolás.

A vezérlő a pozíciókat nem a referenciapont-hoz képest tatja nyilván, hanem a gépi koordinátarendszerben.

A **gépi koordinátarendszer** nullpontját a gép építője határozza meg, amely egy kitüntetett pont a szerszámgépen.

A vezérlő a gépi koordinátarendszerben tartja nyilván a cserehelyzeteket, a forgó tengelyek [9-1](#) ábra

forgáspontjait, stb. Ugyancsak a gépi koordinátarendszerben tartja nyilván a gép mérőrendszerének összes kompenzációját (menetemelkedési, egyenességi, stb.).

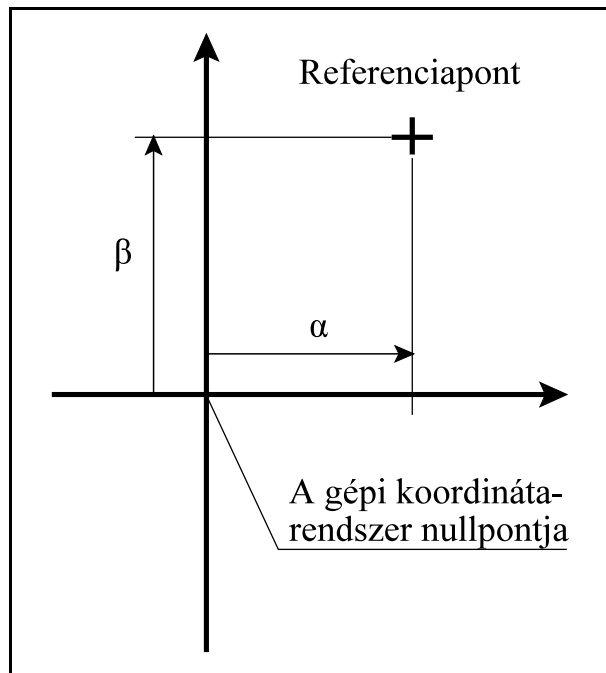
A vezérlő a **referenciapont helyzetét is a gépi koordinátarendszerben** tartja nyilván, amelyet a gép építője paraméteren állít be.

Ha a tengely **inkrementális mérőrendszerrel** van szerelve, bekapcsolás után fel kell venni a referenciapontot. A referenciapont felvétel során a szánok paraméteren meghatározott irányban ráfutnak egy kapcsolóra, majd onnan lejöve megkeresik a mérőrendszer nullimpulzusát, és bejegyzik a referenciapont meglétét. A pozíció a gépi koordinátarendszerben a paraméteren megadott érték lesz.

Ha a tengely **távolságkódolt mérőrendszerrel** van szerelve, bekapcsolás után fel kell venni a referenciapontot. A referenciapontfelvétel során a szánok paraméteren meghatározott irányban elindulnak, megkeresnek két nullimpulzust, és bejegyzik a referenciapont meglétét. A pozíció a gépi koordinátarendszerben a másodikként megtalált nullimpulzus helyzete lesz.

Ha a tengely **abszolút mérőrendszerrel** van szerelve, nincs szükség bekapcsolás után referenciapontfelvételre.

Távolságkódolt és abszolút mérőrendszer esetén is a referenciapont az a pont, ahol a mérőrendszer 0 pozíciót ad fel. Ez a pont általában **nem esik a gép munkatartományába**. Ezért ezt a pontot a gép építője paramétermegadással a munkatér belsejébe, pl a pozitív végállás közelébe tolja el, majd az eltolt referenciapontot méri be a gépi koordinátarendszer origójához. Erre az eltolásra pl. akkor van szükség, ha a G28 utasítást akarják használni az alkatrészprogramban, pl. szerszám-



9-1 ábra

mokkal való kiálláshoz.

9.1 Automatikus referenciapont felvétel (G28)

A

G28 v

utasítás a v vektorban meghatározott tengelyeken a referenciapontra áll rá. A mozgás két fázisból tevődik össze.

Első fázis

Először a v vektor által meghatározott tengelyeken, az **aktuális munkadarab koordináta-rendszerben** megadott koordinátákat **közbülső pont**nak véve, gyorsmenettel a v vektor által meghatározott közbülső pontra áll. A megadott koordinátaértékek lehetnek abszolút, illetve inkrementális értékek is. A közbülső pontra úgy áll rá, hogy a síkbeli szerszámsugár korrekció törlődik.

Második fázis

Ezután a közbülső pontról a v vektor által meghatározott tengelyeken egyidejűleg referenciapontot vesz fel.

Ha még nem történt az adott tengelyen referenciapontfelvétel, a kézi referenciapontfelvétel által meghatározott menet szerint, felveszi a referenciapontot. Ilyen esetben,

- ha a tengely **inkrementális mérőrendszerrel** van szerelve a mozgás végén a gépi pozíció a paraméteren meghatározott referenciapont pozíció lesz,
- ha a tengely **távolságkódolt mérőrendszerrel** van szerelve a mozgás végén a gépi pozíció a második nullimpulzus helyzete lesz.

Ha már történt az adott tengelyen referenciapontfelvétel, vagy a tengely abszolút mérőrendszerrel van szerelve, a tengely gyorsmenettel, a gépi koordináta-rendszerben megadott referenciapont pozícióra áll.

G28 nem öröklődő kód.

Például:

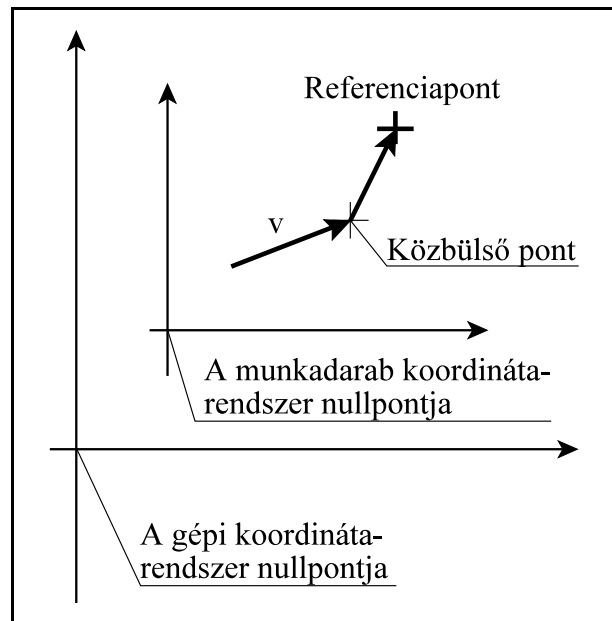
G90 G28 X100 Z50 (a közbülső pont: X=100, Z=50)

Ha X pozíció X=20, Z pozíció Z=50:

G91 G28 X100 Z50 (a közbülső pont: X=120, Z=100)

☞ **Megjegyzések:**

- Ha még nincs érvényes referenciapont, a G28 parancsban szereplő v közbülső koordinátáknak inkrementális értéket kell adni.



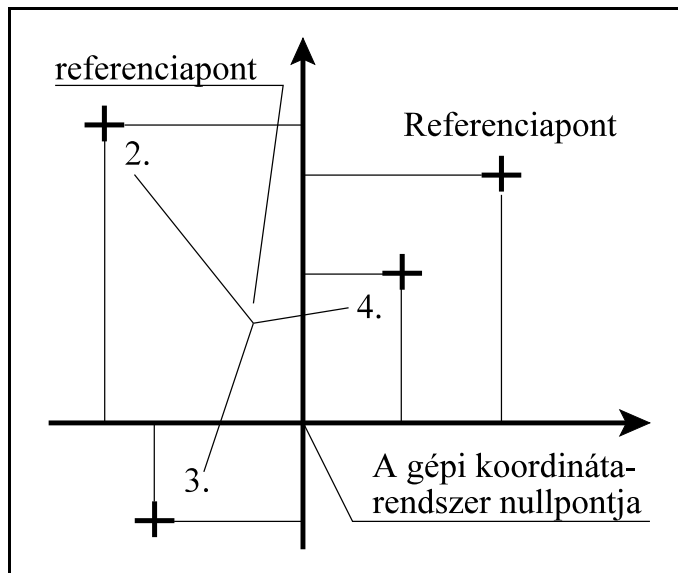
9.1-1 ábra

9.2 A 2., 3., 4. referenciapontra állás (G30)

A *gépi koordinátarendszerben paraméteren* ki lehet jelölni 3 további nevezetes pontot, amit **2., 3., 4. referenciapontnak** nevezünk.

Ezeket a referenciapontokat használjuk a gépen cserehelyzetek, pl. szerszámcserehely, palettacserehely stb. tárolására.

Ezekre a cserehelyekre csak a referenciapont felvétele után lehet mozogni.



A

9.2-1 ábra

G30 v P

utasítás a P címen meghatározott referenciapontra küldi a v vektor címein meghatározott koordinátájú tengelyeket.

P2: 2. referenciapont

P3: 3. referenciapont

P4: 4. referenciapont

A mozgás a G28 utasításhoz hasonlóan két részből tevődik össze.

Először a v vektor által meghatározott koordinátákat közbülső pontnak véve, gyorsmenettel a v vektor által meghatározott **közbülső koordinátákra áll** lineáris mozgással. A megadott koordinátaértékek lehetnek abszolút, illetve inkrementális értékek is. A mozgás mindig az aktuális koordinátarendszerben történik. A lineáris mozgás végpontjára úgy áll rá, hogy a síkbeli szerszámsugar korrekciósvektor törlődik.

A második fázisban a közbülső pontról a v vektor által kijelölt tengelyek gyorsmeneti mozgással a **P címen kiválasztott referenciapontra állnak**.

A referenciapontra állás a korrekciós vektorok (hossz, eltolás, 3 dimenziós sugár) figyelmen kívül hagyásával történik, azokat a G30 utasítás kiadása előtt törölni nem kell, viszont a további mozgások programozásánál a vezérlés érvényesíti azokat. A síkbeli szerszámsugar korrekció automatikusan visszakapcsolódik az első mozgásmondatban.

Nem öröklődő kód.

A G30 v P1 utasítás hatására a gép a referenciapontra mozog, hatása megegyezik a G28 utasításával.

Például:

G90 G30 X100 Z50 P3 (a közbülső pont: X=100, Z50, P3-ra áll)

Ha X pozíció X=20, Y pozíció Z=50:

G91 G30 X100 Z50 P4 (a közbülső pont: X=120, Z=100, P4-re áll)

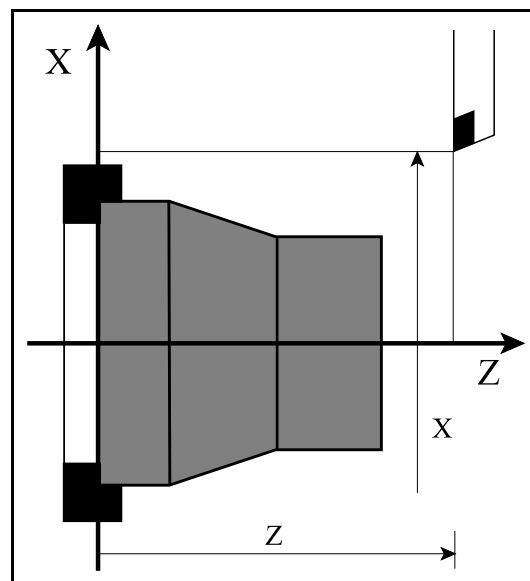
10 Koordinátarendszerek, síkválasztás

A programban egy pozíciót, ahova a szerszámot akarjuk mozgatni, koordinátaadatokkal adunk meg. Ha 2 tengelyünk van (X, Z) a szerszám pozícióját két koordinátaadat X____ Z____ fejezi ki.

Ahány tengely van a gépen a szerszám pozícióját annyi különböző koordinátaadat fejezi ki. A koordinátaadatok mindig egy adott koordinátarendszerben értendők.

A vezérlő háromféle koordinátarendszert különböztet meg:

1. a gépi koordinátarendszert,
2. a munkadarab koordinátarendszert,
3. a lokális koordinátarendszert.



10-1 ábra

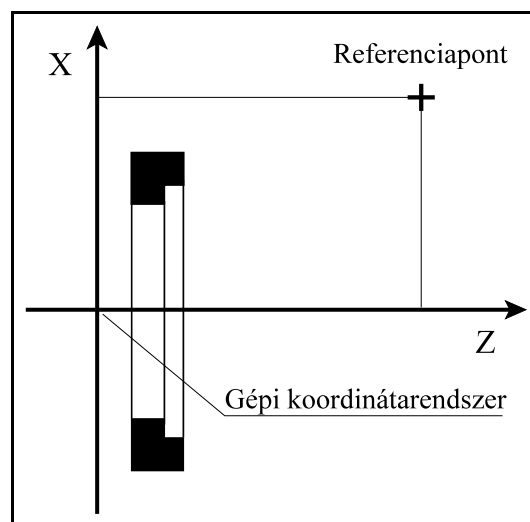
10.1 A gépi koordinátarendszer

A vezérlő a pozíciókat nem a referenciaponthoz képest tartja nyilván, hanem a gépi koordinátarendszerben.

A gépi koordinátarendszer nullpontját a gép építője határozza meg, amely egy kitüntetett pont a szerszámgépen.

A vezérlő a gépi koordinátarendszerben tartja nyilván a cserehelyzeteket, a forgó tengelyek forgáspontjait, stb. Ugyancsak a gépi koordinátarendszerben tartja nyilván a gép mérőrendszerének összes kompenzációját (menetemelkedési, egyenességi, stb.).

A vezérlő a **referenciapont helyzetét is a gépi koordinátarendszerben** tartja nyilván, amelyet a gép építője paraméteren állít be.



10.1-1 ábra

A gépi koordinátarendszer helyzetét semmilyen utasítással, eltolással nem lehet megváltoztatni.

10.1.1 Pozicionálás a gépi koordinátarendszerben (G53)

A

G53 v

utasítás hatására a szerszám a gépi koordinátarendszer v pozíciójú pontjára mozog.

- G90, G91 állapottól függetlenül a v koordinátákat **mindig abszolút** koordinátaként kezeli,
- I operátor a koordináták címe után, vagy U, V, W cím inkrementális megadásra való használata esetén a 2097 Illegális inkrementális mozgás ... tengelyen üzenetet adja,
- a mozgás mindig **gyorsmenettel** történik G00 utasításhoz hasonlóan,
- a pozicionálás mindig a beállított szerszámkorrekciók (hossz, sugár) figyelmen kívül hagyásával történik.

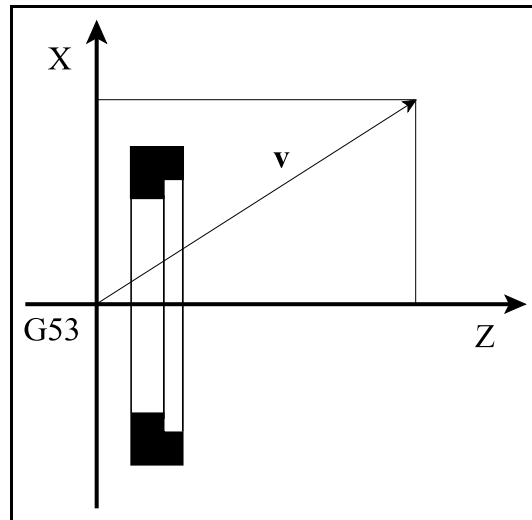
G53 utasítást csak referenciapontfelvétel után lehet végrehajtani. A G53 parancs **egylövetű**, csak abban a mondatban hatásos, ahol megadásra került.

Példa: A

```
G53 X200 Z20
```

utasításra a megadott pontra mozog a gépi koordinátarendszerben.

☞ **Figyelem!** A G53 utasítás felfüggeszti a mondatok előreolvasását (pufferelését). Ezért önmagában álló G53 utasítás, koordinátamegadás nélkül, a mondatok előreolvasásának felfüggesztésére is használható!



10.1.1-2 ábra

10.2 A munkadarab koordinátarendszerek

Azt a koordinátarendszert, amelyben az alkatrészprogramot írjuk, munkadarab koordinátarendszernek nevezzük. A munkadarab koordinátarendszer origóját a gépi koordinátarendszerhez képest tartja nyilván a vezérlő.

A munkadarab koordinátarendszer nullpontját a darab egy megfelelő pontjához állítjuk be. Ez a pont esztergáláskor X irányban a darab forgástengelye, Z irányban pl. a tokmány felfekvő felülete, vagy a darab eleje, maráskor például a darab egyik sarka, egy furat, vagy váll középpontja stb. A beállítás történhet:

- a gépen belül kézi, vagy tapintóval történő beméréssel, vagy
- a gépen kívül.

Ez utóbbi esetben a kívül bemért értékeket be kell vinni a vezérlő memóriájába. Ez történhet kézi adatbevitellel, vagy programból NC utasítások használatával.

A munkadarab koordinátarendszerek és a csatornák kapcsolata

A munkadarab koordinátarendszerek eltolásai az egyes tengelyekre vonatkoznak. Mivel minden tengely egy adott csatornához van paraméteren kijelölve, ezért minden csatornához külön munkadarab eltolási táblázat tartozik.

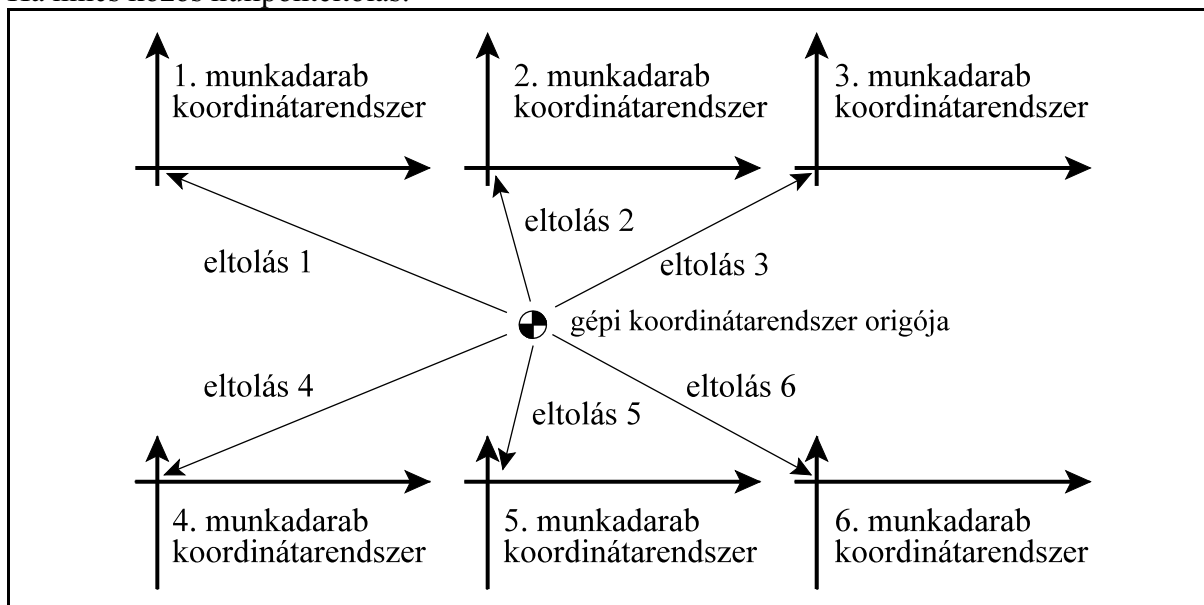
Ha két csatorna között felcserélünk egy vagy több tengelyt, a tengelyek viszik magukkal az új csatornába a nullponteltolásaikat. Az ilyen esetekben a tengelycsere után célszerű egy új munka-

darab koordinátarendszert leírni abszolút pozícionálással együtt és a megmunkálást ezután folytatni.

10.2.1 A munkadarab koordinátarendszer kiválasztása (G54...G59)

Alapkitételben 6 különböző munkadarab koordinátarendszert tart nyilván a vezérlő. A munkadarab koordinátarendszerek *eltolásait a gépi koordinátarendszer origójához képest* tengelyenként kell megadni.

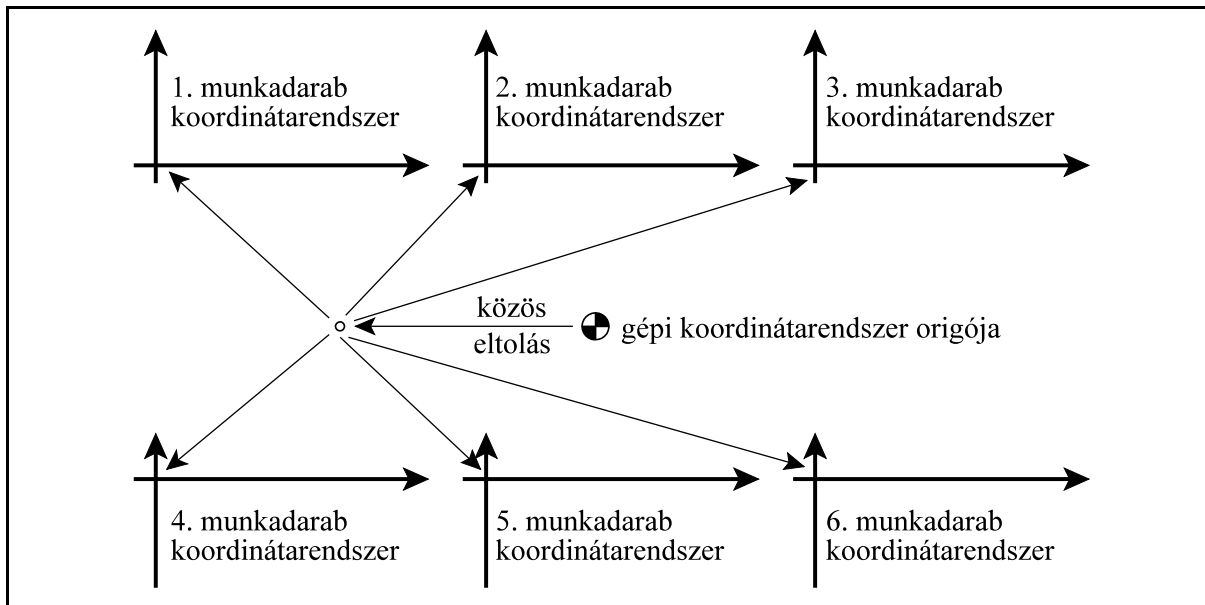
Ha nincs közös nullponteltolás:



10.2.1-1 ábra

A gépi koordinátarendszer origójához képest az összes munkadarab koordinátarendszert el lehet tolni. A *közös nullponteltolás az összes munkadarab koordinátarendszer origóját eltolja* a gépi koordinátarendszerhez képest.

Ha van közös nullponteltolás:



10.2.1-2 ábra

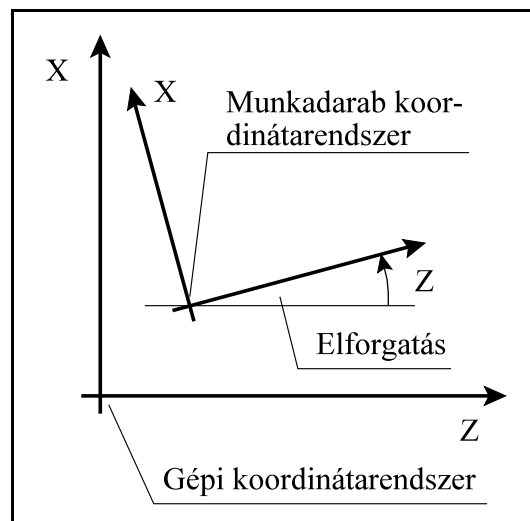
A munkadarab koordinátarendszereket nem csak eltolni lehet a gépi nullponthoz képest, hanem **el is lehet forgatni** azokat az origójuk körül. Az elforgatás szögét fokban kell megadni.

Az elforgatásra akkor lehet szükség, ha a darab nem lehet pontosan kiállítani pl. az Z tengellyel párhuzamosan, annak nagy tömege miatt.

Az alkatrészprogramot úgy lehet megírni, mintha a darab párhuzamos lenne a fő tengelyekkel.

A munkadarab koordinátarendszereket csak egy fő síkban lehet elforgatni, tehát vagy az XY, vagy a ZX, vagy az YZ síkban.

A közös nullponteltolásra elforgatást nem adhatunk meg.



10.2.1-3 ábra

A G54...G59 utasításokkal lehet a különböző munkadarab koordinátarendszereket kiválasztani.

- G54: 1. munkadarab koordinátarendszer
- G55: 2. munkadarab koordinátarendszer
- G56: 3. munkadarab koordinátarendszer
- G57: 4. munkadarab koordinátarendszer
- G58: 5. munkadarab koordinátarendszer
- G59: 6. munkadarab koordinátarendszer

Öröklődő funkciók.

Bekapcsolás, referenciapontfelvétel, reset, vagy program vége után a G54-es koordinátarendszer kerül kiválasztásra.

Az interpolációs mondatok abszolút koordinátaadatait a vezérlő az aktuális munkadarab koordinátarendszerben veszi figyelembe.

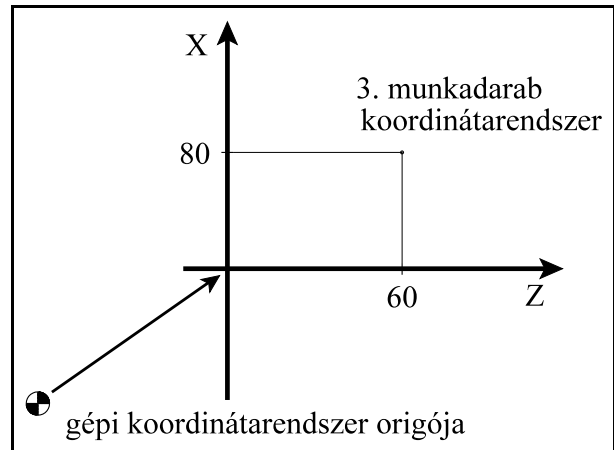
Például a

G56 G90 G00 X80 Z60

utasítás esetén a 3. munkadarab koordinátarendszer

X=80, Z=60

pontjára áll rá.



10.2.1-4 ábra

A munkadarab koordinátarendszer váltással a szerszám pozíciója az új koordinátarendszerben kerül kijelzésre. Például az ábra szerint a **G54** munkadarab koordinátarendszer *eltolása* a gép koordinátarendszerében

X=260 Z=80

A **G55** munkadarab koordinátarendszer *eltolása* a gép koordinátarendszerében

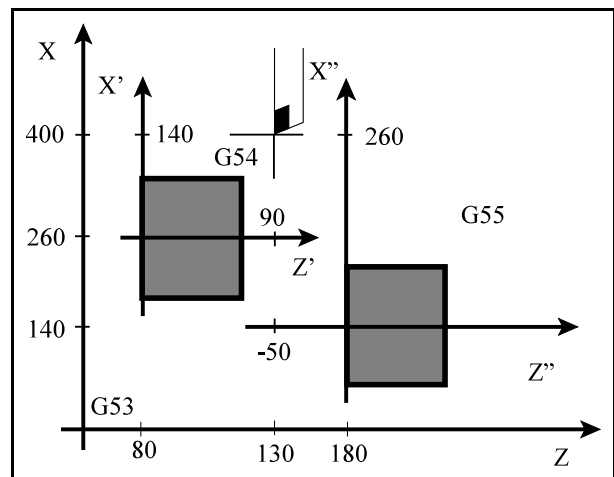
X=140, Z=180.

A **G54** X', Z' koordinátarendszerben a *szerszám pozíciója*

X'=140, Z'=90.

A **G55** utasítás hatására a *szerszám pozíciója* az X'', Z'' koordinátarendszerben kerül értelmezésre:

X''=260, Z''=-50.



10.2.1-5 ábra

10.2.2 A munkadarab koordinátarendszerek eltolásának beállítása (G10 L2)

A munkadarab koordinátarendszerek eltolását, elforgatását, és a közös nullponteltolást be lehet állítani programutasítással is.

A beállítás a

G10 L2 P v I J K

utasítással történik, ahol

P = 0

a közös nullponteltolás eltolás állítása

P = 1...6

a G54, ..., G59 munkadarab koordinátarendszer választása

v (X, Y, Z, ...):

a tengelyenkénti eltolási értékek

A tengelyeltolások mindig derékszögű értéként kerülnek beolvasásra. A hosszadatok mm-ben, vagy inch-ben, a szögadatok fokban.

I: elforgatás szöge a G17 síkban

J: elforgatás szöge a G18 síkban

K: elforgatás szöge a G19 síkban

Az elforgatás szögét fokban kell megadni.

A közös nullponteltolásra nem adható meg elforgatás. I, J, K cím közül csak egy adható meg.

A G10 utasítás egylövetű.

G90 abszolút adatmegadás parancsállapotban a koordinátacímekre és az I, J, K címre írt érték kerül a megfelelő eltolási regiszterbe.

G91 inkrementális adatmegadás parancsállapotban, vagy I operátor használata esetén a címekre írt adat hozzáadódik a megfelelő eltolási regiszter tartalmához. I operátort csak a koordinátacímekre lehet használni, I, J, K címre nem.

10.2.3 A bővített munkadarab koordinátarendszerek kiválasztása (G54.1 P)

Opcionálisan további 99 db. munkadarab koordinátarendszert lehet a vezérlőben használni. Ezek a koordinátarendszerek a 6 db. alapkoordinátarendszer bővítései, ezért nevezzük őket bővített munkadarab koordinátarendszereknek.

A **közös nullponteltolás** a bővített koordinátarendszereket is eltolja. A bővített munkadarab koordinátarendszereket éppúgy **el lehet forgatni**, mint az alapkoordinátarendszereket.

A

G54.1 Pp

utasítással lehet egy bővített munkadarab koordinátarendszert kiválasztani, ahol P címen adható meg a bővített koordinátarendszer sorszáma:

$$P = 1, 2, \dots, 99$$

Öröklődő funkció.

☞ **Figyelem!** A P cím több célra használható. Ügyelni kell arra, hogy a mondatban a P cím használata egyértelmű legyen:

G0 G54.1 P16 X100 Z20 M98 P1 (HIBÁS! Két P egy mondatban)

G0 G54.1 P16 X100 Z20 (Egyértelmű)

10.2.4 A bővített munkadarab koordinátarendszerek eltolásának beállítása (G10 L20)

A bővített munkadarab koordinátarendszerek eltolását és elforgatását be lehet állítani programutasítással is.

A beállítás a

G10 L20 P v I J K

utasítással történik, ahol

$P = 1 \dots n$ a G54.1 P1, G54.1 P2, ..., G54.1 Pn munkadarab koordinátarendszer választása

v (X, Y, Z, ...): a tengelyenkénti eltolási értékek

A tengelyeltolások mindig derékszögű értéként kerülnek beolvasásra. A hosszadatok mm-ben, vagy inch-ben, a szögadatok fokban.

I: elforgatás szöge a G17 síkban

J: elforgatás szöge a G18 síkban

K: elforgatás szöge a G19 síkban

Az elforgatás szögét fokban kell megadni.

I, J, K cím közül csak egy adható meg.

A G10 utasítás egylövetű.

G90 abszolút adatmegadás parancsállapotban a koordinátacímekre és az I, J, K címre írt érték kerül a megfelelő eltolási regiszterbe.

G91 inkrementális adatmegadás parancsállapotban, vagy I operátor használata esetén a címekre

írt adat hozzáadódik a megfelelő eltolási regiszter tartalmához. I operátort csak a koordinátacímekre lehet használni, I, J, K címre nem.

10.2.5 Új munkadarab koordinátarendszer létrehozása (G92)

A

G92 v

utasítás hatására új munkadarab koordinátarendszert hozhatunk létre úgy, hogy egy kijelölt pont, például a szerszám hegye lesz az új munkadarab koordinátarendszer v koordinátájú pontja. Ezután bármely következő abszolút parancs ebben az új munkadarab koordinátarendszerben értendő, és a pozíciókijelzés is ebben a koordinátarendszerben képződik. A G92 parancsban megadott koordináták mindig derékszögű, abszolút értékűként kerülnek értelmezésre.

Ha például a szerszám az

$$X=200, Z=150$$

koordinátájú ponton van az aktuális X, Z munkadarab koordinátarendszerben, a

$$G92 X120 Z90$$

utasítás hatására egy új X', Y' koordinátarendszer jön létre, amelyben a szerszám az

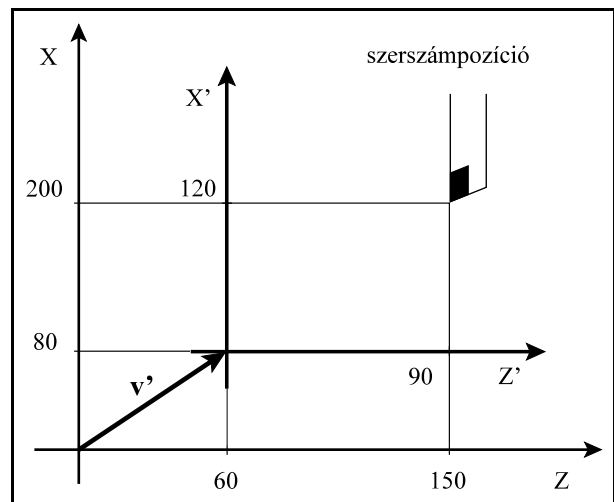
$$X'=120, Z'=90$$

koordinátájú pontra kerül. Az X, Z-X', Z' koordinátarendszerek közötti v' eltolásvektor tengelyirányú komponensei:

$$v'_x = 200 - 80 = 120, \text{ illetve}$$

$$v'_z = 150 - 90 = 60$$

lesznek.



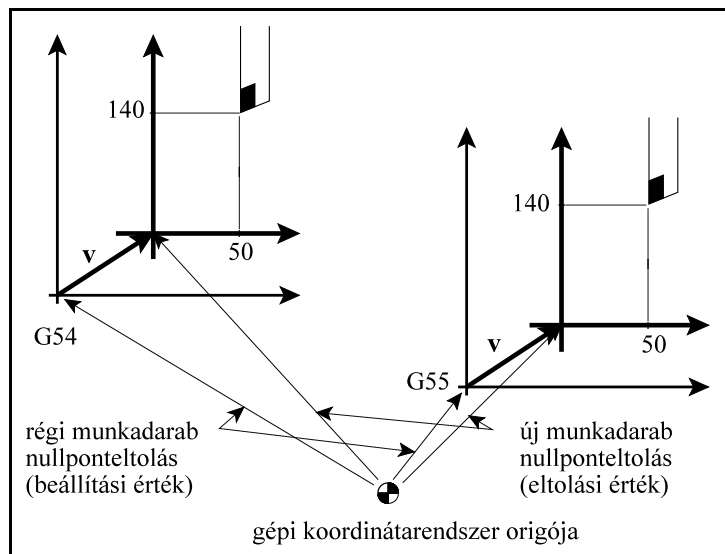
10.2.5-1 ábra

A G92 parancs minden munkadarab koordinátarendszerben érvényesül, azaz az egyikben kiszámított v eltolást a többiben is figyelembe veszi.

A G92 utasítással beállított munkadarab koordinátarendszer eltolása bekapcsolásra, programvégén, és reszetre törlődik.

A **G92** utasítás **törli a szerszámsugár korrekciós vektort**, azt nem számítja bele az eltolás képzésébe.

Az utasítás **nem törli a hosszkorrekciós vektort**, mindig a szerszám hegyének pozíciójához számolja az eltolást.



10.2.5-2 ábra

A **G92** utasítás azokon a tengelyeken, amelyek az utasításban szerepelnek, **törli** a lokális koordinátarendszer **G52** utasítással programozott **eltolásait**.

A mozgáskor a G92 utasításban megadott eltolásvektort is elforgatva veszi figyelembe az érvényben lévő munkadarab koordinátarendszer elforgatása.

10.3 A lokális koordinátarendszer (G52)

Az alkatrészprogram írása közben bizonyos esetekben könnyebb a koordinátaadatokat egy úgynevezett lokális koordinátarendszerben megadni, mint a munkadarab koordinátarendszerben.

A

G52 v

utasítás egy lokális koordinátarendszert hoz létre.

Ha a v koordináta **abszolút érték**ként van megadva, a lokális koordinátarendszer origója a munkadarab koordinátarendszer v koordinátájú pontjára kerül.

Ha a v koordináta **inkrementális érték**ként van megadva a lokális koordinátarendszer origóját v -vel eltolja.

Ettől kezdve minden abszolút koordinátákkal megadott mozgásparancs az új koordinátarendszerben kerül végrehajtásra. A pozíciókijelzés is az új koordinátarendszerben történik.

v koordináták **értékeit** mindig **derékszögű** adatként kezeli.

A

G90 G52 v0

utasítás törli a v koordinátájú pontokon az eltolásokat. Bekapcsolásra, program végén, reset hatására a G52-vel beállított eltolásértékek törlődnek.

Ha a szerszám az

$$X=200, Z=150$$

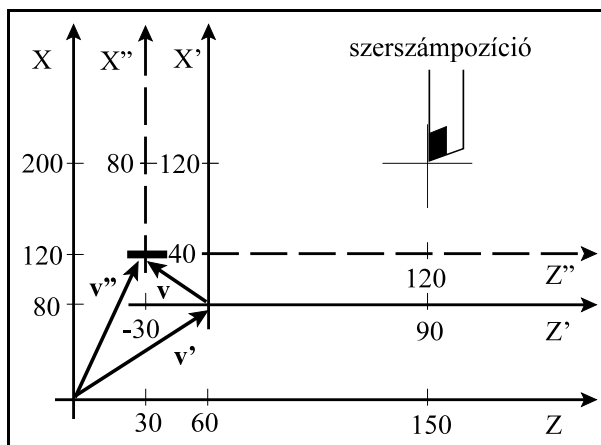
koordinátájú ponton tartózkodik az aktuális X, Z munkadarab koordinátarendszerben, a

$$G90 G52 X80 Z60$$

utasítás hatására egy új, X', Z' lokális koordinátarendszer képződik, amelyben a szerszám az

$$X'=120, Z'=90$$

koordinátájú pontra kerül. Az X, Z- X', Z' koordinátarendszerek közötti v' eltolásvektor tengelyirányú komponenseit határozzuk meg a G52 utasítással: $v'_x=80$, illetve $v'_z=60$.



10.3-1 ábra

Ha új, X'', Z'' koordinátarendszert akarunk létrehozni az eljárás kétféle lehet:

Abszolút adatmegadással:

$$G90 G52 X120 Z30$$

utasítás az **X, Z munkadarab koordinátarendszerben** az X'', Z'' koordinátarendszer origóját az $X=120, Z=30$ koordinátájú pontra **állítja**. A v'' vektor komponensei $v''_x=120, v''_z=30$ értékadással képződnek.

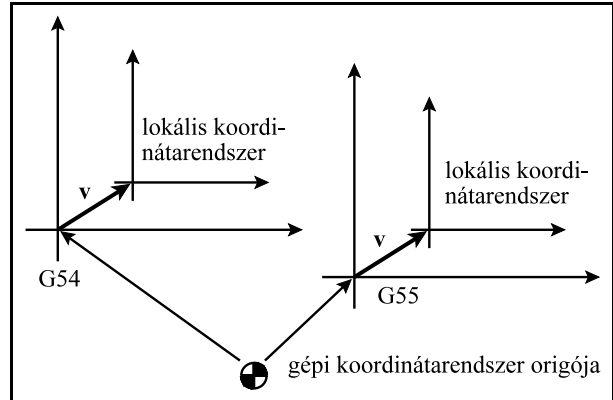
Inkrementális adatmegadással:

$$G91 G52 X40 Z-30$$

utasítás az **X', Z' lokális koordinátarendszer origóját eltolja** $X'=40, Z'=-30$ értékkel. A v vektor komponensei $v_x=40, v_z=-30$ értékadással képződnek. A v'' vektor, amely az új, lokális koordinátarendszer helyzetét mutatja az X, Z munkadarab koordinátarendszerben: $v''=v'+v$. Ennek komponensei: $v''_x=80+40=120, v''_z=60+(-30)=30$.

A szerszám pozíciója az X'', Z'' koordinátarendszerben: $X''=80, Z''=120$.

A lokális koordinátarendszer eltolása az összes munkadarab koordinátarendszerben érvényesül.



10.3-2 ábra

A **G92** utasítás programozása azokon a tengelyeken, amelyeknek értéket adtunk **törli a G52** utasítás által képzett eltolásokat, mintha G52 v0 parancsot adtunk volna ki.

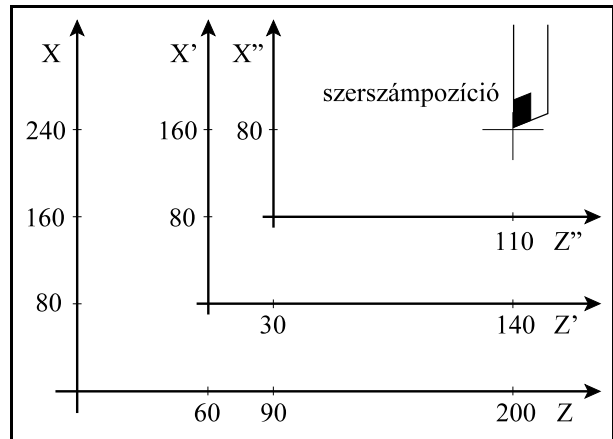
Ha a szerszám az X, Z munkadarab koordinátarendszer $X=240, Z=200$ koordinátájú pontján tartózkodik, a

```
G52 X80 Z60
```

utasítás hatására az X', Z' lokális koordinátarendszerben a pozíciója $X'=160, Z'=140$ lesz. Ezután a

```
G92 X80 Z110
```

parancs hatására az X'', Z'' új munkadarab koordinátarendszerben a szerszám pozíciója $X''=80, Z''=110$ lesz. Tehát az X', Z' lokális koordinátarendszer G92 parancs hatására törlődik, mintha G52 X0 Z0 parancsot adtunk volna ki.



10.3-3 ábra

10.4 Síkválasztás (G17, G18, G19)

A sík, amelyben

- a körinterpoláció,
- a polárkoordinátákkal történő adatmegadás,
- a koordinátarendszer síkbeli elforgatása,
- a síkbeli szerszámsugár korrekció,
- a fűróciklusok pozicionálásai,
- az esztérgáló ciklusok

történnék, a következő G kódokkal választható ki:

```
G17 XpYp sík
```

```
G18 ZpXp sík
```

```
G19 YpZp sík,
```

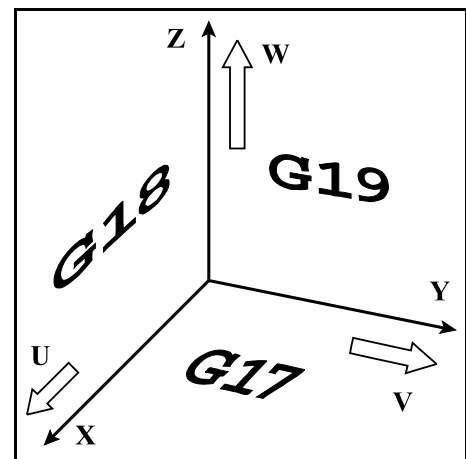
ahol: X_p : X, vagy a vele párhuzamos tengely,

Y_p : Y, vagy a vele párhuzamos tengely,

Z_p : Z, vagy a vele párhuzamos tengely.

A kiválasztott síkot nevezzük a fősíknak.

Az hogy a párhuzamos tengelyek közül melyik kerül kiválasztásra a G17, G18, vagy a G19



10.4-1 ábra

utasítással egy mondatba programozott tengelycímetől függ:

Ha például X és U, Y és V, Z és W párhuzamos tengelyek:

G17 X___ Y___ az XY síkot,
 G17 X___ V___ az XV síkot,
 G17 U___ V___ az UV síkot,
 G18 X___ W___ az XW síkot,
 G19 Y___ Z___ az YZ síkot,
 G19 V___ Z___ a VZ síkot,

választja ki.

Ha G17, G18, G19 nincs megadva egy mondatban a síkkijelölés változatlan marad:

G17 X___ Y___ XY sík
 U___ Y___ marad az XY sík.

Ha a G17, G18, G19 mondatban nincs tengelycím megadva, akkor a fő tengelyeket választja ki a vezérlő:

G17 az XY síkot,
 G17 X az XY síkot,
 G17 U az UY síkot,
 G17 V az XV síkot,
 G18 a ZX síkot,
 G18 W a WX síkot

választja ki.

A mozgásparancs nem befolyásolja a síkválasztást:

G90 G17 G00 Z100

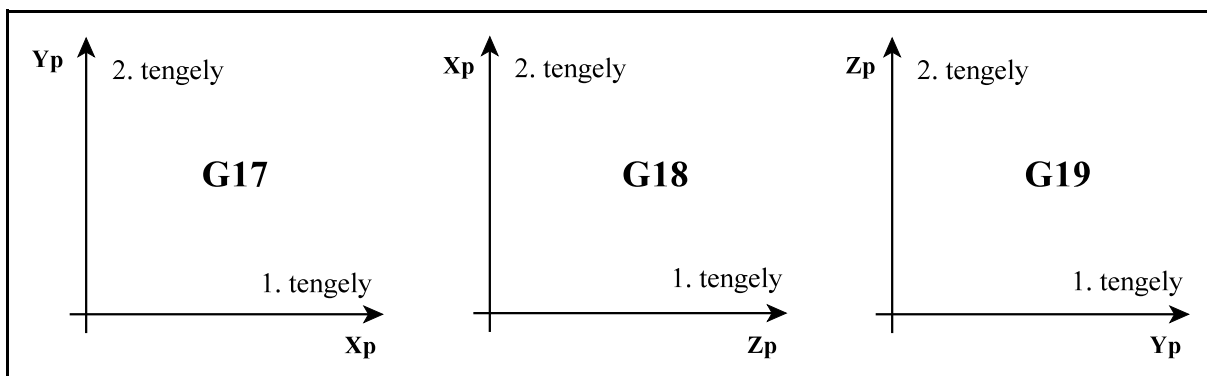
hatására az XY sík kerül kiválasztásra, és a Z tengely a 100 koordinátájú pontra mozog.

Egy programon belül többször is válthatunk síkot.

Bekapcsolás után, program végén, vagy resetre az N1300 DefaultG1 paraméter #1 G18 és #2 G19 bitjei alapján dől el, hogy melyik sík lesz érvényben.

Azt, hogy mely tengelycímet használja a vezérlő fő- és párhuzamos tengelyeknek az N0103 Axis to Plane paraméteren jelölhető ki.

A leírás gyakran hivatkozik a kiválasztott sík első és második tengelyére. Ezek értelmezése az alábbi ábrán található.



10.4-2 ábra

11 Az orsófunkciók

11.1 Az orsó fordulatszám parancs (S kód)

Az

S nnnnnnnn

címre egy maximum 8 jegyű számot írva a kódot az NC átadja a PLC-nek.

S címet a PLC, az adott szerszámgép felépítésének függvényében értelmezheti kódként, vagy fordulatszám/perc dimenziójú értéként is.

Ha mozgásparancsot és orsó fordulatszámot (S) programozunk ugyanabba a mondatba az S funkció a mozgásparancs végrehajtása alatt, vagy után kerül végrehajtásra. A végrehajtás mikéntjét a gép építője határozza meg.

Az S címen megadott fordulatszámértékek öröklődnek. Bekapcsolás után a vezérlő S0 kóddal áll fel.

A különböző orsó áttételi tartományokban az orsó fordulatszámoknak van egy minimális és egy maximális határa. Ezeket a határokat a szerszámgép építője határozza meg a paramétermezőben, és ezen tartományon kívülre nem engedi a vezérlő a fordulatszámot.

A vezérlő maximum 8 fordulatszám tartományt tud kezelni.

11.1.1 Hivatkozás több orsóra. Az S cím kiterjesztése

A vezérlő maximum 16 orsót tud kezelni.

Ha egy gépen, vagy egy csatornában több orsó is van, az S cím nem elég az orsók megkülönböztetéséhez. Több orsó kezeléséhez két lehetőséget biztosít a vezérlő.

Hivatkozás orsóra S és P címen

Az első lehetőség az S cím mellé P címen megadni az orsó számát. Az

Snnnnnnnn Pp

utasítás hatására a vezérlő átadja az S kódot és a P címre írt orsószámot a PLC-nek.

A P címet a programnyelv különböző célra használja, például várakozásra, hívott alprogram számára, stb. P címen történő orsóhivatkozásokat ezért külön mondatba kell írni, különben a P cím értelmezése nem lesz egyértelmű.

Hivatkozás több orsóra az S cím kiterjesztésével

A másik lehetőség az orsók címének kiterjesztése. Az orsókra hivatkozhatunk maximum 3 karakter megadásával is.

Az **orsók címének mindig S betűvel kell kezdődniük**. Az N0605 Spindle Name2 és az N0606 Spindle Name3 paraméteren két további karaktert lehet megadni, amelyek lehetnek az angol ABC betűi: A, B, C, D, ... Y, Z, illetve számok: 0, 1, 2, ..., 9. Ha a 2., vagy a 3. orsónevet nem használjuk a paraméterek értéke 0.

Így megadhatunk SSB orsónevet is például, de használhatjuk az S1 és S2 nevet is.

Ha az orsónév betűre végződik, a hozzá tartozó értéket mellé írhatjuk. Az

SSB12500

jelentése: az SSB orsó forogjon 12500/min fordulaton.

Ha az orsónév számra végződik a név után mindig = jelet kell írni. Az

S1=8700

jelentése: a S1 orsó forogjon 8700/min fordulaton.

A kiterjesztett orsócímmel való hivatkozás esetén a vezérlő átadja a PLC-nek a címre programozott fordulatszám értéket és a hivatkozott orsó számát.

Programban **egy mondaton belül csak egy orsóra lehet hivatkozni**. Ha több orsót kell indítani, külön mondatba kell írni a parancsokat:

S1=500 M3 (S1 500/min, óramutató járása szerint)
S2=1000 M4 (S2 1000/min, óramutató járásával ellentétesen)

Az NC az orsókat mindig a számuk alapján tartja nyilván. Az **orsók számozása és elnevezése is globális**, csatornafüggetlen.

11.1.2 Orsók csatornákhöz rendelése

Az egyes **orsókat** mindig **a PLC program rendeli** hozzá az egyes **csatornához**.

A hozzárendelés azt jelenti, hogy az adott orsóra csak abban a csatornában futó programból adhatunk ki fordulatszám parancsot. Például, ha az S4 orsó a 2. csatornához van rendelve, az 1. csatornában nem programozhatjuk az S4 címet.

Programfutás közben a PLC programnak lehetősége van egy adott orsót áthelyezni egy másik csatornába, például M funkció hatására.

Az orsók csatornákhöz rendelését és másik csatornába helyezését mindig a szerszámgép építője határozza meg.

Minden csatornában kijelölhetünk egy alaporsót az N0604 Default Spindle paraméteren. Az így kijelölt orsóra mindig hivatkozhatunk S címen is, még akkor is, ha a neve többkarakteres. Például az S2 legyen a 2-es számú orsó. Ha a 2. csatornában N0604 Default Spindle=2, akkor a 2. csatornában hivatkozhatunk az orsóra S2 és S címen is.

11.2 Az orsóvezérlő M funkciók

Beépített M funkciók

A vezérlő az alábbi beépített M kódokkal kezeli az orsókat:

M3: orsó be, óramutató járása szerinti irányba

M4: orsó be, óramutató járásával ellentétes irányba

M5: orsó ki (állj)

M19: orientálás

Az irány mindig a motor felől az orsó irányába tekintve értendő. Az M3, M4, M5 helyett írható M03, M04, M05 is.

Ezek azért beépített orsóvezérlő M kódok, mert a vezérlő a fűróciklusok végrehajtása során az orsó megállítására, irányváltására, orientálására a fenti kódokat adja át a PLC-nek.

Opcionális M funkciók

A fenti, beépített M kódokon kívül kijelölhetők az orsók kezelésére további M kódok is. Ezeket a kódokat egy tömbben az N0689 Spindle M Low és az N0690 Spindle M High paramétereken lehet beállítani. Az Spindle M Low paraméterre az M kód tömb legkisebb értékét, a Spindle M high paraméterre a tömb legnagyobb értékét írjuk.

Például legyen a Spindle M Low S2=20, a Spindle M High S2=24. Az M kódok funkciója legyen a következő:

M20: orsó C tengellyé alakítása

M21: orsó szinkronizálása

M22: orsó szinkronizálása fázistolással

M23: orsó előkészítése sokszögessztergáláshoz

M24: orsó pozícióhurok zárás orientálás nélkül (M Code for Closing S Loop)

☞ **Figyelem!** A fenti M kód tömb csak egy példa. Az orsók opcionális M funkcióit mindig a gép építője határozza meg, ezért ezek leírását keresse az adott gép gépkönyvében!

A beépített és az opcionális, paraméteren kijelölt M funkciókat a vezérlő egymást kizáró funkcióknak tartja nyilván. Ez azt jelenti, hogy egy mondatba csak egy ilyen M kód írható.

Az orsóvezérlő M funkciók mindig az utoljára programozott orsóra vonatkoznak:

S1=1500 M4 (S1 be M4 irányba)

S2=2000 M3 (S2 be M3 irányba)

M5 (S2 állj)

M19 (S2 orientálás)

S1=0 M5 (S1 állj)

A fenti példában látszik, ha már hivatkoztunk S2 orsóra, a további, orsóvezérlő M kódok az S2 orsóra vonatkoznak. Ha viszont az S1 orsót le akarjuk állítani, hivatkozni kell az orsó címére!

11.3 A fordulatszám tartományok kezelése

Az orsómotor és az orsó között **változtatható fokozatú áttétel** lehet, amellyel az **orsó fordulatszám tartománya változtatható**. A vezérlő orsónként maximum 8 fordulatszám tartományt tud kezelni. Minél kisebb fordulatszám tartományban van az orsó, annál nagyobb nyomatékkal tud forgácsolni.

Minden áttételi tartományhoz beállítható egy megengedhető minimális és maximális fordulatszám, amely alá, illetve fölé nem engedi a vezérlő az orsó fordulatszámát.

Az egyes fokozatok között a megengedhető fordulatszámok átfedhetik egymást.

Ha a fordulatszám tartományok nem fedik át egymást

Például:

1. tartomány minimális fordulata: 50/min
1. tartomány maximális fordulata: 1000/min
2. tartomány minimális fordulata: 1001/min
2. tartomány maximális fordulata: 4000/min

A fenti esetben az S900 kód programozása alapján egyértelmű, hogy az orsót az 1. tartományban kell forgatni.

Ha a tartományok nem fedik át egymást, a vezérlő átadja a PLC-nek az **S cím** értékével és a hivatkozott orsó számával együtt a tartomány kódját is és a PLC **automatikusan beváltja** a szükséges tartományt.

Ha a fordulatszám tartományok átfedik egymást

Például:

1. tartomány minimális fordulata: 50/min
1. tartomány maximális fordulata: 1000/min
2. tartomány minimális fordulata: 800/min
2. tartomány maximális fordulata: 4000/min

Az S900 kód programozása alapján nem egyértelmű, hogy melyik, az 1. vagy a 2. tartományban kell az orsót forgatni.

Ha a tartományok átfedik egymást a programozónak **M funkcióval kell kiválasztania** azt a tartományt, amelyben az orsót forgatni szeretné.

Ezek a beépített M funkciók a következők:

M11: 1. tartomány választása

M12: 2. tartomány választása

...

M18: 8. tartomány választása

Egy adott gépen a tartománykezelést a gép építője határozza meg, amelynek leírását az adott gép gépkönyve tartalmazza.

11.4 A főorsó. A főorsó kiválasztása

Ha egy gépen, vagy egy csatornában több orsó van, el kell dönteni, hogy melyik legyen az orsók közül a "főorsó". A **főorsó**nak kijelölt orsóra vonatkoznak az alábbi funkciók:

előtolás engedélyezése,
fordulatonkénti előtolás,
konstans vágósebességszámítás,
menetvágás,
merevszárú menetfúrás,
sokszögesztergálás mesterorsója.

Minden csatornában **ki kell jelölni a főorsót**. Olyan orsót is ki lehet jelölni főorsónak egy adott csatornában, amely egy másik csatornához tartozik, ezért ebben a csatornában nem programozható, de pl. a fordulatonkénti előtolást erről az orsóról kell venni.

Azt, hogy a **főorsót hogyan kell kiválasztani**, az adott gép PLC programja dönti el, és **a gépgyártó leírása** tartalmazza. Történet például M funkcióval:

M31 (az 1. orsó a főorsó)

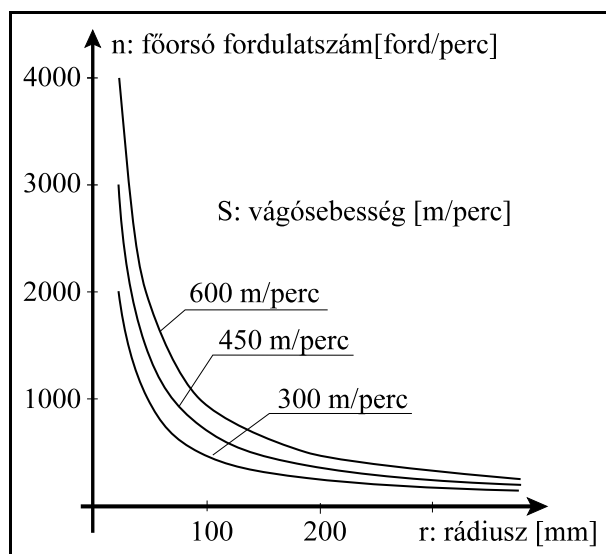
M32 (a 2. orsó a főorsó)

11.5 A konstans vágósebességszámítás

A konstans vágósebesség számítási funkciót csak fokozatmentes orsójajtás esetén lehet használni. A vezérlő ekkor a főorsó fordulatszámát úgy változtatja, hogy a szerszámnak a darab felületéhez viszonyított sebessége mindig állandó, és egyenlő a programozott értékkel.

Mindig a főorsónak kijelölt orsó fordulatszámát kezeli.

A konstans vágósebesség értékét a bemenő mértékrendszer függvényében az alábbi táblázat alapján kell megadni:



11.5-1 ábra

bemenő mértékegység	konstans vágósebesség mértékegysége
mm (G21 metrikus)	m/min (méter/perc)
inch (G20 inches)	feet/min (láb/perc)

11.5.1 A konstans vágósebesség számítás megadása (G96 S, G97)

A

G96 S

utasítás bekapcsolja a konstans vágósebesség számítást. S címen a konstans vágósebesség értékét kell megadni a fenti táblázatban megadott mértékegységben.

A konstans vágósebesség értékét mindig S címen kell megadni és a fordulatszám számítása a főorsónak kijelölt orsóra vonatkozik.

Például:

M32 (2. orsó kijelölése főorsónak)
G96 S300 (300 m/min vágósebesség)

(M32 itt csak egy példa, a főorsónak való kijelölés módját az adott szerszámgép gépkönyve tartalmazza.)

A

G97 S

utasítás kikapcsolja a konstans vágósebesség számítást.

S címen a kívánt főorsó fordulatot lehet megadni (fordulat/perc mértékegységben). Több orsó esetén G97 programozása után mindig a főorsó címét kell használni S helyett. Pl:

G97 S2=1200 (2. orsó fordulatszáma 1200/min)

- A konstans vágósebesség számításához annak a tengelynek a nullpontját, amelynek pozíciója alapján a főorsó fordulatszámát változtatni kell, a főorsó forgástengelyére kell állítani.
- A konstans vágósebesség számítás csak azután határos, hogy a főorsót M3-mal, vagy M4-gyel elindítottuk.
- A konstans vágósebesség értéke öröklődik, még azután is, hogy G97 utasítással kikapcsoltuk a számítását.
 - G96 S100 M3 (100m/min, vagy 100 láb/min)
 - G97 S1500 (1500 ford/min)
 - G96 X260 (100m/min, vagy 100 láb/min)
- A konstans vágósebességszámítás érvényes G94 (előtolás/perc) módban is.
- Ha a konstans vágósebesség számítást kikapcsoltuk G97 paranccsal és nem adtunk meg új főorsó fordulatot akkor a G96 állapotban felvett utolsó főorsófordulat marad érvényben.
 - G96 S100 (100m/min, vagy 100 láb/min)
 - .
 - .
 - .
 - G97 (a kiadódó X átmérőhöz tartozó fordulatszám)
- Gyorsmeneti pozicionálás esetén (G0 mondat) az orsó fordulatszáma nem kerül folyamatosan kiszámításra, hanem a vezérlő a pozicionálás végpontjában esedékes pozícióhoz tartozó fordulatszámot állítja be.

- A konstans vágósebesség értékét bekapcsolás után az az N0686 Default Surf Speed paraméter határozza meg.

11.5.2 A fordulatszám korlátozása konstans vágósebességszámításkor (G92 S)

A

G92 S

utasítással a konstans vágósebesség számításkor megengedhető legmagasabb főorsó fordulatszámot állíthatjuk be. A vezérlő a konstans vágósebességszámítás bekapcsolt állapotában az itt megadott értéknél nagyobb főorsó fordulatot nem enged kiadni. S mértékegysége ebben az esetben: ford/perc.

A főorsó fordulatszám maximális értékét mindig S címen kell megadni és a fordulatszám korlátozása a főorsónak kijelölt orsóra vonatkozik.

- Bekapcsolás után, illetve, ha a fordulatszám értékét nem határoltuk be G92 paranccsal a főorsó fordulatszám felső határa konstans vágósebességszámítás esetén az adott tartományra megengedhető maximális érték.
- Konstans vágósebességszámítás esetén a főorsó fordulatszámának megadhatunk egy alsó határértéket is az N0688 Min Spindle Speed G96 paraméteren, amely nagyobb lehet, mint a tartományhoz tartozó minimális fordulatszám értéke.
- A maximális fordulatszám értéke öröklődik, mindaddig amíg újat nem programozunk.

11.5.3 Tengely kijelölése konstans vágósebesség számításához (G96 P)

Azt a tengelyt, amelynek pozíciója alapján az orsó fordulatszámot számolja a vezérlő G96 állapotban az N0687 Default G96 Axis paraméter jelöli ki.

Ha a kijelölt tengelytől el akarunk térni, a

G96 P

utasítással adhatjuk meg azt a tengelyt, amelyikről a vágósebességet számítani akarjuk.

A **P cím** értelmezése **tengelyszám**.

A G96 utasításban együtt lehet programozni S és P címet is:

G96 S300 P4 (300m/min vágósebesség a 4. tengellyel)

A P címen beállított érték öröklődik.

11.6 Az orsók fordulatszám ingadozás figyelése

Minden orsó fordulatszám ingadozását figyeli a vezérlő. Az override-dal és a fordulatszámhatárokkal módosított programozott fordulatszám és az aktuális, jeladóról mért fordulatszám különbségéből határozza meg az ingadozást.

Ha a gép építője által, paraméteren beállított túréstartományból kiesik az orsó fordulatszáma, az NC üzenetet küld a PLC-nek.

Ezután a PLC program üzen hibát és intézkedik az orsó és a forgácsolás leállításáról. Ezeket a gép építőjének leírása tartalmazza.

- A fordulatszám ingadozás figyelési funkció csak akkor működik, ha az orsóra jeladó van szerelve.
- A fordulatszám ingadozás figyelés csak forgó orsónál (M3, vagy M4 állapotban) hatásos.

11.7 Az orsók pozicionálása

Normál megmunkálás esetén az NC az orsók hajtásainak a programozott fordulatszámmal arányos fordulatszám parancsot ad ki. Az orsóhajtás ekkor fordulatszám szabályzott üzemmódban dolgozik.

Bizonyos technológiai feladatoknál szükség lehet arra, hogy egy orsót meghatározott szöghelyzetbe állítsunk. Ezt nevezzük orsó pozicionálásnak, vagy indexelésnek.

Pozicionáltatás előtt az NC az orsót pozíciószabályzott üzemmódba kapcsolja. Ez a gyakorlatban azt jelenti, hogy többé már nem az S kóddal arányos fordulatszám parancsot ad ki az NC, hanem az orsóra szerelt szöghelyzetadó (jeladó) segítségével méri az orsó helyzetét és a kívánt szögelfordulás függvényében ad ki parancsot a hajtásnak, mint a többi pozíciószabályzott tengelyen. Ez a pozícióvisszacsatolás.

Ahhoz, hogy egy adott gépen az orsót pozicionálni lehessen az orsóra jeladót kell szerelni, illetve az orsóhajtásnak olyannak kell lennie, hogy pozícióvisszacsatolt üzemmódban is működjön.

11.7.1 Az orsók orientálása

Orsó orientálásnak, vagy orientált orsó megállásnak nevezzük azt a funkciót amikor az orsót adott szöghelyzetben állítjuk meg. Erre például automata szerszámcsere esetén, illetve egyes fűróciklusok végrehajtásához lehet szükség.

Azt, hogy egy adott orsó orientálható az N0607 Spindle Config paraméter #1 ORI=1 bitállítása jelenti.

Az orientálás parancsot az

M19

funkcióval adjuk ki. Ha a gépen, vagy a csatornában több orsó is van, az M19 mellett az orsót is ki kell választani. Például:

S2=0 M19

Az orientáció műszakilag kétféleképp történhet.

Ha az orsó nem csatolható vissza pozíciószabályzásra (N0607 Spindle Config #2 INX=0 paraméterállás) az orientáció például a gépre szerelt helyzetkapcsolóra való ráfordulás segítségével történhet.

Ha az orsó visszacsatolható pozíciószabályzásra (N0607 Spindle Config #2 INX=1 paraméterállítás) az M19 parancs hatására a vezérlő megkeresi az orsó jeladó nullimpulzusát. Ezután a vezérlő automatikusan elvégzi a pozíciószabályzó kör zárását.

Ez egyben az orsó retpontfelvételét is jelenti, azaz orientálás után abszolút szöghelyzetre küldhető az orsó.

11.7.2 Az orsók megállítása és a pozíciószabályzó hurok zárása

Az N0607 Spindle Config #2 INX=1 paraméterállítás azt jelenti, hogy a pozíciószabályzó hurok zárható.

Ebben az esetben az N0823 M Code for Closing S Loop paraméteren megadható egy M kód, amelynek hatására az orsó megáll, zárja a pozíciószabályzó hurkot, de nem áll orientálási pozícióra (nem keresi meg a jeladó nullimpulzusát).

Például, ha a paraméter értéke 24 a hurokzárás az

M24

parancs hatására történik. Ha a gépen, vagy a csatornában több orsó is van, az M funkció mellett az orsót is ki kell választani. Például:

S2=0 M24

A funkció kódjáról és működéséről a szerszámgép építője tud felvilágosítást adni. Ez a funkció pl. a merevszárú menetfűróciklusok végrehajtását gyorsíthatja.

11.7.3 Az orsók pozícionálásának programozása

Az N0607 Spindle Config #2 INX=1 paraméterállítás azt jelenti, hogy a pozíciószabályozó hurok zárható. Csak ebben az esetben van lehetőség az orsók pozícionálására.

A paramétert a gép építője állítja be abban az esetben, ha az adott orsón a funkció meg lett valósítva.

Pozícionálás tengelynév alapján

Minden orsónak adhatunk egy maximum 3 karakter hosszúságú tengelynevet, amire a pozíciószabályozó hurok zárása után hivatkozhatunk.

Az N0817 Spindle Axis Name1 paraméteren kötelezően A, B, vagy C betűt kell beállítani. A második (N0818 Spindle Axis Name2 paraméter) és a harmadik karakteren (N0819 Spindle Axis Name3) megadhatjuk az angol ABC betűit: A, B, C, D, ... Y, Z, illetve számokat: 0, 1, 2, ..., 9. Például:

CS: az 1. orsótengely neve

CS2: a 2. orsótengely neve

ABC: a 3. orsótengely neve

Számra végződőnevek esetén az = jelet kell használni. A megadott név, természetesen nem ütközhet más nevekkel.

A cím után a **pozíciót fokban** kell megadni. A pozícionálás az orsótengelyre beállított gyorsmeneti sebességgel történik. A gyorsmenet mértékegysége 1/min.

Abszolút adatmegadással való **pozícionálás** előtt M19-et kell programozni. Például:

```
S2=0 M19
```

```
G90 CS2=30
```

utasítással a CS2 orsót a 30 fokos pozícióra állítja, az orsót pozitív irányba forgatva.

Abszolút pozícionálás esetén a 360 foknál nagyobb adatból **levágja az egész fordulatokat**, pl.

```
S2=0 M19
```

```
G90 CS2=750
```

megadására 30 fokra forgatja az orsót. **Mindig a rövidebb út irányában** forgat:

```
S2=0 M19
```

```
G90 CS2=270
```

esetén negatív irányban forgat 270 fokra.

Inkrementális adatmegadással való **pozícionálás** előtt nem kell M19-et programozni, elég csak a pozícióhurkot zárni. Ha a hurokzárás M24-re történik:

```
S2=0 M24
```

```
G91 CS2=3600
```

vagy

```
S2=0 M24
```

```
CS2=I3600
```

adatmegadás esetén az orsó a megadott (a példában pozitív irányban) 10 fordulatot tesz meg. Inkrementális adatmegadás esetén a vezérlő **nem vágja le az egész fordulatokat és a megadott előjel szerinti irányba megy**.

Orsók indexelése M funkcióval

M kódok használatával az orsókat lehet indexelni. Indexelésnek nevezzük azt, amikor az orsót diszkrét, előre beállított pozíciókra forgatjuk.

Ehhez a következő paramétereket kell beállítani:

Az N0607 Spindle Config paraméter #7 IDS bitjén az indexelés irányát: =0 pozitív, =1 negatív.

Az N0820 Start M of Spnd. Pos. paraméteren az M kódok kezdőértékét, az N0821 No. of M Code for Spnd. Pos. paraméteren az M kódok számát.

Az N0822 Basic Angle of Spnd. Pos. paraméteren az indexelés szögét.

Példa:

Az orsó az orientálási pozíciótól számítva 18 fokként rögzíthető. A beállítás a következő:

Spindle Config: #7 IDS=0: pozitív irányban indexel

Start M of Spnd. Pos.=201 (m=201)

No. of M Code for Spnd. Pos.=20 (n=20)

Basic Angle of Spnd. Pos.=18 ($\varphi=18$)

Az alábbi táblázat a különböző M kódok jelentését adja meg:

M kód	Elforgatott szög α
Mm (M201)	$\alpha=\varphi=18^\circ$
M(m+1) (M202)	$\alpha=2\varphi=36^\circ$
M(m+2) (M203)	$\alpha=3\varphi=54^\circ$
....	
M(m+n) (M220)	$\alpha=n\varphi=360^\circ$

Vegyük az alábbi példát a fenti beállítások alapján:

M19 (orientálás)
M205 (elforgatás 90° -kal)
... (fúrás)
M210 (elforgatás 180° -kal)
... (fúrás)

Mielőtt pozicionáltató M kódot adnánk ki, M19-cel orientáljuk az orsót. Ekkor az orsó az $\alpha=0$ pozícióba kerül.

90° -ra lyukat kell fúrni, M205 parancsra $\alpha=5\varphi=5*18=90^\circ$ ra áll az orsó.

A következő furatot 270° -os pozícióban kell fúrni. Mivel M kóddal csak inkrementálisan mozgathatjuk az orsót M210-et programozunk, mert ekkor az elmozdulás $10*18^\circ=180^\circ$ lesz.

11.7.4 Két orsó pozícióhelyes szinkronizálása

Két orsót úgy lehet szinkronizálni, hogy mindkét orsó ugyanazon a fordulaton forogjon és egymáshoz képest egy beállított fázistolásban legyenek forgás közben.

Két orsó pozícióhelyes szinkronizálását a PLC program általában egy M kóddal valósítja meg. A szinkronizálás módját a gép építőjének a leírása tartalmazza.

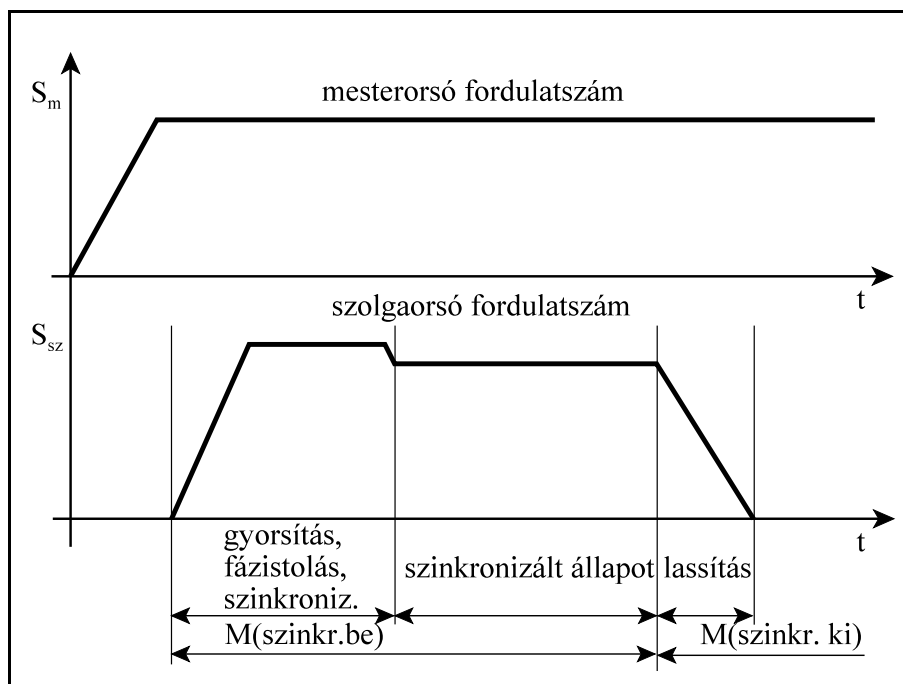
Két orsó között a szinkronizáláshoz az kell, hogy mindkét orsót jeladóval szereljék és a pozíciószabályozó hurok zárható legyen.

Szinkronizálás alatt az *orsók maximális fordulatszámát* az orsótengelyre beállított *gyorsmeneti*

sebesség korlátozza. Ez általában alacsonyabb, mint a maximális orsó fordulatszám. A szinkronizálás során megkülönböztetjük a mester és a szolga orsót. Mindig a szolga szinkronizálódik rá a mesterre.

A szinkronizáció menete a következő:

- ha a mester, fordulatszáma nagyobb, mint a két orsóra (mester és szolga) megadott gyorsmeneti fordulatszám közül a kisebb, lelassít a megfelelő fordulatra,
- zárja a pozíciószabályozó hurkot,
- a szolgálorsó felpörög a mester fordulatszámára, a mesterrel azonos, vagy ellentétes forgásirányban,
- a szolgálorsó zárja a pozíciószabályozó hurkot,
- majd a szolga ráhúzza a nullimpulzusát a mester nullimpulzusára, vagy a mester nullimpulzusától az N0685 Spindle Phase Shift paraméteren jeladóimpulzusban megadott távolságra, fázistolásra áll.



11.7.4-1 ábra

Példa

Ellenorsós esztergán a főorsóban (jelöljük S1-gyel) megmunkált darab másik oldalát is forgácsolni kell. A főorsó rúdból dolgozik. Először az ellenorsót, a szolgát (jelöljük S2-vel) szinkronizálni kell a főorsóhoz (mester), ráfogni a darabra, majd leszúrni.

Legyen M21 a nullimpulzusok együttfutásával megvalósított szinkronizálás és M22 a fázistolásos szinkronizálás.

Ha nem kell fázistolás a programrészlet a következő:

```
S1=3000 M3
S2=0 M21 (S2 szinkronizálása az S1-hez)
... (ráfogás a darabra az S2 tokmánnal)
... (leszúrás)
S2=1200 M4 (szinkronizálás kikapcsolása, S2 forgatása)
```

Ha alakos darabot kell megfogni az M22 kódot kell használni és az N0685 Spindle Phase Shift paraméteren kell a fázistolást beállítani.

☞ **Figyelem!** A fenti példa csak egy minta. Konkrét esetben a gép építőjének leírása szerint kell eljárni!

11.7.5 Az orsók pozíciószabályozott üzemmódjának kikapcsolása

Az orsókat orientálás, pozícionálás, vagy szinkronizálás után az alábbi funkciókkal kell kikapcsolni pozíciószabályozott üzemmódból:

M3, M4, vagy M5

Például:

S1=0 M19 (pozíciószabályozott üzem be, S1 orientálás)
CS1=60

...

S1=0 M5 (pozíciószabályozott üzem ki, orsó áll)

vagy

S1=2400 M3 (pozíciószabályozott üzem ki, orsó be)

11.8 Orsó tengellyé, tengely orsóvá alakítása

Az orsótengelyek csak korlátozott módon használhatók megmunkálásra, mert csak pozícionálni lehet őket, más tengelyekkel való interpolációban nem vehetnek részt. Ezenkívül az orsótengelyek kis felbontású, nagy fordulatszámú működő jeladót használnak, míg egy körasztalhoz nagy felbontású útmérő szükséges, ami kis fordulatszámú működik.

A megmunkálás során ezért szükség lehet egy orsó tengellyé, vagy egy tengely orsóvá alakítására. Eszterga orsóját C tengellyé kell alakítani ahhoz, hogy polárkoordináta interpolációval marni tudjunk a darab homlokfelületére, vagy hengerinterpolációval gravírozunk valamit az oldalára. Ezután a C tengelyt vissza kell alakítani orsóvá, hogy további esztergálást végezhesünk.

Horizontális megmunkálóközpont B körasztalát, ha ezt a gép kialakítása lehetővé teszi, orsóvá lehet alakítani, hogy a darabon esztergálási műveletet tudjon a gép végezni.

Orsónak tengellyé, illetve tengelynek orsóvá alakításához a szerszámgép megfelelő mechanikai és elektronikai felépítésére van szükség. Azt, hogy az adott gépen ez lehetséges-e, az adott gép leírása tartalmazza. Az átváltást mindig a PLC program végzi az adott gép igényeinek megfelelően.

Egy **orsó tengellyé** alakításának menete általában a következő:

- leállítja az orsót, ha az forog,
- leállítja a hajtás működését,
- lekapcsolja az orsót a hajtásról,
- nagyfelbontásúvá alakítja a jeladót,
- átállítja a hajtáson a paramétereket,
- visszakapcsolja a hajtást,
- rákapcsolja a tengelyt a hajtásra és a kijelzését láthatóvá teszi.

Egy **tengely orsóvá** alakításának menete általában a következő:

- megvárja, amíg leáll a tengely,
- leállítja a hajtás működését,
- lekapcsolja a tengelyt a hajtásról és a kijelzését láthatatlanná teszi,
- kifelbontásúvá alakítja a jeladót,

- állítja a hajtáson a paramétereket,
- visszakapcsolja a hajtást,
- rákapcsolja az orsót a hajtásra.

Példa:

Esztergán alakítsuk az S1 orsót C1 tengellyé maráshoz, majd alakítsuk vissza a C1 tengelyt S1 orsóvá további esztergáláshoz:

```
S1=0 M20      (S1 átalakítása C1-gyé)
G28 G91 C1=0  (refpontfelvétel a C1 tengelyen)
...          (marás C1 tengely használatával)
S1=3000 M3    (C1 visszakapcsolása S1-gyé, forgatás 3000-
rel)
...          (esztergálás S1-gyel)
```

☞ **Figyelem!** A fenti példa csak egy minta. Konkrét esetben a gép építőjének leírása szerint kell eljárni!

12 A T funkció

Szerszámszámra **T kód**dal hivatkozunk az alkatrészprogramban. A T kódot maximum **8 decimális számjegyen** lehet megadni:

Tnnnnnnnn

A vezető nullák elhagyhatók.

12.1 A szerszámváltás programozására

Az alkatrészprogramban alapvetően kétféle módon lehet a szerszámváltásra hivatkozni. A kétféle módszer a szerszám gép felépítésétől függ. Az alkatrészprogramban alkalmazható szerszámhívási technikát a szerszám gép építője adja meg.

A. eset: csere T kódra

A szerszámcsere a gépen kézzel, vagy revolver típusú szerszámváltóval történik.

Az N1414 Comp. Config on Lathes paraméter #0 TCM=0 bitállásánál a **T kód a szerszámkorrekciós kódját is tartalmazza** az alsó helyiértékű számjegyeken. A maradék felső helyiértékeken adható meg a szerszám száma. A TCM=0 bitállásnál a szerszámszám hívására szerszámcsere is történik.

Azt, hogy a szerszámkorrekciót hány számjegyen kell megadni, az N1413 No. of Digits of Offs. No. in T Code paraméter dönti el. A paraméter értéke 0, 1, 2, 3 lehet.

Tnnnnnnnk: korrekció 1 számjegyen

Tnnnnnnkk: korrekció 2 számjegyen

Tnnnnnkkk: korrekció 3 számjegyen

ahol:

nnnnnn: a szerszám száma

kk: a korrekciós rekesz száma

Ha a korrekciót 2 számjegyen adjuk meg a

T1236

parancs jelentése:

12-es számú szerszámot váltsd be és a

36-os számú korrekciós csoportot

hívd le mellé.

Ha a szerszám számára 0-t programozunk, vagy csak annyi számjegyet programozunk T címre, ahány számjegyen kell a korrekciót megadni, szerszámváltás nem történik, csak új korrekciót hívunk le. A fenti beállításnál maradva, például

T12

jelentése: a 12-es korrekciós csoportot hívd le és az előző szerszám marad beváltva.

Ha mozgásparancsot és szerszámszámot (T) programozunk ugyanabba a mondatba, a T funkció a mozgásparancs végrehajtása közben, vagy után kerül végrehajtásra. A végrehajtás mikéntjét a gép építője határozza meg.

A szerszám száma a PLC programnak kerül átadásra.

B. eset: csere M6 funkcióra

Ha a szerszámcsere a gépen szerszámelőkészítést igényel, azaz a szerszámok magazinban vannak és egy karos cserélővel lehet őket beváltani, a szerszámszám nem tartalmazhatja a korrekció kódját, mert a T kód csak előkészíti a szerszámot cseréhez és a csere később M6 funkció hatására fog megtörténni.

Ebben az esetben az N1414 Comp. Config on Lathes paraméter #0 TCM=1 bitállásánál a **T kód a szerszámkorrekció kódját nem tartalmazza.**

A szerszámcsere a gépen szerszámelőkészítést igényel. Ezek lépései a következők:

- A szerszámtárban a beváltandó szerszámot meg kell keresni. Ekkor az alkatrészprogramban

Tnnnnnnnn

címre történő hivatkozás a megfelelő szerszámot cserehelyzetbe hozza. Ez a művelet a háttérben folyik párhuzamosan a megmunkálással.

- A szerszámokat cserepozícióba kell küldeni.

- A szerszámcsere végrehajtása

M6

funkcióval történik a programban. (M06 is használható.) A szerszámcsere végrehajtására addig vár a vezérlő, amíg az előkészítés alatt levő T szerszám cserehelyzetbe nem kerül. Ennek hatására az új szerszámot beteszi a szerszámtartóba. Innen folytatódhat a forgácsolás.

- A régi szerszámot visszateszi a szerszámtárba. Ez a tevékenység a háttérben zajlik, a forgácsolással párhuzamosan.

- Elkezd keresni a szerszámtárban az új szerszámot.

M6 funkcióra történő csere esetén az alábbi paraméterbeállítások szükségesek:

Mondatkereséshez N1338 Block No Search #0 M06=1 bitállítás,
a szerszáméltartam kezeléshez N2901 Search Config #1 TLC=0 bitállítás.

Mintapélda:

```
T12 M6      (beváltja a T12-es szerszámot)
T15         (a T15-ös szerszám hívása, a PLC megmunkálás köz-
            ben keres)
...         (forgácsolás a T12-es szerszámmal)
M6          (a T15-ös szerszám beváltása)
T8          (a T8-as szerszám hívása, a PLC megmunkálás köz-
            ben keres)
...         (forgácsolás a T15-ös szerszámmal)
T9          (a T9-es szerszám hívása, a PLC megmunkálás köz-
            ben keres)
M6          (a T8-as szerszám beváltása)
...         (forgácsolás a T8-as szerszámmal)
```

Az egy mondatba írt T és M6 parancs esetén általában a PLC először a T, majd az M6 funkciót hajtja végre, azaz beváltja a szerszámot. Szerszámváltás, M6 után, célszerű a következő mondatban szerszámhívást T-t programozni, hogy a gép mellékidejét minimalizáljuk.

☞ **Figyelem!** A fenti példa csak egy minta. Konkrét esetben a gép építőjének leírása szerint kell eljárni!

13 Vegyes és segédfunkciók

13.1 Vegyes funkciók: M kódok

M cím után egy legfeljebb 8 jegyű számértéket írva az NC a kódot átadja a PLC-nek.

Mnnnnnnnn

Az M kódokban a vezető nullák elhagyhatók.

A vezérlő egyszerre 8 különböző M kódot képes átadni a PLC-nek, vagyis *egy mondatba maximum 8 M kód* írható.

Az egy mondatba írt M funkciók *végrehajtási sorrendjét* a gép építője határozza meg a **PLC programban**.

Ha mozgásparancsot és vegyes funkciót (M kódot) programozunk ugyanabba a mondatba a vegyes funkció a mozgásparancs végrehajtásával párhuzamosan, vagy a mozgásparancs végrehajtása után kerül végrehajtásra.

Minden M kódot átad a vezérlő a PLC-nek, még azokat is, amelyeket a vezérlő hajt végre.

A végrehajtás mikéntjét a gép építője határozza meg.

A programvezérlő M kódok:

M0: programozott stop

A PLC hajtja végre. Azon mondat végén, amelyikben az M0 megadásra került általában az NC Stop állapotba kerül, az orsókat leállítja, a hűtővizet kikapcsolja.

Az összes öröklődő funkció változatlan marad. Start hatására újraindítja az orsókat, visszakapcsolja a hűtővizet és folytatja a programot.

M1: feltételes stop

A PLC hajtja végre. Hatása azonos az M0 kód hatásával. Csak a FELTÉTELES ÁLLJ gomb bekapcsolt állapotában áll meg. Ha a megfelelő gomb nincs bekapcsolva az M1 hatástalan.

M2, M30: program vége

A PLC hajtja végre. A főprogram végét jelenti. A gép funkcióinak alaphelyzetbe hozásáról a PLC program gondoskodik, általában leállítja az orsók forgását és kikapcsolja a hűtővizet.

Minden végrehajtott M2 vagy M30 eggyel növeli a munkadarab-számlálókat, hacsak az N2305 Part Count M paraméteren más M kódot nem jelöltünk ki a számláló léptetésére.

M96: megszakítási makró engedélyezés

A kód átadódik a PLC-nek, de az NC hajtja végre. Engedélyezi a PLC-től jövő megszakítási jelet, amelynek hatására meghívódik a megszakítási makró.

M97: megszakítási makró tiltás

A kód átadódik a PLC-nek, de az NC hajtja végre. Tiltja a PLC-től jövő megszakítási jel érvényesülését és a megszakítási makró futását.

M98: alprogram hívás

A kód átadódik a PLC-nek, de az NC hajtja végre. Hatására alprogramhívás történik.

M99: alprogram vége

A kód átadódik a PLC-nek, de az NC hajtja végre.. Hatására a végrehajtás visszatér a hívás helyére.

Orsóvezérlő M kódok

M3, M4, M5, M19: orsókezelés kódjai

A PLC hajtja végre. Lásd “Az orsóvezérlő M funkciók” fejezetet.

Paraméteren megadható orsókezelő M kódok

További **orsóvezérlő M kódok**at jelölhetünk ki egy tömbben az N0689 Spindle M Low és az N0690 Spindle M High paramétereken. Lásd “Az orsóvezérlő M funkciók” fejezetet. Az **orsóvezérlő M kódok** az **M3, M4, M5, M19** kódokkal együtt **egymást kizáró funkciók**, egy mondatba csak egy ilyen kód adható meg.

A PLC hajtja végre.

Orsók indexelését végző M kódok kezdőértékét az N0820 Start M of Spnd. Pos. paraméteren, az indexelést végző M kódok számát N0821 No. of M Code for Spnd. Pos. paraméteren lehet megadni. Lásd “Az orsók pozicionálásának programozása” fejezetet.

Az NC hajtja végre.

Fordulatszám tartományokat kezelő M kódok

M11, ..., M18: orsók tartományváltási kódjai

A PLC hajtja végre. Lásd “A fordulatszám tartományok kezelése” fejezetet.

Szerszámcsere M kódja

M6: szerszámcsere kódja

A PLC hajtja végre.

Paraméteren beállítható M kód csoportok

16 paraméterpáron **16 különböző M kód csoport** jelölhető ki. A PLC hajtja végre.

Az N1341 M GR Low 1, ..., N1356 M GR Low 16 paraméterekre a csoport legkisebb számú, az N1357 M GR High 1, ..., N1372 M GR High 16 paraméterekre a csoport legnagyobb számú kódját kell írni.

Az **M kód csoportok**at úgy kell megadni, hogy azok **egymást kizáró gépállapotot** jelentő kódok legyenek.

Program végrehajtásakor az NC az **M kódokat úgy szűri**, hogy az egy csoportba tartozó M kódok közül csak egy lehet az adott mondatban, egyébként *Ellentmondó M-kódok* hibaiüzenetet ad.

A paraméteren beállított értékeket **mondatkeresés** során is figyelembe veszi a vezérlő, az M kódok kigyűjtésénél. Az egy csoportba tartozó M kódok közül csak az utolsónak megadott kódot gyűjti ki.

Ezeknek az M kódoknak az értékeit a PLC program az **M kódok ablakban** a képernyőre is kiírja. Például:

Legyenek a tokmány nyitás, zárás M kódjai:

M51: tokmány zárás

M52: tokmány nyitás

M53: tokmány nyitás forgó orsónál

A paraméterek a következőképp vannak beállítva:

N1341 M GR Low 1=51

N1357 M GR High 1=53

A tokmány szorítás M kódjai:

M54: befele szorítás

M55: kifele szorítás

A paraméterek a következőképp vannak beállítva:

N1342 M GR Low 2=54

N1358 M GR High 2=55

Alkatrészprogramban egy mondatba csak M51, vagy M52, vagy M53 írható, különben a vezérlő programfutáskor hibát jelez. Ugyanez vonatkozik az M54, M55 csoportra is.

Mondatkeresés során M51, M52, M53 kódok közül csak az utolsónak programozottat gyűjti ki és hajtja végre. Ugyanez vonatkozik az M54, M55 csoportra is.

A megfelelő gépállapotok közül mindig csak a csoportban érvényes állapotkód kerül kiírásra az M funkciók ablakban.

Csatornák szinkronizálását végző M kódok

Az N2201 Waiting M Codes Min és az N2202 Waiting M Codes Max paraméteren ki lehet jelölni egy M kód csoportot, amellyel a csatornák közötti szinkronizálást, összevárást lehet megvalósítani. Az NC hajtja végre.

13.2 Segédfunkciók (A, B, C, U, V, vagy W)

Az M, S, T cím mellé paraméteren kijelölhetünk még további 3 címet, amelyen segédfunkciót lehet átadni a PLC program számára. A vezérlő mindhárom segédfunkciót képes egyszerre átadni. Az N1333 Aux Fu Addr1, N1334 Aux Fu Addr2, N1335 Aux Fu Addr3 paraméteren az A, B, C, U, V, W címek közül kiválaszthatunk egyet-egyet, amely címeken segédfunkciót adhatunk át. A segédfunkcióknak maximum 8 decimális számjegyen adhatunk értéket.

Ha mozgásparancsot és segédfunkciót programozunk ugyanabba a mondatba, a segédfunkció a mozgásparancs végrehajtásával párhuzamosan, vagy a mozgásparancs végrehajtása után kerül végrehajtásra.

A végrehajtás sorrendjét a szerszámgép építője dönti el, és a szerszámgép specifikációja tartalmazza.

B címen például osztóasztal indexelése valósítható meg.

13.3 Pufferürítő funkciók

A vezérlőben a mondatfeldolgozó **előre olvassa a mondatokat, feldolgozza, majd puffereli** azokat. A végrehajtó (az interpolátor és a PLC) a pufferből szedi elő a már feldolgozott mondatokat, és hajtja végre a mondatokban megadott mozgásokat és funkciókat.

Bizonyos esetekben szükség lehet a mondatok **előreolvasásának leállítására**. Erre az **NC és a PLC közötti tevékenység szinkronizálása miatt** lehet szükség.

Például, ha a PLC egy funkció végrehajtásához elkér egy, vagy több tengelyt az NC-től, hogy mozgassa azokat, fel kell függeszteni az előreolvasást. Az előreolvasást a funkció végrehajtása után lehet csak folytatni, miután a PLC visszaadta a tengelyeket az NC-nek. Ezután az NC a már a PLC által megváltoztatott tengelypozíciótól számolva tudja folytatni a megmunkálást.

Paraméteren ki lehet jelölni 10 db. egyedi M kódot, 8 db. M kód csoportot, valamint mindhárom segédfunkciót és az S és T kódokat pufferürítésre.

A pufferürítő funkciókat a gép építője határozza meg és állítja be.

A programvégrehajtás során például a szerszámsugár korrekció figyelembe vételéhez (G41, G42) szükség van a mondatok előreolvasására. Ha G41, G42 közben pufferürítő funkciót programozunk, a vezérlő felfüggeszti a szerszámsugár korrekciósámítást, ezért a kontúr sérülni fog.

14 Az alkatrészprogram szervezése

A bevezető részben már láttuk az alkatrészprogram felépítését és formátumát. Ebben a fejezetben az alkatrészprogramok szervezéséről lesz szó.

14.1 A mondatszám (N cím)

A program mondatait sorszámmal láthatjuk el. A mondatszámokat címkeként is kezelhetjük, amelyekre a program más részein hivatkozhatunk. A mondatok számozása

Nnnnnnnnn

utasítással történik. N címre legfeljebb 8 számjegyet írhatunk. N cím használata nem kötelező. Egyes mondatokat beszámozhatunk, másokat nem. A mondatok számozásának nem kötelező növekvő sorrendben követniük egymást.

14.2 Feltételes mondatkihagyás (/ cím)

Feltételes mondatkihagyást programozhatunk törtvonal

/n

utasítással. A törtvonal / *cím értéke n=1-8* lehet. Az 1-8 számok kapcsoló sorszámokat jelentenek. Az 1-es sorszámú feltételes mondat kapcsoló a vezérlő kezelőpanelén található.

A többi kapcsoló felszerelése opcionális, a gép építője határozza meg.

Abban az esetben, ha egy mondat elejére /n feltételes mondatkihagyást programozunk, akkor

- ha az n-edik kapcsoló **bekapcsolt állapotban van kihagyja** a végrehajtásból a mondatot,
- ha az n-edik kapcsoló **kikapcsolt állapotban van végrehajtja** a mondatot.

Ha csak / címet programozunk a mondat elején, az az 1-es kapcsolóra vonatkozik:

/ N1200 G0 X200 (kihagyja a mondatot ha az 1. kapcsoló be van kapcsolva)

Programozhatjuk így is:

/1 N1200 G0 X200 (kihagyja a mondatot ha az 1. kapcsoló be van kapcsolva)

Ha azt akarjuk, hogy a feltételes mondat kapcsolót akár a mondat végrehajtása előtti mondatban is figyelembe vegye a vezérlő, állítsuk az N1337 Execution Config paraméter **#4 CBB=0**-ba. Ekkor a feltételes mondat utasítás (/ jellel kezdődő mondatok) **elnyomja** a mondat előreolvasást. Ebben az esetben **G41, G42** esetén a kontúr **torzul**, viszont a feltételes mondat kapcsolót elég az **előző mondat végrehajtása közben kapcsolni**, hogy hatásos legyen.

Ha azt akarjuk, hogy a / utasítás ne nyomja el a mondat előreolvasást, állítsuk az N1337 Execution Config paraméter **#4 CBB=1**-be. Ekkor a feltételes mondat utasítás (/ jellel kezdődő mondatok) **nem nyomja el** a mondat előreolvasást. Ebben az esetben **G41, G42** esetén a kontúr **nem torzul**, viszont a feltételes mondat kapcsolót a biztos hatás miatt **a program végrehajtása előtt be kell állítani**.

Egyes kapcsolókat a PLC program is használhat a programfutás vezérlésére.

Például munkadarab adagolóval felszerelt gép esetén a főprogramot M99-cel végteleníthetjük:

```
... (alkatrészprogram)
M90 (munkadarab számláló léptetése)
/8 M30 (a 8-as kapcsolót a munkadarab számláló vezérli)
```

M99

Ha a munkadarab számláló elérte a kívánt darabszámot, a PLC a 8-as feltételes mondatkihagyás kapcsolót kikapcsolja, a program M30-ra fut, és megáll a végrehajtás.

A fenti példa csak az N1337 Execution Config paraméter #4 CBB=0 paraméterállása esetén működik helyesen, tehát, ha a / kapcsolók elnyomják a mondat előreolvasást.

Az N1337 Execution Config paraméter #4 CBB=1 paraméterállás esetén a program akkor működik helyesen, ha az M90 kód pufferürítésre van beállítva.

A paraméter megváltoztatása előtt kérdezze meg a gép építőjét a hatásokról!

14.3 Megjegyzések írása az alkatrészprogramba: (komment)

Ha a program egy részét **gömbölyű zárójelek ()** közé tesszük, a zárójelek közötti szakaszt a mondatfeldolgozó nem veszi figyelembe.

Így az alkatrészprogramokba megjegyzéseket (kommenteket) írhatunk.

Ha azt szeretnénk, hogy az alkatrészprogram egy részét a vezérlő ne hajtsa végre, de a programrészletet nem akarjuk a programból törölni, tegyük a kívánt részt gömbölyű zárójelek közé.

Például:

```
N10 G0 X100      (pozícionálás X100-ra)
N20 Z30
   (N30 G1 Z60 F0.3
N40 X300)
...

```

Az N10 mondatba kommentet írtunk. Az N30 N40 mondatot bezárójeleztük, ezért a vezérlő ezt a két mondatot nem veszi figyelembe.

14.4 Főprogram és alprogram

Kétféle programot különböztetünk meg: főprogramot és alprogramot. Makrónak nevezzük az olyan alprogramot, amelynek argumentumokat lehet átadni.

Egy alkatrész megmunkálása során adódhatnak ismétlődő tevékenységek, amelyeket ugyanazzal a programrészlettel lehet leírni. Annak érdekében, hogy az ismétlődő részeket ne kelljen többször leírni a programban, ezekből a részekből alprogramot készíthetünk, amelyet a főprogramból hívhatunk.

A fő- és alprogram felépítése teljes egészében megfelel a bevezetőben mondattaknak.

Kettejük között a különbség az, hogy míg a főprogram végrehajtása után a megmunkálás befejeződik, és a vezérlő arra vár, hogy újra elindítsák, az alprogram befejezése után a végrehajtás visszatér a hívó programba és onnan folytatja a megmunkálást.

Programozástechnikailag a különbség a kétféle program között, a program lezárásából adódik. A főprogram végét M02, vagy M30 kóddal jelezzük (használatuk nem kötelező), az alprogramot pedig M99 kóddal kell kötelezően lezárni.

14.4.1 Programok azonosítása a tárban. A programszám (O)

A tárban a programok a felhasználó által definiált, különböző nevű mappákban helyezkednek el. A mappákban a programokat a fájlnevük azonosítja. A vezérlő akkor tekint egy fájl alkatrészprogramnak, vagyis akkor lehet egy fájl alkatrészprogramként futtatni, ha a kiterjesztése (a “.” utáni rész):

fájlnev.txt

fájlnev.prg
 fájlnev.nct
 vagy
 fájlnev.nc

A programok fájlneve karakterekből és számokból állhat.
 Az alprogramokat külön fájlban tároljuk, azok nem lehetnek a főprogrammal egy fájlban.

A programszám

A programszám

Onnnn.ext

vagy

Onnnnnnnn.ext

egy speciális fájlnev, amely kötelezően O betűvel kezdődik és amelyet 4, vagy 8 decimális számjegy követ. A kiterjesztésükre (.ext) a fent elmondottak érvényesek.

Onnnn: **O betű, utána 4 számjegy** a vezető nullákkal együtt.

O1 érvénytelen fájlnev,
 O0001 érvényes fájlnev.

vagy

Onnnnnnnn: **O betű, utána 8 számjegy** a vezető nullákkal együtt.

O01234 érvénytelen fájlnev,
 O00001234 érvényes fájlnev.

14.4.2 Alprogram hívása (M98)

Alprogramokat kétféle módon hívhatunk: programszám, vagy fájlnev szerint.

Alprogram hívása programszám alapján

Az

M98 P....

utasítássor alprogramhívást generál. P címre a hívott program programszámát írjuk. P címen a vezető nullák elhagyhatók és a szám után kiterjesztést nem szabad írni. Az utasítás hatására a program végrehajtása a P címen meghatározott számú alprogramon folytatódik:

hívó program		alprogram	megjegyzés
O0010			az O0010 program végrehajtása
M98 P11	--->	O0011	az O0011 alprogram hívása az O0011 alprogram végrehajtása
következő mondat	<---	M99	visszatérés a hívó programba
.....			az O0010 program folytatása

A P címen, programszámmal hívott alprogramok esetén a mapparendszerben elfoglalt helyükre és a fájlnevre a következő megkötések érvényesek:

- Az alprogramoknak ugyanabban a mappában kell lenniük, mint az őket hívó programnak.
- A programszám alapján hívott alprogramok fájlnevére az előző fejezetben leírt megkötések

érvényesek.

- Az **alprogramok kiterjesztésének meg kell egyeznie az őket hívó program kiterjesztésével.**:

Például, ha a főprogram fájlneve:

Foprogram.**prg**

A “Foprogram.prg”-ből hívott alprogram kiterjesztésének is .prg-nek kell lennie:

O1234.**prg** esetén a hívás megtörténik,

O1234.**nct** esetén hibaüzenet keletkezik.

Alprogram hívása fájlnev alapján

Az

M98 <alprogram.nct>

utasításkor meghívja a programmal egy mappában lévő “alprogram.nct” nevű alprogramot.

A **fájlnevet** a < **kisebb** és a > **nagyobb** jel közé kell írni.

Ebben az esetben a főprogram és az alprogram kiterjesztésének nem kell megegyezniük.

A < és a > jel között megadhatjuk a fájl relatív elérési útvonalát is. Az elérési útvonalat mindig a hívó program mappájától kell megadni.

Ha a hívott alprogram az “alprogram.prg” az őt hívó program a “program1.nct” mappájától **egy szinttel lejjebb** van, egy “alprogramok” nevű almappában:

...

alprogramok (mappa)

program1.nct (fájl)

az alprogramhívás az

M98 <\alprogramok\alprogram.prg>

utasításra történik.

Ha a hívott alprogram az “alprogram.prg” az őt hívó program mappájától “foprogramok” **egy szinttel feljebb** van:

...

foprogramok (mappa)

alprogram.prg (fájl)

az alprogramhívás az

M98 <..\alprogram.prg>

utasításra történik.

A visszaléptetések és az előre léptetések több szintig mehetnek.

Például:

<..\..\mappa1\mappa2\mappa3\fájl.txt>

A < és a > **jelek közötti szöveg hossza** viszont **maximu 60 karakter** lehet

☞ **Megjegyzés:** az elérési útvonal megadásánál mindig a \ (backslash) karaktert kell használni, ami nem tévesztendő össze a / (per) karakterrel.

A mapparendszerben való visszalépéshez (..) mindig 2 db. “.” pontot kell használni.

Alprogramhívás ismétlési szám megadásával

Az

M98 P.... L....

vagy az

M98 <elérési út \ fájlnev> L....

utasítás sor az L címen megadott számban hívja egymás után a megadott alprogramot.

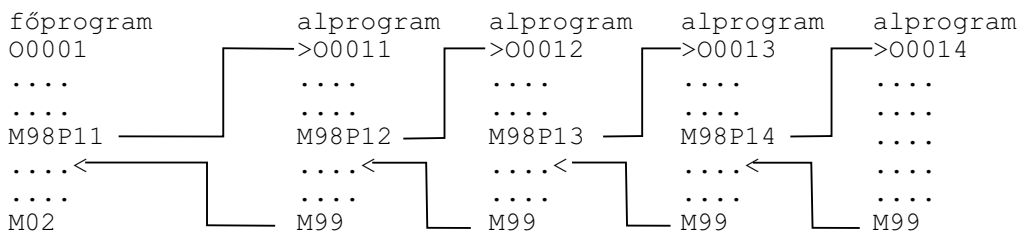
L címet maximum 8 decimális számjegyen adhatjuk meg.

Ha L-nek nem adunk értéket az alprogram egyszer hívódik meg, azaz L=1-et tételez fel a vezérlő.

Az

M98 P11 L6

utasítás azt jelenti, hogy hívd meg az O0011-es alprogramot egymás után 6-szor.

Többszintű alprogramhívásAlprogramból is lehet alprogramot hívni. Alprogramhívások (a makrohívásokkal együtt) **16 szint-ig** skatulyázhatók egymásba.**14.4.3 Visszatérés alprogramból (M99)**Visszatérés a hívást követő mondatra

Az

M99

utasítás alprogramban történő használata az alprogram végét jelenti, és a vezérlést visszaadja a hívó program hívást követő mondatára:

hívó program		alprogram	megjegyzés
O0010			az O0010 program végrehajtása
N101 M98 P11	--->	O0011	az O0011 alprogram hívása az O0011 alprogram végrehajtása
N102	<---	M99	visszatérés a hívó program következő mondatára
.....			az O0010 program folytatása

Visszatérés egy adott mondatra

Az

M99 P...

utasítás alprogramban történő használata az alprogram végét jelenti, és a vezérlést visszaadja a hívó program P cím alatt megadott számú mondatára. P címen maximum 8 decimális számjegyet lehet megadni.

hívó program		alprogram	megjegyzés
O0010			az O0010 program végrehajtása
N101 M98 P11	--->	O0011	az O0011 alprogram hívása
		az O0011 alprogram végrehajtása
N250	<---	M99 P250	visszatérés a hívó program N 2 5 0 mondatára
.....			az O0010 program folytatása

Visszatérés a ciklusszámláló átírásával

Az

M99 (P...) L...

utasítás átírja a hívó program ciklusszámlálóját. Ha L-re 0-t írunk az alprogram csak egyszer kerül meghívásra. Például ha az

M98 P11 L20

utasítással hívjuk meg az O0011 alprogramot és onnan

M99 L5

utasítással térünk vissza az O0011 alprogram összesen 6-szor hívódik meg.

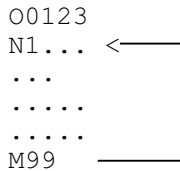
L címet maximum 8 decimális számjegyen adhatjuk meg.

14.4.4 Ugrás a főprogramon belül

Az

M99

utasítás főprogramban történő használata feltétel nélküli ugrást eredményez a főprogram első mondatára, és a programvégrehajtást innen folytatja. Az utasítás használata végtelen ciklust eredményez:



Az

M99 P.....

utasítás főprogramban történő használata feltétel nélküli ugrást eredményez a főprogram P cím alatt megadott számú mondatára, és a programvégrehajtást innen folytatja. Az utasítás használata végtelen ciklust eredményez:



A végtelen ciklusból vagy resettel vehetjük ki a programot, vagy, ha az M99 utasítást tartalmazó mondatot feltételes mondatkihagyással programozzuk,

/ M99

a feltételes mondatkihagyás kapcsoló állásától függően, vagy kihagyja az ugrást, vagy nem.

14.5 A csatornák közötti összevárás M funkciói

Többcsatornás működés esetén szükség lehet arra, hogy az egyik csatornában a program futása egy adott ponton megvárja, hogy egy, vagy több, másik csatornában futó program eljusson bizonyos műveletek végrehajtásáig. Ezt a csatornák közötti szinkronizálásnak nevezzük. Ezt a szinkronizálást az u.n. összevárós M kódokkal lehet megvalósítani.

Az összevárós M kódok **programszervező M kódok**, az **NC dolgozza fel** azokat, a PLC-nek nem kerülnek átadásra. **Pufferürítő M kódok**, azaz, amíg a szinkronizálás nem történt meg az összes csatornában, a mondatok előfeldolgozása szünetel és csak a szinkronizáció után folytatódik.

Az összevárós M kódokat paraméteren lehet kijelölni, maximum 100-at. Az N2201 Waiting M Codes Min paraméteren adhatjuk meg a csoport kezdőértékét és az N2202 Waiting M Codes Max paraméteren a végértékét.

Ha pl.

N2201 Waiting M Codes Min=500 és

N2202 Waiting M Codes Max=599

akkor az M500, M501, M502, ... , M599 M kódokat használhatjuk összevárásra.

Az

Mm Ppppppppp

utasítás összevárást programoz két, vagy több csatorna között. Az összevárást külön sorba kell programozni.

m: a paraméteren megadott egyik összevárós M kód,

pppppppp: azoknak a csatornáknak a száma, amelyek között össze kell várni. Mivel a rendszerben maximum 8 csatorna lehet, ezért P cím max. 8 számjegyű lehet.

Ha például az 1. és a 2. csatorna között kell összevárni, mindkét csatorna programjába, a megfelelő pontra írjunk

M501 P12

utasítást.

A PLC program pl. M funkcióval, vagy nyomógombbal a CP_NOWT jelző segítségével hatástalaníthatja az összevárást. Ez akkor lehet hasznos, ha csak az egyik csatornában akarjuk a programot futtatni és nem akarjuk az összevárós M funkciókat kikommentezni.

Példa:

Legyen az M kódok minimális értéke 500, a maximális értéke 599, és legyen 3 csatornánk:

1. csatorna programja

... megmunkálás

N60 M501 P12

vár a 2. csatornára

... megmunkálás

N130 M502 P123

... megmunkálás

2. csatorna programja

... megmunkálás

N100 M501 P12

...megmunkálás

M502 P123

vár az 1. és a 3. csatornára

... megmunkálás

3. csatorna programja

... megmunkálás

N110 M502 P123

vár az 1. és a 2. csatornára

... megmunkálás

Magyarázat:

Az 1. csatorna fut először az M501 kódra és vár addig, amíg a 2. csatorna is ráfut. A szinkronizálás után mindkét csatorna folytatja saját programját. Közben a 3. csatorna folyamatosan dolgozik.

Az M502 kódra legelőször a 3. csatorna fut rá majd a 2. és a legvégén az 1. A 3. csatornának be kell várnia, hogy az 1. és a 2. csatorna is ráfusszon kódra. Ezután a pont után indulhat mindhárom csatornában a megmunkálás.

15 A szerszámkorrekció

Ahhoz, hogy az alkatrész programban a különféle szerszámokhoz tartozó kinyúlásértékeket, szerszám sugarakat stb. ne kelljen a programban, a koordináták megadásánál figyelembe venni, a szerszámjellemzőket egy táblázatban, az ún. korrekciós táblázatban gyűjtjük össze. Valahányszor egy szerszámot lehívunk az alkatrész programban, meg kell adni, hogy az adott szerszám adatai hol találhatóak a korrekciós táblázatban. Ezek után a vezérlő már a hivatkozott korrekciók figyelembe vételével vezeti a szerszámot a programozott pályán.

15.1 A korrekciós tár. Hivatkozás szerszámkorrekcióra (T, vagy D)

Azt, hogy milyen címen hivatkozhatunk a szerszámkorrekcióra a szerszámcsere-élő mechanizmusa dönti el. Lásd "A T funkció" c. fejezetet.

A eset

Revolver típusú szerszámcsere-élők esetén az N1414 Comp. Config on Lathes paraméter #0 TCM=0 bitállásánál a **T kód a szerszámkorrekció kódját is tartalmazza** az alsó helyiértékű számjegyeken. A maradék felső helyiértékeken adható meg a szerszám száma.

Azt, hogy a szerszámkorrekciót hány számjegyen kell megadni, az N1413 No. of Digits of Offs. No. in T Code paraméter dönti el. A paraméter értéke 0, 1, 2, 3 lehet.

Tnnnnnnnk: korrekció 1 számjegyen

Tnnnnnnkk: korrekció 2 számjegyen

Tnnnnnnkkk: korrekció 3 számjegyen

ahol:

n...: a szerszám száma

k...: a korrekció száma

B eset

Ha a szerszámcsere M6 kódra történik, az N1414 Comp. Config on Lathes paraméter #0 TCM=1 bitállásánál a **T kód a szerszámkorrekció kódját nem tartalmazza**. Ekkor a korrekciós rekeszre, hossz- és sugárkorrekcióra együtt D címen hivatkozhatunk:

a szerszámhossz- és szerszám sugar-korrekciókra: **D** címen

A cím utáni szám, a korrekció száma mutatja meg, hogy melyik korrekciós érték kerül lehívásra. D cím értékhatára: 0-999. A vezető nullák elhagyhatók.

A korrekciós tár felosztása a csatornák között

Az N1400 No. of Tool Offsets paraméteren, **csatornánként** meg lehet adni, hogy az adott csatornában **hány szerszámkorrekciós csoport** legyen elérhető. Minden csoporthoz hossz- és sugárkorrekció is tartozik.

A teljes rendszerben az **esztergacsatornánkénti korrekciós csoportok összege** nem haladhatja meg a **999** csoportot. Minden csatornában a korrekciós csoportra való hivatkozás, akár programból T, D címen, akár a korrekciós táblázatot nézve történik, 1-től indul és a beállított paraméter értékig tart.

Az N1412 No. of Common Tool Offsets T paraméteren a **közös, minden eszterga csatornából** lehívható korrekciós csoportok számát adja meg. Maró korrekciókra eszterga csatornából nem lehet hivatkozni.

Minden csatornában az 1-től a No. of Common Tool Offsets T számú korrekciós csoportig terjedő

hivatkozás akár programból T, vagy D címen, akár korrekciós táblázatot nézve történik, a közös korrekciós értékekre mutat.

Példa:

Legyen 3 csatornánk. Az 1. csatornában legyen 30 korrekció (No. of Tool Offsets L1=30) a 2.-ban 40 és a 3.-ban 60. A közös korrekciók száma legyen 10 (No. of Common Tool Offsets T=10).

	1. csatorna	2. csatorna	3. csatorna
N001	N001-től N010-ig mindhárom csatorna a közös korrekciókat (T1-T10, D1-D10) olvassa		
...			
N010			
...	N011-től N030-ig a saját korrekcióit (T11-T30, D11-D30) olvassa	N011-től N040-ig a saját korrekcióit (T11-T40, D11-D40) olvassa	N011-től N060-ig a saját korrekcióit (T11-T60, D11-D60) olvassa
N030			
...			
N040			
...			
N060			

Az így felhasznált korrekciós csoportok száma: $10+(30-10)+(40-10)+(60-10)=110$ darab.

A korrekciós tár felosztása egy csatornán belül

A korrekciószámmal a szerszámkorrekciós táblázat egy csoportját jelöljük ki a vezérlő számára. Ennek a táblázatnak az elemei a következők:

sor- szám	X		Y		Z		R		Q
	geom.	kopás	geom.	kopás	geom.	kopás	geom.	kopás	
1	123.500	-0.234	87.450	-0.129	267.400	-0.036	1	-0.010	3
2									
3									
...									

A szerszámkorrekciós táblázat tartalmazza a szerszám X, Y és Z irányú kinyúlását, a szerszám sugarát (R), illetve a szerszámállítás kódját (Q).

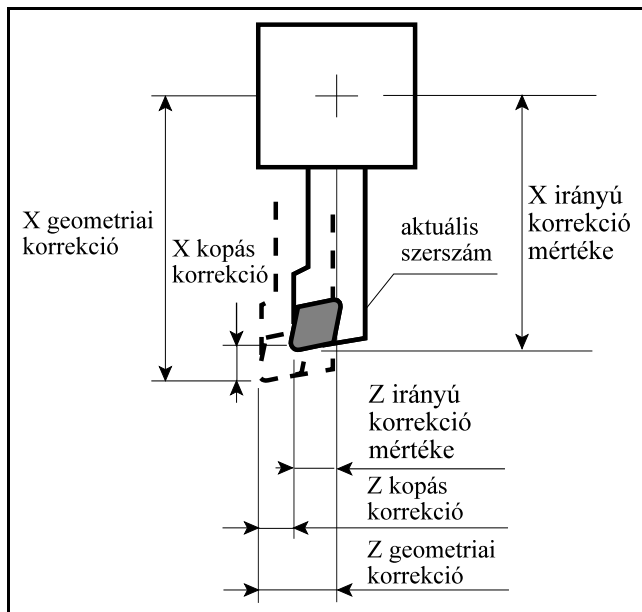
☛ Figyelem!

A 00 korrekciószám nem szerepel a táblázatban, az ezen lévő korrekciós értékek mindig nullák, tehát a T0 utasítás a korrekciók törlését jelenti.

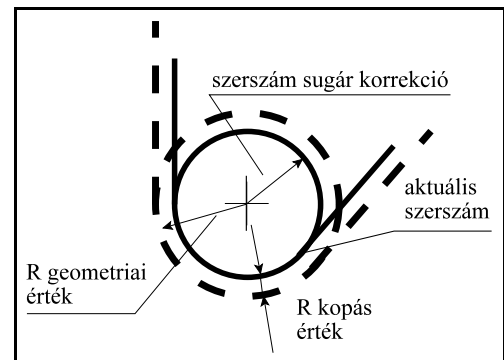
Az X, Y, Z irányú korrekció és a sugárkorrekció (R) két részből tevődik össze: a geometriai és kopásértékből.

Geometriai érték: a bemért szerszám hossza/sugara. Előjeles szám.

Kopás érték: a megmunkálás folyamán fellépő kopások mértéke. Előjeles szám. Kézi adatbevitel esetén a vezérlő a bevitt érték abszolút értékét az N1415 Max. Amount of Wear Comp. paraméteren megadott érték szerint korlátozza.



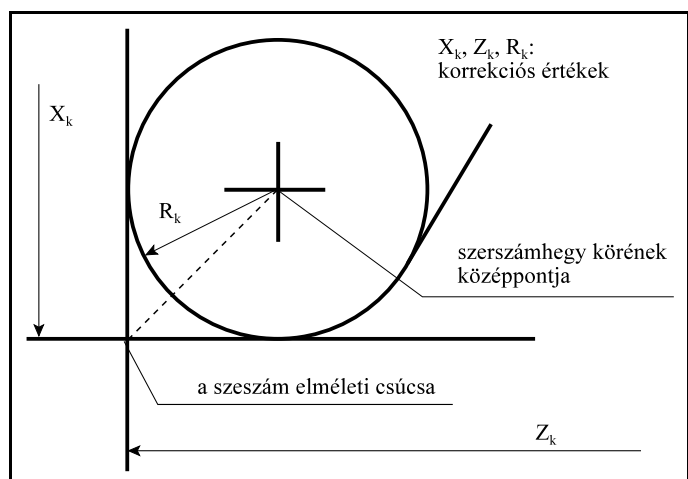
15.1-2 ábra



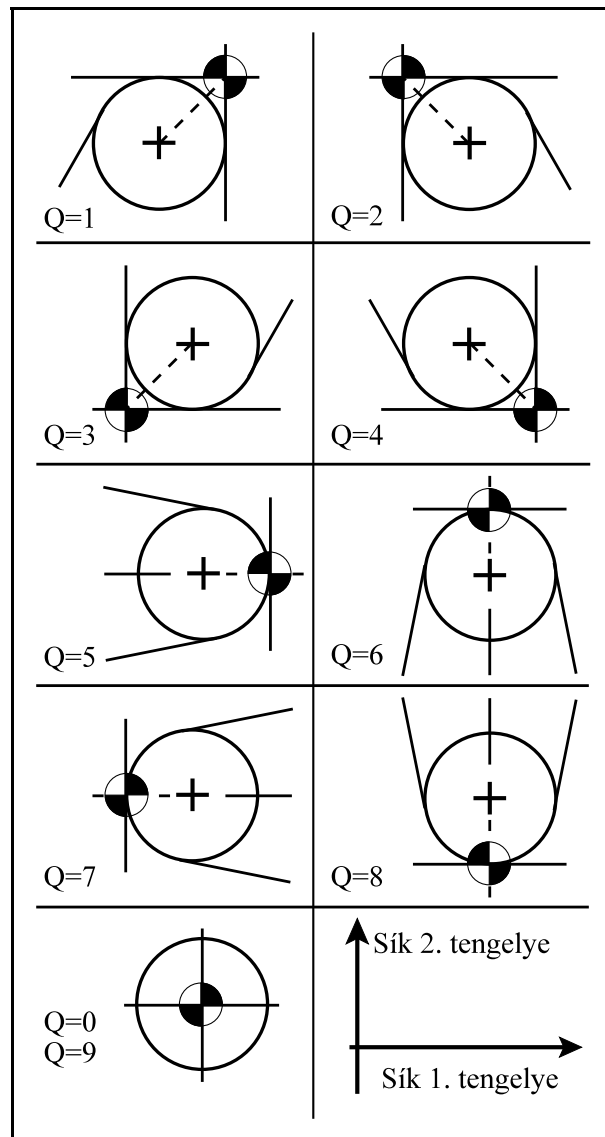
15.1-3 ábra

Szerszámállás: A *szerszámállás kódja (Q)* egyjegyű szám, értéke 0,1,...9 lehet.

A szerszám állás kódja azt mutatja meg, hogy a szerszámhegy körének középpontjából nézve a szerszám elméleti csúcsa milyen irányban található. Mindig a kiválasztott sík első és második tengelyét figyelembe véve kell megadni. Marószerszám használata esetén Q értéke mindig 0, vagy 9.



15.1-4 ábra



15.1-5 ábra

A lehívott **korrekciós kód öröklődik**, vagyis a vezérlő mindaddig ugyanazt a korrekciós értéket veszi figyelembe, amíg egy másik T, vagy D parancsot nem kap, azaz amikor egy T, vagy D paranccsal a korrekciós értéket kiolvastuk, úgy a korrekciós táblázat módosítása (például G10 programozásával) már nincs hatással a kiolvasott értékre.

A korrekciós tár korrekciós értékei kikapcsolásra megőrződnek.

A szerszámkorrekciós értékeket be lehet állítani, illetve módosítani a kezelőpanelről adatbevitellel és programból a G10 beállító utasítás használatával. Ha a G10 paranccsal módosítjuk az aktuális korrekciós értéket, akkor ismételten hivatkoznunk kell az aktuális korrekciós regiszterre, mert csak ebben az esetben kerül figyelembevételre a módosított érték.

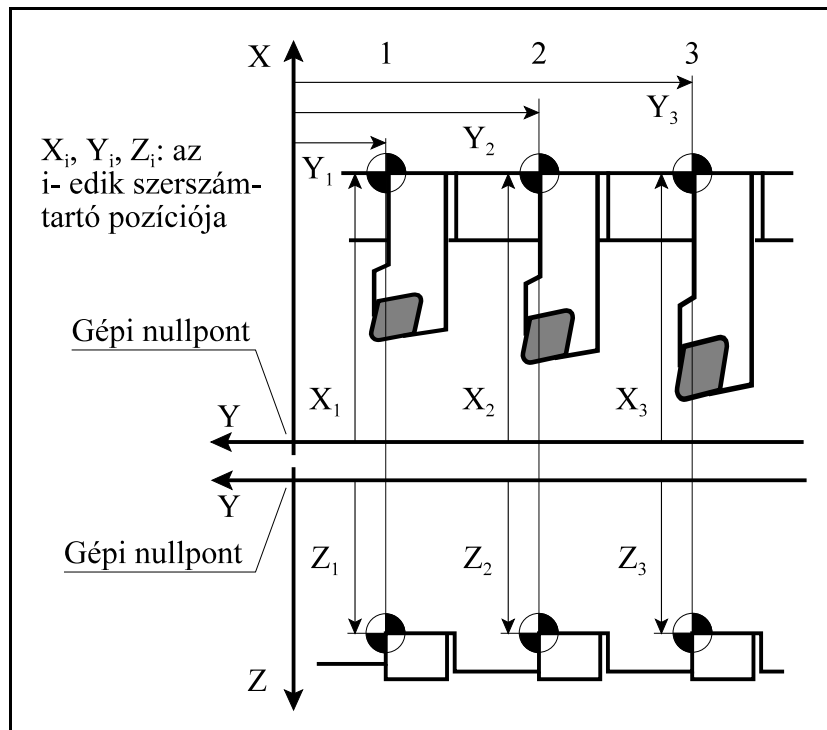
15.2 A 2. geometriai korrekciós tár

Kiterített szerszámozást használó esztergákon célszerű bevezetni egy második geometriai korrekciós tárat, amely a **hosszkorrekciókra vonatkozik**. A második geometriai korrekciós tár ugyanolyan méretű, mint az első, és a geometriai hosszkorrekciókra való hivatkozás során a második

geometriai korrekciós tárban tárolt értékeket, meghatározott feltételek teljesülése esetén, hozzáadja az első tárban tárolt értékekhez.

A második geometriai korrekciós tárban a szerszám-tartók gépi koordinátarendszerben elfoglalt helyzetét X, Y, Z -t lehet megadni. Ezzel lehetővé válik, hogy az első geometriai korrekciós tárban a szerszám valódi x, y, z hosszát adjuk meg, azaz az első geometriai tárban a külső szerszámbe-mérőn mért kinyúlási értékeket közvetlen meg lehessen adni. A figyelembe vett korrekció értéke tehát:

korrekció = 1. geometriai korrekció + 2. geometriai korrekció + kopás korrekció



15.2-1 ábra

A második geometriai korrekciós tár elérhetővé tételét az N1414 Comp. Config on Lathes paraméter #4 SGC=1 bitállása engedélyezi.

Ezen túlmenően az adott gép **PLC programja engedélyezi**, vagy **tiltja tengelyenként a 2. geometriai korrekciók értékének figyelembe vételét és a figyelembe vétel előjelét.**

2. geometriai korrekciós tár bemérését és kezelését a gép építője végzi.

15.3 Szerszámkorrekciós értékek módosítása programból (G10)

A

G10 L P X Y Z R Q

utasítással lehet a szerszámkorrekciók értékeit módosítani programból. A G10 utasítás egyelővetű.

A címek és értékek jelentése:

L=10: geometriai érték beírása

L=11: kopás érték beírása

P címen adjuk meg, hogy melyik korrekciós csoportot akarjuk módosítani:

P: a korrekciós csoport száma

X, Y, Z, R címen visszük be a hossz- és sugárkorrekció értékét:

X: hosszkorrekció értéke az X tengelyen

Y: hosszkorrekció értéke az Y tengelyen

Z: hosszkorrekció értéke a Z tengelyen

R: a sugárkorrekció értéke

G90 abszolút adatmegadás parancsállapotban az X, Y, Z, R címre írt érték kerül a megfelelő korrekciós regiszterbe.

G91 inkrementális adatmegadás parancsállapotban, vagy I operátor használata esetén a címre írt adat hozzáadódik a megfelelő korrekciós regiszter tartalmához.

Q: a szerszámállítás kódja (0...9)

A kódot L10 és L11 megadásával is elfogadja.

A G10 utasítással a 2. geometriai korrekciós tár adatait nem lehet módosítani.

15.4 A szerszámhossz-korrekció figyelembe vétele T kódra

Az N1414 Comp. Config on Lathes paraméter #0 TCM=0 bitállásánál a T kód a szerszámkorrekció kódját is tartalmazza az alsó helyiértékű számjegyeken.

Az N1413 No. of Digits of Offs. No. in T Code paraméter dönti el, hogy a T kódban a korrekció hány számjegyet foglal el.

A hivatkozás

Tnnnnnnnk

Tnnnnnnkk

Tnnnnnnkkk

lehet, ahol:

n...: a szerszám száma

k...: a korrekció száma

A korrekció száma a geometriai- és kopáskorrekciót együtt hívja

Ha az N1414 Comp. Config on Lathes paraméter paraméter #1 GTN=0 bitállásánál a **T címen hivatkozott korrekció száma a geometriai és a kopás korrekciót együtt** hívja le.

A Tnnkk hivatkozás esetén:

– az nn értéket a vezérlő átadja a PLC-nek,

– az **kk geometriai** és a **kk kopásregiszter** értékét összeadja és ez lesz a lehívott korrekció értéke:

Korrekció = Geometriai_{kk} + Kopás_{kk}

A

Tnn00

utasítás törli a korrekciót

A szerszámszám a geometriai, a korrekciószám a kopás korrekciót hívja

Ha az N1414 Comp. Config on Lathes paraméter paraméter #1 GTN=1 bitállásánál a **T címen hivatkozott szerszám száma a geometriai, a korrekció száma a kopás korrekciót** hívja le.

A Tnnkk hivatkozás esetén:

- az nn értéket a vezérlő átadja a PLC-nek,
- az **nn geometriai** és a **kk kopásregiszter** értékét összeadja és ez lesz a lehívott korrekció értéke:

$$\text{Korrekció} = \text{Geometriai}_{nn} + \text{Kopás}_{kk}$$

A fenti megoldás két problémát vet fel. Melyik regiszterből nn-ből, vagy kk-ből vegye a szerszámállás kódját, azaz **Q értékét**, illetve a **Tnn00** hivatkozást hogyan értelmezze a vezérlő.

Ha az N1414 Comp. Config on Lathes paraméter #2 QGW bitje:

- =0: a **szerszámállás kódját (Q)** a **kopás kód regiszteréből** veszi,
- =1: a **szerszámállás kódját (Q)** a **geometriai kód regiszteréből** veszi.

Ha az N1414 Comp. Config on Lathes paraméter #3 ZCG bitje:

- =0: a **0-ás korrekciószám (Tnn00)** nem törli a geometriai, csak a **kopás korrekciót**,
- =1: a **0-ás korrekciószám (Tnn00)** a **geometriai és a kopás korrekciót** is törli.

A T kód hatására a koordinátarendszer eltolódik a T kódban megadott korrekciós csoporthoz tartozó X_k , Y_k , Z_k korrekciós értékkel. Ez azt jelenti, hogy a pillanatnyi X, Y, Z pozícióból levonásra kerül a kiválasztott korrekciós csoporthoz tartozó X_k , Y_k , Z_k irányú korrekciós érték. Ettől kezdve nem a szerszám tartó vonatkoztatási pontjának, hanem a szerszám elméleti csúcsának koordinátáit látjuk a kijelzőn. Korrekciós értéken a továbbiakban mindig a geometriai-, és kopáskorrekció összegét értjük.

Például a

```
T0
N10 G0 G90 X700 Z350
N20 T202
N30 X300 Z150
```

utasításcsoport hatására az N10 mondatban a szerszám tartó vonatkoztatási pontját az

X700; Z350

koordinátájú pontra vezeti. Az N20 mondatban beváltja a szerszámot és levonja a pozícióból a 2-es korrekciós csoportba tartozó $X_k=340$, $Z_k=30$ korrekciós értékeket. Ekkor a pozíciókijelző az

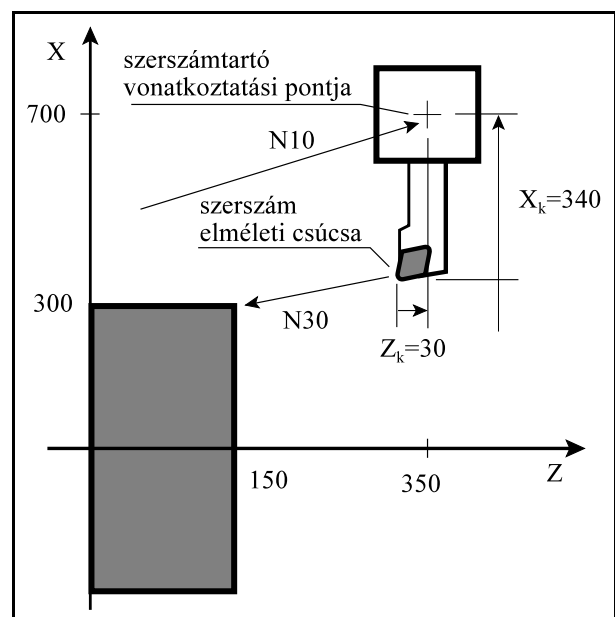
$X=700-340=360$; $Z=350-30=320$

értékre vált. *Az N20 mondatban semmilyen mozgás nem történik.* Az N30 mondatban már a szerszám elméleti csúcsát vezeti a programozott X300; Z150 pontra, vagyis az elmozdulás

$X=300-360=-60$; $Z=150-320=-170$

lesz.

Ha a **T kódot mozgásmondattal együtt** adjuk meg, a mondatvégi pozícióra már a T kóddal meghatározott új korrekciós érték figyelembe vételével áll rá. A szerszámcsere viszont, ha a T kód szerszámváltás parancsot is tartalmaz, a mozgással párhuzamosan, vagy a mozgás végén



15.4-1 ábra

hajtódik végre. (A szerszámcsere időpontját a gép építője határozza meg.)

Nézzük a fenti példát:

```
T0
N10 G0 G90 X700 Z350
N20 X300 Z150 T202
```

Most az N30 mondatot elhagytuk és az N20 és N30 mondat parancsait összevontuk. Az N20 mondatban a mozgás során már a szerszám elméleti csúcsát vezeti az

$X=300, Z=150$

koordinátájú pontra, mint az előző példában az N30 mondat esetén.

Ha a **T kódon csak korrekciószámot** programozunk

```
N10 G0 G90 X700 Z350
N20 X300 Z150 T2
```

szerszámváltás nem történik, csak a 2-es korrekciót hívtuk le az N20 mondatban.

A hosszkorrekció törlése

A 0-ás számú korrekciós csoportnak kitüntetett szerepe van., és a korrekció törlését jelenti.

A

Tnn00, vagy a

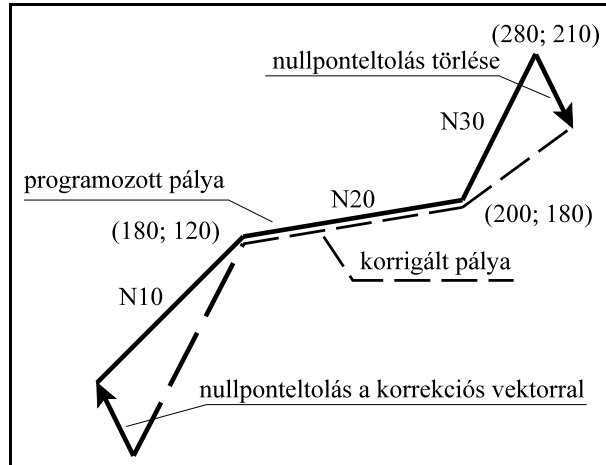
T0

utasítás hatására törlődik a hosszkorrekció (nn: tetszőleges szerszámszám). A folyamat pont fordított, mint a korrekció lehívása. Ez azt jelenti, hogy a szerszám elméleti csúcsának X, Y, Z koordinátájához hozzáadódnak az előzőleg érvényben volt X_k, Y_k, Z_k korrekciós értékek, és a továbbiakban a szerszám tartó vonatkoztatási pontjának koordinátáit látjuk a kijelzőn. Ha a korrekció törlése mozgásmondatdal együtt történik a mondat programozott végpontjára a szerszám tartó vonatkoztatási pontját küldi.

Például a

```
N10 X180 Z120 T202
N20 X200 Z180
N30 X280 Z210 T200
```

mintapéldában az N10 mondatban a lehívott korrekcióval korrigálja a mozgást a programozotthoz képest, az N30 mondatban pedig törli a korrekciót, vagyis az N10 mondatban beállított nullponteltolást.



15.4-2 ábra

15.5 A szerszámhossz-korrekció figyelembe vétele G kódra (G43.7, G49)

Az N1414 Comp. Config on Lathes paraméter #0 TCM=1 bitállásánál a hosszkorrekció hívása G kódra történik.

A

G43.7 D

utasítás bekapcsolja a szerszámhossz-korrekciót. Ha tengelycímmel együtt programozzuk, a megadott pozícióra már az új korrekcióval áll rá.

D cím jelentése: a szerszámhossz-korrekció értékét az ezen a címen megadott korrekciós csoportból veszi.

Abszolút adatmegadás esetén a

G43.7 G0 G90 X100 Z10 D5

utasítás a szerszám hegyét a D5 korrekció figyelembe vételével az X100 Z50 koordinátájú pontra mozgatja.

Inkrementális adatmegadás esetén a

G43.7 G0 G91 X20 Z10 D1

utasítás *eltolja a kezdőpontot a korrekciós értékkel*, majd *innen, a korrigált kezdőponttól számítódik az inkrementális elmozdulás*.

Értelemszerűen a

G43.7 G91 X0 Y0 Z0 D2

utasítás hatására nem végez az X, Y és a Z tengely mozgást, csak a pozíció veszi fel az új korrekciónak megfelelő értéket.

G43.7 hatása *öröklődik* egészen addig, amíg ebből a csoportból egy másik parancsot nem kap. Ha a korrekciós értéket *új D cím* lehívásával megváltoztatjuk, a régi érték törlődik, és az új érték kerül érvényesítésre:

G43.7 X120 Z100 D1 (az X150 Z=110 gépi pozícióra mozog)

...

X120 Z100 D2 (az X170 Z=120 gépi pozícióra mozog)

Az X120 Z100 D2 mondat megörökölte a G43.7 állapotot, ezért az új, D2 korrekciót veszi figyelembe.

D0 törli a korrekciót. A fenti példát folytatva:

G43.7 X120 Z100 D2 (az X170 Z=120 gépi pozícióra mozog)

...

X120 Z100 D0 (az X120 Z100 gépi pozícióra mozog)

A

G49

parancs az összes tengelyen kikapcsolja a szerszámhossz-korrekciót, mozgással, ha a mondatba tengelycímet is programoztunk, vagy transzformációval, ha a mondatba nincs mozgás programozva.

Bekapcsolásra, resetre, vagy program vége után az N1300 DefaultG1 paraméter #5 G43 és #6 G44 bitjei döntenek el, hogy G43.7 kód van érvényben. Ha mindkét bit 0, akkor G49.

15.6 Maró korrekciók hívása (G43, G44)

Minden csatornához (vezérlőhöz), függetlenül attól, hogy az adott csatornát (vezérlőt) eszterga, vagy maró változatra konfiguráltuk az N0001 Control Type paraméteren, tartozik egy maró és egy eszterga korrekciós táblázat.

Eszterga változatra konfigurált vezérlőkön felmerülhet az igény, hogy **marási feladatokhoz maró korrekciós táblázatot** használjunk.

Annak, hogy eszterga csatornában (vezérlőben) **maró korrekciót** is lehessen használni, **két feltétel van:**

1. Az N1414 Comp. Config on Lathes paraméter #0 TCM bitje legyen 1, vagyis a **G kódra történő korrekcióhívás legyen bekapcsolva,**
2. Az N1414 Comp. Config on Lathes paraméter #6 TOM bitje legyen 1. Ennek a bitnek az értelmezése:
 =0: a csatorna (vezérlő) csak eszterga korrekciós táblát használ,
 =1: a csatorna (vezérlő) **maró korrekciós táblát is használ.**

A maró korrekciós tár felosztása

Minden korrekciós csoporthoz 7 érték tartozik:

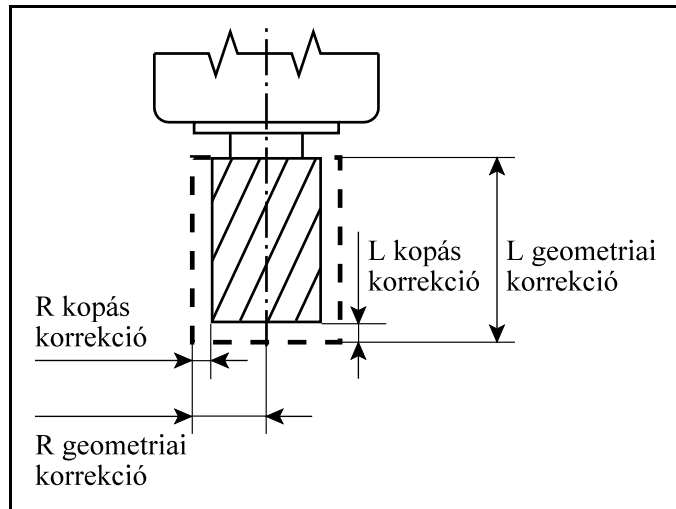
korrekciós szám	Hosszkorrekció L:		Sugárkorrekció R és lekerekítés r:				Szerszámállás: Q
	L geometriai	L kopás	R geometriai	R kopás	r geometriai	r kopás	
1	350.000	-0.130	5.000	-0.012	5.000	-0.010	0
2	830.000	-0.102	10.000	-0.008	6.000	-0.006	0
3	400.000	.	30.000	-0.160	0.000	0.000	0
.
.

A szerszámkorrekciós táblázat tartalmazza a szerszám hosszát (L), a szerszám sugarát (R), a szerszám lekerekítési sugarát illetve a szerszámállás kódját (Q).

A **0 korrekciós szám** nem szerepel a táblázatban, az ezen lévő **korrekciós értékek mindig nullák.**

Geometriai érték: a bemért szerszám hossza/sugara. Előjeles szám.

Kopás érték: a megmunkálás folyamán fellépő kopások mértéke. Előjeles szám. Kézi adatbevitel esetén a vezérlő a bevitt érték abszolút értékét az N1415 Max. Amount of Wear Comp. paraméteren megadott érték szerint korlátozza.



15.6-1 ábra

A lekerekítési sugár: r

Abban az esetben, ha

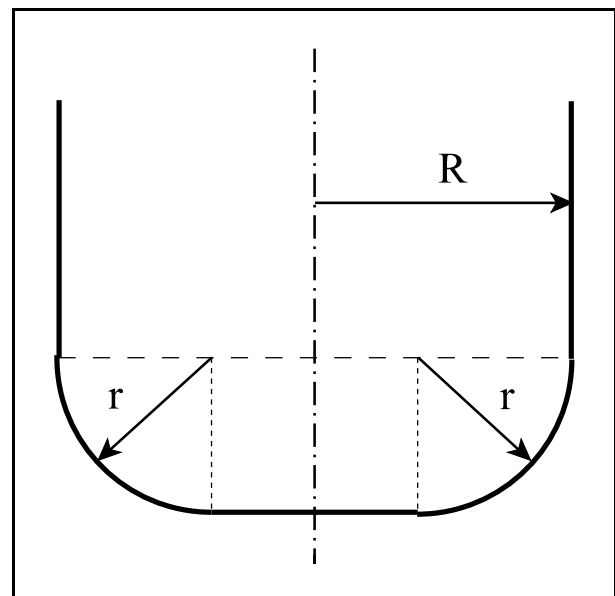
$r=0$: a szerszám vége szögletes, és a sugara R .

Ha

$r < R$: a szerszám r sugárral le van kerekítve, és a sugara R .

Ha

$r=R$: a szerszám R sugarú, gömbvégű újjmáró.



15.6-2 ábra

☞ **Figyelem:** A korrekcióra mindig **D címen** hivatkozunk, akár hossz-, akár sugárkorrekciót akarunk hívni.

A

G43 Dd q

vagy a

G44 Dd q

utasítás eszterga csatornában

átkapcsol a maró korrekciós táblára és

lehívja a **D címen megadott hosszkorrekciós rekesz** geometriai és kopás értékének összegét a **q címen megadott tengelyre**.

G43 utasítás a hosszkorrekció figyelembe vételére kijelölt tengely mentén *pozitív irányban* tolja el a programozói koordinátarendszert a D címen megadott korrekciós értékkel:

G43: + korrekció

G44 utasítás a hosszkorrekció figyelembe vételére kijelölt tengely mentén *negatív irányban* tolja el a programozói koordinátarendszert a D címen megadott korrekciós értékkel:

G44: – korrekció

A G43, G44 a 8-as G kód csoport tagja.

Azt, hogy a D címen megadott *hosszkorrekciót* a vezérlő *melyik tengelyen* vegye figyelembe, az **N1416 Comp. Config on Mills paraméter** #0 ZAX, #1 PLN bitjei alapján háromféle módon lehet megadni:

PLN	ZAX	A hosszkorrekció figyelembe vétele G43, G44 mondatban
0	0	a mondatban megadott összes tengelyen (alapeset)
0	1	mindig a Z tengelyen
1	0	G17 esetén: Z tengelyen G18 esetén: Y tengelyen G19 esetén: X tengelyen

A *korrekció törlését* a

G49

utasítás végzi.

A G43.7 utasítás visszakapcsol az eszterga korrekciós táblázat használatára.

Sugárkorrekció hívása: G41, vagy G42 kód mellett már D címet nem kell megadni, a hosszkorrekcióra lehívott D korrekciós rekeszből veszi elő a szerszámsugár értékét.

Maró korrekciós tár írása G10 utasítással:

A

G10 L P R Q

utasítással lehet a szerszámkorrekciók értékeit módosítani programból. G10 utasítás egylovétű. A címek és értékeik jelentése:

L címen adjuk meg, hogy milyen korrekciós értéket kívánunk módosítani:

L=10 jelentése: a beállítás a hosszkorrekció geometriai értékére vonatkozik,

L=11 jelentése: a beállítás a hosszkorrekció kopásértékére vonatkozik,

L=12 jelentése: a beállítás a sugárkorrekció geometriai értékére vonatkozik,

L=13 jelentése: a beállítás a sugárkorrekció kopásértékére vonatkozik,

L=110 jelentése: a szerszám lekerekítési sugarának, r-nek (D kód) geometriai értékére vonatkozik,

L=111 jelentése: a szerszám lekerekítési sugarának, r-nek (D kód) kopásértékére vonatkozik.

P címen adjuk meg, hogy melyik korrekciós csoportot akarjuk módosítani:

P: a korrekciós csoport száma

R címen adjuk meg a korrekció értékét.

R: a korrekció értéke

Q: a szerszámállás kódja (0...9)

Maró korrekciós tár írása és olvasása makrováltozók használatával

G43, vagy G44 állapotban a maró korrekciós tárat az [294.](#) oldalon leírtak szerint lehet írni, vagy olvasni.

Az aktuális hosszkorrekció értékének kiolvasása

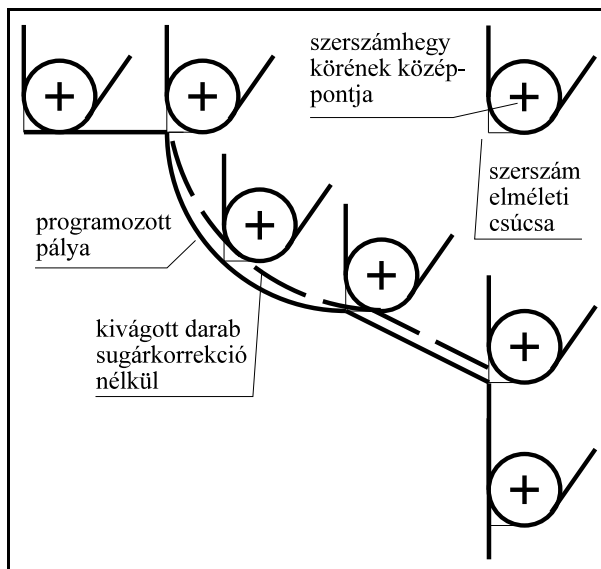
G43, vagy G44 állapotban az aktuális hosszkorrekció értékét makrováltozókon keresztül az [293.](#) oldalon leírtak szerint lehet kiolvasni.

Mintapélda:

...	(esztergálás)
...	
G49 D0	(Eszterga korrekciók törlése)
G10.9 X0	(X programozása sugárban)
G43	(Maró korrekciók használata)
#10001=200.1	(L geometriai)
#11001=-0.1	(L kopás)
#12001=20.3	(R geometriai)
#13001=-0.3	(R kopás)
G54 G17 G94 G0 X-50 Y-50	
Z5 D1	(D1 hosszkorrekció Z-re)
M3 S1500	
G1 Z-5 F1000	
G41 G1 X0 Y0	(sugárkorrekció be D1-gyel)
Y60	
X60	
Y0	
X0	
G40 G0 X-50 Y-50	
G49 D0 Z100	(maró korrekciók ki)
G10.9 X1	(X programozása átmérőben)
G43.7	(eszterga korrekciók használata)
...	(esztergálás)

15.7 A szerszámsugár korrekció (G40, G41, G42)

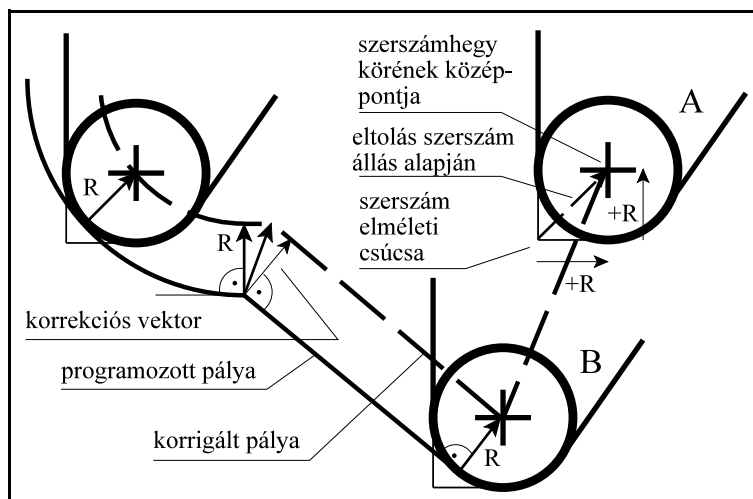
Abban az esetben, ha csak szerszámhossz korrekciót használunk, kúpos felületet, vagy körívet pontosan esztergálni nem tudunk. A szerszámhossz korrekció használata esetén a vezérlő a szerszám elméleti csúcsát vezeti a programozott pályán. Mivel minden szerszám csúcsa kisebb nagyobb lekerekítéssel rendelkezik, ezért a forgácsolt felület csak a tengelyekkel párhuzamos mozgásoknál lesz pontos, amint az a mellékelt ábrán látható. Kúpos, vagy íves felületeknél a darab átmérője minden ponton nagyobb lesz a programozottnál. Az eltérést az ábrán a vonalkázott terület mutatja.



Ahhoz, hogy egy tetszőleges alakzatot pontosan lehessen esztergálni, és az alakzatnak a rajz szerinti pontjait kelljen a programban megadni, függetlenül az alkalmazott szerszámhegy körének sugarától, a vezérlőnek a **szerszámhegy körének középpontját a programozott kontúrral párhuzamosan, attól sugárnyi távolságra kell vezetnie**. A vezérlő a **T**, vagy a **D** címen lehívott korrekciószámot bejegyzett **szerszámsugár korrekció** értékének függvényében állapítja meg, hogy a szerszámközpont pályáját milyen távolságra vezesse a programozott kontúrtól.

15.7-3 ábra

A kontúrra való ráállás előtt a szerszám elméleti csúcsának pozícióját tartja nyilván a vezérlő. Annak eldöntéséhez, hogy az elméleti csúcshoz képest milyen irányban található a szerszámhegy körének középpontja, szükség van a **szerszám állás kódjára**, amit a korrekciós táblázatban Q címen adtunk meg. A szerszámállás kódja alapján az elméleti csúcshoz X és Z irányban hozzáad, vagy kivon egy szerszámsugárnyi értéket (R). (Ábrán az "A" helyzet.) Ezután a **szerszámhegy körének középpontját** vezeti rá a programozott pálya kezdőpontjára merőlegesen állított R hosszúságú vektor, a **korrekciós vektor**, végpontjára. (Ábrán a "B" helyzet.) Ezután a programozott pályával párhuzamosan, attól R távolságra vezeti a szerszámhegy körének középpontját.



15.7-4 ábra

A **korrekciós vektor** egy olyan **síkbeli vektor**, amit a vezérlő minden mondatban újraszámol, és a programozott elmozdulásokat a mondat eleji és végi korrekciós vektorokkal módosítja. A ki-

adódó korrekciós vektorok hossza és iránya a T címen lehvívott korrekciós értéktől és a két mondat közti átmenet geometriájától függ.

A korrekciós vektorokat a **G17, G18, G19 utasítások által kiválasztott síkban** számolja. Ez a szerszámugár korrekció síkja. Ezen síkon kívüli mozgásokat a sugárkorrekció nem befolyásolja. Például: ha G18 állapotban X, Z sík van kiválasztva, akkor a korrekciós vektorok az X, Z síkban kerülnek kiszámításra.

Szerszámugár korrekció számítása közben a korrekciós sík váltása nem megengedett. Ha mégis megkíséreljük, akkor hibaüzenetet ad a vezérlő.

Abban az esetben, ha nem a fősíkbba eső tengelyek mentén akarunk korrekciós síkot definiálni a melléktengelyeket a paramétermezőben párhuzamos tengelyekként kell definiálni. Például, ha U párhuzamos tengelynek van felvéve, és a Z, U, síkban akarjuk a szerszámugár korrekciót alkalmazni G18 U__Z__ megadással lehet a síkot kijelölni.

A szerszámugár korrekció számítását programból ki, és be lehet kapcsolni:

G40: szerszámugár korrekciós számítás **kikapcsolása**

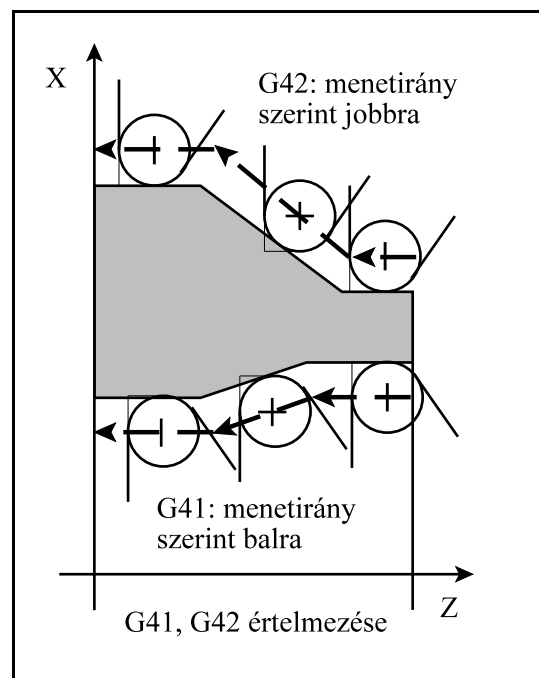
G41: szerszámugár korrekciós számítás **menetirány szerint balra**

G42: szerszámugár korrekciós számítás **menetirány szerint jobbra**

G40, G41, G42 parancsok **öröklődnek**. Bekapcsolás után, program végén, vagy reset hatására a vezérlő a G40 állapotot veszi fel.

A G41, vagy G42 parancs a korrekciós számítást bekapcsolja. G41 állapotban a szerszám a programozott kontúrt a menetirány szerint balról, G42 állapotban pedig jobbról követi. Az alkalmazott szerszámugár **korrekciós értéket T, vagy D címen** kell megadni. **A korrekciós számítás a G00, G01, G02, G03 interpolációs mozgásokra történik.**

Az eddig elmondottak pozitív szerszámugár korrekció megadásakor érvényesek. A **szerszámugár korrekció értéke** viszont **negatív is lehet**. Ennek gyakorlati értelme akkor van, ha például ugyanazzal az alprogrammal akarunk egy anya, majd egy ehhez illeszkedő apa munkadarabot körbejárni. Ezt úgy is meg lehet oldani, hogy G41-gyel forgácsoljuk például az anyát, és G42-vel az apát. Nem kell ezt a váltást azonban beszerkeszteni a programba, ha az anyadarabot például pozitív, az apadarabot pedig negatív sugárkorrekcióval munkáljuk meg. Ekkor a szerszámközpont pályája a programozott G41, vagy G42-vel ellentétesre vált:



15.7-5 ábra

	sugárkorrekció: pozitív	sugárkorrekció: negatív
G41	balról	jobbról
G42	jobbról	balról

☞ **Megjegyzés:** A további leírásokban és ábrákban az egyszerűség kedvéért mindig pozitív sugárkorrekcióval dolgozunk.

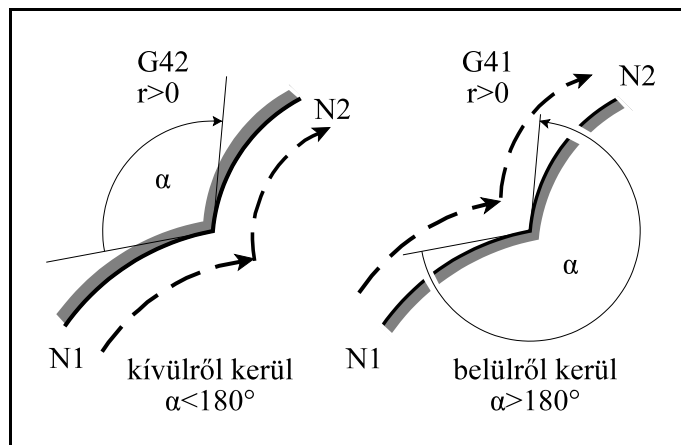
G40 vagy **Tnn00**, **T0**, **D0** parancs kikapcsolja a korrekciószámítást. A két parancs közti különbség, hogy a T0 utasítás csak 0 hosszúságú korrekciós vektorral számol és a G41 vagy G42 állapotot változatlanul hagyja. Ha ezek után új Tnnmm (mm≠0), D címre történik hivatkozás a G41 vagy a G42 állapot függvényében az új szerszámugárral kerül kiszámításra a korrekciós vektor.

Ha viszont G40 utasítást használunk, a korrekciós vektorokat nem számolja tovább a vezérlő. A sugárkorrekció bekapcsolásának, illetve kikapcsolásának, meghatározott szabályai vannak, amit a következő fejezetek tárgyalnak részletesen.

A sugárkorrekciós utasításokat csak automata, vagy kézi adatbeviteli üzemmódban hajtja végre a vezérlő. Kézi üzemmódban egyedi mondatokon nem határos. Ennek oka a következő. Ahhoz, hogy egy mondat végpontjában a korrekciós vektort ki tudja számítani a vezérlő a következő, a kiválasztott síba eső mozgást tartalmazó mondatot is be kell olvasnia. A két mondat közötti átmenet függvénye a korrekciós vektor. Ebből látható, hogy a korrekciós vektor számításához több mondat előfeldolgozására van szükség.

Mielőtt a korrekciószámítás részleteinek tárgyalásába fognánk bele, be kell vezetni egy segédadatot. Két szakasz, azaz két mondat metszéspontjában a két görbéhez húzott érintők által bezárt szöget: α -t. α értelmezése attól függ, hogy a kontúrt balról, vagy jobbról járjuk körül.

A vezérlő az α szög függvényében választja ki a metszéspontoknál a fordulási stratégiát. Ha $\alpha > 180^\circ$, azaz belül dolgozik a szerszám, a két szakasz között metszéspontot számít. Ha $\alpha < 180^\circ$, azaz a szerszám kívülről kerül, akkor további e-



15.7-6 ábra

gyenes szakaszokat iktathat be a kerüléshez.

A továbbiakban ez a leírás a sugárkorrekció számítás eseteit elsősorban az esztergálási feladatok szempontjából tárgyalja. Marási feladatok szempontjából a sugárkorrekció számítás eseteit az NCT2xxM Marógép és megmunkáló központ vezérlő Programozási leírás című gépkönyv tartalmazza.

15.7.1 A sugárkorrekció számítás bekapcsolása. Ráállítás a kontúrra.

G40 állapotból G41, vagy G42 utasítás hatására a vezérlő belép a sugárkorrekció számítási üzemmódba. A korrekció értékét a T címen megadott korrekciós rekeszből veszi.

A **G41** vagy **G42** állapotot *csak pozícionáló G0, vagy egyenes interpolációt végző G1* mondatban veszi fel.

Ha körmondatban (G2, G3) akarjuk a korrekciószámítást bekapcsolni a vezérlő hibajelzést ad. A *kontúrra való ráállítás stratégiáját* csak akkor választja a vezérlő, ha **G40 állapotból G41, vagy G42 állapotba** kapcsolunk.

A korrekció bekapcsolásának alapesetei α szög és a lehetséges átmenetek: egyenes–egyenes, egyenes–kör függvényében alább láthatók. Az ábrák G42 esetre vannak felrajzolva, **pozitív sugárkorrekciót** tételezve föl.

☞ *Megjegyzés:* Az ábrák jelöléseinek jelentése most, és a továbbiakban:

r: a sugárkorrekció értéke,

L: egyenes szakasz,

C: körív,

S: mondatonkénti üzemmódban a megállás helye,

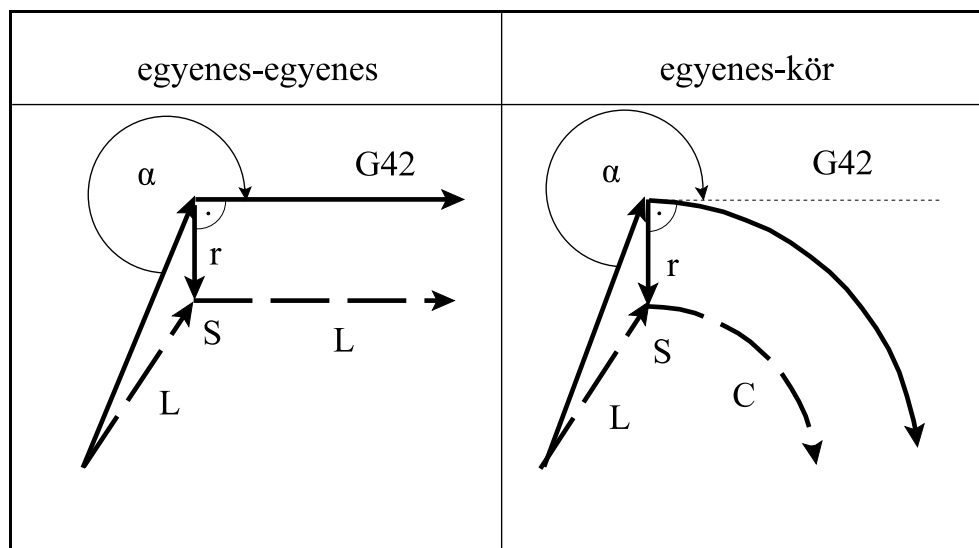
szaggatott vonal: a szerszámközéppont pályája,

folyamatos vonal: a programozott pálya.

A sugárkorrekció bekapcsolásának alapesetei:

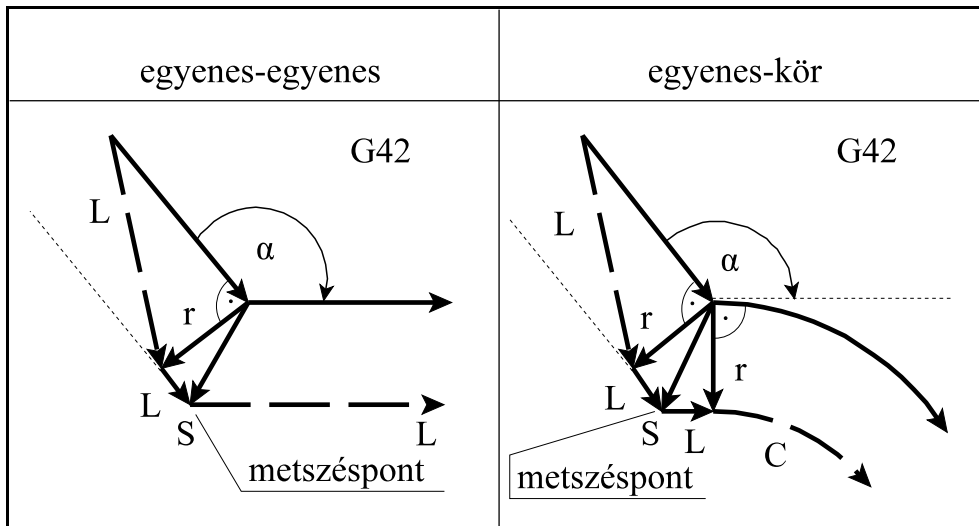
G40	G40
G42 G1 X_ Z_	G42 G1 X_ Z_
X_ Z_	G2 X_ Z_ R_

Belső sarokra való ráállítás: $180^\circ < \alpha < 360^\circ$



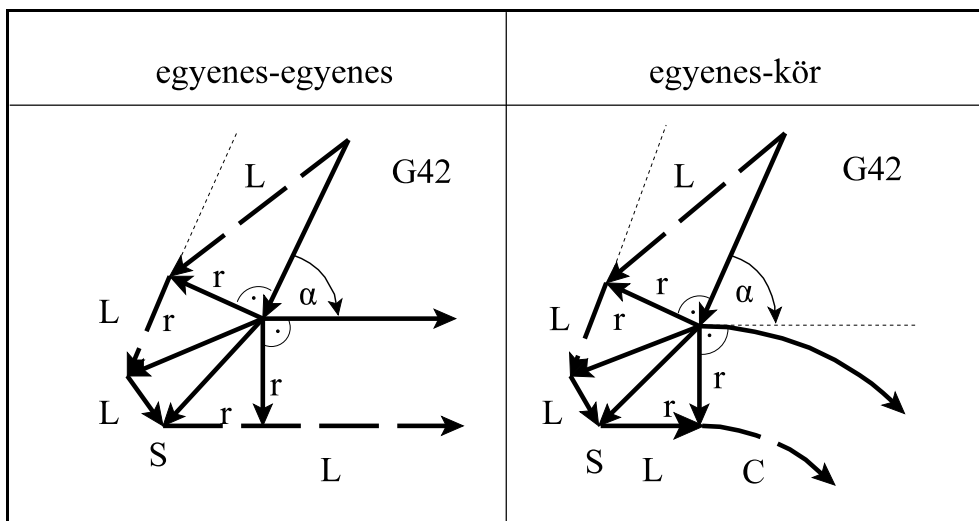
15.7.1-1 ábra

Külső sarokra való ráállás tompaszög alatt: $90^\circ \leq \alpha \leq 180^\circ$



15.7.1-2 ábra

Külső sarokra való ráállás hegyesszög alatt: $0^\circ \leq \alpha < 90^\circ$



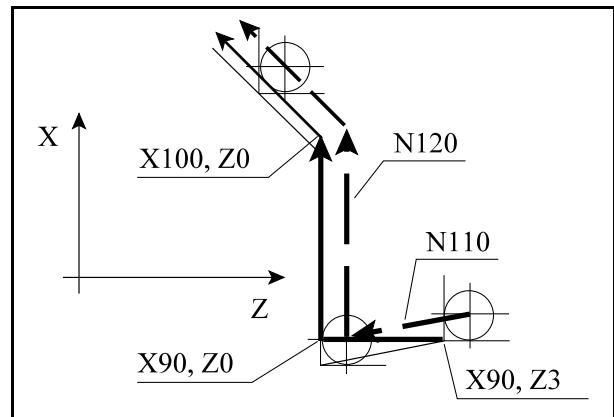
15.7.1-3 ábra

Példa

Az alábbi program egy példát mutat a sugárkorrekció helyes bekapcsolására Q3 szerszámállásnál.

Az N110 mondat kapcsolja be a sugárkorrekciót. Az N110 mondat elejére még a szerszám elméleti csúcspontjával áll rá, majd a mondat végére már a szerszámhegy körét érinti az N120 mondat kezdőpontjához:

```
G54 G18
...
N50 G0 X200 Z50
N60 G92 S3000
N70 G96 S600
N80 T101 M3
N90 X160 Z3
N100 X90
N110 G1 G42 Z0
N120 X100
N130 X110 Z-5
N140 Z-25
N150 G2 X130 Z-35 R10
N160 G1 X140
N170 Z-45
N180 G40 X160
N190 G0 X200 Z50
...
```



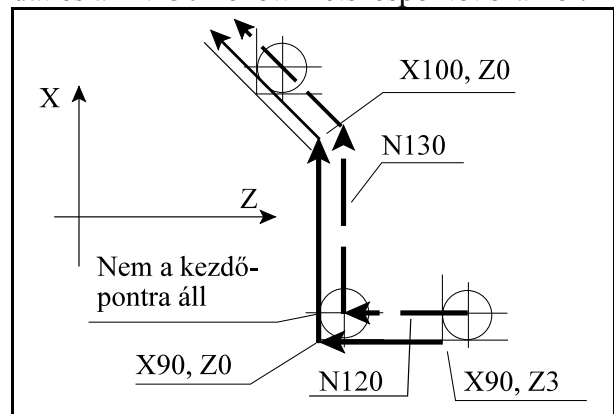
15.7.1-4 ábra

A sugárkorrekció bekapcsolása mozgás programozása nélkül:

Ha a **korrekció bekapcsolását** (G41, G42) olyan mondatban végezzük, amelyben a kiválasztott **síkhoz tartozó egyik tengelycím sem szerepel**, mint az alábbi példában az N110 mondatban sem X, sem Z cím nincs megadva, akkor az N110 mondatban nem végez mozgást. Az N120 mondat végpontjára úgy mozog, hogy az N120 mondat és az N130 között metszéspontot számol.

Ez azért nem jó, mert a így az X90 Z0 pont kimarad a megmunkálásból! Javítható az N100 és az N120 mondat összevonásával.

```
G54 G18
...
N50 G0 X200 Z50
N60 G92 S3000
N70 G96 S600
N80 T101 M3
N90 X160 Z3
N100 X90
N110 G1 G42
N120 Z0
N130 X100
N140 X110 Z-5
```



15.7.1-5 ábra

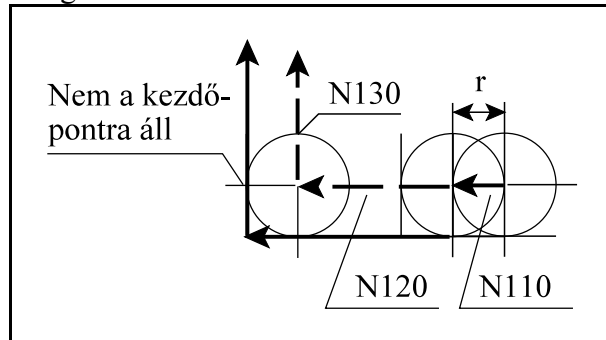
```

N150 Z-25
N160 G2 X130 Z-35 R10
N170 G1 X140
N180 Z-45
N190 G40
N200 X160
N210 G0 X200 Z50
...

```

Ha a **korrekció bekapcsolását** (G41, G42) olyan mondatban végezzük, amelyben a kiválasztott **síkhöz tartozó egyik tengelyen sincs elmozdulás**, mint az alábbi példában az N110 mondatban a Z cím ugyan ki van töltve, de a hozzá tartozó inkrementális 0 miatt nincs elmozdulás, akkor az N110 mondatban szerszámugárnyi elmozdulást végez.

Ilyenkor az N120 és az N130 mondatok között metszéspontot számol és úgy vezeti a szerszám középpontját. Ez azért nem jó mert a így az X90 Z0 sarok kimarad a megmunkálásból! Javítható az N110 és az N120 mondat összevonásával.



15.7.1-6 ábra

```

G54 G18
...
N50 G0 X200 Z50
N60 G92 S3000
N70 G96 S600
N80 T101 M3
N90 X160 Z3
N100 X90
N110 G1 G42 G91 Z0
N120 G90 Z0
N130 X100
N140 X110 Z-5
N150 Z-25
N160 G2 X130 Z-35 R10
N170 G1 X140
N180 Z-45
N190 G91 G40 X0
N200 G90 X160
N210 G0 X200 Z50
...

```

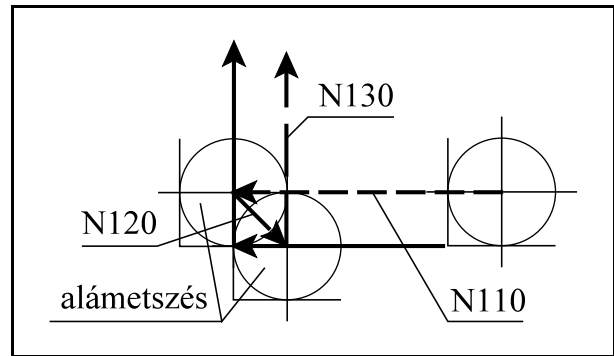
A sugárkorrekció bekapcsolását követő mondatban nincs elmozdulás

Ha a korrekció bekapcsolását követő mondatban, a kiválasztott síkban, 0 elmozdulás adódik, mint az alábbi program N120 mondatában, a szerszámhegy körét a bekapcsolást végző mondat (N110) végpontjára vezeti. Ezután a következő mozgást végző mondat (N130) kezdőpontjára vezeti a szerszámhegy körét, átmozog erre a pontra, majd párhuzamosan viszi tovább a szerszámot. Ez a darabon alámetszést okozhat, ezek az esetek kerülendők! Javítható az N120 mondat törlésével.

```

G54 G18
...
N50 G0 X200 Z50
N60 G92 S3000
N70 G96 S600
N80 T101 M3
N90 X160 Z3
N100 X90
N110 G1 G42 Z0
N120 Z0
N130 X100
N140 X110 Z-5
N150 Z-25
N160 G2 X130 Z-35 R10
N170 G1 X140
N180 Z-45
N190 G91 G40 X0
N200 G90 X160
N210 G0 X200 Z50
...

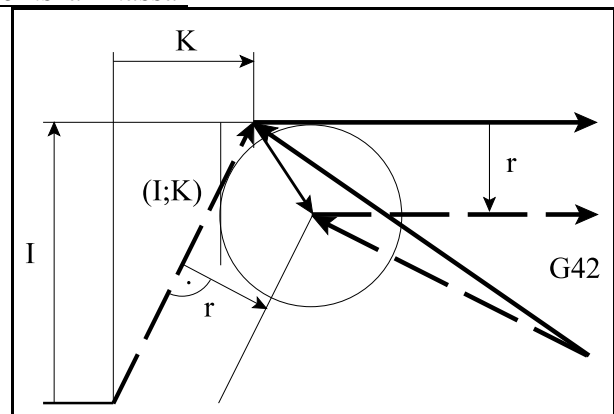
```



15.7.1-7 ábra

Sugárkorrekció bekapcsolása metszéspontszámítással

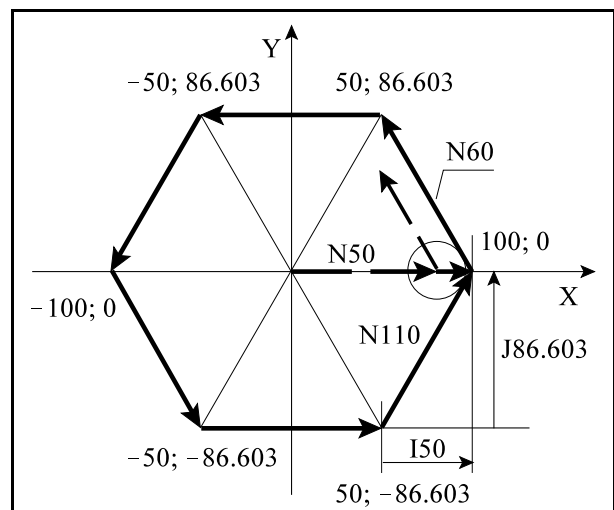
Ha a korrekció bekapcsolását végző mondatban (G41, vagy G42) I, J, K-nak értéket adunk, de csak a kiválasztott síkban lévőeknek (például: G18 esetén I, K-nak), akkor a következő mondat és az I, J, K által meghatározott egyenes közti metszésponttra áll a vezérlő, a sugárkorrekció figyelembe vételével. I, J, K értéke mindig inkrementális, és az általuk megadott vektor annak a mondatnak a végpontjára mutat, amelyikben programoztuk. Ez a lehetőség például belső sarokra való ráállás esetén hasznos.



15.7.1-8 ábra

Példa:

Marjunk meg egy 100 mm élhosszúságú hat szöget belülről. A szerszámmal álljunk be az X100 Y0 koordinátájú ponton lévő sarokba. A mellékelt program N50 mondata metszéspontszámítással áll rá a kontúrra, I és J megadásával. Az I és J koordinátáit a program N110 mondatában végzett elmozdulás alapján számítottuk ki.

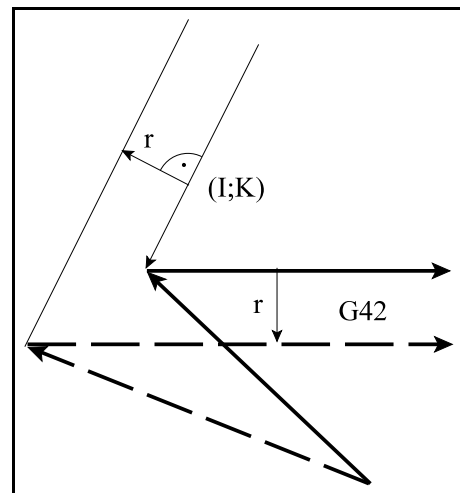


15.7.1-9 ábra

```
N10 G54 G17 M3 S300
N20 G0 X0 Y0 Z100
N30 Z5
N40 G1 Z-5 F1000
N50 G41 G0 X100 Y0 I50 J86.603 D1
N60 G1 X50 Y86.603
N70 X-50
N80 X-100 Y0
N90 X-50 Y-86.603
N100 X50
N110 X100 Y0
N120 G40 G0 X0 I-50 J86.603
N130 M30
```

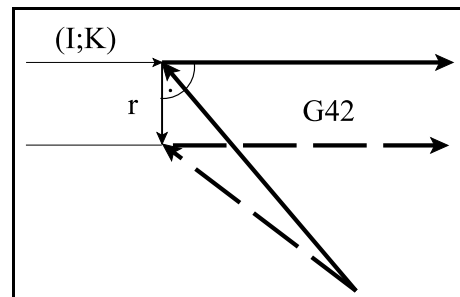
A metszésponttal való ráállás esetei

I, J, K megadás esetén a vezérlő mindig metszéspontot számol, függetlenül attól, hogy belső, vagy külső sarkot munkálunk meg.



15.7.1-10 ábra

Ha nem talál metszéspontot, a következő mondat kezdőpontjára merőlegesen áll rá.



15.7.1-11 ábra

15.7.2 A sugárkorrekció számítás bekapcsolt állapotában. Haladás a kontúron.

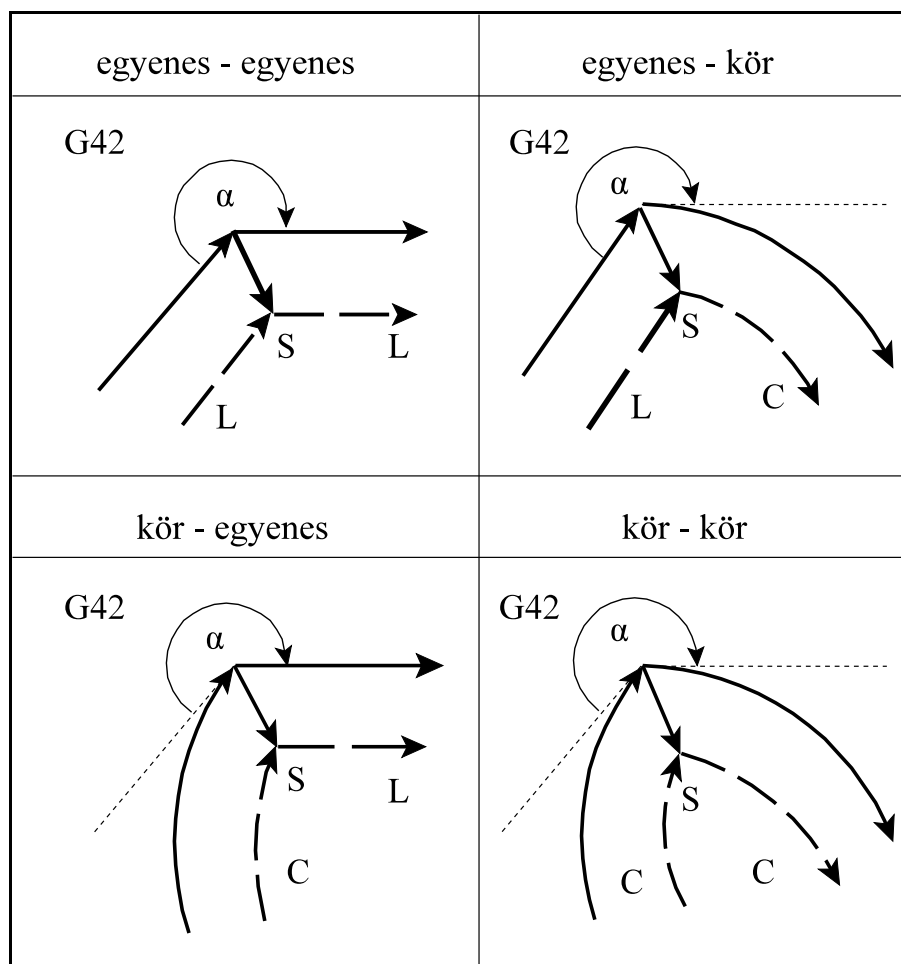
A sugárkorrekció számítás bekapcsolt állapotában a korrekciós vektorok a kijelölt síkban folyamatosan kiszámításra kerülnek a G0, G1, G2, G3 mondatok között. Ahhoz hogy ezeket a vektorokat számítani lehessen, a mondatokat folyamatosan előre kell olvasni.

Az N1404 BK No. Interf paraméteren megadott számú + 2 mondatot olvas előre a vezérlő. Ez azt is jelenti, hogy ha két, a kijelölt síkhoz tartozó mozgásmondat között, a paraméteren megadott számú egyéb mondat van, például funkció, várakozás, síkon kívüli mozgás stb., a korrekciószámítás még folyamatos lesz.

A folyamatosságot megszakítják az olyan G kódok és funkciók, amelyek pufferürítést kényszerítenek ki, azaz felfüggesztik a mondatok előreolvasását.

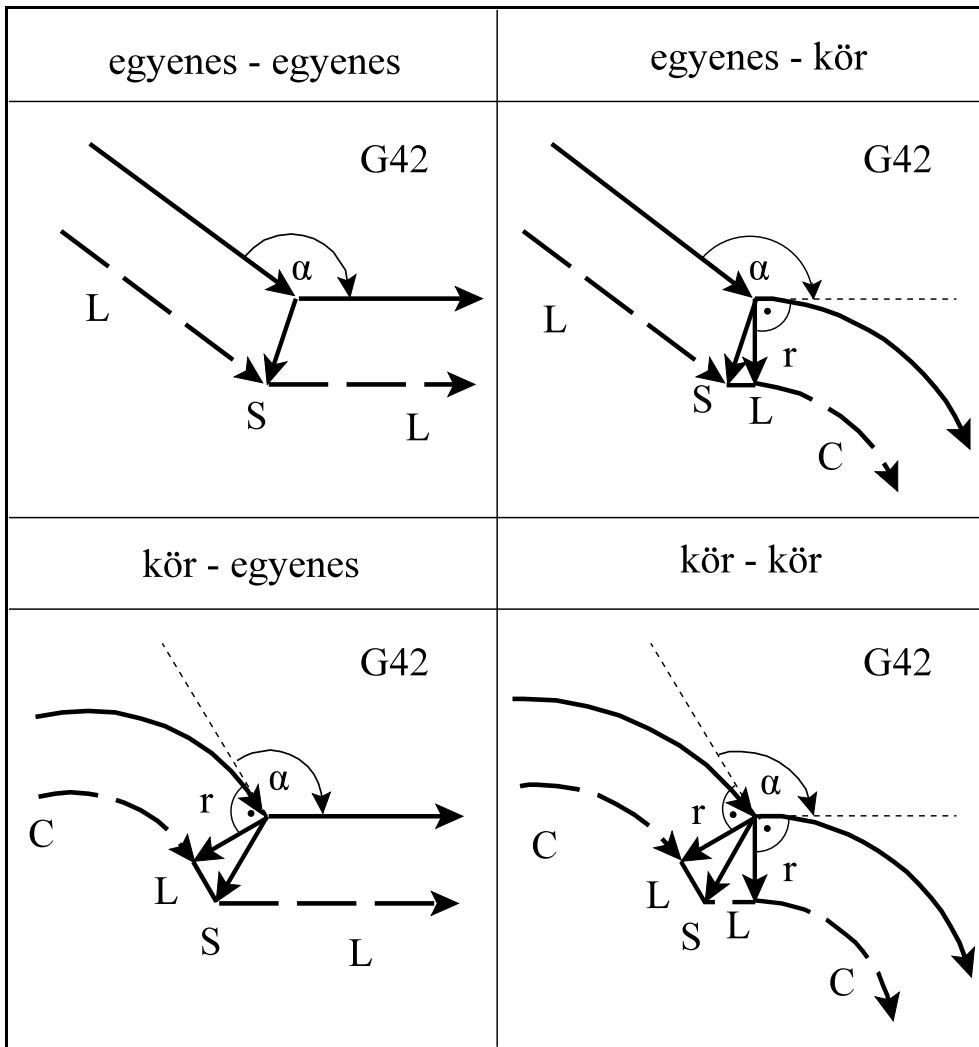
A sugárkorrekció számítás bekapcsolt állapotának alapesetei:

Metszéspontszámítás belső sarkok esetén: $180^\circ < \alpha < 360^\circ$



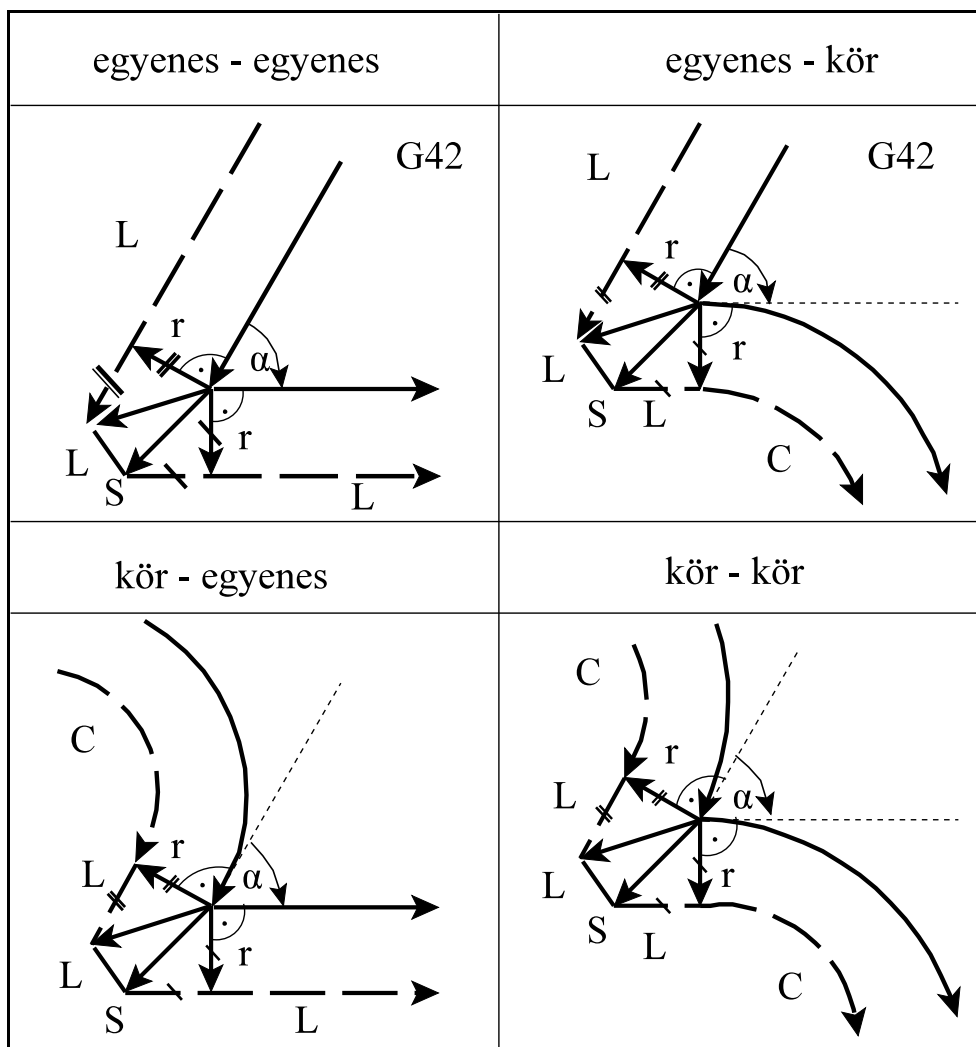
15.7.2-1 ábra

Tompaszögű külső sarkok kerülése: $90^\circ \leq \alpha \leq 180^\circ$



15.7.2-2 ábra

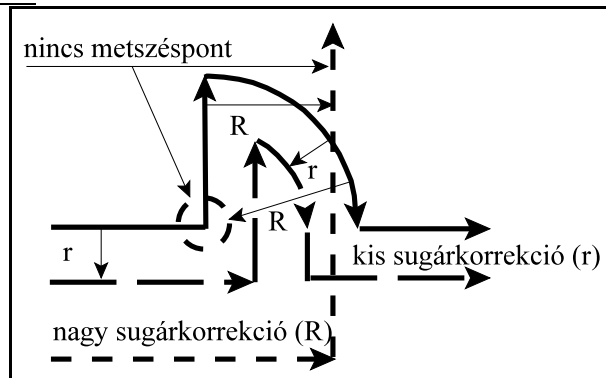
Hegyeszögű külső sarkok kerülése: $0^\circ \leq \alpha < 90^\circ$



15.7.2-3 ábra

Ha nincs metszéspont belső sarkok esetén

Előfordulhat, hogy bizonyos szerszámsugár értékeknél nem adódik metszéspont. Ekkor a vezérlő az előző mondat végrehajtása alatt megáll és 2047 Nincs metszéspont G41, G42 hibajelzést ad.



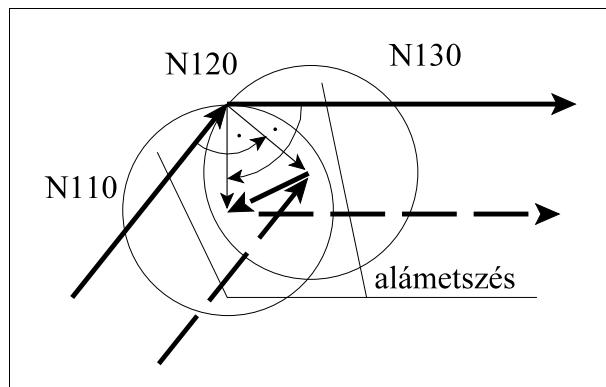
15.7.2-4 ábra

A kiválasztott síkban nulla elmozdulás adódik

Abban az esetben, ha G41, vagy G42 bekapcsolt állapotában, a kiválasztott síkban az egyik mondatban nulla elmozdulást programozunk, vagy nulla elmozdulás adódik, mint a példa N120 mondatában, a következő történik. Az előző mondat (N110) végpontjára és a következő mondat (N130) kezdőpontjára állít egy merőleges vektort, amelynek hossza megegyezik a sugárkorrekcióval, majd a két vektort egyenes interpolációval köti össze. Az ilyen esetekre vigyázni kell, mert szándékolatlan alámetszést, kör esetén torzulást okoz.

Például:

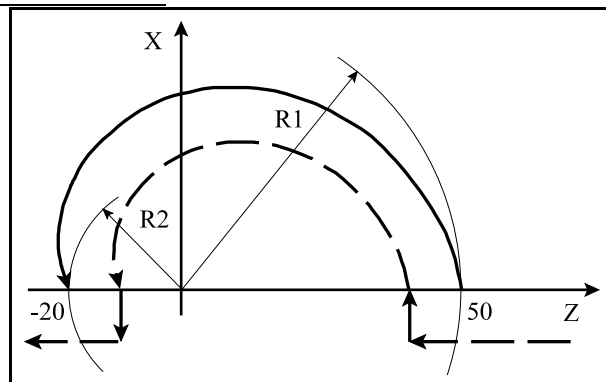
```
... G91 G18 G42 ...
N110 G1 X100 Z40
N120 Z0
N130 Z90
N140 X-40 Z50
...
```



15.7.2-5 ábra

Sugárkorrekció spirális és változó sugarú kör esetén

Ha spirálra, vagy változó sugarú körre alkalmazzuk a sugárkorrekciót a kör kezdőpontjában a korrekciós vektor(oka)t olyan képzeletbeli körhöz számítja ki a vezérlő, amelynek sugara megegyezik a programozott kör kezdőponti sugarával (az ábrán $R1=50$), középpontja pedig egybeesik a programozott középponttal ($X0, Z0$). A kör végpontjában a korrekciós vektor(oka)t olyan képzeletbeli körhöz számítja ki, amelynek sugara megegyezik a programozott kör végponti sugarával ($R2=20$), középpontja pedig egybeesik a programozott kör középponttal.



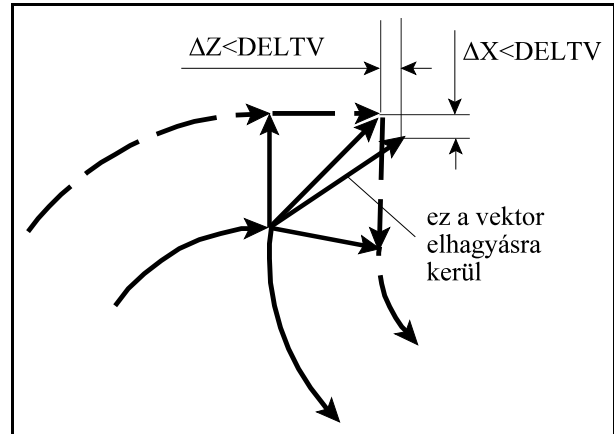
15.7.2-6 ábra

Példa:

```
G0 G18 G41
G1 Z50
G3 Z-20 K-50 L1
G1 Z-30
...
```

Kerülőszakaszok elhagyása

Tompa- és hegyesszögű sarkok kerülése esetén két vagy több korrekciós vektor is képződhet. Ha ezek végpontjai közel esnek egymáshoz, alig adódik mozgás a két pont között. Abban az esetben, ha a két vektor közötti távolság mindkét tengelyen kisebb, mint az N1405 DELTV paraméteren beállított érték, az ábrán látható vektor elhagyásra kerül, és a szerszám pályája az ábra szerint módosul.



15.7.2-7 ábra

- ☞ *Megjegyzés:* A *DELTV* paraméter indokolatlanul nagy értéke mellett külső hegyesszögű sarkok kerülésekor előfordulhat, hogy a sarkot megsérti a szerszám!

A kiválasztott síkban, több mondaton keresztül nincs mozgásparancs

Ahhoz, hogy a szerszám sugar korrekciót helyesen kezelje a vezérlő, például, hogy a beolvasott mondat végpontja és a következő kontúrmondat kezdőpontja között a metszéspontot ki tudja számolni, a **mondatokat előre be kell olvasni és feldolgozni**. Az előre feldolgozott mondatok a mondatpufferbe kerülnek.

A gyakorlatban szükség lehet arra, hogy két síkbeli mozgásmondat közé mozgást nem tartalmazó mondatot, vagy nem a kiválasztott síkba eső mozgást tartalmazó mondatot programozzunk. Ezek például a következők lehetnek:

funkciók: M, S, T

várakozás: G4 P

a kiválasztott síkon kívüli mozgás: (G18) G1 Y

alprogram hívás: M98 P, stb.

Ez azt is jelenti, hogy ha két, a kijelölt síkhoz tartozó mozgásmondat között egyéb mondatok vannak, például funkciók, várakozás, síkon kívüli mozgás stb., a korrekciószámítás még folyamatos lesz, mindaddig, amíg a mondatpuffer meg nem telik.

Ha a **puffer megtelt** a vezérlő **az utolsó, a kijelölt síkhoz tartozó mozgásmondat elején** a

2090 Sugárkorrekciószámítás nem folytatható. Puffer tele.

üzenetet adja.

Ha G41, G42 állapotban pufferürítő funkciót programoztunk

A korrekciószámítás folyamatosságát, azaz a mondatok **előreolvasását megszakítják** az olyan G kódok és funkciók (pl. bizonyos M funkciók stb.), amelyek **pufferürítést** kényszerítenek ki. Amikor az előreolvasás során a vezérlő ilyen kódot olvas be, felfüggeszti a további mondatok előreolvasását és megvárja, hogy kiürüljön a mondatpuffer, azaz a pufferben lévő összes mondat végre legyen hajtva. Ez azzal a következménnyel jár, hogy a **sugárkorrekció számítását felfüg-**

geszti a vezérlő. Ezután végrehajtja a funkciót, majd utána kezdi meg a következő mondatok beolvasását és pufferelesét.

A sugárkorrekció számítás felfüggesztő G kódok a következők:

G22, G23,
G54-G59, G54.1, ...,
G52, G92,
G53,
G28, G30

A T funkció is, akár szerszámcsere, akár korrekcióhívásra vonatkozik, felfüggeszti a korrekciószámítást :

Tnnkk, Tnn00, Tkk

A sugárkorrekció számítás felfüggesztő M funkciók a következők:

M0, M1, M2, M30

A fenti M funkciókon kívül pufferrütést végző M funkciókat és M kód csoportokat paramétereken is ki lehet jelölni. Az M funkciókon kívül az S, és a vegyes funkciók (A, B, C, U, V, W) is kijelölhetők pufferrütésre.

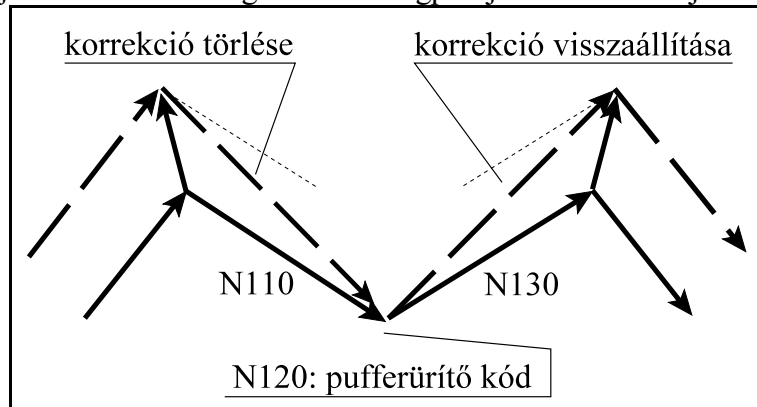
Azt, hogy mely funkciók végeznek pufferrütést, a gép építője határozza meg, és közli a gép leírásában!

Abban az esetben, ha G41, vagy G42 bekapcsolt állapotában két mozgásmondat között a fenti G kódokat, vagy funkciókat programozzuk, az ezt megelőző mondat végpontján törli a korrekciós vektort, végrehajtja a parancsot, majd a következő mozgásmondat végpontjában visszaállítja azt.

Például:

```
...G91 G18 G41...
N110 G1 X-100 Z80
N120 G92 X0 Z0
N130 X100 Z80
...
```

A fenti példában a G92 kód miatt üríti a puffert. Hasonló a helyzet a többi pufferrütő kód esetén is.



15.7.2-8 ábra

Ha G41, G42 állapotban feltételes mondatkihagyást programoztunk

Az N1337 Execution Config paraméter **#4 CBB=0** állásánál a feltételes mondat utasítás (/ jellel kezdődő mondatok) **elnyomja** a mondat előreolvasást. Ebben az esetben **G41, G42** esetén a kontúr **torzul**, viszont a feltételes mondat kapcsolót elég az **előző mondat végrehajtása közben kapcsolni**, hogy hatásos legyen.

Az N1337 Execution Config paraméter **#4 CBB=1** állásánál a feltételes mondat utasítás (/ jellel kezdődő mondatok) **nem nyomja el** a mondat előreolvasást. Ebben az esetben **G41, G42** esetén a kontúr **nem torzul**, viszont a feltételes mondat kapcsolót a biztos hatás miatt **a program végrehajtása előtt be kell állítani**.

A paraméterállítás hatását lásd a Feltételes mondatkihagyás fejezetet.

Példa:

```

...G91 G18 G41...
N110 G1 X-100 Z80
/ N120 S2500
N130 X100 Z80
...

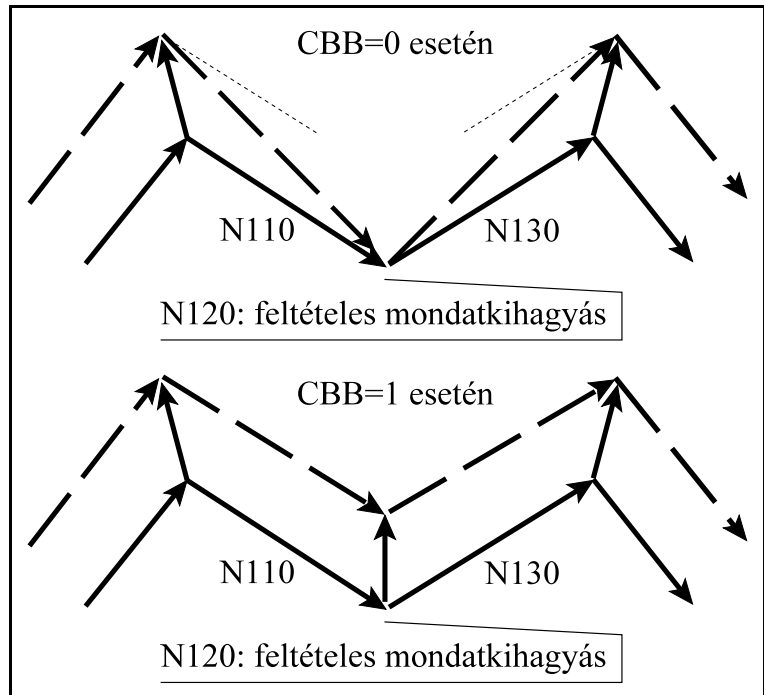
```

A fenti példában az N120 mondat feltételes.

CBB=0 esetén az N110 mondat végpontjában törli a korrekciót és az N130 mondatban helyreállítja azt.

CBB=1 állásnál nem függeszti fel az előreolvasást, a korrekciószámitás folyamatos lesz.

G41, G42 állapotban tanácsos kerülni a feltételes mondat programozását.



15.7.2-9 ábra

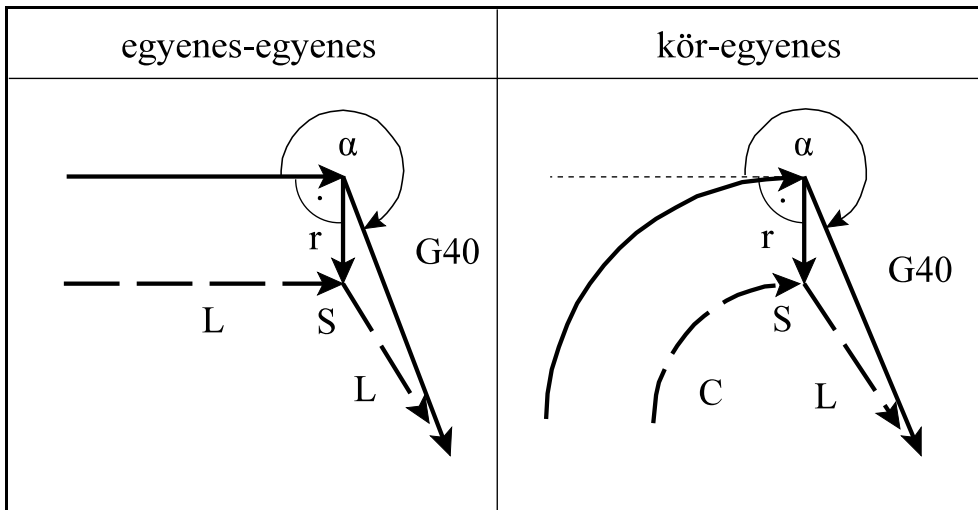
15.7.3 A szerszámsugár korrekciósámítás kikapcsolása. Leállítás a kontúrról.

A G40 parancs kikapcsolja a szerszámsugár korrekciósámítást. G40 parancsot csak lineáris interpolációval lehet kiadni. Ha körmondattban programozunk G40-et, 2043 G40 körinterplációban hibajelzést ad a vezérlő.

A sugárkorrekció kikapcsolásának alapesetei:

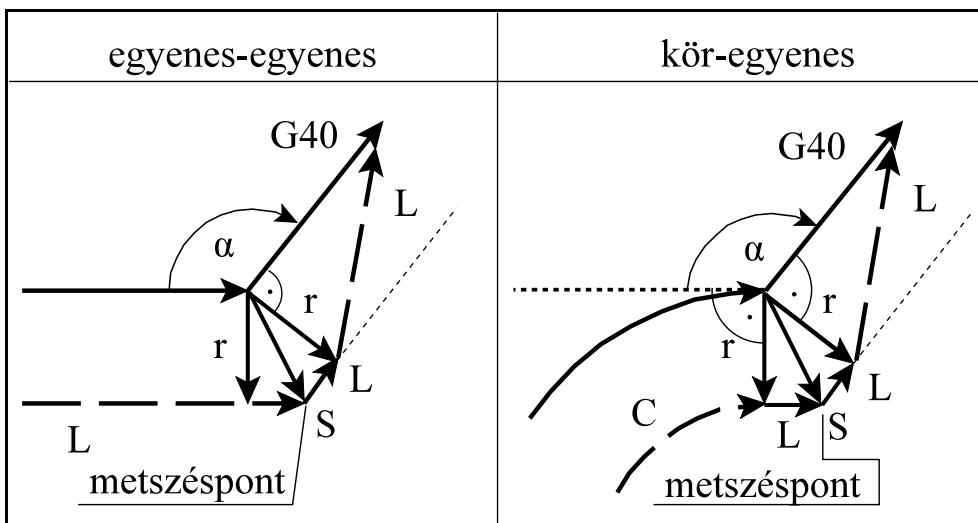
(G42)	(G42)
G01 X_ Z_	G02 X_ Z_ R_
G40 X_ Z_	G40 G1 X_ Z_

Belső sarokból való kiállás: $180^\circ < \alpha < 360^\circ$



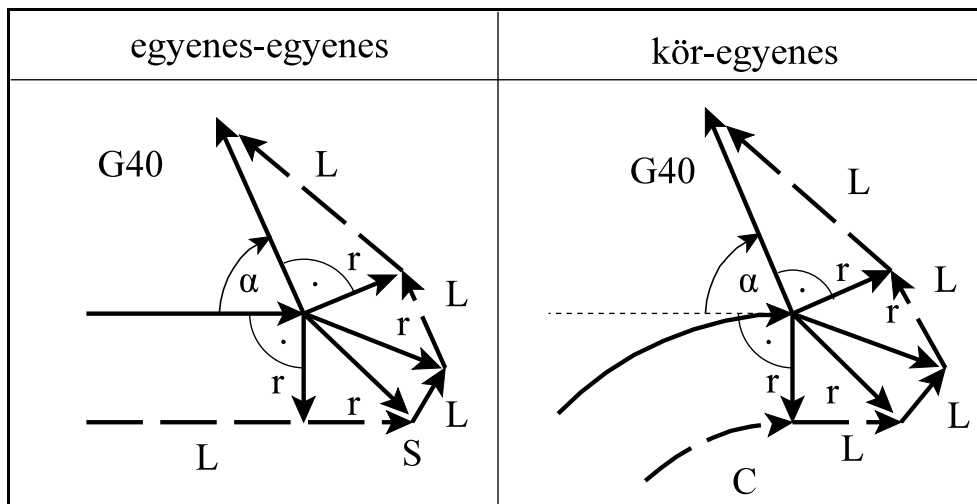
15.7.3-1 ábra

Külső sarokból való kiállás tompaszög alatt: $90^\circ \leq \alpha \leq 180^\circ$



15.7.3-2 ábra

Külső sarokból való kiállítás hegyesszög alatt: $0^\circ \leq \alpha < 90^\circ$



15.7.3-3 ábra

Példa

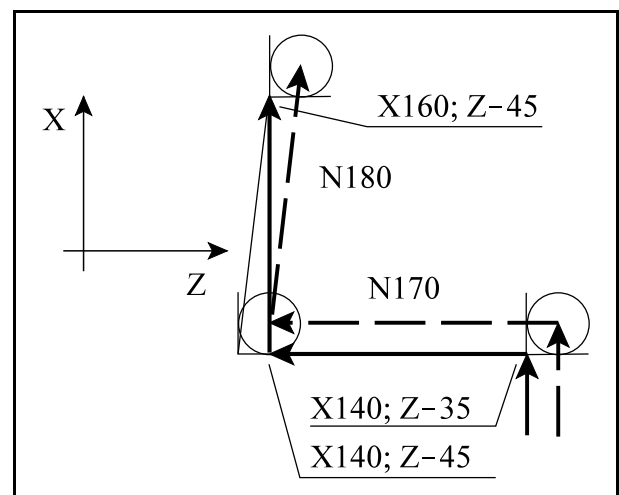
Az alábbi program egy példát mutat a sugárkorrekció helyes kikapcsolására, Q3 szerszámállásnál.

Az N180 mondat kapcsolja ki a sugárkorrekciót. Az N170 mondat végére a szerszám sugár szerint áll rá, a szerszám elméleti csúcsa túlnyúl a végponton, majd az N180 mondat végére már a szerszám elméleti csúcsa lesz a végponti koordinátán:

```

G54 G18
...
N50 G0 X200 Z50
N60 G92 S3000
N70 G96 S600
N80 T101 M3
N90 X160 Z3
N100 X90
N110 G1 G42 Z0
N120 X100
N130 X110 Z-5
N140 Z-25
N150 G2 X130 Z-35 R10
N160 G1 X140
N170 Z-45
N180 G40 X160
N190 G0 X200 Z50
...

```



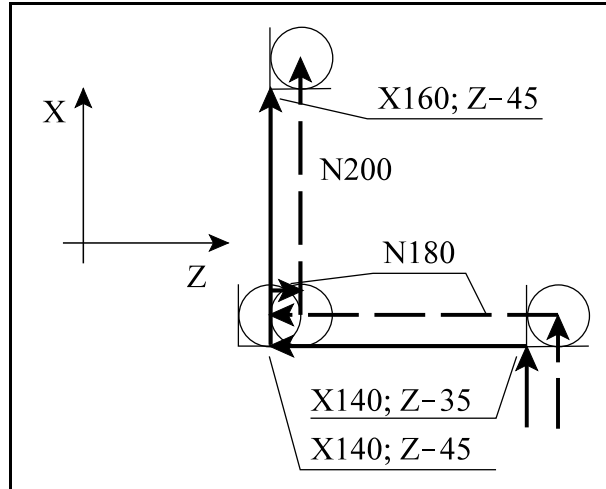
15.7.3-4 ábra

A sugárkorrekció kikapcsolása mozgás programozása nélkül:

Ha a **korrekció kikapcsolását** (G40) olyan mondatban végezzük, amelyben a kiválasztott **síkhoz tartozó egyik tengelycím sem szerepel**, mint az alábbi példában az N190 mondatban sem X, sem Z cím nincs megadva, akkor az N190 mondatban nem végez mozgást. Az N180 mondatban először a szerszámsugarat rávezeti a végpontra, majd visszahúzza lelépi a szerszámsugárnyi távolságot..

Javítható az N190 és az N200 mondat összevonásával.

```
G54 G18
...
N50 G0 X200 Z50
N60 G92 S3000
N70 G96 S600
N80 T101 M3
N90 X160 Z3
N100 X90
N110 G1 G42
N120 Z0
N130 X100
N140 X110 Z-5
N150 Z-25
N160 G2 X130 Z-35 R10
N170 G1 X140
N180 Z-45
N190 G40
N200 X160
N210 G0 X200 Z50
...
```



15.7.3-5 ábra

Ha a **korrekció kikapcsolását** (G40) olyan mondatban végezzük, amelyben a kiválasztott **síkhoz tartozó egyik tengelyen sincs elmozdulás**, mint az alábbi példában az N190 mondatban az X cím ugyan ki van töltve, de a hozzá tartozó inkrementális 0 miatt nincs elmozdulás, akkor ugyanaz játszódik le, mint a fenti esetben, amikor nem programoztunk tengelycímet.

```
G54 G18
...
N50 G0 X200 Z50
N60 G92 S3000
N70 G96 S600
N80 T101 M3
N90 X160 Z3
N100 X90
N110 G1 G42 G91 Z0
N120 G90 Z0
N130 X100
N140 X110 Z-5
N150 Z-25
N160 G2 X130 Z-35 R10
N170 G1 X140
N180 Z-45
```

```

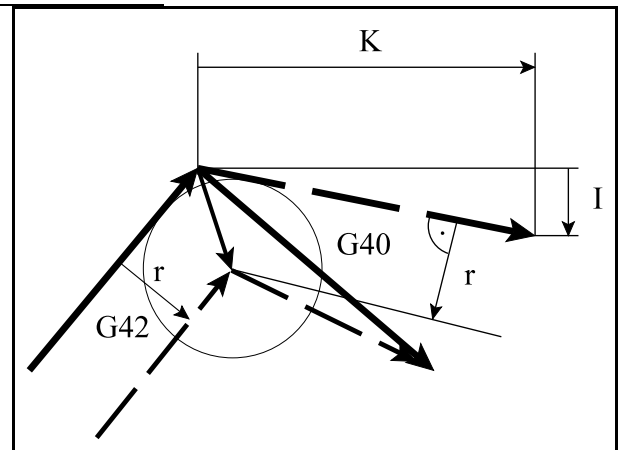
N190 G91 G40 X0
N200 G90 X160
N210 G0 X200 Z50
...

```

Sugárkorrekció kikapcsolása metszéspontszámítással

Ha a korrekció kikapcsolását végző mondatban (G40) I, J, K-nak értéket adunk, de csak a kiválasztott síkban lévőknek (például: G18 esetén I, K-nak), akkor a megelőző mondat és az I, J, K által meghatározott egyenes közti metszéspontra áll a vezérlő. I, J, K értéke mindig inkrementális, és az általuk megadott vektor a megelőző mondat végpontjától elfele mutat.

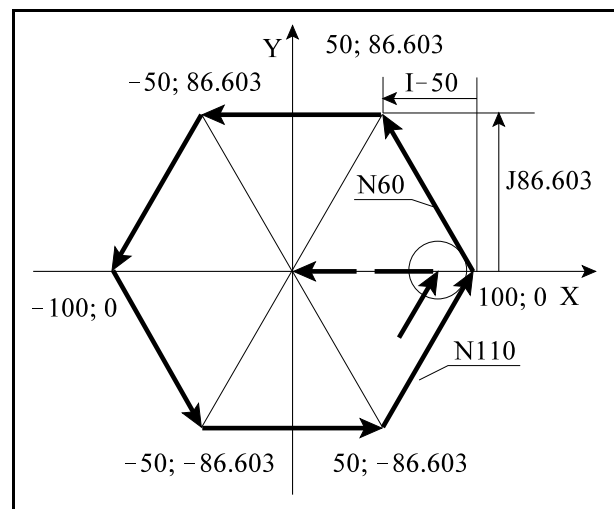
Ez a lehetőség például belső sarokból való kiállítás esetén hasznos.



15.7.3-6 ábra

Példa:

Mmarjunk egy 100 mm élhosszúságú hatszöget belülről. A szerszámmal álljunk ki az X100 Y0 koordinátájú ponton lévő sarokból. A mellékelt program N120 mondata metszéspontszámítással áll le a kontúrról, I és J megadásával. Az I és J koordinátáit a program N60 mondatában végzett elmozdulás alapján számítottuk ki.



15.7.3-7 ábra

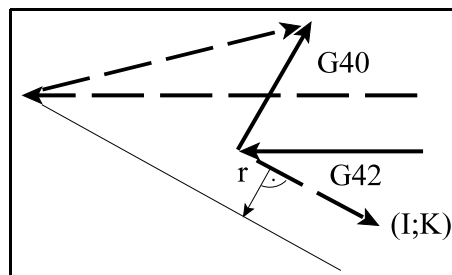
```

N10 G54 G17 M3 S300
N20 G0 X0 Y0 Z100
N30 G43 Z5 H1
N40 G1 Z-5 F1000
N50 G41 G0 X100 Y0 I50 J86.603 D1
N60 G1 X50 Y86.603
N70 X-50
N80 X-100 Y0
N90 X-50 Y-86.603
N100 X50
N110 X100 Y0
N120 G40 G0 X0 I-50 J86.603
N130 M30

```

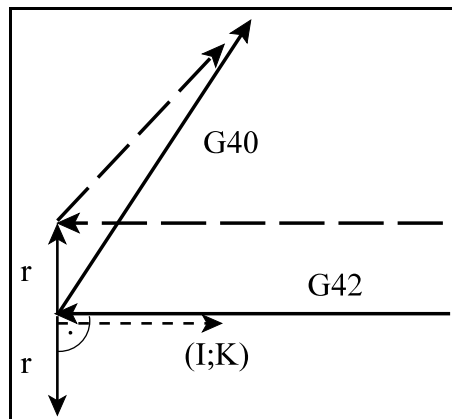
A metszésponttal való leállítás esetei

I, J, K megadás esetén a vezérlő mindig metszéspontot számol, függetlenül attól, hogy belső, vagy külső sarkot munkálunk meg.



15.7.3-8 ábra

Ha nem talál metszéspontot, az előző mondat végpontjára merőlegesen áll rá.



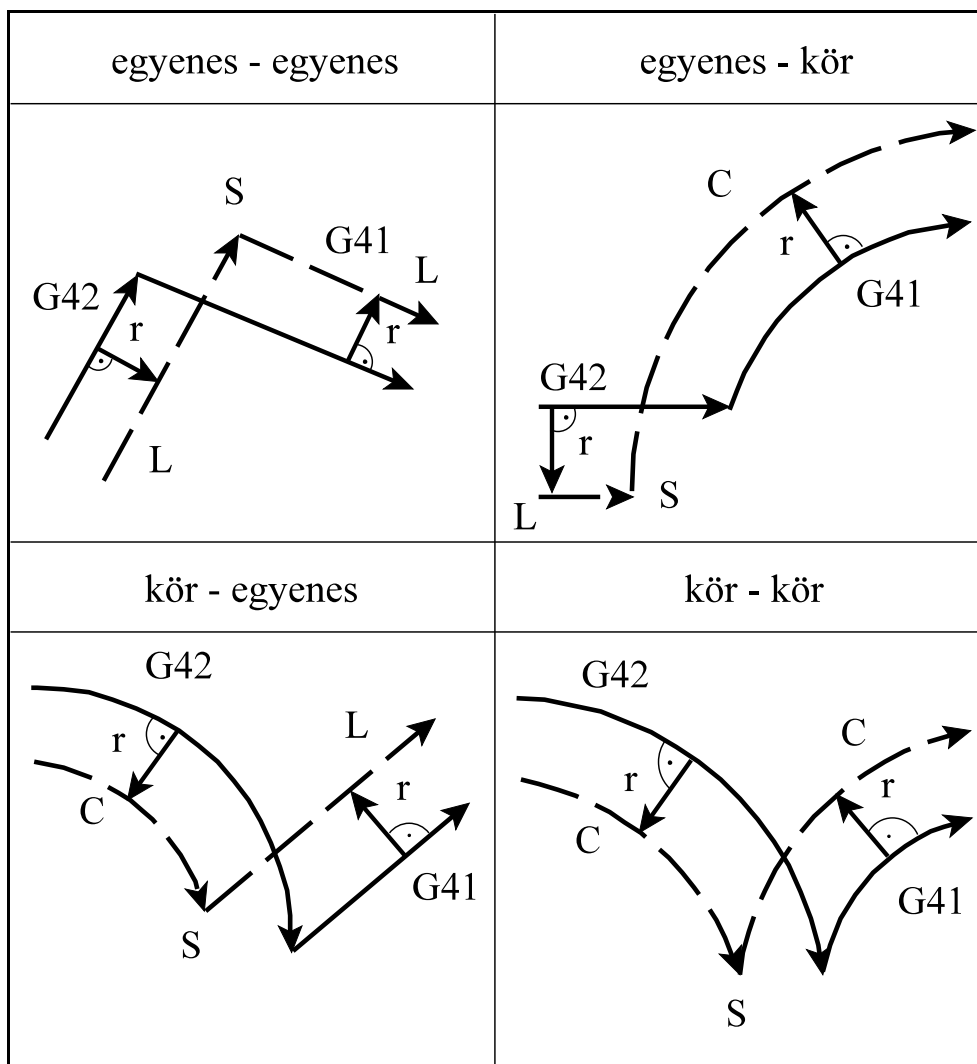
15.7.3-9 ábra

15.7.4 Irányváltás a sugárkorrekció számításban.

A szerszámsugar korrekció számításának, vagyis a kontúr követésének irányát az alábbi táblázat tartalmazza:

	sugárkorrekció: pozitív	sugárkorrekció: negatív
G41	balról	jobbról
G42	jobbról	balról

A kontúrkövetés iránya megfordítható a szerszámsugar korrekció számítás bekapcsolt állapotában is. Ez történhet G41, vagy G42 programozásával. Amikor a kontúrkövetés iránya megfordul a vezérlő nem vizsgálja, hogy "kívül", vagy "belül" van, hanem első lépésben mindig metszéspontot számít. Az alábbi ábrákon pozitív szerszámsugarat tételeztünk fel és G42-ből G41-be történő kapcsolást:



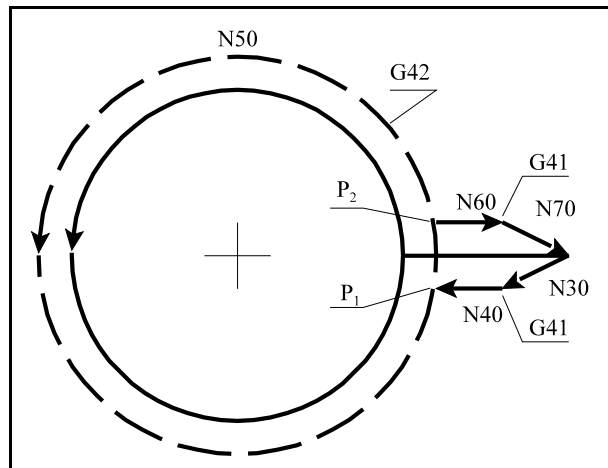
15.7.4-1 ábra

Példa

Egy R50 sugarú körre egyenes mentén, balról, G41-gyel állunk az N30 és az N40 mondattal. Ezután a kört az N50 mondatban jobbról G42-vel követjük. A leállás szintén egyenes mentén, balról, G41-gyel történik.

```

...
N10 G17 G0 X100 Y0
N20 T101 M3 S500
N30 G41 X80
N40 G1 X50
N50 G42 G3 I-50
N60 G41 G1 X80
N70 G40 X100
...
    
```



15.7.4-2 ábra

Ha teljes körön irányváltást programozunk

Teljes kört programozva G41 - G42 váltással, számos olyan eset adódhat, amikor a szerszám pályája többet tesz meg egy teljes kör fordulatnál.

A fenti példa ezt az esetet mutatja.

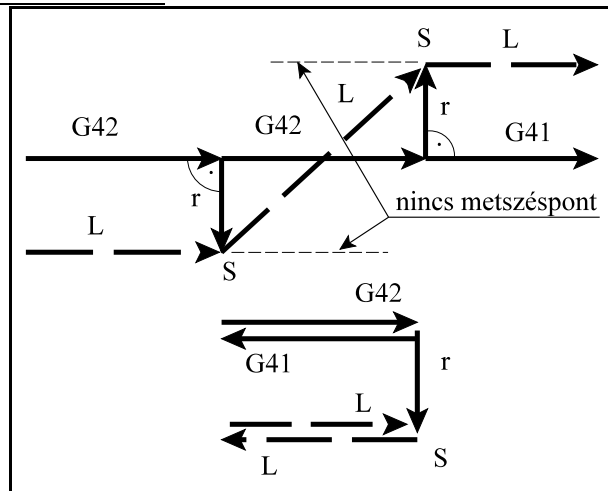
A szerszámközpont tesz egy teljes körívet a P₁ ponttól P₁ pontig, majd egy ívet P₁ ponttól P₂ pontig.

Ha nincs metszéspont egyenes - egyenes átmenetnél

Ha egyenes - egyenes átmenetnél nem adódik metszéspont, a szerszám pályája a mellékelt ábrán látható lesz. Az irányváltás az ábra szerint G42-ről G41-re történik.

Ilyenkor az irányváltást megelőző mondat végpontjában a következő egyenes kezdőpontjára merőlegesen állított, szerszámugárnyi vektor lesz a végpont.

A felső ábra azt mutatja, ha a pálya iránya nem változik, az alsó azt, ha a pálya irányváltása 180 fokos.

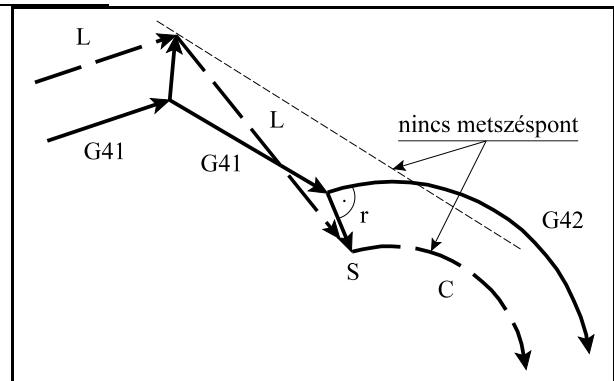


15.7.4-3 ábra

Ha nincs metszéspony egyenes - kör átmenetnél

Ha egyenes - kör átmenetnél nem adódik metszéspony, a szerszám pályája a mellékelt ábrán látható lesz. Az irányváltás az ábra szerint G41-ről G42-re történik.

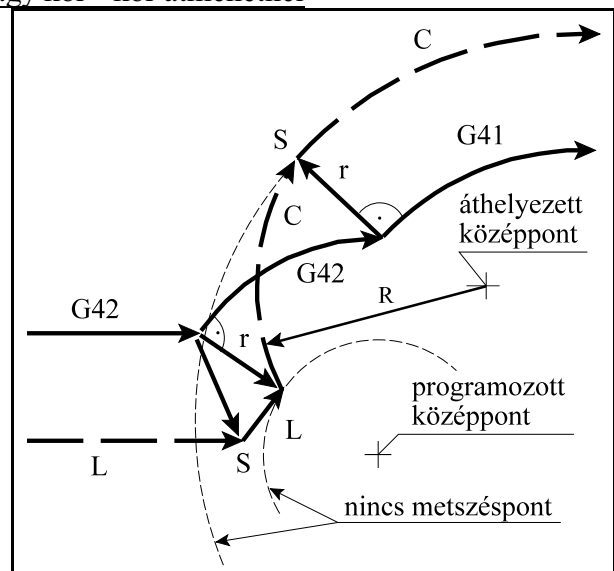
Ilyenkor az irányváltást megelőző egyenes végpontjában a következő kör kezdőpontjára merőlegesen állított, szerszám sugarányi vektor lesz a végpony.



15.7.4-4 ábra

Ha nincs metszéspony kör - egyenes, vagy kör - kör átmenetnél

Ha kör - egyenes, vagy kör - kör átmenetnél nem adódik metszéspony, akkor az első körmondat kezdőpontjában kiadóó korrekciós vektor végpontját és a második mondat kezdőpontjára merőlegesen állított korrekciós vektor végpontját összeköti egy korrigálatlan, programozott R sugarú körívvel. Ekkor az összekötő körív középpontja nem fog egybeesni a programozott körív középpontjával. Ha az irányváltás még a fent vázolt kör - középpont áthelyezéssel sem végezhető el hibajelzést ad a vezérlő.



15.7.4-5 ábra

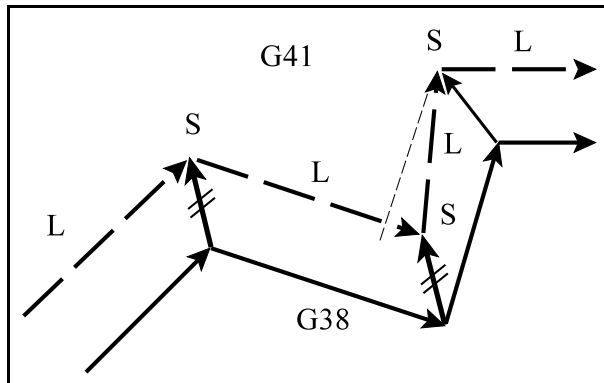
15.7.5 A vektor megtartás programozása (G38)

A

G38 v

parancs hatására a vezérlő a síkbeli szerszám-sugár korrekció számítás bekapcsolt állapotában az *előző mondat és a G38 mondat közötti utolsó korrekciós vektort megtartja és azt érvényesíti a G38 mondat végén*, függetlenül a G38 mondat, és a következő mondat közti átmenettől.

A G38 kód egylovétű, azaz nem öröklődik. Ha több egymást követő mondatban szükséges a vektor megtartása a G38-at újra programozni kell.



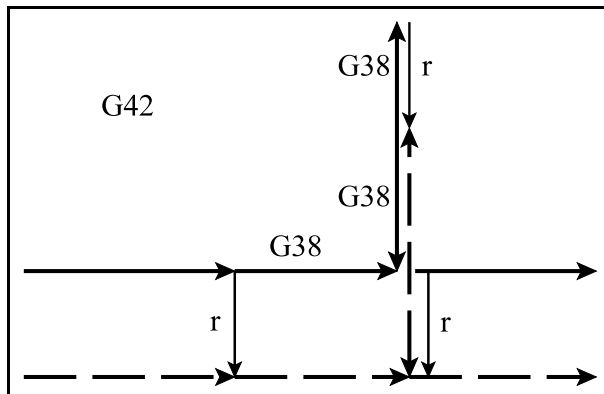
15.7.5-1 ábra

Példa

Programozunk beszúrást a kontúrkövetés ki-kapcsolása nélkül:

```

...G18 G42 G91...
N110 G1 Z40
N120 G38 Z50
N130 G38 X140
N140 G38 X-140
N150 Z60
...
    
```



15.7.5-2 ábra

15.7.6 Sarokív programozása (G39)

A

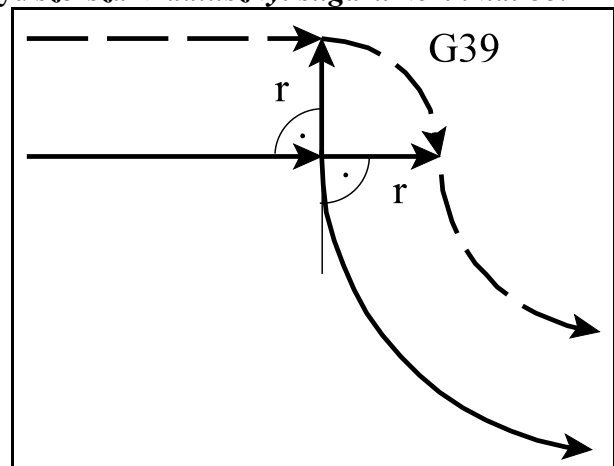
G39 (I J K)

mondat programozásával a síkbeli szerszámugár korrekciószámítás bekapcsolt állapotában elérhető, hogy külső sarkok kerülése esetén a vezérlő ne számoljon automatikusan metszéspontot, vagy ne iktasson be egyenes szakaszokat a kerüléshez, hanem egy szerszámugárnyi köríven mozogjon a szerszám középpontja.

G41 állapotban G02, G42 állapotban G03 irányú *szerszámrádiusznyi sugarú kört iktat be.*

A kör kezdőpontját az előző mondat pályájának végpontjára merőleges szerszámrádiusznyi hosszúságú vektor, végpontját pedig a következő mondat kezdőpontjára merőleges szerszámrádiusznyi hosszúságú vektor adja. **G39-et külön mondatban kell programozni:**

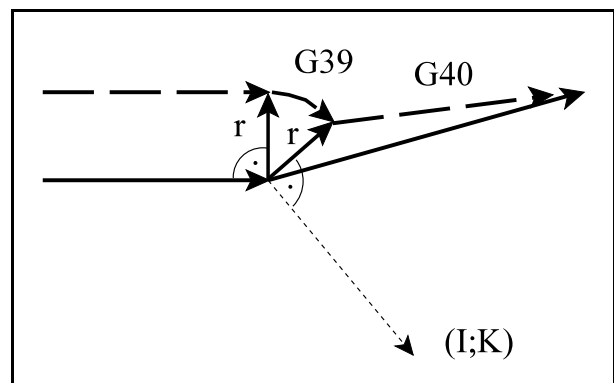
```
...G18 G91 G41...
N110 G1 Z100
N120 G39
N130 G3 X-160 Z80 K80
...
```



15.7.6-1 ábra

Ha a G39-es mondatban I-t, J-t vagy K-t programozunk, a kiválasztott síknak megfelelően, a kör végpontját az előző mondat végpontjától az I, J vagy K által meghatározott vektorra merőleges irányú, szerszámrádiusznyi hosszúságú vektor adja:

```
...G18 G91 G41...
N110 G1 Z100
N120 G39 I-60 K50
N130 G40 X60 Z110
...
```



15.7.6-2 ábra

Az I, J vagy K által meghatározott vektorra érvényesek az előzőleg beállított tükrözési, vagy elforgatási parancsok. A léptékezési parancs értelemszerűen az irányt nem befolyásolja.

A G39-es típusú mondatban semmilyen mozgásparancsot nem lehet programozni.

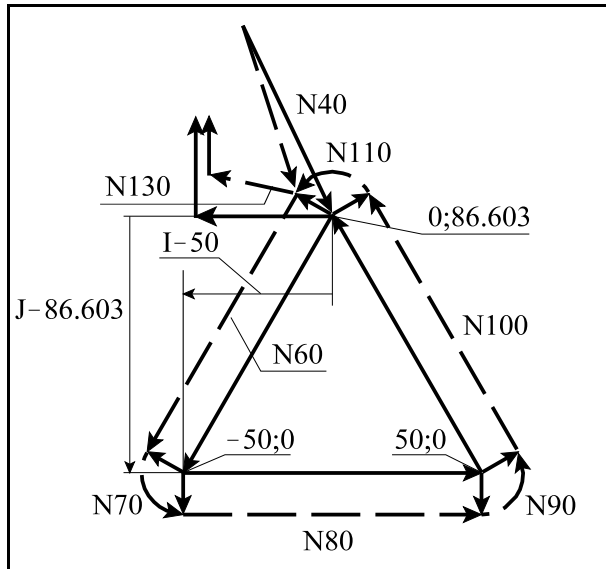
Példa

Marjunk egy háromszög alakú hornyot, ahol a sarkoknál lekerekítés kell.

Ahhoz, hogy a kontúr zárt legyen az N110 mondatban a G39-et I, J-vel adtuk meg. I, J adatait az N60 mondat elmozdulásából határoztuk meg.

```

...
N20 G0 G17 G40 G54
N30 X-30 Y150 Z5 M3 S300
N40 G42 X0 Y86.603 D1
N50 Z-2
N60 G1 X-50 Y0 F1000
N70 G39
N80 X50
N90 G39
N100 X0 Y86.603
N110 G39 I-50 J-86.603
N120 Z5
N130 X-40
N140 G40 Y120
...
    
```

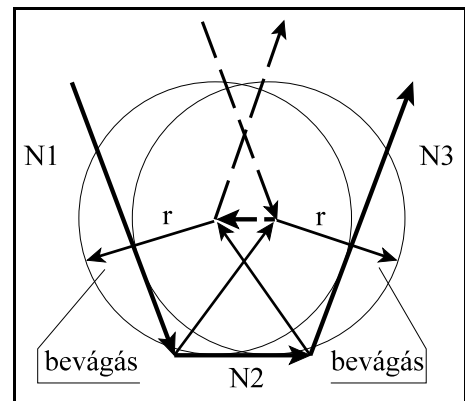


15.7.6-3 ábra

15.7.7 A kontúrkövetés zavarproblémái. Interferenciavizsgálat.

A kontúrkövetés végrehajtása során számos esetben előfordulhat, hogy a szerszám pályája ellentétes lesz a programozott pályával. Ebben az esetben, **a programozói szándékkal ellentétesen, a szerszám belevághat a munkadarabba.** Ezt a jelenséget nevezzük a **kontúrkövetés zavarának**, vagy **interferenciának**.

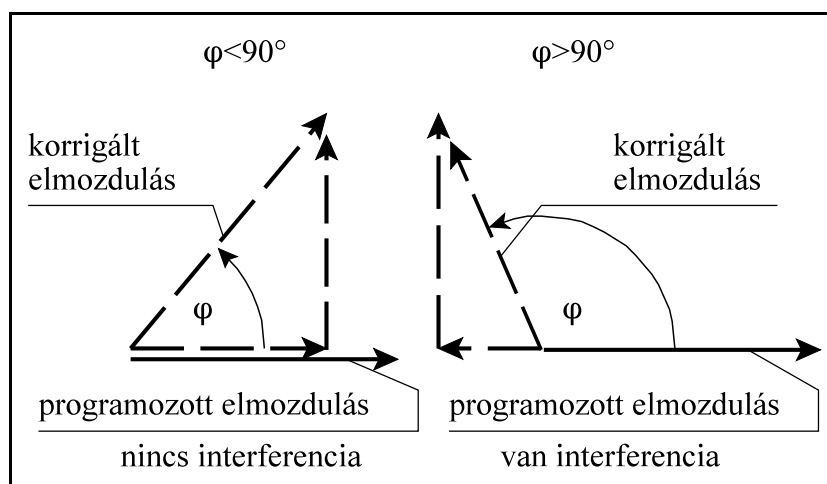
Az ábrán látható esetben a metszéspontok kiszámítása után az N2 mondat végrehajtása során a programozottal (folyamatos vonal) ellentétes szerszám-pálya (szaggatott vonal) adódik. A szerszám belevág a munkadarabba.



15.7.7-1 ábra

Az ilyen esetek elkerülése érdekében a vezérlő interferenciavizsgálatot végez, az N1403 Interference paraméter #0 IEN=1 bitállásánál. Ebben az esetben **a vezérlő azt vizsgálja, hogy a programozott elmozdulás és a sugárkorrekcióval korrigált elmozdulás közötti φ szögre teljesül-e a következő feltétel: $-90^\circ \leq \varphi \leq +90^\circ$.**

Más szavakkal, a vezérlő azt vizsgálja, hogy a korrigált elmozdulásvektornak van-e a programozott elmozdulásvektorral ellentétes irányú komponense.



15.7.7-2 ábra

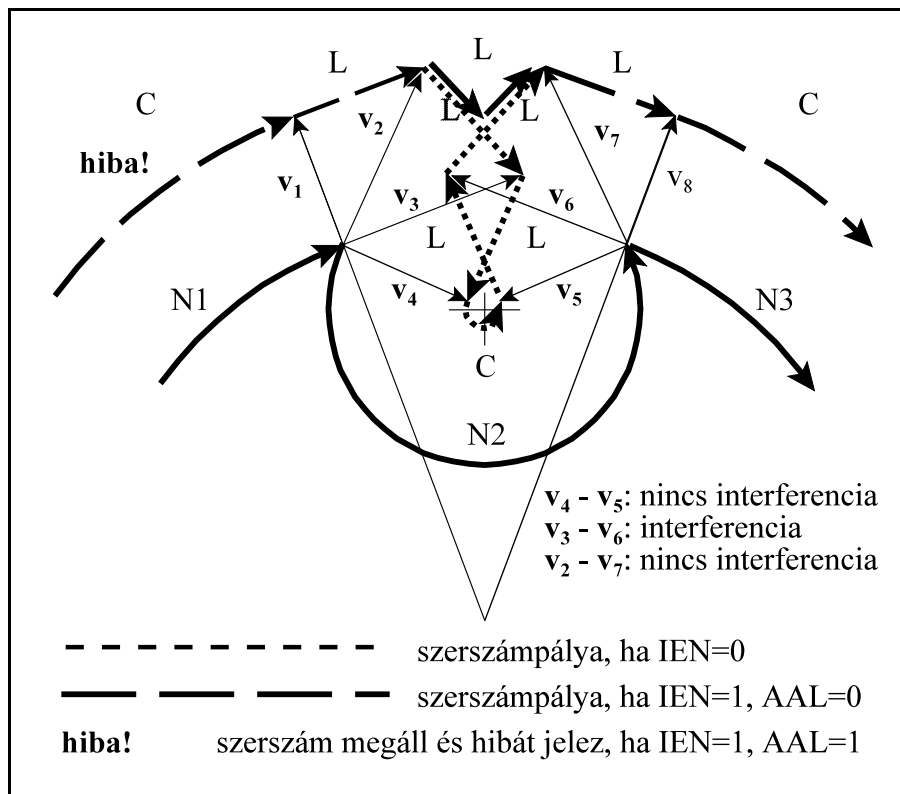
Ha a vezérlő interferenciahibát állapít meg, az N1403 Interference paraméter #1 AAL bitállásának függvényében **hibát jelez, vagy megpróbálja az interferenciahibát javítani.**

Ha az $AAL=1$, akkor a vezérlő **mindig 2049 Interferenciahiba üzenetet ad** az interferenciahibát kiváltó mondatot megelőző mondat végén.

Ha az $AAL=0$, akkor a vezérlő **megpróbálja a hibát automatikusan korrigálni** és csak akkor jelez hibát, ha az automatikus korrekció nem vezet eredményre.

Az **interferenciahiba figyeléséhez**, illetve automatikus javításához **a vezérlő előre beolvas alapesetben 3 mondatot.** Ha az **N1404 BK No. Interf paraméter értéke nagyobb, mint 0, akkor 3 + paraméterérték számú mondatot olvas előre** és végez interferenciavizsgálatot.

Az interferencia hiba automatikus javítása



15.7.7-3 ábra

AAL=0 esetben, a vezérlő nem jelez hibát, hanem automatikusan korrigálni próbálja a kontúrt azzal a céllal, hogy a bevágásokat elkerülje. A korrekció menete a következő:

A szerszámugár korrekció (G41) be van kapcsolva az N1, N2, és N3 mondaton.

N1 és N2 mondat között a kiszámított korrekciós vektorok: v_1 , v_2 , v_3 és v_4 .

N2 és N3 mondat között a kiszámított korrekciós vektorok: v_5 , v_6 , v_7 és v_8 .

Ha v_4 és v_5 között interferencia van (az N2 elmozdulásával ellentétes elmozdulás adódik), a v_3 és a v_4 közötti és a v_5 és v_6 közötti egyenesek között metszéspontot számol, és az N2 körívet kihagyja

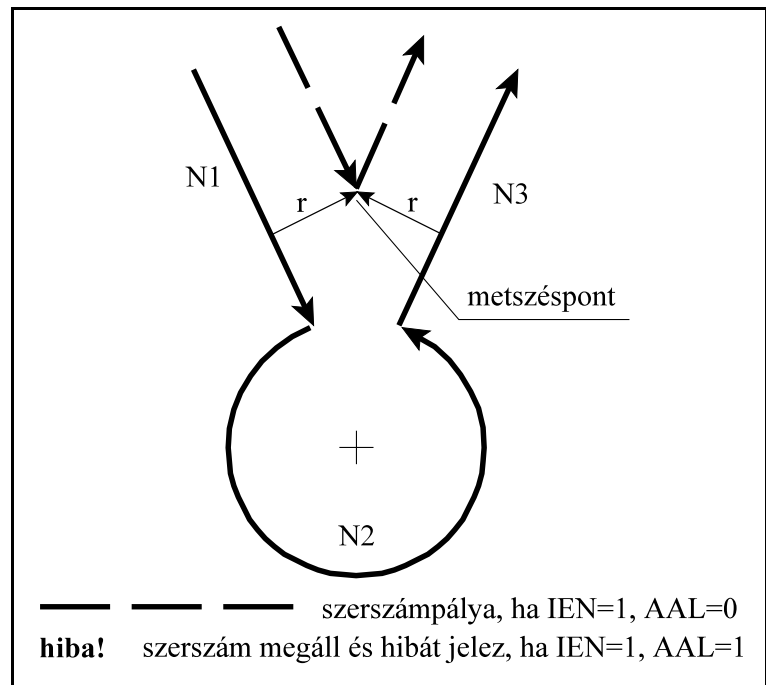
Ha v_3 és v_6 között interferencia van (az N2 elmozdulásával ellentétes elmozdulás adódik), a v_2 és a v_3 közötti és a v_6 és v_7 közötti egyenesek között metszéspontot számol, és a közöttük lévő mozgásokat kihagyja (a fenti ábra ezt az esetet mutatja),

Ha v_2 és v_7 között interferencia van (az N2 elmozdulásával ellentétes elmozdulás adódik), a v_1 és a v_2 közötti és a v_7 és v_8 közötti egyenesek között metszéspontot számol, és a közöttük lévő mozgásokat kihagyja,

Ha v_1 és v_8 között interferencia van az N1 és az N3 mondat között próbál metszéspontot számítani.

A fenti példából látható, hogy az N1 mondat végrehajtás csak akkor kezdődik el, ha az N2 mondatra elvégezte a vezérlő az interferenciavizsgálatot. Ehhez azonban az N3 mondatot is be kellett olvasni a pufferbe, és a N2 - N3 átmenetnél a korrekciós vektorokat kiszámítani.

Ha az N2 mondatban a kerülőszakaszok közötti metszéspontszámítás nem tudja megszüntetni az interferenciahibát AAL=0 esetben, megpróbál az N1 és az N3 mondat között metszéspontot számítani. Ha létezik metszéspont továbbmegy, ha nem 2049 Interferenciahiba üzenetet ad.



15.7.7-4 ábra

Interferenciafigyelés több mondattal előre

A fenti példákban 3 mondatot vizsgál előre a vezérlő. Az N1 és az N2 közötti átmenetet, illetve az N2 és az N3 közöttit. Ez felel meg az N1404 BK No. Interf=0 paraméterállításnak.

Az N1404 BK No. Interf paraméter maximális értéke 8 lehet. Ekkor a fenti vizsgálatokat az

N1 és N2 és az N2 és N3 közötti

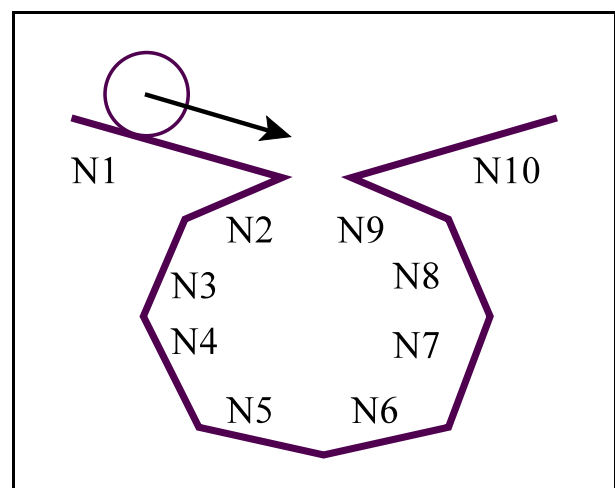
N1 és N2 és az N3 és N4 közötti

...

N1 és N2 és az N [BK No. Interf+2] és N [BK No. Interf+3] közötti mondatokra (elmozdulásokra) végzi el. Az AAL paraméter állásától függően ezután az N1 mondatban hibát jelez, vagy megpróbál javítani.

Ez akkor használható, ha egy üregbe megy a szerszám és meg kell vizsgálni, hogy az adott átmérővel elfér-e abban.

Ha a mellékelt ábrán látható üreget akarjuk ellenőriztetni a vezérlővel, az N1404 BK No. Interf=7 értéket kell beállítani.



15.7.7-5 ábra

Tipikus interferenciahibák

Az alábbiakban felsorolunk néhány tipikus esetet, amikor a vezérlő interferenciahibát érzékel.

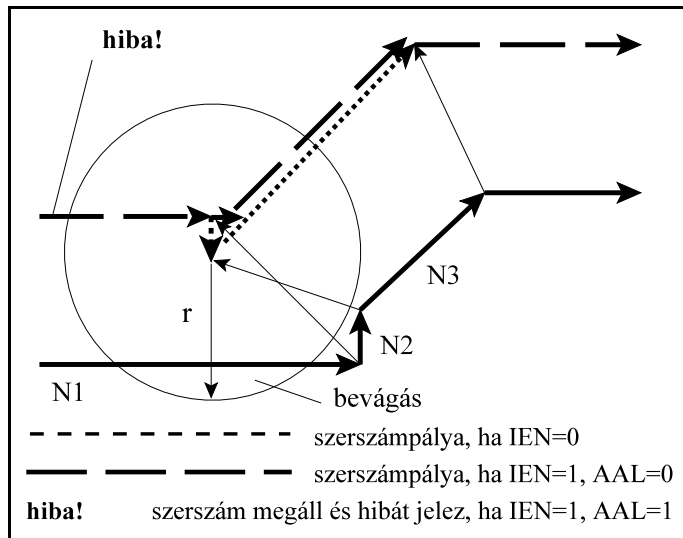
A szerszámsugárnál kisebb lépcső forgácsolása. A vezérlő

IEN=0 esetén belevág a darabba.

IEN=1 esetén:

ha AAL=0 az N1 és az N3 mondat között metszéspontot számolva kerüli el a bevágást,

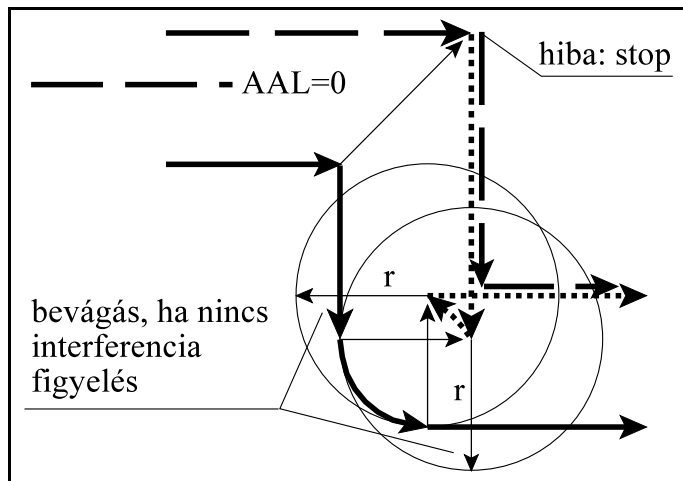
ha AAL=1, 2049 Interferenciahiba üzenetet ad, mert belevágna a darabba.



15.7.7-6 ábra

A szerszámsugárnál kisebb rádiuszú belső sarok megmunkálása. A vezérlő AAL=1 esetén 2049 Interferenciahiba üzenetet ad.

AAL=0 esetén a kört kihagyva a két egyenes között metszéspontot számolva javítja a hibát.

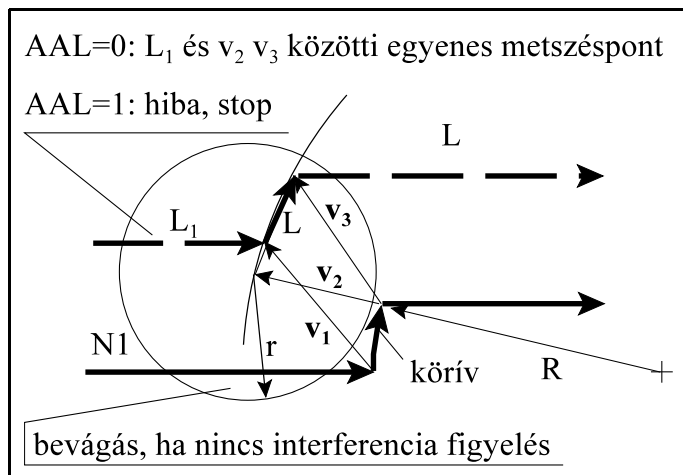


15.7.7-7 ábra

A szerszámsugárnál kisebb lépcső forgácsolása körív mentén.

Ha AAL=0, a vezérlő az L_1 egyenes és a v_1 és a v_3 vektort összekötő egyenesek között metszéspontot számol, hogy elkerülje a bevágást.

Ha AAL=1, akkor 2049 Interferenciahiba üzenetet ad, és megáll az előző mondatban.

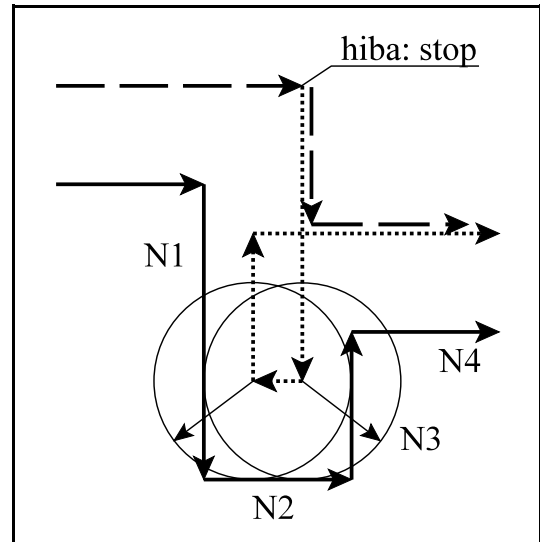


15.7.7-8 ábra

A mellékelt rajzon látható pálya esetén az N2 mondatban a korrigált pályán az elmozdulás ellentétes a programozottal.

A vezérlő nem tud javítani, ha az N1404 BK No. Interf paraméter értéke 0, mert az N1 és az N3 mondat között nincs metszéspont, ezért AAL=0 és AAL=1 esetben is hibát jelez.

Ha az **N1404 BK No. Interf paraméter** értéke **nagyobb, mint 0**, a vezérlő továbbolvassa a mondatokat, ezért az **interferenciahibát** az N1 és az N4 közötti metszésponttal **javítja**, kihagyva az N2, N3 mondatokat.



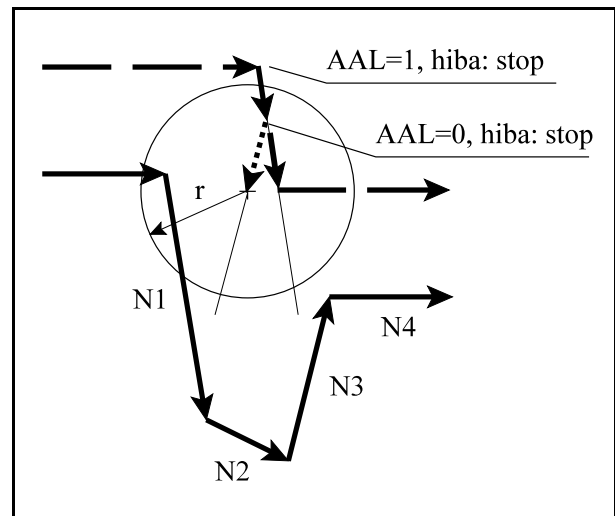
15.7.7-9 ábra

A mellékelt rajzon látható pálya esetén az N2 mondatban a korrigált pályán az elmozdulás ellentétes lenne a programozottal.

AAL=1 esetben az N1 mondatkezdőpontján hibát jelez.

AAL=0 esetben metszéspontot számol az N1 és az N3 mondat között és a metszéspontig halad. Ha tovább menne a metszésponttól az N3 mondatdal ellentétes irányú mozgás alakulna ki, a korrigált pályán, ezért hibát jelez, ha az N1404 BK No. Interf paraméter értéke 0.

Ha az **N1404 BK No. Interf paraméter** értéke **nagyobb, mint 0**, a vezérlő továbbolvassa a mondatokat, ezért az **interferenciahibát** az N1 és az N4 közötti metszésponttal **javítja**, kihagyva az N2, N3 mondatokat.



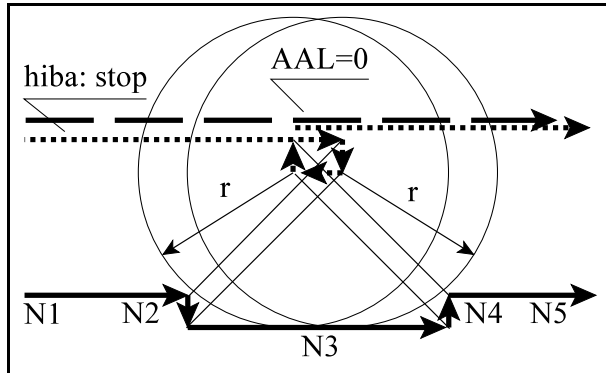
15.7.7-10 ábra

Interferenciahibát jelez, de nem vágna bele a darabba

Vannak esetek, amikor valójában nem vágna bele a szerszám a darabba, de az interferenciavizsgálat hibát jelez.

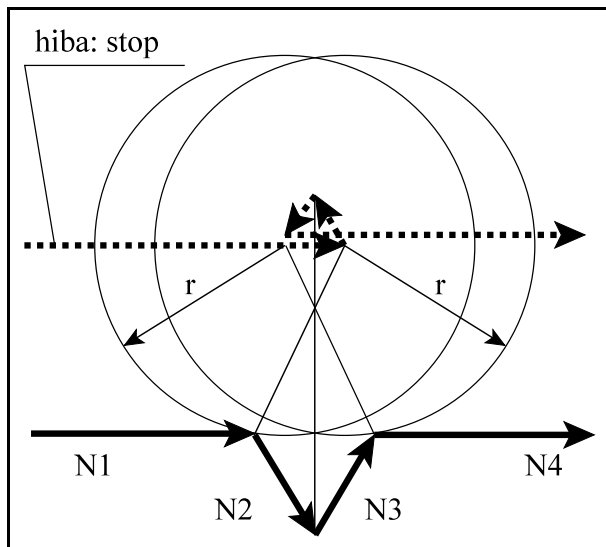
Ha a sugárkorrekciónál kisebb süllyesztést munkálunk meg, a valóságban esetleg belevágás nem történik, mint az az ábrán is látható, de a vezérlő 2049 Interferenciahiba üzenetet ad AAL=1 esetén, mert az N3 mondatban a korrigált pályán az elmozdulás iránya ellentétes a programozottal.

AAL=0 esetén az ábrán látható módon az N2, N3, N4 mondatokat kihagyva, az N1 és az N5 mondatokat összekötve folytatja a megmunkálást.



15.7.7-11 ábra

A mellékelt ábrán látható példában szintén interferencia hibát jelez, mivel az N2 mondatban a korrigált pálya elmozdulása ellentétes a programozottal.



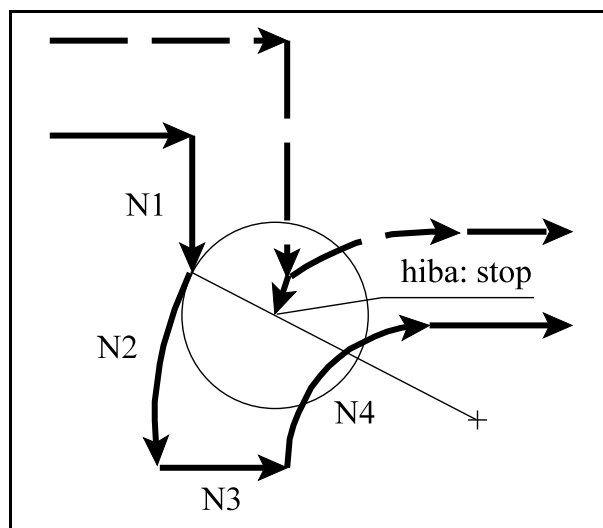
15.7.7-12 ábra

Az interferenciafigyelés ellenére belevág a darabba

Vannak olyan, a pálya geometriájából adódó esetek, amikor az interferenciafigyelés ellenére belevág a darabba. Az alábbiakban felsorolunk néhány esetet.

A mellékelt ábrán látható esetben az N3 mondatban a programozottal ellentétes irányú elmozdulás adódik, ezért az N2 mondat kezdőpontján interferenciahibát jelez és megál, ha az N1404 BK No. Interf paraméter értéke 0. Az N4 mondat geometriájából (G2) adódóan belevág a darabba.

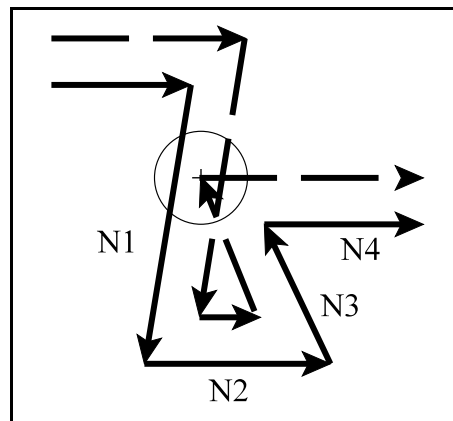
Ha az N1404 BK No. Interf paraméter értéke nagyobb, mint 0, a vezérlő továbbolvassa a mondatokat, ezért az **interferenciahibát** az N1 és az N4 közötti metszésponttal **javítja**, kihagyva az N2, N3 mondatokat, ha AAL=0, egyébként hibát jelez az N1 mondat elején.



15.7.7-13 ábra

A mellékelt ábrán látható esetben az N1, N2, N3 mondat egyikén sem alakul ki a programozottal ellentétes elmozdulás, ezért a vezérlő nem jelez interferenciahibát, ha az N1404 BK No. Interf paraméter értéke 0, mégis a pálya geometriájából adódóan belevág a darabba.

Ha az N1404 BK No. Interf paraméter értéke nagyobb, mint 0, a vezérlő továbbolvassa a mondatokat, ezért az **interferenciahibát** az N1 és az N4 közötti metszésponttal **javítja**, kihagyva az N2, N3 mondatokat, ha AAL=0, egyébként hibát jelez.



15.7.7-14 ábra

A mellékelt rajzon látható pálya esetén az N2 mondatban a korrigált pályán az elmozdulás ellentétes lenne a programozottal.

AAAL=1 esetben az N1 mondatkezdőpontján hibát jelez.

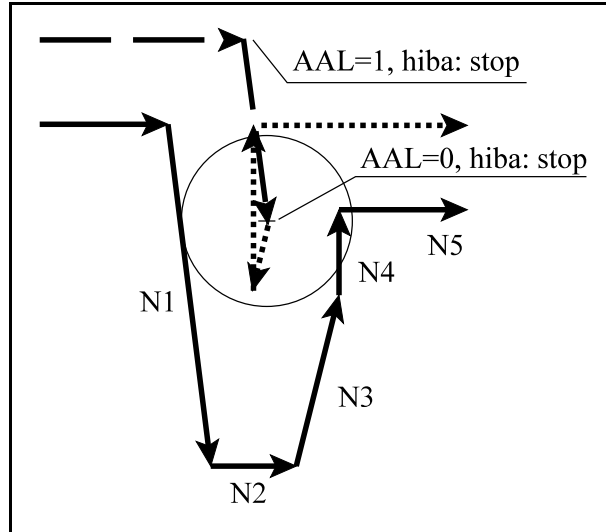
AAAL=0 esetben metszéspontot számol az N1 és az N3 mondat között és a metszéspontig halad. Ha tovább menne a metszésponttól az N3 mondatdal ellentétes irányú mozgás alakulna ki, ezért hibát jelez.

A mondat végpontján azonban már bevágás történik, ha az N1404 BK No. Interf paraméter értéke 0.

Ha az N1404 BK No. Interf paraméter értéke 1 interferenciahibát jelez.

Ha az **N1404 BK No. Interf paraméter értéke**

2 az N1 és az N5 mondatok között metszéspontot számolva javítja a hibát.



15.7.7-15 ábra

16 Különleges transzformációk

16.1 Alakzat elforgatása adott pont körül (G68, G69)

A

G68 p q R

paranccsal egy programozott alakzat a **G17, G18, G19 által kijelölt síkban elforgatható**.

p és q címen

adjuk meg az **elforgatás középpontjának koordinátáit**. Csak a kiválasztott sík p és q koordinátáira írt értékeket értelmezi.

Polárkoordináta adatmegadás bekapcsolt állapotában is az itt beírt **p, q koordinátaadatokat derékszögű koordinátarendszerben** értelmezi. Az elforgatás középpontjának p, q koordinátáit megadhatjuk **abszolút és növekményes** adatként is G90, G91, vagy I operátor használatával. A növekményes adat mindig a programozói (nem elforgatott) utolsó tengelypozíciótól értendő.

Ha p és q valamelyikének, vagy egyikének sem adunk értéket, az elforgatás középpontjának a pillanatnyi tengelypozíciót veszi.

Az alakzat elforgatásának egyenlete G17 esetén az XY síkban:

$$X' = (X - p) \cos R - (Y - q) \sin R + p$$

$$Y' = (X - p) \sin R + (Y - q) \cos R + q$$

Ahol: p, q: a forgatás középpontja,

R: a forgatás szöge,

X, Y a programozott pont koordinátái,

X', Y': az elforgatott pont koordinátái.

R címen

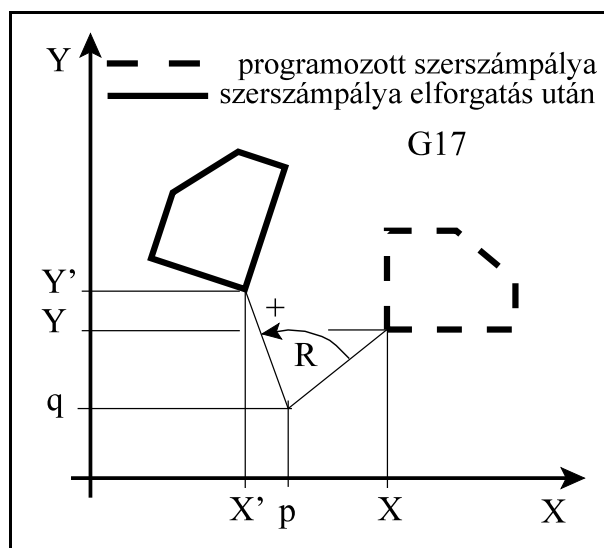
adjuk meg az **elforgatás szögét**. A címre írt **pozitív** érték az **óramutató járásával ellentétes** irányt, a **negatív** érték az **óramutató járásával megegyező** irányt jelent, amit a kiválasztott síkban kell értelmezni.

R-re adott érték **abszolút és inkrementális** is lehet. Ha az elforgatás szögét inkrementálisan adjuk meg, akkor az előzőleg programozott elforgatási szögekhez hozzáadódik R értéke.

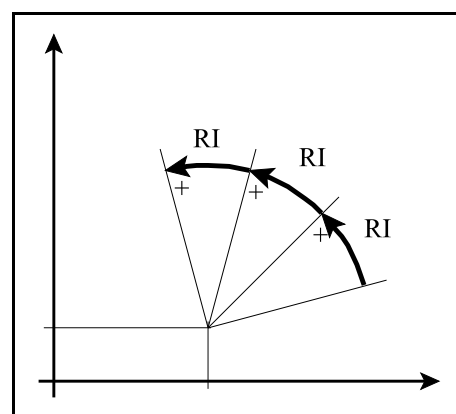
A

G69

parancs az **elforgatást kikapcsolja**. Törli az elforgatás középpontjának koordinátáit, és az elforgatási szöveget is. Ez az utasítás más parancsok mellett is állhat.



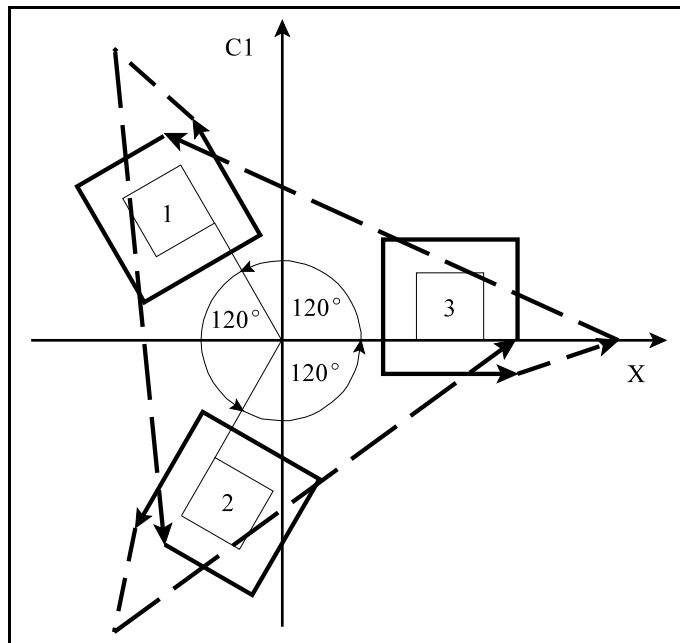
16.1-1 ábra



16.1-2 ábra

Mintapélda:

Plárinterpolációval 3 db. egymáshoz képest 120 fokos lévő négyzetet kell marni egy darab homlokfelületére. A négyzetet az O0001 alprogram írja le. Az alprogram háromszoros hívásával és az alprogram végén történő 120 fokos inkrementális fogratással alakul ki a darab. Az ábrán az 1, 2, 3 számok jelzik a végrehajtási sorrendet.



16.1-3 ábra

Főprogram:

```
G17 G59 G94 G0 X100 Z20 C1=0
M3 S2=500
G12.1
G68 X0 C1=0 R120
M98 P1 L3
G69
G13.1
```

Alprogram:

```
O0001
G42 G0 X60 C1=0
G1 Z-2
C1=10
X40
C1=0
X70
G40 G0 X100 C1=0
Z20
G68 X0 C1=0 RI120
M99
```

16.2 Alakzat léptékezése adott ponthoz viszonyítva (G50, G51)

A léptékezést a G51 kóddal kapcsolhatjuk be. Megadása kétféleképp lehetséges: egy közös, minden hossz tengelyre érvényes arányszámmal, illetve az X, Y, Z fő tengelyekre érvényes különböző arányszámmal.

Mindkét megadási módot a G50 utasítás kapcsolja ki.

Léptékezés minden tengelyre érvényes arányszámmal (P)

A

G51 v P

parancsal egy programozott alakzat *kicsinyíthető, vagy nagyítható*.

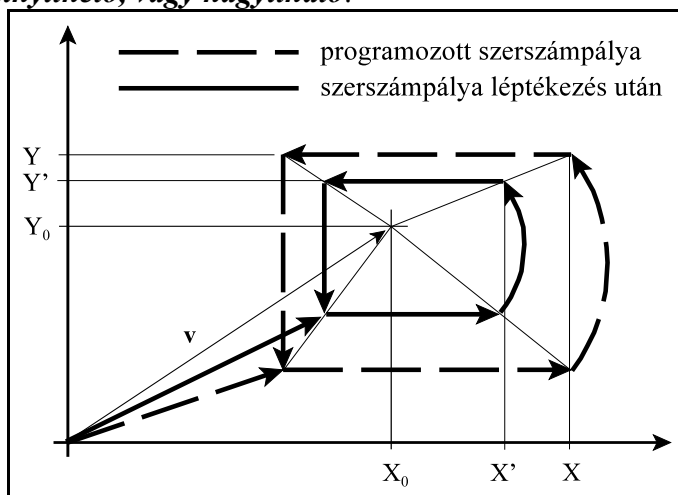
A

v

koordinátákon adhatjuk meg a *léptékezés középpontjának helyzetét*. A felhasználható címek: X, Y, Z alaptengelyek, illetve a párhuzamos tengelyek.

Polárkoordináta adatmegadás bekapcsolt állapotában is az itt beírt v *koordináta-adatokat derékszögű koordinátarendszerben* értelmezi.

A léptékezés középpontjának v koordinátáit megadhatjuk *abszolút és növekményes* adatként is G90, G91, vagy I operátor használatával.



16.2-1 ábra

Ha valamelyik tengelycímnek, vagy egyiknek sem adunk értéket, a léptékezés középpontjának a pillanatnyi tengelypozíciót veszi.

P

címen a *léptékezés arányszámát* állíthatjuk be.

Ha $P < 1$ kicsinyítés, ha $P > 1$ nagyítás történik.

Az alakzat léptékezésének egyenlete az XYZ térben:

$$X' = (X - X_0)P + X_0$$

$$Y' = (Y - Y_0)P + Y_0$$

$$Z' = (Z - Z_0)P + Z_0$$

Ahol: X_0, Y_0, Z_0 : a léptékezés középpontjának v koordinátái,

P: a léptékezés arányszáma,

X, Y, Z: a programozott pont koordinátái.

X', Y', Z' : a léptékezés utáni koordináták.

A

G50

parancs a *léptékezést kikapcsolja*.

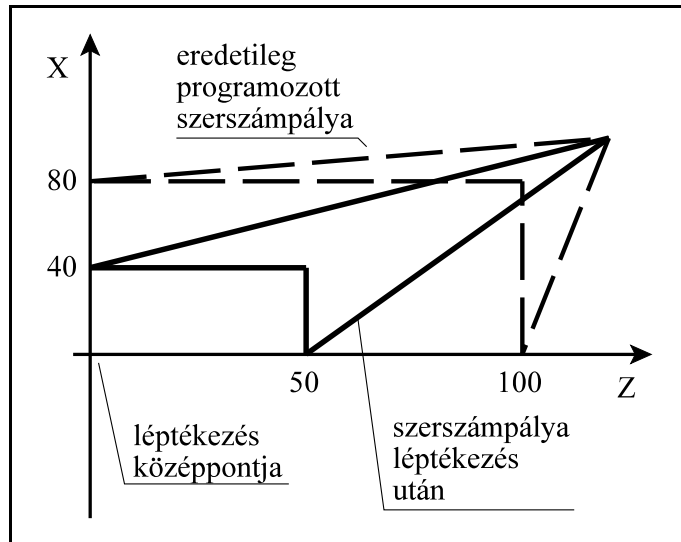
☞ **Megjegyzés:**

A léptékezés arányszáma (P) az alkalmazott sugárkorrekció értékét nem befolyásolja.

Mintapélda:

Ha a darab Z irányú nullpontja a tokmány felfekvő felületén, van egyszerűen esztergálhatunk G51-gyel kisebb, vagy nagyobb darabot.

```
G90 G0 X100 Z120
G51 X0 Z0 P0.5
G1 X0 Z100 F150
X80
Z0
G50
G0 X100 Z120
```



16.2-2 ábra

Léptékezés az X Y Z tengelyen különböző arányszámmal (I J K)

A

G51 X Y Z I J K

paranccsal egy programozott alakzat *kicsinyíthető, vagy nagyítható.*

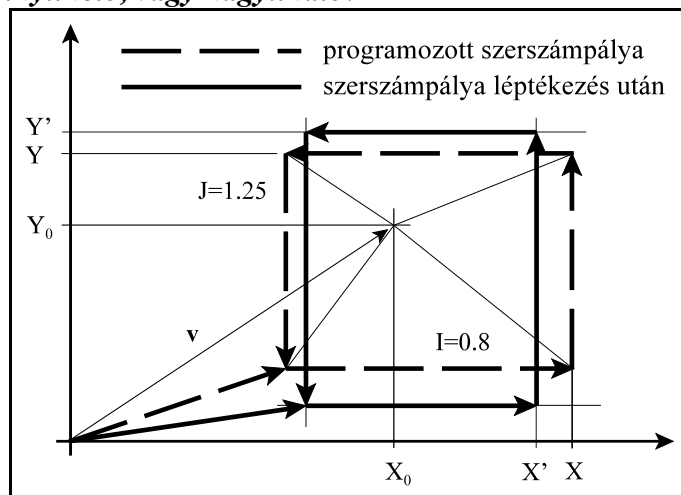
Az

X Y Z

koordinátákon adhatjuk meg a *léptékezés középpontjának helyzetét*. A felhasználható címek: az X, Y, Z fő tengelyek. Polárkoordináta adatmegadás bekapcsolt állapotában is az itt beírt *koordinátaadatokat derékszögű koordinátarendszerben* értelmezi.

A léptékezés középpontjának X Y Z koordinátáit megadhatjuk *abszolút és növekményes* adatként is G90, G91, vagy I operátor használatával.

Ha valamelyik tengelycímnek, vagy egyiknek sem adunk értéket, a léptékezés középpontjának a pillanatnyi tengelypozíciót veszi.



16.2-3 ábra

I J K

címen a *léptékezés I-X, J-Y, K-Z tengelyre érvényes arányszámát* állíthatjuk be.

Ha I J K < 1 kicsinyítés, ha I J K > 1 nagyítás történik.

Az alakzat léptékezésének egyenlete az XYZ térben:

$$X' = (X - X_0)I + X_0$$

$$Y' = (Y - Y_0)J + Y_0$$

$$Z' = (Z - Z_0)K + Z_0$$

Ahol: X_0, Y_0, Z_0 : a léptékezés középpontjának v koordinátái,
I J K: a léptékezés arányszáma,

X, Y, Z: a programozott pont koordinátái.
 X', Y', Z': a léptékezés utáni koordináták.

A

G50

parancs a *léptékezést kikapcsolja*.

☞ *Megjegyzés:*

- A léptékezés arányszámai (I J K) az alkalmazott sugárkorrekció értékét nem befolyásolják.
- I J K értéke lehet negatív szám is. Ekkor az adott tengelyen nem csak léptékezést, hanem tükrözést is végez a vezérlő. Ha I J K értéke -1, az adott tengelyre csak tükrözést végez.
- Ha I J K értéke különböző, a körinterpoláció (G1, G3) és a ,R-en lekerekítés értéke az alábbi példa szerint módosul:

```
G17
G51 X0 Y0 Z0 I0.5 J0.75 K1.25
G1 X100 ,R10 (R=,R*(I+J)/2)
Y100
X10
G3 X0 Y90 R10 (R=R*I)
G1 Y0
G50
```

A lekerekítés sugarát I és J átlagával szorozza, a G3 mondatban a körsugarat I-vel.

16.3 Alakzat tükrözése egy, vagy több egyenesre (G50.1, G51.1)

A

G51.1 v

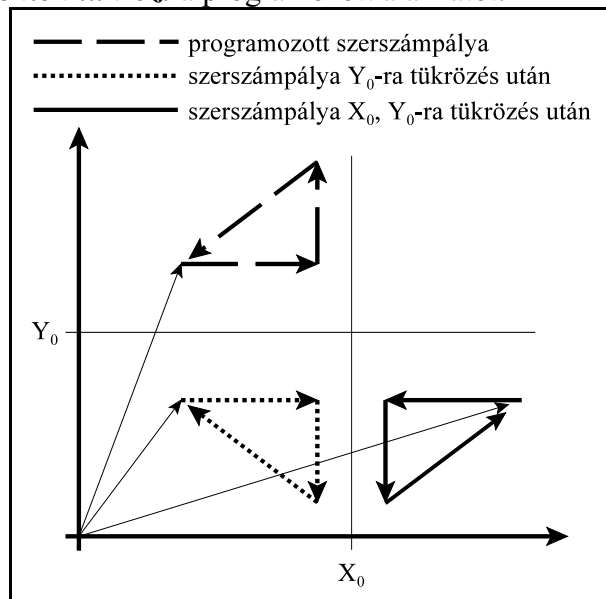
paranccsal a v-ben *kiválasztott koordináták mentén tükrözi* a programozott alakzatot.

v

koordinátákon adhatjuk meg azt, hogy *mely tengellyel párhuzamos egyenesre, vagy egyenesekre akarunk tükrözni*.

v lehet: X, Y, Z, illetve további tengelycímek. Ha az X tengellyel párhuzamos egyenesre akarunk tükrözni Y címen, ha az Y tengellyel párhuzamos egyenesre X címen adjuk meg az egyenes helyzetét, stb.

Polárkoordináta adatmegadás bekapcsolt állapotában is az itt beírt v *koordinátaadatokat derékszögű koordinátarendszerben értelmezi*. A tükrözés tengelyeinek v koordinátáit megadhatjuk *abszolút és növekményes adatként* is G90, G91, vagy I operátor használatával. Ha valamelyik *tengelycímet, nem adunk értéket arra nem végez tükrözést*.



16.3-1 ábra

Az alakzat tükrözésének egyenlete az XYZ térben:

$$X' = -(X - X_0) + X_0$$

$$Y' = -(Y - Y_0) + Y_0$$

$$Z' = -(Z - Z_0) + Z_0$$

Ahol: X_0, Y_0, Z_0 : a tükrözés középpontjának v koordinátái,

X, Y, Z : a programozott pont koordinátái.

X', Y', Z' : a tükrözés utáni koordináták.

A páratlan számú tengelyre való tükrözés megváltoztatja az alakzat körüljárásának irányát. Ennek következménye, hogy megváltoznak a körirányok (G2-G3) és a szerszégűgár korrekció iránya (G41-G42) illetve az elforgatás szöge (G68 R) is, amit a vezérlő automatikusan figyelembe vesz.

A

G50.1 v

parancs a v-n *megadott koordinátatengely(ek)en kikapcsolja a tükrözést*. A v koordinátákra tet-szőleges adat írható, hatása csak a kikapcsolás tényét rögzíti.

Tükrözés be-, illetve kikapcsolásakor nem állhat fenn sem elforgatási (G68), sem léptékezési (G51) állapot. Ellenkező esetben 2001 Tükrözést be-/kikapcsolni nem lehet G51, vagy G68 állapotban! hibajelzést ad.

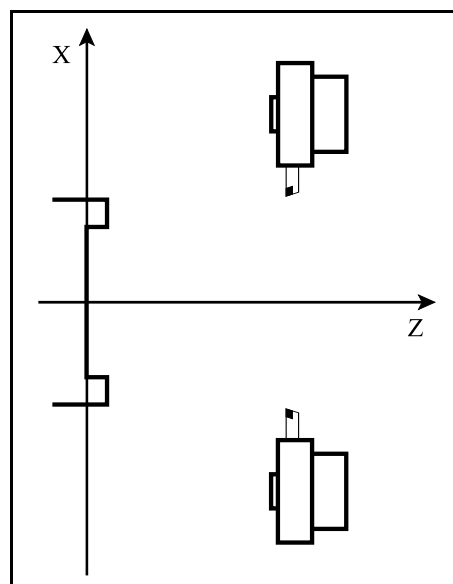
Tükrözés kettős szerszámtartó esetén

Ha az esztergán két szerszámcserélő van, egyik a felső, az X+, a másik az alsó, az X- térfélen, a Z tengelyre való tükrözést

G51.1 X0

lehet használni az alsó szerszámcserélőben lévő szerszámokkal történő megmunkálás során.

Ha az alsó, az X- térfélen lévő szerszámcserélő szerszámaival akarunk dolgozni, megírhatjuk a programot az X+ térfélre és a programrésztletet tükrözzük a Z tengelyre.

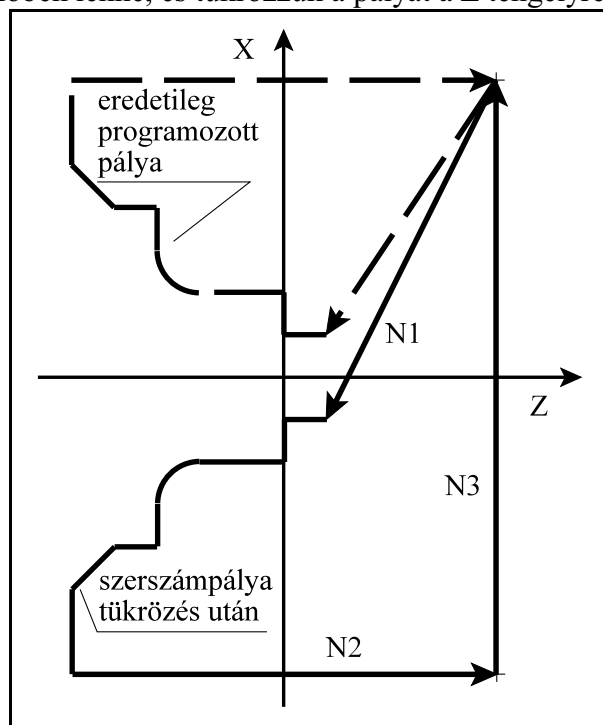


16.3-2 ábra

Mintapélda:

Végezzük a darab simítását az alsó szerszámcserélőben lévő T2-es szerszámmal. A programot úgy írjuk meg, mintha a szerszám a felső cserélőben lenne, és tükrözzük a pályát a Z tengelyre.

```
G0 X140 Z50
T202 M3 (simítókés, Q2)
G51.1 X0
N1 G0 G42 X20 Z10
Z0
X40
Z-20
G2 X60 Z-30 R10
G1 X80
Z-40
X100 Z-50
X140
N2 G40 G0 Z50
G50.1 X0
N3 X140
```



16.3-3 ábra

16.4 A különleges transzformációk programozási szabályai

A **G68 elforgatás** és a **G51 léptékezés** utasítások *sorrendje tetszőleges* lehet.

Vigyázni kell viszont arra, hogy ha először elforgatunk utána léptékezünk, akkor **a léptékezés középpontjának koordinátáira is érvényes az elforgatási parancs**. Ha viszont először léptékezünk és utána forgatunk el, **az elforgatás középpontjának koordinátáira a léptékezési parancs** lesz érvényes.

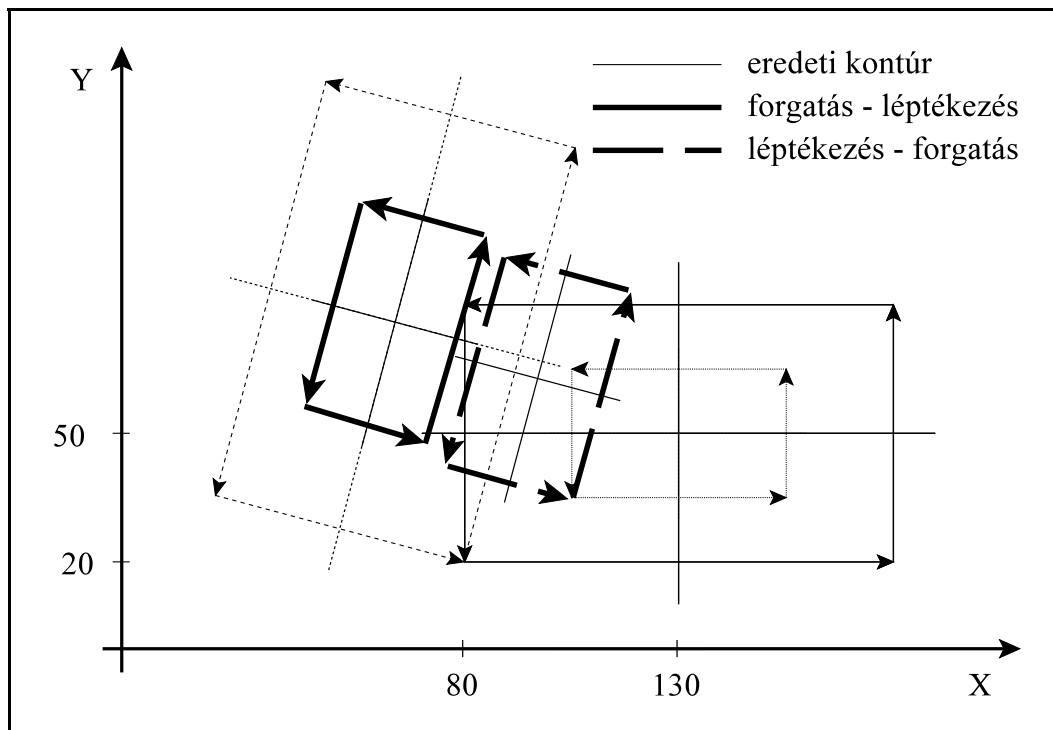
A két művelet bekapcsolási és kikapcsolási parancsainak viszont egymásba kell skatulyázódnuk, egymást nem lapolhatják át:

Elforgatás–léptékezés

```
N1 G90 G17 G0 X0 Y0
N2 G68 X80 Y20 R75
N3 G51 X130 Y50 P0.5
N4 X180 Y20
N5 G1 Y80 F200
N6 X80
N7 Y20
N8 X180
N9 G50
N10 G69 G0 X0 Y0
```

Léptékezés–elforgatás

```
N1 G90 G17 G0 X0 Y0
N2 G51 X130 Y50 P0.5
N3 G68 X80 Y20 R75
N4 X180 Y20
N5 G1 Y80 F200
N6 X80
N7 Y20
N8 X180
N9 G69
N10 G50 G0 X0 Y0
```



16.4-1 ábra

Az ábrából látható, hogy nem mindegy milyen sorrendben alkalmazzuk a különböző transzformációkat.

Más a helyzet a tükrözéssel. Tükrözést bekapcsolni csak G50 és G69 állapotban lehet, azaz ha nincs sem léptékezési sem elforgatási parancsállapot.

A tükrözés bekapcsolt állapotában viszont mind a léptékezés, mind az elforgatás bekapcsolható. A tükrözésre is érvényes, hogy sem a léptékezési, sem az elforgatási parancsokkal nem lapolódhat át, tehát először a megfelelő sorrendben az elforgatást és a léptékezést kell kikapcsolni, és csak utána a tükrözést.

G51.1 ...	(tükrözés bekapcsolása)
G51 ...	(léptékezés bekapcsolása)
G68 ...	(elforgatás bekapcsolása)
...	
G69 ...	(elforgatás kikapcsolása)
G50 ...	(léptékezés kikapcsolása)
G50.1 ...	(tükrözés kikapcsolása)

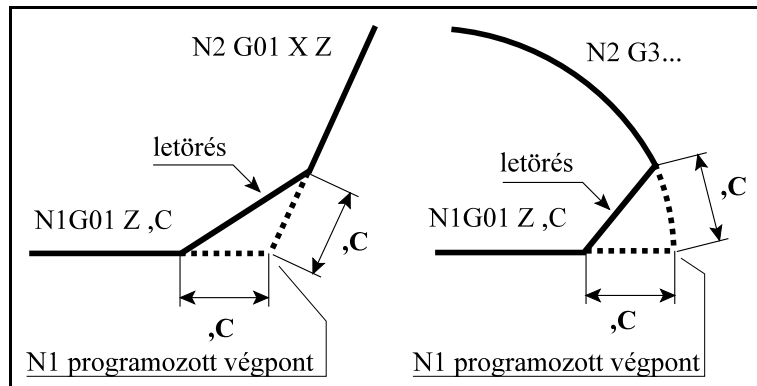
17 Automatikus geometriai számítások

17.1 Letörés és lekerekítés programozása

Két egyenes interpolációt (G01), vagy körinterpolációt (G02, G03) tartalmazó mondat közé a vezérlő automatikusan *letörést*, vagy *lekerekítést* tud beiktatni *a kiválasztott síkban*.

A

,C
(vessző és C) *címen megadott értéknek megfelelő hosszúságú egyenlő szárú letörést* iktat a ,C címet tartalmazó mondat végpontja és a következő mondat kezdőpontja közé.

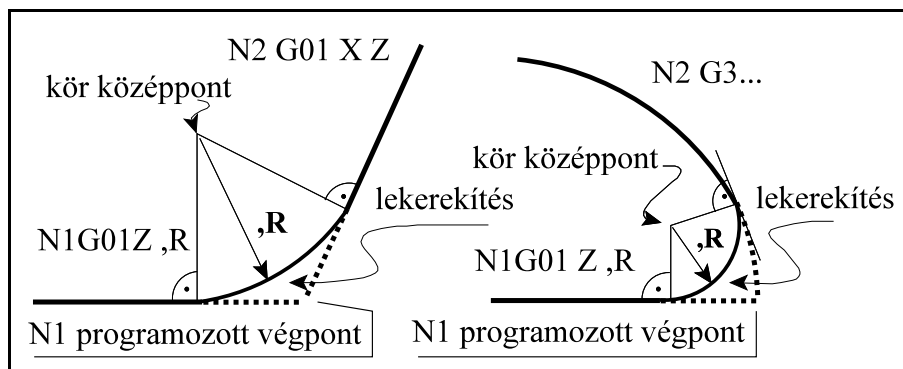


17.1-1 ábra

Például:

```
N1 G18 G1 G91 Z30 ,C10
N2 X80 Z10
```

A ,C címen megadott érték azt mutatja meg, hogy a két egymást követő mondat feltételezett metszéspontjától mekkora távolságra kezdődik illetve fejeződik be a letörés. Letörést körök, vagy kör és egyenes közé is be lehet iktatni. Ekkor a ,C érték a metszésponttól húzott húr hossza.



17.1-2 ábra

A

,R
(vessző és R) *címen megadott értéknek megfelelő sugarú lekerekítést* iktat a ,R címet tartalmazó mondat végpontja és a következő mondat kezdőpontja közé.

Például:

```
N1 G18 G91 G01 Z30 ,R8
N2 G03 X60 Z-30 R30
```

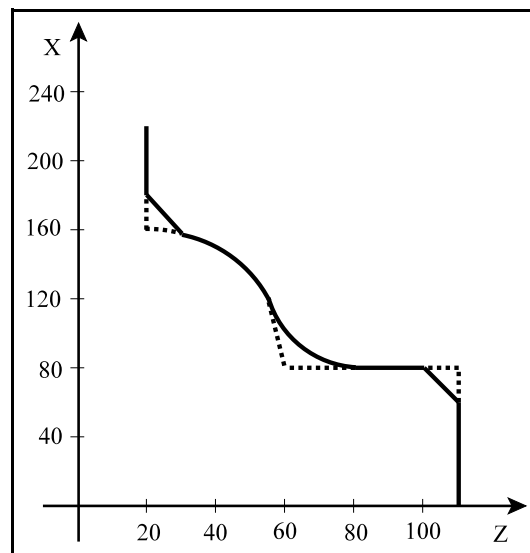
A ,R sugarú körívet úgy iktatja a két mondat közé, hogy a kör mindkét pályaelemhez érintőlegesen símuljon.

Egymást követő, több mondat végére is írható letörést, vagy lekerekítést tartalmazó utasítás, mint az alábbi példa mutatja:

```

...
G18 G1 X80 ,C10
Z60 ,R22
G3 X160 Z20 R40 ,C10
G1 X220
...

```



17.1-3 ábra

☞ Megjegyzés:

- Letörés, vagy lekerekítés csak a kiválasztott síkban (G17, G18, G19) fekvő elemek között programozható, ellenkező esetben a vezérlő “Illegális letörés vagy lekerekítés” üzenetet ad.
- Ha a letörés szárhossza, vagy a lekerekítés sugara olyan nagy, hogy nem illeszthető a programozott mondatokhoz, a vezérlő “Letörés vagy lekerekítés túl nagy” hibajelzést ad.
- Ha egy mondatba programozunk ,C-t és ,R-t a vezérlő 2082 “Vagy letörés vagy lekerekítés” hibüzenetet ad.
- Mondatonkénti üzemben a vezérlő a letörés, vagy a lekerekítés végrehajtása után áll meg, és vesz fel stop állapotot.

17.2 Egyenes megadása irányszögével

Egyenest a G17, G18, G19 utasítás által meghatározott síkban *meg lehet adni* a kiválasztott sík *egyik koordinátájával és ,A címen megadva az egyenes irányszögével.*

$$G17 \left\{ \begin{array}{l} G0 \\ G1 \end{array} \right\} \left\{ \begin{array}{l} X_{p,A} \\ Y_{p,A} \end{array} \right\} qF$$

$$G18 \left\{ \begin{array}{l} G0 \\ G1 \end{array} \right\} \left\{ \begin{array}{l} Z_{p,A} \\ X_{p,A} \end{array} \right\} qF$$

$$G19 \left\{ \begin{array}{l} G0 \\ G1 \end{array} \right\} \left\{ \begin{array}{l} Y_{p,A} \\ Z_{p,A} \end{array} \right\} qF$$

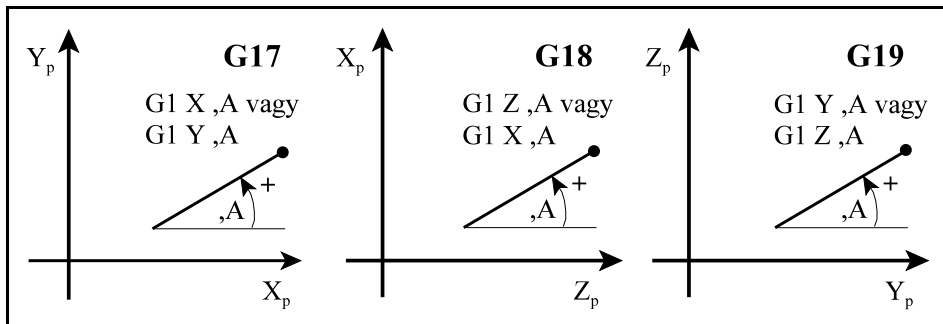
A fenti képletekben X_p, Y_p, Z_p az X, Y, Z , vagy a velük párhuzamos tengely mentén az egyenes

végponti koordinátáját jelöli, q tetszőleges egy, vagy több, a kiválasztott síkon kívül eső tengelyt jelöl.

A ,A címen történő megadás G0 és G1 kód mellett is használható.

,A szög a kiválasztott sík első tengelyétől számítódik, és a pozitív irány az óramutató járásával ellentétes.

,A értéke lehet pozitív és negatív is, valamint lehet 360° -nál nagyobb, illetve -360° -nál kisebb érték is.



17.2-1 ábra

Például:

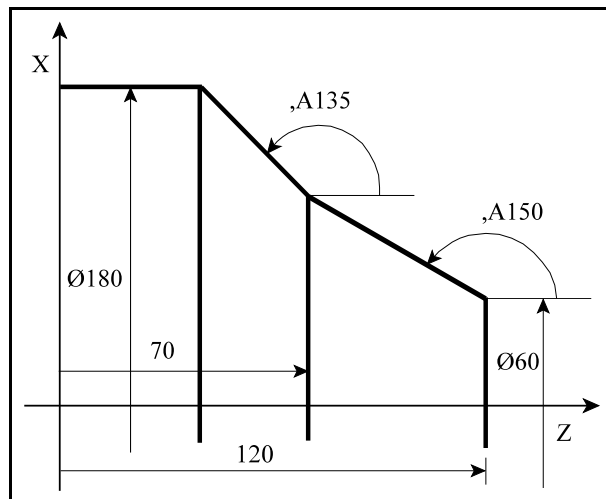
(G18 G90) G1 X60 Z120 ...

Z70 ,A150

(ez a megadás ekvivalens az X117.735 Z70 megadással)

X180 ,A135

(ez a megadás ekvivalens az X180 Z38.868 megadással)



17.2-2 ábra

☞ **Megjegyzés:**

– Egy mondatban megadható egyenes irányszögével és letörés, vagy lekerekítés is. Például:

Z100 ,A30 ,C5

X100 ,A120 ,R10

Z-100 ,A210

– ,A címen történő irányszög megadása használható fűróciklusokban is. Ekkor a kiválasztott síkban történő pozicionálás végrehajtásakor veszi figyelembe a fent leírt módon. Például

a

G81 G91 X100 ,A30 R-2 Z-25

mondat ekvivalens az alábbi mondattal:

G81 G91 X100 Y57.735 R-2 Z-25

17.3 Síkbeli metszéspontszámítások

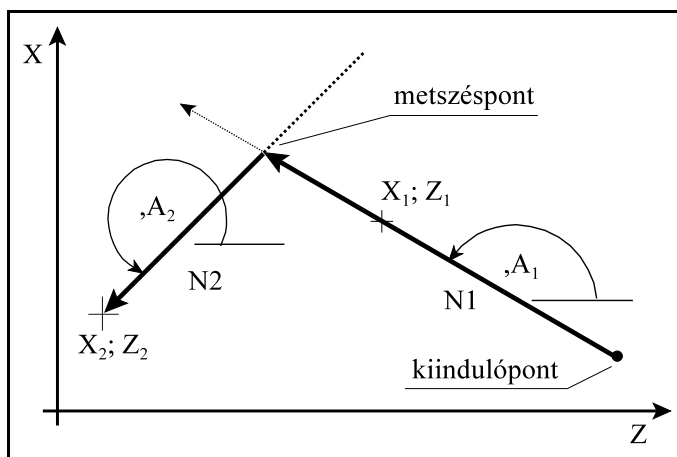
Az itt közölt metszéspontszámításokat a vezérlő csak **a szerszámsugár korrekció számítás bekapcsolt állapotában (G41, vagy G42)** végzi el. Ha esetleg a programban nem akarunk szerszámsugár korrekciót figyelembe venni, akkor is kapcsoljuk be azt és a megfelelő szerszámsugár korrekció értékét töröljük.

17.3.1 Két egyenes metszéspontja

Ha két, egymást követő, egyenes interpolációt végző mondat közül a **második egyenest** úgy adjuk meg, hogy **megadjuk mindkét, a kijelölt síkba eső tengelyen az egyenes végpontját, és megadjuk az egyenes irányszögét is**, a vezérlő kiszámítja az első mondatban kijelölt egyenes és a második mondatban megadott egyenes metszéspontját.

A második mondatban így megadott egyenest **túlhátrózott egyenesnek** nevezük a továbbiakban.

Az **első mondat végpontja**, illetve a **második mondat kezdőpontja a kiszámított metszéspont** lesz.



17.3.1-1 ábra

G17 G41 (G42)

N1 G1 ,A₁ vagy

X₁ Y₁

N2 G1G90 X₂ Y₂ ,A₂

G18 G41 (G42)

N1 G1 ,A₁ vagy

X₁ Z₁

N2 G1G90 X₂ Z₂ ,A₂

G19 G41 (G42)

N1 G1 ,A₁ vagy

Y₁ Z₁

N2 G1G90 Y₂ Z₂ ,A₂

A **metszéspontot mindig a G17, G18, G19 által kijelölt síkban** számítja ki.

Az **első mondatot** (N1) vagy **csak irányszögével** (,A₁) adjuk meg, és ebben az esetben a kiindulópontból a megfelelő irányszögben húz egy egyenest a metszéspontig, **vagy az egyenes egy tetszőleges**, a kiindulóponttól különböző **pontját adjuk meg** (X₁, Y₁; X₁, Z₁; vagy Y₁, Z₁) és ekkor a két ponton áthaladó egyenessel számítja a metszéspontot.

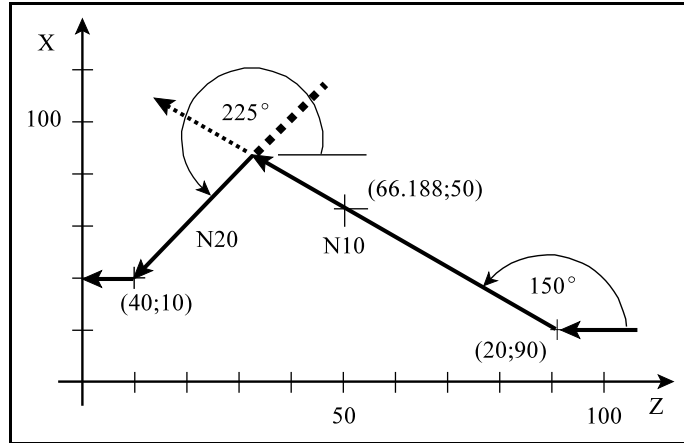
A **második mondatban** (N2) megadott **koordinátákat** mindig **abszolút** (G90) adatként értelmezi a vezérlő.

Például:

```
(G18) G90 G41 ...
G0 X20 Z90
N10 G1 ,A150
N20 X40 Z10 ,A225
G0 Z0
...
```

Az N10 mondatot megadhatjuk az egyenes egy pontjának koordinátaival is:

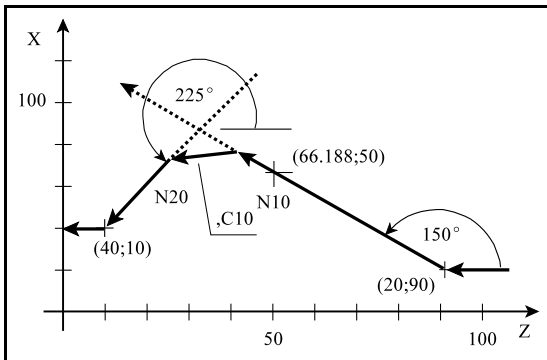
```
(G18) G90 G41 ...
G0 X20 Z90
N10 G1 X66.188 Z50
N20 X40 Z10 ,A225
G0 X0 Y20
...
```



17.3.1-2 ábra

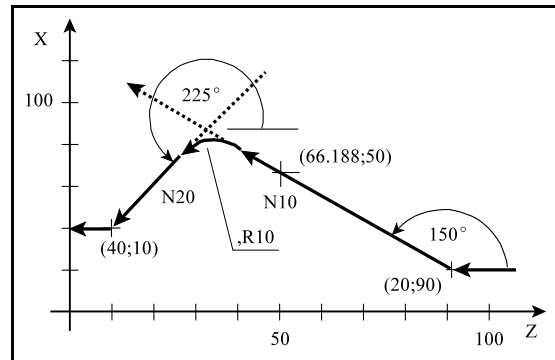
Figyeljük meg, hogy ebben az esetben az N10 mondatban megadott X, Z koordinátát (X66.18 Z50) nem végpontnak tekinti a vezérlő, hanem csak az egyenes kezdőpontját a megadott ponttal összekötő átmenő pontnak.

A metszéspontszámítást kombinálhatjuk letörés, vagy lekerekítés megadásával is. Például:



17.3.1-3 ábra

```
(G18) G90 G41 ...
G0 X20 Z90
N10 G1 X66.188 Z50 ,C10
N20 X40 Z10 ,A225
G0 X0 Y20
...
```



17.3.1-4 ábra

```
(G18) G90 G41 ...
G0 X20 Z90
N10 G1 X66.188 Z50 ,R10
N20 X40 Z10 ,A225
G0 X0 Y20
...
```

A fenti példákban a kiszámított metszésponttól méri vissza a letörés hosszát, illetve a kiszámított metszésponthoz igazítja a lekerekítést.

17.3.2 Egyenes és kör metszéspontja

Ha egyenes mondatot követően körmondatot úgy adunk meg, hogy **a körnek megadjuk a végponti és középponti koordinátáját és a kör sugarát is**, vagyis **a kört túlhatározzuk**, a vezérlő az egyenes és kör között metszéspontot számol. Az **első mondat végpontja**, illetve a **második mondat kezdőpontja a kiszámított metszéspont lesz**.

G17 G41 (G42)

N1 G1 ,A vagy

 $X_1 Y_1$ N2 G2 (G3) G90 $X_2 Y_2 I J$

R Q

G18 G41 (G42)

N1 G1 ,A vagy

 $X_1 Z_1$ N2 G2 (G3) G90 $X_2 Z_2 I K$

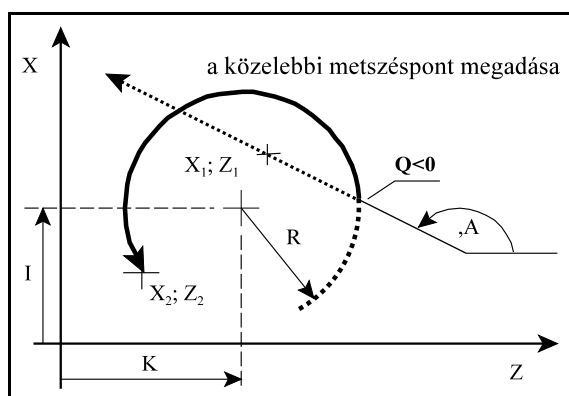
R Q

G19 G41 (G42)

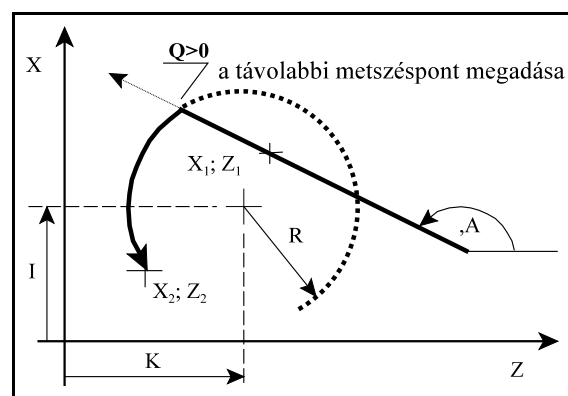
N1 G1 ,A vagy

 $Y_1 Z_1$ N2 G2 (G3) G90 $Y_2 Z_2 J K$

R Q



17.3.2-1 ábra



17.3.2-2 ábra

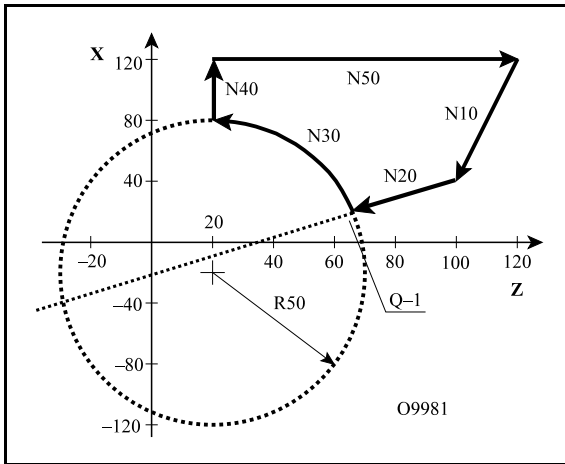
A **metszéspontot mindig a G17, G18, G19 által kijelölt síkban** számítja ki.

Az **első mondatot** (N1) vagy **csak irányszögével** (,A) adjuk meg, és ebben az esetben a kiindulópontból a megfelelő irányszögben húz egy egyenest a metszéspontig, vagy **az egyenes egy tetszőleges**, a kiindulóponttól különböző **pontját adjuk meg** (X_1, Y_1 ; X_1, Z_1 ; vagy Y_1, Z_1) és ekkor a két ponton áthaladó egyenessel számítja a metszéspontot.

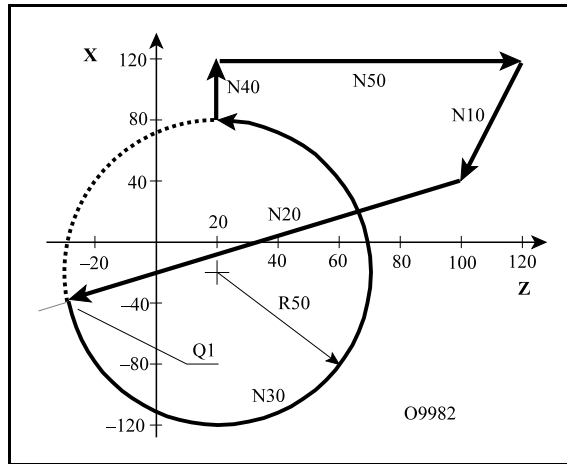
A **második mondatban** (N2) **megadott koordinátákat**, így a **kör középpontját** meghatározó **I, J, K koordinátákat is mindig abszolút (G90) adatként** értelmezi a vezérlő.

Azt, hogy a kiadódó két metszéspont közül melyiket számolja ki a vezérlő a **Q címen** lehet megadni. **Ha a cím értéke kisebb, mint nulla ($Q < 0$) az egyenes irányába eső közelebbi**, ha a cím értéke nagyobb, mint nulla ($Q > 0$) **az egyenes irányába eső távolabbi metszéspontot számolja ki**. Az egyenesen való haladás irányát az irányszög jelöli ki.

Nézzük a következő példát:



17.3.2-3 ábra



17.3.2-4 ábra

```
O9981
N10 (G18) G42 G0 X40 Z100
N20 G1 X-40 Z-30
N30 G3 X80 Z20 I-10K20 R50 Q-1
N40 G40 G0 X120
N50 Z120
```

```
O9982
N10 (G18) G42 G0 X40 Z100
N20 G1 X-40 Z-30
N30 G3 X80 Z20 I-10 K20 R50 Q1
N40 G40 G0 X120
N50 Z120
```

Az N30 G3 körmondat túlhatározott, mivel a középpont koordináták (I-10 K20 abszolút értékben), és a körsugár (R50) is meg van adva, a vezérlés az N20 mondatban megadott egyenes és az N30 mondatban megadott kör metszéspontját számolja.

Az O9981 programban az egyenes irányába eső közelebbi metszéspontot számítja ki, mert az N30 körmondatban Q-1-et programoztunk.

Az O9982 programban viszont az egyenes irányába eső távolabbi metszéspontot számítja ki, mert az N30 körmondatban Q1-et adtunk meg.

Az egyenes - kör metszéspontszámítást kombinálhatjuk letörés, vagy lekerekítés megadásával is.

Például:

```
N10 (G18) G42 G0 X40 Z100 S200 M3
N20 G1 X-40 Z-30 ,R15
N30 G3 X80 Z20 I-10 K20 R50 Q-1
N40 G40 G0 X120
N50 Z120
```

A vezérlő az N20 és N30 mondat metszéspontját kiszámolja és a metszésponthoz egy 15 mm sugarú lekerekítést illeszt az N20 mondatban megadott ,R15 hatására.

17.3.3 Kör és egyenes metszéspontja

Ha körmondatot követően egyenes mondatot úgy adunk meg, hogy *az egyenest túlhatározzuk*, azaz *megadjuk az egyenes végponti koordinátáját és az irányszögét is*, a vezérlő a kör és az egyenes között metszéspontot számol. *Az első mondat végpontja, illetve a második mondat kezdőpontja a kiszámított metszéspont lesz.*

G17 G41 (G42)

N1 G2 (G3) X₁ Y₁ I J

vagy R

N2 G1 G90 X₂ Y₂ ,A Q

G18 G41 (G42)

N1 G2 (G3) X₁ Z₁ I K

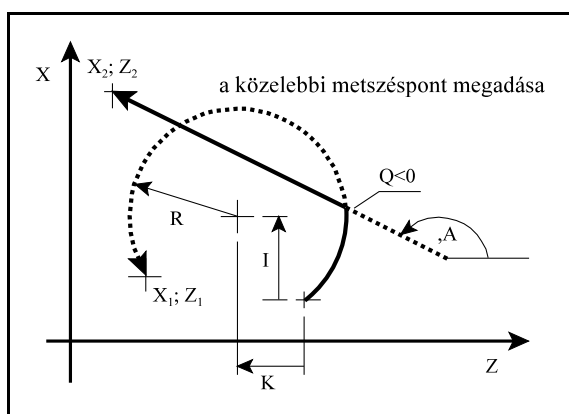
vagy R

N2 G1 G90 X₂ Z₂ ,A Q

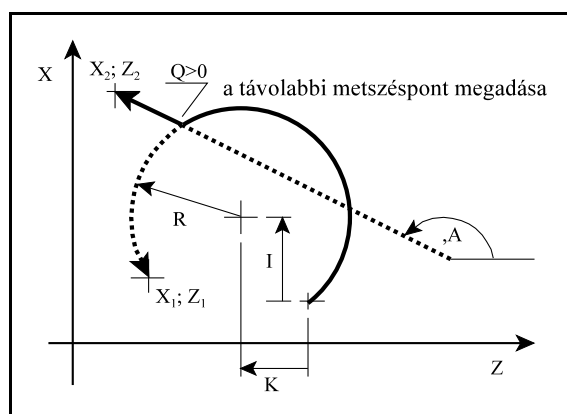
G19 G41 (G42)

N1 G2 (G3) Y₁ Z₁ J K

vagy R

N2 G1 G90 Y₂ Z₂ ,A Q

17.3.3-1 ábra



17.3.3-2 ábra

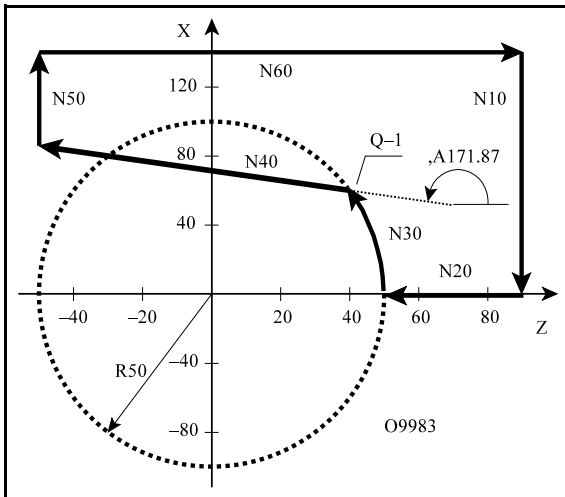
A metszéspontot mindig a G17, G18, G19 által kijelölt síkban számítja ki.

Az első mondatot (N1), vagyis *a kört egy tetszőleges pontjával* (X₁, Y₁; X₁, Z₁; vagy Y₁, Z₁) és *a középponti koordinátájával* (I J; I K; vagy J K) adjuk meg, *vagy* a középponti koordináta helyett *megadhatjuk a kör sugarát* (R) is.

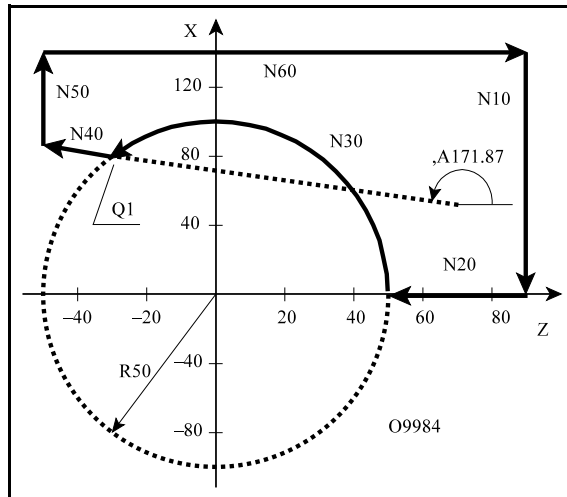
A második mondatban (N2) *az egyenest túlhatározzuk*, vagyis *megadjuk az egyenes végponti koordinátáit* (X₂ Y₂; X₂ Z₂; vagy Y₂ Z₂) és *az egyenes irányszögét* (,A). Az egyenes *végponti koordinátáit* mindig *abszolút* (G90) adatként értelmezi a vezérlő. Mindig a kiadódó *metszéspontból* a megadott *végpontba* mutató egyenes *vektor irányszögét* kell megadni ,A címen, ellenkező esetben a programozói szándékkal ellentétes mozgások következnek be.

Azt, hogy a kiadódó két metszéspont közül melyiket számolja ki a vezérlő a *Q címen* lehet megadni. *Ha a cím értéke kisebb, mint nulla (Q<0, pl: Q-1) az egyenes irányába eső közelebbi, ha a cím értéke nagyobb, mint nulla (Q>0, pl: Q1) az egyenes irányába eső távolabbi metszéspontot számolja ki. Az egyenesen való haladás irányát az irányszög jelöli ki.*

Nézzük a következő példát:



17.3.3-3 ábra



17.3.3-4 ábra

```
O9983
N10 (G18) G0 X0 Z90
N20 G42 G1 Z50
N30 G3 X0 Z-50 R50
N40 G1 X85.714 Z-50 ,A171.87
Q-1
N50 G40 G0 X140
N60 Z90
```

```
O9984
N10 (G18) G0 X0 Z90 M3 S200
N20 G42 G1 Z50
N30 G3 X0 Z-50 R50
N40 G1 X85.714 Z-50 ,A171.87
Q1
N50 G40 G0 X140
N60 Z90
```

Az N40 egyenes mondat túlhatározott, mert az egyenes végponti koordinátái (X85.714 Z-50) is és irányszöge is (,A171.87) meg van adva. Ezért az előző, N30 mondatban programozott kör X0 Z-50 koordinátáit nem tekinti végponti értékeknek, hanem csak egy pontnak, amin a kör áthalad, és a végpont a kiszámított metszéspont lesz.

Az O9983 számú programban a haladási irány szerinti közelebbi metszéspontot adtuk meg (Q-1), míg az O9984-ben a haladási irány szerinti távolabbit (Q1).

Kör és egyenes metszéspontjának megadását kombinálhatjuk letörés, vagy lekerekítés megadásával. Például:

```
O9983
N10 (G18) G0 X0 Z90 M3 S200
N20 G42 G1 Z50
N30 G3 X0 Z-50 R50 ,R15
N40 G1 X85.714 Z-50 ,A171.87 Q-1
N50 G40 G0 X140
N60 Z90
```

Példánkban az N30 mondatban megadtunk egy 15 mm-es lekerekítést (,R15). A vezérlő kiszámítja az N30 és N40 mondat közötti metszéspontot, és az így kiadódó kontúrhoz beilleszti a programozott lekerekítést.

17.3.4 Két kör metszéspontja

Ha kéz egymást követő körmondatot úgy adunk meg, hogy **a második körnek megadjuk a végponti és középponti koordinátáját és a sugarát is**, vagyis **a második kört túlhatározzuk**, a vezérlő a két kör között metszéspontot számol. **Az első mondat végpontja, illetve a második mondat kezdőpontja a kiszámított metszéspont lesz.**

G17 G41 (G42)

N1 G2 (G3) X₁ Y₁ I₁ J₁
vagy X₁ Y₁ R₁

N2 G2 (G3) G90 X₂ Y₂ I₂
J₂ R₂ Q

G18 G41 (G42)

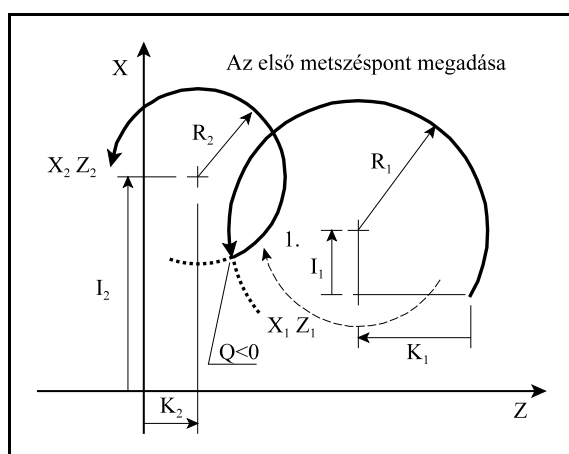
N1 G2 (G3) X₁ Z₁ I₁ K₁
vagy X₁ Z₁ R₁

N2 G2 (G3) G90 X₂ Z₂ I₂
K₂ R₂ Q

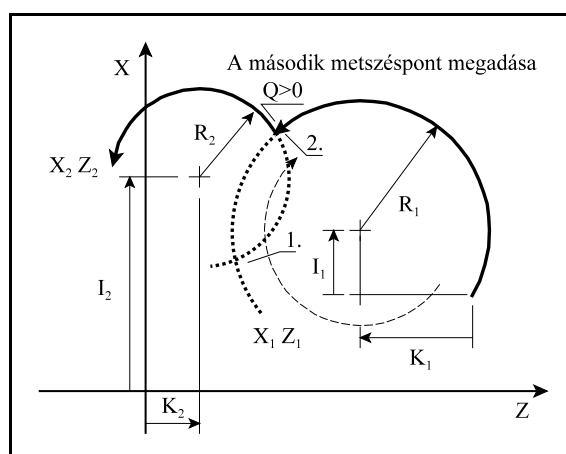
G19 G41 (G42)

N1 G2 (G3) Y₁ Z₁ J₁ K₁
vagy Y₁ Z₁ R₁

N2 G2 (G3) G90 Y₂ Z₂ J₂
K₂ R₂ Q



17.3.4-1 ábra



17.3.4-2 ábra

A metszéspontot mindig a G17, G18, G19 által kijelölt síkban számítja ki.

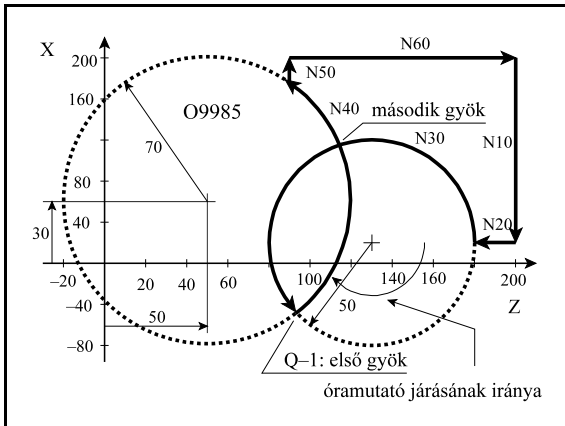
Az első mondatot (N1) **vagy a kör középponti koordinátájával** (I₁ J₁; I₁ K₁; J₁ K₁), **vagy a kör sugarával** (R₁) adjuk meg. Ebben a mondatban a középponti koordináták értelmezése megegyezik a körmegadás alapértelmezésével, vagyis a kezdőponttól mért relatív távolság.

A második mondatban (N2) megadott koordinátákat, így a **kör középpontját** meghatározó **I, J, K koordinátákat** is mindig **abszolút** (G90) adatként értelmezi a vezérlő.

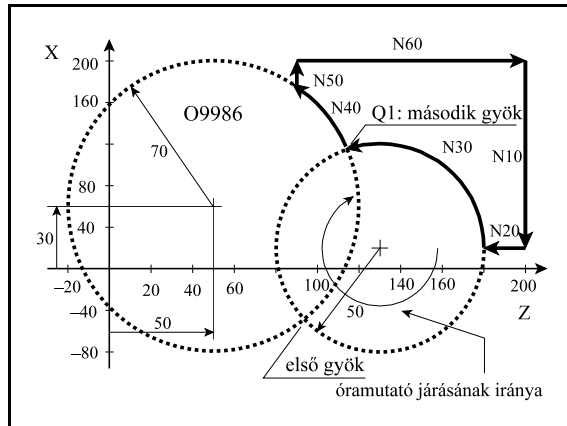
Azt, hogy a kiadódó két metszéspont közül melyiket számolja ki a **Q címen** lehet megadni. Ha a **cím értéke kisebb, mint nulla** (Q < 0, pl: Q = -1) **az első**, ha a **cím értéke nagyobb, mint nulla** (Q > 0, pl: Q = 1) **a második metszéspontot** számolja ki.

Első az a metszéspont amelyiken az óramutató járásának irányában haladva (függetlenül a programozott G2, G3 iránytól) elsőnek haladunk át.

Nézzük a következő példát:



17.3.4-3 ábra



17.3.4-4 ábra

```
O9985
N10 (G18) G0 X20 Z200
N20 G42 G1 Z180
N30 G3 X-80 Z130 R-50
N40 X174.892 Z90 I30 K50 R70
Q-1
N50 G40 G0 X200
N60 Z200
```

```
O9986
N10 (G18) G0 X20 Z200
N20 G42 G1 Z180
N30 G3 X-80 Z130 R-50
N40 X174.892 Z90 I30 K50 R70
Q1
N50 G40 G0 X200
N60 Z200
```

Az N40 körmondattól túlhatározott, mert középpontjának koordinátái is (I30 K50 abszolút értéként, valamint I sugárban értendő), és sugara is (R70) meg vannak adva. Ezért az előző, N30 mondatban programozott kör X-80 Z130 koordinátáit nem tekinti végponti értékeknek, hanem csak egy pontnak, amin a kör áthalad, és a végpont a kiszámított metszéspont lesz.

Az O9985 számú programban az óramutató járási iránya szerinti közelebbi metszéspontot adtuk meg (Q-1), míg az O9986-ban a távolabbit (Q1).

Két kör metszéspontjának megadását kombinálhatjuk letörés, vagy lekerekítés megadásával.

Például:

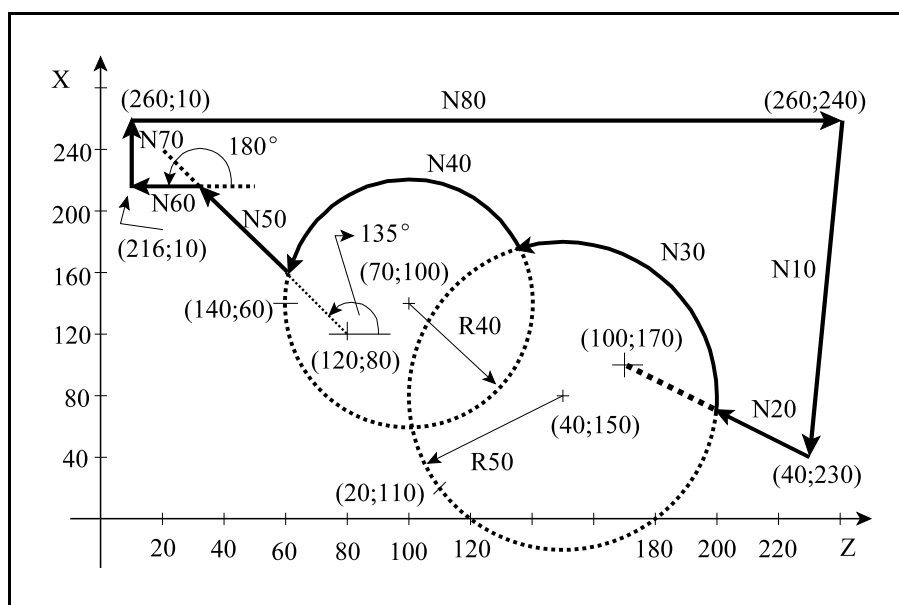
```
O9986
N10 (G18) G0 X20 Z200 M3 S200
N20 G42 G1 Z180
N30 G3 X-80 Z130 R-50 ,R20
N40 X174.892 Z90 I30 K50 R70 Q1
N50 G40 G0 X200
N60 Z200
```

Példánkban az N30 mondatban megadtunk egy 20 mm-es lekerekítést (,R20). A vezérlő kiszámítja az N30 és N40 mondat közötti metszéspontot, és az így kiadódó kontúrhoz beilleszti a programozott lekerekítést.

17.3.5 A metszéspontszámítások láncolása

A *metszéspontszámító mondatokat lehet láncolni*, vagyis *több, egymást követő mondatot is kijelölhetünk metszéspontszámításra*. A vezérlő *addig számít metszéspontot, amíg a programban túlhatározott egyeneseket, vagy köröket talál*.

Tekintsük az alábbi példát:



17.3.5-1 ábra

```

N10 (G18) G0 G42 X40 Z230 F300
N20 G1 X100 Z170
N30 G3 X20 Z110 I40 K150 R50 Q-1
N40 X140 Z60 I70 K100 R40 Q1
N50 G1 X120 Z80 ,A135 Q1
N60 X216 Z10 ,A180
N70 G40 G0 X260
N80 Z240

```

A fenti példában az N30, N40, N50, N60 mondat túlhatározott.

Az N20 egyenest nem a programozott végpontjáig (X100 Z170) vezeti, mert az N30 körmondat túlhatározott, vagyis I K R címek mind ki vannak töltve, és Q címen megadtuk, hogy melyik metszéspontot keresse.

Az N30 körmondatot sem a programozott végpontig (X20 Z110) vezeti, mert az N40 körmondat szintén túl van határozva.

A programban az utolsó túlhatározott mondat az N60 egyenes. Mivel az utána következő N70 egyenes mondat nincs túlhatározva, ezért az N60 mondatban programozott X216 Z10 koordinátákat nem az egyenes egy átmenő pontjának, hanem az N60 mondat végponti koordinátáinak tekintjük.

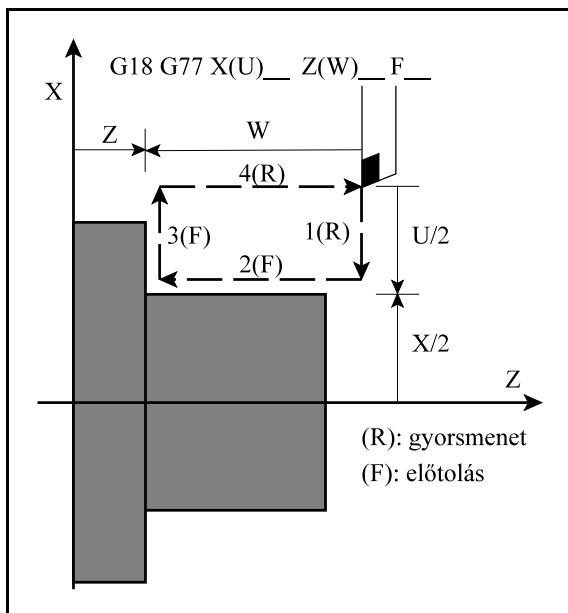
Általában elmondhatjuk, hogy a túlhatározott egyenes és kör mondatok kijelölt síkba eső koordinátapontjait csak akkor tekinti a vezérlő végponti koordinátának, ha utána már nem következik túlhatározott mondat.

18 Esztergáló ciklusok

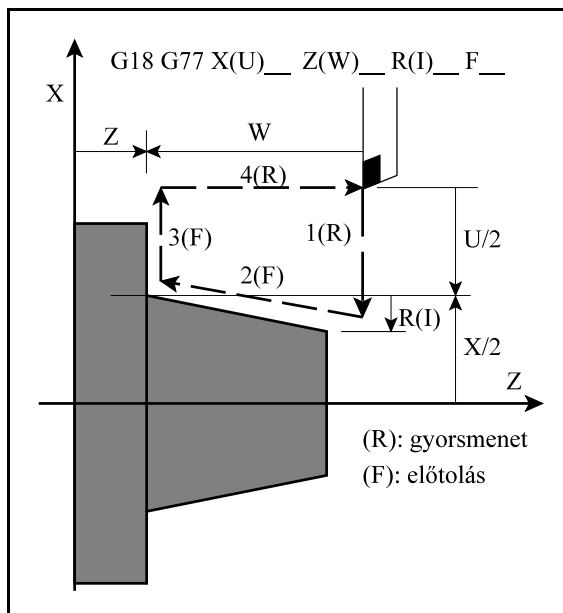
18.1 Egyszerű ciklusok

Az egyszerű ciklusok a G77 hosszsztergáló, a G78 egyszerű menetvágó és a G79 oldalazó ciklus.

18.1.1 A hosszsztergáló ciklus (G77)



18.1.1-1 ábra



18.1.1-2 ábra

Hengeres hosszsztergáló ciklust a következőképp adhatunk meg:

G18 (Z-X sík)

G77 X_p(U) Z_p(W) F

G19 (Y-Z sík)

G77 Y_p(V) Z_p(W) F

G17 (X-Y sík)

G77 X_p(U) Y_p(V) F

Mindig a kijelölt sík 2. tengelye mentén vesz fogást és a sík 1. tengelye mentén esztergál.

Inkrementális adatmegadás az I operátorral, G91 programozásával, illetve U, V, W cím megadásával is lehetséges.

Inkrementális adatmegadás esetén az adat előjele az 1-es és a 2-es pálya irányát határozza meg. Ábránkon, amely a G18 síkban történő hosszsztergálást mutatja, úgy az U, mind a W cím előjele negatív.

A mondatban F címen programozott, vagy megörökölt előtolással mozog a 2-es és a 3-as pályán, gyorsmenettel az 1-esen és a 4-esen.

Kúpos hosszesztergáló ciklust a következőképp adhatunk meg:

G18 (Z-X sík)

G77 X_p(U)___ Z_p(W)___ R(I)___ F___

G19 (Y-Z sík)

G77 Y_p(V)___ Z_p(W)___ R(K)___ F___

G17 (X-Y sík)

G77 X_p(U)___ Y_p(V)___ R(J)___ F___

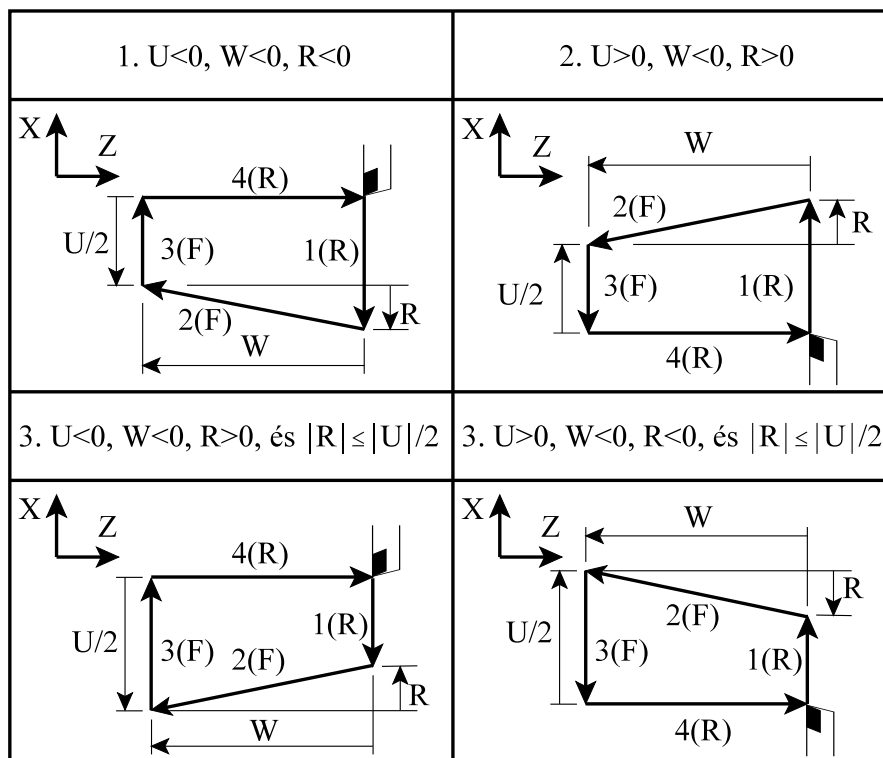
A kúposágot vagy R, vagy a kiválasztott sík függvényében az I, J, K címen adhatjuk meg. Mindkét esetben az adat értelmezése ugyanaz. Az **R(I, J, K)** címen megadott adat **mindig inkrementális** adatként kerül értelmezésre, és sík 2. tengelyének címén megadott pozíciótól **sugárban megadva** értendő. Az R(I, J, K) cím előjele határozza meg a kúp lejtési irányát.

A többi cím értelmezése megegyezik a hengeres hosszesztergáló ciklusnál elmondottakkal.

A G77 kód, és a G77 mondatban programozott adatok öröklődőek.

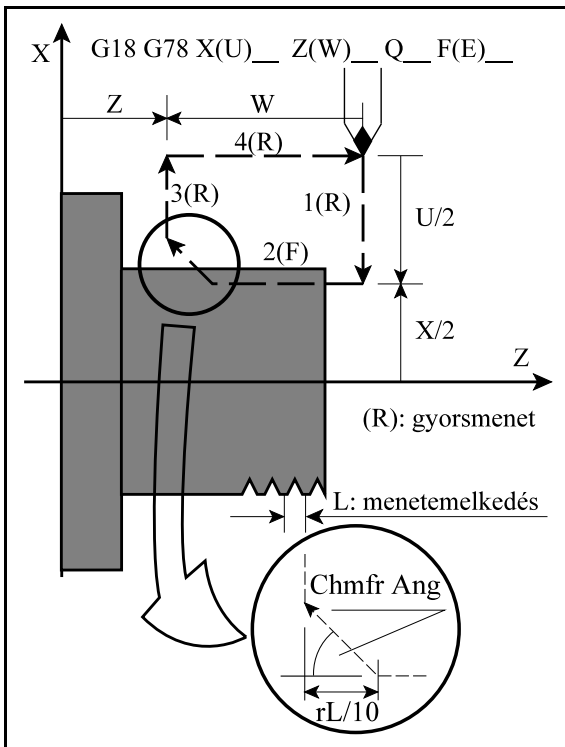
Mondatonkénti üzemben mind a négy művelet (1, 2, 3, 4) végén megáll.

Inkrementális programozás esetén U, W, és R(I) címek előjele az alábbiak szerint befolyásolják a mozgások irányát G18 állapotban:

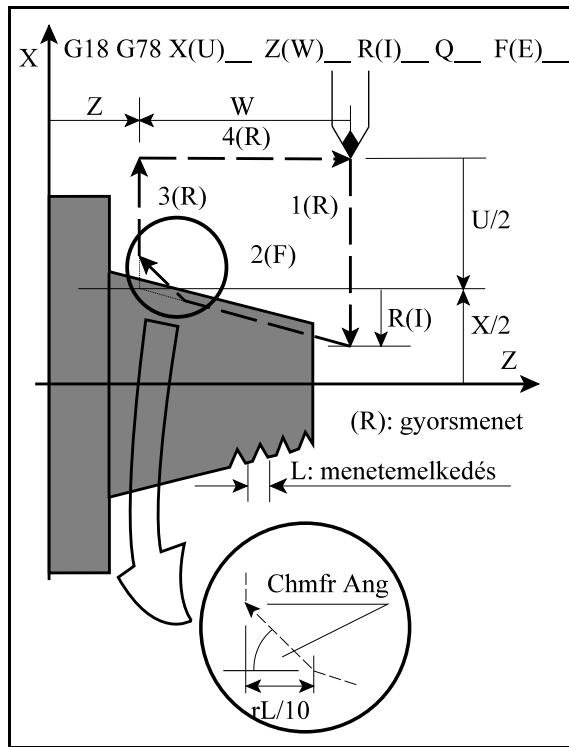


18.1.1-3 ábra

18.1.2 Az egyszerű menetvágó ciklus (G78)



18.1.2-1 ábra



18.1.2-2 ábra

Egyszerű menetvágó ciklust a következőképp adhatunk meg:

G18 (Z-X sík)

G78 X_p(U)___ Z_p(W)___ Q___ F(E)___

G19 (Y-Z sík)

G78 Y_p(V)___ Z_p(W)___ Q___ F(E)___

G17 (X-Y sík)

G78 X_p(U)___ Y_p(V)___ Q___ F(E)___

Mindig a kijelölt sík 2. tengelye mentén vesz fogást és a sík 1. tengelye mentén vágja a menetet. Inkrementális adatmegadás az I operátorral, G91 programozásával, illetve U, V, W cím megadásával is lehetséges.

Inkrementális adatmegadás esetén az adat előjele az 1-es és a 2-es pálya irányát határozza meg. Ábránkon, amely a G18 síkban történő menetvágást mutatja, úgy az U, mind a W cím előjele negatív.

A mondatban F címen programozzuk a menet emelkedését, vagy E címen az inchenkénti menetek számát, valamint Q címen a menet kezdetének a jeladó nullimpulzusától számított szögértét °-ban megadva, a G33 mondatban leírtak szerint.

Az 1, 3, 4 mozgás gyorsmenettel történik.

A 2-es szakasz végén, az N1607 Chmfr Ang paraméteren beállított szögű letörést végez.

A letörési szakasz hosszát az N1606 ThrdChmfr paraméter határozza meg, amit az ábrán r-rel jelöltünk. A szakasz hossza

$$r \cdot L / 10$$

ahol: r: az N1606 ThrdChmfr paraméter értéke

L: a programozott menetemelkedés

Ha a paraméter értéke pl. 4 és a programozott menetemelkedés F2, akkor a letörés hossza:

$$2 \cdot (4/10) = 0.8 \text{ mm}$$

Kúpos menetvágó ciklust a következőképp adhatunk meg:

G18 (Z-X sík)

G78 X_p(U)___ Z_p(W)___ R(I)___ Q___ F(E)___

G19 (Y-Z sík)

G78 Y_p(V)___ Z_p(W)___ R(K)___ Q___ F(E)___

G17 (X-Y sík)

G78 X_p(U)___ Y_p(V)___ R(J)___ Q___ F(E)___

A kúposágot vagy R, vagy a kiválasztott sík függvényében az I, J, K címen adhatjuk meg. Mindkét esetben az adat értelmezése ugyanaz. Az **R(I, J, K)** címen megadott adat **mindig inkrementális** adatként kerül értelmezésre, és sík 2. tengelyének címen megadott pozíciótól **sugarban megadva** értendő. Az R(I, J, K) cím előjele határozza meg a kúp lejtési irányát.

A többi cím értelmezése megegyezik a hengeres ciklusnál elmondottakkal.

A letörés szöge ebben az esetben is 45°-os és a letörés hosszát r-t a tengelyekkel párhuzamos egyenes mentén mérjük.

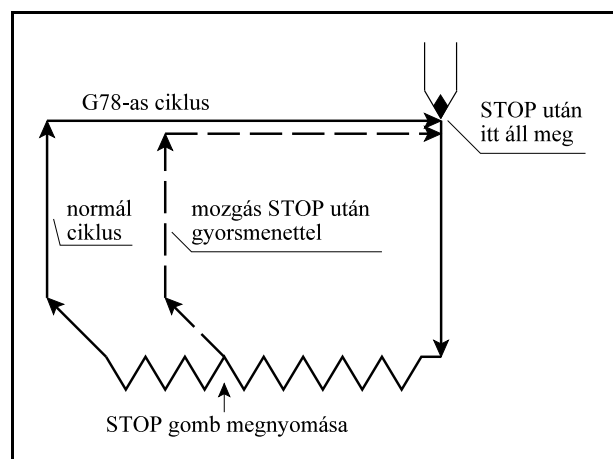
A G78 kód, és a G78 mondatban programozott adatok öröklődőek.

Mondatonkénti üzemben mind a négy művelet (1, 2, 3, 4) végén megáll.

A STOP gomb megnyomásának hatása a ciklus 2-es műveletében

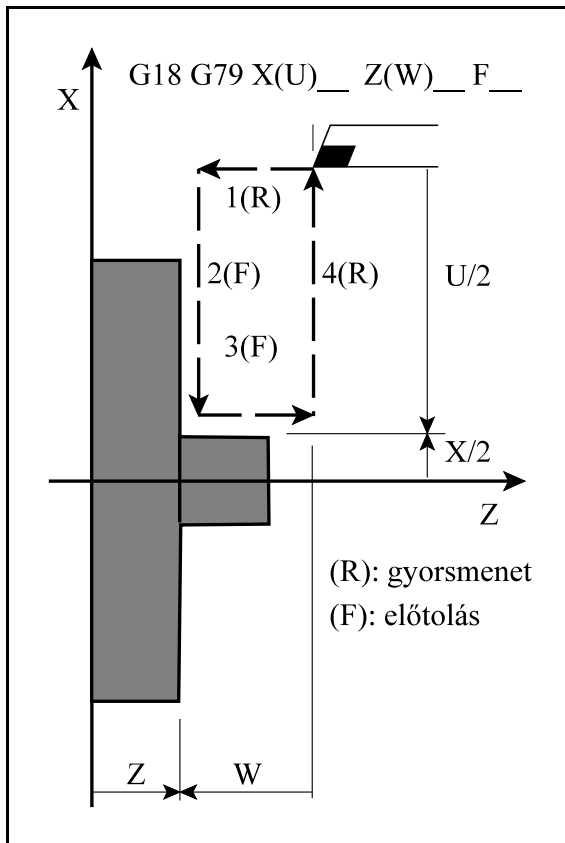
A ciklus 1-es, 3-as és 4-es műveletét a STOP gomb használatával bármikor meg lehet állítani és a szánok úgy állnak meg, mint normál G0 interpoláció esetén.

A 2-es menetvágó részben is hatásos a STOP gomb megnyomása, ám ekkor a vezérlő először megcsinálja ugyanazt a letörést, mint a 2-es művelet végén, azután először gyorsmenettel kiemel az X tengely mentén, majd Z-ben a kiindulási pontra áll. A menekülési útvonalon már nem hatásos a STOP gomb.

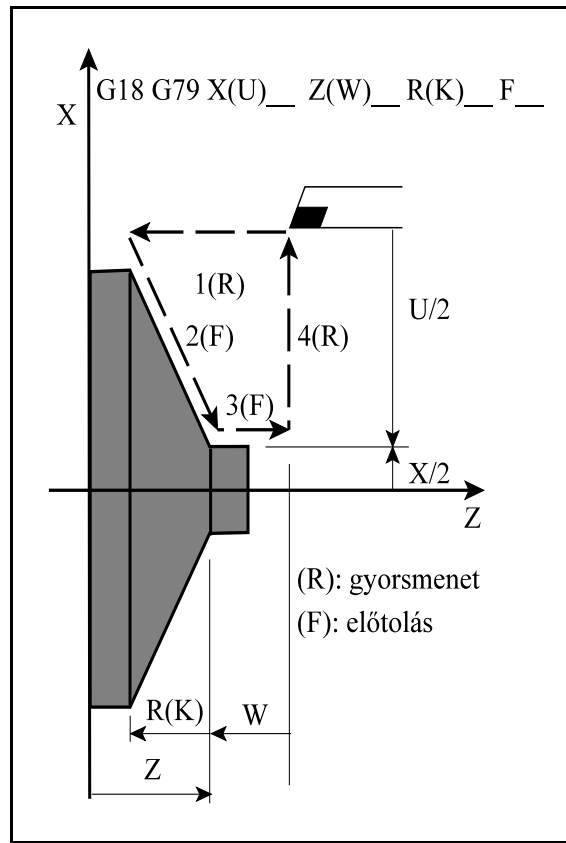


18.1.2 -3 ábra

18.1.3 Az oldalazó ciklus (G79)



18.1.3 -1 ábra



18.1.3-2 ábra

Oldalazó ciklust a következőképp adhatunk meg:

G18 (Z-X sík)
G79 X_p(U) Z_p(W) F

G19 (Y-Z sík)
G79 Y_p(V) Z_p(W) F

G17 (X-Y sík)
G79 X_p(U) Y_p(V) F

Mindig a kijelölt sík 1. tengelye mentén vesz fogást és a sík 2. tengelye mentén oldalaz.
 Inkrementális adatmegadás az I operátorral, G91 programozásával, illetve U, V, W cím megadásával is lehetséges.

Inkrementális adatmegadás esetén az adat előjele az 1-es és a 2-es pálya irányát határozza meg. Ábránkon, amely a G18 síkban történő oldalazást mutatja, úgy az U, mind a W cím előjele negatív.

A mondatban az F címen programozott, vagy a megörökölt előtolással mozog a 2-es és a 3-as pályán, gyorsmenettel az 1-esen és a 4-esen.

Kúpos oldalazó ciklust a következőképp adhatunk meg:

G18 (Z-X sík)

G79 X_p(U)___ Z_p(W)___ R(K)___ F___

G19 (Y-Z sík)

G79 Y_p(V)___ Z_p(W)___ R(J)___ F___

G17 (X-Y sík)

G79 X_p(U)___ Y_p(V)___ R(I)___ F___

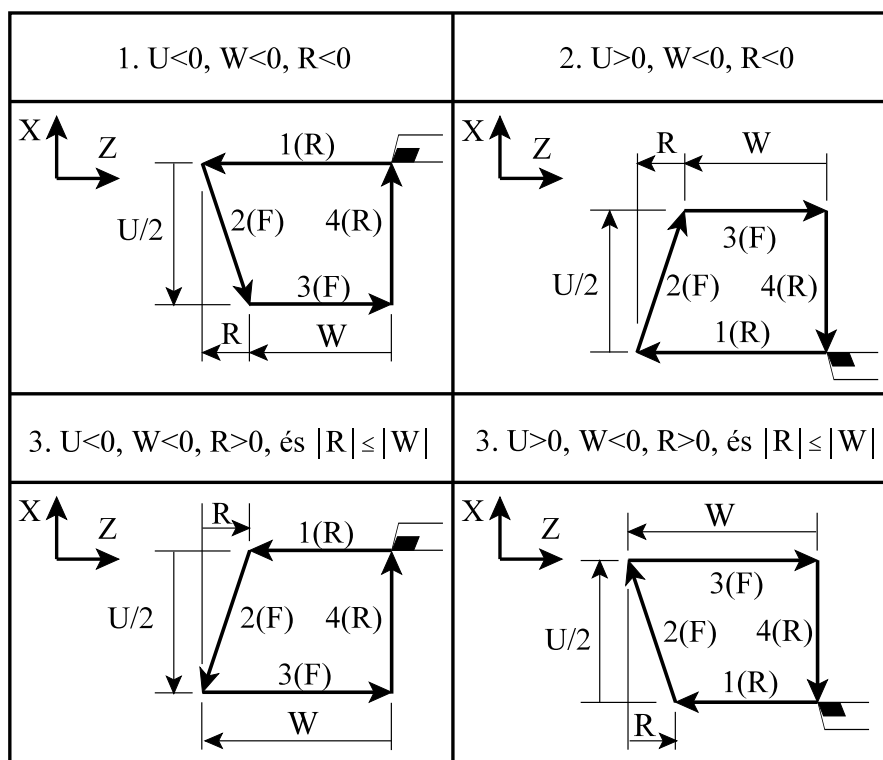
A kúposágot vagy R, vagy a kiválasztott sík függvényében az I, J, K címen adhatjuk meg. Mindkét esetben az adat értelmezése ugyanaz. Az **R(I, J, K)** címen megadott adat **mindig inkrementális** adatként kerül értelmezésre, és sík 1. tengelyének címén megadott pozíciótól **sugárban megadva** értendő. Az R(I, J, K) cím előjele határozza meg a kúp lejtési irányát.

A többi cím értelmezése megegyezik a hengeres oldalazó ciklusnál elmondottakkal.

A G79 kód, és a G79 mondatban programozott adatok öröklődőek.

Mondatonkénti üzemben mind a négy művelet (1, 2, 3, 4) végén megáll.

Inkrementális programozás esetén U, W, és R(K) címek előjele az alábbiak szerint befolyásolják a mozgások irányát G18 állapotban:



18.1.3 -3 ábra

18.1.4 Az egyszerű ciklusok használata

Úgy **a ciklusok G kódja**, mind a **bemenő paramétereit öröklődnek**. Ez azt jelenti, hogy, ha a ciklus változóit, X(Z, Y), U(W V), Z(Y, X), W(V, U), vagy R(I, J vagy K), már megadtuk egyszer és értékük változatlan, nem kell újra beírni a programba azokat. Például:

```
G91 G18...
G77 X-20 Z-50 F0.5
X-30
X-40
X-50
G0... (ciklus törölve)
...
```

A fenti példában csak a fogásvétel értéke (X) változik ezért csak ezt a címet kell újra kitölteni, a többi értéke változatlan marad.

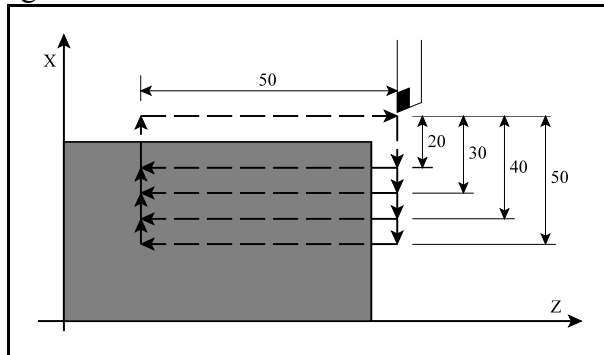
Ciklust csak akkor hajt végre a ciklus bekapcsolt állapotában, ha valamelyik mozgásra utaló változó, X(U), Y(V), Z(W) is ki van töltve.

Ha például ciklusállapotban önálló mondatban funkciót hajtunk végre a ciklusállapot bekapcsolva marad, de a ciklust nem ismétli meg:

```
G18...
G77 U-20 W-50 F0.5 (ciklust bekapcsolja és végrehajtja)
T202 (ciklus bekapcsolva, de nem hajtja végre)
U-30 (ciklust végrehajtja)
G1... (ciklus törölve)
...
```

A ciklust és az öröklődő változókat az 1-es csoportba tartozó interpolációs G kódok törlik.

Az egyszerű ciklusokat tartalmazó mondatokba írható M, S, T funkció is. A funkciók mindig a ciklus 1-es műveletében hajtódnak végre, vagy a mozgással párhuzamosan, vagy a mozgás végén. Ha ez bizonyos esetekben nem jó, a funkciót írjuk külön mondatba.



18.1.4 -1 ábra

18.2 Összetett ciklusok

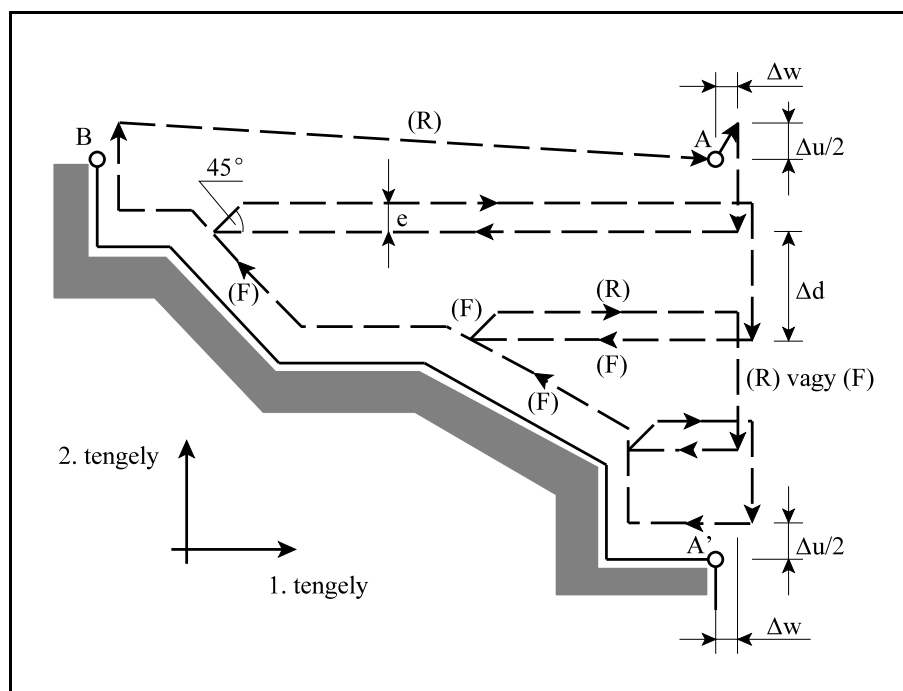
Az összetett ciklusok leegyszerűsítik a munkadarab program írását. Például a munkadarab kész méretének kontúrját a símitáshoz le kell írni. Ez a kontúr egyben meghatározza az alapját a munkadarab nagyolását végző ciklusoknak (G71, G72, G73). A nagyoló ciklusokon kívül rendelkezésre áll egy símitó (G70), egy menetvágó (G76) és két beszűrő ciklus (G74, G75) is.

18.2.1 Nagyoló ciklus (G71)

Két nagyoló ciklus van az 1-es és a 2-es típusú.

Az 1-es típusú nagyolóciklus

Ha adott egy munkadarab kész méretének kontúrja, amit az ábrán az A–A'–B pontokkal jelöltünk, a G71-es ciklus Δd nagyságú fogásokkal kinagyolja a nyers darabot $\Delta u/2$ és Δw nagyságú símitási ráhagyással.



18.2.1-1 ábra

1. Megadási módszer:

A nagyolóciklus megadása két, egymást követő mondat paramétereinek kitöltésével történik.

G18 (Z-X sík)

G71 U(Δd) R (e)

G71 P (n_s) Q (n_f) U(Δu) W(Δw) F(f) S(s) T(t)

N(n_s) X(U) ...

...

N(n_f)...

G19 (Y-Z sík)

G71 W(Δd) R (e)

G71 P (n_s) Q (n_f) W(Δu) V(Δw) F(f) S(s) T(t)

N(n_s) Z(W) ...

...

N(n_f)...

G17 (X-Y sík)

G71 V(Δd) R (e)

G71 P (n_s) Q (n_f) V(Δu) U(Δw) F(f) S(s) T(t)

N(n_s) Y(V) ...

...

N(n_f)...

Ahol:

Δd : fogásmélység. Mindig *sugárban* értendő *pozitív* szám. Ha a fogásmélység értékét nem adjuk meg a programban, akkor a vezérlő azt az N1600 Depth of Cut paraméterről veszi.

e: a kiemelés mértéke. Mindig *sugárban* értendő *pozitív* szám. Ha a kiemelés értékét nem adjuk meg a programban, a vezérlő azt az N1601 Escape a paraméterről veszi.

n_s : a program símitást leíró részének (A–A'–B szakasz) kezdő mondatszám.

n_f : a program símitást leíró részének (A–A'–B szakasz) befejező mondatszám.

Δu : a símitási ráhagyás nagysága és iránya *a kiválasztott sík 2. tengelyén*. A 2. tengely értelmezésének függvényében *átmérőben*, vagy *sugárban* értendő *előjeles* szám.

Δw : a símitási ráhagyás nagysága és iránya *a kiválasztott sík 1. tengelyén*. Az 1. tengely értelmezésének függvényében *átmérőben*, vagy *sugárban* értendő *előjeles* szám.

f, s, t: a ciklus folyamán a program símitást leíró n_s -tól n_f -ig tartó részében (A–A'–B szakasz) programozott F, S, T funkciókat nem hajtja végre, hanem a G71 mondatban megadott, vagy korábbról örökölt f, s, t értékeket érvényesíti.

Azt, hogy az U(W, V) címen megadott érték jelentése Δd , vagy Δu az dönti el, hogy az adott mondatban programoztunk-e P-t és Q-t. Ha nem, az U(W, V) cím jelentése Δd , ha igen az U(W, V) cím jelentése Δu .

A nagyoló ciklust az a mondat hajtja végre, amelyikben P-t és Q-t adtunk meg.

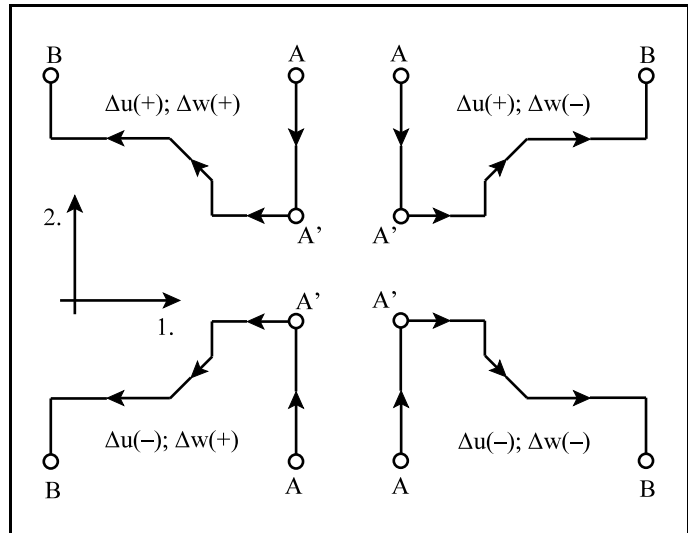
Az $A-A'$ pontok közötti mozgást a P címen megadott n_s számú mondatban kell megadni **kötelezően G0**, vagy **G1** programozásával. Az itt megadott kód dönti el, hogy a nagyolás során a **fogásvétel** (az $A-A'$ irányú mozgás) **gyorsmenettel** (G0 programozása esetén), vagy **előtolással** (G1 programozása esetén) történjék. Ebben a mondatban P(n_s) mindig G18 esetén X (G19 esetén Z, G17 esetén Y) **irányú mozgást kell megadni**, a másik tengelyre hivatkozni nem szabad.

Az $A'-B$ szakasz a tulajdonképpeni, egyenesekből és körívekből álló kontúr. Az **1-es típusú nagyológiklusnál a kontúrnak** úgy X (Z, Y), mind Z (Y, X) irányban **monoton növekvőnek, vagy csökkenőnek kell lennie**, ami azt jelenti, hogy **visszafordulás egyik irányban sem lehetséges**. A ciklus mind a négy síknegyedben használható. A mellékelt ábra a simítási ráhagyás előjelét is feltünteti.

Az n_s és n_f mondatok közötti programrészben programozott F, S, T funkciókat figyelmen kívül hagyja, és azokat érvényesíti, amelyeket a G71 mondatban (f, s, t), vagy előbb programoztunk.

Ugyanez vonatkozik az n_s és n_f mondatok között programozott konstans vágósebességre is, vagyis a **G71 mondat előtti G96, vagy G97 állapotot**, és **konstans vágósebesség értéket** érvényesíti. Az n_s -től n_f -ig terjedő mondatok nem tartalmazhatnak alprogramhívást.

A szerszám sugár korrekció számítás (G41, G42) bekapcsolható a ciklus végrehajtása alatt azzal a megkötéssel, hogy n_s -től n_f -ig terjedő mondatok között kell bekapcsolni (G41, vagy G42) és kikapcsolni (G40) a sugár korrekció számítást:



18.2.1 -2 ábra

HELYES
 $N(n_s)$ X(U) G41 ...
 (G41) ...

...
 (G40)
 $N(n_f)$ G40 ...

vagy

G41
 $N(n_s)$ X(U) ...
 ...
 $N(n_f)$...
 G40

HELYTELEN

G41
 $N(n_s)$ X(U) ...

...
 ...
 G40

$N(n_f)$...

vagy

$N(n_s)$ G41 X(U) ...
 ...
 $N(n_f)$...
 G40

2. Megadási módszer:

G18 (Z-X sík)

G71 P (n_s) Q (n_f) U(Δu) W(Δw) D(Δd) F(f) S(s) T(t)
 N(n_s) X(U) ...
 ...
 N(n_f)...

G19 (Y-Z sík)

G71 P (n_s) Q (n_f) W(Δu) V(Δw) D(Δd) F(f) S(s) T(t)
 N(n_s) Z(W) ...
 ...
 N(n_f)...

G17 (X-Y sík)

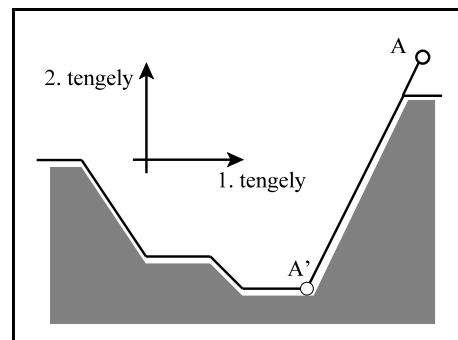
G71 P (n_s) Q (n_f) V(Δu) U(Δw) D(Δd) F(f) S(s) T(t)
 N(n_s) Y(V) ...
 ...
 N(n_f)...

A második megadási módszer bemenő paraméterei megegyeznek az elsőével.

A 2-es típusú nagyolóciklus

A 2-es típusú nagyoló ciklust ugyanúgy kell megadni, mint az 1-es típusút, a kódja G71 és a bemenő paraméterei is ugyanazok, mint az 1-es típusúnak.

A különbség a kontúr kezdő mondatának (n_s számú mondat) megadásában van. Amíg az 1-es típus hívása esetén ebben a mondatban az 1. tengelyre Z(Y,X) címre nem lehet hivatkozni, vagyis az A–A' szakasz mozgásának merőlegesnek kell lennie az 1. Z(Y, X) tengelyre, addig a 2-es típus hívása esetén ebben a mondatban kötelezően hivatkozni kell Z(Y, X) címre. Az A–A' szakasznak tehát nem kell merőlegesnek lennie az 1., Z(Y, X) tengelyre.



18.2.1 -3 ábra

1-es típus megadása

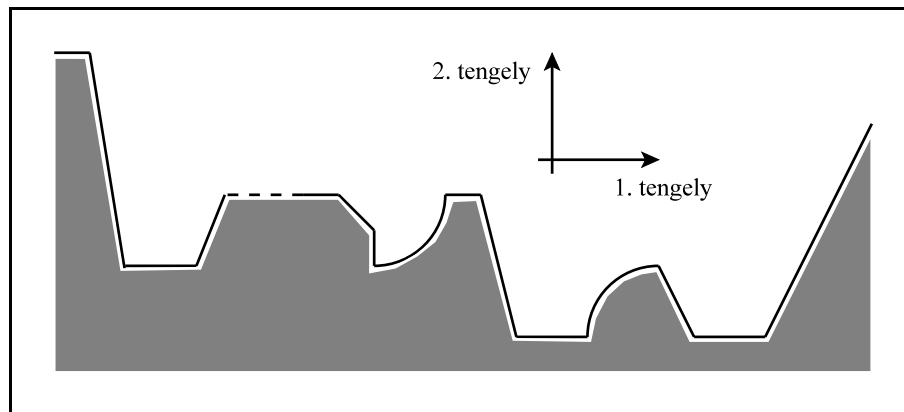
G18
 G71 U8 R1
 G71 P100 Q200 U0.5 W0.2
 N100 X(U) ____
 ...
 ...
 ...
 N200

2-es típus megadása

G18
 G71 U8 R1
 G71 P100 Q200 U0.5 W0.2
 N100 X(U) ____ Z(W) ____
 ...
 ...
 ...
 N200

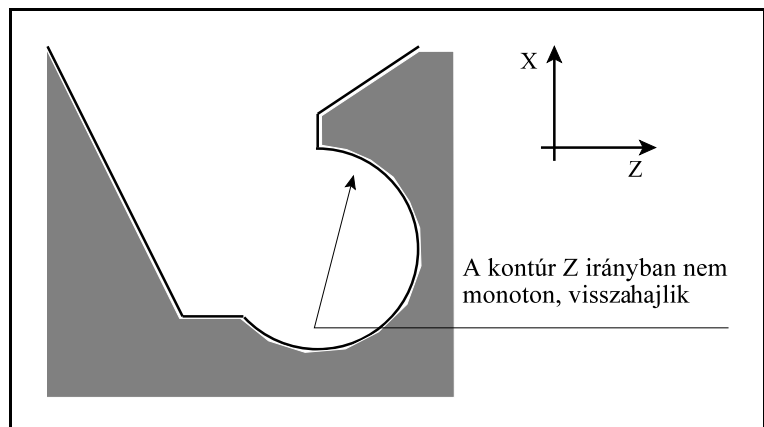
Abban az esetben, ha 2-es típusú ciklust kell használnunk, ám a kontúrt bevezető mondatban csak $X(Z, Y)$ irányban kell mozognunk, vagyis merőlegesen a $Z(Y, X)$ tengelyre, programozzuk a $Z(Y, X)$ tengely mentén inkrementális 0 elmozdulást, azaz $Z(Y, X) I0$ -t, vagy $W(V, U) 0$ -t.

A 2-es típusú nagyoló ciklusnál a kontúrnak a 2. tengely $X(Z, Y)$ irányában nem kell monoton növekvőnek, vagy csökkenőnek lennie, vagyis a kontúr lehet visszahajló, zsebeket tartalmazhat. A 1. tengely irányú símitási ráhagyást (Δw) nem szabad programozni $W(V, U) 0$ kell legyen, ellenkező esetben a szerszám belevághat a zseb egyik oldalába.



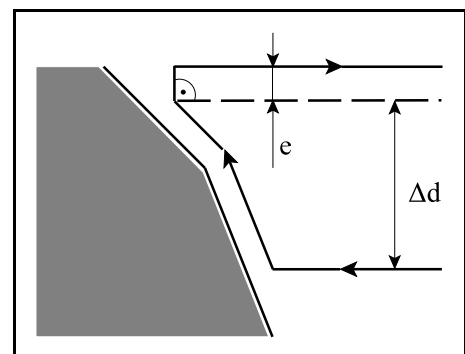
18.2.1 -4 ábra

Az 1. tengely $Z(Y, X)$ irányában viszont a kontúrnak továbbra is **monotonnak kell lennie**, nem lehet visszahajlás benne.



18.2.1 -5 ábra

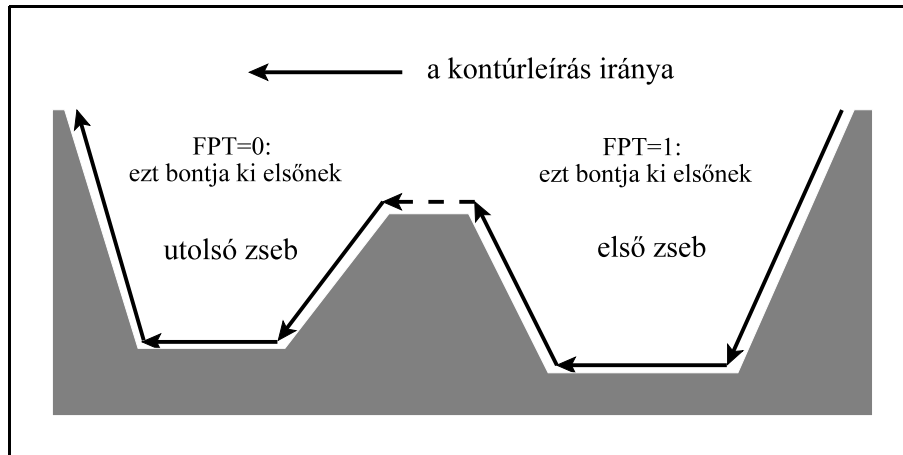
A 2-es típusú nagyoló ciklusnál a kiemelés a Z tengelyre merőlegesen történik, az érvényes kiemelési "e" értékkel.



18.2.1 -6 ábra

A 2-es típusú nagyolóciklus az N1611 Turning Cyc. Config. paraméter #2 FPT bitállása alapján kétféleképp bonthatja ki a zsebeket. Ha FPT:

- =0: a *kontúrleírás iránya szerinti utolsó zseb* kibontásával kezdi a ciklust,
- =1: a *kontúrleírás iránya szerinti első zseb* kibontásával kezdi a ciklust.



18.2.1 -7 ábra

Az alábbiakban leírtak úgy az 1-es, mint a 2-es típusú nagyolóciklusokra érvényesek.

A program folytatása a nagyolóciklus végrehajtása után

A nagyolóciklus végrehajtása után vagy a **G71 P Q mondatot követő mondatok** végrehajtásával, vagy a **Q címen megadott számú mondat után** folytatja a megmunkálást.

Az első esetben, ha a kontúr kezdőmondata a G71 P Q mondatot követi, a végrehajtás ráfut a kontúrleírásra és besímítja a darabot.

A második esetben a program végrehajtása a kontúr Q címen megadott zárómondata utáni mondatokkal folytatódik, ezért a símitáshoz G70 símitóciklust kell programozni. Ez olyankor hasznos, amikor ugyanazzal a nagyoló szerszámmal több nagyoló, vagy oldalazó ciklust hajtunk végre, majd a símitószerszámot beváltva, G70 símitó ciklus használatával ugyanazzal a szerszámmal besímítjük a különböző kontúrokat.

A fenti két lehetőség közül az N1611 Turning Cyc. Config. paraméter #1 SKP bitje alapján választhatunk:

- =0: a G71, G72, G73 utasítás után következő mondatokkal folytatja,
- =1: a G71, G72, G73 utasításban Q címen megadott mondatot követő mondatokkal folytatja.

SKP=0

```
G18
G71 U8 R1
G71 P100 Q200 U0.5 W0.2
(a program itt folytatódik)
N100
...
N200
(símitás befejezve)
```

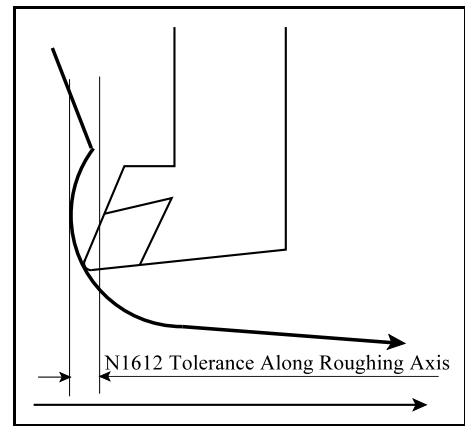
SKP=1

```
G18
G71 U8 R1
G71 P100 Q200 U0.5 W0.2
N100
...
N200
(a program itt folytatódik)
...
G70 P100 Q200 (símitás)
```

A pálya monotonitásának vizsgálata

Úgy az 1-es, mind a 2-es típusú nagyolóciklusban a **nagyoló tengely mentén a kontúrnak monotonnak kell lennie**, azaz a pálya visszahajló elemet nem tartalmazhat. Ha a pálya nem monoton, a vezérlő "A kontúr nem monoton" hibát jelez.

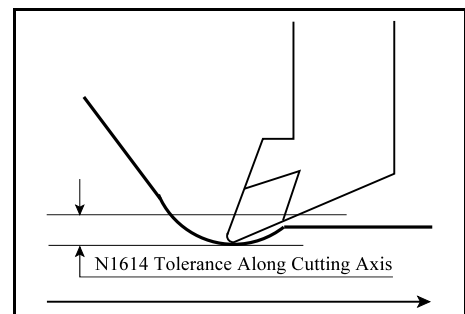
A monotonításra az N1613 Tolerance Along Roughing Axis paraméteren be lehet állítani egy tűrési értéket, amelyet a G71-es és a G72-es ciklusra is figyelembe vesz a vezérlő. Abban az esetben, ha a visszahajlás mértéke kisebb, mint a paraméteren megadott érték a vezérlő nem jelez hibát.



18.2.1-8 ábra

Az 1-es típusú nagyolóciklusnak a **forgácsoló (fogást vevő) tengely mentén is monotonnak kell lennie**, azaz a kontúr zsebet nem tartalmazhat. Ha a pálya nem monoton, a vezérlő "A kontúr nem monoton" hibát jelez.

A monotonításra az N1614 Tolerance Along Cutting Axis paraméteren be lehet állítani egy tűrési értéket, amelyet az 1-es típusú G71-es és a G72-es ciklusra is figyelembe vesz a vezérlő. Abban az esetben, ha a visszahajlás mértéke kisebb, mint a paraméteren megadott érték a vezérlő nem jelez hibát.



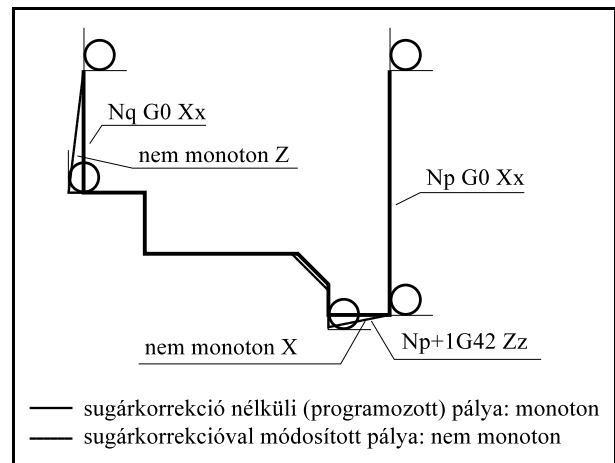
18.2.1-9 ábra

Úgy az 1-es, mind a 2-es típusú nagyolóciklusban visszahajló lehet a pálya a **szerszámsugárkorrekció bekapcsolásakor** (G41, G42), illetve **kikapcsolása esetén** (G40), még akkor is, ha a programozott kontúr monoton.

Ennek az az oka, hogy a vezérlő G40 állapotban a szerszám elméleti csúcsát vezeti a pályán, míg G41, G42 állapotban a csúcskört vezeti párhuzamosan a kontúrral.

Az N1611 Turning Cyc. Config. paraméter #3 FCK bitjével az alábbiakat lehet beállítani: G71, G72 nagyoló ciklusokban, a P és Q címen megadott számú mondatok között leírt **kontúr monotonitás vizsgálatát**, ha a paraméter értéke:

- =0: a **sugárkorrekcióval módosított**,
- =1: az eredeti, **sugárkorrekció nélküli pályára** végzi.



18.2.1-10 ábra

Mintapéldák

Az alább közölt mintapéldák N1611 Turning Cyc. Config. paraméter alábbi bitállításait tételezik fel:

```
#1 SKP=1
#3 FCK=1
```

Példa I-es típusú G71 ciklusra:

```
G18...
N10 G54 G0 X200 Z50
N20 G92 S3000
N30 G96 S400
N40 T101 M3 (NAGYOLÓKÉS, Q3)
N50 G0 X160 Z2
/N60 G71 U4 R1
/N70 G71 P80 Q160 U1 W0.5 F0.5 (D4) (NAGYOLÁS)
N80 G0 X100
N90 G1 G42 Z0
N100 X110 ,C2
N110 Z-25
N120 G2 X130 Z-35 R10
N130 G1 X140 ,R2
N140 Z-45 ,C3
N150 X155
N160 G40 X160
N170 G0 X200 Z50
N180 T202 (SÍMÍTÓKÉS, Q3)
N190 G0 X160 Z2
/N200 G70 P80 Q160 (SÍMÍTÁS)
N210 G0 X200 Z50
...
```

Példa II-es típusú G71 ciklusra:

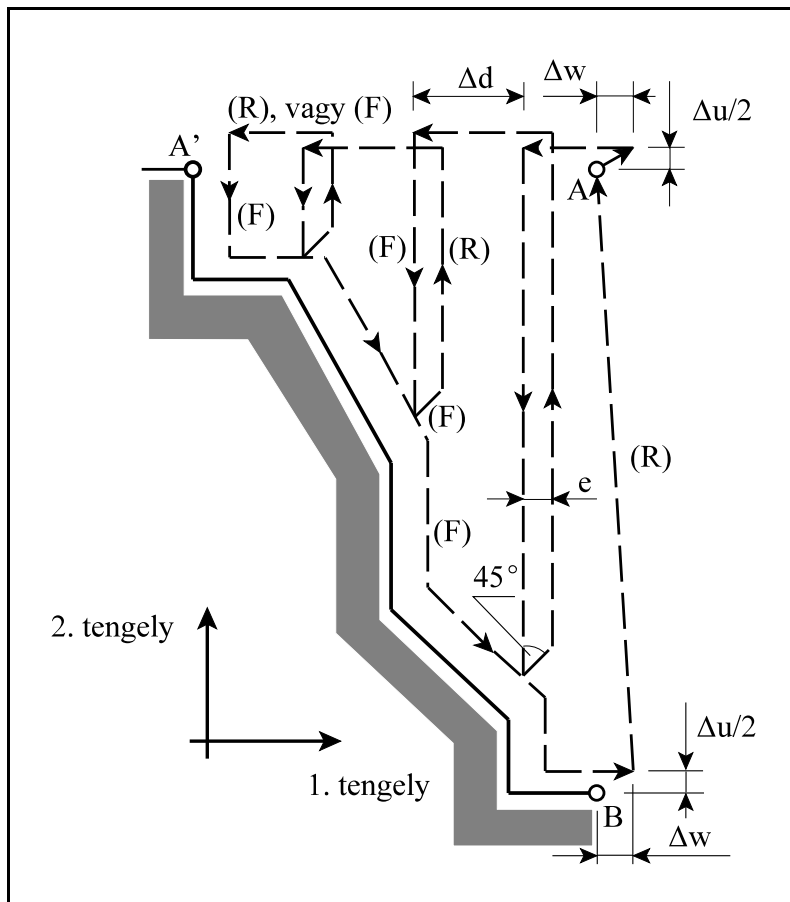
```
G18...
N10 G54 G0 X200 Z50
N20 G92 S3000
N30 G96 S400
N40 T606 M3 (NAGYOLÓKÉS, Q8)
N50 G0 X180 Z2
/N60 G71 U6 R4
/N70 G71 P80 Q210 U1 (D6) (NAGYOLÁS)
N80 G42 G0 X98 Z2
N90 G1 Z-10
N100 X100
N110 ,A150 (metszéspont N110-N120 között)
N120 X110 Z-40 ,A225
N130 Z-50
N140 ,A150 (metszéspont N140-N150 között)
```

N150 G3 X110 Z-110 I55 K-80 R30 Q1 (metszéspont N150-N160 között)
N160 G1 X100 Z-130 ,A210 Q1 (metszéspont N160-N170 között)
N170 G2 X100 Z-150 R10 Q-1 (metszéspont N170-N180 között)
N180 X110 Z-180 I55 K-160 R20 Q-1
N190 G1 X170
N200 Z-190
N210 G40 G0 X180
N220 G0 X200 Z50
N230 T707 (SÍMÍTÓKÉS Q8)
/N240 **G70 P80 Q210** (SÍMÍTÁS)
N250 G0 X200 Z50
...

18.2.2 Homlok nagyoló ciklus (G72)

A homlok nagyoló ciklus (G72) ugyanaz, mint a G71-es nagyoló ciklus, azzal a különbséggel, hogy a forgácsolás a sík 2. tengelyével párhuzamosan történik. *A G71-es ciklus leírásánál mondottak érvényesek érvényesek a G72-es ciklusra is*, ezért ezeket itt külön nem részletezzük.

Az 1-es típusú homlok nagyoló ciklus



18.2.2 -1 ábra

1. Megadási módszer:**G18** (Z-X sík)**G72** **W**(Δd) **R** (e)**G72** **P** (n_s) **Q** (n_f) **U**(Δu) **W**(Δw) **F**(f) **S**(s) **T**(t) $N(n_s)$ $Z(W)$...

...

 $N(n_f)$...**G19** (Y-Z sík)**G72** **V**(Δd) **R** (e)**G72** **P** (n_s) **Q** (n_f) **W**(Δu) **V**(Δw) **F**(f) **S**(s) **T**(t) $N(n_s)$ $Y(V)$...

...

 $N(n_f)$...**G78** (X-Y sík)**G72** **U**(Δd) **R** (e)**G72** **P** (n_s) **Q** (n_f) **V**(Δu) **U**(Δw) **F**(f) **S**(s) **T**(t) $N(n_s)$ $X(U)$...

...

 $N(n_f)$...

A bemenő paraméterek jelentése teljes egészében megegyezik a G71 ciklusnál elmondottakkal.

2. Megadási módszer:**G18** (Z-X sík)**G72** **P** (n_s) **Q** (n_f) **U**(Δu) **W**(Δw) **D**(Δd) **F**(f) **S**(s) **T**(t) $N(n_s)$ $Z(W)$...

...

 $N(n_f)$...**G19** (Y-Z sík)**G72** **P** (n_s) **Q** (n_f) **W**(Δu) **U**(Δw) **D**(Δd) **F**(f) **S**(s) **T**(t) $N(n_s)$ $Y(V)$...

...

 $N(n_f)$...**G78** (X-Y sík)**G72** **P** (n_s) **Q** (n_f) **V**(Δu) **U**(Δw) **D**(Δd) **F**(f) **S**(s) **T**(t) $N(n_s)$ $X(U)$...

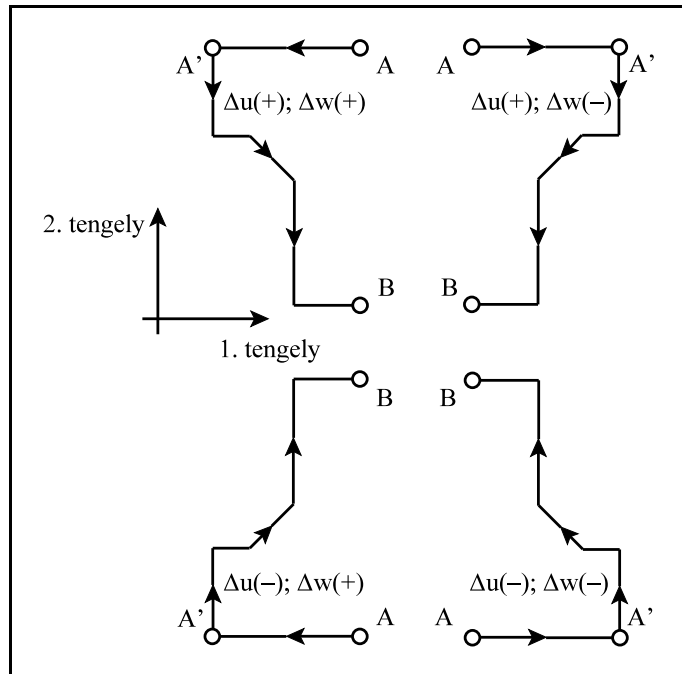
...

 $N(n_f)$...

18.2.2 Homlok nagyoló ciklus (G72)

Az $A-A'$ pontok közötti mozgást a P címen megadott n_s számú mondatban kell megadni. Ebben a mondatban P(n_s) mindig G18 esetén Z (G19 esetén Y, G17 esetén X) **irányú mozgást kell megadni**, a másik tengelyre hivatkozni nem szabad.

A ciklus mind a négy síknegyedben használható. Az ábra a simítási ráhagyás előjelét is megadja mind a négy esetre.



18.2.2 -2 ábra

A 2-es típusú homlok nagyoló ciklus

A **2-es típusú nagyoló ciklust ugyanúgy kell megadni, mint az 1-es típusút**, a kódja G72 és a be-menő paraméterei is ugyanazok, mint az 1-es típusúnak.

A **különbség a kontúr kezdő mondatának** (n_s számú mondat) **megadásában van**. Amíg az 1-es típus hívása esetén ebben a mondatban a 2. tengelyre X(Z, Y) címre nem lehet hivatkozni, vagyis az $A-A'$ szakasz mozgásának merőlegesnek kell lennie a 2. X(Z, Y) tengelyre, addig a 2-es típus hívása esetén ebben a mondatban kötelezően **hivatkozni kell X(Z, Y) címre**. Az $A-A'$ szakasznak tehát nem kell merőlegesnek lennie a 2., X(Z, Y) tengelyre.

Mintapéldák

Az alább közölt mintapéldák N1611 Turning Cyc. Config. paraméter alábbi bitállításait tételezik fel:

```
#1 SKP=1  
#3 FCK=1
```

Példa I-es típusú G72 ciklusra:

```
G18 . . .  
N10 G54 G0 X200 Z50  
N20 G92 S3000  
N30 G96 S600  
N40 T101 M3 (NAGYOLÓKÉS, Q3)  
N50 G0 X161 Z2  
/N60 G72 W5 R1  
/N70 G72 P80 Q140 U1 W0.5 (D5) F0.5 (NAGYOLÁS)
```

```

N80 G41 G0 Z-25
N90 G1 X120 ,C3
N100 Z-17 ,R2
N110 G3 X100 Z-7 R10
N120 G1 Z-3 ,C1
N130 X80
N140 G40 Z2
N150 G0 X200 Z50
N160 T202 (SÍMÍTÓKÉS Q3)
N170 X161 Z2
/N180 G70 P80 Q140 (SÍMÍTÁS)
N190 G0 X200 Z50
...

```

Példa II-es típusú G72 ciklusra:

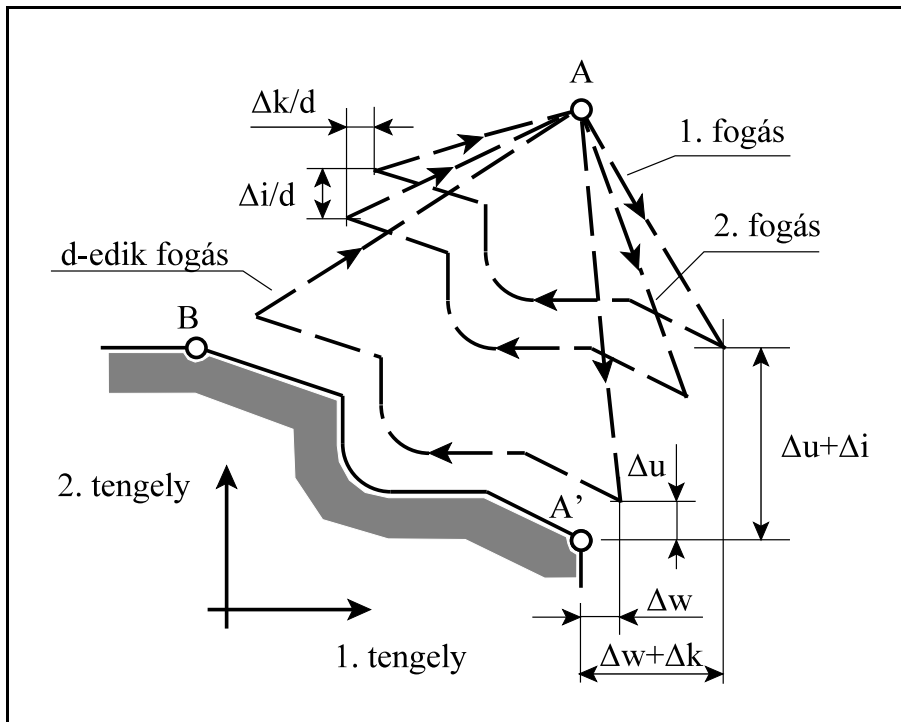
```

G18...
N10 T505 (NAGYOLKÉS, Q7)
N20 G54 G0 X200 Z50
N30 G92 S3000
N40 G96 S200 M3
N50 G0 X160 Z5
/N60 G72 W4 R1
/N70 G72 P80 Q150 W0.5 (D4) (NAGYOLÁS)
N80 G41 G0 X160 Z-25
N90 G1 X145 ,R4
N100 X125 Z-10 ,R3
N110 X105 Z-25 ,R4
N120 X95
N130 G2 X15 Z-25 R20 ,R5
N140 G1 X0
N150 G40 Z5
N160 G0 X200 Z50
N170 T202 (SÍMÍTÓKÉS? Q7)
N180 X160 Z5
/N190 G70 P80 Q150 (SÍMÍTÁS)
N200 G0 X200 Z50
...

```

18.2.3 Minta ismétlő ciklus (G73)

Ez a ciklus előkovácsolt, öntött, vagy előnagyolt darabok nagyolásánál alkalmazható, ahol a végleges forma körvonalai már rendelkezésre állnak. A ciklus fogásról fogásra ismétel egy, a programban leírt kontúrt.



18.2.3 -1 ábra

1. Megadási módszer:**G18** (Z-X sík)**G73** U(Δi) W(Δk) R (d)**G73** P (n_s) Q (n_f) U(Δu) W(Δw) F(f) S(s) T(t)N(n_s) ...

...

N(n_f) ...**G19** (Y-Z sík)**G73** W(Δi) V(Δk) R (d)**G73** P (n_s) Q (n_f) W(Δu) V(Δw) F(f) S(s) T(t)N(n_s) ...

...

N(n_f) ...**G17** (X-Y sík)**G73** V(Δi) U(Δk) R (d)**G73** P (n_s) Q (n_f) V(Δu) U(Δw) F(f) S(s) T(t)N(n_s) ...

...

N(n_f) ...

utasítással történik, ahol:

- Δi :** a nagyolási ráhagyás nagysága és iránya a kiválasztott sík 2. tengelyén. Mindig *sugárban* értendő *előjeles* szám. Ha a nagyolási ráhagyás értékét nem adjuk meg a programban, akkor a vezérlő azt az N1603 Relief Ax2 paraméterről veszi.
- Δk :** a nagyolási ráhagyás nagysága és iránya a kiválasztott sík 1. tengelyén. Mindig *sugárban* értendő *előjeles* szám. Ha a nagyolási ráhagyás értékét nem adjuk meg a programban, akkor a vezérlő azt az N1602 Relief Ax1 paraméterről veszi.
- d:** a fogások száma. Ha a fogások számát nem adjuk meg a programban, akkor a vezérlő azt az N1604 Numb of Div paraméterről veszi.
- n_s :** a program símitást leíró részének (A–A'–B szakasz) kezdő mondat száma.
- n_f :** a program símitást leíró részének (A–A'–B szakasz) befejező mondat száma.
- Δu :** a símitási ráhagyás nagysága és iránya a kiválasztott sík 2. tengelyén. A koordináta értelmezésének függvényében *átmérőben*, vagy *sugárban* értendő *előjeles* szám.
- Δw :** a símitási ráhagyás nagysága és iránya a kiválasztott sík 1. tengelyén. A koordináta értelmezésének függvényében *átmérőben*, vagy *sugárban* értendő *előjeles* szám.

Az, hogy a G73 mondatban megadott U(W, V) és W(V, U) cím jelentése Δi és Δk , vagy Δu és Δw , attól függ, hogy az adott mondatban programoztunk-e P-t és Q-t. Vagyis, ha programoztunk P-t és Q-t, akkor U és W jelentése Δu és Δw , ha nem programoztunk, akkor Δi és Δk .

A ciklus a P-t és Q-t tartalmazó mondatban hajtja végre. A ciklus mind a négy síknegyedben végrehajtható a Δi , Δk , Δu , Δw értékek előjelének függvényében. Minden fogásvétel után és a ciklus végén a szerszám az "A" pontra tér vissza.

Az n_s és n_f mondatok közötti programrészben programozott **F, S, T** funkciókat figyelmen kívül hagyja és azokat érvényesíti, amelyeket a G73 mondatban (f, s, t), vagy előbb programoztunk. Ugyanez vonatkozik az n_s és n_f mondatok között programozott konstans vágósebességre is, vagyis a **G73 mondat előtti G96, vagy G97 állapotot, és konstans vágósebesség értéket** érvényesíti. Az n_s -től n_f -ig terjedő mondatok nem tartalmazhatnak alprogramhívást. A szerszám sugár korrekció számítás megadása lehetséges a ciklust leíró mondatokban a G71 funkció tárgyalásakor említett megszorításokkal.

A program folytatása a minta ismétlődő ciklus végrehajtása után

A ciklus végrehajtása után vagy a **G73 P Q mondatot követő mondatok** végrehajtásával, vagy a **Q címen megadott számú mondat után** folytatja a megmunkálást. Az utóbbi esetben G70 símitó ciklussal lehet a darabot besímitani.

A fenti két lehetőség közül az N1611 Turning Cyc. Config. paraméter #1 SKP bitje alapján választhatunk:

- =0: a G71, G72, G73 utasítás után következő mondatral folytatja,
- =1: a G71, G72, G73 utasításban Q címen megadott mondatot követő mondatral folytatja.

2. Megadási módszer:

G18 (Z-X sík)

G73 P (n_s) **Q** (n_f) **U**(Δu) **W**(Δw) **I**(Δi) **K**(Δk) **D**(d) **F**(f) **S**(s) **T**(t)
 $N(n_s)$...
 ...
 $N(n_f)$...

G19 (Y-Z sík)

G73 P (n_s) **Q** (n_f) **W**(Δu) **V**(Δw) **K**(Δi) **J**(Δk) **D**(d) **F**(f) **S**(s) **T**(t)
 $N(n_s)$...
 ...
 $N(n_f)$...

G17 (X-Y sík)

G73 P (n_s) **Q** (n_f) **V**(Δu) **U**(Δw) **J**(Δi) **I**(Δk) **D**(d) **F**(f) **S**(s) **T**(t)
 $N(n_s)$...
 ...
 $N(n_f)$...

A második megadási módszer bemenő paramétereinek megegyeznek az elsőével.

Mintapélda

Az alább közölt mintapélda N1611 Turning Cyc. Config. paraméter #1 SKP=1 bitállítást tételez fe.

```
G18...
N10 G96 S200
N20 G92 S3000
N30 G54 G0 X70 Z20
N40 T111 M3 (NAGYOLÓKÉS Q3)
/N50 G73 U10 W4 R4
/N60 G73 P70 Q130 U1 W0.5 (I10 K4 D4)
N70 G42 X10 Z5 F1.4
N80 G1 Z0
N90 X20 ,C1
N100 Z-10
N110 G2 X40 Z-20 R10
N120 G1 Z-30
N130 G40 G0 X42
N140 G0 X70 Z20
N150 T212 (SÍMÍTÓKÉS Q3)
/N160 G70 P70 Q130
N170 G0 X70 Z20
...
```

18.2.4 Símító ciklus (G70)

A G71, G72, vagy a G73-mal történő, az n_s -től n_f -ig terjedő mondatok által leírt kontúr nagyolása után, a kontúrt a G70 utasítással símithatjuk be. A símítás megadása a következő utasítással történik:

G70 P (n_s) **Q** (n_f)

n_s : a program símítást leíró részének kezdő mondat száma.

n_f : a program símítást leíró részének befejező mondat száma.

A ciklus folyamán a program símítást leíró n_s -től n_f -ig tartó részében programozott **F**, **S**, **T** funkciókat, illetve a **konstans vágósebességszámítás**ra vonatkozó **parancsokat végrehajtja**, ellentétben a G71, G72, G73 ciklussal.

A símító ciklus végén a szerszám visszatér a kiindulási pontra és a következő mondat kerül beolvasásra.

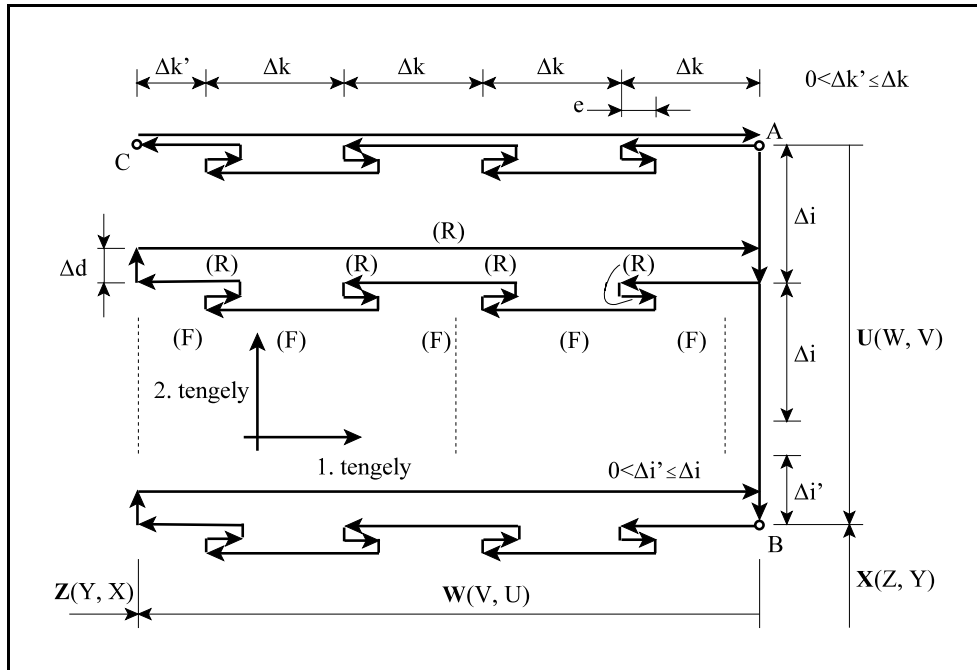
A símító ciklus folyamán a **szerszámsugár korrekció** számítása **működik**.

Az n_s -től n_f -ig terjedő mondatok nem tartalmazhatnak alprogramhívást.

A símító ciklusok használatára a G71, G72, G73 ciklusoknál található példa.

18.2.5 Homlok beszúró ciklus (G74)

A mellékelt ábra a G74 típusú homlok beszúró ciklus mozgását mutatja. A beszúrás a kiválasztott sík 1. tengelye mentén történik.



18.2.5 -1 ábra

1. Megadási módszer:

G18 (Z-X sík)

G74 R (e)

G74 X_p(U) Z_p(W) P (Δi) Q (Δk) R (Δd) F

G19 (Y-Z sík)

G74 R (e)

G74 Y_p(V) Z_p(W) P (Δi) Q (Δk) R (Δd) F

G17 (X-Y sík)

G74 R (e)

G74 X_p(U) Y_p(V) P (Δi) Q (Δk) R (Δd) F

utasítással történik, ahol:

e: a visszahúzás mértéke. Öröklődő, **sugárban megadott pozitív szám**. Addig nem változik, amíg át nem írjuk. Ha a visszahúzás mértékét nem adjuk meg a programban, akkor a vezérlő azt az N1605 Retr G74 G75 paraméterről veszi.

X(Z, Y): a "B" pontnak a sík 2. tengelyén megadott abszolút koordinátája

U(W, V): az AB pont távolsága inkrementálisan

Z(Y, X): a "C" pontnak a sík 1. tengelyén megadott abszolút koordinátája

W(V, U): az AC pont távolsága inkrementálisan

Δi: a fogásvétel nagysága a sík 2. tengelyén. *Sugárban megadott* pozitív szám.

Δk: a fogásvétel nagysága a sík 1. tengelyén. *Sugárban megadott* pozitív szám.

Δd: a szerszám elhúzásának mértéke alul. *Sugárban megadott* szám. A mozgás iránya mindig az AB vektor előjelével ellentétes. Az 1. beszűrés végén a Δd elmozdulást elhagyja.

F: előtolás

Az ábrán (F)-fel jelöltük azokat a szakaszokat, amelyeket előtolással és (R)-rel azokat, amelyeket gyorsmenettel tesz meg.

Azt, hogy a G74-es mondatban az R cím kitöltése e-t, vagy Δd-t határoz meg, a Z(Y, X), vagy a W(V, U) cím kitöltése dönti el. Ha a cím ki van töltve R cím jelentése Δd.

Ha az X(Z, Y), vagy az U(W, V) cím kitöltését és P(Δi) cím kitöltését is elhagyjuk, akkor mozgás csak a Z tengely mentén történik, vagyis egy fűróciklus játszódik le.

2. Megadási módszer:

G18 (Z-X sík)

G74 X_p(U) Z_p(W) I (Δi) K (Δk) D (Δd) F

G19 (Y-Z sík)

G74 Y_p(V) Z_p(W) K (Δi) J (Δk) D (Δd) F

G17 (X-Y sík)

G74 X_p(U) Y_p(V) J (Δi) I (Δk) D (Δd) F

A második megadási módszer bemenő paramétereinek értelmezése megegyezik az elsőével.

Mintapélda

G18 . . .

G0 X100 Z5 S1=1500 M3 F1

G74 R6

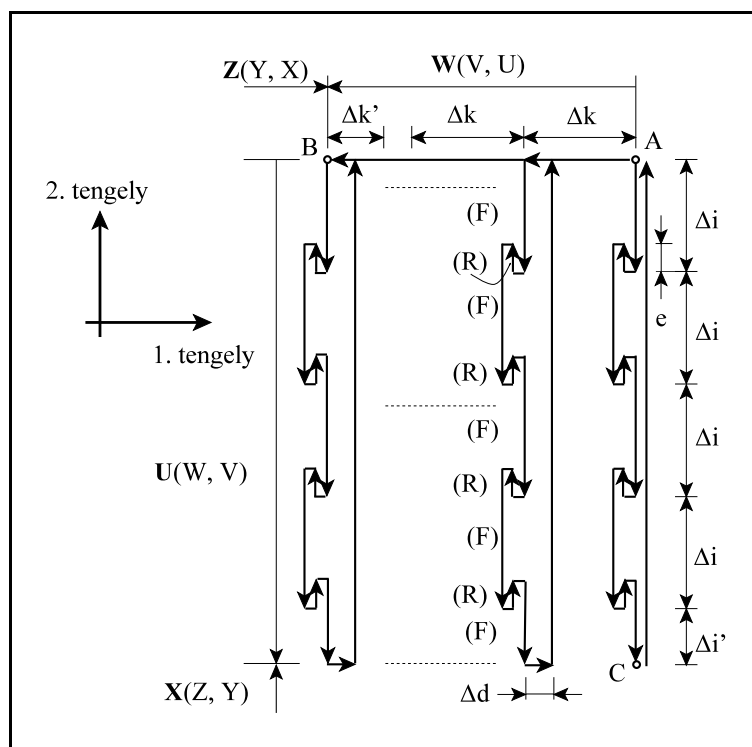
G74 X70 Z-100 P6 (I6) Q25 (K25) R2 (D2) F1

G0 X120 Z10

. . .

18.2.6 Beszúró ciklus (G75)

A mellékelt ábra a G75 típusú beszúró ciklus mozgását mutatja. A beszúrás a kiválasztott sík 2. tengelye mentén történik.



18.2.6 -1 ábra

1. Megadási módszer:

G18 (Z-X sík)

G75 R (e)

G75 X_p(U) Z_p(W) P (Δi) Q (Δk) R (Δd) F

G19 (Y-Z sík)

G75 R (e)

G75 Y_p(V) Z_p(W) P (Δi) Q (Δk) R (Δd) F

G17 (X-Y sík)

G75 R (e)

G75 X_p(U) Y_p(V) P (Δi) Q (Δk) R (Δd) F

A ciklus változóinak értelmezése megegyezik a G74 cikluséval, azzal a különbséggel, hogy a beszúrás X(Z, Y) irányban történik, ezért X(Z, Y) U(W, V) és Z(Y, X) W(V, U) címek értelmezése felcserélődik.

2. Megadási módszer:

G18 (Z-X sík)

G75 X_p(U) Z_p(W) I (Δi) K (Δk) D (Δd) F

G19 (Y-Z sík)

G75 Y_p(V) Z_p(W) K (Δi) J (Δk) D (Δd) F

G17 (X-Y sík)

G75 X_p(U) Y_p(V) J (Δi) I (Δk) D (Δd) F

A második megadási módszer bemenő paramétereinek értelmezése megegyezik az elsőével.

Mintapélda

G18 ...

G0 X105 Z-15 S1=1500 M3 F1

G75 R6

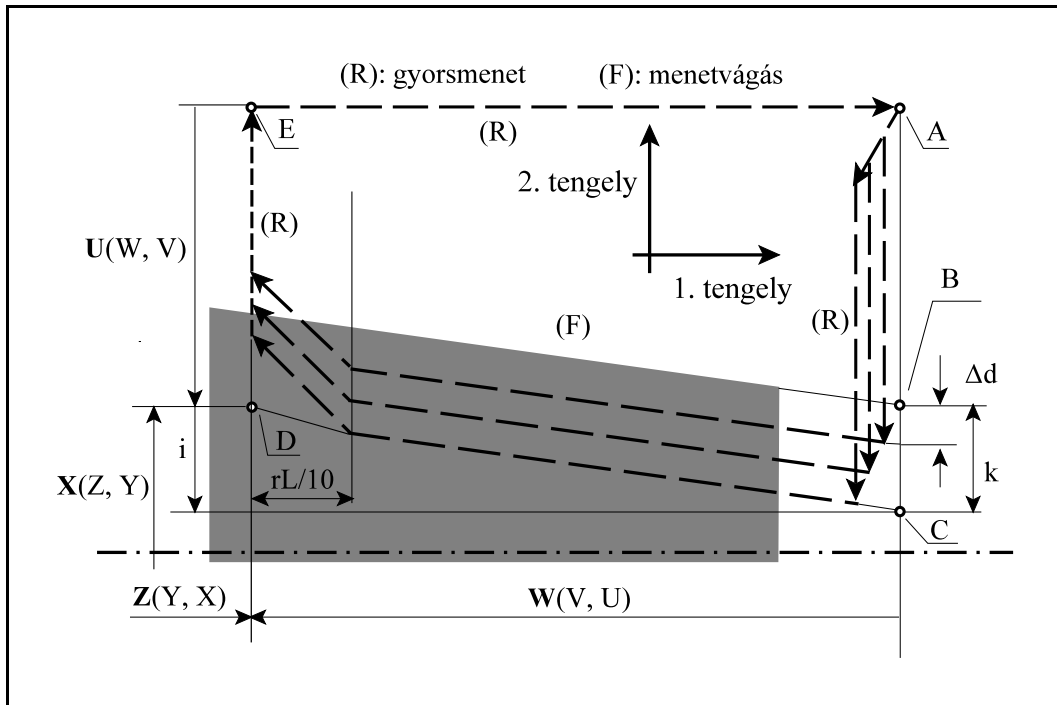
G75 X20 Z-30 P15 (I15) Q6 (K6) R2 (D2) F1

G0 X120 Z10

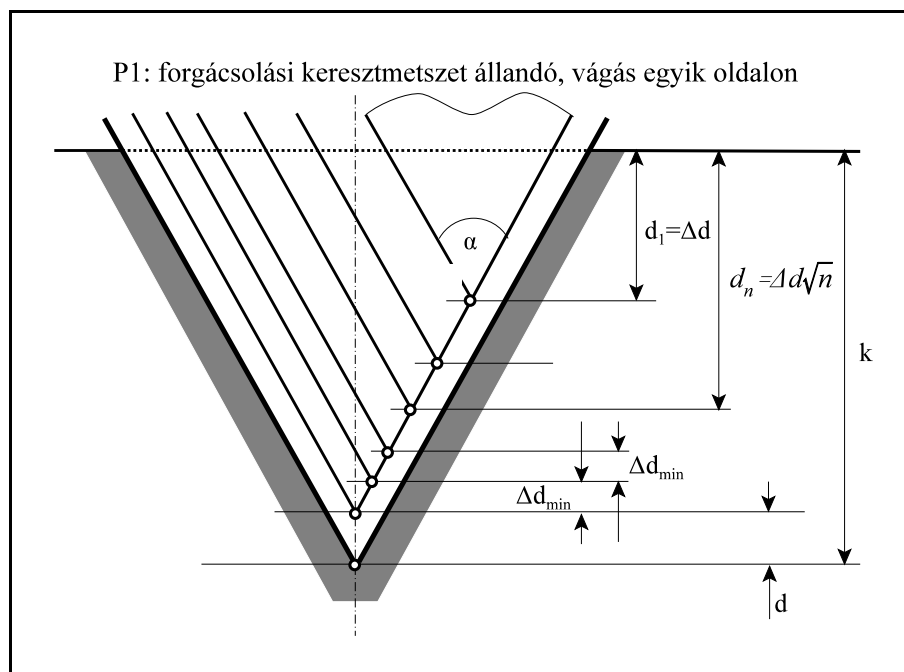
...

18.2.7 A menetvágó ciklus (G76)

A mellékelt ábra egy G76 típusú menetvágó ciklus mozgását mutatja.



18.2.7 -2 ábra



18.2.7 -3 ábra

1. Megadási módszer:

G18 (Z-X sík)

G76 P (n) (r) (α) **Q** (Δd_{\min}) **R** (d)

G76 X_p(U) Z_p(W) P (k) **Q** (Δd) **R** (i) **F(E)(L)**

G19 (Y-Z sík)

G76 P (n) (r) (α) **Q** (Δd_{\min}) **R** (d)

G76 Y_p(V) Z_p(W) P (k) **Q** (Δd) **R** (i) **F(E)(L)**

G17 (X-Y sík)

G76 P (n) (r) (α) **Q** (Δd_{\min}) **R** (d)

G76 X_p(U) Y_p(V) P (k) **Q** (Δd) **R** (i) **F(E)(L)**

utasítással történik.

Az alább felsorolt paraméterek az első **G76 P** (n) (r) (α) **Q** (Δd_{\min}) **R** (d) mondat bemenő adatai:

n: *a símító ciklusok száma* (n=01...99)

Ez az érték öröklődő, és addig nem változik, amíg át nem írjuk. Ha a símító ciklusok számát nem adjuk meg a programban, akkor a vezérlő azt az N1608 Count Fin paraméterről veszi.

r: *a letörés mértéke* (r=01...99)

A menetből való kifizáskor a vezérlő az N1607 Chmfr Ang paraméteren megadott szögben emeli ki a szerszámot. "r" segítségével a kiemelési letörés hosszát adjuk meg a menetemelkedés függvényében. A szakasz hossza

$$r \cdot L / 10$$

ahol: L: a programozott menetemelkedés

Ez az érték öröklődő, és addig nem változik, amíg át nem írjuk. Ha a letörés mértékét nem adjuk meg a programban, akkor a vezérlő azt az N1606 ThrdChmfr paraméterről veszi.

α : *a menetvágó kés élszöge* fokban (α =01...99)

Ez az érték öröklődő, és addig nem változik, amíg át nem írjuk. Ha a menetvágó szerszám élszögét nem adjuk meg a programban, akkor a vezérlő azt az N1612 Tool Tip Angle paraméterről veszi.

*n, r és α értékét a P címen adhatjuk meg egyszerre. Mivel mindegyik értéket egy kétjegyű szám fejez ki, P címre egy hatjegyű számot kell írni. Például, ha a símító ciklusok száma n=2, a letörés értéke 1.5L (r=15), és 60°-os késsel dolgozunk, akkor P értéke: **P021560**.*

Δd_{\min} : *a minimális fogás értéke* (mindig *sugárban* értendő, *pozitív* szám)

Ha a menetvágás során az n-edik ciklusban a fogásvétel értéke $d_n - d_{n-1} < \Delta d_{\min}$, akkor a fogásvétel értékét mindig Δd_{\min} értékre korlátozza be. Ez az érték öröklődő, és addig nem változik, amíg át nem írjuk. Ha a minimális fogás értékét nem adjuk meg a programban, akkor a vezérlő azt az N1609 Min Thrd Cut paraméterről veszi.

d: *símítási ráhagyás* (mindig *sugárban* értendő, *pozitív* szám)

Ez az érték öröklődő, és addig nem változik, amíg át nem írjuk. Ha a símítási ráhagyás értékét nem adjuk meg a programban, akkor a vezérlő azt az N1610 Fin Allow paraméterről veszi.

A vezérlő a G76 kódra akkor végzi el a fentebb felsorolt paraméterek átvételét, ha a G76 mondatban nincs sem X(Z, Y) U(W, V), sem Z(Y, X) W(V, U) cím kitöltve.

Az alábbi paraméterek a második **G76 X(U) Z(W) R(i) P(k) Q(Δd) F(E)(L)** mondat bemenő adatai:

X_p(Z_p, Y_p): a "D" pontnak a sík 2. tengelyén megadott abszolút koordinátája

U(W, V): az AD pont távolsága inkrementálisan

Z_p(Y_p, X_p): a "D" pontnak a sík 1. tengelyén megadott abszolút koordinátája

W(V, U): az AD pont távolsága inkrementálisan

i: **a kúposág mértéke** (mindig **sugárban** értendő)
Ha i=0, vagy az R címet nem töltjük ki, hengeres menetet vág.

k: **a menet mélysége** (mindig **sugárban** értendő, **pozitív** szám)

Δd: **az első fogás mélysége** (mindig **sugárban** értendő, **pozitív** szám)

L: **a menet emelkedése**
Programozása megegyezik a G33-nál elmondottakkal. Az **F címre** írt érték **menetemelkedést**, az **E címre** írt érték **inchenkénti menetszámot** jelöl.

Menetvágást csak a fenti módon kitöltött mondatra végez, ami azt jelenti, hogy az X(Z, Y) U(W, V), és a Z(Y, X) W(V, U) címeket ki kell tölteni. Ha egyik koordinátacím sincs kitöltve, akkor a mondatot paraméterbeállító mondatnak értelmezi.

A ciklus végrehajtása során a **fogásvételt gyorsmenettel** hajtja végre, ha a ciklus alatt **G0** kód van érvényben, illetve a megörökölt **előtolással**, ha a ciklus alatt **G1** kód van érvényben.

A C és D pontok között előtoló mozgást végez, az F(E) címen megadott L menetemelkedésnek megfelelően. A többi szakaszt gyorsmenettel hajtja végre.

A menetet mindig az ábra szerinti módszerrel, az egyik oldalon vágja a

$$d_n = \Delta d \sqrt{n}$$

összefüggés szerint úgy, hogy a forgácsolási keresztmetszet állandó legyen.

A **menet végén** mindig **letörést** vág a beállított paramétereknek megfelelően.

Az X(Z, Y) U(W, V), Z(Y, X) W(V, U) címeken programozott mozgás iránya határozza meg a fogásvétel, illetve a menetvágás irányát, az R cím előjele pedig a kúp lejtési irányát.

STOP hatására a G78 kódnál leírtak szerint a programozott letörésnek megfelelően kiemeli a szerszámot, majd a kezdőpontra (A) áll. START gomb hatására a megszakított fogásvételt előlről kezdi.

2. megadási módszer:

G18 (Z-X sík)

G76 X_p(U) Z_p(W) I(i) K(k) D(Δd) A(α) F(E)(L) Q P

G19 (Y-Z sík)

G76 Y_p(V) Z_p(W) K(i) J(k) D(Δd) A(α) F(E)(L) Q P

G17 (X-Y sík)

G76 X_p(U) Y_p(V) J(i) I(k) D(Δd) A(α) F(E)(L) Q P

A vezérlő akkor tekinti a mondatmegadást a **2. módszer szerint**inek, ha a mondatban **K(J, I) cím ki van töltve**.

Az n, r, α, Δd_{min} és d bemenő paramétereket a vezérlő éppúgy figyelembe veszi a 2. módszerrel történő menetmegadásnál is, mint az 1-nél, vagy a **G76 P (n) (r) (α) Q (Δd_{min}) R (d)** paraméter-beállító mondatl, vagy paraméterről.

“A” címen (α) az 1. módszerhez hasonlóan a menetvágó kés élszögét lehet megadni. A különbség az, hogy amíg az 1. módszernél csak 1°-os felbontással, egész számmal tudjuk megadni, addig a 2. módszernél törtszámmal is. Ha “A” cím nincs kitöltve α értékét az N1612 Tool Tip Angle paraméterről veszi.

Az i, k, Δd, L adatok értelmezése megegyezik az 1. megadási módszernél leírtakkal. A további címek értelmezése:

Q: *a menet kezdetének a jeladó nullimpulzusától számított szögértéke °-ban* megadva. A cím értelmezése megegyezik a G33-nál elmondottakkal.

P: *a menetvágás módszere.*

Négyféle fogásvételi módszer között lehet választani az alábbi ábrák alapján

P1: forgácsolási keresztmetszet állandó, vágás egyik oldalon

P2, P5: forgácsolási keresztmetszet állandó, vágás mindkét oldalon

P3: fogásvétel állandó vágás egyik oldalon

P4: fogásvétel állandó vágás mindkét oldalon

Mintapélda

G18 . . .

G97 S1000 M3

G0 X36 Z4

G76 P010560 Q0.3 R0.2

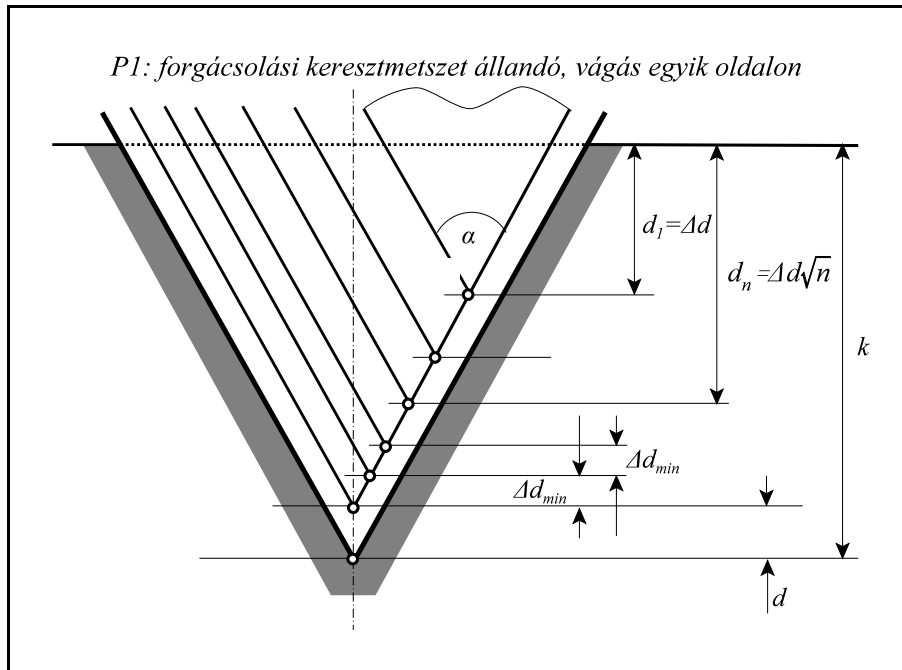
G76 X24 Z-50 P3 Q1 R-4 F3 (1. megadási módszer)

vagy

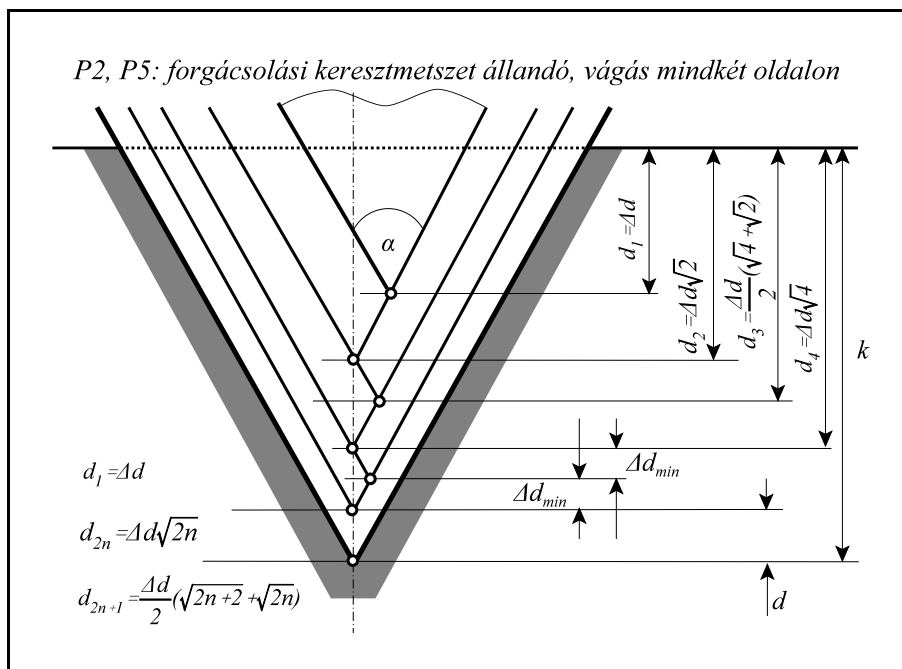
G76 X24 Z-50 K3 D1 I-4 A60 F3 P2 (2. megadási módszer)

G0 X100 Z50

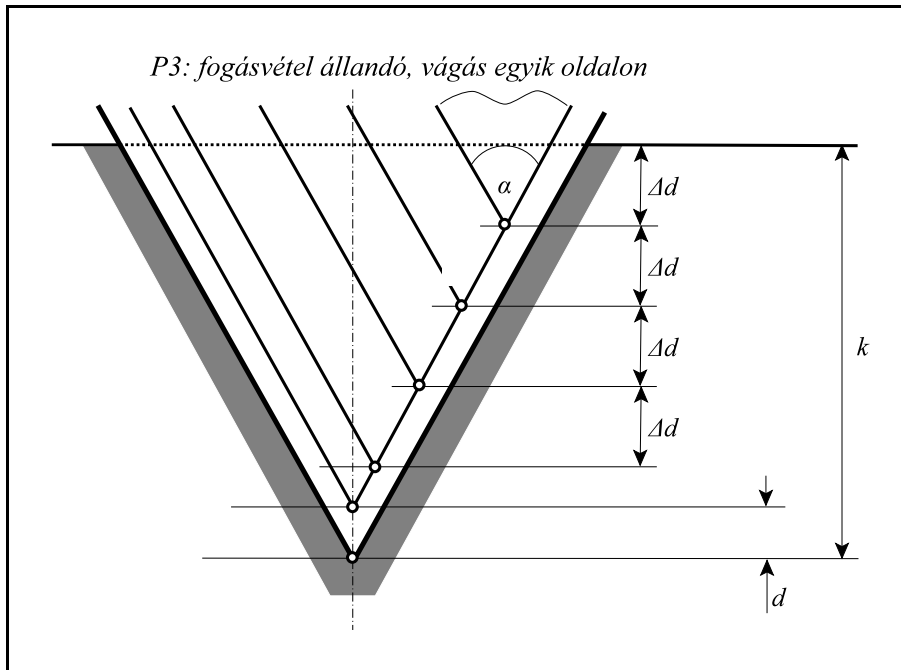
. . .



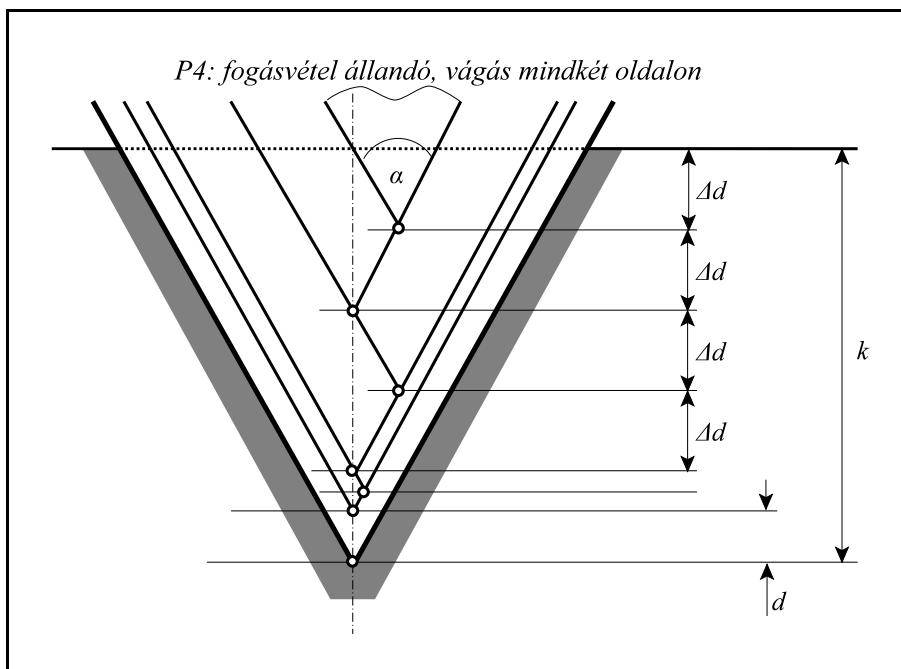
18.2.7 -5 ábra



18.2.7 -6 ábra



18.2.7 -7 ábra



18.2.7 -8 ábra

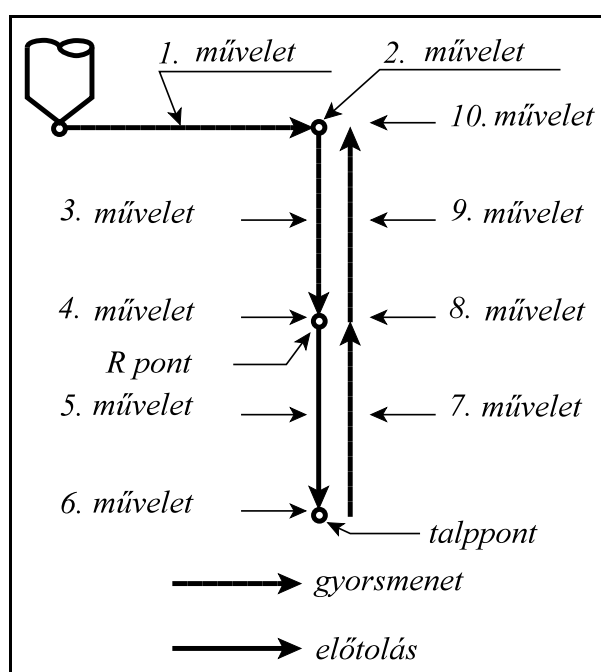
19 Fúróciklusok

A fúróciklusokat a következő műveletekre lehet bontani:

1. művelet: pozicionálás a kiválasztott síkban
2. művelet: tevékenység pozicionálás után
3. művelet: gyorsmeneti mozgás az R pontig (megközelítési pont)
4. művelet: tevékenység az R ponton
5. művelet: fúrás a talppontig
6. művelet: tevékenység a talpponton
7. művelet: visszahúzás az R pontig
8. művelet: tevékenység az R ponton
9. művelet: gyorsmeneti visszahúzás a kiindulási pontig
10. művelet: tevékenység a kiindulási ponton

R pont, megközelítési pont: A szerszám a munkadarabot eddig a pontig közelíti meg gyorsmeneti mozgással.

Kiindulási pont: A fúrótengelynek az a pozíciója, amelyet a ciklus indítása előtt felvesz.



19-9 ábra

A fenti műveletek a fúróciklusok általános leírását adják, a konkrét esetekben műveletek elmaradhatnak.

A fúróciklusoknak van **pozicionálási síkja**, és **fúrótengelye**. A fúrótengelyt a G17, G18, G19 síkválasztó utasítások jelölik ki. Az összes többi tengelyt a pozicionálási síkban mozgatja.

G kód	pozícionálási sík	fúrótengely
G17	$X_p Y_p$ sík	Z_p
G18	$Z_p X_p$ sík	Y_p
G19	$Y_p Z_p$ sík	X_p

ahol: X_p : X, vagy azzal párhuzamos tengely

Y_p : Y, vagy azzal párhuzamos tengely

Z_p : Z, vagy azzal párhuzamos tengely

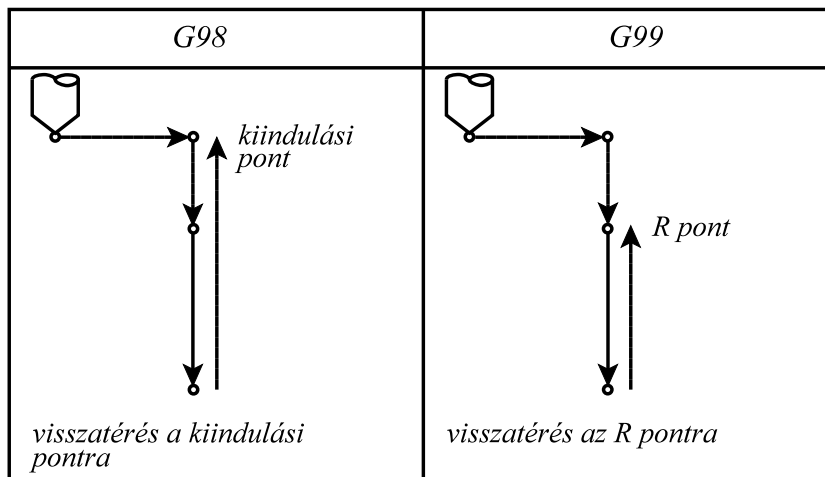
U, V, W tengelyeket akkor tekinti párhuzamos tengelyeknek, ha az N0103 Axis to Plane paraméteren annak vannak definiálva.

Ha homlokfúrást akarunk programozni, ahol a fúrótengely a Z, válasszuk a G17-es síkot, ha oldalfúrást, ahol a fúrótengely az X, válasszuk a G19-es síkot.

A fúróciklusok konfigurálása a **G98** és **G99** utasításokkal lehetséges:

G98: a számszám a fúróciklus során **a kiindulási pontig kerül visszahúzásra**. Alaphelyzet, amelyet a vezérlő bekapcsolás, reset vagy a ciklus üzemmód törlése után vesz fel.

G99: a számszám a fúróciklus során **az R pontig kerül visszahúzásra**, következésképp ekkor a 9., 10. művelet elmarad.



19-10 ábra

A fúróciklusok kódjai: G83.1, G84.1, G86.1, G81, ..., G89

Ezek a kódok bekapcsolják a ciklus üzemmódot, amely lehetővé teszi a ciklusváltóók öröklődését.

A G80 kód kikapcsolja a ciklus üzemmódot és törli az eltárolt ciklusváltóókat.

A fúróciklusokban használt címek és értelmezésük:

G17	G_	X_p_	Y_p_	q_	I_	J_	Z_p_	R_	Q_	E_	P_	F_	S_	L_
G18	G_	Z_p_	X_p_	q_	K_	I_	Y_p_	R_	Q_	E_	P_	F_	S_	L_
G19	G_	Y_p_	Z_p_	q_	J_	K_	X_p_	R_	Q_	E_	P_	F_	S_	L_

ismétlési szám
fúrési adatok
elmozdulás orsó orientálás után
a furat pozíciója
a fúrás kódja

A fúrás kódja:

Az egyes kódok értelmezését lásd később.

A kódok öröklődnek mindaddig amíg G80 utasítást, vagy az 1-es G kód csoportba (interpolációs csoport: G01, G02, G03, G33) tartozó kódot nem programozunk, illetve törlődik üzemmódváltásra.

Amíg a ciklusállapot be van kapcsolva, a G83.1, G84.1, G86.1, G81, ..., G89 utasításokkal, addig az öröklődő ciklusváltozók a különböző típusú fúróciklusok között is átöröklődnek.

A kezdőpont vagy kiindulási pont:

A **kezdőpont a fúrásra kijelölt tengely pozíciója**, amely bejegyzésre kerül:

– ciklusüzemmód bekapcsolásakor. Például:

```
N1 G17 G90 G0 Z200
N2 G81 X0 C0 Z50 R150
N3 X100 C30 Z80
```

esetén a kezdőpont pozíciója $Z=200$ az N2 és N3 mondatban is.

– vagy új fúrótengely kijelölésénél. Például:

```
N1 G17 G90 G0 Z200 W50
N2 G81 X0 C0 Z50 R150
N3 X100 C30 W20 R25
```

N2 mondatban a kezdőpont pozíciója $Z=200$

N3 mondatban a kezdőpont pozíciója $W=50$

Abban az esetben, ha változik a fúrótengely kijelölése R programozása kötelező, ellenkező esetben 2053 Nincs talppont vagy R pont üzenetet ad.

A furat pozíciója: X_p, Y_p, Z_p, q

A beírt koordinátaértékek közül a fúrótengelyt leszámítva a többi koordinátaadatot veszi a furat pozíciójának. Ezek lehetnek **a kiválasztott sík fő tengelyei**, azokkal **párhuzamos tengelyek**, vagy **tetszőleges egyéb, nem fúrásra kijelölt tengely (q: pl. C)**.

A beírt értékek lehetnek inkrementális, vagy abszolút, derékszögű vagy polárkoordinátában megadott értékek, dimenziójuk pedig metrikus vagy inches.

A beírt koordinátaértékekre érvényesek a tükrözési, elforgatási, és léptékezési parancsok.

A furat pozíciójára a vezérlés gyorsmeneti pozicionálással áll rá, függetlenül attól, hogy melyik kód volt érvényben az 1-es csoportból.

Elmozdulás orsó orientálás után: I, J, K

Ha az adott gépen lehetőség van az orsó orientálására, a G76, és G87 kiesztergáló ciklusokban a szerszámot a felülettől eltávolítva lehet visszahúzni, hogy a szerszám hegye ne karcolja azt. Ekkor I, J és K címen lehet megadni hogy a szerszámot milyen irányban távolítsa el a felülettől a vezérlő. A kiválasztott síknak megfelelően értelmezi a vezérlő a címekeket:

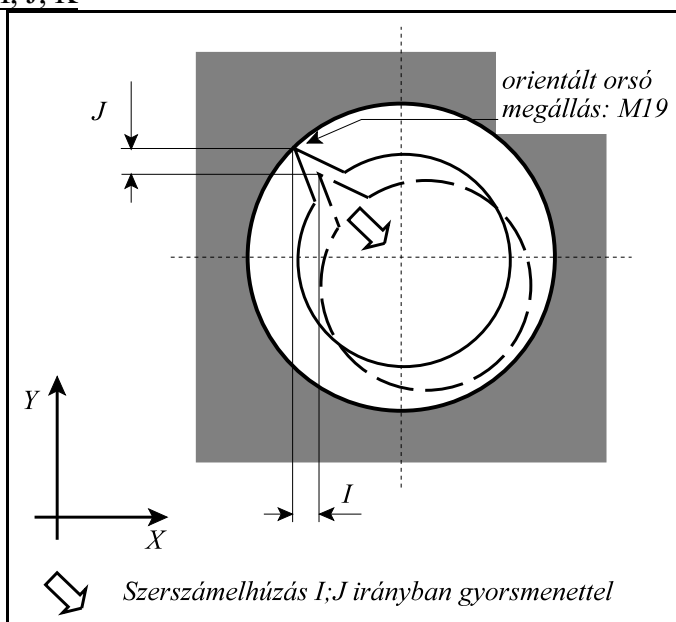
G17: I, J

G18: K, I

G19: J, K

A címek **mindig inkrementális, derékszögű** adatként kerülnek értelmezésre. A cím lehet metrikus és inch-es.

I, J, K adatokra nem érvényesek a tükrözési, elforgatási, vagy léptékezési parancsok. I, J és K öröklődő értékek. G80, vagy az interpolációs csoport kódjai értékeit törlik. Az elhúzás gyorsmenettel történik.

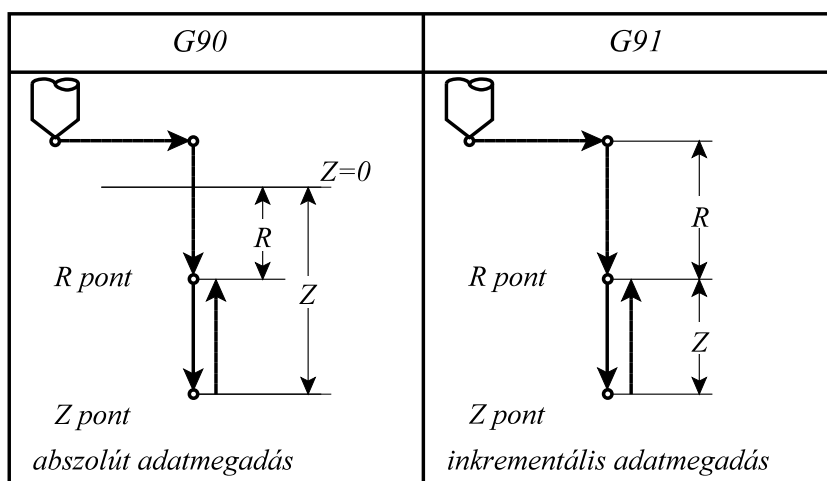


19-11 ábra

Fúrési adatok:

A furat talppontja: X_{p2}, Y_{p2}, Z_p

A furat talppontját a fúró tengely címén kell megadni. A furat talppontjának koordinátája mindig derékszögű adatként kerül értelmezésre. Lehet inch-es, vagy metrikus, abszolút, vagy inkrementális. Ha inkrementálisan adjuk meg a talppont értékét az elmozdulást az R ponttól számítja.



19-12 ábra

A talppont adataira érvényesek a tükrözési és léptékezési parancsok. A talppont adata öröklődő érték. G80, vagy az interpolációs csoport kódjai értékét törlik. A talppontot mindig az érvényes előtolással közelíti meg a vezérlő.

A megközelítési pont, R pont: R

A megközelítési pontot R címen adjuk meg. Az R cím mindig derékszögű adat amely lehet inkrementális és abszolút, metrikus, vagy inch-es. Ha R adat inkrementális, értékét a kiindulási ponttól számítjuk. Az R pont adataira érvényesek a tükrözési és léptékezési adatok. Az R pont adata öröklődik. G80, vagy az interpolációs csoport kódjai értékét törlik. Az R pontot mindig gyorsmeneti mozgással közelíti meg a vezérlő.

A fogásmélység értéke: Q

G83.1-es és G83-as ciklusokban a fogásmélység értéke. Mindig inkrementális, derékszögű, pozitív adat. A fogásmélység értéke öröklődő adat. G80, vagy az interpolációs csoport kódjai értékét törlik. A fogásmélységre nem érvényes a léptékezési parancs.

Segédadat: E

G83.1-es ciklusban a visszahúzás mértéke, illetve a G83-asban pedig a fogásvétel előtt ekkora értékig megy gyorsmenettel. Mindig inkrementális, derékszögű, pozitív adat. A segédadatra nem érvényes a léptékezési parancs. A segédadat értéke öröklődő. G80, vagy az interpolációs csoport kódjai értékét törlik. Ha nem programozták, akkor az N1500 Return Val G73 , illetve N1501 Clearance Val G83 paraméterről veszi a vezérlő a szükséges értéket.

Várakozás: P

A várakozási időt adja meg a furat alján. Megadására a G04-nél elmondott szabályok érvényesek. A várakozás értéke öröklődő. G80, vagy az interpolációs csoport kódjai értékét törlik.

Előtolás: F

Az előtolást határozza meg. Értéke öröklődő. Csak egy másik F adat programozása írja felül, G80 vagy más kód nem törli.

Kihúzási override: I (%)

G85, G89, G84.2, G84.3 ciklusokban a kihúzás alapvetően az F címen programozott előtolással történik. I címen a kihúzási az override-ot százalékos értékben lehet megadni. Ha nincs programozva, az override értékét paraméterről veszi.

Orsó fordulatszám: S

Értéke öröklődő. Csak egy másik S adat programozása írja felül, G80 vagy más kód nem törli.

Ismétlési szám: L

A ciklus ismétlésének számát határozza meg. Értékhatára: 1–99999999. Ha L nincs kitöltve L=1 értéket vesz figyelembe. L=0 esetén a ciklus adatai eltárolódnak, de nem hajtódnak végre. L értéke csak abban a mondatban érvényes, ahol megadtuk.

Példa a fúrési kódok és a ciklusváltozók öröklődésére:

```
N1 G17 G0 Z_ M3
N2 G81 X_ C_ Z_ R_ F_
```

A ciklus üzemmód kezdetén a fúrési adatok (Z, R) meghatározása kötelező)

```
N3 X_
```

Mivel az N2 mondatban a fúrési adatok meg lettek határozva és az N3 mondatban ugyanazok szükségesek, kitöltésük felesleges, azaz G81, Z_, R_, F_ elhagyható. A furat pozíciója csak X

irányban változik, a fúró ebben az irányban mozog, majd ugyanazt a furatot fúrja, mint az N2 mondatban.

```
N4 G82 C_ Z_ P_
```

A furat pozíciója C irányban mozog. A fúrás metódusa G82-nek megfelelően alakul, a talppont Z új értéket vesz fel, a megközelítési pont és előtolás (R, F) N2 mondatból öröklődnek.

```
N5 G80 M5
```

Törli a ciklusüzemmódot és az öröklődő ciklusváltozókat, F kivételével.

```
N6 G85 C_ Z_ R_ P_ M3
```

Mivel az N5 mondatban törölődtek a fúrás adatok G80 utasítás hatására Z, R, és P értékeket újra meg kell adni.

```
N7 G0 X_ C_
```

Törli a ciklusüzemmódot és az öröklődő ciklusváltozókat, F kivételével.

Példák a ciklus ismétlésének használatára:

Ha ugyanolyan furatokat kell egyenlő távolságra készíteni ugyanolyan paraméterekkel, az ismétlési számot az L címen adhatjuk meg. L csak abban a mondatban érvényes, amelyben megadtuk.

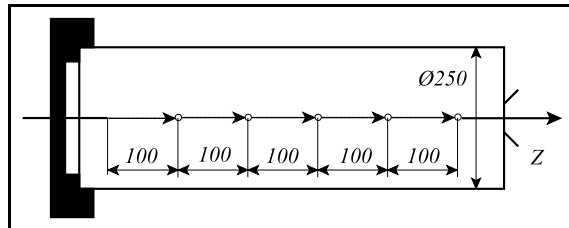
```
N1 G90 G19 G0 X300 Z40 C0 M3
N2 G91 G81 X-40 Z100 R-20 F50 L5
```

A fenti utasítások hatására a vezérlő az Z tengely mentén 100 mm távolságra egymástól 5 db egyforma furatot fúr. Az első furat pozíciója Z=140, C=0. Mivel oldalfúrásról van szó (X tengellyel fúrunk) a G19 síkot választottuk.

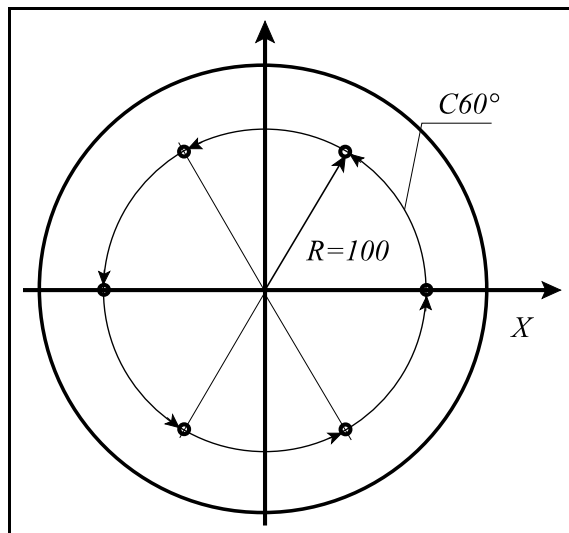
A furat pozíciója G91 hatására inkrementálisan lett megadva.

```
N1 G90 G17 G0 X200 C-60 Z50
N2 G81 CI60 Z-40 R3 F50 L6
```

A fenti utasítások hatására a vezérlő egy 100 mm-es lyukkörön 60 fokként 6 db furatot fúr. Az első furat pozíciója az X=200 C=0 koordinátájú pontra esik. Mivel homlokfúrásról van szó (Z tengellyel fúrunk) a G17 síkot választottuk.



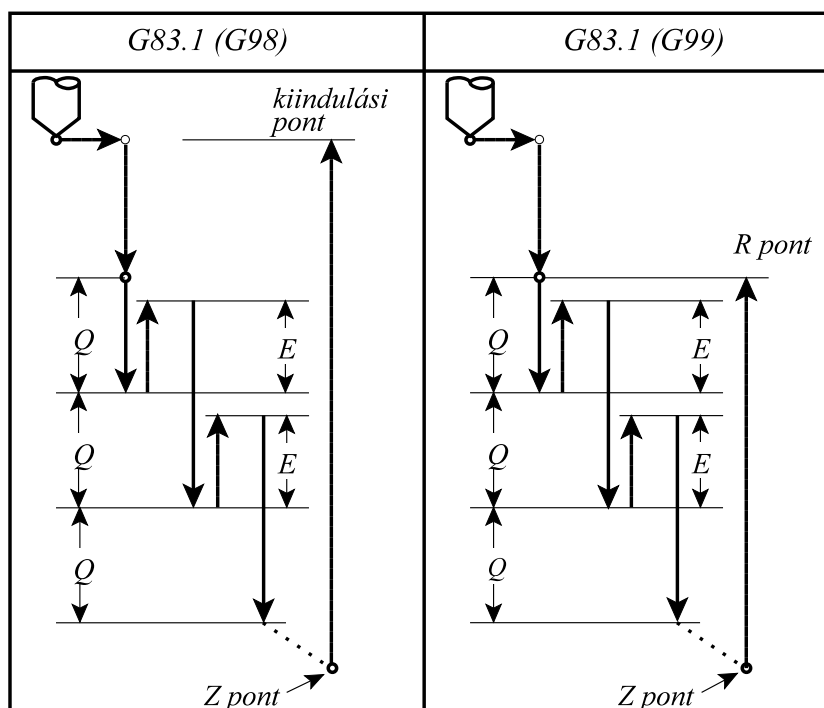
19-13 ábra



19-14 ábra

19.1 A fúróciklusok részletes leírása.

19.1.1 Nagysebességű mélyfúróciklus (G83.1)



19.1.1-1 ábra

A ciklusban felhasznált változók:

G17 **G83.1** X_p__ Y_p__ Z_p__ R__ Q__ E__ F__ L__

G18 **G83.1** Z_p__ X_p__ Y_p__ R__ Q__ E__ F__ L__

G19 **G83.1** Y_p__ Z_p__ X_p__ R__ Q__ E__ F__ L__

A ciklus műveletei:

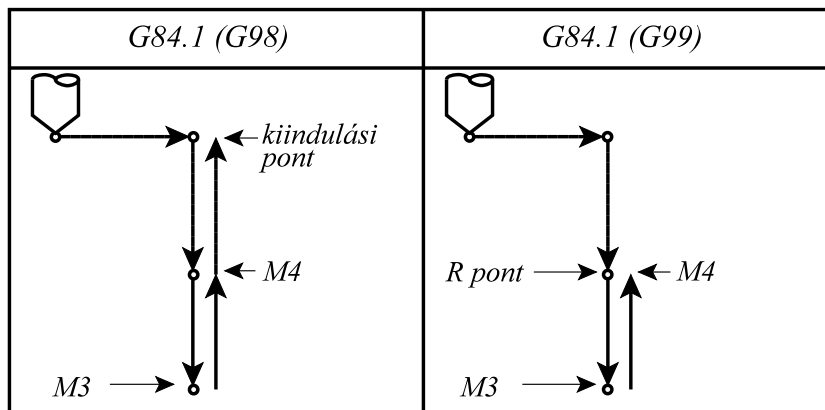
1. művelet: pozicionálás a kiválasztott síkban gyorsmenettel
2. művelet: –
3. művelet: gyorsmeneti mozgás az R pontig (megközelítési pont)
4. művelet: –
5. művelet: fúrás a talppontig F előtolással
6. művelet: –
7. művelet: G99 esetén: visszahúzás az R pontig gyorsmenettel
8. művelet: –
9. művelet: G98 esetén: visszahúzás a kiindulási pontig gyorsmenettel
10. művelet: –

Az 5. fúrási művelet leírása:

- a **Q** címen megadott **fogásmélységet** előtolással belefúrja az anyagba,
- az **E** címen, vagy az N1500 Return Val G73 paraméteren megadott értékkel gyorsmenettel **visszahúz**,
- az előző befúrás talppontjától számítva **Q** mélységet ismételten befúr,
- az **E** címen megadott értékkel gyorsmenettel visszahúz.

Az eljárás a **Z** címen megadott talppontig folytatódik.

19.1.2 Balmenet fúrása kiegyenlítő betéttel (G84.1)



19.1.2-1 ábra

A ciklus csak kiegyenlítőbetéttel ellátott menetfúróval alkalmazható.

A ciklusban felhasznált változók:

G17 **G84.1** X_p__ Y_p__ Z_p__ R__ P__ F__ L__

G18 **G84.1** Z_p__ X_p__ Y_p__ R__ P__ F__ L__

G19 **G84.1** Y_p__ Z_p__ X_p__ R__ P__ F__ L__

A ciklus indítása előtt M4 (óramutató járásával ellentétes) főorsó forgásirányt kell bekapcsolni illetve programozni.

Az előtolás értékét a fúró menetemelkedésének függvényében kell megadni:

– G94 percnkénti előtolás állapotban:

$$F=P \times S$$

ahol: P: a menetemelkedés mm/ford, vagy inch/ford dimenzióban

S: az orsó fordulatszáma ford/perc dimenzióban

– G95 fordulatonkénti előtolás állapotban:

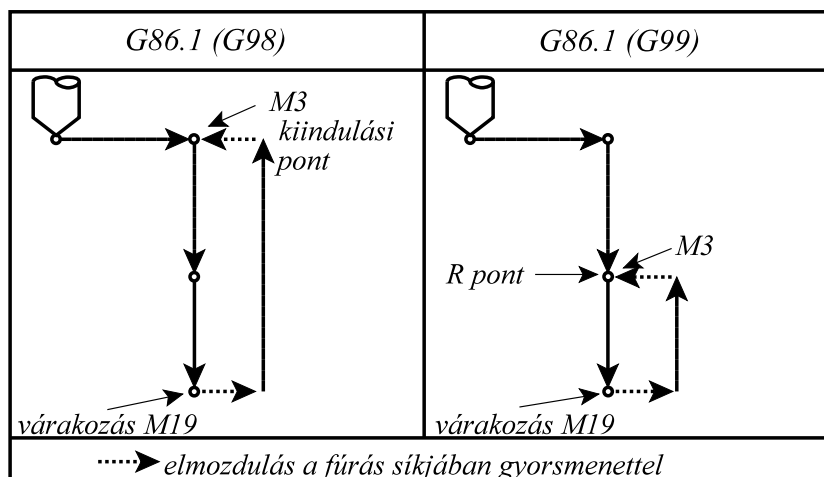
$$F=P$$

ahol: P: a menetemelkedés mm/ford, vagy inch/ford dimenzióban

A ciklus műveletei:

1. művelet: pozicionálás a kiválasztott síkban gyorsmenettel
2. művelet: –
3. művelet: gyorsmeneti mozgás az R pontig (megközelítési pont)
4. művelet: –
5. művelet: fúrás a talppontig F előtolással, **override és stop tiltva**
6. művelet: – várakozás P címen megadott értékkel
– főorsó forgásirányváltás: M3
7. művelet: visszahúzás az R pontig F előtolással, **override és stop tiltva**
8. művelet: főorsó forgásirányváltás: M4
9. művelet: G98 esetén: gyorsmeneti visszahúzás a kiindulási pontig
10. művelet: –

19.1.3 Kiesztergálás automatikus szerszámelhúzással (G86.1)



19.1.3-1 ábra

A G86.1 ciklust csak akkor lehet használni, ha a főorsó orientálás be van építve a szerszámgépbe. A vezérlő számára ezt a tényt az N0607 Spindle Config paraméter ORI bitjének 1 állapota jelzi. Ellenkező esetben 2137 G76, G87 hiba jelzést ad.

Mivel a ciklus a kiesztergálás után főorsó orientálást végez és a szerszámot elhúzza a felülettől I, J és K-n megadott értékkel, a szerszám kihúzásakor nem karcolódik a felület.

A ciklusban felhasznált változók:

G17 **G86.1** X_p__ Y_p__ I__ J__ Z_p__ R__ P__ F__ L__

G18 **G86.1** Z_p__ X_p__ K__ I__ Y_p__ R__ P__ F__ L__

G19 **G86.1** Y_p__ Z_p__ J__ K__ X_p__ R__ P__ F__ L__

A ciklus indítása előtt M3 parancsot kell kiadni.

A ciklus műveletei:

1. művelet: pozicionálás a kiválasztott síkban gyorsmenettel
2. művelet: –
3. művelet: gyorsmeneti mozgás az R pontig (megközelítési pont)
4. művelet: –
5. művelet: kiesztergálás a talppontig F előtolással
6. művelet:
 - várakozás P címen megadott értékkel
 - főorsó orientálás: M19
 - szerszámelhúzás a kiválasztott síkban I, J, K értékkel gyorsmenettel
7. művelet: G99 esetén: visszahúzás az R pontig, gyorsmenettel
8. művelet: G99 esetén
 - szerszámvisszahúzás a kiválasztott síkban I, J, K-n megadott értékkel ellentétesen, gyorsmenettel,
 - főorsó újraindítása M3 irányban
9. művelet: G98 esetén: gyorsmeneti visszahúzás a kiindulási pontig
10. művelet: G98 esetén
 - szerszámvisszahúzás a kiválasztott síkban I, J, K-n megadott értékkel ellentétesen, gyorsmenettel,
 - főorsó újraindítása M3 irányban

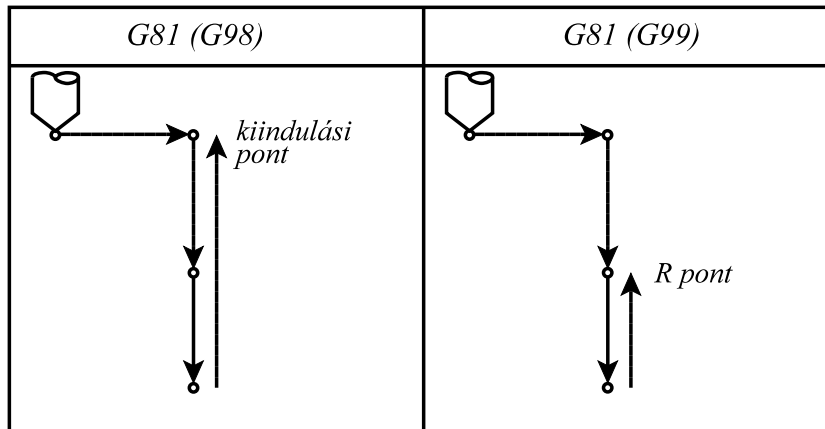
19.1.4 A ciklusállapot kikapcsolása (G80)

A kód hatására a ciklusállapot kikapcsolódik, a ciklusváltozók törlődnek.

Z és R inkrementális 0 értéket vesz fel, a többi változó 0-t.

Ha a **G80** mondatba koordinátákat programozunk, és más utasítást nem adunk, akkor a ciklus bekapcsolása előtt érvényes interpolációs kód (az 1-es G kód csoport, vagy interpolációs csoport) alapján hajtódik végre a mozgás.

19.1.5 Fúróciklus, kiemelés gyorsmenettel (G81)



19.1.5-1 ábra

A ciklusban felhasznált változók:

G17 **G81** X_p__ Y_p__ Z_p__ R__ F__ L__

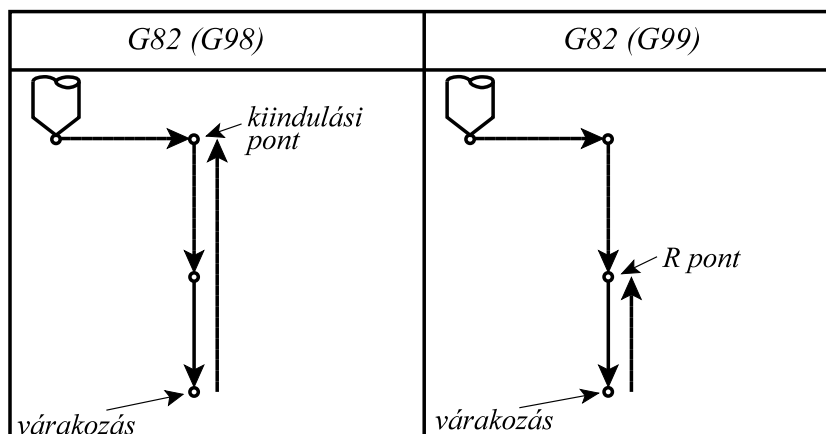
G18 **G81** Z_p__ X_p__ Y_p__ R__ F__ L__

G19 **G81** Y_p__ Z_p__ X_p__ R__ F__ L__

A ciklus műveletei:

1. művelet: pozicionálás a kiválasztott síkban gyorsmenettel
2. művelet: –
3. művelet: gyorsmeneti mozgás az R pontig (megközelítési pont)
4. művelet: –
5. művelet: fúrás a talppontig F előtolással
6. művelet: –
7. művelet: G99 esetén: visszahúzás az R pontig, gyorsmenettel
8. művelet: –
9. művelet: G98 esetén: gyorsmeneti visszahúzás a kiindulási pontig
10. művelet: –

19.1.6 Fúróciklus várakozással, kiemelés gyorsmenettel (G82)



19.1.6-1 ábra

A ciklusban felhasznált változók:

G17 **G82** X_p__ Y_p__ Z_p__ R__ P__ F__ L__

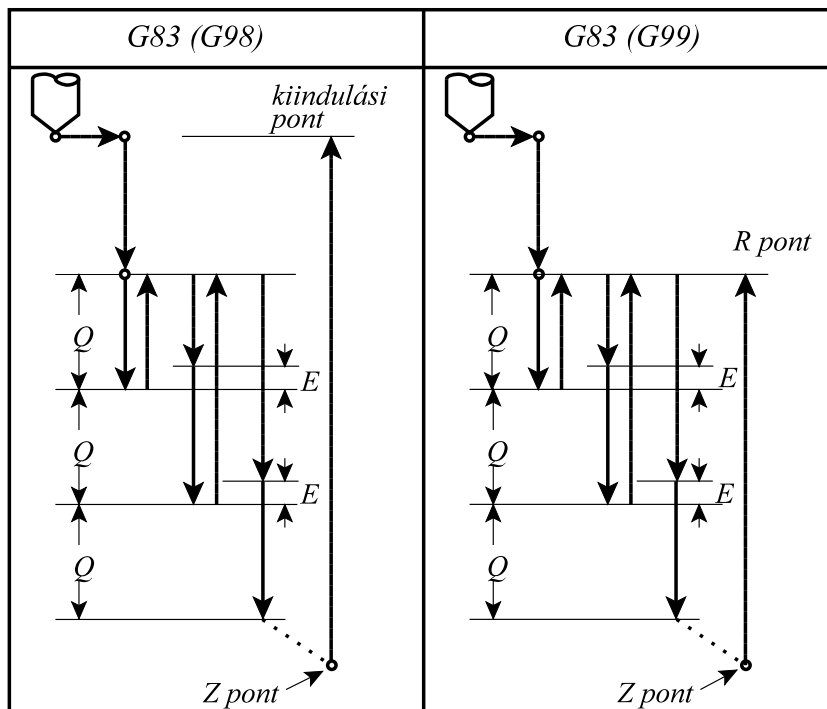
G18 **G82** Z_p__ X_p__ Y_p__ R__ P__ F__ L__

G19 **G82** Y_p__ Z_p__ X_p__ R__ P__ F__ L__

A ciklus műveletei:

1. művelet: pozicionálás a kiválasztott síkban gyorsmenettel
2. művelet: –
3. művelet: gyorsmeneti mozgás az R pontig (megközelítési pont)
4. művelet: –
5. művelet: fúrás a talppontig F előtolással
6. művelet: várakozás P címen megadott ideig
7. művelet: G99: esetén: visszahúzás az R pontig, gyorsmenettel
8. művelet: –
9. művelet: G98 esetén: gyorsmeneti visszahúzás a kiindulási pontig
10. művelet: –

19.1.7 Mélyfúróciklus (G83)



19.1.7-1 ábra

A ciklusban felhasznált változók:

G17 **G83** X_p Y_p Z_p R Q E F L
 G18 **G83** Z_p X_p Y_p R Q E F L
 G19 **G83** Y_p Z_p X_p R Q E F L

A ciklus műveletei:

1. művelet: pozicionálás a kiválasztott síkban gyorsmenettel
2. művelet: –
3. művelet: gyorsmeneti mozgás az R pontig (megközelítési pont)
4. művelet: –
5. művelet: fúrás a talppontig F előtolással
6. művelet: –
7. művelet: G99 esetén: visszahúzás az R pontig gyorsmenettel
8. művelet: –
9. művelet: G98 esetén: visszahúzás a kiindulási pontig gyorsmenettel
10. művelet: –

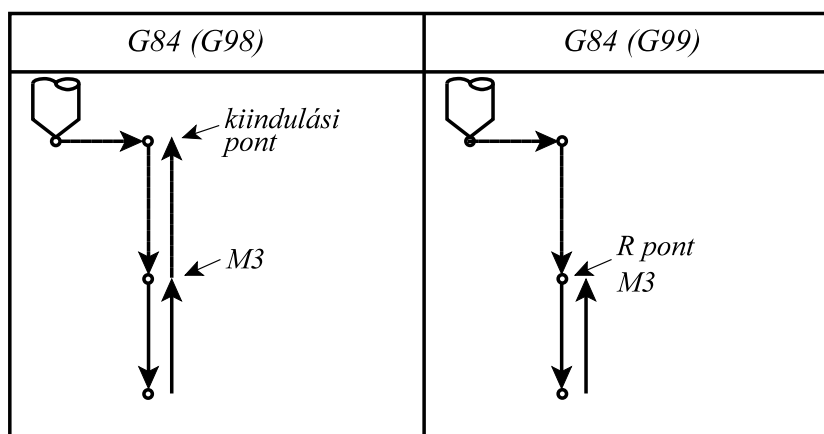
Az 5. fúrási művelet leírása:

- a **Q** címen megadott **fogásmélységet** előtolással belefúrja az anyagba,
- gyorsmenettel visszahúz az R pontig,
- gyorsmenettel **megközelíti** az előző mélységet **“E” távolságig**,
- az előző befúrás talppontjától számítva Q mélységet ismételtlen befúr, F előtolással (elmozdulás E+Q)
- gyorsmenettel visszahúz, az R pontig

Az eljárás a Z címen megadott talppontig folytatódik.

“E” távolságot vagy a programból E címről, vagy az N1501 Clearance Val G83 paraméterről vesz.

19.1.8 Jobbmenet fúrása kiegyenlítő betéttel (G84)



19.1.8-1 ábra

A ciklus csak kiegyenlítőbetéttel ellátott menetfúróval alkalmazható.

A ciklusban felhasznált változók:

G17 **G84** X_p__ Y_p__ Z_p__ R__ P__ F__ L__

G18 **G84** Z_p__ X_p__ Y_p__ R__ P__ F__ L__

G19 **G84** Y_p__ Z_p__ X_p__ R__ P__ F__ L__

A ciklus indítása előtt M3 (óramutató járásával megegyező) főorsó forgásirányt kell bekapcsolni.

Az előtolás értékét a fúró menetemelkedésének függvényében kell megadni:

– G94 percenkénti előtolás állapotban:

$$F=P \times S$$

ahol: P: a menetemelkedés mm/ford, vagy inch/ford dimenzióban

S: az orsó fordulatszáma ford/perc dimenzióban

– G95 fordulatonkénti előtolás állapotban:

$$F=P$$

ahol: P: a menetemelkedés mm/ford, vagy inch/ford dimenzióban

A ciklus műveletei:

1. művelet: pozicionálás a kiválasztott síkban gyorsmenettel
2. művelet: –
3. művelet: gyorsmeneti mozgás az R pontig (megközelítési pont)
4. művelet: –
5. művelet: fúrás a talppontig F előtolással, **override és stop tiltva**
6. művelet: – várakozás P címen megadott értékkel
– főorsó forgásirányváltás: M4
7. művelet: visszahúzás az R pontig F előtolással, **override és stop tiltva**
8. művelet: főorsó forgásirányváltás: M3
9. művelet: G98 esetén: gyorsmeneti visszahúzás a kiindulási pontig
10. művelet: –

19.1.9 Menetfűrés kiegyenlítő betét nélkül (G84.2, G84.3)

A kiegyenlítő betét nélküli menetfűró ciklusok csak olyan gépeken alkalmazhatóak, ahol az orsóra pozíciójeladó van szerelve, és az orsójárat visszacsatolható pozíció szabályzásra. (N0607 Spindle Config paraméter INX bitje 1). Ellenkező esetben 2138 Főorsó nem indexelhető hibaüzenetet küld.

Menetfűrés esetén a fűrőtengely előtolása és az orsó fordulat hányadosának egyenlőnek kell lennie a menetfűró menetemelkedésével. Másképpen fogalmazva menetfűrésnél ideális esetben az alábbi hányadosnak pillanatról pillanatra állandónak kell lenni: $F=P/S$

ahol: P: a menetemelkedés (mm/ford, vagy inch/ford),
F: előtolás (mm/min, vagy inch/min),
S: orsó fordulatszám (ford/min).

A G84.1-es balmenet, és a G84-es jobbmenet fűróciklusban az orsó fordulatszáma és a fűrőtengely előtolása egymástól teljesen függetlenül vezérlődik. A fenti feltétel ennek megfelelően nem teljesülhet pontosan. Különösen igaz ez a furat alján, ahol a fűrőtengely előtolásának és az orsó fordulatszámának egymással szinkronban kellene lelassulnia és megállnia, majd az ellenkező irányban felgyorsulnia. Ez a feltétel egyáltalán nem tartható vezérléstechnikailag a fenti esetben. A fenti problémát úgy lehet kikerülni, hogy a menetfűrőt egy rugós kiegyenlítőbetéttel tesszük be az orsóba, amely kiegyenlíti az F/S hányados értékében bekövetkező ingadozást.

Más a vezérlés elve a kiegyenlítőbetét kiküszöbölését lehetővé tevő G84.2, G84.3 fűróciklusoknál. Ezeknél a vezérlő folyamatosan gondoskodik, hogy az F/S hányados pillanatról pillanatra állandó legyen.

Vezérléstechnikailag az előző esetben a vezérlő csak az orsó fordulatszámát szabályozza, míg az utóbbiban annak pozícióját is. A G84.2, G84.3 ciklusokban a fűrőtengely és az orsó mozgását lineáris interpolációval kapcsolja össze. Ezzel a módszerrel a gyorsítási és lassítási szakaszokban is biztosítható az F/S hányados állandósága.

G84.2: jobbmenet fűrésa kiegyenlítő betét nélkül

G84.3: balmenet fűrésa kiegyenlítő betét nélkül

A ciklusban felhasznált változók:

G17 **G84.** X_p Y_p Z_p R P F I S L

G18 **G84.** Z_p X_p Y_p R P F I S L

G19 **G84.** Y_p Z_p X_p R P F I S L

A ciklus végén a főorsó indexelt (pozíciószabályozó hurok zárt) állapotban marad, szükség esetén újraindításáról a programozónak kell gondoskodni.

Az előtolás értékét a fűró menetemelkedésének függvényében kell megadni:

– G94 percenkénti előtolás állapotban:

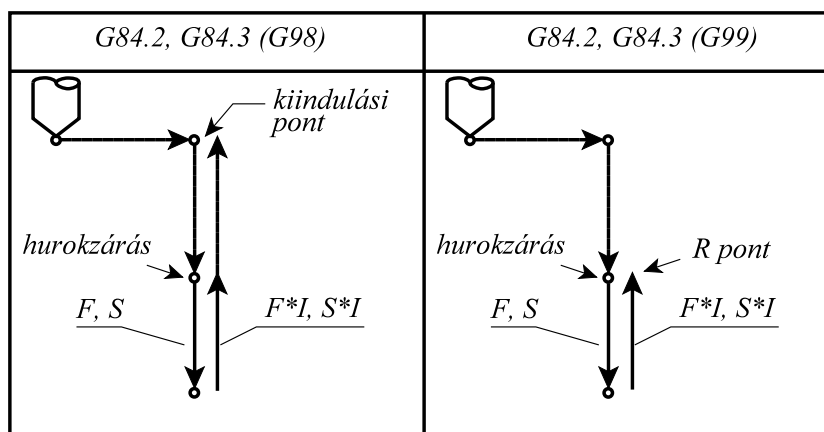
$$F=P \times S$$

ahol: P: a menetemelkedés mm/ford, vagy inch/ford dimenzióban
S: az orsó fordulatszáma ford/perc dimenzióban

– G95 fordulatonkénti előtolás állapotban:

$$F=P$$

ahol: P: a menetemelkedés mm/ford, vagy inch/ford dimenzióban



19.1.9-1 ábra

A ciklus műveletei G84.2 estén:

1. művelet: pozicionálás a kiválasztott síkban gyorsmenettel
2. művelet: -
3. művelet: gyorsmeneti mozgás az R pontig (megközelítési pont)
4. művelet: Ha az N1503 Drilling Cycles Config. paraméter TSC bitje:
 - =0: az orsót nem orientálja, hanem csak a pozíciószabályozó hurkot zárja. A hurokzárás kódját az N0823 M Code for Closing S Loop paraméter határozza meg. Ezt a kódot adja át a PLC-nek. (Végrehajtása gyorsabb)
 - =1: a menetfrúrás megkezdése előtt az orsót orientálja és M19 parancsot ad át a PLC-nek. (Visszatál a menetbe)
5. művelet: lineáris interpoláció a fúrótengely és az orsó között óramutató járásával megegyező (+) G84.2, illetve ellentétes G84.3 orsó forgásirányban
6. művelet: várakozás a P címen megadott értékkel
7. művelet: lineáris interpoláció a fúrótengely és a főorsó között óramutató járásával ellentétes (-) főorsó forgásirányban G84.2 esetén, illetve az óramutató járása szerint G84.3 esetén.

A kihúzási override I (%)

A kihúzási előtolás override-ja, akár I címen programoztuk, akár paraméterről kell venni, csak akkor hatásos, ha az N1503 Drilling Cycles Config. paraméter #4 EOE bitjén 1 értékadással engedélyeztük azt.

Ha I címnek nem adunk a mondatban értéket, akkor a 7. műveletben a kihúzási előtolás override-ja az N1506 Extraction Override in Tapping paraméter %-ban megadott értéke lesz. Az így kiszámított sebességre még az előtolás százalékkapcsoló is hatással van, vagyis az előtolás az

$$F_{\text{kihúzás}} = F_{\text{programozott}} * \text{Előtolás Override} * \text{Extraction Override in Tapping} / 100$$

érték lesz.

Gyorsítás kihúzáskor

Kihúzás során az N1507+n Rn Acc in Tapping paraméteren az orsóra beállított gyorsulástól különböző (kisebb) értéket is megadhatunk, ha az N1503 Drilling Cycles Config. paraméter #6 EAE bitjén 1 értékadással engedélyeztük azt. Ekkor a tartományonként különböző N1523+n Rn Acc in Extract (n=1...8) paraméterről veszi a gyorsítás értékét.

8. művelet: -

9. művelet: G98 esetén: gyorsmeneti visszahúzás a kiindulási pontig
 10. művelet: -

19.1.10 Menetfúrás kiegyenlítő betét nélkül, forgácstöréssel (G84.2, G84.3)

A ciklus kódja:

G84.2 Q__: jobbmenet fúrása kiegyenlítő betét nélkül, forgácstöréssel

G84.3 Q__: balmenet fúrása kiegyenlítő betét nélkül, forgácstöréssel

A ciklusban *akkor alkalmaz* a vezérlő *forgácstörést*, ha a *Q címen fogásmélységet* programoztunk.

A ciklusban felhasznált változók:

G17 **G84.** X_p Y_p Z_p R Q E P F S I L

G18 **G84.** Z_p X_p Y_p R Q E P F S I L

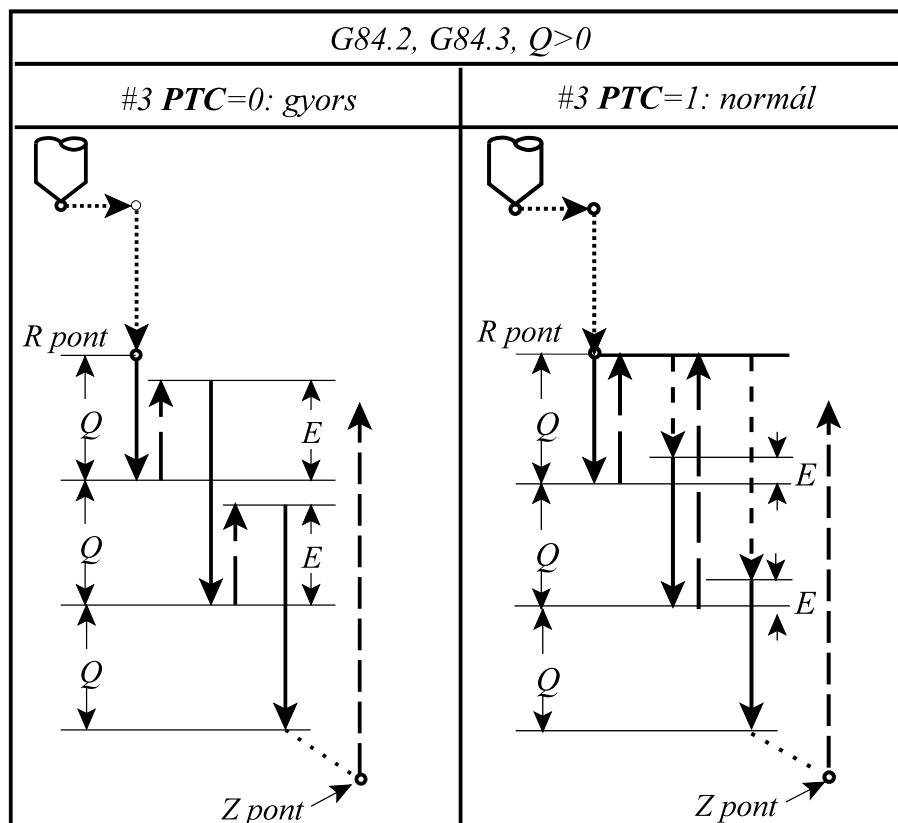
G19 **G84.** Y_p Z_p X_p R Q E P F S I L

A ciklus végén az orsó indexelt (pozíciószabályozó hurok zárt) állapotban marad, szükség esetén újraindításáról a programozónak kell gondoskodni. A **G98**, **G99** kód jelentése ugyanaz, mint az összes többi fúróciklus esetében.

Azt, hogy a *forgácstörés módja* milyen, az N1503 Drilling Cycles Config. paraméter #3 PTC bit-je dönti el. Ha a PTC bit:

=0: (G83.1 szerinti) gyors forgácstörést alkalmaz.

=1: (G83 szerinti) normál forgácstörést alkalmaz, és a kiemelés az R pontig történik.



19.1.10-1 ábra

Az E cím értelmezése

Gyors forgácstörés esetén (PTC=0) **E** címen adhatjuk meg a **szerszám visszahúzásának mértékét**. Ha E címet nem töltöttük ki a vezérlő az N1504 Return Val in Tapping paraméterről veszi a távolságot.

Normál forgácstörés esetén (PTC=1) **E** címen adhatjuk meg a visszatérés utáni **megközelítési távolságot**. Ha E címet nem töltöttük ki, a vezérlő az N1505 Clearance Val in Tapping paraméterről veszi a távolságot.

A kihúzási override I (%)

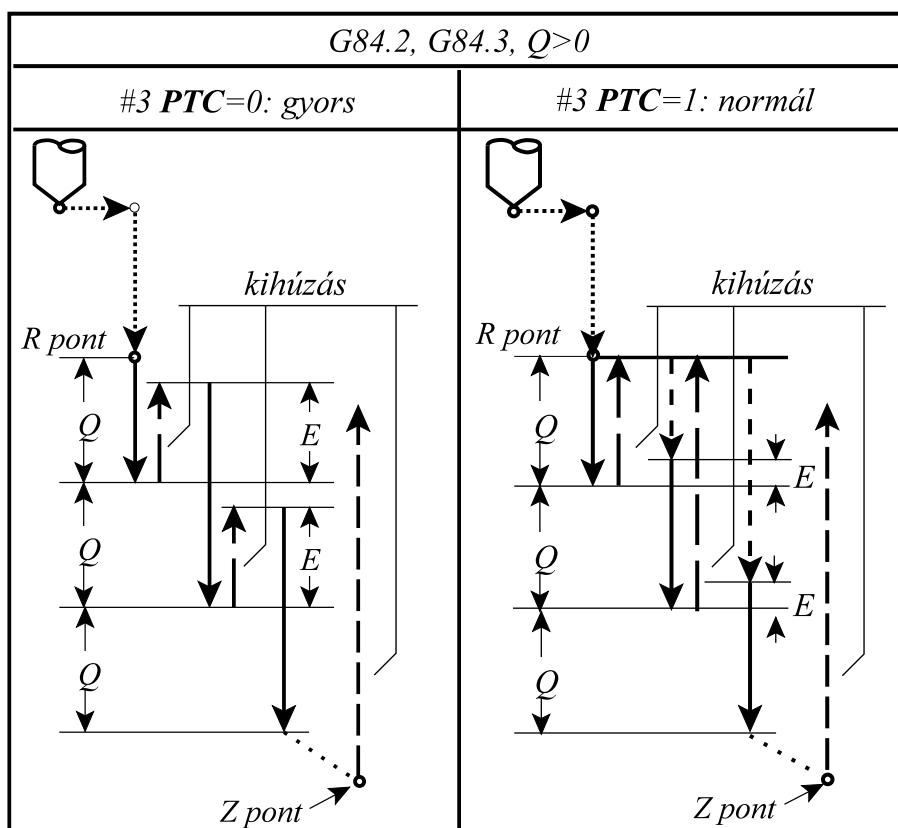
Az összes kihúzási szakaszban a kihúzási előtolás override-ja, akár I címen programoztuk, akár paraméterről kell venni, csak akkor hatásos, ha az N1503 Drilling Cycles Config. paraméter #4 EOE bitjén 1 értékadással engedélyeztük azt.

Ha I címnek nem adunk a mondatban értéket, akkor a kihúzási előtolás override-ja az N1506 Extraction Override in Tapping paraméter %-ban megadott értéke lesz. Az így kiszámított sebességre még az előtolás százalékkapcsoló is hatással van, vagyis az előtolás az

Fkihúzás = Fprogramozott * Előtolás Override * Extraction Override in Tapping/100 érték lesz.

Gyorsítás kihúzáskor

Kihúzás során az N1507+n Rn Acc in Tapping paraméteren az orsóra beállított gyorsulástól különböző (kisebb) értéket is megadhatunk, ha az N1503 Drilling Cycles Config. paraméter #6 EAE bitjén 1 értékadással engedélyeztük azt. Ekkor a tartományonként különböző N1523+n Rn Acc in Extract (n=1...8) paraméterről veszi a gyorsítás értékét.

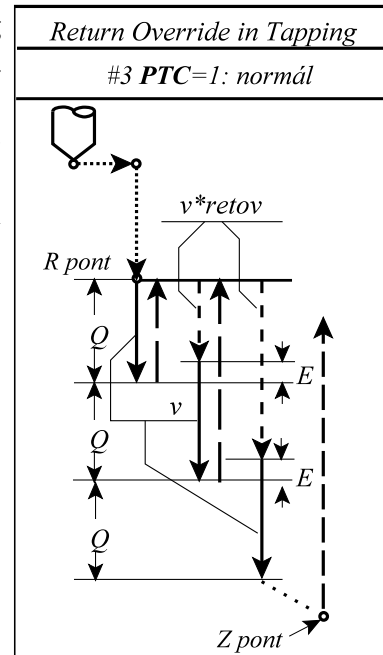


19.1.10-2 ábra

Visszatérési override (%) normál forgácstörésnél

Normál forgácstörés alkalmazása **esetén** (az N1503 Drilling Cycles Config. paraméter #3 PTC bitje 1) a **visszatérés** során, a programozott F-re külön, 100%-tól eltérő **override**-ot alkalmazhatunk, ha az N1503 Drilling Cycles Config. paraméter #5 ROE bitjén 1 értékadással engedélyeztük azt.

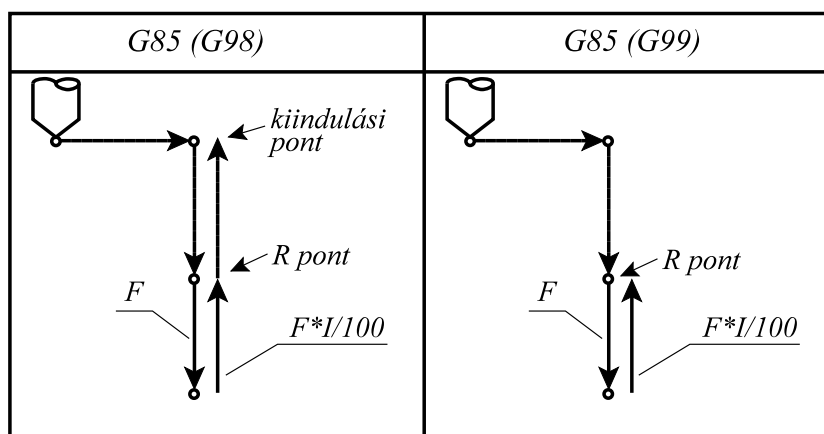
Ekkor az N1507 Return Override in Tapping paraméter %-ban megadott értékéről veszi az override értékét.



19.1.10-3 ábra

A ciklus további bemenő paramétereire az előző fejezetben elmondottak érvényesek.

19.1.11 Fúróciklus, kiemelés előtolással (G85)



19.1.11-1 ábra

A ciklusban felhasznált változók:

G17 **G85** X_p Y_p Z_p R F I L

G18 **G85** Z_p X_p Y_p R F I L

G19 **G85** Y_p Z_p X_p R F I L

A ciklus műveletei:

1. művelet: pozicionálás a kiválasztott síkban gyorsmenettel
2. művelet: –
3. művelet: gyorsmeneti mozgás az R pontig (megközelítési pont)
4. művelet: –
5. művelet: fúrás a talppontig F előtolással
6. művelet: –
7. művelet: visszahúzás az R pontig, $F \cdot I / 100$ előtolással

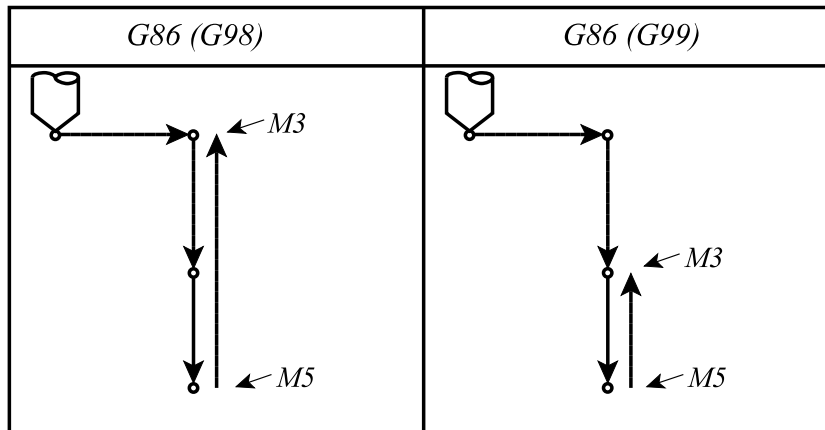
A kihúzási override I (%)

Ha I címnek nem adunk a mondatban értéket, akkor a 7. műveletben a visszahúzási előtolás override-ja az N1502 Extraction Override in G85, G89 paraméter értéke lesz. Az így kiszámított sebességre még az előtolás százalékkapcsoló is hatással van, vagyis az előtolás az

Fkihúzás = Fprogramozott * Előtolás Override * Extraction Override in G85, G89/100 érték lesz.

8. művelet: –
9. művelet: G98 esetén: gyorsmeneti visszahúzás a kiindulási pontig
10. művelet: –

19.1.12 Fúróciklus, gyorsmeneti kiemelés álló főorsóval (G86)



19.1.12-1 ábra

A ciklusban felhasznált változók:

G17 **G86** X_p Y_p Z_p R F L

G18 **G86** Z_p X_p Y_p R F L

G19 **G86** Y_p Z_p X_p R F L

A ciklus indításakor M3-as forgásirányt kell adni a főorsónak.

A ciklus műveletei:

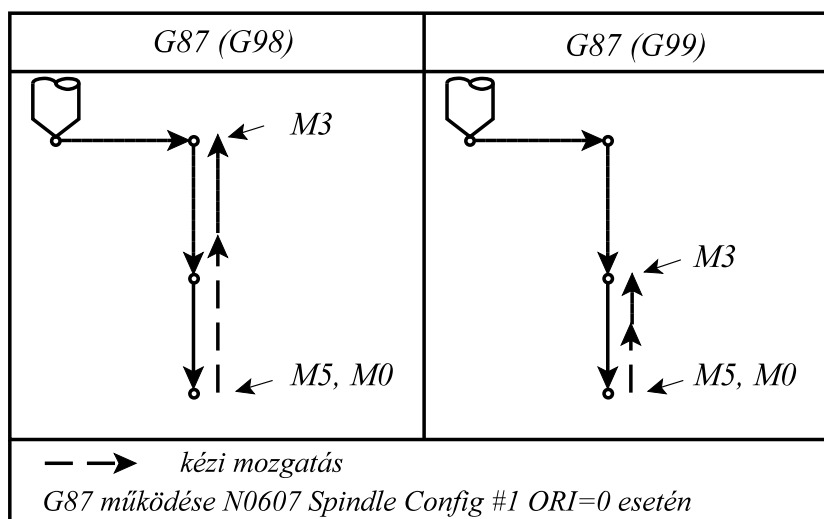
1. művelet: pozicionálás a kiválasztott síkban gyorsmenettel
2. művelet: –
3. művelet: gyorsmeneti mozgás az R pontig (megközelítési pont)
4. művelet: –
5. művelet: fúrás a talppontig F előtolással
6. művelet: főorsó leállítás: M5
7. művelet: G99 esetén: visszahúzás az R pontig, gyorsmenettel
8. művelet: G99 esetén: főorsó újraindítás: M3
9. művelet: G98 esetén gyorsmeneti visszahúzás a kiindulási pontig
10. művelet: G98 esetén: főorsó újraindítás: M3

19.1.13 Kézi működtetés a talpponton/ Kiesztergálás visszafelé (G87)

A ciklust a vezérlő kétféleképp hajtja végre:

A. Fúróciklus, kézi működtetés a talpponton

Abban az esetben, ha **a gépen nincs kiépítve az orsó orientálás lehetősége**, azaz az N0607 Spindle Config paraméter #1 ORI bitje 0, a vezérlő az A eset szerint jár el.



19.1.13-1 ábra

A ciklusban felhasznált változók:

G17 **G87** X_p Y_p Z_p R F L

G18 **G87** Z_p X_p Y_p R F L

G19 **G87** Y_p Z_p X_p R F L

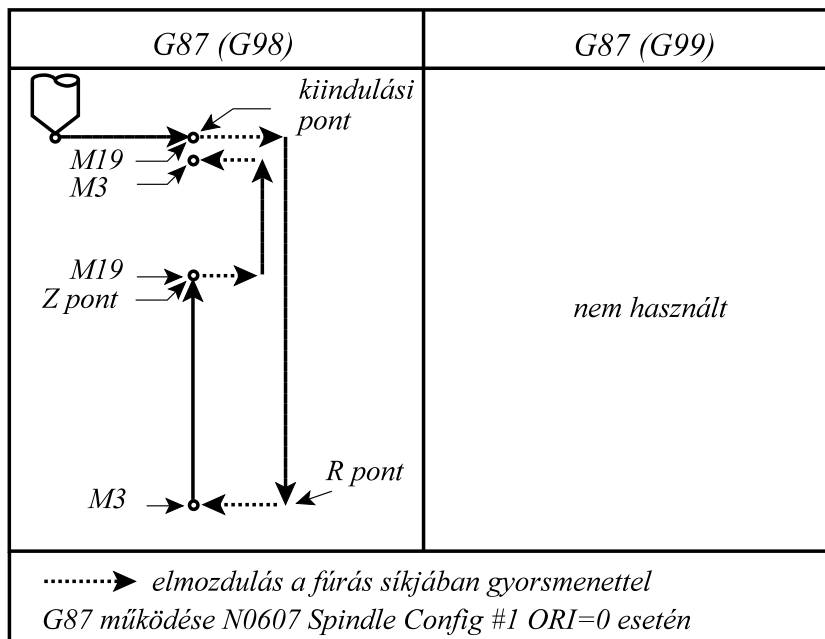
A ciklus indításakor M3-as forgásirányt kell adni a főorsónak.

A ciklus műveletei:

1. művelet: pozicionálás a kiválasztott síkban gyorsmenettel
2. művelet: –
3. művelet: gyorsmeneti mozgás az R pontig (megközelítési pont)
4. művelet: –
5. művelet: fúrás a talppontig F előtolással
6. művelet: – orsó leállítás: M5
– a vezérlő STOP állapotot vesz fel (M0), ahonnan a kezelő valamelyik kézi működtető üzembe (MOZGATÁS, LÉPTETÉS, KÉZIKERÉK) kilépve kézzel működtetheti a gépet, vagyis elhúzhatja a szerszám hegyét a furat felületétől, és kihúzhatja a szerszámot a furatból. Ezután visszalépve AUTOMATA üzembe, startra továbbmegy a megmunkálás.
7. művelet: G99 esetén: START után visszahúzás az R pontig, gyorsmenettel
8. művelet: G99 esetén: főorsó újraindítás: M3
9. művelet: 98 esetén: START után gyorsmeneti visszahúzás a kiindulási pontig
10. művelet: G98 esetén: főorsó újraindítás: M3

B. Kiesztergálás visszafelé, automatikus szerszámelhúzással

Abban az esetben, ha **a gépen ki van építve az orsó orientálás lehetősége**, azaz az N0607 Spindle Config paraméter #1 ORI bitje 1, a vezérlő a B eset szerint jár el.



19.1.13-2 ábra

A ciklusban felhasznált változók:

G17 **G87** X_p Y_p I J Z_p R F L

G18 **G87** Z_p X_p K I Y_p R F L

G19 **G87** Y_p Z_p J K X_p R F L

A ciklus indításakor M3-as forgásirányt kell adni a főorsónak.

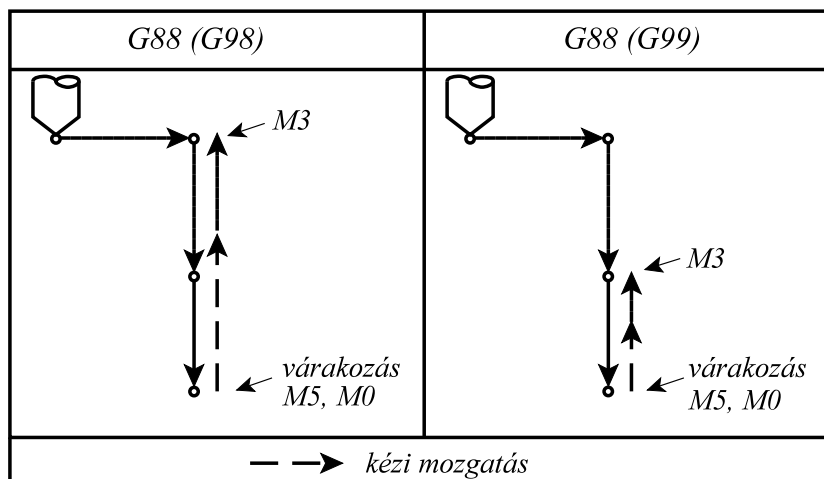
A ciklus műveletei:

1. művelet: pozicionálás a kiválasztott síkban gyorsmenettel
2. művelet:
 - orsó orientálás
 - szerszámelhúzás a kiválasztott síkban I, J, K értékkel gyorsmenettel
3. művelet: gyorsmeneti mozgás az R pontig (megközelítési pont)
4. művelet:
 - szerszámvisszahúzás a kiválasztott síkban I, J illetve K-n megadott értékkel ellentétesen, gyorsmenettel,
 - orsó újraindítása M3 irányban
5. művelet: kiesztergálás a talppontig F előtolással
6. művelet:
 - orsó orientálás: M19
 - szerszámelhúzás a kiválasztott síkban I, J, K értékkel gyorsmenettel
7. művelet: –
8. művelet: –
9. művelet: gyorsmeneti visszahúzás a kiindulási pontig
10. művelet:
 - szerszámvisszahúzás a kiválasztott síkban I, J illetve K-n megadott értékkel ellentétesen, gyorsmenettel,
 - orsó újraindítása M3 irányba

A ciklus természetéből következően az eddigiekkel ellentétben a megközelítési pont, azaz **az R pont mélyebben fekszik, mint a talppont**. Ezt a fúrótengely és R címek programozásánál figyelembe kell venni.

Mivel a ciklus a kiesztergálás előtt orsó orientálást végez és a szerszámot elhúzza a felülettől az I, J illetve K-n megadott értékkel, a behatolásakor elkerülhető a szerszám törése.

19.1.14 Fúróciklus, várakozás után kézi működtetés a talpponton (G88)



19.1.14-1 ábra

A ciklusban felhasznált változók:

G17 **G88** X_p__ Y_p__ Z_p__ R__ P__ F__ L__

G18 **G88** Z_p__ X_p__ Y_p__ R__ P__ F__ L__

G19 **G88** Y_p__ Z_p__ X_p__ R__ P__ F__ L__

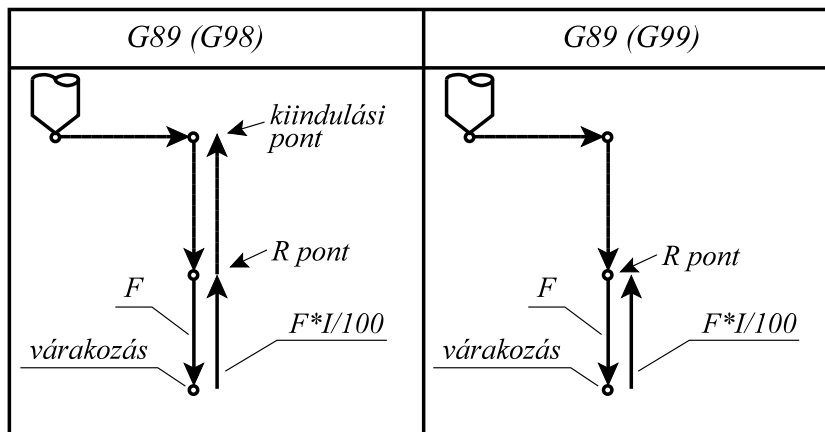
A ciklus indításakor M3-as forgásirányt kell adni az orsónak.

A ciklus műveletei:

1. művelet: pozicionálás a kiválasztott síkban gyorsmenettel
2. művelet: —
3. művelet: gyorsmeneti mozgás az R pontig (megközelítési pont)
4. művelet: —
5. művelet: fúrás a talppontig F előtolással
6. művelet:
 - várakozás P értékkel
 - orsó leállítás: M5
 - a vezérlő STOP állapotot vesz föl (M0), ahonnan a kezelő valamelyik kézi működtető üzembe (MOZGATÁS, LÉPTETÉS, KÉZIKERÉK) kilépve kézzel működtetheti a gépet, vagyis elhúzhatja a szerszám hegyét a furat felületétől, és kihúzhatja a szerszámot a furatból. Ezután visszalépve AUTOMATA üzembe startra továbbmegy a megmunkálás.
7. művelet: G99 esetén: START után visszahúzás az R pontig, gyorsmenettel
8. művelet: G99 esetén: orsó újraindítás: M3
9. művelet: G98 esetén: gyorsmeneti visszahúzás a kiindulási pontig
10. művelet: G98 esetén: orsó újraindítás: M3

A ciklus ugyanaz, mint G87 A esete, csak várakozik az orsó leállítása előtt.

19.1.15 Fúróciklus várakozással, kiemelés előtolással (G89)



19.1.15-1 ábra

A ciklusban felhasznált változók:

G17 **G89** X_p Y_p Z_p R P F I L

G18 **G89** Z_p X_p Y_p R P F I L

G19 **G89** Y_p Z_p X_p R P F I L

A ciklus műveletei:

1. művelet: pozicionálás a kiválasztott síkban gyorsmenettel
2. művelet: -
3. művelet: gyorsmeneti mozgás az R pontig (megközelítési pont)
4. művelet: -
5. művelet: fúrás a talppontig F előtolással
6. művelet: várakozás P címen megadott értékig
7. művelet: visszahúzás az R pontig, $F \cdot I / 100$ előtolással

A kihúzási override I (%)

Ha I címnek nem adunk a mondatban értéket, akkor a 7. műveletben a visszahúzási előtolás override-ja az N1502 Extraction Override in G85, G89 paraméter értéke lesz. Az így kiszámított sebességre még az előtolás százalékkapcsoló is hatással van, vagyis az előtolás az

$F_{kihuzás} = F_{programozott} \cdot El\otol\acute{a}s\ Override \cdot Extraction\ Override\ in\ G85,\ G89/100$
érték lesz.

8. művelet: -
9. művelet: G98 esetén: gyorsmeneti visszahúzás a kiindulási pontig
10. művelet: -

A ciklus megegyezik G85-tel, a várakozást kivéve.

19.2 Megjegyzések a fúróciklusok használatához

- Ciklusüzemmódban ha egy G kód nélküli mondat a következő címek valamelyikét tartalmazza, a fúróciklus végrehajtásra kerül:

X_p, Y_p, Z_p , vagy q

ahol q egy tetszőleges, nem fúró tengely.

Ellenkező esetben a fúróciklus nem hajtódik végre.

- Ciklusüzemmódban G04 P várakozási mondatot programozva a parancs végrehajtódik a programozott P szerint, de a várakozásra vonatkozó ciklusváltozó **n e m** törlődik, és **nem** íródik át.

- I, J, K, Q, E, P értékét olyan mondatban kell megadni, ahol fúrás is történik, különben **n e m** tárolódnak el az értékek.

A fentiek illusztrálására tekintsük a következő példát:

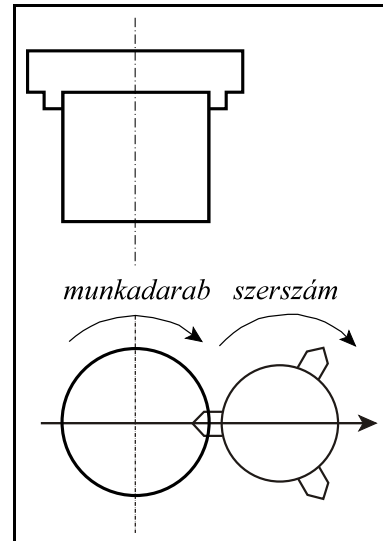
G81 X_ C_ Z_ R_ F	(a fúróciklus végrehajtásra kerül)
X	(a fúróciklus végrehajtásra kerül)
F_	(a fúróciklus nem kerül végrehajtásra, F felülíródik)
M S_	(a fúróciklus nem kerül végrehajtásra, az M kód végrehajtásra kerül)
G4 P_	(a fúróciklus nem kerül végrehajtásra, a várakozás igen, a várakozási ciklusváltozó nem íródik át)
I Q	(a fúróciklus nem kerül végrehajtásra, a programozott értékek nem kerülnek ciklusváltozóként bejegyzésre)

- Ha a fúróciklus mellé funkciót is programozunk a funkció az első művelet végén kerül végrehajtásra a pozicionálás befejezése után. Ha a ciklusban L-et is programoztak, a funkció csak az első menetben kerül végrehajtásra.
- Az L ismétlési szám nem öröklődik.
- A vezérlő mondatonkénti üzemmódban cikluson belül az 1., 3., és a 10. művelet után áll meg.
- A STOP gomb nem hatásos G74, G84, ciklusok 5., 6., és 7. műveletében. Ha ezen műveletek közben nyomnak STOP-ot a vezérlés folytatja működését és csak a 7. művelet végén állítható meg.
- Az előtolás és az orsó override függetlenül a kapcsoló állásától mindig 100% a G84.1, G84 ciklusok 5., 6., és 7. műveletében.
- Ha ciklusmondatban Tnnmm kerül programozásra, az új hosszkorrekciót mind a síkbeli pozicionáláskor, mind a fúrásakor figyelembe veszi.

20 Tengelyvezérlő funkciók

20.1 A sokszögesztergálás

Sokszögesztergáláskor a szerszámot is és a munkadarabot is forgatjuk egymáshoz képest egy meghatározott fordulatszámaránytal. A fordulatszámarányt és a forgó szerszám vágóéleinek számát változtatva különböző oldalszámú sokszöget kapunk. A fordulatszámarány és a vágóélek számának szorzata adja meg a kiadódó sokszög oldalszámát. Ha pl. a szerszám és a munkadarab fordulatszámaránya 2:1, és a forgó szerszám vágóéleinek száma 2 négyszöget, ha a vágóélek száma 3 hatszöget esztergálhatunk. Így egyszerűen lehet esztergályozással pl. hatlapfejű csavart, vagy anyát készíteni. Az így történő megmunkálás sokkal gyorsabb, mintha polárkoordináta interpoláció alkalmazásával marnánk a sokszöget, azonban a kiadódó oldallapok nem pontosan síkok.



20.1-1 ábra

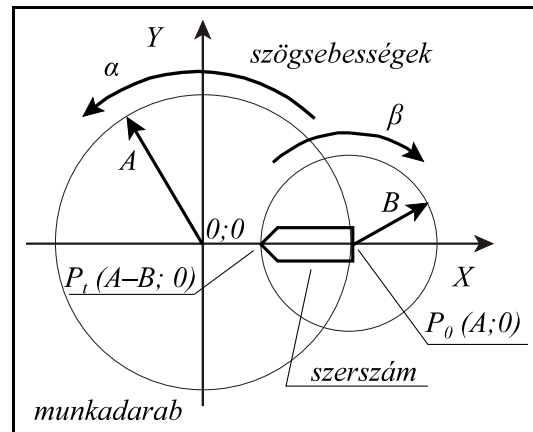
20.1.1 A sokszögesztergálás működési elve

Legyen a munkadarab forgástengelye az $X=0, Y=0$ koordinátán.

A szerszám forgástengelye legyen a P_0 pont, amelynek koordinátái: $X=A, Y=0$, vagyis a szerszám forgástengelyének a munkadarab forgástengelyétől mért távolsága A .

A forgó szerszám sugara legyen B . A szerszám hegyét jelölje P_t . A szerszám hegyének koordinátája a $t=0$ pillanatban $X=A-B; Y=0$.

A munkadarab forgásának szögsebessége legyen α , a szerszám forgásé pedig β .



20.1.1-1 ábra

A t időpontban a szerszám hegyének koordinátái $P_t(x_t; y_t)$ a következők lesznek:

$$x_t = A \cos \alpha t - B \cos(\beta - \alpha) t$$

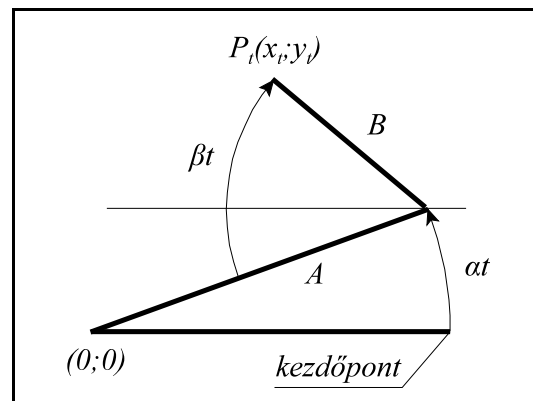
$$y_t = A \sin \alpha t + B \sin(\beta - \alpha) t$$

Ha feltételezzük, hogy a szerszám és a munkadarab fordulatszám aránya 2:1, akkor $\beta=2\alpha$. Helyettesítsük be a fenti egyenletbe:

$$x_t = A \cos \alpha t - B \cos \alpha t = (A - B) \cos \alpha t$$

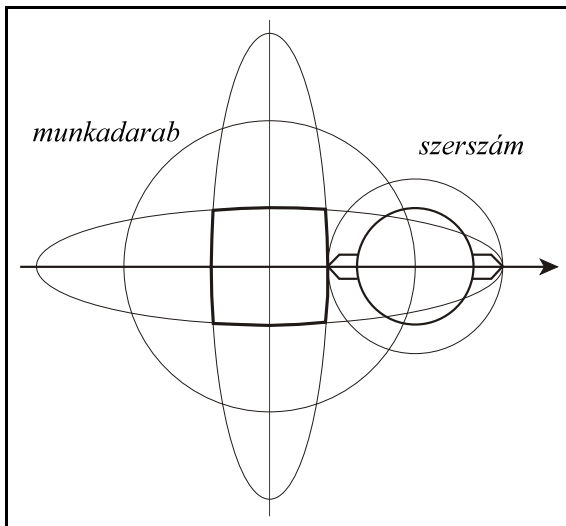
$$y_t = A \sin \alpha t + B \sin \alpha t = (A + B) \sin \alpha t$$

A fenti egyenletek egy olyan ellipszis egyenletei, amelynek nagytengelye $A+B$, kistengelye $A-B$ hosszúságúak.

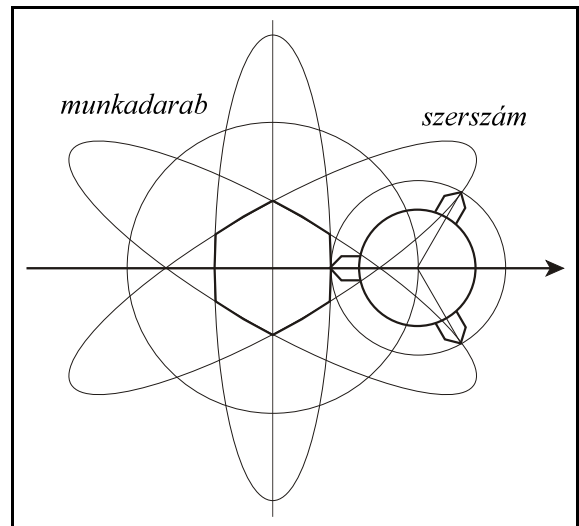


20.1.1-2 ábra

Ha a szerszámok szimmetrikusan egymáshoz képest 180° -ra állnak négyszöget, ha 120° -ra akkor hatszöget esztergálhatunk, feltéve, hogy a szerszám és a darab fordulatszámaránya 2:1.

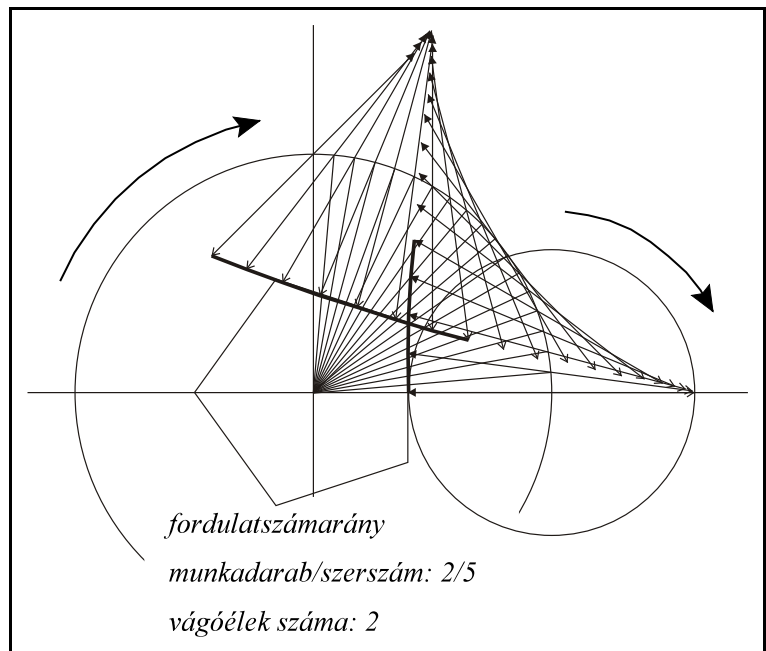


20.1.1-3 ábra



20.1.1-4 ábra

Más fordulatszámarányokkal a kiadódó görbék ellipszistől különbözőek lesznek, de még ezekkel is esetleg jól közelíthetőek sokszögek oldalai.



20.1.1-5 ábra

20.1.2 A sokszögesztergálás programozása (G51.2, G50.2)

A

G51.2 P_ Q_ R__

utasítás bekapcsolja a sokszögesztergálás funkcióját.

A

P és Q

címeiken lehet beállítani a munkadarabnak a szerszámhoz viszonyított fordulatszámarányát:

$P/Q = \text{darab fordulatszám} / \text{szerszám fordulatszám}$

Ha pl. a munkadarabnak és a szerszámnak 1:2 arányban kell forognia programozzunk

G51.2 P1 Q2-t.

P címen mindig *pozitív egész* számot kell megadni.

Q címen *pozitív és negatív egész* számot is megadhatunk. Negatív szám megadása esetén a szerszám forgásiránya a darab forgásirányával ellentétes lesz.

P és címet mindig meg kell adni.

Az

R (fázistolás fokban)

cím kitöltése *opcionális*. R címen a fázistolást, azaz a szerszámorsó nullimpulzusának a munkadarab orsó nullimpulzusához képesti *fáziseltolását* adhatjuk meg *fokban*. Ha az R címet nem programozzuk, a munkadarab- és a szerszámorsó nullimpulzusa egybeesik a szerszámorsó minden Q/P fordulatra. A G51.2 utasítást sokszögesztergálás közben is kiadhatjuk, ha a szolga nullimpulzusát akarjuk eltolni:

G51.2 R_

Ha például négyszöget akarunk esztergálni egyélű késsel, első lépésben a

G51.2 P1 Q2

utasítással két szemben fekvő oldalt munkálunk meg. A két orsó nullimpulzusa közötti távolság 0. Ha ezután

G51.2 R90

utasítást programozunk a kés élét a munkadarabhoz képest 90°-kal elforgatva a másik két szemben fekvő oldalt lehet megmunkálni.

Sokszögesztergálás során úgy a darabot, mind a szerszámot forgató orsón jeladónak kell lennie, és a pozíciószabályozó hurok zárható kell legyen mindkét orsón.

Sokszögesztergálás során az *orsók maximális fordulatszámát* az orsótengelyre beállított *gyorsmeneti sebesség* korlátozza. Ez általában alacsonyabb, mint a maximális orsó fordulatszám.

Sokszögesztergálás során megkülönböztetjük a *darabot forgató mesterorsót* és a *szerszámot forgató szolgálorsót*. **Mindig a szolga szinkronizálódik rá a mesterre.**

A sokszögesztergálás mesterorsója mindig a csatornában főorsónak kijelölt orsó. A főorsó kiválasztásának módját a gépgyártó határozza meg és az adott gép leírása tartalmazza.

A sokszögesztergálás szolgálorsóját programból, pl. orsóhivatkozással és M funkcióval lehet kiválasztani, amit a gépgyártó határoz meg és az adott gép leírása tartalmazza.

A szinkronizáció menete sokszögesztergálás során a következő:

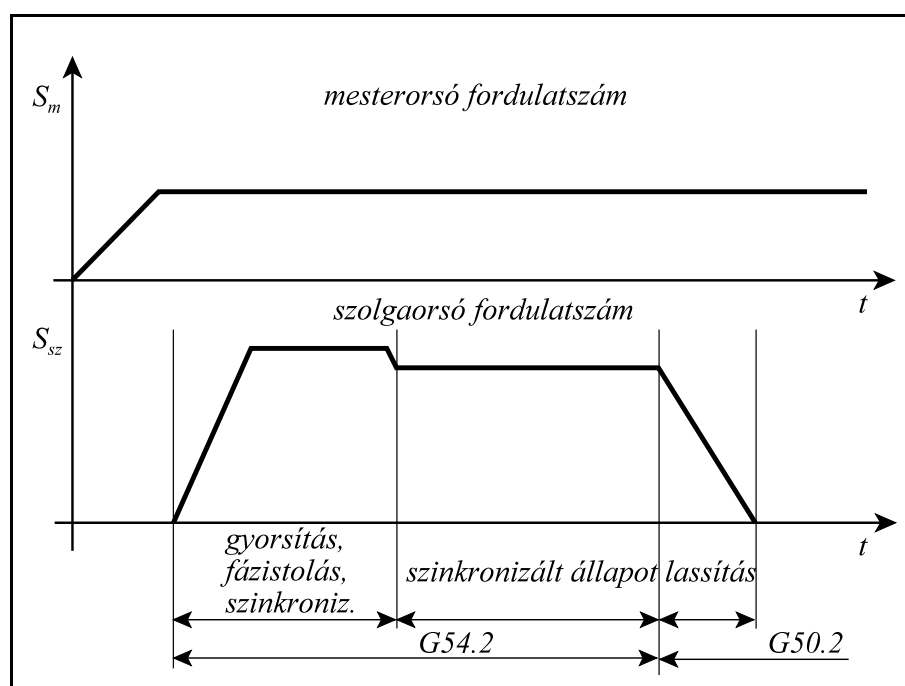
- ha a mesterorsó fordulata nagyobb, mint a gyorsmeneti fordulatszám, vagy a kiadódó szolga fordulatszám ($S \cdot Q/P$) nagyobb, mint a szolga gyorsmeneti fordulatszám, lelassít a megfelelő fordulatra,
- zárja a pozíciószabályozó hurkot,
- a szolgaorsó felpörög a Q/P aránnyal megszorozott mester fordulatszámra ($S \cdot Q/P$), a mesterrel azonos, vagy ellentétes forgásirányban,
- a szolgaorsó zárja a pozíciószabályozó hurkot,
- majd a szolga ráhúzza a nullimpulzusát a mester nullimpulzusára, vagy a mester nullimpulzusától az R címen fokban megadott fázistolásnyira.

A

G50.2

utasítás kikapcsolja a sokszögesztergálást.

A G51.2 és a G50.2 utasítást mindig külön mondatban kell megadni.



20.1.2-1 ábra

A szinkronozás hatása hasonló, mint a menetvágásnál a főorsó nullimpulzusé. Amíg a munkadarabot nem vesszük ki a tokmányból és amíg az ütőkést ugyanolyan szöghelyzetben hagyjuk a szerszámorsóban, addig a munkadarab és a szerszám mozgása szinkron marad, többször végigmehetünk ugyanazon a felületen, például nagyolás majd símítás céljából. Ugyancsak kikapcsolhatjuk a szinkronizálást a G50.2 utasítással, majd visszakapcsolva azt ugyanúgy végigmehetünk az előzőleg esztergált sokszög felületén, feltéve, hogy a programozott fordulatszámarány ugyanaz, mint előzőleg.

☞ *Figyelem!*

Sokszögesztergálás programozása során ügyeljünk arra, hogy a szerszámorsóra kiadódó fordulatszám $n_{szerszámorsó} = S \cdot (Q/P)$ soha ne haladja meg az arra engedélyezett maximális fordulatszámot.

A szinkronfutást kikapcsolja

- a vészstop,
- a szervoköri hibák.

Mintapélda

Esztergáljunk hatszöget. A darab legyen az S1 orsóban, a szerszámot forgassa az S3 orsó. Az alábbi példában megadott M funkciók esetlegesek, a különböző gépeken más kódok lehetnek.

Háromélű szerszámmal:

```
...
M31 (S1 ORSÓ KIJELÖLÉSE FŐORSÓNAK)
S1=1000 M3
S3=0 M23 (S3 ORSÓ BEKAPCSOLÁSA SOKSZÖGESZTERGÁLÁSHOZ)
G0 X20 Z1
G51.2 P1 Q-2 (SOKSZÖGESZTERGÁLÁS BEKAPCSOLÁSA)
X15
G1 Z-10 F0.5
G0 X30
Z1
G50.2
S1=0 M5 (S1 LEÁLLÍTÁSA)
S3=0 M5 (S2 LEÁLLÍTÁSA)
...
```

Egyélű szerszámmal:

```
...
G0 X20 Z1
G51.2 P1 Q-2 (SOKSZÖGESZTERGÁLÁS BEKAPCSOLÁSA)
X15
G1 Z-10 F0.5
G0 X30
Z1
G51.2 R120 (FÁZISTOLÁS 120 FOKKAL)
X15
G1 Z-10 F0.5
G0 X30
Z1
G51.2 R240 (FÁZISTOLÁS 240 FOKKAL)
X15
G1 Z-10 F0.5
G0 X30
Z1
G50.2
...
```

20.2 Fogaskerekek lefejtőmarása (G81.8)

A fogaskerekek lefejtőmarását az NCT2xxM Marógép és megmunkáló központ vezérlő Programozási leírása tartalmazza.

20.3 Tengelyek szinkron mozgatása

Egy darab megmunkálása során szükség lehet két tengely szinkron mozgatására. A tengelyek lehetnek ugyanabban, vagy különböző csatornában is.

Mesternek nevezzük azt a tengelyt, amelyre az alkatrészprogramban az adatokat megadjuk.

Szolgának nevezzük azt a tengelyt, amely szinkron mozog a mesterrel. Amíg egy tengely szinkron szolgál, addig azt a tengelyt sem programból, sem kézi mozgatással nem mozgathatjuk.

A szinkron mozgatást a PLC program kezdeményezi pl. egy M kód végrehajtásával.

A szinkron mozgatásnál alkalmazott paraméterek és PLC jelzők

Ha két tengelyt össze akarunk kapcsolni szinkron mozgatásra, a *szolga tengelyhez tartozó* N2101 Synchronous Master *paraméteren meg kell adni, hogy hányadik tengely a mestere.*

A mester lehet ugyanabban, de különböző csatornában is. Egy mesternek lehet több szolgálja is. Egy szolgálja lehet egy másik tengelynek a mestere is.

Tengelyek *szinkron vezérlésénél csak a mester tengelyt lehet programozni*, vagy kézi mozgásparancsot (jog, kézikerek) kiadni.

A szinkron mozgatást a PLC kéri a szolgálhoz tartozó AP_SYNCR jelző 1-be kapcsolásával.

A PLC megvárja, hogy az NC nyugtássa a kérést az AN_SYNCA jelzőn keresztül. Ettől kezdve a mester tengely elmozdulásait a szolgálja is megkapja.

Ezt a jelzőt *általában M funkciók kapcsolhatják* be, vagy ki.

Például:

```

...
M41 (szinkron mozgatás be)
...
M40 (szinkron mozgatás ki)
...

```

A továbbiakban a szinkronizálás ki-, bekapcsolására az M40, M41 kódpárt használjuk ebben a leírásban. Konkrét gép esetén *a funkció kódját és működési leírását kérje a gép építőjétől.*

A *szinkron mozgatás* csak szigorúan összevárós, *pufferürítő funkció* hatására indulhat.

Ha a szinkron mozgatás *két csatorna között* történik, a másik csatornán összevárós kódot kell programozni:

1-es csatorna programja	2-es csatorna programja
...	...
M502 P12	M502 P12
M41 (szinkron be)	(vár a szinkronra)
M503 P12	M503 P12
...	...
M504 P12	M504 P12
M40 (szinkron ki)	(vár a kikapcsolásra)
M505 P12	M505 P12
...	...

Ezen kívül a szinkron mozgatás kérésének feltétele, hogy úgy a mester tengelyen mind a szolgálja legyen érvényes referenciapont.

Szinkron mozgatsnál a *szolga mozgásiránya* megegyezhet a mesterével, de lehet tőle különböző is. A szolga mozgásirányát az N2102 Synchron Config paraméternek a szolga tengelyhez tartozó #0 MSY bitjén lehet beállítani. Ha a paraméterbit értéke:

- =0: a mester és a szolga ugyanabba az irányba mozog
- =1: a mester és a szolga ellentétes irányba mozog.

Példa tengelyek szinkron mozgására

Tengelyek szinkron mozgására vegyünk példának egy 2 csatornás hosszesztergát.

Az 1. csatornában lévő S1 orsóba rúd-dagoló tölti a darabot és ott munkáljuk meg a darab elejét. A 2. csatornában lévő S2 orsóban, a darab átvétele után pedig a darab végét.

Az 1. csatornában a darabot az S1 orsóval együtt a Z_1 tengely mozgatja. A függőlegesen álló T01, ..., T06 esztergáló szerszámokat pedig az X_1, Y_1 tengelyek.

A 2. csatornában az S2 orsó a darabbal együtt X_2, Z_2 irányban mozog.

Miközben az S1 orsóban a következő darab elejét, addig az S2 orsóban az előző darab végét munkáljuk meg.

Tegyük fel, hogy a darab mindkét végét ugyanúgy kell megmunkálni.

Legyen a T01 és a T02 szerszámtartóban ugyanolyan hosszúságú és lekerekítési sugarú szerszám. Ha az X_1, Y_1 tengellyel az 1. csatornában a T01 szerszám elé állunk be, az X_2 tengellyel pedig a 2. csatornában a T02 szerszám elé, **egyszerre is megmunkálhatjuk a darab mindkét végét a Z_1 és a Z_2 tengelyek szinkron mozgásával.**

Legyen 2. csatorna Z_2 tengelye az 1. csatorna Z_1 tengelyének a szinkron szinkron szolgája, amit paraméteren jelölünk ki.

Tegyük fel, hogy az

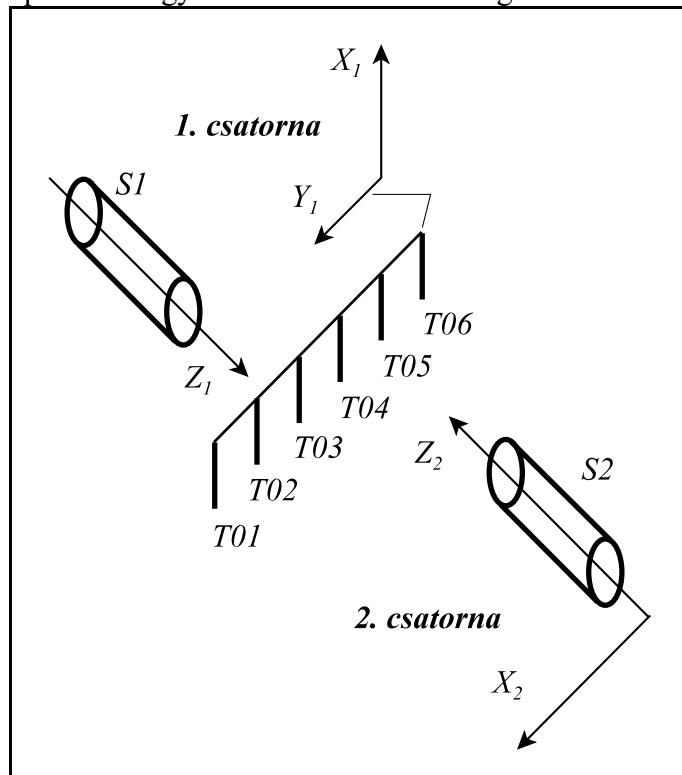
M41 kód kapcsolja össze szinkron mozgásra a Z_1 és a Z_2 tengelyeket, illetve szinkron forgatásra az S1 és S2 orsót,

az

M40 pedig megszünteti a szinkron működést.

Az M41, M40 kódokat mindig az 1. csatornában adjuk meg.

A megmunkáláshoz az S1 orsóban lévő darabot a **T01** szerszámmal szembeni pozícióra kell küldeni az X_1, Y_1, Z_1 tengelyen az 1. csatornában, míg a 2. csatornában az X_2, Z_2 tengelyeket a **T02** szerszámmal szembeni pozícióra. Ezután következik a szinkron forgácsolás. A Z_1 és a Z_2 tengelyek szinkron mozognak. Az X_2 tengely áll és az S2 orsóban lévő darab végét az X_1 és a Z_2 mozgása munkálja meg. A T01 és a T02 szerszám kinyúlásának és lekerekítési sugarának egyezniük kell.



20.3-1 ábra

Az alkatrészprogram a két csatornában az alábbi:

Az 1. csatornában futó program

```

...
T101
G0 X100 Y0 Z10
M501 P12 (összevárás)
M41 (Z1, Z2 szinkron be)
M502 P12 (összevárás)
G42 G0 X0 Z2
G1 Z0 F2
X10 ,C1
Z-5
X15
G40 G0 X100 Z10
M40 (szinkron ki)
M503 P12 (összevárás)
... (normál megmunkálás)

```

A 2. csatornában futó program

```

...
T202
G0 X0 Z10 (szinkron pozíció)
M501 P12 (összevárás)
M502 P12 (összevárás)
(2-es csatorna nem dolgozik)
M503 P12 (összevárás)
... (normál megmunkálás)

```

Tengelyek parkoltatása szinkron tengelymozgatásnál

Szükség lehet arra, hogy egy szinkron mozgatóval megírt program csak az egyik, a mester, vagy a szolga oldalán fusson és ne kelljen új programot írni.

Ekkor azt az oldalt, amelyiket nem kívánjuk mozgatni “parkoltatni” lehet. Úgy a szolga, mind a mester oldalt lehet parkoltatni. Ilyen esetben is a vezérlő kiszámítja a megfelelő elmozdulásokat úgy a mester, mind a szolga tengelyre, de a parkoló oldalnak nem ad ki mozgásparancsot.

Egy, a szinkron mozgásban részt vevő tengely parkoltatását a PLC kéri az AP_PARKR parkolás kérés jelző 1-be írásával. A vezérlő a parkolás kérés nyugtázását az AN_PARKA nyugtázó jelző 1-be írásával nyugtázza.

A parkoltatást lehet kezdeményezni ***nyomógombról, vagy M funkcióról.*** Konkrét gép esetén ***a parkoltatás működési leírását kérje a gép építőjétől.***

20.4 Tengelycsere

Egy darab megmunkálása során szükség lehet két tengely felcserélésére. A tengelyek lehetnek ugyanabban, vagy különböző csatornában is.

Mesternek nevezzük azt a tengelyt, amelyikkel a szolgát felcseréljük.

Szolgának nevezzük azt a tengelyt, amelyik a cserét kezdeményezi.

A tengelycsere a PLC program kezdeményezi pl. egy M kód végrehajtásával.

A tengelycsereénél alkalmazott paraméterek és PLC jelzők

Ha két tengelyt fel akarunk cserélni, a **szolga tengelyhez tartozó** N2104 Composit Axis **paraméteren meg kell adni, hogy hányadik tengely a mestere.**

A mester lehet ugyanabban, de különböző csatornában is.

Tengelyek **cseréjénél a mester tengelyt is és a szolgát is lehet programozni**, vagy kézi mozgásparancsot (jog, kézikerek) kiadni.

A tengelycsere a PLC kéri a szolgálhoz tartozó AP_MIXR jelző 1-be kapcsolásával. A PLC megvárja, hogy az NC nyugtázza a kérést az AN_MIXA jelzön keresztül. Ettől kezdve a két tengely fel van cserélve.

Ezt a jelzőt **általában M funkciók kapcsolhatják** be, vagy ki.

Például:

```
...
M42 (tengelycsere be)
```

```
...
M40 (tengelycsere ki)
```

```
...
```

A továbbiakban a tengelycsere ki-, bekapcsolására az M40, M42 kódpárt használjuk ebben a leírásban. Konkrét gép esetén **a funkció kódját és működési leírását kérje a gép építőjétől.**

A **tengelycsere** csak szigorúan összevárós, **pufferürítő funkció** hatására indulhat.

Ha a tengelycsere **két csatorna között** történik, a másik csatornán összevárós kódot kell programozni:

1-es csatorna programja

```
...
M502 P12
M42 (tengelycsere be)
M503 P12
```

```
...
M504 P12
M40 (tengelycsere ki)
M505 P12
```

```
...
```

2-es csatorna programja

```
...
M502 P12
(vár a cserére)
M503 P12
```

```
...
M504 P12
(vár a csere törlésére)
M505 P12
```

```
...
```

Tengelycsere után a **szolga mozgásiránya** megegyezhet az eredeti irányával, de lehet vele ellentétes is. A szolga mozgásirányát az N2105 Composit Config paraméternek a szolga tengelyhez tartozó #0 MMI bitjén lehet beállítani. Ha a paraméterbit értéke:

=0: tengelycsere után a szolga az eredeti irányába mozog

=1: tengelycsere után a szolga az eredetivel ellentétes irányba mozog.

Tengelycsere után a **mester mozgásiránya** megegyezhet az eredeti irányával, de lehet vele ellentétes is. A mester mozgásirányát is az N2105 Composit Config paraméternek a mester tengelyhez tartozó #0 MMI bitjén lehet beállítani. Ha a paraméterbit értéke:

=0: tengelycsere után a mester az eredeti irányába mozog

=1: tengelycsere után a mester az eredetivel ellentétes irányba mozog.

Tengelycsere után **a programot megírhatjuk a valós és a virtuális (tükrözött) oldalon is.**

Az N2105 Composit Config paraméternek a szolga tengelyhez tartozó #2 MCO bitjén lehet a virtuális oldalon történő programozást beállítani. Ha a paraméterbit értéke:

=0: tengelycsere után a szolgát a valós oldalon programozzuk

=1: tengelycsere után a szolgát a virtuális oldalon programozzuk.

Az N2105 Composit Config paraméternek a mester tengelyhez tartozó #2 MCO bitjén lehet a virtuális oldalon történő programozást beállítani. Ha a paraméterbit értéke:

=0: tengelycsere után a szolgát a valós oldalon programozzuk

=1: tengelycsere után a szolgát a virtuális oldalon programozzuk.

Tengelycsere után **a tengelyek megöröklik az eredeti nullponttolásaikat és hosszkorrekcióikat.**

Ha például felcseréljük két csatorna között az X_1 - X_2 és a Z_1 - Z_2 tengelyeket, a csere után az X_1 tengelyre beállított hosszkorrekció a 2. csatornába kerül és fordítba. Ezért célszerű tengelycsere után új, a tengelycsere után bemért nullponttolást, illetve új hosszkorrekciót beváltani.

Példa tengelycsere alkalmazására

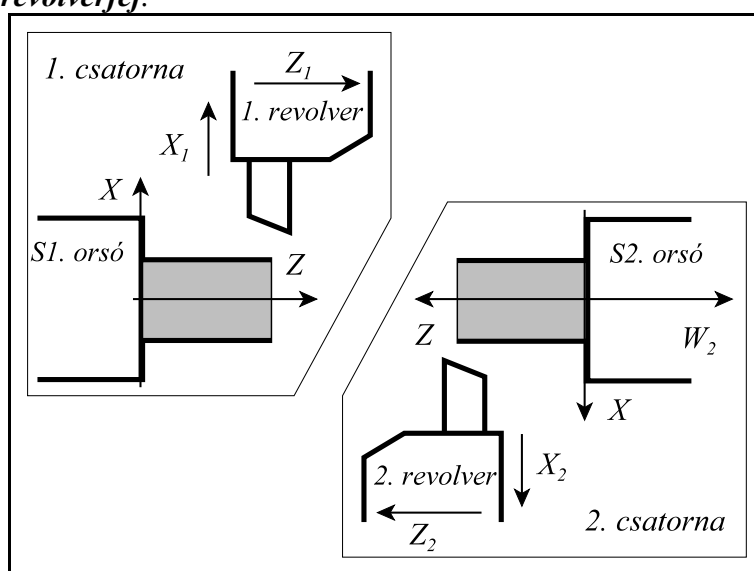
Legyen egy **2 csatornás gépen két revolverfej.**

Az **első revolvert az 1. csatorna kezeli**, és az X_1 , Z_1 tengelyek mentén mozgatható. A **másodikat a 2. csatorna kezeli** és az X_2 , Z_2 tengelyek mentén mozgatható.

Darabátvétel miatt a 2. csatornában lévő S2 ellenorsó a W_2 tengellyel mozgatható.

Az S1 orsóban a darab egyik, az S2 orsóban a darab másik oldalát munkáljuk meg.

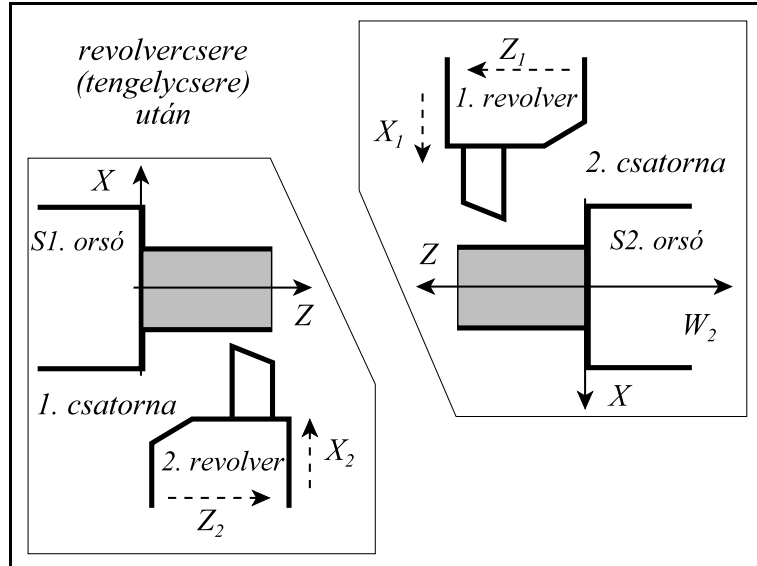
Mindkét csatornában a megmunkáláshoz szükség lehet azonos szerszámokra.



20.4-1 ábra

Bonyolult darab esetén, amikor sokféle szerszámmal kell dolgozni, felmerülhet, hogy nincs annyi férőhely az egyik, vagy a másik revolverfejben, hogy mindkettőben azonos szerszámokat tartunk. Ekkor, ha valamelyik csatornában szükség van a másik revolverfejben tárolt szerszámra, **fel kell cserélni a csatornák között a revolverfejeket**, vagyis az X_1, Z_1 és az X_2, Z_2 tengelyeket. Ezután mindkét csatornában lehet megmunkálást végezni.

Mint a mellékelt ábrán is látszik mind az X_1, Z_1 mind az X_2, Z_2 tengelyek irányát fordítani kell eredeti irányukhoz képest az MMI bit 1-be írásával, azért, hogy az irányok alkalmazkodjanak a csatornában felvett koordinátairányokhoz.



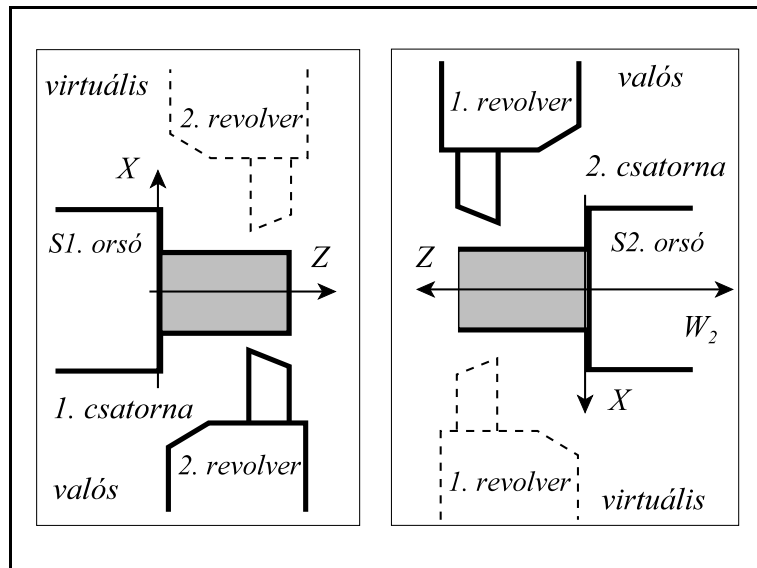
20.4-2 ábra

Ezután eldönthetjük, hogy a programot a valós vagy a virtuális oldalon akarjuk megírni.

Ha akár az 1., akár a 2. csatornában írunk tengelycsere után a **valós oldalon** programot, az X tengely pozitív irányú mozgása közelíti a szerszámot a darabhoz, a negatív irányú mozgása pedig távolítja attól. Épp ellenkezően kell programozni, mint tengelycsere előtt!

Ha a programot a **virtuális oldalon** akarjuk megírni, tehát hasonló módon, mint a tengelycsere előtt, akkor az X_1 és az X_2 tengelyekhez tartozó MCO bitet 1-be kell írni.

Ekkor a vezérlő az így megírt **programot automatikusan tükrözi a Z tengelyre**.



20.4-3 ábra

Az ábrán a 2. csatornában lévő darab látható, amelyet tengelycsere után az X_1, Z_1 tengelyekkel mozgatott revolverfejjben lévő szerszámmal akarunk megmunkálni.

A szerszám valós, abszolút pozíciója a darabhoz rögzített koordinátarendszerben:

$$x_2 = -120, z_2 = 80.$$

Az alábbiakban két példán mutatjuk, be hogyan kell írni a programot a *valós oldalon*

$$MCO_{x_2} = 0$$

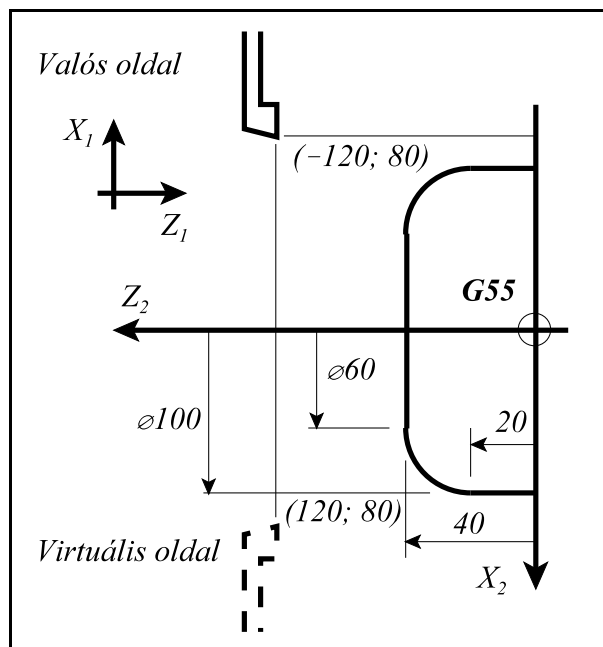
paraméterállásnál, illetve hogyan kell írni a *virtuális oldalon*

$$MCO_{x_2} = 1$$

paraméterállásnál.

A programot a virtuális oldalon írni olyan, mintha nem vettünk volna tudomást a tengelycsereről, és az eredeti revolverfejjel dolgozánk.

A szerszámsugár korrekcióban a *szerszámállítás Q kódját aszerint kell megadni, hogy melyik oldalra irtuk meg a programot*. Az alábbi példában a valós oldalon Q2-t, a virtuális oldalon Q3-at kell megadni.



20.4-4 ábra

Valós oldal, $MCO_{x_2} = 0$:

```
... (megmunkálás  $X_2, Z_2$  tengelyekkel)
(tengelycsere, megmunkálás  $X_1, Z_1$  tengelyekkel)
M502 P12
G55 T101 (R0.8, Q2)
G0 X-120 Z80
G41 G0 X0 Z40
G1 X-60 F0.5
G2 X-100 Z20 R20
G1 Z0
G0 X-120
G40 Z80
M503 P12
...
```

Virtuális oldal, $MCO_{x_2} = 1$:

```
... (megmunkálás  $X_2, Z_2$  tengelyekkel)
(tengelycsere, megmunkálás  $X_1, Z_1$  tengelyekkel)
M502 P12
G55 T101 (R0.8, Q3)
G0 X120 Z80
G42 G0 X0 Z40
G1 X60 F0.5
G3 X100 Z20 R20
G1 Z0
G0 X120
G40 Z80
M503 P12
...
```

A szerszám, természetesen, mindig a valós oldalon fog mozogni.

Valós és hipotetikus tengely cseréje

Létezik olyan eset, amikor **egy valódi és egy nem létező tengelyt kell felcserélni**. Ilyen esetet mutat a mellékelt ábra, amely egy hosszeszterga kinematikáját mutatja.

Az 1. csatornában lévő S1 orsóba rúdadagoló tölti a darabot és ott munkáljuk meg a darab elejét. A 2. csatornában lévő S2 orsóban, a darab átvétele után pedig a darab végét.

Az 1. csatornában a darabot az S1 orsóval együtt a Z_1 tengely mozgatja. A függőlegesen álló T01, ..., T06 esztergáló szerszámokat pedig az X_1, Y_1 tengelyek.

A 2. csatornában az S2 orsó a darabbal együtt X_2, Z_2 irányban mozog.

Miközben az S1 orsóban a következő darab elejét, addig az S2 orsóban az előző darab végét munkáljuk meg.

A 2. csatornában nincs Y_2 tengely. Tegyük fel, hogy a 2. csatornában a T01, ..., T06 szerszámcsoport valamilyik szerszámával kell a darab végét megmunkálnunk.

Ekkor az **1. és a 2. csatorna között fel kellene cserélni az X_1, Y_1 és az X_2, Y_2 tengelyeket**. Annak érdekében, hogy a csere mechanizmusa meg tudjon történni, **a 2. csatornában ki kell jelölni egy hipotetikus Y_2 tengelyt**.

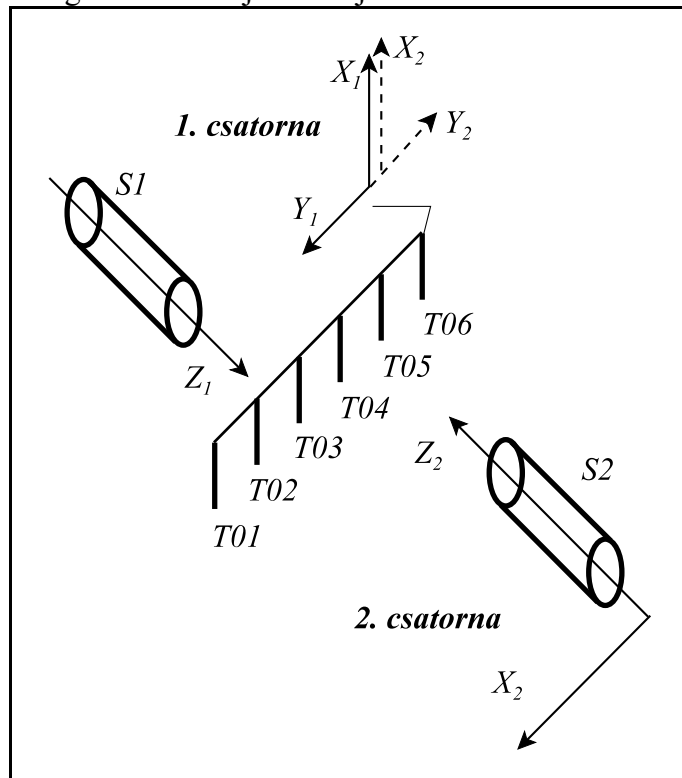
Az N0106 Axis Properties paraméter #7 HYP=1 bitállítással **jelölhetünk ki hipotetikus tengelyt**.

A hipotetikus tengelynek van neve (esetünkben Y) és **száma**, de a szervo paraméterek között **fizikai tengely ki- és bemenet nincs hozzá rendelve**, azaz nem tartozik hozzá jeladó bemenet és hajtás kimenet.

Tengelycsere, után esetünkben a 2. csatornában, hivatkozhatunk az Y tengelyre programból és kézi mozgatással egyaránt.

Az ábrán látszik, hogy az Y_2 tengely pozitív iránya ellentétes az Y_1 tengely pozitív irányával azért, hogy az X_2, Y_2, Z_2 koordináta-rendszer jobbsodrású legyen. Ezért az N2105 Composit Config paraméternek az Y_2 tengelyhez tartozó #0 MMI bitjét 1-be kell állítani.

Mivel tengelycsere után az 1. csatornába került a hipotetikus tengely, ezért itt nem hivatkozhatunk az Y tengelyre se programból se kézi mozgatással. Ugyanúgy nem hivatkozhatunk az 1. csatornában az X tengelyre sem mert csere után X tengely az S2 orsót mozgatná.



20.4-5 ábra

20.5 Tengelyek szuperponált mozgása

A darab megmunkálása során szükség lehet arra, hogy egy tengely mozgásához egy másik tengely mozgása is hozzáadódjon. Ezt nevezzük a tengelyek szuperponált mozgásának. A tengelyek lehetnek ugyanabban, vagy különböző csatornában is.

Mesternek nevezzük azt a tengelyt, amelynek a mozgása hozzáadódik a szolga mozgásához..

Szolgának nevezzük azt a tengelyt, amelynek a mozgásához hozzáadódik a mester mozgása. Programból úgy a mestert, mind a szolgát mozgathatjuk.

A szuperponált mozgást a PLC program kezdeményezi pl. egy M kód végrehajtásával.

A szuperponált mozgásnál alkalmazott paraméterek és PLC jelzők

Ha két tengelyt össze akarunk kapcsolni szuperponált mozgásra, a **szolga tengelyhez tartozó N2107 Superimposed Master paraméteren meg kell adni, hogy hányadik tengely a mestere.**

A mester lehet ugyanabban, de különböző csatornában is. Egy mesternek lehet több szolgálja is. Egy szolga lehet egy másik tengelynek a mestere is.

Tengelyek **szuperponált vezérlésénél mindkét tengelyt, úgy a mestert, mind a szolgát lehet programozni**, vagy kézi mozgásparancsot (jog, kézikerek) kiadni.

A szuperponált mozgást a PLC kéri a szolgálhoz tartozó AP_SPRPNR jelző 1-be kapcsolásával. A PLC megvárja, hogy az NC nyugtázza a kérést az AN_SPRPNA jelzón keresztül. Ettől kezdve a mester tengely elmozdulásait a szolga tengely is megkapja és a mester elmozdulásai hozzáadódnak a szolga elmozdulásaihoz.

Ezt a jelzőt **általában M funkciók kapcsolhatják** be, vagy ki.

Például:

```

...
M43 (szuperponált mozgás be)
...
M40 (szuperponált mozgás ki)
...

```

A továbbiakban a szuperponálás ki-, bekapcsolására az M40, M43 kódpárt használjuk ebben a leírásban. Konkrét gép esetén **a funkció kódját és működési leírását kérje a gép építőjétől.**

A **szuperponált mozgás** csak szigorúan összevárós, **pufferürítő funkció** hatására indulhat.

Ha a szuperponált mozgás **két csatorna között** történik, a másik csatornán összevárós kódot kell programozni:

1-es csatorna programja	2-es csatorna programja
...	...
M502 P12	M502 P12
M43 (szuperponálás be)	(vár a szuperponálásra)
M503 P12	M503 P12
...	...
M504 P12	M504 P12
M40 (superponálás ki)	(vár a kikapcsolásra)
M505 P12	M505 P12
...	...

Ezen kívül a szuperponált mozgás kérésének feltétele, hogy úgy a mester tengelyen, mind a szolgán legyen érvényes referenciapont.

Szuperponált mozgatsnál a *szolga mozgásához* hozzáadódhat a mester elmozdulása, de ki is vonódhat belőle. Azt, hogy a szolga mozgáshoz hozzáadódjon, vagy abból kivonódjon a mester elmozdulása, az N2108 Superimposed Config paraméternek a szolga tengelyhez tartozó #0 MSU bitjén lehet beállítani. Ha a paraméterbit értéke:

=0: a szolga elmozdulásához hozzáadódik a mester elmozdulása

=1: a szolga elmozdulásából kivonódik a mester elmozdulása.

Példa tengelyek szuperponált mozgására

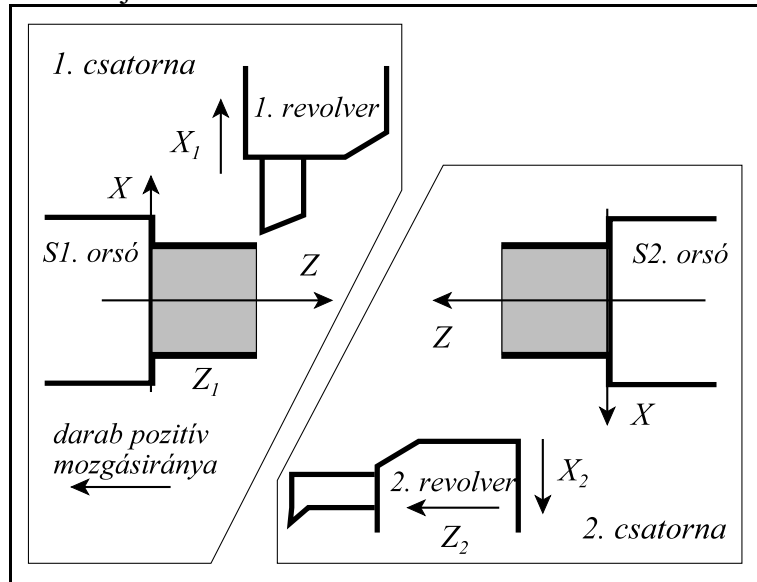
Legyen egy 2 csatornás gépen két revolverfej.

Az első revolvert az 1. csatorna kezeli, és az X_1 tengely mentén mozgatható, míg az $S1$ orsót a darabbal együtt a Z_1 tengely mozgatja.

A második revolverfejet a 2. csatorna kezeli és az X_2, Z_2 tengelyek mentén mozgatható. A 2. csatornában lévő $S2$ ellenorsó áll.

Az $S1$ orsóban a darab egyik, az $S2$ orsóban a darab másik oldalát munkáljuk meg.

Tegyük fel, hogy az $S1$ orsóban lévő darabot kívülről az 1. revolverfejben lévő szerszámmal, míg belülről a 2. revolverfejben lévő szerszámmal munkáljuk meg.

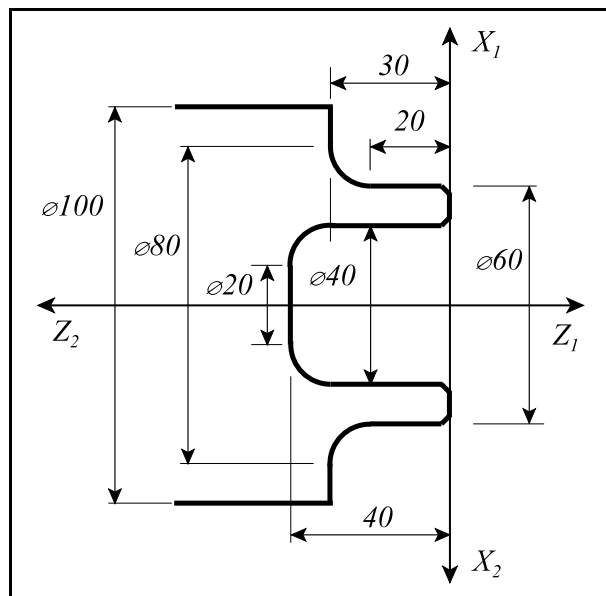


20.5-1 ábra

Az 1. csatornában futó program az X_1, Z_1 tengelyeket mozgatja. Ezzel egy időben a 2. csatornában futó program az X_2, Z_2 tengelyeket. Mivel a Z_1 tengely az $S1$ orsóban lévő darabot mozgatja jobbra balra, ezért a Z_1 tengely elmozdulásához hozzá kell adni a Z_2 tengely elmozdulásához, hogy a 2. revolverfejben lévő szerszám pozíciója a darabhoz képest ne változzon Z_1 mozgása miatt.

Ebben a felállásban Z_1 a mester, Z_2 a szolga. Ha a Z_1 mester pozitív irányba mozog a Z_2 szolgát is pozitív irányba kell mozgatni, hogy a 2. revolverfejben lévő szerszám pozíciója az $S1$ orsóban lévő darabhoz képest ne változzon. Ezért a szolga tengelyen az N2108 Superimposed Config paraméteren #0 MSU=0 bitállítást kell beállítani.

A mellékelt munkadarab rajzot leíró programok a következők:



20.5-2 ábra

Az 1. csatornában futó program

```

...
T101 (szerszámállítás Q3)
G55 G90 G40 G0 X120 Z10
M3 S1=1000
M501 P12
M43 (szuperponált mozgatás)
M502 P12
G42 G0 X50 Z2
G1 X60 ,C3 F0.6
Z-20
G2 X80 Z-30 R10
G1 X100
G40 G0 Z10

M503 P12
M40 (normál megmunkálás)
M504 P12
G54 G0 X200 Z20
...

```

A 2. csatornában futó program

```

...
T2121 (szerszámállítás Q1)
G55 G90 G40 G0 X120 Z-10

M501 P12

M502 P12
G42 G0 X60 Z-5
G1 Z0 F0.8
X40 ,C1
Z30
G2 X20 Z40 R10
G1 X0
G40 G0 Z-10
X120
M503 P12

M504 P12
G54 G0 X150 Z10
...

```

20.6 Tengelyirány váltás

A vezérlőben, PLC programból, meg lehet változtatni egy tengely mozgásirányát: a pozitív irányú mozgásparancsból negatív irányú mozgás lesz és fordítva.

A tengelyirány váltást a PLC kéri az adott tengelyhez tartozó AP_MIRR jelző 1-be kapcsolásával. A PLC megvárja, hogy az NC nyugtázza a kérést az AN_MIRA jelzőn keresztül. Ettől kezdve a tengely mozgásirány a paraméteren beállítottal ellentétes lesz.

Ezt a jelzőt *általában M funkciók kapcsolhatják* be, vagy ki.

Konkrét gép esetén **a funkció kódját és működési leírását kérje a gép építőjétől.**

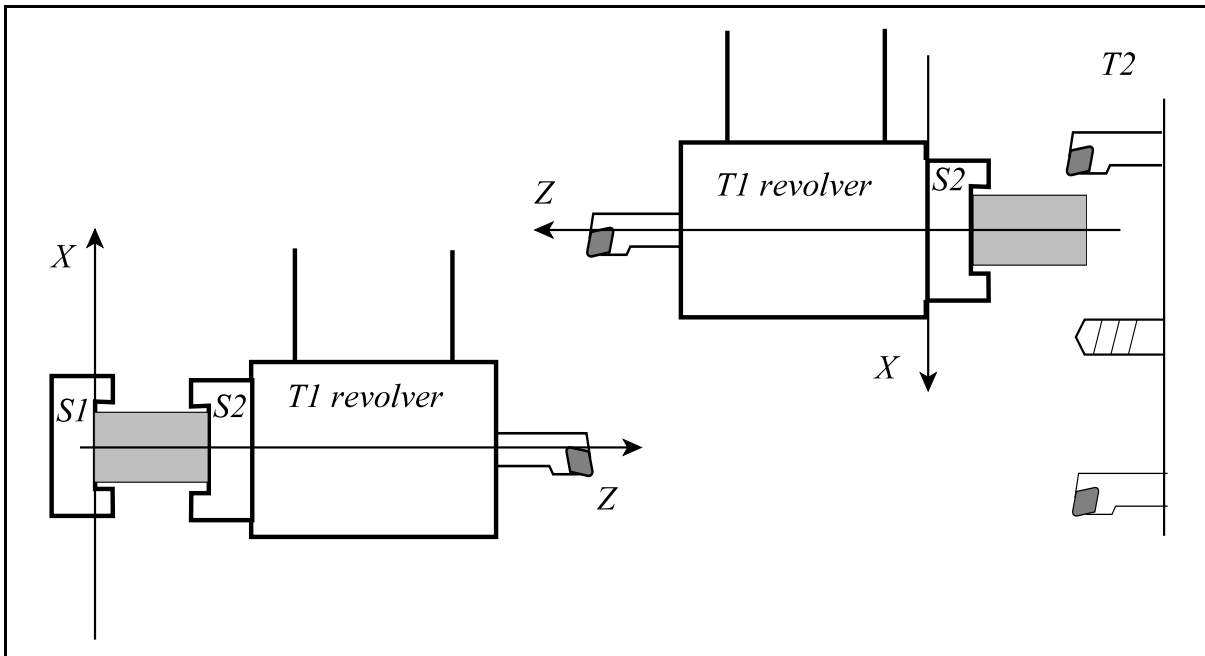
A **tengelyirány váltás** csak szigorúan összevörös, **pufferürítő funkció** hatására indulhat.

A tengelyirány váltás nem érinti a gépi koordinátarendszerben programozott pozicionálások (G53) és a referenciapontra való küldés (G28, G30 P) mozgásirányát és pozícióit.

Példa tengelyirány váltásra

A T1 revolverfej a benne lévő szerszámokkal az S1 orsóban lévő darab elejét munkálja meg. Ekkor a szerszám mozog X-Z irányban a darabhoz képest. Ezután a darabot átveszi a T1 revolverfejben lévő S2 orsóba (lásd ábra). A revolverfej forgatása után S2 szembe kerül a T2 kiterített szerszámokat tartalmazó szerszámcsoporthal, amivel a darab hátulját munkáljuk meg. Ekkor az S2-ben orsóban lévő darab mozog a T2 szerszámcsoporthoz képest.

Annak érdekében, hogy a programokat hagyományos módon lehessen megírni (pozitív mozgásirányban a szerszám távolodik a darabtól), illetve, hogy a koordinátarendszer jobbsodrású maradjon, vagyis a körirányok, sugárkorrekció iránya stb. ne változzék, az X és a Z tengely mozgásirányát is fordítani kell.



20.6-1 ábra

Legyen például M44 a tengelyirány váltását végző M funkció, az irányváltást pedig törölje az M40 kód.

```

...
G54 T101
...
M44 (irányváltás az X-Z tengelyen)
G55 T2020 (új munkadarab nullpont és hosszkorrekció)
...
M40 (irányváltás törlése az X-Z tengelyen)
G54 T101 (új munkadarab nullpont és hosszkorrekció)
...

```

Tengelyirány váltás után új, az irányváltás után bemért munkadarab koordinátarendszert és új, a T2 csoportba tartozó hosszkorrekciót kell beváltani. Hasonló a helyzet az irányváltás törlése után is.

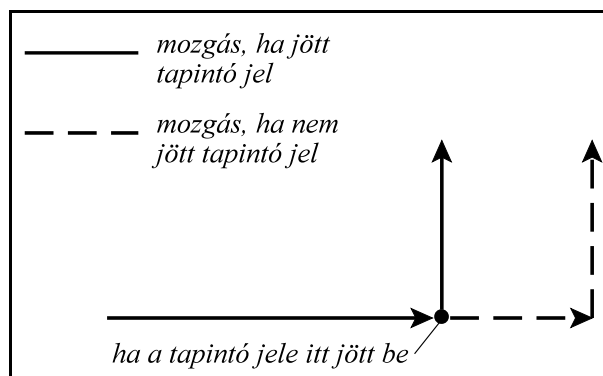
21 MÉRŐFUNKCIÓK

21.1 Mérés a maradék út törlésével (G31)

A

G31 v (P) (F)

utasítás hatására a v koordinátájú pontra **lineáris interpolációval** elindul a **mozgás**. A mozgás addig tart, amíg egy **külső törlőjel** (pl. egy mérő-tapintó jele) be nem érkezik, vagy a **v koordinátákon** megadott **végponti pozíciót** el nem érte a vezérlő. A törlőjel beérkezése után, vagy a mondat programozott végpontján a vezérlő lelassít és megáll.



21.1-2 ábra

P címen lehet megadni, hogy a vezérlőbe bevezethető 8 tapintójel közül **hányadik jelét** vegye figyelembe a mozgás során:

P1: 1. tapintójel figyelembe vétele

P2: 2. tapintójel figyelembe vétele

...

P8: 8. tapintójel figyelembe vétele.

P cím kitöltése nem kötelező, ha a **P** címet nem töltjük ki, az 1. sz. tapintó jelét veszi figyelembe.

A **G31** funkciót mindig **G94 (percenkénti előtolás)** állapotban kell használni. A mozgás során az **előtolás F**:

- a megadott, vagy öröklődött **F** érték, ha az N3001 G31 Config paraméter #0 SKF bitje 0,
- az N0311 G31 Feed paraméterről felvett előtolásérték, ha az N3001 G31 Config paraméter #0 SKF bitje 1.

A **G31** utasítás **nem öröklődik**, csak abban a mondatban érvényes, amelyikben programozták.

A külső jel bejövetele pillanatában a **tengelyek pozíciója eltárolódik** az alábbi **makrováltozókon**:

#5061, vagy #100151, vagy #_ABSKP[1]: 1. tengely pozíciója

#5062, vagy #100152, vagy #_ABSKP[2]: 2. tengely pozíciója

...

A fenti makrováltozókon **eltárolt pozíció**:

- ha **jött külső jel**, a **jel bejövételének pillanatában** felvett pozíció,
- ha **nem jött külső jel** a **G31** mondat programozott **végpontjának** pozíciója.

A pozícióadatok

- mindig az **aktuális munkadarab koordinátarendszerben**,
- az aktuális **hosszkorrekció** (G43, G44) **figyelembe vétele nélkül** kerül eltárolásra.

A külső jel bejövetele után a mozgás lassítással leáll. A **G31** mondat végponti pozíciója ekkor a mondatban alkalmazott előtolás függvényében kis mértékben eltér a jel bejövetelekor a #5061... változókon letárolásra került pozícióktól. A mondat végponti pozíciói a #5001... változókon érhetők el. A következő mozgásmondat ezektől a végponti pozícióktól kezdve fog érvényesülni.

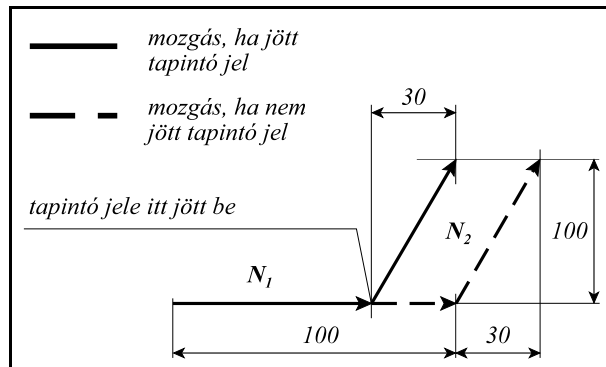
A G31 mondat végrehajtása csak **G15, G40, G50, G50.1 G69, G94** állapotban lehetséges. Ellenkező eset “2055 Tapintás Gnn állapotban” hibajelzést vált ki.

A v koordinátákon megadott érték lehet **inkrementális és abszolút** is. Ha a következő mozgásparancs koordinátamegadása inkrementális, az elmozdulást a G31 mondat azon pontjától számítja, ahol a mozgás az előző mondatban abbamaradt.

Például:

```
N1 G31 G91 Z100
N2 X100 Z30
```

N1 mondatban elindít egy inkrementális mozgást Z irányban. Ha a külső jel bejövetele után a vezérlő az Z=86.7 koordinátájú ponton áll meg, ettől a ponttól számítva lép inkrementálisan 100-at X irányban, 30-at a Z irányban az N2 mondatban.

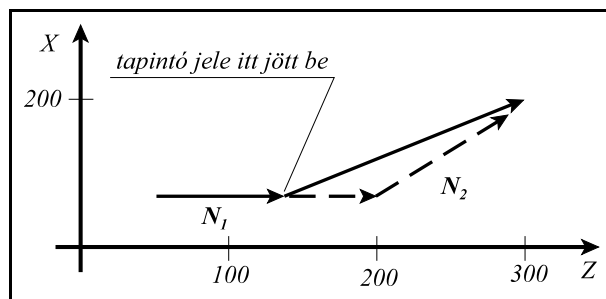


21.1-3 ábra

Ha abszolút adatmegadást programoztunk a mozgás a következőképp alakul:

```
N1 G31 G90 Z200
N2 X200 Z300
```

N1 mondat elindít Z irányban egy mozgást az Z=200 koordinátájú pontra. Ha a külső jel bejövetele után a vezérlő a Z=130 koordinátájú ponton áll meg, az N2 mondatban a Z irányú elmozdulás Z=300-130, azaz Z=170 lesz.



21.1-4 ábra

21.2 Maradék út törlése nyomatékhatárra (G31)

A G31 funkciót használhatjuk úgy is, hogy a **mozgás megállítása és a maradék út törlése** nem egy külső jelre (pl. tapintó jelére) történik, hanem arra hogy, egy **kijelölt tengelyen a motor nyomatéka elér egy megadott értéket**, például úgy, hogy a szerszámot nekinyomjuk egy rögzített felületnek. A nyomatékhatár elérése után, vagy a mondat programozott végpontján a vezérlő megáll, majd továbblép a következő mondat végrehajtására.

A fent leírtakat a

G31 P98 Qq v Ff

funkció valósítja meg, ahol:

Q: a programozható nyomatékhatár értéke, a motor maximális nyomatékának százalékában. Q0 0%-nak, Q255 100%-nak felel meg.

Megadható értékek: Q1-Q254

v: a programozott tengely neve és végpozíciója, amelyiken a nyomatékhatárt figyeli a vezérlő. **Csak egy tengelycím adható meg.**

F: az előtolás értéke mm/perc-ben, vagy inch/perc-ben. (G94 állapot kell).

A mozgás során az **előtolás F:**

– a megadott, vagy öröklődött F érték, ha az N3001 G31 Config paraméter #0 SKF bitje 0,

– az N0311 G31 Feed paraméterről felvett előtolásérték, ha az N3001 G31 Config paraméter #0 SKF bitje 1.

A **G31 P98** utasítás *nem öröklődik*, csak abban a mondatban érvényes, amelyikben programozták.

Mintapélda:

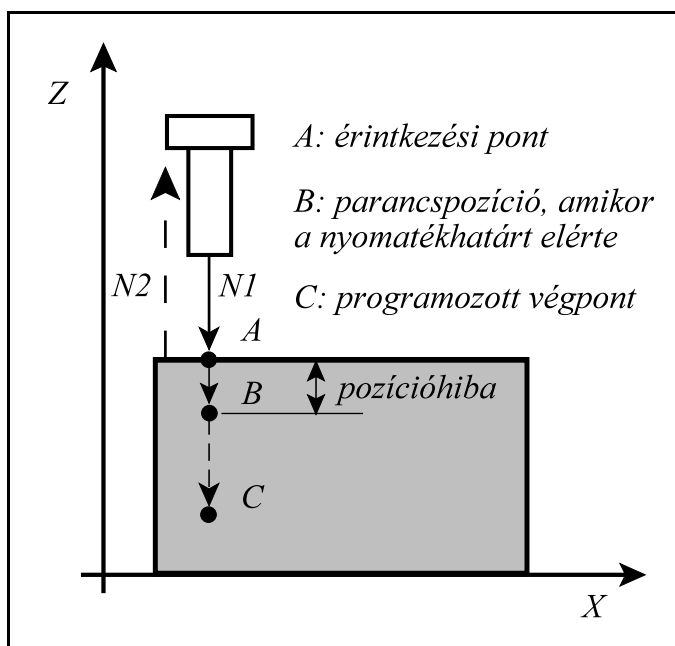
```
N1 G31 P98 Q50 Z30 F100
N2 G0 Z200
```

Az N1 mondatban az **“A” pontban éri el a szerszám a darab felületét**. Ettől a ponttól számítva a Z tengely már nem mozog. Mivel a Q50 (19.6%) nyomatékhatárt még nem érte el a motor, a vezérlő nem állítja meg a mozgást.

A **pozícióhiba** a Z tengelyen az **“A” ponttól a “B” pontig** folyamatosan növekszik, miközben folyamatosan nő a motor nyomatéka.

A mozgásparancs kiadása a **“B” pontig** folytatódik, ahol **eléri a motor a nyomatékhatárt**.

A vezérlő a **“B” pontban** megállítja a mozgást, **bejegyzi az “A” pont pozícióját, majd megkezd a következő, az N2 mondat végrehajtását**.



21.2-1 ábra

A pozícióhiba az **“A” ponttól a “B” pontig** folyamatosan növekszik, miközben a motor nyomatéka nő, mert a szerszám nyomja a darabot. Azért, hogy a vezérlő ne fusson szervó hibára, a funkció végrehajtása közben, egy az N0520 Serrl2 paraméteren beállítottnál nagyobb hibahatár állítható be az N3019 Servo Limit during Torque Limit Skip paraméteren. Csak ennek a határnak a túllépése után küld a vezérlő **“3157 Szervo hiba G31 P98 közben”** üzenetet. Ez a szervóhiba nem okoz vészállapotot, de felfüggeszti a program végrehajtását.

A nyomatékhatár elérésének pillanatában a **tengely “A” pontban bejegyzett pozíciója eltárolódik** az alábbi **makrováltozókon**:

#5061, vagy #100151, vagy #_ABSKP[1]: 1. tengely pozíciója

#5062, vagy #100152, vagy #_ABSKP[2]: 2. tengely pozíciója

...

A fenti makrováltozókon **eltárolt pozíció**:

– ha **jött nyomatékhatár jel**, az **“A” pont** pozíciója,

– ha **nem jött nyomatékhatár jel**, a G31 P98 mondat programozott **végpontjának** pozíciója.

A pozícióadat

– mindig az **aktuális munkadarab koordinátarendszerben**,

– az aktuális **hosszkorrekció** (G43, G44) **figyelembe vétele nélkül** kerül eltárolásra.

Hibajelzések

Ha a Q cím nincs kitöltve, **“2004 Q adat hiányzik”** üzenetet ad.

Ha a Q cím értéke kisebb, mint 1, vagy nagyobb, mint 254, “2039 Q megadási hiba” üzenetet küld.

Ha a funkcióban egynél több tengelycímre történik hivatkozás “2035 <tengelycím> tengelyadat megadása hibás” üzenetet ad., ahol a <tengelycím> a tengelyszám szerinti sorrendben a 2. programozott tengely címe.

Ha a programozott tengelyről nem kap a vezérlő nyomaték (áram) információt a “2156 Tapintó állapot hiba a(z) 98 csatornán” üzenetet küld.

21.3 Automatikus szerszámhossz mérés (G36, G37)

A

G36 X_

G37 Z_

utasítás az X koordinátán G36 esetén, illetve a Z koordinátán G37 esetén méri meg és korigálja a beváltott szerszám hosszkorrekcióját. X, Z értéke mindig abszolút adatként kerül értelmezésre, az aktuális munkadarab koordinátarendszerben.

A mozgás a *q* - **Rapid Distance pozícióig** megy **gyorsmenettel**, ahol Rapid Distance az X tengelyre az N3006 Rapid Distance X, a Z tengelyre pedig az N3010 Rapid Distance Z paraméteren beállított távolság, q pedig az X, Z címen programozott érték.

Ezután a mozgás **előtolással** folytatódik, amíg a **mérő-tapintó jele be nem jön, vagy** amíg a vezérlő **hibát nem jelez**. “2104 Tapintási pozíció határon kívül” hibajelzést akkor ad a készülék, ha q (az X,

vagy a Z címen programozott) jószolt mérési pozíció Alarm Distance sugarú környezetén kívül jön meg a tapintó jele. Az Alarm Distance távolság az X tengelyre az N3007 Alarm Distance X, a Z tengelyre pedig az N3011 Alarm Distance Z paraméteren beállított érték.

A **G36, G37 mondatban az előtolás értéke:**

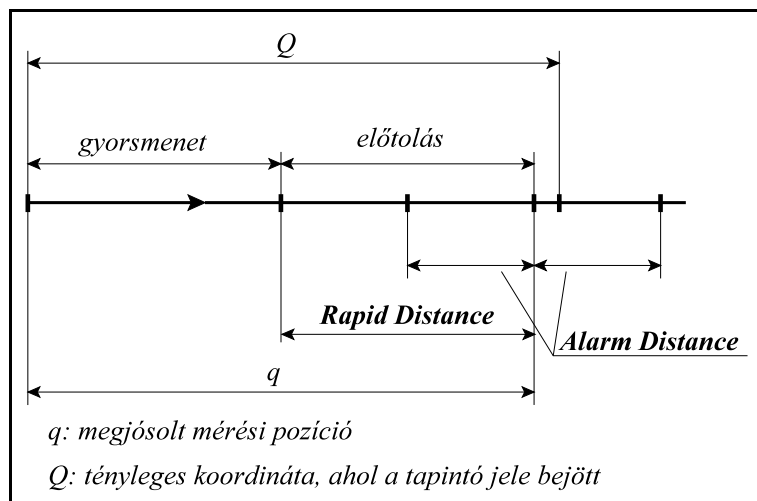
- a programból örökölt előtolás, ha az N3003 G36, G37 Config paraméter #0 TLF bitje 0, vagy
- az N0312 G37 Feed paraméteren beállított érték, ha az N3003 G36, G37 Config paraméter #0 TLF bitje 1.

Ha a mérés sikerrel járt, és a tapintó jele a Q koordinátájú ponton jött be, a következő korrekció módosítási esetek lehetségesek, az előzőleg lehívott hosszkorrekciós regiszterekben:

- ha az N3003 G36, G37 Config paraméter #1 TMW bitje 0, a **geometriai korrekció**,
- ha az N3003 G36, G37 Config paraméter #1 TMW bitje 1, a **kopáskorrekció**

kerül módosításra.

A hosszkorrekció módosítása:



21.3-1 ábra

- ha az N3003 G36, G37 Config paraméter #2 TCA bitje 0, a *q-Q különbséget levonja* a megfelelő *korrekcióból*,
- ha az N3003 G36, G37 Config paraméter #2 TCA bitje 1, a *q-Q különbséget hozzáadja* a megfelelő *korrekcióhoz*.

A mérés megkezdése előtt a megfelelő *a hosszkorrekciót le kell hívni*.

- A G36, G37 utasítás egylovétű.
- A G36, G37 ciklus mindig az aktuális munkadarab koordinátarendszerben kerül végrehajtásra.
- A Rapid Distance és az Alarm Distance paraméterek mindig pozitív értékek. A két paraméterre a következő feltételnek kell teljesülni: Rapid Distance > Alarm Distance.
- A funkciót csak G15, G50, G50.1, G69, G94 állapotban lehet hívni, ellenkező esetben hibajelzést ad.

Mintapélda

```
G55 G15 G50 G50.1 G69 G94
...
G0 X300 Z200
T505
X20 Z100
G37 Z50 F200
Z100
X200
Z50
G36 X40
X200
...
```

22 Biztonsági funkciók

A vezérlőn három biztonsági zóna állítható be:

- **Végállások:** a tengelyek löketének határait határozza meg. Mindig kívülről tiltott területet határol, kapcsolók, vagy paraméterek segítségével.
- **Munkatér határolás:** programból G22, G23 funkcióval kapcsolható be, ki. Beállítható paraméterek megadásával is. Meghatározhat belülről és kívülről tiltott területet is.
- **Belülről tiltott terület:** olyan területet határol, amelynek a belsejébe nem lehet belépni. Paraméterek segítségével állítható be.

22.1 Végállás

A végállások a tengelyek löketét határolják be. A végálláskezelés történhet kapcsolóról, vagy paraméterről. A végállások mindig külső területet tiltanak.

Végálláskezelés kapcsolóról

A végálláskapcsolók jeleit a PLC program kezeli és továbbítja az NC felé. Ha valamelyik tengely végálláskapcsolóra futott “3018 Pozitív végálláskapcsolóra futott # tengelyen”

vagy

“3019 Negatív végálláskapcsolóra futott # tengelyen”

üzenetet küld a vezérlő, ahol # a tengely neve.

A végálláskapcsolóval történő lökethatárolás **hátránya, hogy a kapcsolóra való ráfutás után kezd a vezérlő lassítani**. A nagy gyorsmenettel rendelkező gépeken a hosszú **lassítási út miatt nagyon le kellene a löketet csökkenteni**, hogy még gyorsmenetről is meg tudjon a tengely állni. Ezen kívül a mozgásindítás előtti végállásfigyelés sem működik.

Végálláskezelés paraméterről

A paraméterről történő végálláskezelés esetén a vezérlő minden pillanatban tudja, milyen messze jár a végállástól, és a tengely sebességének függvényében mindig a megfelelő pillanatban kezdi a lassítást. Ezért **a paraméteres végállás maximálisan ki tudja használni a gép löketét**.

Ha valamelyik tengely paraméteres végállásra futott

“3010 Pozitív végállásra futott # tengelyen” vagy

“3011 Negatív végállásra futott # tengelyen”

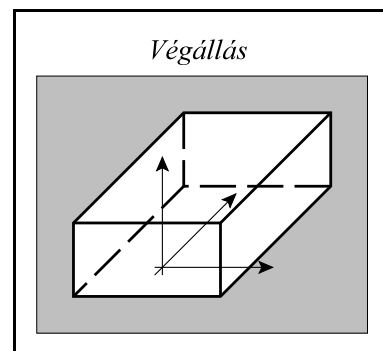
üzenetet küld a vezérlő, ahol # a tengely neve.

A paraméteres végállásokat a gép építője határozza meg.

A paraméteres végállásfigyelés csak referenciapont felvétel után hatásos. Azokon a gépeken, amelyeken nincs abszolút mérőrendszer azaz referenciapontot kell felvenni, referenciapont felvétel előtt a gyorsmenet nagysága be van korlátozva biztonsági okokból. Abszolút mérőrendszerrel szerelt tengelyeken a paraméteres végállás bekapcsolás után azonnal hatásos.

Paraméteren **két végállástartományt**, A-t és B-t lehet beállítani. A gép működése közben a **PLC program határozza meg**, hogy melyik tengelyen és melyik irányban legyen az **A, vagy a B végállás hatásos**.

Például normál esetben a Z tengelyen az “A” végállás hatásos. Ha szerszámcserekor a cserélőkar ráfog a szerszámra, a PLC program átkapcsol a B végállás tartományra és ekkor a Z tengely



22.1-1 ábra

végállástománya olyan szűkre van állítva, hogy a cserélőkart ne tudja leszakítani a Z tengely mozgása.

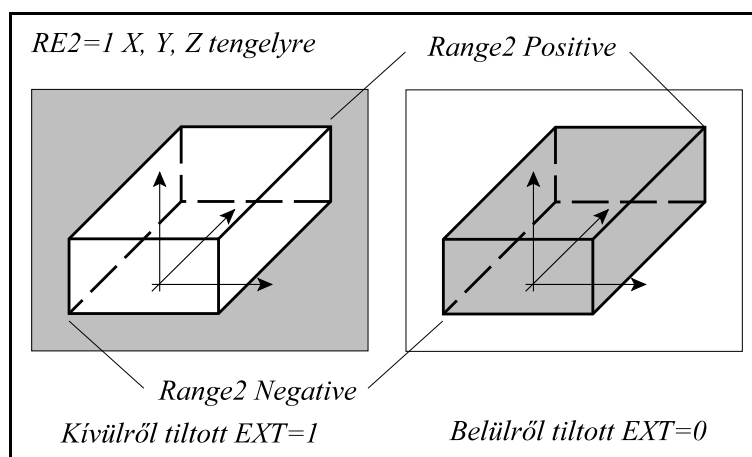
A végállásról való lejövetel

Ha programfutás, vagy kézi mozgatás közben végállásra fut valamelyik tengely, **csak kézi mozgatással lehet lejönni** onnan.

22.2 Paraméterről állítható/programozható munkatér határolás (G22, G23)

Munkatér határolás paraméterről

A következő paraméterek beállításával lehet kijelölni munkatér határolást. A paramétereket csak Szerkesztés (Edit) üzemmódban lehet állítani.



22.2-1 ábra

Az N1000 Range Enable paraméter #1 RE2 paraméterbitjének 1-be állításával kell **kijelölni tengelyenként**, hogy az adott tengely vegyen-e részt a munkatér határolásban.

Az N1006 Range2 Positive és az N1007 Range2 Negative paraméteren kell megadni **gépi koordinátarendszerben** a korlátozni kívánt munkatér pozitív és negatív irányú pontját tengelyenként.

Az

N1006 Range2 Positive > N1007 Range2 Negative
feltételnek az összes kijelölt tengelyre teljesülnie kell!

Paraméteren a gépen létező **összes tengely kijelölhető** az engedélyező bit és a gépi pozíciók segítségével munkatérhatárolásra.

Az N1001 StrkContr paraméter #0 EXT bitjén lehet megadni, hogy a **kijelölt munkatér belülről** (EXT=0 esetén), **vagy kívülről** (EXT=1 esetén) **legyen tiltva**.

Az N1001 StrkContr paraméter #1 STE bitjének 1-be írásával **kell a munkatér határolást engedélyezni**. Ha az STE bit 0, a munkatér határolás nem működik.

Az EXT és az STE biteket csatornánként kell beállítani. A munkatér határolás mindig az adott csatornához tartozó tengelyekre vonatkozik.

A paraméterek beírása, vagy átírása után valamelyik kézi üzemben ki kell adni a

G23

egyedi mondatot. Ennek hatására veszi figyelembe a beírt, vagy módosított paramétereket.

Az ábra az X, Y, Z tengelyekre mutatja a munkatér határolást. Természetesen ennél kevesebb és több tengelyt is ki lehet jelölni erre a funkcióra.

Ha a gép belülről tiltott területre fut (EXT=0)

“3042 Belülről tiltott terület 2”

hibát üzen a vezérlő.

Ha a gép kívülről tiltott területre fut (EXT=1) pozitív, vagy negatív irányban

“3040 Tiltott terület 2 # +” vagy

“3041 Tiltott terület 2 # -”

üzenetet küld a vezérlő, ahol # a tengely neve.

Kijövetel paraméterrel tiltott területből

Ha programfutás, vagy kézi mozgatás közben tiltott területre kerül valamelyik tengely az eljárás a következő:

Ha a *munkatér kívülről van tiltva*.

A területről *kézi mozgatással* le lehet jönni, hasonlóan a végállásokról való lejövetelhez.

Ha a *munkatér belülről van tiltva*.

PLC jelző bekapcsolásával: A programot *reseteljül* le, a *hibát töröljük*. A CP_LIM2DIS PLC jelző bekapcsolásával (például egy *nyomógomb megnyomásával és nyomva tartásával*) fel lehet függeszteni a 2-es tiltott terület figyelését, és ki lehet jönni onnan a paraméterek átírása nélkül. *A részletekről kérdezze a gép építőjét.*

Paraméterállítással: az *STE=0 paraméterállítással kapcsoljuk ki* a munkatér határolás figyelését, egyedi mondatban adjunk ki G23 parancsot, majd *kézi mozgatással hagyjuk el a területet*, majd *kapcsoljuk vissza a figyelést* STE=1 paraméterállítással és újból adjunk ki G23 parancsot.

Munkatér határolás programból

A

G22 X Y Z I J K P

utasítás bekapcsolja a munkatér határolás figyelését. Az utasítással a tengelyek mozgástartománya korlátozható be. Az utasítás címeinek jelentése:

X: az X tengelyen a pozitív irányú határ,

I: az X tengelyen a negatív irányú határ,

Y: az Y tengelyen a pozitív irányú határ,

J: az Y tengelyen a negatív irányú határ,

Z: az Z tengelyen a pozitív irányú határ,

K: az Z tengelyen a negatív irányú határ,

A következő feltételeknek kell teljesülni a megadott adatokra:

$$X \geq I, Y \geq J, Z \geq K$$

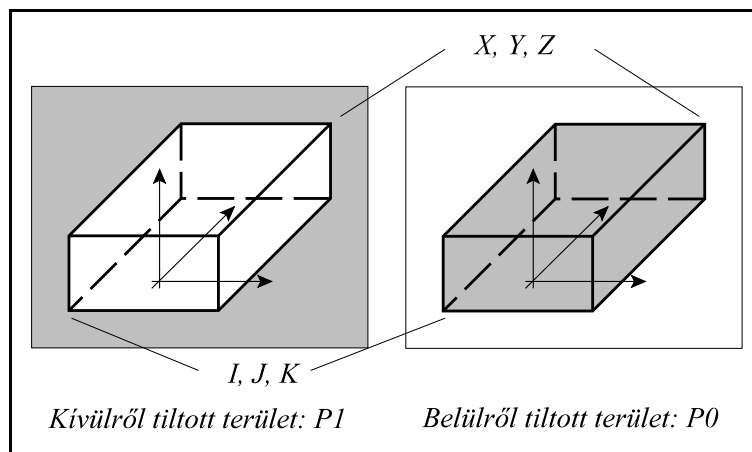
Az összes koordinátaadatot (X, Y, Z, I, J, K) a gépi koordinátarendszerben kell megadni.

P címen adható meg, hogy a kijelölt téren kívülre, vagy belülre nem szabad menni.

P=0 esetén a kijelölt tér belseje van tiltva,

P=1 esetén a kijelölt tér külseje van tiltva.

A G22 utasítás hatására a vezérlő nem veszi figyelembe a 2-es tartomány paraméteren beállított értékeit, csak azokat az értékeket, amelyeket a G22 utasításban megadtunk.



22.2-2 ábra

A

G23

utasítás kikapcsolja a munkatérhatárolás figyelését.

A G23 utasítás törli a G22 utasításban beállított tartományhatárokat, és egyben visszaállítja a 2-es tartományfigyelés paraméteren beállított értékeit.

- Munkatérhatárolást csak a **főtengelyekre** lehet megadni.
- A G22, G23 utasításokat önálló mondatban kell megadni.
- A munkatér határolása bekapcsolás és gépi referenciapont felvétel után lesz hatásos.
- Ha $X=I, Y=J, Z=K$ és $P=0$ a teljes tér engedélyezve van.
- Ha $X=I, Y=J, Z=K$ és $P=1$ a teljes tér tiltva van.

Ha a gép belülről tiltott területre fut (G22 P0)

“3042 Belülről tiltott terület 2”

hibát üzen a vezérlő.

Ha a gép kívülről tiltott területre fut (G22 P1) pozitív, vagy negatív irányban

“3040 Tiltott terület 2 # +” vagy

“3041 Tiltott terület 2 # -”

üzenetet küld a vezérlő, ahol # a tengely neve.

Kijövetel G22 funkcióval programozott tiltott területből

Ha programfutás, vagy kézi mozgás közben tiltott területre kerül valamelyik tengely az eljárás a következő:

Ha a **munkatér kívülről van tiltva.**

A területről **kézi mozgással** le lehet jönni, hasonlóan a végállásokról való lejövetelhez.

Ha a **munkatér belülről van tiltva.**

PLC jelző bekapcsolásával: A programot **reseteljül** le, a **hibát töröljük**. A CP_LIM2DIS PLC jelző bekapcsolásával (például egy **nyomógomb megnyomásával és nyomva tartásával**) fel lehet függeszteni a 2-es tiltott terület figyelését, és ki lehet jönni onnan. *A részletekről kérdezze a gép építőjét.*

Programból: Kézi üzemben **G23 funkcióval kapcsoljuk ki** a munkatér határolás figyelését, **kézi mozgatóval hagyjuk el a területet**, majd **kapcsoljuk vissza a figyelést** a teljes G22 utasítás újbóli kiadásával.

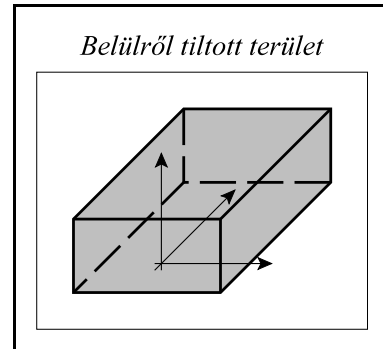
Ha a vezérlőn a 2. tiltott terület be van állítva paraméterről, de G22 utasítással kiadunk munkatér határolást, mindaddig a G22-vel megadott terület lesz tiltva, amíg G23-mal nem töröljük azt. Ezután újra a paraméteren megadott 2-es terület lesz tiltva.

22.3 Belülről tiltott terület

A vezérlőn paraméterek segítségével be lehet állítani egy mindig belülről tiltott területet. Ha valamelyik tengely, esetleg több tengely erre a területre, vagy annak határára fut, a vezérlő "3043 Belülről tiltott terület 3" üzenetet küld.

Kijövetel belülről tiltott területről

PLC jelző bekapcsolásával: A programot *reseteljük* le, a *hibát töröljük*. A CP_LIM3DIS PLC jelző bekapcsolásával (például egy *nyomógomb megnyomásával és nyomva tartásával*) fel lehet függeszteni a 3-as, belülről tiltott terület figyelését, és ki lehet jönni onnan a paraméterek átírása nélkül. *A részletekről kérdezze a gép építőjét.*



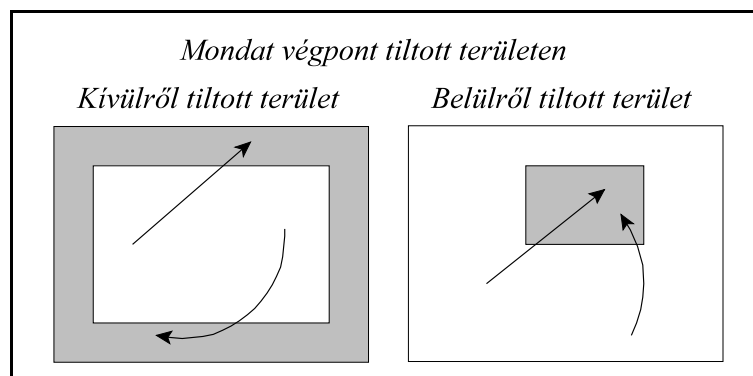
22.3-1 ábra

Paraméterállítással: Ha a fenti, belülről tiltott területre fut a gép, az N1000 Range Enable paraméteren a #2 RE3 bitet, a belülről tiltott terület figyelésének engedélyezését, ki kell kapcsolni az összes tengelyen, az RE3 bitek 0-ba írásával. Ezután kézi mozgatóval ki kell jönni a területről, majd a #2 RE3 biteket vissza kell kapcsolni.

Az RE3 biteket csak Szerkesztés (Edit) üzemmódban lehet állítani.

22.4 Tiltott tartomány figyelés mozgásindítás előtt.

Az N1001 StrkCont paraméter #2 CBM=1 állásánál, *mielőtt egy mozgásparancsot elindítana* a vezérlő automata, vagy kézi adatbeviteli üzemben, vagy egyedi mondat indításánál, *megvizsgálja, hogy az adott mondat végpontja nem esik-e valamelyik tiltott területre.*



22.4-1 ábra

Ha a mondat végpontja az **1-es tiltott területen, végálláson túlra esik:**

“2056 Végpont pozitív végálláson az # tengelyen”

“2057 Végpont negatív végálláson az # tengelyen”

üzenetet küld a vezérlő, ahol # a vonatkozó tengely neve, és **a mozgás nem indul el.**

Ha a mondat végpontja a **2-es, belülről tiltott területre esik**

“2060 Végpont 2-es belülről tiltott területen”

üzenetet küld a vezérlő és **a mozgás nem indul el.**

Ha a mondat végpontja a **2-es, kívülről tiltott területre esik**

“2058 Végpont tiltott területen 2 # +”

“2059 Végpont tiltott területen 2 # -”

üzenetet küld a vezérlő, ahol # a vonatkozó tengely neve, és **a mozgás nem indul el.**

Ha a mondat végpontja a **3-as, belülről tiltott területre esik**

“2061 Végpont 3-as belülről tiltott területen”

üzenetet küld a vezérlő és **a mozgás nem indul el.**

Az N1001 StrkCont paraméter #2 CBM=1 állásánál, **mielőtt egy mozgásparancsot elindítana** a vezérlő automata, vagy kézi adatbeviteli üzemben, vagy egyedi mondat indításánál, **megvizsgálja, hogy az adott mondat pályája nem metsz-e belülről tiltott területet.**

Ha a **mondat pályája metszi a 2-es, belülről tiltott területet**, de a végpont nincs a tiltott terület belsejében

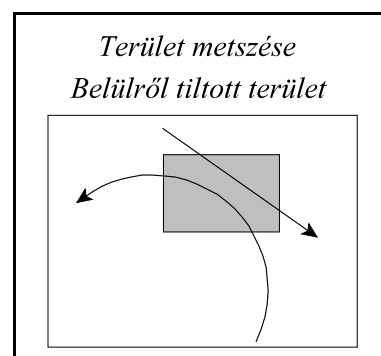
“2062 Belépés 2-es belülről tiltott területre”

üzenetet küld a vezérlő és **a mozgás nem indul el.**

Ha a **mondat pályája metszi a 3-as, belülről tiltott területet**, de a végpont nincs a tiltott terület belsejében

“2062 Belépés 3-as belülről tiltott területre”

üzenetet küld a vezérlő és **a mozgás nem indul el.**



22.4-2 ábra

A fenti esetekben a programozott koordináták átírásával, nullpontok, vagy szerszámkorrekciók módosításával tudjuk a hibát elhárítani.

23 A makróprogramozás

A hagyományos NC programnyelv G, M, S, T kódok megadásával írja le a kívánt pályát, kapcsolja be-, vagy ki a különböző funkciókat. Az egyes címeknek konkrét számértéket adunk meg. Például, ha az X50 Y100 pozícióba akarjuk küldeni a tengelyeket, akkor

```
G0 X50 Y100
```

mondatot programozunk.

A makrónyelv utasításait alkalmazva, *nem szükséges egy adott címnek konkrét számértéket adni*, például X50, hanem egy *változó értékét is adhatjuk neki*, például írhatjuk a programba azt is, hogy

```
G#105 X#102 Y#110
```

ahol #105, #102, #110 három különböző változó értéke, amelyeknek korábban valamilyen értéket adtunk.

A programnyelvben használhatunk különböző *aritmetikai kifejezéseket* és *függvényeket* is, mint például összeadás, négyzetgyök, szinusz, stb.

Lehet *értékadó*, *feltételvizsgáló*, *elágazó*, *ciklust* végrehajtó utasításokat használni.

A programnyelv lehetővé teszi olyan alprogramok, *makrók*, *hívását*, amelyeknek *argumentumokat* (paramétereket) *lehet átadni* a hívó mondatból.

Paramétermegadással létrehozhatunk olyan u.n. *rendszermakrókat* illetve *rendszerprogramokat*, amelyekkel a hagyományos G kódos programnyelv utasításait a felhasználó saját igényei alapján bővítheti, módosíthatja.

23.1 A programnyelv változói

A főprogramban, az alprogramokban és a makrókban az egyes *címeknek változókat is lehet adni* konkrét számértékek helyett.

A *változóknak érték adható* a megengedett értékhatáron belül. Változók használatával a programozás sokkal rugalmasabbá tehető.

A változók lehetnek

lokális változók, amelyeket makróhívásokban argumentumátadásra használhatunk,
globális változók, amelyeket minden makróhívás szinten el lehet érni, illetve a
rendszerváltozók.

A rendszerváltozók az NC olyan belső adatai, amelyeket ki lehet olvasni, vagy át lehet írni az alkatrészprogramból.

23.1.1 Hivatkozás változókra

Változókra hivatkozhatunk *számmal*, a rendszerváltozókra pedig vagy számmal, vagy *szimbolikusan*.

Változóra való hivatkozást mindig a #jel vezeti be.

Hivatkozás számmal megadva

A *#jel után következő szám* jelöli ki a változó azonosítóját:

#<szám>

Például:

#12

#138

#5106

Változóra *indirekt* módon, formulával is *hivatkozhatunk*: #[<formula>]

Például:

#[#120]

jelentése: a 120-as változóban található az a szám, ahányadik változóra hivatkozunk.

#[#120-4]

jelentése: a 120-as változóban található számból 4-et levonva kapjuk a hivatkozott változó számát.

Hivatkozás rendszerváltozókra szimbólummal

A szimbolikus hivatkozást is a *#jel* vezeti be, majd utána következik az *_ (aláhúzás)* karakter:

#_<szimbólum>

Például:

#_ALM (hibaüzenet)

A szimbolikus változóknak egyes esetekben *indexet* is kell adni. Az indexet a *[és a] szögletes zárójelek* közé kell tenni:

#_<szimbólum>[index]

Például:

#_ABSIO[3]

jelentése a 3. tengely mondatvégi pozíciója, tengelyindex: [3].

A programmondat szavaiban a különböző címek nem csak számértéket vehetnek föl, hanem változók értékeit is. A címek után változóra való hivatkozás esetén is használható a "-" mínusz előjel, illetve az I operátor, ahol ez megengedett számértékek esetén. Például:

G#102

ha #102=1.0, akkor ez a hivatkozás G1-gyel egyenértékű

XI-#24

ha #24=135.342, akkor ez a hivatkozás XI-135.342-vel egyenértékű

– mondatszám: N, feltételes mondat: / címei után változóra való hivatkozás nem megengedett.

Mindig a mondatba írt első N címet tekinti mondatszámnak.

– Egy változó számát nem helyettesíthetjük változóval, azaz nem írhatjuk, hogy

##120 (hibás).

A helyes megadás:

#[#120] (jó).

– Ha cím után használjuk a változót, a változó értéke nem haladhatja meg az adott címre megengedett értéktartományt. Például,

#112=123456789

értékkadás után az

M#112

hivatkozás hibajelzést eredményez.

– Ha cím után használjuk a változót, a változó értéke a címnek megfelelő értékes jegyre kerekítődik. Például:

#112=3.23 esetén M#112 M3 lesz,

#112=3.6 esetén M#112 M4 lesz.

23.1.2 A makróváltozók számbábrázolása

A makróváltozók kevés kivételtől eltekintve lebegőpontos számok. A nem lebegőpontos makróváltozókat a leírásában külön jelöljük.

A vezérlőben a lebegőpontos számok ábrázolása az *IEEE754*-es szabvány *dupla pontosságú, lebegőpontos* számbábrázolását követi. Ezeket a számokat 64 biten ábrázoljuk.

A dupla pontosságú, lebegőpontos számbábrázolással a

$\pm 5.0 \times 10^{-324}$ -tól a $\pm 1.7 \times 10^{308}$ -ig

terjedő nagyságú számokat és a 0-t lehet ábrázolni, 15-16 decimális jegy pontossággal. A bevitel során **tizedespontot** (.) kell használni, de egész számok bevitelénél nem kell kitenni:

#100=256

Sem vezető, sem követő nullákat nem kell beadni. A + (pozitív) előjel elhagyható:

#100=134.89654

23.1.3 A lokális változók: #1 – #33

A lokális változókat a makróprogram adott helyen, lokálisan használja.

A lokális változókat általában *argumentumátadásra* használjuk.

A lokális változók *többszintűek*, úgy a főprogramhoz, mind a különböző makróhívásokhoz más és más szint tartozik, ezért nevezzük őket lokálisnak. Vagyis például #1 értéke a főprogramban más lehet mint, mondjuk a makróhívások 2. szintjén.

Makróhívásból való visszatérés után az adott szint lokális változói megsemmisülnek #0-ra, üresre törődnek. A *főprogram lokális változói a program végén kerülnek megsemmisítésre.*

Az argumentumok címe és a lokális változók közötti megfeleltetést, illetve a szintek kezelését a [23.3](#) Makrók, rendszermakrók, rendszeralprogramok hívása fejezet tárgyalja a [319.](#) oldalon. Az a lokális változó, amelynek címe nem szerepel argumentumkijelölésben üres és szabadon felhasználható.

23.1.4 Globális változók: #100 - #499, #500 - #999

A **globális változók** a lokális változókkal ellentétben, **az egész csatornában azonosak** függetlenül, hogy főprogramban, alprogramban, vagy makróban használjuk őket, illetve, hogy a makróhívás melyik szintjén.

A globális változók a rendszerben **teljesen szabad felhasználásúak**, nincs semmi kitüntetett szerepük.

A globális változók között két csoportot különböztetünk meg:

A **#100-tól #499-ig terjedő globális változók törlődnek kikapcsolásra.**

A **#500-tól #599-ig terjedő globális változók értéke kikapcsolás után is megőrződik.**

A **#500-tól #999-ig terjedő makrováltozók írásvédetté tehetők** az N1702 Write Prt Low és az N1703 Write Prt Hig paraméterek segítségével.

Az N1702 Write Prt Low paraméterre a védeni kívánt tömb első elemét, az N1703 Write Prt Hig paraméterre a védetté nyilvánított tömb utolsó elemét írjuk.

Például, ha a #530-tól a #540-ig tartó változókat írásvédetté akarjuk tenni az N1702 Write Prt Low=530-ra és N1703 Write Prt Hig=540-re kell állítani a paramétereket.

Ha a vezérlő több csatornát kezel, a **globális változók egy-egy tömbje** paraméter segítségével **hozzáférhetővé tehető az összes csatornában.**

Az N1700 No. of Common #100 paraméter a globális, minden csatornából lehívható #100-tól #499-ig tartó makrováltozók számát adja meg. Minden csatornában a 100-tól a 100 + No. of Common #100 paraméterig terjedő számú makrováltozók lesznek közösek. Ennek a paraméternek kisebbnek kell lennie, mint 400.

Ha a paraméter értéke pl. 40, a #100-tól #139-ig terjedő makrováltozók az összes csatornára közösek.

Az N1701 No. Of Common #500 a globális, minden csatornából lehívható #500-tól #999-ig tartó makrováltozók számát adja meg. Minden csatornában az 500-tól az 500 + No. of Common #500 paraméterig terjedő számú makrováltozók lesznek közösek. Ennek a paraméternek kisebbnek kell lennie, mint 500.

Ha a paraméter értéke pl. 30, a #500-tól #529-ig terjedő makrováltozók az összes csatornára közösek.

23.1.5 Rendszerváltozók leírásánál használt jelölések

A rendszerváltozók az NC olyan belső adatai, amelyeket ki lehet olvasni, vagy át lehet írni az alkatrészprogramból.

A tengelyekhez tartozó, 10000 alatti számmal azonosított makrováltozók csak 1-től 20-ig azonosíthatók a tengelyek. Nagyobb tengelyszámok esetére be lettek vezetve a 100000 fölötti azonosítószámok. Ezek a számok 1-től 50-ig hivatkozhatunk tengelyekre. Például:

#100001 az 1. tengely mondatvégi pozícióját

azonosíthatjuk a fenti számmal is. Természetesen hivatkozhatunk a fenti adatokra 10000 alatti számokkal és szimbólummal is.

A rendszerváltozók leírásánál használt jelölések a következők:

[n]: a változó indexe. Lehet pl. tengely-, vagy orsószám

R: a változó tulajdonsága: csak olvasható változó

W: a változó tulajdonsága: csak írható változó

R/W: a változó tulajdonsága: olvasható és írható változó.

23.1.6 Az üres változó. Konstansok

Száma	Szimbóluma	Tulajd.	Leírása
#0, #3100	#_EMPTY	R	“üres” konstans
#3101	#_PI	R	$\pi = 3.14159\dots$
#3102	#_E	R	Természetes szám: $e = 2.71828\dots$

Üres változó #0, #3100, #_EMPTY (R)

Makróhíváskor, ha egy címnek nem adtunk értéket, a makró törzsében az ahhoz a címhez tartozó lokális változó értéke üres lesz.

Például:

G65 P100 X20 Y30

hívás után az O0100 makróban a #1 lokális változó értéke üres lesz, mert a G65 hívásban az “A” címnek nem adtunk értéket. A makró törzsében a

#1 EQ #0

vizsgálattal dönthetjük el, hogy “A” címet kitöltötték-e a makró hívásakor.

Az üres változó és a 0 szám nem ugyanaz, különböző!

Az üres változónak a természete összevetve azzal, ha egy változó értéke 0:

Üres változóra való *hivatkozás* címben:

ha #1=<üres>

G90 X20 Y#1

|
G90 X20

ha #1=0

G90 X20 Y#1

|
G90 X20 Y0

Üres változó értékadó utasításban:

ha #1=<üres>	ha #1=0
#2=#1	#2=#1
#2=<üres>	#2=0
#2=#1*3	#2=#1*3
#2=0	#2=0
#2=#1+#1	#2=#1+#1
#2=0	#2=0

Feltételvizsgálat esetén az **üres** és **0** értékű változó közti különbség:

ha #1=<üres>	ha #1=0
#1 EQ #0	#1 EQ #0
teljesült	nem teljesült
#1 NE 0	#1 NE 0
teljesült	nem teljesült
#1 GE #0	#1 GE #0
teljesült	nem teljesült
#1 GT 0	#1 GT 0
teljesült	nem teljesült

23.1.7 Az alaktrészprogram és a PLC program közötti változók

Az alaktrészprogram és a PLC program között az információcsere az alábbi változókon keresztül valósítható meg.

☞ **Figyelem!** Azt, hogy a különböző rendszerváltozókon keresztül a PLC program milyen információkat ad át az alaktrészprogramnak, illetve milyen információkat fogad az alaktrészprogramtól, a gép építője határozza meg.

Száma	Szimbóluma	Tulajd.	Leírása
#1000...#1031	#_UI[n] n=0 - 31	R	A PLC programnak az NC felé átadott 32 db. bites változója . Értékkészlet: 0, 1
#1032	#_UIL[n] n=0	R	A PLC programnak az NC felé átadott 32 db. bites változója egy, 32 bites egész számként átadva. Értékkészlet: 0 ... 2 ³² -1

Száma	Szimbóluma	Tulajd.	Leírása
#1033...#1035	#_UIL[n] n=1, 2, 3	R	A PLC programnak az NC felé átadott 3 db. <i>lebegőpontos</i> változója.
#1100...#1131	#_UO[n] n=0 - 31	R/W	Az alkatrész programnak a PLC felé átadott 32 db. <i>bites változója</i> . Értékkészlet: 0, 1
#1132	#_UOL[n] n=0	R/W	Az alkatrész programnak a PLC felé átadott 32 db. bites változója egy, 32 bites egész számként átadva. Értékkészlet: 0 ... $2^{32}-1$
#1133...#1135	#_UOL[n] n=1, 2, 3	R/W	Az alkatrész programnak a PLC felé átadott 3 db. <i>lebegőpontos</i> változója.

23.1.8 Az akatrészprogram üzenetei

Makroprogramból hibát jelezhetünk, illetve üzenetet küldhetünk a kezelőnek:

Száma	Szimbóluma	Tulajd.	Leírása
#3000	#_ALM	W	Hibaüzenet, Cancel gombbal törölhető
#3006	#_MSGSTP	W	Megállás üzenettel, Startra továbbmegy
#3106	#_MSG	W	Üzenet kiírása a programlista ablakba
#3107	#_MSGBOX	R/W	Üzenetküldés Windows ablakba

Hibajelzés: #3000, #_ALM (W)

A

#3000=nnn(HIBAJELZÉS)

vagy a

#_ALM=nnn(HIBAJELZÉS)

értékadással számmal jelzett (nnn: max. három számjegy), és/vagy szöveges hibaüzenetet adhatunk. A szöveget (,) gömbölyű zárójelek közé kell tenni.

Ha a makróban hibát észlel a program, azaz olyan ágra fut, ahol a #3000 változónak értéket adtunk, az előző mondatig végrehajtja a programot, majd a végrehajtást felfüggeszti és a képernyőn megjelenik a zárójelek között megadott hibaüzenet, illetve az üzenet kódja

ii4nnn00 (ii: a csatorna száma, amelyben a hiba előfordult)

formában, vagyis a #3000 értéken megadott nnn számhoz hozzáad 4000-et. Ha számot nem adtunk az üzenet kódja 4000 lesz, ha szöveget nem adtunk, csak a kód jelenik meg. A hibaüzenet a CANCEL gombbal szüntethető meg.

Megállás üzenettel: #3006, #_MSGSTP (W)

A

#3006=nnn(ÜZENET)

vagy a

#_MSGSTP=nnn(ÜZENET)

értékkadás hatására a program végrehajtása megáll (STOP állapot lesz), és a gömbölyű zárójelek közé zárt üzenet megjelenik a képernyőn, illetve az üzenet kódja

ii5nnn00 (ii: a csatorna száma, amelyben az üzenetet kiadták)

formában, vagyis a #3006 értéken megadott nnn számhoz hozzáad 5000-et. Ha számot nem adtunk az üzenet kódja 5000 lesz, ha szöveget nem adtunk, csak a kód jelenik meg. A program végrehajtása a START gomb lenyomására folytatódik, ekkor az üzenet letörlődik a képernyőről. Az utasítás hasznosan alkalmazható olyan esetben, amikor a program végrehajtása közben kezelői beavatkozás szükséges.

Üzenet kiírása a programlista ablakba: #3106, #_MSG (W)

A

#3106=nnn(ÜZENET)

vagy

#_MSG=nnn(ÜZENET)

értékkadás hatására a *program megállás nélkül folytatódik*, és az *üzenet szövege a Program lista ablak legfelső sorában* megjelenik a következő módon:

MSGnnn: (ÜZENET)

Az üzenet szövege addig marad ott, amíg egy újabb #3106, vagy #_MSG utasítással felül nem írjuk. RESET, program vége (M30) hatására az üzenet törlődik.

Azt, hogy az üzenet szövege megjelenjen a programlista legfelső sorában, *a kiíratást engedélyezni kell* a funkció billentyűk használatával a következő módon:

F5 Nézet - F1 Program lista - F9 Beállítások - F3 #_MSG info

Felhasználható például programrészek végrehajtásának jelzésére:

```
...
#3106=1 (nagyolás D30 maróval)
...
...
#_MSG=2 (símítás D20 maróval)
...
```

Üzenetküldés Windows ablakba: #3107, #_MSGBOX (R/W)

A

#3107=nnn(ÜZENET)

vagy

#_MSGBOX=nnn(ÜZENET)

értékkadás hatására a program végrehajtása megáll (STOP állapot lesz) és a gömbölyű zárójelek közé zárt üzenet szövege, valamint az üzenet kódja megjelenik a vezérlő üzenet sorában

ii6nnnjj (ii: a csatorna száma, amelyben az üzenetet kiadták)

formában. Az üzenetkiírás nem csak a felső státuszszámban jelenik meg, hanem a képernyő közepén egy *Windows üzenetablakban* is.

Az hogy az üzenetablak milyen típusú legyen azt az üzenet kódja határozza meg. Ha az értékkadás **nnn=100 - 199** közötti, akkor az üzenetablak egy „Ok” gombbal jelenik meg,

így a kezelő csak elfogadhatja a kiírt üzenetet. Ha az értékadás

nnn=200 - 299 közötti, akkor az üzenetablak egy „Igen” „Nem” gombbal jelenik meg. Ebben az esetben a kezelőnek lehetősége van választani. Minden más értékadásra hibajelzést ad a vezérlő.

A Windows ablakban megjelenő üzenetekre adott válasz után az üzenet törlődik és a #3107 változó az alábbi értéket veheti fel a különböző esetekben :

- #3107=0: az ablakon **X (bezár)** gombra kattintottak, vagy **CANCEL** gombot nyomtak,
- #3107=1: „OK” gombra kattintottak,
- #3107=2 : „Igen” gombra kattintottak,
- #3107=3 : „Nem” gombra kattintottak,

Üzenetkiírásnál ha az üzenetablakban több sorban szeretnék megjeleníteni az adott üzenetrészt, lehetőség van „\n” karakterrel szétdarabolni a szöveget és így beszúrni soremelés karaktert. Érték kiírásánál a **DPRNT utasításnál leírt makróváltozó formázás** alkalmazható.

1. példa:

```
##3107=100 (Mérési eredmény értékei\nX hossz: #140[53]\nY
hossz: #141[53])
```

2. példa

```
#140=0.3458
#141=0.9123
(mérés)
(korrekció számítás eredmény kiírása)
#_MSGBOX=200 (Szerszám kopás értékei\nX átméro kopás:
#140[53]\nZ irányú kopás: #141[53]\nAkarja az új
korrekciót beírni?)
IF [[#_MSGBOX] EQ 2] GOTO10
GOTO20
N10 (Igen ág)
(kopás beírása)
N20
(program folytatása)
```

23.1.9 Idők, munkadarab számlálók

Száma	Szimbóluma	Tulajd.	Leírása
#3001	#_CLOCK1	R/W	Szabadfelhasználású, msec-es időmérő
#3002	#_CLOCK2	R/W	Forgácsolási idő (msec)
#3011	#_DATE	R	Dátum: év/hó/nap
#3012	#_TIME	R	Óra: óra/perc/másodperc
#3901	#_PRTSA	R/W	Elkészült darabok száma
#3902	#_PRTSN	R/W	Elkészítendő darabok száma

Milliszekundumos időmérő: #3001, #_CLOCK1 (R/W)

A változó értéke írható és olvasható.

Két időpont között eltelt időt mérhetjük meg milliszekundumban. A változó értéke a vezérlő bekapcsolásakor nulláról indul, és felfele számol. Mindig számol, amikor a vezérlő be van kapcsolva. Értékadással, programból lenullázhatjuk a

```
#3001=0,
```

vagy a

```
#_CLOCK1=0
```

utasítással a változót, amely ilyenkor 0-ról indul, majd később le lehet kérdezni programból az értékét a

```
#100=#3001
```

vagy a

```
#100=#_CLOCK1
```

utasítással.

Forgácsolási idő: #3002, #_CLOCK2 (R/W)

A változó értéke írható és olvasható. Az automata üzemmódban, start állapotban, előtolással (G1, G2, stb) eltöltött időt méri milliszekundumban, a vezérlő élete elejétől halmozva. A változó értéke az Idő/számlálók képernyő Forgácsolási idő kijelzőn olvasható ki.

Dátum: #3011, #_DATE (R)

A pillanatnyi dátum olvasható ki a változóból év/hó/nap formátumban.

A

```
#100=#3011,
```

vagy a

```
#100=#_DATE
```

utasítás használata után, ha a #100 változó értéke, pl

```
20140518,
```

az azt jelenti, hogy a dátum: 2014 (év), 05 (május), 18 (18-a).

Óra: #3012, #_TIME (R)

A pillanatnyi idő olvasható ki a változóból óra/perc/másodperc formátumban.

A

```
#100=#3012,
```

vagy a

```
#100=#_TIME
```

utasítás használata után, ha a #100 változó értéke, pl

```
153241,
```

az azt jelenti, hogy az idő: 15 ó (3 ó du.), 32 perc, 41 mp.

Elkészült/elkészítendő darabok száma: #3901, #_PRTSA / #3902, #_PRTSN (R/W)

A változók értékei írhatók és olvashatók. Az elkészült darabok számát a #3901-es #_PRTSA számlálóban gyűjti a vezérlő. A számláló tartalmát minden M02, M30 vagy az N2305 Part Count M paraméteren kijelölt M funkció végrehajtásakor növeli 1-gyel a vezérlő.

Amikor az elkészült darabok száma elérte az elkészítendő darabszámot (#3902-es #_PRTSN számláló) a végrehajtó értesíti a PLC-t.

```
elkészült darabok száma    #3901, #_PRTSA
```

```
elkészítendő darabok száma #3902, #_PRTSN
```

Az elkészült és az elkészítendő darabszám értéke az Idő/számlálók képernyő Elkészült és Elkészítendő kijelzőjén olvasható ki.

23.1.10 Az automata üzem működését befolyásoló változók

Száma	Szimbóluma	Tulajd.	Leírása
#3003	#_CNTL1	R/W	1. kontrol változó (mondatonkénti)
#3003 bit 0	#_M_SBK	R/W	Mondatonkénti végrehajtás elnyomása. Értékkészlet: 0, 1
#3004	#_CNTL2	R/W	2. kontrol változó
#3004 bit 0	#_M_FHD	R/W	Stop elnyomása. Értékkészlet: 0, 1
#3004 bit 1	#_M_OV	R/W	Override és Stop elnyomása. Értékkészlet: 0, 1
#3004 bit 2	#_M_EST	R/W	Pontos megállás elnyomása. Értékkészlet: 0, 1

☞ **Figyelem:** A kontrol változókon beállított bitek értékei resetre és program végére törlődnek!

1. kontrol változó: #3003, #_CNTL1 (R/W)

Ha a #3003, vagy a #_CNTL1 változó értéke 1 (vagy páratlan szám), a mondatonkénti végrehajtás állapotában mindaddig nem áll meg egy mondat végrehajtása után, amíg ennek a változónak az értéke 0 nem lesz.

A mondatonkénti végrehajtás elnyomására bitesen (0, vagy 1 megadásával) is hivatkozhatunk a

#_M_SBK

változó írásával.

Bekapcsolásra, resetre a változó értéke 0.

A mondatonkénti végrehajtás, ha a változó értéke

0: nincs elnyomva

1: el van nyomva.

2. kontrol változó: #3004, #_CNTL2 (R/W)

A #3004, vagy a #_CNTL2 változónak az alábbi értékeket adhatjuk:

#3004, vagy #_CNTL2 értéke	Pontos megállás G61, G9 elnyomása	Override és Stop elnyomása	Stop gomb elnyomása
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

A pontos megállás üzem, az override és a stop gomb, ha a megfelelő oszlopba írt szám

0: nincs elnyomva

1: el van nyomva.

A fenti elnyomásokat bites változókkra történő 0, vagy 1 írásával is megadhatjuk:

#_M_FHD: stop gomb elnyomása:

0: nincs elnyomva

1: el van nyomva.

A mondatonkénzi üzemet nem nyomja el.

#_M_OV: override és stop elnyomása (G63 állapot):

0: nincs elnyomva

1: el van nyomva.

A mondatonkénti üzemet is elnyomja

#_M_EST: Pontos megállás G61, G9 elnyomása (G64 állapot)

0: nincs elnyomva

1: el van nyomva.

23.1.13 A főprogram száma

Száma	Szimbóluma	Tulajd.	Leírása
#4000	#_MAIN0	R	A végrehajtás alatt álló főprogram száma

Mindig az aktuális, végrehajtás alatt álló főprogram számát mutatja, még akkor is, ha éppen egy alprogram végrehajtása van folyamatban. Ha megszakítjuk az automata üzemet és MDI-ben végrehajtunk egy programot, az MDI-ben végrehajtott program számát mutatja.

Ha a főprogram neve nem O-val kezdődik és nem max. 8 számjegyből áll #4000=0 lesz.

23.1.14 Öröklődő információk

A *megelőző mondatban érvényes öröklődő információk* a #4001–#4199–es rendszerváltozók olvasásával állapíthatók meg.

Száma	Szimbóluma	Tulajd.	Leírása
#4001...#4039	#_BUFG[n] n=1-39	R	Csoportonként az utoljára megadott öröklődő G funkció kódja. n: a G kód csoport száma
#4101	#_BUFA	R	Az utoljára megadott segédfunkció értéke, amelynek címét az N1333 Aux Fu Addr1 paraméteren definiáltunk.
#4102	#_BUFB	R	Az utoljára megadott segédfunkció értéke, amelynek címét az N1334 Aux Fu Addr2 paraméteren definiáltunk.
#4103	#_BUFC	R	Az utoljára megadott segédfunkció értéke, amelynek címét az N1334 Aux Fu Addr3 paraméteren definiáltunk.
#4107	#_BUFD	R	Az utoljára megadott D cím értéke
#4108	#_BUFE	R	Az utoljára megadott E cím értéke
#4109	#_BUFF	R	Az utoljára megadott F cím értéke
#4111	#_BUFH	R	Az utoljára megadott H cím értéke
#4113	#_BUFM	R	Az utoljára megadott M cím értéke
#4114	#_BUFN	R	Az utoljára megadott N mondatszám értéke
#4115	#_BUFO	R	Az utoljára meghívott O programszám értéke
#4119	#_BUFS	R	Az utoljára megadott S cím értéke
#4120	#_BUFT	R	Az utoljára megadott T cím értéke
#4130	#_BUFWZP	R	Az utoljára megadott kiterjesztett koordináta-rendszer száma (G54.1 Pn)

A pillanatnyilag *végrehajtás alatt álló mondatban érvényes öröklődő információk* a #4201–4399-es változók olvasásával állapíthatók meg.

Száma	Szimbóluma	Tulajd.	Leírása
#4201...#4239	#_ACTG[n] n=1-39	R	Csoportonként a végrehajtás alatt álló mondatban érvényes öröklődő G funkció kódja. n: a G kód csoport száma
#4301	#_ACTA	R	A végrehajtás alatt álló mondatban érvényes segédfunkció értéke, amelynek címét az N1333 Aux Fu Addr1 paraméteren definiáltunk
#4302	#_ACTB	R	A végrehajtás alatt álló mondatban érvényes segédfunkció értéke, amelynek címét az N1334 Aux Fu Addr2 paraméteren definiáltunk.
#4303	#_ACTC	R	A végrehajtás alatt álló mondatban érvényes segédfunkció értéke, amelynek címét az N1334 Aux Fu Addr3 paraméteren definiáltunk.
#4307	#_ACTD	R	A végrehajtás alatt álló mondatban érvényes D cím értéke
#4308	#_ACTE	R	A végrehajtás alatt álló mondatban érvényes E cím értéke
#4309	#_ACTF	R	A végrehajtás alatt álló mondatban érvényes F cím értéke
#4311	#_ACTH	R	A végrehajtás alatt álló mondatban érvényes H cím értéke
#4313	#_ACTM	R	A végrehajtás alatt álló mondatban érvényes M cím értéke
#4314	#_ACTN	R	A végrehajtás alatt álló mondatban érvényes N mondatszám értéke
#4315	#_ACTO	R	A végrehajtás alatt álló mondatban érvényes O programszám értéke
#4319	#_ACTS	R	A végrehajtás alatt álló mondatban érvényes S cím értéke
#4320	#_ACTT	R	A végrehajtás alatt álló mondatban érvényes T cím értéke
#4330	#_ACTWZP	R	A végrehajtás alatt álló mondatban érvényes

A megszakítási makró hívásának pillanatában **a megszakított mondatban érvényes öröklődő információk** a #4401–4599-es változók olvasásával állapíthatók meg.

Száma	Szimbóluma	Tulajd.	Leírása
#4401...#4439	#_INTG[n] n=1-39	R	Csoportonként a megszakított mondatban érvényes öröklődő G funkció kódja. n: a G kód csoport száma
#4501	#_INTA	R	A megszakított mondatban érvényes segéd-funkció értéke, amelynek címét az N1333 Aux Fu Addr1 paraméteren definiáltunk
#4502	#_INTB	R	A megszakított mondatban érvényes segéd-funkció értéke, amelynek címét az N1334 Aux Fu Addr2 paraméteren definiáltunk.
#4503	#_INTC	R	A megszakított mondatban érvényes segéd-funkció értéke, amelynek címét az N1334 Aux Fu Addr3 paraméteren definiáltunk.
#4507	#_INTD	R	A megszakított mondatban érvényes D cím értéke
#4508	#_INTE	R	A megszakított mondatban érvényes E cím értéke
#4509	#_INTF	R	A megszakított mondatban érvényes F cím értéke
#4511	#_INTH	R	A megszakított mondatban érvényes H cím értéke
#4513	#_INTM	R	A megszakított mondatban érvényes M cím értéke
#4514	#_INTN	R	A megszakított mondatban érvényes N mondat-szám értéke
#4515	#_INTO	R	A megszakított mondatban érvényes O programszám értéke
#4519	#_INTS	R	A megszakított mondatban érvényes S cím értéke
#4520	#_INTT	R	A megszakított mondatban érvényes T cím értéke
#4530	#_INTWZP	R	A megszakított mondatban érvényes kiterjesztett koordinátarendszer száma (G54.1 Pn)

☞ **Figyelem!** A pontot tartalmazó öröklődő G kódok esetén pl. G51.1, pont nélkül adja vissza a kódot úgy, hogy a pont utáni számot az elejére írja:

#100=#4022

utasítás hatására #100 értéke 151 lesz G51.1 állapotban és 150 lesz G50.1 állapotban.

Az öröklődő G kódok csoportszám szerinti rendezését az alábbi táblázat mutatja. Az egylövetű, nem öröklődő, G kódok a 0-ás csoportba tartoznak. A 0-ás csoportot nem lehet lekérdezni, ezért az oda tartozó G kódokat nem tüntettük fel a táblázatban.

Csoport-szám	A csoportba tartozó öröklődő G kód	Funkció
1	G0, G1, G2, G3, G33, G77, G78, G79	interpoláció
2	G17, G18, G19	síkválasztás
3	G90, G91	abszolút/inkrementális programozás
4	G22, G23	tiltott terület be/ki
5	G94, G95	percenkénti/fordulatonkénti előtolás
6	G20, G21	inch/metrikus adatmegadás
7	G40, G41, G42	sugárkorrekció ki/be
8	G43, G44, G49	hosszkorrekció be/ki
9	G83.1, G84.1, G86.1, G80, G81, G82, G83, G84, G84.2, G84.3, G85, G86, G87, G88, G89	fűróciklusok
10	G98, G99	visszatérés fűróciklusból kiindulási/R pontra
11	G50, G51	léptékezés ki/be
12	G66, G66.1, G67	öröklődő makrohívás be/ki
13	G96, G97	konstans vágósebesség be/ki
14	G54, G54.1, G55, G56, G57, G58, G59	munkadarab koordináta rendszer választás
15	G61, G62, G63, G64	előtolás vezérlő funkciók
16	G68, G69	síkbeli forgatás be/ki
17	G15, G16	adatmegadás polárkoordinátával ki/be
18		

23.1 A programnyelv változói

Csoport- szám	A csoportba tartozó öröklődő G kód	Funkció
19		
20		
21	G12.1, G13.1	polárinterpoláció be/ki
22	G50.1, G51.1	tükrözés ki/be
23		
24		
25		
26		
27		
28		
29		
30		
31	G50.2, G51.2	sokszög esztergálás ki/be
32		
33		
34	G80.8, G81.8	elektronikus hajtómű ki/be
35		
36		
37		
38		
39		

23.1.15 Pozícióinformációk

Száma	Szimbóluma	Tulajd.	Leírása
#5001...#5020	#_ABSIO[n] n=1-50	R	Az 1...n tengely mondatvégi pozíciója
#100001...#100050			
#5021...#5040	#_ABSMT[n] n=1-50	R	Az 1...n tengely pozíciója a gépi koordinátarendszerben
#100051...#100100			
#5041...#5060	#_ABSOT[n] n=1-50	R	Az 1...n tengely pozíciója a munkadarab koordinátarendszerben
#100101...#100150			
#5061...#5080	#_ABSKP[n] n=1-50	R	Az 1...n tengely tapintási pozíciója a munkadarab koordinátarendszerben
#100151...#100200			

☞ **Figyelem!** A pozícióinformációk kiolvasása tengelyszám alapján történik. Mielőtt az utasítást használja kérjen felvilágosítást a gépépítőjétől, hogy melyik tengelynek mi a száma.

Például vegyünk egy kétcsatornás vezérlőt. Az első csatornában van X, Y, Z tengely, a másodikban X,Z. A gép építője a következőképp osztotta ki a tengelyeket:

1. csatorna:

X – 1. tengely

Y – 2. tengely

Z – 3. tengely

2. csatorna

X – 4. tengely

Z – 5. tengely

Ha az 1. csatornában akarjuk a pozícióinformációkat lekérdezni a Z tengelyen a 3-as indexet, ha a másodikban az 5-ös indexet kell használni.

Tengelyek mondatvégi pozíciója: #5001...#5020, #100001...#100050, #_ABSIO[n] (R)

A tengelyek mondatvégi pozíciója

az aktuális munkadarab koordinátarendszerben (G54, G55, ...),

derékszögű koordinátákban,

az összes korrekció (hossz, sugár) *figyelembe vételével* nélkül

kerül visszaadásra.

Tengelyek gépi pozíciója: #5021...#5040, #100051...#100100, #_ABSMT[n] (R)

A tengelyek gépi pozíciója

a gép koordinátarendszerében (G53),

kerül visszaadásra. Ha az adott tengely mozog, a visszaadott érték állandóan változik.

Tengelyek munkadarab pozíciója:#5041...#5060, #100101...#100150, #_ABSOT[n] (R)

A tengelyek munkadarab pozíciója

- az aktuális munkadarab koordinátarendszerben (G54, G55, ...),
- derékszögű koordinátákban,
- az összes korrekció (hossz, sugár) *figyelembe vételével*

kerül visszaadásra. Ha az adott tengely mozog, a visszaadott érték állandóan változik.

Tapintási pozíciók:#5061...#5080, #100151...#100200, #_ABSKP[n] (R)

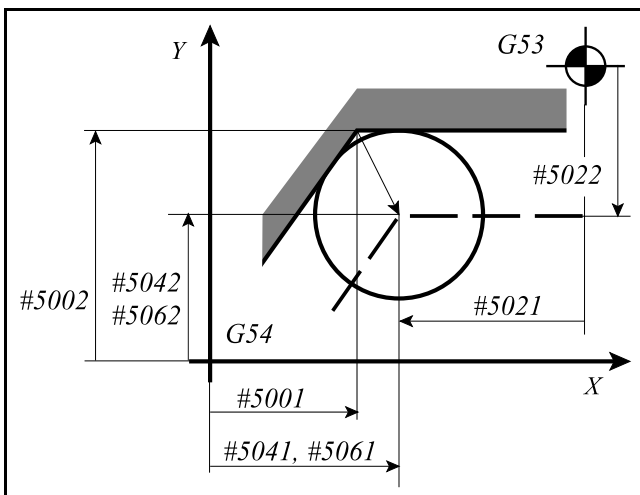
G31 mondatban az a pozíció, ahol a tapintó jele megjött

- az aktuális munkadarab koordinátarendszerben (G54, G55, ...),
- derékszögű koordinátákban,
- az összes korrekció (hossz, sugár) *figyelembe vételével*

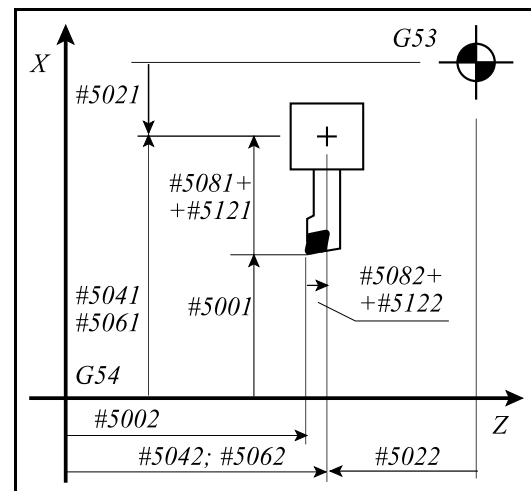
kerül visszaadásra.

Ha a tapintó jele nem jött meg, a fenti változók a G31 mondatban programozott végponti pozíciót veszik fel.

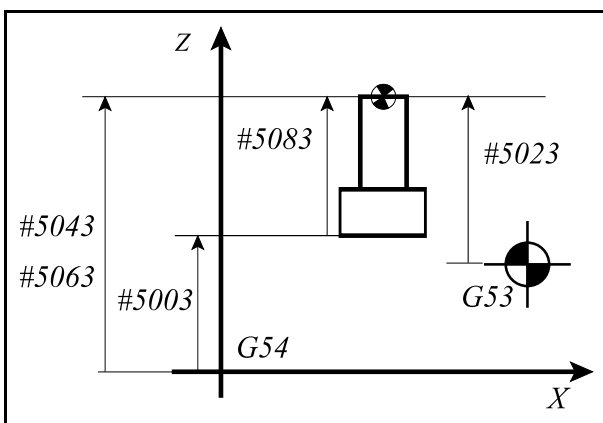
A tapintó jelének megjövetele után a tengely lelassít és megáll. A G31 mondat végponti pozíciója #_ABSIO(n) a lassítás és megállás utáni értéket veszi fel.



23.1.15-1 ábra



23.1.15-2 ábra



23.1.15-3 ábra

23.1.16 Az aktuális hosszkorrekció értéke

Maró korrekciók használata esetén

Száma	Szimbóluma	Tulajd.	Leírása
#5081...#5100	#_TOFS[n]	R	Az 1...n tengelyen figyelembe vett hosszkorrekció értéke
#100201...#100250	n=1-50		

A fenti változókról a végrehajtás alatt álló mondatban tengelyenként figyelembe vett hosszkorrekció (geometriai + kopás) értéke olvasható ki.

Eszterga korrekciók használata esetén

Száma	Szimbóluma	Tulajd.	Leírása
#5081	#_TOFSWX	R	Az X tengelyen figyelembe vett kopáskorrekció értéke
#100201			
#5082	#_TOFSWZ	R	Az Z tengelyen figyelembe vett kopáskorrekció értéke
#100202			
#5083	#_TOFSWY	R	Az Y tengelyen figyelembe vett kopáskorrekció értéke
#100203			
#5121	#_TOFSGX	R	Az X tengelyen figyelembe vett geometriai hosszkorrekció értéke
#100901			
#5122	#_TOFSGZ	R	Az Z tengelyen figyelembe vett geometriai hosszkorrekció értéke
#100902			
#5123	#_TOFSGY	R	A Y tengelyen figyelembe vett geometriai hosszkorrekció értéke
#100903			

A fenti változókról a végrehajtás alatt álló mondatban az X, Z és Y tengelyen figyelembe vett hosszkorrekció (geometriai + kopás) értéke olvasható ki.

23.1.17 Egyéb pozícióinformációk

Száma	Szimbóluma	Tulajd.	Leírása
#5101...#5120	#_SVERR[n]	R	Az 1...n tengely lemaradása
#100251...#100300	n=1-50		
#100651...#100700	#_MIRTP[n]	R	Az 1...n tengelyen a kézikerékkel eltolt pozíció
	n=1-50		

23.1 A programnyelv változói

Száma	Szimbóluma	Tulajd.	Leírása
#5181...#5200	#_DIST[n]	R	Az 1...n tengelyen a maradékút értéke
#100801...#100850	n=1-50		

Tengelyek lemaradása: #5101...#5120, #100251...#100300, #_SVERR[n] (R)

A változókról kiolvashatók a tengelyek szervokörének lemaradása (követési hibája), bemeneti mértékegységben.

A kézikerékkel eltolt pozíciók: #100651...#100700, #_MIRTP[n] (R)

Ha automata üzemben, akár mozgás közben, kézikerék mozgatással korrigáljuk a tengelyek pozícióját (nullpontját), a korrekció mértéke a fenti változókról olvashatók ki, tengelyenként bemeneti mértékegységben.

A maradékút értéke: #5181...#5200, #100801...#100850, #_DIST[n] (R)

A változókról kiolvashatók tengelyenként a maradékút értékei, vagyis, hogy a végrehajtás alatt álló mondatban mennyi a hátralévő út a végpozícióig, bemeneti mértékegységben. Az az érték, amit a vezérlő a pozíció képernyőn a maradék útra kijelez.

23.1.18 Szerszámkorrekciós tár értékei

Maró korrekciós tár esetén

Száma	Szimbóluma	Tulajd.	Leírása
#10001...#10999	#_OFSHG[n] n=1...999	R/W	A hosszkorrekció geometriai értéke. n=1...999: a korrekciós rekesz száma
#11001...#11999	#_OFSHW[n] n=1...999	R/W	A hosszkorrekció kopás értéke. n=1...999: a korrekciós rekesz száma
#12001...#12999	#_OFSDG[n] n=1...999	R/W	A sugárkorrekció geometriai értéke. n=1...999: a korrekciós rekesz száma
#13001...#13999	#_OFSDW[n] n=1...999	R/W	A sugárkorrekció kopás értéke. n=1...999: a korrekciós rekesz száma

A fenti makrováltozókon keresztül az összes korrekciós rekesznek értéket adhatunk, illetve az összes korrekciós rekeszt kiolvashatjuk programból.

Marószerszám lekerekítési sugara

Száma	Szimbóluma	Tulajd.	Leírása
#21001...#21999	#_CORR_G[n] n=1...999	R/W	A lekerekítési sugár geometriai értéke. n=1...999: a korrekciós rekesz száma
#22001...#22999	#_CORR_W[n] n=1...999	R/W	A lekerekítési sugár kopás értéke. n=1...999: a korrekciós rekesz száma

Eszterga korrekciós tár esetén

Száma	Szimbóluma	Tulajd.	Leírása
#10001...#10999	#_OFSXW[n] n=1...999	R/W	Az X irányú hosszkorrekció kopás értéke. n=1...999: a korrekciós rekesz száma
#15001...#15999	#_OFSXG[n] n=1...999	R/W	Az X irányú hosszkorrekció geometriai értéke. n=1...999: a korrekciós rekesz száma
#14001...#14999	#_OFSYW[n] n=1...999	R/W	Az Y irányú hosszkorrekció kopás értéke. n=1...999: a korrekciós rekesz száma
#19001...#19999	#_OFSYG[n] n=1...999	R/W	Az Y irányú hosszkorrekció geometriai értéke. n=1...999: a korrekciós rekesz száma
#11001...#11999	#_OFSZW[n] n=1...999	R/W	Az Z irányú hosszkorrekció kopás értéke. n=1...999: a korrekciós rekesz száma
#16001...#16999	#_OFSZG[n] n=1...999	R/W	Az Z irányú hosszkorrekció geometriai értéke. n=1...999: a korrekciós rekesz száma
#12001...#12999	#_OFSRW[n] n=1...999	R/W	A sugárkorrekció kopás értéke. n=1...999: a korrekciós rekesz száma
#17001...#17999	#_OFSRG[n] n=1...999	R/W	A sugárkorrekció geometriai értéke. n=1...999: a korrekciós rekesz száma
#13001...#13999	#_OFST[n] n=1...999	R/W	A szerszámállás kódja. n=1...999: a korrekciós rekesz száma

A fenti makrováltozókon keresztül az összes korrekciós rekesznek értéket adhatunk, illetve az összes korrekciós rekeszt kiolvashatjuk programból.

23.1.19 Munkadarab nullponteltolásokA nullponteltolások értékei

Száma	Szimbóluma	Tulajd.	Leírása
#5201...#5220	#_WZCMN[n] n=1-50	R/W	Közös munkadarab nullponteltolás tengelyenként
#100301...#100350			
#5221...#5240	#_WZG54[n] n=1-50	R/W	G54 munkadarab nullponteltolás tengelyenként
#100351...#100400			
#5241...#5260	#_WZG55[n] n=1-50	R/W	G55 munkadarab nullponteltolás az 1...n tengelyen
#100401...#100450			
#5261...#5280	#_WZG56[n] n=1-50	R/W	G56 munkadarab nullponteltolás az 1...n tengelyen
#100451...#100500			
#5281...#5300	#_WZG57[n] n=1-50	R/W	G57 munkadarab nullponteltolás az 1...n tengelyen
#100501...#100550			
#5301...#5320	#_WZG58[n] n=1-50	R/W	G58 munkadarab nullponteltolás az 1...n tengelyen
#100551...#100600			
#5321...#5340	#_WZG59[n] n=1-50	R/W	G59 munkadarab nullponteltolás az 1...n tengelyen
#100601...#100650			

A fenti makrováltozókon keresztül az összes nullponteltolásnak tengelyenként értéket adhatunk, illetve az összes nullponteltolást tengelyenként kiolvashatjuk programból.

Az alapelforgatások értékei

Száma	Szimbóluma	Tulajd.	Leírása
#120001	#_WRG54[1]	R/W	G54 alapelforgatás az XY síkban
#120002	#_WRG54[2]	R/W	G54 alapelforgatás az ZX síkban
#120003	#_WRG54[3]	R/W	G54 alapelforgatás az YZ síkban
#120011	#_WRG55[1]	R/W	G55 alapelforgatás az XY síkban
#120012	#_WRG55[2]	R/W	G55 alapelforgatás az ZX síkban
#120013	#_WRG55[3]	R/W	G55 alapelforgatás az YZ síkban
#120021	#_WRG56[1]	R/W	G56 alapelforgatás az XY síkban
#120022	#_WRG56[2]	R/W	G56 alapelforgatás az ZX síkban

Száma	Szimbóluma	Tulajd.	Leírása
#120023	#_WRG56[3]	R/W	G56 alapelforgatás az YZ síkban
#120031	#_WRG57[1]	R/W	G57 alapelforgatás az XY síkban
#120032	#_WRG57[2]	R/W	G57 alapelforgatás az ZX síkban
#120033	#_WRG57[3]	R/W	G57 alapelforgatás az YZ síkban
#120041	#_WRG58[1]	R/W	G58 alapelforgatás az XY síkban
#120042	#_WRG58[2]	R/W	G58 alapelforgatás az ZX síkban
#120043	#_WRG58[3]	R/W	G58 alapelforgatás az YZ síkban
#120051	#_WRG59[1]	R/W	G59 alapelforgatás az XY síkban
#120052	#_WRG59[2]	R/W	G59 alapelforgatás az ZX síkban
#120053	#_WRG59[3]	R/W	G59 alapelforgatás az YZ síkban

A fenti makrováltozókon keresztül munkadarabkoordinátarendszerenként tetszőleges fősíokban (de csak egyben) elforgatást adhatunk meg, ahol a forgatás középpontja a koordinátarendszer origója.

A kiiterjesztett nullponteltolások értékei

Száma	Szimbóluma	Tulajd.	Leírása
#7001...#7020	#_WZP1[n]	R/W	G54.1 P1 munkadarab nullponteltolás az 1...n tengelyen
#101001...#101050	n=1-50		
#7021...#7040	#_WZP2[n]	R/W	G54.1 P2 munkadarab nullponteltolás az 1...n tengelyen
#101051...#101100	n=1-50		
...
#7941...#7960	#_WZP48[n]	R/W	G54.1 P48 munkadarab nullponteltolás az 1...n tengelyen
#103351...#103400	n=1-50		
#103401...#103450	#_WZP49[n]	R/W	G54.1 P49 munkadarab nullponteltolás az 1...n tengelyen
	n=1-50		
#103451...#103500	#_WZP50[n]	R/W	G54.1 P50 munkadarab nullponteltolás az 1...n tengelyen
	n=1-50		
...

A fenti makrováltozókon keresztül az összes nullponteltolásnak tengelyenként értéket adhatunk, illetve az összes nullponteltolást tengelyenként kiolvashatjuk programból.

A kiterjesztett alapelforgatások értékei

Száma	Szimbóluma	Tulajd.	Leírása
#120061	#_WRP1[1]	R/W	G54.1 P1 alapelforgatás az XY síkban
#120062	#_WRP1[2]	R/W	G54.1 P1 alapelforgatás az ZX síkban
#120063	#_WRP1[3]	R/W	G54.1 P1 alapelforgatás az YZ síkban
#120071	#_WRP2[1]	R/W	G54.1 P2 alapelforgatás az XY síkban
#120072	#_WRP2[2]	R/W	G54.1 P2 alapelforgatás az ZX síkban
#120073	#_WRP2[3]	R/W	G54.1 P2 alapelforgatás az YZ síkban
...

A fenti makrováltozókon keresztül minden kiterjesztett munkadarabkoordinátarendszerben tetszőleges fősíkban (de csak egyben) elforgatást adhatunk meg, ahol a forgatás középpontja a koordinátarendszer origója.

A dinamikus nullponteltolások értékei

A dinamikus nullponteltolás táblázat értékei az változók segítségével írhatók és olvashatók:

Száma	Szimbóluma	Tulajd.	Leírása
#5521...#5540	#_FOFS1[n] n=1-50	R/W	G54.2 P1 dinamikus nullponteltolás tengelyenként
#117051...#117100			
#5541...#5560	#_FOFS2[n] n=1-50	R/W	G54.2 P2 dinamikus nullponteltolás tengelyenként
#117101...#117150			
...
#5661...#5680	#_FOFS8[n] n=1-50	R/W	G54.2 P8 dinamikus nullponteltolás tengelyenként
#117401...#117450			

Az aktuális dinamikus nullponteltolás száma

Program futása közben az érvényes dinamikus nullponteltolás száma az alábbi utasítás segítségével olvasható ki:

Száma	Szimbóluma	Tulajd.	Leírása
#5500	#_FOFSP	R	Az aktuális dinamikus nullponteltolás száma (G54.2 Pp P=1-8)
#117000			

Az aktuális dinamikus nullponteltolás értéke

Program futása közben az aktuális dinamikus nullponteltolás értéke az alábbi változók segítségével olvashatók ki. Ezek az értékek nem azonosak a dinamikus nullponttáblázatba írt értékekkel, hanem az asztal(ok) pillanatnyi pozíciója alapján módosított (forgatott) értékek.

Száma	Szimbóluma	Tulajd.	Leírása
#5501...#5520	#_FOFSVAL[n]	R	Az aktuális dinamikus nullponteltolás értéke tengelyenként
#117001...#117050	n=1-50		

23.1.20 Az orsó és készletlét magazinokban lévő szerszám adatainak kiolvasása

Száma	Szimbóluma	Tulajd.	Leírása
#8400		R/W	Orsó, vagy készletlét magazin címe

A

#8400 (10, 11, 20, 21, ... írható, olvasható)

makrováltozón kell megadni, hogy melyik orsó, vagy készletlét magazinban lévő szerszám szerszámkezelői adatait kívánjuk kiolvasni a #8401, #8402, ... makrováltozókon keresztül. Csak a zárójelben megadott magazinszámok definiálhatók. Ha több orsó, vagy készletlét magazin van a gépen, információért forduljon a gép építőjéhez.

Ha a gépen csak egy orsómagazin van és készletlét magazin nincs, a #8400 makrováltozónak nem kell értéket adni, a #8401, ... makrováltozók mindig az orsóban lévő szerszámmra vonatkoznak.

A kiolvasható adatok a következők:

Száma	Szimbóluma	Tulajd.	Leírása
#8401		R	Adatszám (a szerszámkezelő táblázat sorszáma)
#8402		R	A szerszám típuszáma (T kód)
#8403		R	Az éltartam számláló értéke
#8404		R	A szerszám max. éltartama
#8405		R	A figyelmeztető éltartam értéke
#8406		R	Az éltartam státusza
#8407		R	Felhasználói bites adat
#8408		R	Szerszám info
#8409		R	H: a hosszkorrekciós rekesz száma (maró csoportban)

23.1 A programnyelv változói

Száma	Szimbóluma	Tulajd.	Leírása
#8410		R	D: a sugárkorrekciós rekesz száma (maró csatornában)
#8411		R	S: orsó fordulatszám
#8412		R	F: előtolás
#8413		R	G: a geometriai korrekciós rekesz száma (eszterga csatornában)
#8414		R	W: a kopáskorrekciós rekesz száma (eszterga csatornában)
#8431		R	Felhasználói adat 1
#8432		R	Felhasználói adat 2
#8433		R	Felhasználói adat 3
#8434		R	Felhasználói adat 4
#8435		R	Felhasználói adat 5
#8436		R	Felhasználói adat 6
#8437		R	Felhasználói adat 7
#8438		R	Felhasználói adat 8
#8439		R	Felhasználói adat 9
#8440		R	Felhasználói adat 10
#8441		R	Felhasználói adat 11
#8442		R	Felhasználói adat 12
#8443		R	Felhasználói adat 13
#8444		R	Felhasználói adat 14
#8445		R	Felhasználói adat 15
#8446		R	Felhasználói adat 16
#8447		R	Felhasználói adat 17
#8448		R	Felhasználói adat 18
#8449		R	Felhasználói adat 19
#8450		R	Felhasználói adat 9

A felhasználói adatok számát az N2903 No. of Custom Columns paraméteren lehet megadni.

Az első felhasználói adat mindig bites adat.

A fenti makrováltozók csak olvashatóak.

Ha a #8400 makrováltozón kiválasztott orsó, vagy készletléti magazin üres (nincs benne szer-
szám), a makrováltozók értékei:

#8401=0 (adatszám)

#8402= #8403= ...= #8450= #0 (üres)

A makrováltozók segítségével le lehet hívni a szerszámhoz rendelt korrekciókat (H, D), vagy
technológiai paramétereket (F, S). Ha pl. a szerszámcsere kódjára (M06) alprogram hívást je-
lölünk ki az alprogramba beírhatjuk:

...

M6

#8400=10 (1. orsómagazin)

H#8409 D#8410 S#8411 F#8412

...

23.1.21 A munkahelyen és a szerelőhelyen lévő paletta adatainak kiolvasása

Száma	Szimbóluma	Tulajd.	Leírása
#8500		R/W	Munkahely, vagy szerelőhely címe

A

#8500 (10, 11 írható, olvasható)

makrováltozón kell megadni, hogy munkahelyen (#8500=10), vagy a szerelőhelyen
(#8500=11) lévő paletta adatait kívánjuk kiolvasni a #8501, #8502, ... makrováltozókon
keresztül. Csak a zárójelben megadott magazinszámok definiálhatók.

Ha a gépen csak munkahely van és szerelőhely nincs, a #8500 makrováltozónak nem kell
értéket adni, a #8501, ... makrováltozók mindig a munkahelyen lévő palettára vonatkoznak.
Bekapcsolás után #8500=10 kezdeti értékkel áll fel.

A kiolvasható adatok a következők:

Száma	Szimbóluma	Tulajd.	Leírása
#8501		R	Paletta azonosító
#8502		R	Státusz
#8503		R	Prioritás
#8504		R	Végrehajtott programok száma
#8505		R	Kijelölt programok száma
#8531		R	Felhasználói 1
#8532		R	Felhasználói 2
#8533		R	Felhasználói 3
#8534		R	Felhasználói 4

23.2 A programnyelv utasításai

A különböző utasítások leírásánál a

$$\#i = \langle \text{formula} \rangle$$

kifejezést használjuk. A $\langle \text{formula} \rangle$ tartalmazhat aritmetikai műveleteket, függvényeket, változókat, konstansokat.

Általánosságban a $\langle \text{formula} \rangle$ -ban a $\#j$ és a $\#k$ változókra hivatkozunk.

A $\langle \text{formula} \rangle$ nem csak az értékadó utasítás jobb oldalán állhat, hanem az NC mondatban a különböző címek is felvehetnek konkrét számérték vagy változó helyett formulát is.

A műveletvégzés sorrendjét szögletes zárójelekkel

$$[,]$$

lehet befolyásolni.

23.2.1 Az értékadó utasítás: $\#i = \#j$

Az utasítás kódja: =

Az utasítás hatására a $\#i$ változó a $\#j$ változó értékét veszi fel, vagyis $\#i$ változóba bekerül $\#j$ változó értéke.

Az

$$\#i = \#i \langle \text{művelet} \rangle \#j$$

értékadás is megengedett. Az $\#i$ változó értékén végzünk egy műveletet $\#j$ változóval, aminek az eredménye az $\#i$ változóba kerül.

Például:

$$\#100 = \#100 + 1$$

azt jelenti, hogy $\#100$ új értéke a régi értéke plusz 1 lesz.

Bites változóknak is adhatunk értéket:

$$\#1100 = 1$$

$$\#_UO[3] = 0$$

$$\#_M_SBK = 1$$

Ha a bites változónak adott érték

$$0 < \text{érték} \leq 1$$

az értéket 1-nek veszi. Ha az érték nagyobb, mint 1

$$1 < \text{érték}$$

“2092 Értékhatár túllépve” hibát üzen.

☞ **Figyelem: Értékadó utasítás bal oldalán csak írható, vagy olvasható/írható (W, vagy W/R) változó állhat.** Ha csak olvasható (R) változónak akarunk értéket adni “2161 #nnnn makróváltozó csak olvasható” hibát üzen.

23.2.2 Aritmetikai műveletek

Egy operandusú mínusz: $\#i = - \#j$

A művelet kódja: -

A művelet hatására a $\#i$ változó a $\#j$ változóval abszolút értékben megegyező nagyságú, de ellentétes előjelű lesz.

$$\#100 = - \#12$$

Összeadás: $\#i = \#j + \#k$

A művelet kódja: +

A művelet hatására a #i változó a #j és #k változók értékének összegét veszi fel.

```
#1=#2+3.25
GO X[#100+#101]
```

Kivonás: $\#i = \#j - \#k$

Az művelet kódja: -

A művelet hatására a #i változó a #j és #k változók értékének különbségét veszi fel.

```
#100=#100-#102
G1 Z[25.34-2.48]
```

Szorzás: $\#i = \#j * \#k$

A művelet kódja: *

A művelet hatására a #i változó a #j és #k változók értékének szorzatát veszi fel.

```
#3=#1*#2
#100=#101*5.65
```

Osztás: $\#i = \#j / \#k$

A művelet kódja: /

A művelet hatására a #i változó a #j és #k változók értékének hányadosát veszi fel. #k értéke nem lehet 0. Ellenkező esetben "2093 Nullával osztási kísérlet" hibajelzést ad a vezérlő.

```
#3=#1/#2
#100=542.23/#3
```

Modulusképzés: $\#i = \#j \text{ MOD } \#k$

A művelet kódja: MOD

A művelet hatására a #i változó a #j és #k változók osztási maradékát veszi fel. #k értéke nem lehet 0. Ellenkező esetben "2093 Nullával osztási kísérlet" hibajelzést ad a vezérlő.

```
#120=27MOD4
```

esetén a #120 változó értéke 3 lesz.

Példák aritmetikai műveletek alkalmazására:

Szögletes zárójelek [,] használatával a műveletvégzés sorrendjét befolyásolhatjuk.

```
#100 = #101 + #102
#100 = #100 - 3
#100 = [#101 + #102*5.27]/4.1
GO X[[#100 + 2]/4] Y100
```

23.2.3 Logikai műveletek

A *logikai műveletek a változókat mindig 32 bites, előjeles, egész számként* kezelik. Ha eredendően lebegőpontos változón, pl #100, végzünk logikai műveletet, a művelet elvégzése előtt megvizsgálja, hogy a #i változó értéke teljesíti-e az alábbi feltételt:

$$-(2^{32} - 1) \leq \#i < 2^{32},$$

azaz decimálisan

$$-4294967297 \leq \#i < 4294967296.$$

Ha nem, "2129 Hibás művelet #-val" hibajelzést ad.

Ha a változó értéke a határon belülre esik a 32 bites egész számon végzi el a műveletet. Ha a művelet eredménye eredendően lebegőpontos változóba kerül, pl. #100, az eredményt visszaalakítja lebegőpontos számmá.

Negáció: $\#i = \text{NOT } \#j$

A művelet kódja: **NOT**

A művelet hatására a $\#j$ változót először átalakítja 32 bites fixpontos számmá. Ezután ennek a fixpontos számnak a bitenkénti negált értékét veszi mind a 32 bitre.

Vagy: $\#i = \#j \text{ OR } \#k$

A művelet kódja: **OR**

A művelet hatására a $\#j$ és $\#k$ változót először átalakítja 32 bites fixpontos számmá.

A művelet hatására a $\#i$ változóba a $\#j$ és a $\#k$ változók bitenkénti értékeinek logikai összege kerül, mind a 32 bitre. Ahol a két szám azonos helyiértékén mindkét helyen 0 található, arra a helyiértékre az eredményben 0 kerül, egyébként 1.

Kizáró vagy: $\#i = \#j \text{ XOR } \#k$

A művelet kódja: **XOR**

A művelet hatására a $\#j$ és $\#k$ változót először átalakítja 32 bites fixpontos számmá.

A művelet hatására a $\#i$ változóba a $\#j$ és a $\#k$ változók bitenkénti értékei úgy összegződnek, hogy, ahol azonos helyiértéken azonos számértékek találhatók az eredményben arra a helyiértékre 0, ahol különböző számértékek találhatók oda 1 kerül, mind a 32 biten.

És: $\#i = \#j \text{ AND } \#k$

A művelet kódja: **AND**

A művelet hatására a $\#j$ és $\#k$ változót először átalakítja 32 bites fixpontos számmá.

A művelet hatására a $\#i$ változóba a $\#j$ és a $\#k$ változók bitenkénti értékeinek logikai szorzata kerül, mind a 32 bitre. Ahol a két szám azonos helyiértékén mindkét helyen 1 található, arra a helyiértékre az eredményben 1 kerül, egyébként 0.

Példa logikai műveletek alkalmazására:

A #1132, PLC felé menő, 32 bites makróváltozót manipuláljuk a következőképp: a változó felső 8, (31 - 24) bitjét kell állítani úgy, hogy az alsó bitek változatlanok maradjanak. Az alsó 24 bit változatlan hagyásához és a felső 8 bit törléséhez vegyük a következő bitmaszkt:

bitmaszk hexadecimálisan

00FFFFFF,

decimálissá alakítva:

16777215

Tároljuk el bitmaszkt a #100 változón:

$\#100 = 16777215$

A #101 változóra eltároljuk, a #1132 alsó 24 bitjét változatlanul hagyó, a felső 8 bitjén törölt értéket:

$\#101 = \#100 \text{ AND } \#1132$

A #102 változóra írjuk rá a beállítandó bitmintát:

bitminta hexadecimálisan:

9B000000,

decimálissá alakítva:

-1694498816

Tároljuk el a bitmintát a #102 változón:

$\#102 = -1694498816$

Majd írjuk ki a #1132 változóra úgy, hogy az alsó 24 bit érintetlen maradjon:

$\#1132 = \#101 \text{ XOR } \#102$

Zárójelek megfelelő használatával a fenti négy sor helyett írhatjuk a következőt is:


```
#1132=[16777215AND#1132]XOR[-1694498816]
```

23.2.4 Függvények

Négyzetgyökvonás: #i = SQRT #j

A függvény kódja: **SQRT**

A művelet hatására a #i változó a #j változó négyzetgyökét veszi fel.

```
#101=SQRT[#100+4]
```

A #j változó értéke csak pozitív szám, vagy 0 lehet:

$$\#j \geq 0$$

Ha az argumentum negatív, "2122 Négyzetgyök negatív értékből" hibát üzen.

Színusz: #i = SIN #j

A függvény kódja: **SIN**

A művelet hatására #i változó #j változó szinuszát veszi fel. *#j értéke mindig fokban értendő.*

```
G1 X[SIN30-#101]
```

Koszínusz: #i = COS #j

A függvény kódja: **COS**

A művelet hatására #i változó #j változó koszínuszát veszi fel. *#j értéke mindig fokban értendő.*

```
#102=COS[#1+#2]
```

Tangens: #i = TAN #j

A függvény kódja: **TAN**

A művelet hatására #i változó #j változó tangensét veszi fel. *#j értéke mindig fokban értendő.*

Ha a #j argumentum értéke

$$\#j=(2n+1)*90^\circ, \text{ ahol } n=0, \pm 1, \pm 2, \dots$$

"2116 Argumentum abszolút értéke nem kisebb mint 90" hibát üzen.

```
#1=TAN[#2*15.6]
```

Arkuszszínusz: #i = ASIN #j

A függvény kódja: **ASIN**

A művelet hatására #i változó #j változó arkuszszinuszát veszi fel. *Az eredményt fokban adja meg, #i értéke +90° és -90° közé esik.*

```
#101=ASIN[-0.5] (#101=-30 lesz)
```

#j argumentumnak az alábbi feltételt kell teljesítenie:

$$-1 \leq \#j \leq 1$$

Ellenkező esetben a "2117 Argumentum abszolút értéke nagyobb, mint 1" hibát üzeni.

Arkuszkoszínusz: #i = ACOS #j

A függvény kódja: **ACOS**

A művelet hatására #i változó #j változó arkuszkoszínuszát veszi fel. *Az eredményt fokban adja meg, #i értéke 0° és 180° közé esik.*

```
#101=ACOS[-0.5] (#101=120 lesz)
```

#j argumentumnak az alábbi feltételt kell teljesítenie:

$$-1 \leq \#j \leq 1$$

Ellenkező esetben a "2117 Argumentum abszolút értéke nagyobb, mint 1" hibát üzeni.

Arkusztangens: #i = ATAN #j

A függvény kódja: **ATAN**

A művelet hatására #i változó #j változó arkusztangensét veszi fel. *Az eredményt fokban adja meg, #i értéke +90° és -90° közé esik.*

#101=ATAN[-0.5] (#101=-26.565 lesz)

Exponenciális: #i = EXP #j

A függvény kódja: **EXP**

A művelet hatására a #i változó a természetes szám (e) #j-edik hatványát veszi fel.

#100=EXP1 (#100=2.71828... azaz "e" lesz)

Természetes alapú logaritmus: #i = LN #j

A függvény kódja: **LN**

A művelet hatására #i változó a #j szám természetes alapú logaritmusát veszi fel.

#100=LN2.718281828 (#100=0.999... ln(e)=1 lesz)

Ha a #j argumentum értéke

#j ≤ 0

“2118 Argumentum értéke nem pozitív” hibát üzen.

23.2.5 Konverziós utasítások

Abszolút érték képzés: #i = ABS #j

A függvény kódja: **ABS**

A művelet hatására a #i változó a #j változó abszolút értékét veszi fel.

#100=ABS[-3.1] (#100=3.1 lesz)

#100=ABS[5.25] (#100=5.25 lesz)

#100=ABS[0] (#100=0 lesz)

Kerekítés abszolút értékben lefelé: #i = FIX #j

A függvény kódja: **FIX**

A művelet a #j változó törtrészét eldobja és ez az érték kerül #i változóba.

Például:

#130=FIX4.8 (#130=4 lesz)

#131=FIX[-6.7] (#131=-6 lesz)

Kerekítés abszolút értékben felfelé: #i = FUP #j

A függvény kódja: **FUP**

A művelet a #j változó törtrészét eldobja és abszolút értékben 1-et ad hozzá.

Például:

#130=FUP12.1 (#130=13 lesz)

#131=FUP[-7.3] (#131=-8 lesz)

23.2.6 Összetett aritmetikai műveletek végrehajtási sorrendje

A fentebb felsorolt aritmetikai műveleteket és függvényeket kombinálni lehet. A *műveletek végrehajtásának sorrendje*, vagy precedencia szabálya:

függvény – multiplikatív aritmetikai műveletek – additív aritmetikai műveletek.

Például az alábbi példában a műveletek végrehajtási sorrendje:

$$\begin{array}{c} \#110 = \#111 + \#112 * \text{COS } \#113 \\ \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \text{—————1—————} \\ \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \text{—————2—————} \\ \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \text{—————3—————} \end{array}$$

A műveletek végrehajtási sorrendjének módosítása

Szögletes [és] zárójelek használatával a műveletek végrehajtási sorrendje módosítható. Példa háromszoros mélységű zárójelezésre:

$$\#120 = \text{COS} \left[\left[\left[\#121 - \#122 \right] * \#123 + \#125 \right] * \#126 \right]$$

1
2
3
4
5

A számok a műveletek végrehajtási sorrendjét mutatják. Látható, hogy az azonos szintű zárójeleken belül a műveletek végrehajtási sorrendjére a fent említett precedenciaszabály érvényes.

A nyitó [és a záró] zárójeleket párban kell megadni. Ha a nyitó [zárójelek száma kevesebb, mint a záróké, “2064 Szintaktikai hiba” üzenetet ad. Ha a záró] zárójelek száma kevesebb, mint a nyitóké, “2121 Jobb zárójel nem található üzenetet ad.

23.2.7 Feltételes kifejezések

A programnyelv a következő feltételes kifejezéseket ismeri:

egyenlő (=):	#i EQ #j
nem egyenlő (≠):	#i NE #j
nagyobb, mint (>):	#i GT #j
kisebb, mint (<):	#i LT #j
nagyobb, vagy egyenlő (≥):	#i GE #j
kisebb, vagy egyenlő (≤):	#i LE #j

A feltételes kifejezés mindkét oldalán a változó helyettesíthető formulával is. A fenti feltételes kifejezések IF, vagy WHILE utasítások után állhatnak.

23.2.8 Feltétel nélküli elágazás: GOTO

A **GOTO** utasítás hatására a program végrehajtása feltétel nélkül ugyanannak a programnak az n számú mondatán folytatódik. n szám helyettesíthető változóval, vagy formulával is. **A mondat számnak**, amelyekre a GOTO utasítással ugrunk **a mondat elején kell szerepelnie**. Ha a kijelölt mondat számot nem találja “2125 Mondat nem található Nnnn” hibajelzést ad. Ugorhatunk az utasítással előre:

```
...
GOTO160 (az N160 mondatra ugrik)
```

```
...
N160 GO X100
```

vagy hátra is. Ebben az esetben végtelen ciklus is létrejöhet:

```
...
N160 GO X100
```

```

...
#1=100
#2=60
...
GOTO[#1+#2] (az N160 mondatra ugrik)
...

```

23.2.9 Feltételes elágazás: IF[<feltételes kifejezés>] GOTO n

Ha a kötelezően szögletes zárójelek közé tett [<feltételes kifejezés>] teljesül, a program végrehajtása ugyanannak a programnak az n számú mondatán folytatódik.

Ha a [<feltételes kifejezés>] nem teljesül, a program végrehajtása a következő mondaton folytatódik.

Ha az IF után nem feltételvizsgálat következik, “2064 Szintaktikai hiba”, vagy “2121 Hibás terminátor =” üzenetet ad, a hiba típusától függően.

Például

```

...
#100=52.28
#101=16.87
...
IF[#100GT#101] GOTO210 (Az N210 mondatra ugrik)
...
N210 G0 X0 Y100 Z20
...

```

23.2.10 Feltételes utasítás: IF[<feltételes kifejezés>] (THEN)<utasítás>

Ha a [<feltételes kifejezés>] teljesül, a THEN mögötti utasítás végrehajtásra kerül.

Ha a [<feltételes kifejezés>] nem teljesül, a program végrehajtása a következő mondaton folytatódik.

Az utasításban a **THEN elhagyható** az

IF[<feltételes kifejezés>] utasítás

utasítássor végrehajtása ugyanaz.

```

...
#100=0
#101=0
#1=20
...
IF[#1EQ20] THEN#100=3.15 (#100=3.15 lesz)
IF[#100GT3.15] #101=5 (#101=0 marad)
...

```

23.2.11 Ciklusszervezés: WHILE[<feltételes kifejezés>] DO m ... END m

Amíg a [<feltételes kifejezés>] teljesül, a DO m utáni mondatok az END m mondatig ismételten végrehajtásra kerülnek. Vagyis a vezérő megvizsgálja, hogy a feltétel teljesült-e, ha igen végrehajtja a DO m és END m közötti programrészt, majd END m utasítás hatására a program visszatér a WHILE utáni feltétel újbóli vizsgálatára.

Ha a [<feltételes kifejezés>] nem teljesül a program végrehajtása az END m utáni mondaton folytatódik.

Ha a WHILE [<feltételes kifejezés>] elhagyásra kerül, azaz a ciklust a DO m ... END m uta-

sítások írják le a *DOm és ENDM közötti programrészt végtelen ideig hajtja végre.*

m lehetséges értékei: 1, 2, 3.

m>3 eset “2002 DO adat értékhatáron kívül” hibajelzést eredményez.

Ha a WHILE után nem feltételvizsgálat következik hanem értékadó utasítás, “2121 Hibás terminátor: =” hibajelzést ad.

Ha a WHILE utasítást nem követi DO ugyanabban a mondatban, 2110 WHILE, DO nélkül” hibát üzen.

Ha ENDM utasítás előtt nincs DOm, “2125 Mondat nem található: DOm” üzenetet küld.

Bontsunk ki egy Z irányban 5.4 mm mély, X irányban 20 mm széles üreget, amelynek a közepén van X irányban a nullpont, Z irányban pedig az üreg tetején. A fogásmélység legyen 0.2 mm. A feladat megoldható ciklusszervezéssel:

```
#1=0           (Z irányú célpozíció)
#2=10         (X irányú célpozíció)
G0 X#2        (mozgás X-ben a kezdőpontra)
Z[#1+1]       (mozgás Z-ben a kezdőpontra)
WHILE[#1GT-5.39] DO1
(ciklus, amíg Z-ben nem érjük el az 5.4 mm mélységet)
#1=#1-0.2     (0.2 mm fogásvétel Z)
#2=-#2        (mozgásirány megfordítása X)
G1 Z#1 F100   (fogásvétel Z-ben)
X#2 F500      (marás X mentén)
END1
G0 Z20
...
```

A ciklusszervezés szabályai:

– DOm utasítást ENDM utasítás előtt kell megadni:

```
:
END1
:
:           (HIBÁS)
:
DO1
```

A fenti esetben az END1 mondaton “2125 Mondat nem található DOm” hibát üzen.

– A DOm és ENDm utasításoknak párban kell állniuk:

```
:  
DO1  
:  
DO1           HIBÁS  
:  
END1  
:
```

vagy

```
:  
DO1  
:  
END1          HIBÁS  
:  
END1  
:
```

– Ugyanazt az azonosító számot többször lehet használni:

```
:  
DO1  
:  
END1  
:  
:  
:  
:  
DO1  
:  
END1  
:
```

– DOm ... ENDm párok háromszoros mélységig skatulyázhatók egymásba:

```
:  
DO1  
:  
DO2  
:  
DO3  
:  
:  
:  
:  
END3  
:  
END2  
:  
END1  
:
```

– A DO m ... END m párok nem lapolhatják át egymást:

```
:  
DO1  
:  
DO2  
:  
:                               HIBÁS  
:  
END1  
:  
END2
```

– Ciklus belsejéből a cikluson kívülre el lehet ágni:

```
:  
DO1  
:  
GOTO150  
:  
:                               HELYES  
:  
END1  
:  
N150  
:  
:
```

– Ciklusba kívülről be lehet lépni:

```
:  
GOTO150  
:  
DO1  
:  
:  
:  
N150  
:  
END1  
:  
:
```

vagy

```
:  
DO1  
:  
N150  
:  
:  
:  
END1  
:  
GOTO150  
:  
:
```

- Ciklus belsejéből alprogram, vagy makrohívás lehetséges. Az alprogram, vagy a felhasználói makró belsejében a ciklusok ismét háromszoros mélységig skatulyázhatók egymásba:

```

:
DO1
:
M98 . . .           HELYES
:
G65 . . .           HELYES
:
G66 . . .           HELYES
:
G67 . . .           HELYES
:
END1
:

```

23.2.12 Indirekt tengelyhivatkozások

Az

#i=AXNUM[<tengelycím>]

utasítás segítségével egy tengely címe alapján le lehet kérdezni az adott csatornában a tengely számát. Ha nem létező tengelycímhez akarunk tengelyszámot lekérdezni, “2017 Tiltott cím <tengelynév>” hibát üzen.

Az

AX[<tengelyszám>]=

utasítással indirekt módon, nem a tengely címe, hanem a tengely száma alapján adhatunk ki parancsot. Ha a csatornában nem található a megadott számú tengely, “2018 Hibás fizikai tengelyszám: n” hibát üzen.

Például, a

```
G0 X10 Y20 Z30
```

mondat helyett írhatjuk az alábbi utasítást

```

#101=AXNUM[X]      (X tengely számának lekérdezése)
#102=AXNUM[Y]      (Y tengely számának lekérdezése)
#103=AXNUM[Z]      (Z tengely számának lekérdezése)
G0 AX[#101]=10 AX[#102]=20 AX[#103]=30

```

☞ **Figyelem!** Ha többkarakteres tengelycímeket használunk a tengely címe ne legyen AX, vagy AXN.

23.2.13 Adatkiadási parancsok

A vezérlő a következő adatkiadási parancsokat ismeri:

POPEN	periféria megnyitása
FOPEN	fájl megnyitása
BPRNT	bináris adatkiadás
DPRNT	decimális adatkiadás
PCLOS	periféria zárása
FCLOS	fájl zárása

Ezeket az adatkiadó parancsokat karaktereknek és változók értékeinek a kiadására lehet használni. A kiadás a vezérlés memóriájába történik. Felhasználható például mérési

eredmények eltárolására, naplózásra, stb.

Periféria megnyitása: POPENn

Mielőtt adatkiadó parancsot adunk, meg kell nyitni a megfelelő perifériát, amin keresztül az adatkiadás történni fog. A megfelelő periféria kiválasztása az n számmal történik:

n = 31 a vezérlő memóriája

A periféria megnyitásakor egy % karakter is kiküldésre kerül a perifériára, tehát minden adatkiadás egy % karakterrel kezdődik.

A POPEN utasítást követően DPRNT utasítással egy fájlnevet kell kiadni. Ha a fájl már létezik, kérdezés nélkül felülírja a régit, ha még nem létezik megnyit egy újat. A fájl abba a mappába kerül, ahol a program fut, és a program kiterjesztését veszi át.

Például:

```
...
#100=4567
POPEN31
DPRNT [O#100[4]]      (A fájlnev 04567 lesz)
...
```

vagy

```
...
POPEN31
DPRNT [ABC]           (A fájlnev ABC lesz)
...
```

Periféria lezárása: PCLOSn

A POPEN parancssal megnyitott perifériát a PCLOS parancssal le kell zárni. A PCLOS parancs után meg kell nevezni a lezárandó periféria számát. Esetünkben:

```
...
PCLOS31
...
```

A lezáráskor még egy % karakter is kiküldésre kerül a perifériára, azaz minden adatkiadást egy % karakter zár le.

Fájl megnyitása: FOPENn Ppppp, FOPENn <fájlnev>

Mielőtt adatkiadási parancsot adunk ki, meg kell nyitni egy fájlt, ahova az adatokat be akarjuk írni.

Fájl megnyitása programszám alapján

Az

FOPENn P(programszám)

P címen (programszám) megadott számú fájlt nyitja meg.

A P címen megadott *programszám programozási szabályait* a “[14.4.1 Programok azonosítása a tárban. A programszám \(O\)](#)” című fejezet tárgyalja a [108.](#) oldalon. A

programszámmal megnyitott fájlok kiterjesztésének és tárban való elhelyezkedésének szabályai megegyeznek az alprogramokéval. Lásd a “[14.4.2 Alprogram hívása \(M98\)](#)” című fejezetet a [109.](#) oldalon.

Például:

```
...
FOPEN1 P12           (A fájlnev 00012 lesz)
```

...

Fájl megnyitása fájlnev alapján

A

FOPENn <fájlnev>

<> jelek között megadott nevű fájl nyitja meg.

A *fájlnev megadási szabályait* a “**14.4.1 Programok azonosítása a tárban. A programszám (O)**” című fejezet tárgyalja a [108.](#) oldalon. A *fájlnevvel megnyitott fájlok kiterjesztésének és tárban való elhelyezkedésének szabályai* megegyeznek az alprogramokéval. Lásd a “**14.4.2 Alprogram hívása (M98)**” című fejezetet a [109.](#) oldalon.

Például:

```
...
FOPEN1 <adatok.txt> (A fájlnev adatok.txt lesz)
...
```

A fájlnyitás típusai

A fájlnyitás típusa az “n” helyére írt számmal adható meg:

n = 1: Új fájl létrehozása

Ha a fájl már létezik, akkor a rendszer “2144 Az (Ooooo) fájl már létezik”, vagy a “2146 A megadott fájl már létezik” hibajelzést küld, és a program futása nem folytatódik.

n = 2: Új fájl létrehozása mindenképp

Ha a fájl *nem létezik*, akkor *létrehozza*, *ha létezik*, akkor megnyitja, és *törli a tartalmát*. A BPRNT, vagy a DPRNT utasítással kiadott tartalmakat a fájl elejétől kezdve írja.

n = 3: Fájl megnyitása hozzáfűzésre

Ha a file nem létezik, akkor “2147 Az (Ooooo) fájl nem található”, vagy a “2132 Fájl nem található” hibajelzést küld, és a program futása nem folytatódik. Ha a fájl már létezik, akkor megnyitja, és a *fájl végétől kezdve hozzáfűzi* a BPRNT, vagy a DPRNT utasítással kiadott tartalmakat.

n = 4: Fájl létrehozása, vagy megnyitása hozzáfűzésre

Ha a fájl még *nem létezik*, akkor *létrehozza*, *ha már létezik*, akkor *megnyitja*, és a *fájl végétől kezdve hozzáfűzi* a BPRNT, vagy a DPRNT utasítással kiadott tartalmakat.

n = 5: Fájl megnyitása törléssel

Ha a fájl még nem létezik, akkor “2147 Az (Ooooo) fájl nem található”, vagy a “2132 Fájl nem található” hibajelzést küld és a program futása nem folytatódik. Ha a fájl már létezik, akkor *megnyitja*, és *törli a tartalmát*. A BPRNT, vagy a DPRNT utasítással kiadott tartalmakat a fájl elejétől kezdve írja.

Fájl lezárása: FCLOS

A megnyitott fájl a FCLOS utasítással zárható be. Egyidejűleg egyetlen fájl lehet írásra létrehozni, vagy megnyitni.

Bináris adatkiadás: BPRNT[...]

A BPRNT utasítás formátuma:

```
BPRNT [ a #b [c] ... ]
```

a tizedespont alatti számjegyek száma
változó
karakter

A karaktereket ASCII kódban, a változókat pedig binárisan küldi ki a parancs.

– A kiküldhető karakterek:

alfabetikus karakterek: A, B, ..., Z

numerikus karakterek: 1, 2, ..., 0

speciális karakterek: *, /, +, -

A * karakter helyett a szóköz (space) kódját küldi ki a vezérlő.

– A változók értékeit 4 byte-on, azaz 32 biten adja ki a vezérlő, a legnagyobb helyiértékű byte-tól kezdve. A változók száma után zárójelben [] a tizedespont utáni számjegyek számát kell megadni. Ekkor a vezérlő a változó lebegőpontos értékét olyan fixpontos értékévé alakítja, amelyben az értékes tizedesjegyek száma a [] zárójelben megadott érték. c lehetséges értékei: 1, 2, ..., 8. Például, ha

#120 = 258.647673 és [3] ——— kiadásra kerül 258648=0003F258h

– Az üres változót 00000000h bináris kóddal adja ki.

– Az adatkiadás végén a vezérlő automatikusan egy kocsi vissza (CR) és egy soremelés (LF) karaktert ad ki.

Példa:

#110=318.49362

#120=0.723415

#112=23.9

BPRNT [C*/X#110 [3]Y#120 [3]M#112 [0]]

A #110, #120 és a #112 változók értékei a kerekítés és a hexadecimális átalakítás után a következők lesznek:

#110=318.49362 ——— 318494=0004DC1Eh

#120=0.723415 ——— 723=000002D3h

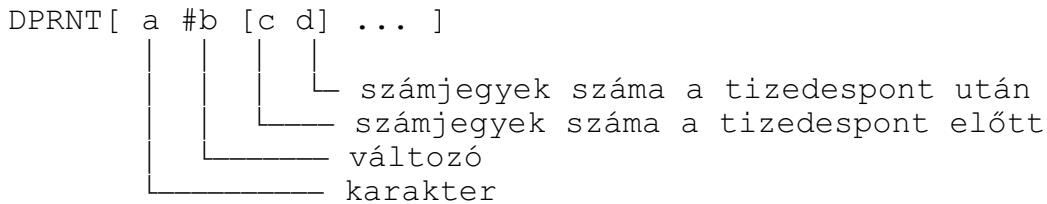
#112=23.9 ——— 24=00000018h

A kiadásra kerülő karakterek:

7	6	5	4	3	2	1	0	
0	1	0	0	0	0	1	1	C
0	0	1	0	0	0	0	0	Szóköz (Space)
0	0	1	0	1	1	1	1	/
0	1	0	1	1	0	0	0	X
0	0	0	0	0	0	0	0	00
0	0	0	0	0	1	0	0	04
1	1	0	1	1	1	0	0	DC
0	0	0	1	1	1	1	0	1E
0	1	0	1	1	0	0	1	Y
0	0	0	0	0	0	0	0	00
0	0	0	0	0	0	0	0	00
0	0	0	0	0	0	1	0	02
1	1	0	1	0	0	1	1	D3
0	1	0	0	1	1	0	1	M
0	0	0	0	0	0	0	0	00
0	0	0	0	0	0	0	0	00
0	0	0	0	0	0	0	0	00
0	0	0	1	1	0	0	0	18
0	0	0	0	1	1	0	1	Kocsi vissza (Carriage Return)
0	0	0	0	1	0	1	0	Soremelés (Line Feed)

Decimális adatkiadás: DPRNT[...]

A DPRNT utasítás formátuma:



Az összes karakter és számjegy ASCII kódban kerül kiadásra.

- A karakterek kiadására vonatkozó szabályokat lásd **BPRNT** utasítás.
- Változók értékeinek kiadásához meg kell adni, hogy a változó hány decimális egész és tört jegyen kerüljön kiadásra. A számjegyek megadását [] zárójelek közé kell tenni. A számjegyek megadására a $0 < c + d < 16$ feltételnek teljesülni kell. A számok kiadása a legmagasabb helyiértéküktől kezdődik.
- A számjegyek kiadásánál a negatív előjel (-) kiadásra kerül.
- Ha a tizedespont definiálva van ($d > 0$), minden nulla, a követő nullák is, kiadásra kerülnek a tizedesponttal (.) együtt.
- Ha $d = 0$, vagy d nincs megadva, sem tizedespontot, sem nullát nem ad ki.
- Ha az N1757 Print Contr paraméter #0 PNT bitje
 - =0 a + előjel és a vezető nullák helyén szóköz (space) kód kerül kiadásra,
 - =1 a + előjel és a vezető nullák nem kerülnek kiadásra.
- Az üres változót 0 kóddal adja ki.

- Az adatkiadás végén a vezérlő automatikusan egy kocsi vissza (CR) és egy soremelés (LF) karaktert ad ki.

Példa:

```
#130=35.897421
#500=-150.8
#10=14.8
DPRNT [X#130 [53] Y#500 [53] T#10 [20]
```

A #130, #500 és a #10 változók értékei a kerekítés után a következők lesznek:

```
#130=35.897421    _____    35.897
#500=-150.8       _____    150.8
#10=14.8          _____    15
```

Adatkiadás N1757 Print Contr paraméter #0 PNT=0 állásánál:

7	6	5	4	3	2	1	0	
0	1	0	1	1	0	0	0	X
0	0	1	0	0	0	0	0	Szóköz (space)
0	0	1	0	0	0	0	0	Szóköz (space)
0	0	1	0	0	0	0	0	Szóköz (space)
0	0	1	0	0	0	0	0	Szóköz (space)
0	0	1	1	0	0	1	1	3
0	0	1	1	0	1	0	1	5
0	0	1	0	1	1	1	0	Tizedespont (.)
0	0	1	1	1	0	0	0	8
0	0	1	1	1	0	0	1	9
0	0	1	1	0	1	1	1	7
0	1	0	1	1	0	0	1	Y
0	0	1	0	1	1	0	1	Negatív előjel (-)
0	0	1	0	0	0	0	0	Szóköz (space)
0	0	1	0	0	0	0	0	Szóköz (space)
0	0	1	1	0	0	0	1	1
0	0	1	1	0	1	0	1	5
0	0	1	1	0	0	0	0	0
0	0	1	0	1	1	1	0	Tizedespont (.)
0	0	1	1	1	0	0	0	8
0	0	1	1	0	0	0	0	0
0	0	1	1	0	0	0	0	0
0	1	0	1	0	1	0	0	T
0	0	1	0	0	0	0	0	Szóköz (space)
0	0	1	1	0	0	0	1	1
0	0	1	1	0	1	0	1	5
0	0	0	0	1	1	0	1	Kocsi vissza (Carriage Return)
0	0	0	0	1	0	1	0	Soremelés (Line Feed)

Adatkiadás N1757 Print Contr paraméter #0 PNT=1 állásnál:

7	6	5	4	3	2	1	0	
0	1	0	1	1	0	0	0	X
0	0	1	1	0	0	1	1	3
0	0	1	1	0	1	0	1	5
0	0	1	0	1	1	1	0	Tizedes pont (.)
0	0	1	1	1	0	0	0	8
0	0	1	1	1	0	0	1	9
0	0	1	1	0	1	1	1	7
0	1	0	1	1	0	0	1	Y
0	0	1	0	1	1	0	1	Negatív előjel (-)
0	0	1	1	0	0	0	1	1
0	0	1	1	0	1	0	1	5
0	0	1	1	0	0	0	0	0
0	0	1	0	1	1	1	0	Tizedes pont (.)
0	0	1	1	1	0	0	0	8
0	0	1	1	0	0	0	0	0
0	0	1	1	0	0	0	0	0
0	1	0	1	0	1	0	0	T
0	0	1	1	0	0	0	1	1
0	0	1	1	0	1	0	1	5
0	0	0	0	1	1	0	1	Kocsi vissza (Carriage Return)
0	0	0	0	1	0	1	0	Soremelés (Line Feed)

☞ **Megjegyzések:**

- Az adatkiadási parancsok sorrendje kötött: először POPEN, vagy FOPEN paranccsal meg kell nyitni a megfelelő fájlt, utána jöhet az adatkiadás BPRNT, vagy DPRNT paranccsal, végül a megnyitott fájlt le kell zárni a PCLOS, vagy az FCLOS utasítással.
- A fájl megnyitása és lezárása bárhol a programban megadható. Például a program elején megnyitható, a program végén lezárható, és közben a program bármely, a két utasítás közé eső részén adat küldhető ki.
- Adatkiadás közben végrehajtott M30, vagy M2 parancs megszakítja az adatátvitelt. Ha ezt el akarjuk kerülni, az M30 parancs végrehajtása előtt várakozni kell adatátvitel közben.

23.3 Makrók, rendszermakrók, rendszerprogramok hívása

A *makróhívás hasonló az alprogramhíváshoz*, azzal a különbséggel, hogy a *makróknak*, szemben az alprogramokkal, *lehetnek bemenő változói*, azaz *argumentumai*.

Úgy a makrókat, mint az alprogramokat, hívhatjuk programszámuk, vagy fájlnevük alapján is.

Rendszermakróknak és rendszerprogramoknak nevezzük az olyan makrókat, illetve alprogramokat, amelyek hívását egy adott, paraméteren kijelölt címhez kötjük (pl: G, M stb.). A rendszermakrók és alprogramok fájlnevére illetve tárbeli helyzetükre speciális szabályok vonatkoznak.

A *makróhívás is lehet többszintű*. Alprogramból is hívhatunk makrókat, makróból is alprogramot. Az alprogramok és makrók hívásának együttes szintje *maximum 16* lehet.

A *makrókból való visszatérés*

M99

kódra történik.

A *makróhívások lehetnek egylovvetűek, vagy öröklődők*. Az öröklődő makróhívásokat a

G67

kóddal lehet törölni.

Argumentumok megadása

A makróprogramnak argumentumokat lehet átadni. *Az argumentumok meghatározott címeknek adott olyan konkrét számértékek, amelyek a makróhívás során a megfelelő lokális változóknak (#1, #2, ..., #33) kerünek eltárolásra*. A makróprogram (speciális alprogram) ezeket a lokális változókat fel tudja használni, vagyis a makróhívás olyan speciális alprogramhívás, ahol az alprogramnak a főprogram változókat (paramétereit) tud átadni.

Kétféle argumentumkijelölés lehetséges:

1. sz. argumentumkijelölés:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Az 1. sz. argumentumkijelölés a paramétereit az angol ABC betűin keresztül adja át a makróknak. A fenti karakterek közül egyes hívások lefoglalhatnak más célra betűket (**G, P, L**), ezeket ilyenkor nem lehet paraméterátadásra használni. A címeket tetszőleges sorrendben lehet kitölteni, nem szükséges, hogy ABC sorrendben írjuk be őket a hívó mondatba.

2. sz. argumentumkijelölés:

A B C I1 J1 K1 I2 J2 K2 ... I10 J10 K10

A 2. sz. argumentumkijelölés az A, B, C címeken kívül az I, J, K címeken keresztül ad át argumentumokat a makróknak. Az I, J, K címeken maximum 10 különböző argumentumcsoport jelölhető ki. A címeket tetszőleges sorrendben lehet kitölteni. Ha egy mondatban több argumentumot jelölünk ki ugyanarra a címre, a változók a kijelölés sorrendjében veszik fel a megfelelő értéket.

A lokális változók és az argumentumok közötti kapcsolatot:

lv	1. ak	2. ak
#1	A	A
#2	B	B
#3	C	C
#4	I	I1
#5	J	J1
#6	K	K1
#7	D	I2
#8	E	J2
#9	F	K2
#10	G	I3
#11	H	J3

lv	1. ak	2. ak
#12	L	K3
#13	M	I4
#14	N	J4
#15	O	K4
#16	P	I5
#17	Q	J5
#18	R	K5
#19	S	I6
#20	T	J6
#21	U	K6
#22	V	I7

lv	1. ak	2. ak
#23	W	J7
#24	X	K7
#25	Y	I8
#26	Z	J8
#27	–	K8
#28	–	I9
#29	–	J9
#30	–	K9
#31	–	I10
#32	–	J10
#33	–	K10

rövidítések: *lv*:lokális változó,
1.ak: 1. sz. argumentumkijelölés,
2.ak: 2. sz. argumentumkijelölés.

Az I, J, K címek utáni indexek az argumentumkijelölés sorrendjét mutatják.
Tizedespont, előjel, is átadható a címeken.

Az N cím kezelése:

A mondatba beírt **első N címet mondatszám**nak veszi, a **második N címet** viszont már **argumentumnak jegyzi be** a #14 lokális változón:

```
/N130 X12.3 Y32.6 N250
N130: mondatszám
#24=12.3
#25=32.6
#14=250 (argumentum)
```

Ha az N cím már egyszer argumentumként került bejegyzésre a következő N címre történő hivatkozás már “2017 Tiltott cím N” hibajelzést eredményez.

Vegyes argumentumkijelölés

Az 1.sz. és a 2. sz. argumentumkijelölés együtt is létezhet egy mondaton belül, a vezérlő elfogadja azt. Hibát akkor jelez, ha egy adott számú változóra kétszer akarunk hivatkozni.

Például:

```
A2.12 B3.213 J36.9 J-12 E129.73 P2200
```

A lokális változók az alábbi sorrendben veszik fel a címek értékeit:

```
A: #1=2.12
B: #2=3.213
J: #5=36.9 (első J)
J: #8=-12 (második J)
E: #8=HIBA, a #8 változó már ki lett töltve
```

Ebben a példában #8-nak a második J cím (értéke -12) már adott értéket. Mivel az E cím ér-

tékét is a #8 változó veszi fel, a vezérlő “2017 Tiltott cím E” hibát üzen.
Ha viszont a második J cím és az E cím megadásának sorrendjét felcseréljük,

```
A2.12 B3.213 J36.9 E129.73 J-12 P2200
nem jelez hibát és a J cím értékét a következő J változójára #11-re írja:
A: #1=2.12
B: #2=3.213
J: #5=36.9 (első J)
E: #8=129.73
J: #11=-12 (a harmadik J változójára kerül, mert a
    második J változója, #8 már foglalt)
```

☞ **Figyelem: Csak egykarakteres címek használhatók argumentumátadásra**, azaz ,C ,R ,A illetve a többkarakteres tengely és orsócémekek nem!

Az argumentumok szintjei

A **makróhívás is**, hasonlóan az alprogramhíváshoz, **lehet többszintű**, de az alprogramok és makrók hívásának együttes szintje maximum 16 lehet.

Emiatt a #1 ... #33 lokális változók is többszintűek. A főprogramhoz tartozó lokális változók szintje 0 (nullás szint), majd a makróhívások sorrendben az 1-es, 2-es, stb szinteket töltik fel, egészen maximum 16-ig.

Az alprogramhívás, vagy a rendszerprogram hívása nem változtatja meg a lokális változók szintjét.

Makróhívásból való visszatérés után az adott szint lokális változói megsemmisülnek #0-ra üresre törlődnek. A **főprogram lokális változói a program végén** kerülnek megsemmisítésre.

Makró hívása programszám alapján

A

P(programszám)

P címen (programszám) megadott számú makróprogram hívódik meg.

A P címen megadott **programszám programozási szabályait** a “[14.4.1 Programok azonosítása a tárban. A programszám \(O\)](#)” című fejezet tárgyalja a [108.](#) oldalon. A **programszámmal megadott makrók kiterjesztésének és tárban való elhelyezkedésének szabályai** megegyeznek az alprogramokéval. Lásd a “[14.4.2 Alprogram hívása \(M98\)](#)” című fejezetet a [109.](#) oldalon.

Makró hívása fájlnev alapján

☞ **Figyelem!** Fájlnev alapján **csak a G65 típusú** nem öröklődő **makrók hívhatók**, G66 és G66.1 típusú öröklődő makrók nem!

A

<fájlnev>

< > jelek között megadott fájlnevű makróprogram hívódik meg.

A **fájlnev megadási szabályait** a “[14.4.1 Programok azonosítása a tárban. A programszám \(O\)](#)” című fejezet tárgyalja a [108.](#) oldalon. A **fájlnévvel megadott makrók kiterjesztésének és tárban való elhelyezkedésének szabályai** megegyeznek az alprogramokéval. Lásd a “[14.4.2 Alprogram hívása \(M98\)](#)” című fejezetet a [109.](#) oldalon.

Rendszermakrók és alprogramok hívása és helyük a tárban

Rendszermakrókat és alprogramokat *paraméteren megadott címen hívhatunk*.

A rendszermakrók és alprogramok tárban való elhelyezkedésükre a makró és alprogramhívásoknál szigorúbb szabályok vonatkoznak.

A rendszermakróknak és alprogramoknak a tárban *csatornánként külön mappában*, a *Programs könyvtárban* lévő *SystemMacros* mappában kell lenniük:

..\Programs\SystemMacros\Channel1\
 ..\Programs\SystemMacros\Channel2\
 ..\Programs\SystemMacros\Channel8\
 ..\Programs\SystemMacros\Channel8\

Fájlnévüknek O betűvel kell kezdődniük, amit *4 decimális számjegynek* kell követni. A fájl *kiterjesztésének .nct*-nek kell lennie. Például:

O9010.nct

23.3.1 Az egyszerű makróhívás (G65)

A

G65 P(programszám) L(ismétlési szám) <argumentum kijelölés>

G65 <fájlnev> L(ismétlési szám) <argumentum kijelölés>

utasítás hatására a P címen (programszám) megadott számú, vagy a < > jelek között megadott fájlnevű makróprogram az L címen megadott számmal egymás után ismételt meghívódik.

Nem öröklődő hívás.

Az argumentumkijelölés szabályai G65 hívás esetén

Mindkét argumentumkijelölés használható.

G65 hívásban a G, P, L címek nem használhatók argumentum átadásra, de a megadott értékek beíródnak a megfelelő lokális változóba.

Többszörös G65 hívás

Ha makróból újra makrókat hívunk a makró szintjével a lokális változók szintje is növekszik.

főprogram	makró	makró	makró	makró
0. szint	1. szint	2. szint	3. szint	4.
szint	O_____	O_____	O_____	O_____
G65 P	G65 P	G65 P	G65 P	
	M99	M99	M99	M99
lokális változók				
0. szint	1. szint	2. szint	3. szint	4.
szint	#1	#1	#1	#1
#1	:	:	:	:
#33	#33	#33	#33	#33

Az első makró hívásakor a főprogram lokális változóit #1-től #33-ig eltárolódnak és az 1. szinten a lokális változók a híváskor megadott argumentumértékeket veszik föl. Az első szint-

ről történő újabb makróhívás esetén az első szint lokális változói #1-től #33-ig eltárolódnak, és a második szinten a lokális változók a híváskor megadott argumentum értékeket veszik föl. Többszörös hívás esetén az előző szint lokális változói eltárolódnak és a következő szinten a lokális változók a híváskor megadott argumentumértékeket veszik fel. M99 esetén, amikor visszatér a hívott makróból a hívó programba, az előző szint eltárolt lokális változói ugyanabban az állapotban visszaállításra kerülnek, mint amilyen állapotban a híváskor eltárolódtak.

Mukáljunk meg egy furatmintát G65-ös hívás segítségével. "A" címen adjuk meg a fúrás kódját a makrónak, a többi cím kitöltése a fúróciklusoknál megszokott módon történik.

A **főprogram** legyen a következő:

```
G54 G17 X0 Y0 Z20
...
G65 P300 A81 Z-2 R2 F300 S500 M3 (pontozás G81-gyel)
G65 P300 A83 Z-30 R2 Q6 E1 F100 S1000 (fúrás G83-mal)
G65 P300 A84.2 Z-30 R2 S1000 F1000 (menetfúrás G84.2-vel)
G0 Z20
...
```

Az O0300 makróban a lokális változókon keresztül átvesszük a hívás argumentumait, az orsóra vonatkozó adatokat

fordulatszámot S#19

forgásirányt M#13

a fúrás bemenő paramétereit

fúróciklus kódját A#1

furatmélységet Z#26

R pont koordinátáját R#18

előtolás értékét F#9

fogásmélység nagyságát Q#17

megközelítési pont távolságát E#8

Ezután a furat X, Y koordinátáinak felsorolása következik.

A fúrást végző **O0300 makró** törzse:

```
S#19 M#13 (orsó indítása)
G#1 X0 Y0 Z#26 R#18 F#9 Q#17 E#8 (fúróciklus beállítása)
X100
Y100
X0
X50 Y50
G80
M99
```

23.3.2 Öröklődő makróhívás minden mozgásparancs után: (G66)

A

G66 P(programszám) L(ismétlési szám) <argumentum kijelölés>
utasítás hatására a P címen (programszám) megadott számú **makróprogram** az L címen megadott számmal egymás után ismételt **meghívódik minden mozgásparancs végrehajtása után**.

Öröklődő hívás.

A kijelölt makró mindaddig meghívódik, amíg a

G67

makróhívás öröklődésének törlőparancsát nem programoztuk.

Az argumentumkijelölés szabályai G66 hívás esetén

Mindkét argumentumkijelölés használható.

G66 hívásban a G, P, L címek nem használhatók argumentum átadásra, de a megadott értékek beíródnak a megfelelő lokális változóba.

Példa

Az alkatrészprogram egy adott szegmensén minden mozgás után egy furatot kell készíteni:
Főprogram

```
...
G66 P1250 Z-100 R-1 X2 F130 (Z: a furat talppontja, R:
    afurat R pontja, X: várakozási idő, F: előtolás)
N100 G91 G0 X100
N110 Y30
...
N180 X150
G67
```

Az N110 mondatától kezdve az N180 mondatig bezárólag a pozicionálás végén meghívódik az O1250 makróprogram és az ott leírt fúrési műveletet végrehajtja, a G66 mondatban megadott bemenő paraméterekkel.

Makróprogram:

```
O1250
G0 Z#18          (gyorsmeneti pozicionálás Z irányban az R
                  címen megadott -1 pozícióra)
G1 Z#26 F#9      (az F címen megadott 130 mm/min
                  előtolással fúrás a Z címen megadott -100
                  talppontig)
G4 P#24          (várakozás a furat alján az X címen megadott
                  2 sec értékkel)
G0 Z-[#18+#26]  (a szerszám visszahúzása a kiindulási
                  pontra)
M99              (visszatérés a hívó programba)
```

Az O1250 makróprogram a főprogram N100 mondatából megörökli a G91 állapotot, ezért a pozicionálások inkrementálisan történnek.

Többszörös G66 hívás

G66 típusú makrók többszörös hívása esetén minden mozgásmondat végrehajtása után **először a legutoljára hívott makró hívódik, és ebből hívódnak meg visszafelé haladva a korábban**

hívott makrók. Lássuk a következő példát:

```

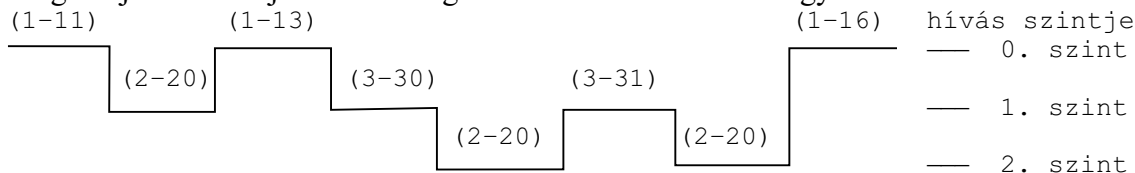
O0001
...
N10 G66 P2
N11 G1 G91 Z10      (1-11)
N12 G66 P3
N13 Z20             (1-13)
N14 G67             (G66 P3 hívás törlése)
N15 G67             (G66 P2 hívás törlése)
N16 Z-5             (1-16)
...

O0002
N20 X4              (2-20)
N21 M99

O0003
N30 Z2              (3-30)
N31 Z3              (3-31)
N32 M99

```

A végrehajtás sorrendje csak a mozgást tartalmazó mondatok figyelembe vételével:



A zárójelbe tett számok közül az első a végrehajtás alatt álló program száma a második pedig a végrehajtás alatt álló mondat száma.

Minden G66 hívást külön G67 utasítással kell törölni. Az első G67 a legutoljára hívott G66-ot törli, majd újabb G67 visszafelé haladva többi hívást. Az N14 mondatban megadott G67 utasítás az N12 mondatban hívott makrót (O0003) törli, az N15 mondatban megadott az N10 mondatban hívottat (O0002).

23.3.3 Öröklődő makróhívás minden mondatból: (G66.1)

A

G66.1 P(programszám) L(ismétlési szám) <argumentum kijelölés> utasítás hatására az összes utána következő mondatot argumentumkijelölésnek értelmezi, és a P címen (programszám) megadott számú **makróprogram** az L címen megadott számmal egymás után ismételt **meghívódik minden mondatra**.

A parancs hatása ugyanaz, mintha minden mondat G65-ös makróhívás lenne:

```

G66.1 P L
X Y Z = G65 P L X Y Z
M S = G65 P L M S
X = G65 P L X
G67

```

Öröklődő hívás.

A kijelölt makró mindaddig meghívódik, amíg a

G67

makróhívás öröklődésének törlőparancsát nem programoztuk.

Az argumentumkijelölés szabályai:

1. *A bekapcsolást végző mondatban* (ahol a G66.1 P L-t programoztuk):

G66.1 hívásban a G, P, L címek nem használhatók argumentum átadásra, de a megadott értékek beíródnak a megfelelő lokális változóba.

2. *A G66.1 utasítást követő mondatokban:*

A G, L, P cím is használható.

G cím (G: #10) azzal a megkötéssel, hogy a vezérlő **egy mondatban csak egy G** címre történő hivatkozást fogad el, ha több G címet programoztunk “2017 Tiltott cím G” hibajelzést ad.

A mondatvégrehajtás szabályai G66.1 esetén:

A kijelölt makró már abból a mondatból meghívódik, ahol a G66.1 kódot megadtuk, figyelembe véve az 1. pont alatti argumentumkijelölési szabályokat.

A G66.1 kódot követő mondatból a G67 kódot tartalmazó mondatig minden NC mondat makróhívást eredményez a 2. pont argumentumkijelölési szabályai alapján. Nem hívódik meg a makró, ha üres mondatot talál, pl.: N1240, ahol csak egy N címre történt hivatkozás, illetve makróutasítást tartalmazó mondatból.

Többszörös G66.1 hívás

G66.1 típusú makrók többszörös hívása esetén **először az legutoljára hívott makró hívódik minden mondat beolvasásakor** argumentumként kezelve ennek a mondatnak a címeit, majd ennek a makrónak a mondatait beolvasva és argumentumként kezelve az eggyel előbb megadott makró hívódik.

Minden G66.1 hívást külön G67 utasítással kell törölni. Az első G67 a legutoljára hívott G66.1-et törli, majd újabb G67 visszafelé haladva többi hívást.

Az alábbi programrészlet az XY síkban lett megírva, amikor a gépen a fej vertikális állású volt.

```
...  
G1 X40 Y30 F1000  
G3 X0 Y50 I-40 J-30  
G1 Y0  
...
```

Közben a fejet átszerelték horizontális állásba. Annak érdekében, hogy az XY síkban megírt programot ne kelljen újra megírni az Y és a Z tengelyeket fel kell cserélni, az X tengely irányát pedig ellentétesre váltani, hogy a koordinátarendszer jobbsodrású maradjon. Az O0400 makró hívásával megfordítjuk az X mozgás irányát és felcseréljük az Y és Z tengelyt:

A főprogram:

```
...  
G66.1 P400 (cseremakró hívása)  
G18 X0 Y0 Z-5 (síkkijelölés, pozicionálás)  
G1 X40 Y30 F1000 (O0400 hívás)
```

```
G3 X0 Y50 I-40 J-30      (O0400 hívás)
G1 Y0                    (O0400 hívás)
G67                       (öröklődő hívás törlése)
...

```

A makró:

```
IF[#10EQ66.1] GOTO10 (rekurzív hívás kivédése)
G#10 X-#24 Y#26 Z#25 I-#4 K#5 R#18 F#9 (cserék)
N10 M99

```

Már a főprogram G66.1 P400 mondatára meghívódik az O0400 makró. Ekkor a #10 változón a 66.1 kód jön át. Az első hívásra azonnal visszatérünk. A makró második mondata veszi át az argumentumokat és dolgozza fel azokat: megcseréli az Y és Z tengelyt és az X irányát.

23.3.4 Rendszermakró hívás paraméteren megadott G kódra

Egyedi G kódokat, illetve G kód tömböket jelölhetünk ki, paraméteren megadva, csatornánként, makróhívásra.

10 egyedi G kód kijelölése rendszermakró hívására

A paramétermezőben, *csatornánként, ki lehet jelölni legfeljebb 10 különböző G kódot*, amelyre makróhívást akarunk kezdeményezni. Ekkor az

Nn G65 Pp <argumentum kijelölés>

utasítássor helyett az

Nn **Gg** <argumentum kijelölés>

utasítássort kell leírni. A paramétermezőben azt kell beállítani, hogy a hívó G kód melyik programszámot hívja. G65, G66, G66.1, G67 kód nem adható meg ilyen célra.

Ezek a paraméterek a következők:

N1704 G(9010): G kód, amelyik az O9010.nct nevű programot hívja

N1705 G(9011): G kód, amelyik az O9011.nct nevű programot hívja

:

N1713 G(9019): G kód, amelyik az O9019.nct nevű programot hívja

Ha a paraméterre 0-át írunk, nem hívódik meg az adott programszámú makró.

Ha **G0-ra** akarunk *makróhívást* kezdeményezni, írjunk a paraméterre **1000**-et.

G kód csoport kijelölése rendszermakró hívására

Az alábbi paraméterek segítségével egy **G kód tömböt jelölhetünk ki**, csatornánként, makróhívásra. Az

N1714 Start G Macro paraméteren a G kód tömb kezdőszámát adjuk meg, decimális egész számmal,

N1715 Start Prg No paraméteren a Start G Macro paraméteren megadott G kódhoz tartozó makró programszámát adjuk meg,

N1716 No. of G Codes paraméteren a G kód tömb elemeinek számát adjuk meg.

Ha a No. of G Codes=0 nem hívódik makró ezekre a G kódokra.

Példa

Legyen például a csoport kezdő kódja G2000. Start G Macro=2000-at kell beállítani.

Ha a G2000 kód az O3400-as programot hívja, akkor Start Prg No=3400.

A No. of G Codes paraméteren a csoporthoz tartozó G kódok számát adhatjuk meg. Ha a csoportban 50 kód van, akkor No. of G Codes=50.

Ezek után a G kódok és programszámok közötti megfeleltetés:

1. G kód	G2000 →	1. programszám	O3400
2. G kód	G2001 →	2. programszám	O3401
3. G kód	G2002 →	3. programszám	O3402
.....			
50. G kód	G2049 →	50. programszám	O3449

Decimális pontot tartalmazó G kód csoport kijelölése rendszermakró hívására

Az alábbi paraméterek segítségével egy olyan G kód tömböt jelölhetünk ki, csatornánként, makróhívásra, amelyek kódja decimális pontot és egy tizedesjegyet tartalmaz. Az

N1717 Start Dec G Macro paraméteren a G kód tömb kezdőszámát adjuk meg, tizedes ponttal és egy tizedesjeggyel,

N1718 Start Prg No. Dec G paraméteren a Start Dec G Macro paraméteren megadott G kódhoz tartozó makró programszámát adjuk meg,

N1719 No. of Dec G Codes paraméteren a decimális pontot tartalmazó G kód tömb elemeinek számát adjuk meg.

Ha No. of Dec G Codes=0 nem hívódik makró ezekre a G kódokra.

Példa

Ha a csoport kezdő kódja pl. G310.5 a paramétert a következőképp töltjük ki: Start Dec G Macro=310.5.

Ha a G310.5 kód az O4000-es programot hívja, akkor Start Prg No. Dec G=4000.

A No. of Dec G Codes paraméteren a csoporthoz tartozó G kódok számát adhatjuk meg. Ha a csoportban 10 kód van, akkor No. of Dec G Codes=10.

Ezek után a G kódok és programszámok közötti megfeleltetés:

1. G kód	G310.5	→	1. programszám	O4000
2. G kód	G310.6	→	2. programszám	O4001
3. G kód	G310.7	→	3. programszám	O4002
.....				
10. G kód	G311.4	→	10. programszám	O4009

A G kód hívások öröklődővé tétele

Ha az egyedi hívásoknál az egyedi G kódokra, tömbös hívásoknál az N1714 Start G Macro, vagy az N1717 Start Dec G Macro **paraméteren negatív értéket adunk meg**, akkor a kijelölt **G kód, vagy G kód csoport öröklődő (modális) hívást** generál. Például, ha a G(9011)=-120, akkor a programban a G120 utasítás öröklődő hívást eredményez. Azt, hogy a hívás típusa milyen legyen a

N1755 Macro Contr #0 MEQ=0: a G hívás G66 típusú

N1755 Macro Contr #0 MEQ=1: a G hívás G66.1 típusú

paraméter állapota határozza meg. Ha a paraméter értéke 0, a makró minden mozgásmondat végén hívódik. Ha a paraméter értéke 1 a makró minden mondatra meghívódik.

G makró törzsében ugyanarra a G kódra történő hivatkozás

Ha egy **sztenderd G kódot jelölünk ki felhasználói hívásra** (pl. G01-et) és a **makró törzsében** ismét erre a kódra hivatkozunk ez a hivatkozás már nem eredményez újabb makróhívást, hanem a vezérlő **közönséges G kódként értelmezi** és hajtja végre azt.

Ha **nem sztenderd G kódot jelölünk ki felhasználói hívásra** (pl.G101-et) és a **makró**

törzsében ismételten **a hívó G kódra hivatkozunk** (esetünkben G101-re) “2123 Nem megvalósított funkció: G<szám> **hibajelzést ad a vezérlő.**

G makró törzsében M, S, ... hivatkozás, M, S, ... hívás törzsében G hivatkozás
 – rendszer G makró hívásból rendszer M, S, T, A, B, C, ASCII alprogram/makró hívása,
 – rendszer M, S, T, A, B, C, ASCII alprogram/ makró hívásból rendszer G makró hívása,
 paraméterállástól függően engedélyezett:

N1755 Macro Contr #1 ENC=0: nem kezdeményez hívást (közönséges M,S, ...G kódként kerülnek végre hajtásra),

N1755 Macro Contr #1 ENC=1: hívást kezdeményez.

A rendszer G makrók argumentumkészlete:

- ha a kód G65 vagy G66 típusú a G65–höz rendelt argumentumkészlet, valamint P és L,
- ha a kód G66.1 típusú, akkor argumentumkészletére az ott elmondottak érvényesek.

Az öröklődő hívás törlése G67 utasítással történik.

23.3.5 Rendszermakró hívás paraméteren megadott M kódra

Egyedi M kódokat, illetve egy M kód tömböt jelölhetünk ki, paraméteren megadva, csatornánként, makróhívásra.

10 egyedi M kód kijelölése rendszermakró hívására

A paramétermezőben, csatornánként, ki lehet jelölni legfeljebb 10 különböző M kódot, amelyre makróhívást akarunk kezdeményezni. Ekkor az

Nn G65 Pp <argumentum kijelölés>

utasítássor helyett az

Nn **Mm** <argumentum kijelölés>

utasítássort kell leírni. **Az Mm kód** ebben az esetben **nem kerül a PLC–nek átadásra**, hanem a megfelelő programszámú makró kerül hívásra.

A paramétermezőben azt kell beállítani, hogy a hívó M kód melyik programszámot hívja.

Ezek a paraméterek a következők:

N1733 M(9020): M kód, amelyik az O9020 sz. programot hívja

N1734 M(9021): M kód, amelyik az O9021 sz. programot hívja

:

N1742 M(9029): M kód, amelyik az O9029 sz. programot hívja

Ha a paraméterre 0-át írunk, nem hívódik meg az adott programszámú makró.

M kód csoport kijelölése rendszermakró hívására

Az alábbi paraméterek segítségével egy M kód tömböt jelölhetünk ki, csatornánként, makróhívásra. Az

N1743 Start M Macro paraméteren az M kód tömb kezdőszámát adjuk meg, decimális egész számmal,

N1744 Start Prg No. M Macro paraméteren a Start M Macro paraméteren megadott M kódhoz tartozó makró programszámát adjuk meg,

N1745 No. of M Macro Codes paraméteren az M kód tömb elemeinek számát adjuk meg.

Ha a No. of M Macro Codes=0 nem hívódik makró ezekre az M kódokra.

Példa

Legyen például a csoport kezdő kódja M500. Start M Macro=500-at kell beállítani.
Ha az M500 kód az O3500-as programot hívja, akkor Start Prg No. M Macro=3500.
A No. of M Macro Codes paraméteren a csoporthoz tartozó M kódok számát adhatjuk meg.
Ha a csoportban 20 kód van, akkor No. of M Macro Codes=20.
Ezek után az M kódok és programszámok közötti megfeleltetés:

1. M kód	M500	→	1. programszám	O3500
2. M kód	M501	→	2. programszám	O3501
3. M kód	M502	→	3. programszám	O3502
.....				
20. M kód	M520	→	20. programszám	O3519

A makróhívást indító M kód helyzete a mondatban

A paramétermezőben kijelölt, makróhívást kezdeményező M kódot a mondatban csak a / és az N cím (mondatszám) előzheti meg, a bemenő argumentumok csak ezután következhetnek.

M kódra hívott makró nem öröklődik

M kóddal mindig G65 típusú, tehát **nem öröklődő hívás** adható meg.

M makró törzsében ugyanarra az M kódra történő hivatkozás

Ha **a makró törzsben** ismételten **ugyanarra az M kódra hivatkozunk**, a makró nem hívódik újra, hanem az **M kód a PLC-nek kerül átadásra**.

M makró törzsében G, S, ... hivatkozás, G, S, ... hívás törzsében M hivatkozás

- M kódra indított makróhívásból rendszer G, S, T, A, B, C, M, ASCII alprogram/makró hívása,
 - rendszer G, S, T, A, B, C, M, ASCII alprogram/makró hívásból M kódra indított makró hívása,
- paraméterállástól függően engedélyezett:

- N1755 Macro Contr #1 ENC=0: nem kezdeményez hívást (közönséges M,S, ...G kódként kerülnek végre hajtásra),
- N1755 Macro Contr #1 ENC=1: hívást kezdeményez.

A makróhívást indító M kódok argumentumkészlete:

M kódra indított makróhívást tartalmazó mondatban **mindkét argumentumkészlet**, az 1.sz. és a 2. sz. is használható.

A hívó M kód megadása után a második M kódot már argumentumként értelmezi és a kódot átadja a #13 változónak.

23.3.6 Rendszeralprogram hívás paraméteren megadott M kódra

Egyedi M kódokat, illetve egy M kód tömböt jelölhetünk ki, paraméteren megadva, csatornánként, alprogramhívásra.

Az **alprogramot**, illetve a **makrót hívó M kódok között az a különbség**, hogy az **alprogramot hívó M kód semmilyen argumentumot nem ad** át az alprogramnak, míg a makrót hívó M kód igen.

10 egyedi M kód kijelölése rendszerprogram hívásra

A paramétermezőben, csatornánként, ki lehet jelölni legfeljebb 10 különböző M kódot, amelyre alprogramot akarunk hívni. Az **Mm kód** ebben az esetben **nem kerül a PLC-nek átadásra**, hanem a megfelelő programszámú alprogram kerül hívásra, vagyis az

Nn Gg Xx Yy M98 Pp

utasítás helyett a következő utasítás adható meg:

Nn Gg Xx Yy **Mm**

A paramétermezőben azt kell beállítani, hogy a hívó M kód melyik programszámot hívja. Ezek a paraméterek a következők:

N1720 M(9000): M kód, amelyik az O9000 sz. programot hívja

N1721 M(9001): M kód, amelyik az O9001 sz. programot hívja

:

N1729 M(9009): M kód, amelyik az O9009 sz. programot hívja

Ha a paraméterre 0-át írunk, nem hívódik meg az adott programszámú alprogram.

M kód csoport kijelölése rendszerprogram hívásra

Az alábbi paraméterek segítségével egy M kód tömböt jelölhetünk ki, csatornánként, alprogram hívásra. Az

N1730 Start M SubP paraméteren az M kód tömb kezdőszámát adjuk meg, decimális egész számmal,

N1731 Start Prg No. M SubP paraméteren a Start M SubP paraméteren megadott M kódhoz tartozó alprogram programszámát adjuk meg,

N1732 No. of M Codes paraméteren az M kód tömb elemeinek számát adjuk meg.

Ha a No. of M Codes=0 nem hívódik alprogram ezekre az M kódokra.

Példa

Legyen például a csoport kezdő kódja M600. Start M SubP=600-at kell beállítani.

Ha az M600 kód az O3600-as alprogramot hívja, akkor Start Prg No. M SubP=3600.

A No. of M Codes paraméteren a csoporthoz tartozó M kódok számát adhatjuk meg. Ha a csoportban 30 kód van, akkor No. of M Codes=30.

Ezek után az M kódok és programszámok közötti megfeleltetés:

1. M kód M600 → 1. programszám O3600

2. M kód M601 → 2. programszám O3601

3. M kód M602 → 3. programszám O3602

.....

30. M kód M630 → 30. programszám O3629

Az alprogram hívást indító M kód helyzete a mondatban

Tetszőleges helyre írhatjuk az M kódot a mondatban.

M alprogram törzsében ugyanarra az M kódra történő hivatkozás

Ha az **alprogramban** ismételten **ugyanarra az M kódra hivatkozunk**, az alprogram nem hívódik újra, hanem az **M kód a PLC-nek kerül átadásra**.

- M alprogram törzsében G, S, ... hivatkozás, G, S, ... hívás törzsében M hivatkozás
- M kódra induló alprogram hívásból rendszer G, S, T, A, B, C, M, ASCII alprogram/makró hívása,
 - rendszer G, S, T, A, B, C, ASCII alprogram/makró hívásból M kódra induló alprogram hívása,

paraméterállástól függően engedélyezett:

- N1755 Macro Contr #1 ENC=0: nem kezdeményez hívást (közönséges M,S, ...G kódként kerülnek végre hajtásra),
- N1755 Macro Contr #1 ENC=1: hívást kezdeményez.

23.3.7 Rendszeralprogram hívás paraméteren engedélyezett A, B, C, S, T kódra

A, B, C, S, T kódra, csatornánként, paraméteren engedélyezve, alprogramot hívhatunk.

A, B, C, S, T kód kijelölése alprogram hívásra

Az N1746 ABCST paraméter

- #0 AM=1 bitállásánál az A kódra az O9030.nct alprogram hívódik,
- #1 BM=1 bitállásánál a B kódra az O9031.nct alprogram hívódik,
- #2 CM=1 bitállásánál a C kódra az O9032.nct alprogram hívódik,
- #3 SM=1 bitállásánál az S kódra az O9033.nct alprogram hívódik,
- #4 TM=1 bitállásánál a T kódra az O9034.nct alprogram hívódik.

Alprogram hívásra kijelölt címek esetén a programba írt A, B, C, S, T érték nem kerül sem az interpolátornak (ha A, B, vagy a C cím tengelynek van kijelölve), sem a PLC-nek átadásra, hanem az A, B, C, S, T kód a fenti alprogramok hívását kezdeményezi.

Ekkor, ha például T kód alprogramhívásra van kijelölve a

Gg Xx Yy Tt

mondat ekvivalens az alábbi két mondattal:

#199=t

Gg Xx Yy M98 P9034

Az A, B, C, S, T kód argumentumának átadása az alprogramnak

Az **A, B, C, S, T** címnek adott értékek argumentumként az alábbi **globális változókra** íródnak be:

A kód → #195

B kód → #196

C kód → #197

S kód → #198

T kód → #199

Ezután az alprogram ezeket a változókat használhatja.

Az alprogram hívást indító A, B, C, S, T kód helyzete a mondatban

Tetszőleges helyre írhatjuk a kódokat a mondatban.

A, B, C, S, T alprogram törzsében a hívó kódra történő hivatkozás

Ha az **A, B, C, S, T** címre hívódó alprogramban ismételten **a hívó címre hivatkozunk**, az alprogram nem hívódik újra, hanem az **A, B, C, S, T** kód az **interpolátornak, vagy a PLC-nek kerül átadásra**.

A, B, C, S, T alprogram törzsében G, S, ... hivatkozás, G, S, ... hívás törzsében A, B, C, S, T hivatkozás

- A, B, C, S, T kódra induló alprogram hívásból rendszer G, M, S, T, A, B, C, ASCII alprogram/makró hívása (ha a hívás nem a hívó címről történik),
- rendszer G, M, S, T, A, B, C, ASCII alprogram/makró hívásból A, B, C, S, T kódra induló alprogram hívása (ha a hívás nem a hívó címről történik),
paraméterállástól függően engedélyezett:

N1755 Macro Contr #1 ENC=0: nem kezdeményez hívást (közönséges M,S, ...G kódként kerülnek végre hajtásra),

N1755 Macro Contr #1 ENC=1: hívást kezdeményez.

23.3.8 Rendszerprogram hívás paraméteren megadott ASCII kódra

Paraméteren megadott 4 különböző ASCII kódra, csatornánként, alprogramot hívhatunk. Az *ASCII kódok* közül az *angol ABC betűit* lehet választani.

ASCII kód kijelölése alprogram hívásra

Az

N1747 ASCII Code SubP1

N1748 ASCII Code SubP2

N1749 ASCII Code SubP3

N1750 ASCII Code SubP4

paraméteren 4 különböző kódot (angol ABC betűit) lehet beállítani. Ezután ezekre a kódokra a vezérlő az alábbi paramétereken beállított számú (Onnnn) alprogramokat hívja:

N1751 Prg No. ASCII Call1

N1752 Prg No. ASCII Call2

N1753 Prg No. ASCII Call3

N1754 Prg No. ASCII Call4

Az ASCII kód argumentumának átadása az alprogramnak

A kijelölt *ASCII címnek adott értékek* argumentumként az alábbi *globális változókra* íródnak be:

1. ASCII kód → #191

2. ASCII kód → #192

3. ASCII kód → #193

4. ASCII kód → #194

Ezután az alprogram ezeket a változókat használhatja.

Az alprogram hívást indító ASCII kód helyzete a mondatban

Tetszőleges helyre írhatjuk a kódokat a mondatban.

ASCII kódra hívódó alprogram törzsében a hívó kódra történő hivatkozás

Ha az *ASCII kódra hívódó alprogramban* ismételten *a hívó címre hivatkozunk*, az alprogram nem hívódik újra, hanem az *ASCII kód az interpolátornak, vagy a PLC-nek kerül átadásra*.

ASCII alprogram törzsében G, S, ... hivatkozás, G, S, ... hívás törzsében ASCII hivatkozás

- ASCII kódra induló alprogram hívásból rendszer G, M, S, T, A, B, C, ASCII alprogram/makró hívása (ha a hívás nem a hívó címről történik),
 - rendszer G, M, S, T, A, B, C, ASCII alprogram/makró hívásból ASCII kódra induló alprogram hívása (ha a hívás nem a hívó címről történik),
- paraméterállástól függően engedélyezett:

N1755 Macro Contr #1 ENC=0: nem kezdeményez hívást (közönséges M,S, ...G kódként kerülnek végre hajtásra),

N1755 Macro Contr #1 ENC=1: hívást kezdeményez.

23.3.9 Makrók és alprogramok mondatainak kijelzése automata üzemmódban.

A végrehajtás alatt álló makrók és alprogramok mondatait alapesetben a vezérlő listázza. Lehetőség van azonban a mondatok listázásának letiltására is. Ilyenkor a vezérlő az egész makrót, vagy alprogramot egy mondatnak tekinti és mondatonkénti üzemben nem áll meg a belső mondatokon.

A listázást az N1756 List Contr paraméter #0 MD8 és a #1 MD9 bitjeivel lehet engedélyezni, vagy tiltani. Az MD8 bit a 8000–tól 8999–ig, az MD9 bit a 9000–tól 9999–ig számozott alprogramok és makrók listázását szabályozza.

Ha az MD8, vagy az MD9 paraméter értéke 0, a 8000–tól 8999–ig, illetve a 9000–tól 9999–ig számozott alprogramok és makrók végrehajtásakor a makró illetve alprogram mondatai nem kerülnek listázásra, mondatonkénti üzemben a belső mondatokon nem áll meg a végrehajtás. Az MD8, vagy az MD9 paraméter 1 állásánál ezek mondatai is listázásra kerülnek, illetve mondatonkénti üzemben a belső mondatokon is megáll a vezérlő.

23.4 A megszakítási makró

Az alkatrészprogram futása közben fel lehet függeszteni az aktuális program végrehajtását, meghívni egy másik programot, majd annak végrehajtása után folytatni a felfüggesztett programot.

A megszakítást kiváltó események lehetnek például, hálózatkimaradás, szerszámtörés, vagy meghatározott időnként történő adatgyűjtés.

A *felfüggesztéskor meghívódó programot* nevezzük *megszakítási makrónak*.

A *megszakítási makró hívását* egy, a *PLC programból állított jel*, a CP_MINT, *indítja*.

Az alkatrészprogramban a *megszakítási makró hívását engedélyezni kell*, különben a PLC jel hatástalan.

Ezen kívül ki kell jelölni azt a programot, amit futtatni akarunk a jel hatására.

Az

M96 P(programszám)

vagy az

M96 <fájlnév>

utasítás *engedélyezi* a PLC programtól jövő CP_MINT *megszakítási jel működését*. A P címen (programszám) *megadott* számú *program hívódik meg* a megszakítási jel hatására.

A P címen megadott *programszám programozási szabályait* a “[14.4.1 Programok azonosítása a tárban. A programszám \(O\)](#)” című fejezet tárgyalja a [108.](#) oldalon. A *programszámmal megadott megszakítási makrók kiterjesztésének és tárban való elhelyezkedésének szabályai* megegyeznek az alprogramokéval. Lásd a “[14.4.2 Alprogram hívása \(M98\)](#)” című fejezetet a [109.](#) oldalon.

A megszakítási makró helye a tárban

Az N1758 Intrrt Contr paraméter #5 SYM bitállása dönti el, hogy az NC a megszakítási makrót hol keresse a tárban:

- #5 SYM=0 esetén: mindig abban a mappában, amelyikben a főprogram van, még akkor is, ha a megszakítási makrót nem a főprogramban engedélyeztük M96 P utasítással,
- #5 SYM=1 esetén: mindig a SystemMacros mappa megfelelő csatornához tartozó könyvtárban.

A megszakítási makróból való *visszatérés M99 kóddal* történik.

Az

M97

utasítás *letiltja a megszakítási makró hívását*, azaz a megszakítási makró többet nem hívódik meg, mégha a jel be is jön. Reset, illetve program vége szintén letiltja a megszakítást.

A megszakítási makró kizárólag

automata, vagy

kézi adatbeviteli

üzemmódban működik, amikor

start

állapot van.

A funkció működésének engedélyezése

Az N1758 Intrrt Contr *paraméter* #0 USD=1 bitállása *engedélyezi* a vezérlőn a *megszakítási makró működését*.

Ha a paraméter értéke #0 USD=0, a megszakítási makró nem működik, az M96, M97 kódokat a PLC szabadon használhatja funkcióként.

Megszakítás engedélyezése más kóddal

Az N1758 Intrrt Contr paraméter #4 MCD=0 bitállásánál a megszakítási makró *engedélyezés/tiltás* az *M96/M97* kódpárral történik.

Az N1758 Intrrt Contr paraméter #4 MCD=1 bitállásánál a megszakítási makró engedélyezés/tiltás az M96/M97-től különböző kódpárral történik.

Ekkor az N1759 M Code MI On *paraméteren* adhatjuk meg, hogy a megszakítást *milyen M kód engedélyezze*, az N1760 M Code MI Off paraméteren pedig azt, hogy a megszakítást *milyen M kód kapcsolja ki*.

Ez akkor lehet hasznos, amikor a PLC az M96, vagy az M97 kódot már valamilyen funkcióra felhasználta.

Alprogram típusú, vagy makró típusú megszakítás

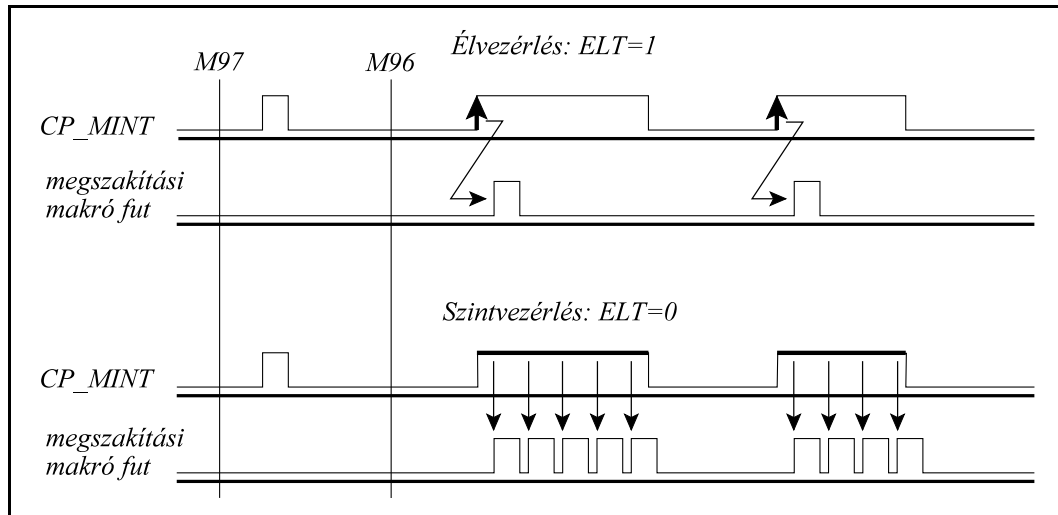
Az N1758 Intrrt Contr paraméter #1 STP=1 bitállásánál a megszakítás *alprogram típusú*, azaz a hívás a lokális változók szintjét nem növeli, a hívott programban a lokális változók értéke ugyanaz, mint a megszakított programban.

Ezzel szemben az STP=0 bitállásnál a megszakítás *makró típusú*, a hívott programban a lokális változók szintje nő, azaz a hívó programban és a megszakítási programban a lokális változók különbözőek lesznek.

Élvezérlésű és szintvezérlésű megszakítás

*Élvezérlésű*nek nevezzük a megszakítást, amikor a megszakítási makró a CP_MINT PLC jel *felfutó élére* hívódik meg. Ha a jel továbbra is bekapcsolva marad a megszakítási makróból való visszatérés után, a makró nem hívódik újra, csak ha a jel kikapcsolódik, majd újra visszakapcsolódik.

Ezzel szemben a *szintvezérlésű* megszakítás esetén, ha a megszakítási makróból való visszatérés után a CP_MINT jel még be van kapcsolva, a megszakítási makró *újra meghívódik*, mindaddig, amíg a CP_MINT jel ki nem kapcsolódik.



23.4-1 ábra

Az N1758 Intrrt Contr paraméter #3 ELT bitje dönti el, hogy a makróhívás milyen legyen. Ha #3 ELT=1, a makróhívás élvezérlésű, ha #3 ELT=0, akkor szintvezérlésű.

Megszakítás a mondat végrehajtása közben, vagy a mondat végén

A felhasználó eldöntheti, hogy a megszakítási makró **a jel bejövetelének pillanatában** elkezdjen futni, vagy várja meg, hogy az aktuális mondatban a mozgás befejeződjék és **csak a mondat végén** kezdje a futást.

Az N1758 Intrrt Contr paraméter #2 TPI=0 bitállásnál a megszakítási makró azonnal meghívódik, míg #2 TPI=1 bitállásnál csak a mondat végén.

Öröklődő információk a megszakítási makróban

A megszakítási makró hívása különbözik a síma alprogramhívástól.

Alprogram hívás esetén (M98 Pp) az alprogram megőrökli a hívó programból a #4001...#4130 a megelőző mondatban érvényes öröklődő információkat és a #4201...#4330 a végrehajtás alatt álló mondatban érvényes öröklődő információkat. A két információ természetesen különböző, hiszen a végrehajtás a puffereles miatt hátrébb tart, mint a mondatok előfeldolgozása.

Más a helyzet a megszakítási makró esetén.

A megszakítás a program végrehajtása közben történik, ezért a **megszakítási makró (M96 Pp) a végrehajtás alatt álló azaz a megszakított mondatban érvényes öröklődő információkat a #4401...#4530**

változókon olvashatja ki.

A #4001...#4130, illetve a #4201...#4330 öröklődő információk a megszakítási makróba történő belépés után inicializálódnak.

A megszakítási makró végrehajtása során a #4001...#4130 és a #4201...#4330 változók a megszokott módon működnek, míg a #4401...#4530 változók megőrzik a megszakítás pillanatában érvényes öröklődő információkat.

Visszatérés a megszakítási makróból M99-cel

A vezérlő **eltárolja a megszakítás pillanatában érvényes öröklődő információkat**, majd **M99-cel való visszatérés után visszaállítja az elmentett öröklődő információkat**.

Ha a programot interpoláció közben szakítottuk meg, visszatérés után befejezi az interpolációt, ha a mondat végén szakítottuk meg, leveszi a következő mondatot és azt hajtja végre.

Ezért, ha a megszakítási makróban mozgást is kell programozni, célszerű a megszakítási makróba való belépés után eltárolni a mondatvégi pozíciót (#5001...), majd visszatérés előtt visszaállni erre a pozícióra.

Visszatérés a megszakítási makróból M99 Pp-vel

Ha M99 Pp paranccsal térünk vissza a megszakítási makróból, a megszakított program P címen megadott számú mondatától folytatja a megmunkálást.

M99 Pp-vel való visszatérés után nem állítja vissza az elmentett öröklődő információkat, hanem a megszakítási makróban kialakult öröklődő információkkal folytatja a programot.

23.5 NC és makró utasítások. A makromondatok végrehajtása

NC és makró mondatok

A programnyelvben megkülönböztethetünk *NC és makró mondatokat*.

*NC mondatok*nak tekintjük a hagyományos, G, M stb. kódokkal leírt mondatokat, még akkor is, ha az egyes címek értékei, nem csak számértéket vesznek fel, hanem változókat, vagy formulát.

*Makró utasítások*nak a következő mondatokat tekintjük:

- az értékadó utasítást tartalmazó mondatot: #i=#j
- a feltételes, vagy ciklusszervező utasítást tartalmazó mondatot: IF, WHILE
- a kontrolparancsokat tartalmazó mondatokat: GOTO, DO, END
- a makrohívást tartalmazó mondatokat: G65, G66, G66.1, G67, vagy azok a G, vagy M kódok, amelyek makrohívást indítanak.
- az alprogramhívást (M98 P, vagy A, B, C, S, T, M-re indított alprogram)
- az alprogramból, vagy makróból való visszatérés kódját (M99)

Az utasítások szinkronizálása (G53)

Az N1301 DefaultG2 paraméter #6 MBM=1 bitállásánál (multibuffer mód) a mondatelőkészítő előre beolvassa úgy az NC, mind a makró mondatokat, feldolgozza azokat, és beteszi a kiegyenlítőtarba (puffer). A végrehajtók, az interpolátor és a PLC, a kiegyenlítőtarból szedik ki a mondatokat a program végrehajtása során. Erre azért van szükség, hogy az interpolátor folyamatosan tudja a tengelyeket mozgatni és ne kelljen várnia a következő mondat feldolgozására.

Az előre történő mondatfeldolgozásnak és pufferelésnek az a következménye, hogy a végrehajtó sok száz mondattal lemarad a mondatelőkészítőtől. Például a mondatelőkészítő már az 1500-ik mondatot dolgozza fel, de a végrehajtó még csak az 1000-ik mondatot hajtja végre.

Ez az oka annak, hogy megkülönböztetjük a #4001...#4130 (#_BUFx) és a #4201...#4330 (#_ACTx) változókat. Míg az előző a legutoljára a kiegyenlítőtarba rakott mondatokról ad információ, addig az utóbbi a végrehajtás alatt álló mondatokról.

Vannak esetek, amikor a mondatok előfeldolgozását és pufferelését fel kell függeszteni, például, ha egy tengely adott pozíciójára akarunk várni programvégrehajtás közben.

A külön mondatba írt

G53

utasítás felfüggeszti a mondatok előreolvasását. Megvárja, amíg a mondatpuffer kiürül és csak azután kezdi el beolvasni és feldolgozni a következő mondatot.

A makromondatok végrehajtása

A makromondatokat végrehajthatja a vezérlő az NC mondatok végrehajtásával párhuzamosan, vagy azt követően. Az NC és makromondatok végrehajtását szabályozó paraméter az N1755 Macro Contr #2 SBM. Ha a paraméter:

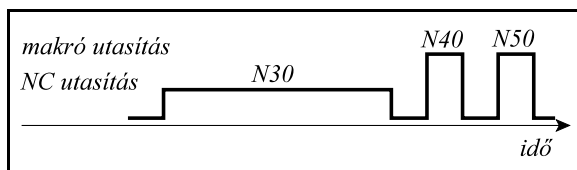
- =0: az NC és makró mondatokat a programban leírt sorrendben hajtja végre,
- =1: az NC mondatok végrehajtása közben végrehajtja a makró utasításokat.

Példa:

SBM=0

```
%O1000
...
N10 #100=50
N20 #101=100
N30 G1 X#100 Y#101
N40 #100=60 (értékadás N30 után)
N50 #101=120 (értékadás N30 után)
N60 G1 X#100 Y#101
```

Az N40 és N50 mondatokban leírt értékadást az N30 mondat végrehajtása után végzi el.



23.5-1 ábra

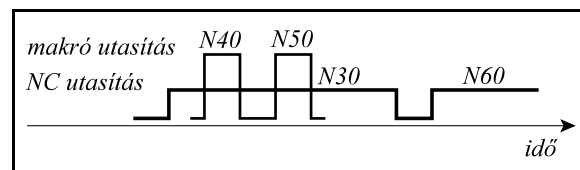
☞ *Következmények:*

- a program végrehajtása lassabb,
- ha az N30 mondat végrehajtását megszakítjuk, majd a megmunkálást újraindítjuk, mivel az N30 mondat változóit még nem írta át az N40, N50 mondat, a megmunkálás egyszerűen folytatható.

SBM=1

```
%O1000
...
N10 #100=50
N20 #101=100
N30 G1 X#100 Y#101
N40 #100=60 (értékadás N30 közben)
N50 #101=120 (értékadás N30 közben)
N60 G1 X#100 Y#101
```

Az N40 és N50 mondatokban leírt értékadást az N30 mondatban történő mozgás közben végzi el.



23.5-2 ábra

☞ *Következmények:*

- a program végrehajtása gyorsabb,
- ha az N30 mondat végrehajtását megszakítjuk, majd a megmunkálást újraindítjuk, mivel az N30 mondat változóit már átírta az N40, N50 mondat, a megmunkálás nem folytatható, csak, ha az N30 mondatra mondatkeresést indítunk.

23.6 Üregmaró makróciklus

A

G65 P9999 X Y Z I J K R F D E Q M S T

utasítás üregmaró ciklust indít.

A ciklus hívása előtt az üreg geometriai közepe fölé kell állni a kiválasztott síkban, az anyag felületétől biztonsági távolságnyra. A ciklus végén ugyanerre a pontra húzza vissza a szerszámot.

A mondat címeinek értelmezése:

X: az üreg X irányú mérete

Y: az üreg Y irányú mérete

Z: az üreg Z irányú mérete

A G17, G18, G19 utasítás dönti el, hogy a három koordináta közül melyik az üreg hosszúsága, szélessége illetve mélysége. Például: G17 esetén Z az üreg mélysége, X és Y közül a hosszabb koordináta az üreg hosszúsága, a rövidebb a szélessége. Ezeket az értékeket abszolút értékben pozitív számként kell beadni.

R: az üreg sarkainak sugara

R címen kell megadni az üreg sarkainak esetleges lekerekítését. Ha az R címet nem töltjük ki az üreg sarkainak lekerekítése a szerszám sugarával lesz egyenlő.

I: biztonsági távolság mélységi irányban G19 esetén

J: biztonsági távolság mélységi irányban G18 esetén

K: biztonsági távolság mélységi irányban G17 esetén

A kiválasztott sík függvényében I

(G19), J (G18), vagy K (G17) címen kell megadni a szerszámirányú biztonsági ráhagyást a mondatban. A vezérlés a ciklus indulásakor feltételezi, hogy a szerszám hegye a darab felületétől ilyen távolságra áll. Az üreg kimarása közben pedig, amikor egy szint kiszedésével végzett, ekkora távolságra emeli ki a szerszámot, hogy a következő szint kiszedéséhez a kezdőpontra álljon.

D: a szerszám sugárkorrekcióját tartalmazó rekesz címe

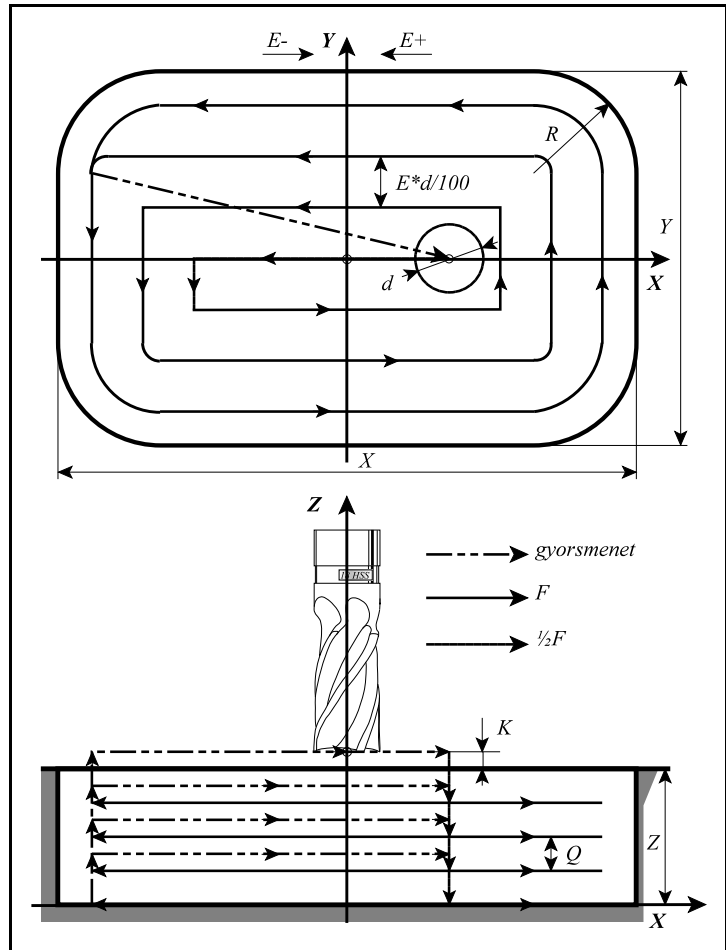
A programban használt szerszám sugárkorrekciós regiszterének számát D címen kötelezően meg kell adni. Az üregmarást egyébként G40 állapotban kell végezni.

E: fogásszélesség a maróátmérő százalékában

+ előjellel: megmunkálás az óramutatóval ellentétes

- előjellel: megmunkálás az óramutatóval egyező irányban

E címen két információt közölhetünk a vezérlővel. E értéke azt adja meg, hogy a fogásszélesség mekkora legyen a maróátmérő százalékában. Ha nincs megadva, a vezérlés automatikusan



23.6-1 ábra

+83%-ot tételez föl. A vezérlés az üreg szélességének függvényében az E címen megadott adatot módosíthatja úgy, hogy egy szint kiszedésénél a fogásvétel értéke egyenletes legyen. A módosítás azonban csak csökkentés lehet. Az E cím előjele a marás irányát mondja meg. Ha E+, azaz pozitív, a megmunkálás az óramutató járásával ellentétes, ha E-, azaz negatív, a megmunkálás az óramutató járásával megegyező irányban történik.

Q: fogásmélység

Q címen adhatjuk meg a fogásmélységet az alkalmazott mértékrendszerben azaz, mm-ben, vagy inch-ben. A vezérlés az üreg mélységének függvényében a programozott értéket az egyenletes fogásfelosztás érdekében felülbíráhatja. A módosítás azonban csak csökkentés lehet.

F: előtolás

F címen adhatjuk meg a ciklus során alkalmazott előtolás nagyságát. Ha F címnek nem adunk értéket az öröklött F értéket veszi figyelembe. Az F érték 50%-t alkalmazza az alábbi esetekben:

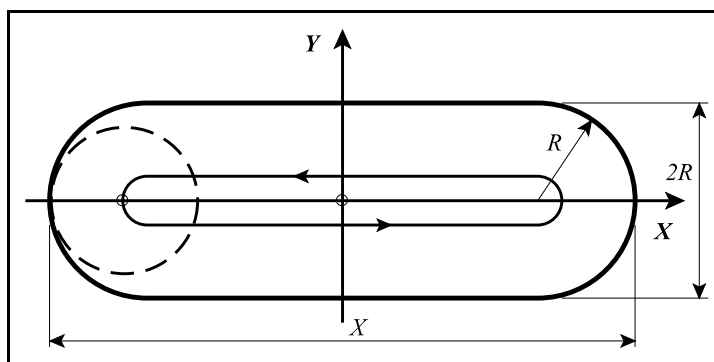
- Amikor egy szintet elkezd kibontani és a szerszám irányában Q mélységet lefúr,
- Az üreg hosszirányban történő marásánál mindaddig, amíg a szerszám mindkét oldalon terhelve van.

M S T: funkció

Az üregmarást hívó mondatban egy db. M, ill. S, T funkciót lehet megadni, amit a vezérlő a marás megkezdése előtt hajt végre.

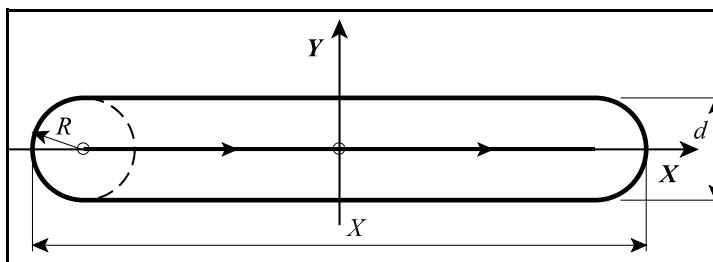
Az üregmarás elfajuló esetei:

Ha az üreg szélessége nem lett megadva az üreg sarkainak sugarát kétszer veszi és ez lesz az üreg szélessége.



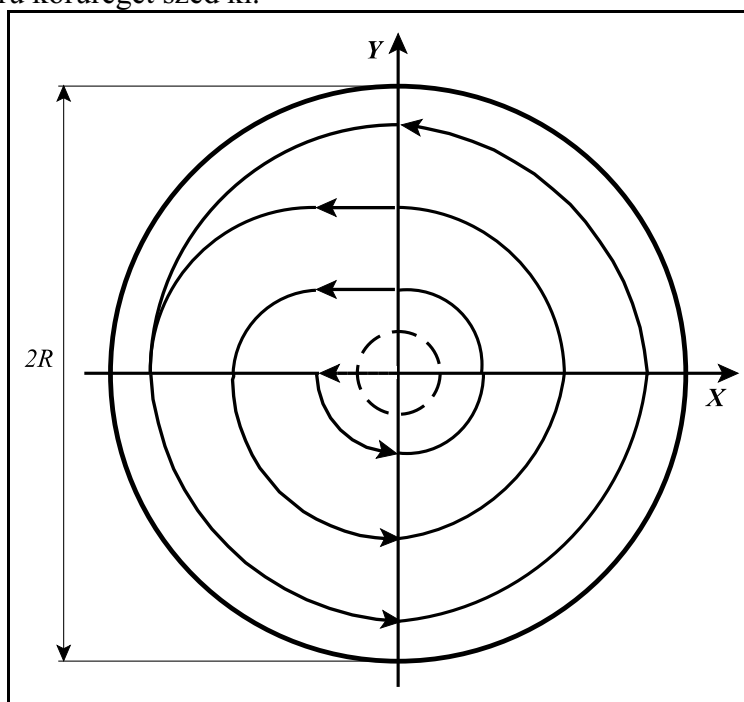
23.6-2 ábra

Ha sem az üreg szélessége, sem a sarok lekerekítési sugara nem lett megadva az alkalmazott szerszám átmérőjét veszi az üreg szélességének (horony).



23.6-3 ábra

Ha sem az üreg hosszúsága, sem a szélessége nem lett megadva, csak R címet programoztunk, akkor egy R sugarú körüregget szed ki.



23.6-4 ábra

Ha sem hosszúság, sem szélesség, sem sugár nem lett megadva akkor a ciklus fúrássá fajul.

Az üregmarás végrehajtása során előforduló hibajelzések:

MACRO ERROR 1: mondatkitöltési hiba. Lehetséges okai:

- Az üregmélység nincs megadva,
- A szerszámsugár nics megadva,
- A fogásmélység nincs megadva.

MACRO ERROR 2: méretmegadási hiba. Lehetséges okai:

- Ha az üreg hosszúságának, vagy szélességének megadott méret kisebb, mint az üreg sugarának kétszerese,
- Ha az üreg hosszúsági, vagy szélességi mérete kisebb, mint a D címen lehvívott szerszámátmérő,

- Ha a fogásszélességre megadott érték 0, vagy a lehívott szerszámátmérő 0,
- Ha a fogásmélység értéke 0, azaz Q címre 0 lett programozva.

24 Paraméterek írása és olvasása

A paraméterek segítségével a vezérlő működését lehet az igényeknek megfelelően beállítani. A paraméterek a vezérlő nem felejtő memóriájában kerülnek eltárolásra. A vezérlő paramétereit alkatrészprogramból át lehet írni, és be lehet olvasni.

Minden *paraméternek* van egy maximum *négyjegyű azonosítószáma*, amely a tárban azonosítja a paramétert. Ezekhez *az azonosítószámokhoz tartozhat egy* (globális paraméterek esetén), *vagy több* (csatornánként, tengelyenként, orsónként indexelt) különböző *érték*. A paramétereket *számbábrázolásuk*, *hatókörük* és *figyelembe vételük* módja szerint osztályozhatjuk.

A paraméterek számbábrázolásuk szerinti osztályozása

Bites típusú paraméterek: értéktartományuk

0, vagy 1
lehet.

Egész típusú paraméterek (DWORD): értéktartományuk

előjel nélkül: 0,...4294967295, vagy
előjelesen: -2147483648... +2147483647
lehet

Lebegőpontos paraméterek (double): értéktartományuk

negatív számok esetén: $-1.7 \times 10^{308} \dots -5.0 \times 10^{-324}$
pozitív számok esetén: $5.0 \times 10^{-324} \dots 1.7 \times 10^{308}$
lehet.

A paraméterek hatókörük szerinti osztályozása

Globális paraméterek: globálisnak nevezzük azokat a paramétereket, amelyek a vezérlő minden csatornájában érvényesek. Ilyen paraméterek például az N2201 Waiting M Codes Min és az N2202 Waiting Codes Max, amely a csatornák közötti összevárást szabályozó M kód csoportot jelölik ki és minden csatornára érvényesek.

Globális paraméterek esetén minden azonosítószámhoz *egy* paraméterérték tartozik.

Csatornánként megadható paraméterek: ezek olyan paraméterek, amelyekre csatornánként különböző értékek írhatók. Ilyen paraméter például az N1500 Return Val G73 csoport, amelyen a visszahúzási távolságot állíthatjuk be G73-as fűrciklusban, és amelyre csatornánként különböző érték írható.

Csatornánként megadható paraméterek esetén minden azonosítószámhoz *maximum 8* paraméterérték tartozhat.

Tengelyenként megadható paraméterek: ezek olyan paraméterek, amelyekre tengelyenként különböző értékek írhatók. Ilyen paraméter például az N0106 Axis properties csoport #0 DIA bitje, amelyen tengelyenként beállítható, hogy az adott tengelyt sugárban, vagy átmérőben kívánjuk programozni.

Tengelyenként megadható paraméterek esetén minden azonosítószámhoz *maximum 32* paraméterérték tartozhat.

Orsónként megadható paraméterek: ezek olyan paraméterek, amelyekre orsónként különböző értékek írhatók. Ilyen paraméter például az N0608 Spindle Encoder Counts csoport, amelyen orsónként meg lehet adni az egyes orsókra szerelt jeladó impulzusainak számát.

Orsónként megadható paraméterek esetén minden azonosítószámhoz *maximum 16* paraméterérték tartozhat.

A paraméterek figyelembe vételük szerinti osztályozása

Futásidőben figyelembe vett paraméterek: Ezeket a paramétereket átírásuk után azonnal figyelembe veszi a vezérlő.

Újraindítás után figyelembe vett paraméterek: Ezeknek a paramétereknek az átírását a vezérlő csak újraindítás (ki-, majd bekapcsolás után) veszi figyelembe.

Vészállapot esetén írható paraméterek: Ezeknek a paramétereknek az átírását csak vészállapotban engedélyezi a vezérlő.

Az "NCT2xx szerszámgép vezérlők Paraméterek" című leírás tartalmazza az egyes paraméterek részletes leírását.

24.1 Paraméterek írása alkatrészprogramból (G10 L52)

A külön sorba írt

G10 L52 (paraméter írás be)

utasítás bekapcsolja a paraméter írás funkcióját. Ezután külön sorban kell megadni az egyes paraméterek sorszámait és értékeit az alábbiak szerint:

N_ (Q_) R_ (globális paraméter írása)

N_ P_ (Q_) R_ (csatorna/tengely/orsó paraméter írása)

...

...

A paraméterírás végét a külön sorba írt

G11 (paraméter írás vége)

utasítás zárja le.

Az N, P, Q, R címek értelmezése:

N: a paraméter azonosítószáma (0-9999). A vezető nullák elhagyhatók.

☞ **Figyelem!** Csak olyan számú paramétert lehet a G10 L52 utasítással átírni, amelyik figyelembe vételéhez nem kell újraindítás, vagy átírásához vészállapot!

P: a csatorna (1-8), a tengely (1-32), vagy az orsó (1-16) száma. Zárójelben adtuk meg, hogy az egyes esetekben milyen P érték adható meg.

☞ **Figyelem!** Ha csatornánként megadható paramétert írunk, és a P címet nem töltjük ki, az utasítás mindig annak a csatornának a paraméterét írja át, amelyikben a program fut.

Q: bites paraméter esetén a beírandó bit száma 0-tól 7-ig megadva,

R: a paraméter értéke. R értékénél elfogadott az I inkrementális operátor, amely a paraméter aktuális értékét inkrementálja az RI címen megadott értékkel.

Lebegőpontos adat esetén tizedespontot (.) lehet használni.

A G10 L52 utasítással módosított paraméterek mentésre kerülnek, tehát a módosítás a következő bekapcsoláskor is érvényes lesz.

Az N, P, Q, R címek hibás kitöltésére a "2002 <N, P, Q, vagy R> adat értékhatáron kívül" hibajelzést adja.

N címre: ha nincs ilyen azonosítószámú paraméter,
 P címre: ha nincs ilyen számú csatorna/tengely/orsó,
 Q címre: ha Q értéke kisebb, mint 0, vagy nagyobb, mint 7,
 R címre: ha az N azonosítójú paraméterre megengedett értékhatáron kívüli értéket írunk R-re.

Ha az N, P, Q, R adatok közül valamelyik hiányzik “2004 <N, P, Q, R> adat hiányzik)” hibát üzen.

Ha olyan paramétert akarunk átírni, amelynek figyelembe vételéhez újraindítás kell, vagy átírásához vészállapot szükséges, akkor “2193 Az Nnnnn paraméter nem módosítható G10 utasítással” hibaüzenetet küldi.

Példa paraméterek írására:

```

...
G10 L52                (paraméterírás bekapcsolása)
N107 P4 Q0 R1         (N0107 RollOver Control A4- REN)
N1339 (P1) R0.5       (N1339 Radius Diff)
N1746 (P1) Q1 R1      (N1746 ABCST BM=1)
G11                   (paraméterírás kikapcsolása)
...

```

24.2 Paraméterek olvasása alkatrészprogramból (PRM)

A

#a=PRM[#b,#c]

és a

#a=PRM[#b,#c] / [#d]

értékadó utasítással a vezérlő *bármely paraméterének értéke korlátozás nélkül kiolvasható.*

Az argumentumok jelentése:

#a: egy írható makrováltozó.

#b: a paraméter azonosítószáma. Lehet indirekten egy makrováltozón, vagy közvetlenül egy számértékkel megadni.

#c: bites paraméter olvasásakor a paraméter bitszáma. Lehet indirekten egy makrováltozón, vagy közvetlenül egy számértékkel megadni.

#d: a csatorna (1-8), a tengely (1-32), vagy az orsó (1-16) száma. Zárójelben adtuk meg, hogy az egyes esetekben #d-n milyen érték adható meg. Lehet indirekten egy makrováltozón, vagy közvetlenül egy számértékkel megadni.

☞ **Figyelem!**

Ha csatornánként megadható paramétert olvasunk, és a / [#d] részt az utasításból elhagyjuk, az utasítás mindig annak a csatornának a paraméterét olvassa be, amelyikben a program fut.

Ha a PRM utasítás argumentumait makrováltozókon adjuk meg és a makrováltozó értéke #0 (üres), az olyan hatású, mintha azt az argumentumot nem adtuk volna meg.

Az argumentumok lehetnek makrokifejezések eredményei is.

Ha a PRM utasítás nem egy értékadó utasítás jobb oldalán áll, “2017 Tiltott cím PRM” hibát üzen.

Ha a PRM utasítás #b argumentuma, azaz a paraméter száma hiányzik, “2064 Szintaktikai hiba” üzenetet küld.

Ha a PRM utasítás #c bitszám, vagy a /[#d] argumentuma hiányzik, “2194 PRM függvény <2., 3.> paraméter hiányzik” hibát üzen.

Példa paraméterek olvasására:

```
...
#101=PRM[107,0]/[4]      (N0107 RollOver Control A4- REN)
#102=PRM[1339] (/ [1])  (N1339 Radius Diff)
#103=PRM[1746,1] (/ [1]) (N1746 ABCST BM=1)
#104=PRM[2201]          (N2201 Waiting M Codes Min)
...
```

Jegyzetek

