

Numerikus vezérlésű szerszámgépek programozása számítógéppel.

Tankó József

Az Egyesült Államokban mintegy 15, Nyugat-Európában mintegy 10, de világszerte több év óta mind szélesebb körben használják a numerikus /számjegyes, digitális/ vezérlésű szerszámgépeket a hagyományos szerszámgépek helyett ill. mellett. Ezeket a számítógépekhez hasonlóan programozni kell és a programozás könnyítésére kezdettől fogva alkalmaznak elektronikus számítógépeket is. Hazánkban most kezdődik a numerikus vezérlésű szerszámgépek fejlesztése és gyártása és így felmerült ezek számítógép segítségével történő programozásának problémája is. Ez teszi aktuálissá és indokolttá azt a munkát, amelyet az Intézet a Villamos Automatika Intézet megbízásából végzett és végez ezen a területen. Az első tevékenység egy tanulmány - "Tanulmány a numerikus vezérlésű szerszámgépek automatikus programozásáról". Összeállította: Tankó József, Fabók Julianna, Szelezsán János. MTA Számítástechnikai Központja, Budapest, 1966. /Kézirat/ - elkészítése volt a probléma lényegéről, a világon e területen eddig elért eredményekről. A jelen ismertető ennek a tanulmánynak egyes részleteiről és az ott bővebben kifejtett problémákról nyújt vázlatos képet.

Bevezetés.

A modern technika napjaink gyors ütemű fejlődésében majdnem egy időre esik az elektronikus számítógépek /továbbiakban: ESz/ fejlődésének megindulásával, a numerikus vezérlésű szerszámgépek /továbbiakban: NSz/ fejlődésének megindulása. Eltekintve attól, hogy mindkét ág csirái már korábban és különböző időben megjelentek, az ugrásszerű fejlődés nem véletlenül kez-

dődött el mintegy 10-15 éve. Az elektronika és kibernetika fejlődése ekkor teremtette meg fejlődésük feltételeit.

Az ESz és NSz egyaránt önszabályozó, programozható elektronikus és többé-kevésbé mechanikai rendszer. Az ESz-nél az elektronikus központi berendezés /az u.n. aritmetikai egység, vezérlő egység és memória együttesen/ a funkcionális fő rész, amelyet a többi, perifériális berendezések kiszolgálnak; A NSz-nél a fő funkcionális rész számológépi felfogásban egy speciális perifériális berendezés, a szorosam vett szerszámgép és az elektronikus vezérlő berendezés ennek "kiszolgálására" hivatott. Hasonlattal élve a NSz-nél az "agy" kevésbé fejlett, mint az ESz-nél, ellenben egyik "szerve" a speciális feladatnak megfelelően magas fejlettségű. Ennek egy fontos következménye a magasabb igényű és racionálisabb gyakorlati használatnál kerül felszínre. A programozás automatizálása az ESz-nél megoldható saját szervezésen belül /"programozó program" konstruálása/, a NSz-nél azonban a programozás automatizálásához munkába kell vonni az univerzális ESz-et is. Az a megoldás, hogy ehelyett fejlesszék ki a NSz "agyát" is úgy, hogy egy univerzális ESz feleslegessé váljék, nem bizonyult ésszerűnek és gazdaságosnak. Jóllehet a korszerű, nagy teljesítményű NSz programozásához szükség van ESz-re, ugyanakkor azonban ez az ESz sok NSz-et tud egyszerre kiszolgálni és emellett számtalan egyéb feladat megoldására használható, így használata gazdaságossá tehető.

A NSz-eket a mind bonyolultabb és pontosabb megmunkálást igénylő, nagyrészt kis szériákban gyártott alkatrészek készítésének szükséglete hívta életre. A NSz-ek számológéppel történő programozását pedig az, hogy az adatelőkészítés és programozás bizonyos határon túl történő megvalósítása számológép nélkül nem gazdaságos, sokszor lehetetlen. A számológéppel történő programozás racionalizálására és tökéletesítésére való törekvés azután létrehozta a különféle automatikus programozó rendszereket és szimbólikus programozó nyelveket.

A következőben a numerikus vezérlésű szerszámgépek számológépekkel történő programozásával kapcsolatos alapkérdéseket érintjük, és áttekintést adunk a programozó rendszerek osztályozási szempontjairól, a rendszerek kialakulásáról és a jelenleg használt rendszerekről.

A NSz-ek programozása.

A NSz-eket mint már a bevezetőben említettük, az ESz-ekhez hasonlóan programozni kell. Ez azt jelenti, hogy előírt formában lépésről-lépésre közölni kell a szerszámgép vezérlő berendezésével, hogy mit csináljon a szerszámgép egy adott munkadarab elkészítése céljából. A közlés kódolt információk alakjában valamilyen információhordozó felhasználásával történik /pl. lyukszalag, lyukkártyaköteg, vagy mágnesszalag/. Az információk megfelelő formában és sorrendben a munkadarab programját képezik és ennek elkészítése a munkadarab-programozás.

Tartalmilag a programnak mindenképpen magában kell foglalnia a munkadarab megmunkálendő részeinek egyértelmű geometriai megadását, a megmunkálás sorrendjének és módjának ugyancsak egyértelmű meghatározását, valamint a szerszámgép működéséhez szükséges funkciók működtetésének ütemezését.

Nem kívánunk itt foglalkozni egyáltalán a szerszámgépek utasításrendszereivel /programozó nyelveivel/ a kódrendszerekkel és információhordozókkal. Ezek ugyanis nem lényegbevágóan érintik a számológéppel történő programozás problémáját.

Nem közömbös azonban problémánk szempontjából az a kérdés, hogy a szerszámgép vezérlő rendszere milyen részletes, vagy kevésbé részletes információkat fogad el a szerszám mozgás pályájára vonatkozóan.

Ez érthetővé válik, ha figyelembe vesszük a következőket: A szerszámgépen a szerszám helyváltoztatása az egyes koordináta-

irányokba elmozdító szervomotorok segítségével történik, az u.n. számokon. A NSz-eknél a motorok impulzusok hatására meghatározott egységekkel mozgatják el a szerszámot a megfelelő számok mentén. Ilymódon egy görbe kontur, vagy felület menti mozgatásnál két, vagy több szán menti összehangolt mozgatásra van szükség, amelyhez a motoroknak időben megfelelően ütemezett vezérlő impulzusok tömegét kell kapniuk. Mármost a vezérlőberendezés és programozás szempontjából a következőket lehet megállapítani:

- a./ Elképzelhetetlen ezeknek az impulzussorozatoknak a közvetlen emberi programozása.
- b./ Az impulzussorozat /nagy mennyisége miatt/ legfeljebb a legnagyobb információsűrűségű mágnesszalagon közvetíthető és tárolható.
- c./ A fentiekből következően az impulzussorozatot automatikusan kell előállítani és célszerű külső információhordozó nélkül továbbítani közvetlenül a motorokhoz. Ez a szerszámgép elektronikus vezérlő berendezésének minimális feladata.
- d./ Az egyes koordinátairányba mozgó impulzusok generálásának pillanatai között olyan függvénykapcsolatnak kell fennállnia, hogy az eredő szerszám-mozgás előírt türesem belül megfeleljen a munkadarab előírt konturjának. Lényegében tehát a munkadarab tervezett konturját lépcsős függvénnyel kell közelíteni. Ez történhet egy, vagy több lépésben a kontur jellegétől, valamint a technikai megoldástól függően.
- e./ Ha ezt az egész feladatot a NSz vezérlőberendezésén kívül oldják meg és pl. mágnesszalagon közlik, akkor külső interpolátoros megoldásról beszélünk. Ha az interpolációból valamely lépést a vezérlő berendezés old meg, akkor ezt belső interpolálásnak nevezzük.

f./ A sokféle lehetséges fokozat ismertetésétől eltekintve csak azt említjük meg, hogy a gyakorlatban - főleg az ESz-ek gazdaságos alkalmazásának elérésére - a legáltalánosabb az a megoldás, amikor a görbét egy külső interpolátor /az ESz pl./ egyenesszakaszokkal közelíti meg, és a belső interpolátor azután ezt dolgozza fel tovább. Az elektronikus vezérlőberendezés felé így tömörebb információt kell továbbítani, melyhez nem okvetlen szükséges mágnesszalag, hanem pl. papírszalag vagy lyukkártya is alkalmazható /a mágnesszalag ugyanis az üzemi körülmények - por, légnedvesség, stb. - miatt kevéssé ellenálló/.

Az általános gyakorlatban a külső interpolátor szerepét a számológép tölti be, melynek tehát a görbét egyenesszakaszok sorozatával kell közelítenie. A számológépet viszont programozni kell arra, hogy ezt a feladatot elvégezze.

Az egyenesszakaszokkal megközelített görbe /vagy eleve csak egyenesszakaszok megmunkálását igénylő munkadarab/ esetén a geometriai adatok segítségével egy u.n. szerszámgépi programot kell összeállítani, amely utasításokat tartalmaz egyéb szerszámgépi funkciókra és paraméterekre vonatkozóan is. Egyszerű esetben, vagy NSz hiányában ezt a programot lehet kézzel is összeállítani. Ezt nevezzük a NSz kézi programozásának. Rendszerint azonban ezt a munkát is az ESz-re bizzák /különösen akkor, ha a görbéközelítést is az ESz végzi/. Ezt nevezzük a NSz-ek számológépi programozásának.

Az NSz-ek számológéppel történő programozását automatikus programozásnak és szimbólikus programozásnak is nevezik. Ez a három kifejezés elvileg takarhatna három különböző eljárást. Ahelyett, hogy az ember esetleg sok ezer szerszámgépi utasításból álló programot kézzel készítené el egy tervezett munkadarabhoz /kézi programozás/, a program készítéséhez elektronikus

számológépet vesz igénybe /számológépes programozás/ de nem csupán a számolás megkönnyítésére segédeszközként, hanem úgy, hogy bizonyos, a munkadarabra és szerszámgépre vonatkozó minimális információ közlése után a számológép a programot automatikusan készítse el /automatikus programozás/. Ehhez természetesen a számológépre egy bonyolult programot kell készíteni, amely ezt a feladatot megoldja. Hogy az ember munkája még könnyebb legyen, a programozás könnyen megtanulható és megbízhatóbb legyen, a számológépi programot /amelyet szerszámgépprogramozó programnak nevezhetnénk/ úgy készítik, hogy a munkadarabra, szerszámgépre és a megmunkálás módjára vonatkozó nélkülözhetetlen információkat szimbólikus formában lehet felírni /szimbólikus programozás/ valamilyen szabályok szerint, az u.n. programozó nyelven.^{x/}

A programozó nyelveknél - a szóbanforgó, szerszámgépprogramozásra szolgáló szimbólikus nyelvénél ugyanúgy, mint bármely más célra alkalmas szimbólikus számológépi nyelvénél - az a törekvés, hogy az minél közelebb álljon a mindennapi emberi nyelvhez. Egy alapvető különbség azonban van: a programozó nyelvnek szigorúan egyértelműnek kell lennie. A hétköznapi nyelvhasználat ugyanis ezt a követelményt - még általában a tudományokban sem - nem teljesíti.

A szimbólikus programozó nyelvek másik közös sajátossága az a törekvés, hogy problémaorientáltak legyenek, ami azt jelenti, hogy a programozónak lényegében csak azt kell tudnia, hogy mi a programozandó feladat. Ne függjön a nyelv attól, hogy milyen számológépen, vagy milyen szerszámgépre használják. /Ne számológép, vagy esetünkben esetleg szerszámgéporientált legyen./

x/ A szerszámgép utasításrendszerét is nevezik a NSz programozó nyelvének, amely analóg egy ESz gépi nyelvével. Az itt tárgyalt szimbólikus programozó nyelv nem tévesztendő össze ezzel.

Ez a törekvés általában nem valósítható meg teljes mértékben. Ahogyan a szimbólikus számológépprogramozó nyelvek /ALGOL, FORTRAN, COBOL, stb./ számológépprogramozásra valóban kész formái /gépi reprezentánsai/ bizonyos mértékben mindig függenek az adott számológépektől, ugyanúgy a szimbólikus szerszámgépprogramozó nyelvek /APT, ADAPT, AUTOPROMT, AUTOSPOT, stb./ kis mértékben mindig függenek attól, milyen szerszámgépek tényleges programozására használják /sőt sok esetben az alkalmazott számológéptől is függhetnek minimálisan/. Általában a problémaorientált nyelv ezért egy alap "szókincset" és "nyelvtani", szintaktikus szabályokat tartalmaz. A szókincset és a szabályokat esetleg bővíteni kell ahhoz, hogy konkrét számológépen és szerszámgépre használható legyen. A nyelv ilyen reprezentálásával együtt jár magának a számológépi programnak az elkészítése, amelyet a szimbólikus nyelv fordító programjának, vagy értelmező programnak is neveznek.

A programozó rendszerek osztályozásának alapja.

Mivel a szóbanforgó programozó rendszerek problémaorientáltak, kézenfekvő, hogy azokat a programozható problémák, nevezetesen a megmunkálendő munkadarabok alapján osztályozzák. A szerszámgépeken megmunkálásra kerülő munkadarabok /alkatrészek/ olyan osztályozása azonban, amely alapul szolgálhatna a programozó rendszerek osztályozásához, sajnos nem alakult ki. Éppen fordítva, a programozó rendszerek osztályozásának és tervezésének szükséglete hívott életre egy bizonyos osztályzást a munkadarabokra vonatkozóan is. Mindkét osztályozás /a munkadaraboké és programozó rendszereké/ egyaránt a szerszámgépek osztályozására vezetett speciális szemszögből.

A programozó rendszerek osztályozásának alapja így minden esetben a programozható szerszámgépek valamilyen szempontból való osztályozása. A gyakorlat megmutatta, hogy az osztályzást

a szerszámfelület és a munkadarab relatív mozgásának bonyolultsága szerint célszerű végezni. A relatív mozgás ellenőrzött és ütemezett módon történő végzését nevezzük a szerszám vezérlésének. Az ütemezett mód az időben történő ellenőrzést, szabályozást jelenti.

A programozó rendszerek osztályozásánál általában azt veszik alapul, hogy a szerszámnak milyen fajta vezérlését lehet a rendszerben programozni.

A megmunkálásnál lejátszódó mozgásokat általában két komponensre bontják: főmozgásra és mellékmozgásra. A főmozgás lehet haladó mozgás is, de rendszerint forgómozgás, amelyet végezhet a munkadarab /esztergálás/ vagy a szerszám /furás, marás/. Feladata a forgácsleválasztás. A mellékmozgás a megmunkálás, forgácsleválasztás helyének változtatását célozza és rendszerint haladó mozgás. Az előtolás az effektív megmunkálás közbeni mellékmozgás. A szerszám és munkadarab relatív mozgását megmunkálás nélkül pozicionálásnak nevezzük. A mellékmozgás vezérlési szempontból a következő fajta lehet:

- a./ Egy adott pontba eljuttatja a szerszámot a munkadarabhoz viszonyítva; a pályára, a sebességre és egyéb körülményekre nézve nem kell tekintettel lenni; ilyenek a pozicionálás, vagy a szerszám visszavonulása egy megmunkálás véghelyzetéből, asztalelfordítás adott helyzetbe, stb.
- b./ Egy tengely mentén egyenletes mozgás megadott szakaszon; az egyenletes sebesség nagyságának és irányának jelentősége van; ez a mozgás mindig megmunkálás közben történik, pl. egyenes körhenger szakasz esztergálása, előírt mélységű lyuk furása /egy menetben/ marás egy szán mozgása közben, egyenletes sebességű adott szögelfordulás megmunkálás közben /ha ilyen lehetséges/.
- c./ Több tengely mentén egyidejű összehangolt /általában változó sebességű/ mozgás; a szerszám a munkadarabhoz viszo-

nyitva meghatározott pályán mozog vezérelt sebességkomponensek mellett; ilyenek a tengelyekkel derékszögtől eltérő szöget bezáró egyenes menti /pl. egyenletes/ mozgás, síkbeli, vagy térbeli görbe mentén történő mozgás.

A programozó rendszereket és vezérlési szempontból a szerszámgépeket aszerint lehet csoportosítani, hogy hány tengely mentén és milyen b/ vagy c/ típusu mozgás lehetséges. Az osztályozást az a/ típusu mozgások nem érintik /mivel azokat automatikusan vezérli a szerszámgép /szervomechanizmusa//.

A szerszámgépek szokásos osztályozása, melyet legtöbbször alapul vesznek programozó rendszerek osztályozásához a következő:

Pontvezérlésű szerszámgép, ami azt jelenti, hogy csak egy, a szerszámtengely menti b/ típusu mozgás lehetséges; szakaszvezérlésű szerszámgépnél legalább egy, a szerszámtengelyre merőleges tengely mentén lehetséges b/ típusu vezérlés, a folyamatos pályavezérlésnél pedig legalább két tengely /dimenzió/ viszonylatában lehetséges c/ típusu vezérlést megvalósítani. A pont-szakaszvezérlésű szerszámgépnél értelemszerűen mind a pont, mind a szakaszvezérlés kritériuma is teljesül.

Ez az osztályozás azonban a számológépi programozás szempontjából nem eléggé részletes és nem a legcélszerűbb.

A munkadarabok bonyolultságát, valamint megmunkálási műveletek sokféleségét és összetettségét, valamint a szerszám mozgás vezérlésének bonyolultságát nyilvánvalóan a mozgások b/ és c/ típusán, ill. a pont-, szakasz-, vagy pályavezérlésen túlmenően lényegesen befolyásolja az, hogy mindezek a mozgások hány tengely mentén valósíthatók meg. Geometriailag ez azt jelenti, hogy lényeges az is, a megmunkálás ill. vezérlés hány dimenziós és haladó, valamint forgó mozgások is lehetségesek-e?

A tengelyek száma szerint a szerszámgépeket többféle elv alapján osztályozzák. Általában egy mozgástengely értékelésébe

a lehetséges mozgás a/, b/, c/ típusát is belevonják, mert egyébként az osztályozás nem egyértelmű programozástechnikai szempontból. Egy szerszámgépen a mozgás két vagy három dimenzióban /2D ill. 3D mozgás ill. programozás/ történhet, azonban megint lényeges a mozgás minősége is. A szerszámgépek vezérlésében 2-3-4-5- tengelyű vezérlést különböztetnek meg. A 2 vagy 3 tengely a síkbeli vagy térbeli haladó mozgást teszi lehetővé /szánok/, míg a 4 és 5 tengelyek forgástengelyek, ha az asztal vagy orsó egy, vagy két tengely körül elfordulhat.

A programozó rendszerek osztályozása.

A szerszámgép vezérlések fenti osztályozási szempontjai alapján röviden ismertetjük a legismertebb programozó rendszereket felsorolva, hogy azok az egyes szempontok szerint milyen szerszámgépek vezérlésére alkalmasak.

AUTOPROPS /Automatic Programming of Positioning Systems/: kétdimenziós pont-vezérlésre alkalmas furó, vagy lyukasztógépen táblázatos, numerikus információk alapján. Az IBM cég dolgozta ki az IBM 1401 ESz-re.

AUTOSPOT /Automatic System for Positioning Tools/: kétdimenziós pont-szakaszvezérlésre alkalmas, egyik legelterjedtebben használt szimbólikus programozó nyelv. Az IBM dolgozta ki az 1620-as ESz-re 1962-ben. Mintegy 60-70 szót és jelet használ a nyelv.

AUTOMAP /Automatic Programming/: kétdimenziós folyamatos pályavezérlésre \mathcal{N} z-irányú pozicionálással alkalmas szimbólikus programozó nyelv. Az IBM 1620-ra dolgozták ki.

AUTOPROMT /Automatic Programming of Machine Tools/: háromdimenziós folyamatos pályavezérlésre alkalmas /5 tengelyig/

szimbólikus nyelv, amelyben egész felületi tartományok programozhatók egyetlen /vagy néhány/ "mondattal". Az IBM 7090 ESz-re, valamint más nagyteljesítményű számológépre készítették el. A szimbólikus nyelv 200-nál több szót használ és fordító programja mintegy 32000 számológépi utasításból áll. Ez tekinthető jelenleg az APT mellett /l.alább/ a leguniverzálisabb és leggazdagabb nyelvnek.

APT /Automatically Programmed Tool/ rendszer: több fokozata van. Az APT I pontvezérlésre, az APT II háromdimenziós görbék szerinti megmunkálás programozására és az APT III háromdimenziós felületi tartományok szerinti programozásra szolgáló rendszer ill. szimbólikus nyelv. A rendszert 1958-ig dolgozták ki a cambridgei műszaki egyetem /Massachusetts Institute of Technology, Cambridge/ Szervomechanizmus Laboratóriumában /ma: Elektronikus Rendszerek Laboratóriuma/ az USA légierővel együttműködve saját Whirwind I számológépre és később átkódolták az IBM 7090 és UNIVAC 1107 számológépekre is. A komplett APT rendszer 5 tengelyes folyamatos pályavezérlésű szerszámgépekre a legbonyolultabb alkatrészek /pl. repülőgéprészek/ programozására is kiválóan alkalmas /az AUTOPROMT mellett/ és állandó továbbfejlesztés alatt áll. A fejlesztés irányaira még visszatérünk.

ADAPT /Air Material Command Developed APT/ az APT II rendszer egyszerűsített változata, amely kétdimenziós folyamatos pályavezérlésű gépek programozására alkalmas és közepes számológépeken is alkalmazható. Eredetileg az IBM 7090 ESz-re programozták, de később sok más számológépre is. Mintegy 85-90 szót és 10 egyéb jelet használ a nyelv /a számjegyeken kívül/.

A felsoroltakon kívül még mintegy 40 féle különböző programozó rendszer ismeretes, melyek nagyrészt az Egyesült Államokban, néhányat pedig más országokban dolgoztak ki. Ezek közül pont-szakaszvezérlés programozására megemlítjük még példaként a CAMP /Compiler for Automatic Machine Programming/ prog-

ramot, melynek II fokozata az IBM 7090 gépekre 2-5 tengelyű pontvezérlésre alkalmas és az AUTOSPOT rugalmasabb változatának tekinthető és a PRONTO /Program for Numerical Tools/ programot, amely háromtengelyű pont-szakaszvezérléshez az IBM 704, GE 225 számológépekre kb. 50 szavas szótárral készült, Folyamatos pályavezérlésre készítették pl. a SYMPAC /Symbolic Programming for Automatic Controls/- Compiler rendszert a közepes UNIVAC UCT számológépekre és később az USS 80/90 gépre kétdimenziós vezérléshez. További ismertebb rendszerek még: WALDO, SPLIT, CLAM, stb.

A szimbólikus programozó rendszerek fejlődése.

Röviden felidézzük a NSz-ek és programozásuk fejlődését, különös tekintettel a szimbólikus programozó rendszerekre.

A szerszámgépek numerikus vezérlésének szükséglete és megvalósításának lehetősége az 1950-es évek elején merült fel. A problémával az USA repülőgépiparában ill. a légierő megbizásából a Massachusetts Institute of Technology /M.I.T./ Szervo-mechanizmus Laboratóriumában kezdtek foglalkozni. A kutatási programot és munkát "Automatikusan Programozott Szerszám Rendszerek" /Automatically Programmed Tool (APT) System/ elnevezés alatt végezték. A munkába a repülőgépipar mintegy 20 legjelentősebb gyártó cége is bekapcsolódott és az 1958-ig elkészített és széleskörűen kipróbált APT rendszer fejlesztésére mintegy 100 évnnyi kutatói munkát és több mint 4 millió dollárt fordítottak.^{x/} Az APT rendszer volt az első numerikus vezérlésű szerszámgép automatikus /számológéppel történő/ programozásra szolgáló program-rendszer és rögtön a legbonyolultabb háromdimenziós

x/ R.A. Thomas: The languages of tape. American Machinist, Special Report No. 545. Jan. 6., 1964.

munkadarabok programozására, hiszen a gyakorlati követelmények ilyen rendszert kívántak. Az APT rendszer egyben cégek és intézmények egy csoportját is jelenti, amelyek közösen résztvesznek a rendszer fejlesztésében, valamint joguk van a rendszer használatához. Ez a társaság 1959-től a civil iparág képviselőivel bővült. Az APT alapelvei szerint, vagy attól tudatosan eltérve egyéb cégek és intézmények, főleg a nagy számológépgyártó cégek speciális igényeknek megfelelően más rendszereket fejlesztettek ki. Így születtek az előző pontban felsorolt programozó rendszerek.

Az APT rendszer fejlesztése ma is folyik az Illinois Institute of Technology, Research Institute /IITRI/ védnöksége alatt, amely koordinálja a különféle erőfeszítéseket a rendszer speciális gyakorlati feladatkörökre való felhasználására és a rendszer általános fejlesztésére /melyre még visszatérünk/, valamint lehetőséget nyújt programozó szakemberek, segéderők képzésére, kíséreltetésére, stb.

A NSz-ek és automatikus programozásuk fejlesztése az amerikai példa nyomán Európában és másutt a világon később indult meg, de körülbelül hasonló fejlődés figyelhető meg Angliában 1953-tól, a Szovjetunióban 1954-től, Csehszlovákiában 1956-tól, Nyugat-Németországban 1957-től foglalkoznak a problémával.^{x/} Hazánkban napjainkban merült fel a probléma kényszerítő szükségességként. Nyugat-Európában - bár az USA-tól még mindig elmaradva - már széles körben alkalmaznak NSz-eket és programozásukra ESz-eket megfelelő programozó rendszerekkel, de a szocialista országokban széleskörű elterjedésről még nem beszélhetünk és az automatikus programozás igénye még csupán napjainkban merül fel. Az előrehaladás komoly akadálya a nem megfelelő ellátottság számológépekkel és az ebből következő felkészületlenség a programozás és programozó rendszerek területén.

x/ W.Simon: Der Entwicklungstand und die Zukunft numerisch gesteuerter Werkzeugmaschinen in Deutschland. IBM Nachrichten, Febr. 1965.

Az APT koncepció.

A NSz-ek automatikus programozásának alapvető problémája, amelyet az M.I.T. az APT rendszer kifejlesztésének keretében megoldott a számos elkerülhetetlen részletprobléma mellett lényegében az ember /munkadarabtervező konstruktőr és programozó/, az elektronikus számológép és a numerikus vezérlésű szerszámgép szerves, racionális és hatékony egységbe foglalása "nyelvek" és számológépi programok segítségével.

Az ember szempontjából a probléma lényege az, hogy minél egyszerűbben, minél természetesebbnek ható nyelven és minél több és bonyolultabb feladatot tudjon megfogalmazni és a korszerű gépgyártással járó munkák minél nagyobb hányadát tudja a gépekre automatikus végrehajtás céljából áthárítani. Első fokozatban az M.I.T.-ben azt tűzték feladatul - és a gyakorlatban a mai napig ezt tudják általában biztosítani -, hogy az ember a gyártással járó munkákból a műszaki terület nyelvéhez "hasonló" nyelven, a munkadarabok egyértelmű, pontos geometriai leírását és a legracionálisabb megmunkálási folyamat minimális technológiai utasításokkal történő megadását vállalja, a többi feladatot a számológép és szerszámgép automatikusan végzi el /az ember-készítette program számológép számára "érthető" formában való "leírása", kódolás és információhordozóra való elhelyezés természetesen még emberi munkát is igényel, de ez elenyésző/. Egy fejlettebb fokozatban, amelyről később még beszélünk, még ennek a munkának egy részét is a számológépre lehet bízni.

A számológép szempontjából többféleképpen vetődik fel probléma. Az egyik az, hogy milyen számológépet alkalmazzunk, ill. milyen áll rendelkezésre? Speciális számológépet /külső interpolátor/ vagy univerzális számológépet? Nagy teljesítményű számológépünk van, vagy kisebb? Miként lehet adott számológépet az ember-számológép-szerszámgép együttesbe a kívánt funkcióval bekapcsolni? Ez a probléma természetesen nem választható el az em-

bertől, ill. a szerszámgéptől. Az ember oldaláról nézve, a számológép memóriakapacitása és működési sebessége korlátozza a szimbólikus nyelv rugalmasságát, egyszerűségét és az emberre háruló feladatok redukálását. A szerszámgép oldaláról nézve a szerszámgépen megmunkálható munkadarabok geometriai bonyolultsága, valamint a megmunkálási módok iránti igénye befolyásolja azt, hogy milyen számológépeket lehet a programozáshoz racionálisan alkalmazni.

A számológép nagyságán kívül programozhatósága az, amely kihat az ember tevékenységére. Ezt a kihatást azonban már nem a szóbanforgó ember-számológép-szerszámgép rendszerben tekintjük, mert ez a mindennapi gyakorlatban a munkadarab programozó munkáját közvetlenül nem befolyásolja. Ez a szerves hármas létrehozásának és működése biztosításának munkájához, vagyis egy egyszerű előkészítéshez tartozik és ez volt az M.I.T. kutatóinak feladata, és ez ismétlődik újra és újra akkor, ha az ember-számológép-szerszámgép hármas utóbbi két tagjának valamelyike - sőt, az ember által használni kívánt programozó nyelv ugyszintén - megváltozik. Természetesen egy ilyen új hármas kialakításában már igen hatásosan felhasználhatók azok a tapasztalatok, melyek az APT és más rendszerek fejlesztésével kapcsolatban nyilvánosságra kerültek.

Az M.I.T.-ben az ember-számológép-szerszámgép hármas első és utolsó tagját különféle "tevékenységi szinteken" elképzelve a középső tagot /számológép/ igyekeztek egy konkrét univerzális számológéptől nagyrészt függetleníteni a következő módon: gondolatilag megalkottak egy APT számológépnek nevezett fiktív számológépet, az összes alapvető egységgel, amellyel egy tényleges, anyagi számológép rendelkezik: bemenő-vezérlő-aritmetikai-memória-kimenő egységekkel. Mindezeknek az egységeknek megadták a funkcióit, működésük kapcsolatát és algoritmusait úgy, hogy azok egy megfelelő univerzális számológépen szimulálhatók legyenek. Vagyis egy konkrét számológépem beprogramozva az egységek prog-

ramjai együtt egy "szimulációs számológépet" realizálnak, amely elvégzi azokat az alapszintű feladatokat, amelyet az ember-számológép-szerszám-gép középső tagjának adott programozó nyelv és szerszám-gépi utasításrendszer mellett el kell végeznie.

Ennek a szimulációs APT számológépnek a "nyelve" nem egyezik sem az ember használt programozó nyelvvel, sem a konkrét számológép valamely programozó nyelvvel, hanem egy célszerűen választott numerikus APT számológépi nyelv. Az APT számológép bemenő egységén csak ilyen nyelven írt utasítások mehetnek be, a vezérlő egysége csak ezeket tudja értelmezni és végrehajtani az aritmetikai egységével. Az APT számológép kimeneti nyelve szintén nem egyezik meg egy konkrét szerszám-gép nyelvvel sem. Az APT számológép bemeneti nyelve olyan, amelyre az emberi programozó nyelven felírt munkadarabprogramok /"forrásprogramok"/ könnyen fordíthatók minden konkrét számológépen /könnyű fordító-programot, translatort készíteni/ és kimeneti nyelve pedig olyan, hogy az - ismét bármely számológéppel - könnyen fordítható le adott szerszám-gép konkrét kódrendszerébe /könnyű u.n. post-processort készíteni/.

Az ember és a szerszám-gép említett "tevékenységi szintjei", amelyekhez az APT számológépet illesztik az emberi oldalon: /1/ szimbólikus nyelven való programozása készre megtervezett munkadaraboknak előre kiválasztott technológiával és szerszám-gépre, és /2/ a számológéppel való "társalgás" útján való munkadarabtervezés és konstrukció csupán a különleges igények közlésével /erre még visszatérünk/, a szerszám-gépi oldalon pedig /1/ pont-szakaszvezérlés, ill. térbeli pontok segítségével való programozhatóság, /2/ folyamatos pályavezérlés térgörbék szerinti programozhatósággal és /3/ folyamatos pályavezérlés térbeli felületek szerinti programozhatósággal.

A szimulációs számológép módszerrel az APT problémája lényegében az APT számológépek "megépítésére", programok elkészí-

tésére redukálódott. Az M.I.T.-ben az első stádiumban az emberi tevékenységet az említett /1/ szinten tétélezték fel. A szer-
számgepi oldalon szinte egyidejüleg mindhárom szintet figyelem-
bevéve háromféle APT számológép kidolgozásába kezdtek. Először
egy APT I számológépet programoztak, amely azonban csak kísérle-
ti célokat szolgált és előkészítette a későbbi "nagyobb telje-
sítőmennyű" APT II, majd APT III számológépek programozását. Az
APT II szimulációs számológép lényegében u.n. kanonikus alakban
megadott térbeli görbék interpolálására, vagyis előírt türezen
belül egyenesszakaszokkal való megközelítésére képes. A gyakor-
lati használatban főként síkbeli görbékkel határolt munkadara-
bok megmunkálására, vagyis kétdimenziós pályavezérlésre alkal-
mazzák. Az APT III számológép kanonikus alakban megadott felü-
letek határolta felületi tartományok megmunkálásának automatikus
programozására képes.

Az emberi tevékenység említett /2/ szintjéhez tervezett
számológépet ill. az így nyert IITRI-ban ma is fejlesztés alatt
álló APT rendszert APT IV rendszernek nevezték el. Egyébként az
APT I-II-III-IV rendszerek mindegyike az előző fokozat kibővíté-
se, vagyis bármely fokozat használata feleslegessé teszi az elő-
ző használatát, mert a megelőző fokozatok minden funkcióját képes
ellátni.

Az ember-számológép-szerszámgep hármast most már a szer-
számgep oldaláról nézve lényegében arra volt szükség, hogy adott NSz-
et bekapcsoljanak az APT rendszerbe, mégpedig úgy, hogy egy szá-
mológépi program /postprocessor/ segítségével a megfelelő APT
számológépfokozat outputjait lefordítsák, konvertálják az illető
szerszámgep "nyelvére". Erre a kérdésre a fordító program prob-
lémájánál még visszatérünk, itt csupán annyit említünk, hogy
természetesen egy számológép számos szerszámgepet kiszolgálhat
egyidejüleg, mivel feladatának elvégzése összehasonlíthatatlanul
rövidebb időt vesz igénybe, mint a szerszámgep működési ideje.

Ezzel kapcsolatban hangsúlyozzuk, hogy az ember-számológép-szerszám-gép hármas általában /1/ több programozó ember és több szerszám-gép együttműködését jelenti egy számológéppel /2/ a hármas minden tagja elláthat emellett a kooperatív munkától különböző feladatokat is /ez főleg a számológépnél jelentős/ és /3/ a hármas tagjai időben és térben teljesen különválva dolgozhatnak; tehát nem szinkron és helyhez kötötten kell együttműködniük /természetesen először az embernek kell programozni, hogy azt a számológép lefordíthassa és ezután a szerszám-gép a programozott munkadarabot elkészíthesse./

A programozó nyelvek.

A NSz-ek programozására tervezett szimbólikus programozó nyelvek egyszerűen közönséges tagjai a ma már mind nagyobb családot alkotó probléma-orientált számológépi nyelveknek, melyekre példák a COGO, STRESS, SIMON, CSL, stb., melyek nem szerszám-gépvezérlésre szolgálnak, és a már általunk korábban említett APT, ADAPT, AUTOMAP, AUTOPROMT, AUTOSPOT, stb., nyelvek, melyek szerszám-gépvezérlésre készültek. Ezek a nyelvek némileg különböznek az ALGOL, FORTRAN, COBOL, stb. nyelvektől, amelyeket program-orientált nyelveknek neveznek, mert a számológépprogramozás egyszerűsítését és szimbolizálását célozzák a programozandó probléma témakörétől függetlenül. Méginkább különböznek a fenti nyelvek az u.n. (számoló)géporientált szimbólikus nyelvektől /autókód, "assembly" nyelvek/, mint pl. az IBM gépeknél a SPS (Symbolic Programming System) AUTOCODER, az Elliott gépeknél pl. a NEAT, stb., amelyek csupán az adott számológépre történő programozáshoz használhatók.

Végül az összehasonlítás láncolatának másik végén állnak esetünkben a szerszám-gépi nyelvek, melyek ugyanugy gépi kódok ill. gépi utasításrendszerek numerikus /ill. alfanumerikus/ formában, mint a számológépi nyelvek ill. utasításrendszerek.

A szimbolikus nyelvekben egyaránt az a közös törekvés tükröződik, hogy a problémák programját valamely elektronikus rendszer vezérlésével történő megoldása céljából az emberi nyelvhez minél közelebb álló, minél természetesebbnek ható, könnyen megtanulható és használható formában lehessen megfogalmazni és leírni. Az így leírt programok kódolását és feldolgozását azután speciális elektronikus /és részben mechanikai/ berendezések bevonásával végezzük a lehető legautomatikusabb fokon.

A problémaorientált nyelvek és programrendszerek fejlődése még messzebb vezet, amely már az ember és a számítógép közti "eszmecsere" lehetőségét teremti meg az információközlési módok széles skáláján /adott esetekben, a legcélszerűbbet megválasztva: írógép, rajzoló és vetítő berendezések, esetleg akusztikai berendezések, stb./. A nagyobb IBM számítógépekre ilyen nyelvek pl. a COGO és a STRESS, valamint az APT még fejlesztés alatt álló fejlettebb fokozata /APT IV/.

A szerszámprogramozó szimbolikus nyelvek általában strukturálisan az ALGOL-hoz, vagy a FORTRAN-hoz hasonlíthatók. Ismeretes pl. már az APT III nyelv ALGOL 60 riporthoz ^{x/} hasonló szintaktikus leírása is. ^{xx/}

A szóbanforgó nyelveken komplett munkadarabok készítéséhez szükséges információk írhatók le. Ezek együttesét munkadarab-
programoknak nevezzük. Egy munkaprogram kijelentésekből, mondatokból áll, amelyek a nyelv szabályainak megfelelően csoportosított "szavakat", írásjeleket, paramétereket, számokat tartalmaznak előírt szabály szerinti sorrendben. A szavak egy része

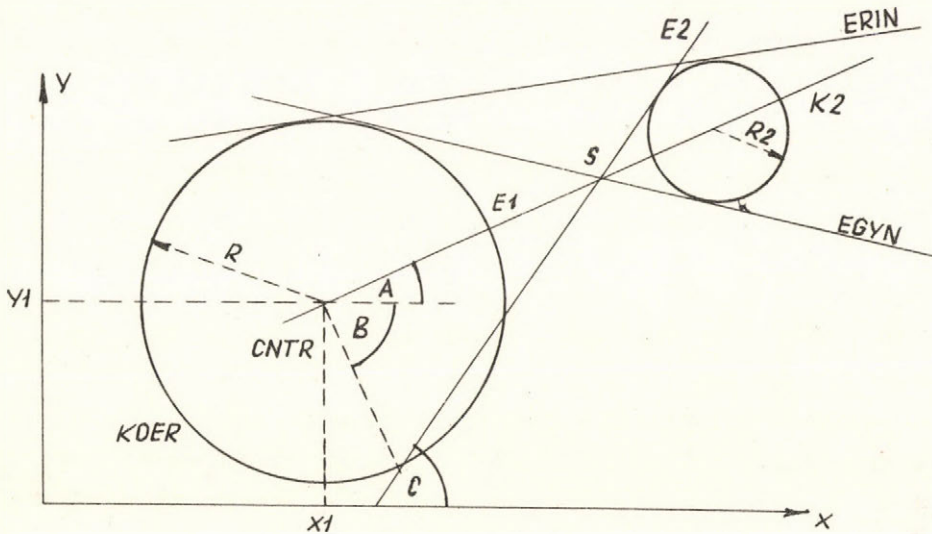
x/ J.W. Bachus és mások: Revised Report on the Algorithmic Language ALGOL 60 /Edited by P. Naur/ Comm. ACM 6 nov. 1. /1963/

xx/ S.A. Brown és mások: A Description of the APT Language. Comm. ACM 6 nov. 11. /1963/

u.n. szótári szó, melyeknek a nyelvben meghatározott, rögzített jelentésük van, más részük azonosító, melyek adott eljárás szerint definiálhatók a nyelv alapjelei-ből. Az alapjelek meghatározott betűk /általában az ABC nagybetűi/, az arab számjegyek és néhány egyéb jel.

Különbéféle információfajták leírására rendszerint más-más mondattípusok használandók, amelyek között általában az alábbi három mindig szerepel:

a/ Geometriai definíciók. Ilyen mondatok keretében írhatók le geometriailag a megmunkálendő alkatrész azon részletei, ahol effektív megmunkálásnak kell történnie, valamint azok egymáshoz, vagy egy koordinátarendszerhez viszonyított helyzetei. Az alkatrész mindig egy hozzá kötött derékszögű /esetleg polár/ koordinátarendszerben írható le és a nyelv lehetőségeinek megfelelő geometriai alapelemeket, összefüggéseket lehet felhasználni a leírásban. Például az APT III nyelvben pont, egyenes, kör, ellipszis, hiperbola, általános kúpszelet, sík, henger, kup, gömb, általános másodrendű felület adhatók meg előírt formában néhány paraméterrel, vagy egymáshoz viszonyított helyzetükkel, sőt még táblázatosan megadott síkmetszetű hengert is lehet használni. A geometriai definíciók általában egymásba is ágyazhatók, valamint beépíthetők más mondatokba, utasításokba. Nagyrészüket azonban általában a program elején adjuk meg és a későbbiekben név szerint hivatkozunk rájuk. Tipikus definiáló \mathcal{G}^1 mondatokat mutatunk be APT III nyelven az alábbi példán:



- 0/ REMARK / GEOMETRIAI DEFINÍCIÓK
- 1/ CNTR = POINT / X1, Y1
- 2/ KOER = CIRCLE / CENTER, CNTR, RADIUS, R
- 3/ E1 = LINE / CNTR, (POINT / KOER, ATANGL, A)
- 4/ E2 = LINE / (POINT / KOER, ATANGL, B), ATANGL, C
- 5/ EGYN = LINE / (S = POINT / INTOF, E 1, E 2), RIGHT, TANTO, KOER
- 6/ K2 = CIRCLE / EGYN, YLARGE, E2, YSMALL, RADIUS, R2
- 7/ ERIN = LINE / LEFT, TANTO, KOER, LEFT, TANTO, K2

A programrészben aláhúztuk az előforduló APT III szótári szavakat. A 0/ címkéjű megjegyzés /REMARK/ hatástalan. Az 1/ definiálja a CNTR-nek elnevezett pontot, melynek koordinátái /X1, Y1/ változópár értéke /a z-koordinátáról feltételezzük, hogy előzőleg egyszer s mindenkorra adott, vagy pedig 0, amikor is nem kell feltüntetni./ A 2/ definíció egy olyan KOER nevű kört definiál, melynek középpontja az előzőleg megadott pont, sugara pedig R értéke. A 3/ alatt E1 egyenest két pontjával definiáljuk, melyek egyike a CNTR pont, a másikat itt definiáljuk még pedig a KOER nevű körön az A (x-tengellyel bezárt) szöghöz /polárszög/

tartozó pontként. A definíciót zárójelbe kell tenni, ha az nem önállóan szerepel, egy mondatként.

A 4/ alatti E2 egyenes a KOER kör B szöghöz tartozó pontján átmenő C irányszögű egyenes. Az EGYN egyenes átmegey azon az S-nek elnevezett ponton, amely E 1 és E 2 egyeneseknek is metszéspontja /INTOF/ és jobbról érinti /RIGHT, TANTO/ a KOER kört. A 6/ alatti K2 nevű kör az EGYN és E2 egyenesek egyik közös érintője R2 sugárral mégpedig az, amelyik EGYN-től a nagyobb Y értékek /YLARGE/ és E2-től a kisebb Y értékek /YSMALL/ felé esik. Végül az ERIN egyenes a két kör közös érintője mégpedig KOER-től a K2 felé nézve mindkét kört balról érinti /LEFT, TANTO/.

b/ Szerszám- és egyéb definíciók. Vannak nyelvek, ahol a szerszámokat szigorúan meghatározott módon a program elején kell definiálni és megszámozni, amely számokkal később az egyes szerszámokra lehet hivatkozni. Más nyelveknél a szerszámokat ott definiálják, ahol a programban használatbavételére sor kerül. Az APT rendszerben is az utóbbi módszert követik. A szerszám-definiáló mondatban megadjuk a szerszám mindazon lényeges adatait, amelyek az APT számológép számításainak biztosításához és a postprocessor részére szükségesek. Pl.: CUTTER /d, r azt jelenti, hogy a szerszám egy r sugárral lekerekített élű d átmérőjű hengeres maró.

A szerszámokon kívül a program elején definiálni kell a geometriai türeést, amely a görbék egyenesszakaszokkal való közelítése közben még megengedhető, az előtolás sebességét, a hűtés módját stb. Ezek a definíciók mindaddig érvényesek, amíg nem változtatjuk meg. Pl.: a SPINDL / 1800, RPM, LOW definícióval /mely utasítás is az orsóforgás elindítására/ meghatározott 1800 fordulat pro perces /RPM/ fordulatszám, mely az alacsony /LOW/fordulatszám tartományba esik, mindaddig érvényes, amíg más hasonló előírás, vagy SPINDL / OFF (orsóforgás leállítása)

előírás elő nem fordult. A mondatoknak ilyen típusához sorolhatjuk még a munkadarab és a szerszám gép azonosításához szükséges adatok és a REMARK megjegyzések megadását; valamint a számolás, vagy mozgás módjára, vagy megszakítására szolgáló utasításokat (STOP, DELAY, END, FINI, CUT, DNTCUT, MULTAX, 2DCAIC, 3DCAIC, stb.)

c/ Mozgási utasítások. A munkadarabprogram nagy részét képezik általában azok az utasítások, amelyek ténylegesen a szerszám mozgását /irányát, sorrendjét, stb./ írják elő a geometriailag definiált munkadarab mentén, hogy a kívánt megmunkálások megvalósuljanak. Sok nyelvnél az ilyen utasításokat megszakítják, vagy teljesen egybeépülnek azokkal a geometriai definíciók, vagy egyéb utasítások /pl. aritmetikai számítások, logikai ⁹ döntések, stb./. A nyelv (ill. program) bonyolultságától függ, hogy egyetlen mozgási utasítással milyen összetett és komplikált mozgássorozatot lehet kiváltani. Pl. az AUTOSPOT-nál egyenesszakasz menti egyenletes mozgásokat, az APT II-nél térgörbeszakaszok menti mozgásokat és pl. az AUTOPROMT-nál felületi tartományok szisztematikus bejárását lehet egyetlen mondatral előírni. Természetesen az előírt utat és határait egyértelműen kell megadnunk.

Az APT III-ban tipikus mozgási utasításokat sorolunk fel.
OO) FROM /PT
INDIRV / VEK
GO/TO, LAP

E három utasítás a mozgási utasítások első három utasításaként fordulhat elő, és pozicionálásra és elindításra szolgál. Azt jelenti, hogy PT /előzőleg megadott/ pontból elindulva VEK /előzőleg definiált/ vektor irányában menjen (GO), addig míg a szerszám érintkezésbe nem kerül a LAP /előzőleg definiált/ felülettel. Hasonló elindító utasítássorozatok még:

01) FROM /PT 1
INDIRP /PT 2
GO / OFFSET, FLIT

amely azt jelenti, hogy PT 1 pontból a szerszám menjen egyenes vonalban abba a helyzetbe, amíg a FLIT felületet nem érinti a PT 1, PT 2 szakasznak a felülettel való metszéspontjánál.

02) FROM / PT
GO / TO, F 1, ON, F 2, PAST, F 3

ami azt jelenti, hogy PT pontból a legrövidebb uton olyan helyzetbe menjen a szerszám, hogy az F 1, F 2 és F 3 felületek metszéspontjában /ha ilyen nincs, hibajelzés!/ az F 1 felületet érintse az innenső /PT felőli/ oldalon /TO/, az F 2 felület normálisa essen egybe a szerszámtengellyel, az F 3 felületet pedig érintse a tulsó oldalról. Ez a legspeciálisabb helymeghatározások egyike az APT III_ban. Geometriailag $3 \times 3 \times 3 = 27$ térbeli helyzet képzelhető el. Mivel azonban a szerszámtengely a munkadarab felületére /közelítőleg/ merőlegesen kell legyen, ezért ahhoz csak TO előírás lehetséges és így az összes lehetőség 9-re csökken.

<u>F2</u>	<u>F3</u>
TO	TO
TO	ON
TO	PAST
ON	TO
ON	ON
ON	PAST
PAST	TO
PAST	ON
PAST	PAST

03) FROM /PT 1
GODITA / X 1, X1, Z1

04) GOTO /PT2

Itt a 03 utasítás után egyszerűen X 1, Y 1 és Z 1 koordinátaértékeknek megfelelő /növekményes/ elmozdulás történik, a 04)-nél pedig a pillanatnyi /álló!/ helyzetből a PT2 pontba történő elmozdulás (és ott megállás) történik.

10) GORGT / GOERBE

Az előző mozgáshoz képest jobbra kanyarodva a GOERBE vonal mentén a következő utasításbeli görbével való metszéspontig. Az ehhez hasonló utasítások folyamatos mozgásra szolgálnak.

20) GO LEFT /GOERBE, PAST, HATAAR

Az előző mozgáshoz képest balra kanyarodva a GOERBE mentén, amíg a HATAAR vonallal való metszés pontnál a szerszám éppen tulmegy (PAST) a HATAAR-on.

Ez egy mozgássorozat befejezésére szolgál.

Az említett három mondat típusokon, illetve kijelentéstípusokon kívül sok nyelven még egyéb kijelentések is megengedettek. Pl. aritmetikai kifejezések, illetve értékadó utasítások, ciklusok, összetett utasítások, feltételes és vezérlésátadó utasítások. Az APT III-ban például még a következők használhatók:

d/ Egyéb utasítások. A legfontosabb talán az u.n. makro-utasítás lehetősége, amely több nyelvben is biztosított. Pl. a

MENET = MACRO / P 1, P2, ...

(utasítások)

TERMAC

utasítássorozat egy MENET nevű összetett utasítás, amelynek P 1, P 2, stb. megadott jelzésű paraméterei vannak.

A definíció után a makro utasítás

CALL / MENET, P 1 = A, P 2 = B,

formában akárhol és akárhányszor hívható és a végrehajtás P 1 = A, P 2 = B, stb. értékekkel történik. Az alábbi utasítássorozat egy ciklusutasítás az APT III-ban:

LOOPST

IND) CALL / MENET, P 1 = A, P 2 = B

IF(MAX-B) VEEGE, VEEGE, TOVAABB

TOVAABB) A = A + . 6

B = B + . 6

JUMPTO / IND

VEEGE) GO TO / HAZA

LOOPND

ez az utasítás példázza az APT III-ban a feltételes átirányító (IF), az értékadó (A=A+.6) és feltétlen vezérátadó (JUMPTO) utasításokat is. A LOOPST (loop start) és LOOPND (loop end) mondatok csupán a ciklus elejét és végét jelzik, egyébként hatástalannak.

A szótári szavak általában egy, vagy két közönséges szó rövidítéséből állnak, amelyek megkönnyítik azok megjegyzését és használatát. Ezért a szerszámgépprogramozó szimbolikus nyelvek szinte természetes kibővítésként hatnak azok számára, akik egyébként a munkadarabtervezés, de különösen programozás területén járatosak /mnemotechnikai szóképzés/. Természetesen a nyelvek a gyártó ország nyelvére épülnek. Így napjainkban a nyelvek majdnem kizárólag angol-szerű szótárakat tartalmaznak.

Az azonosítóknál általában korlátozva van azok hossza /öt, vagy hat betű ill. szám/ és ki vannak zárva a nyelv különleges jelei. Pl. az APT III-ban bármely, legfeljebb hat betűből és

számból álló jelsorozat lehet azonosító, kivéve a szótári szavakat /melyek itt szintén eleget tesznek e korlátozásnak/. Az AUTOPROMT nyelvben azonban nincs korlátozva az azonosítók hossza.

Az APT III nyelvében pl. lehetőség van a szótári szavak szinonimáinak definiálására is. Itt, ha egy szótári szó a programban igen sokszor szerepelne és nehézkes, vagy egy másik szót kifejezőbbnek tartanánk, akkor mód van bármely azonosítót használni helyette. Pl. a

SYN / ÉR, TANTO, METSZ, CROSS, PÁRH, PARLEL

mondat azt jelenti, hogy a programban a továbbiakban a TANTO (tangent to=érintő) helyett ÉR, a CROSS (metsző) helyett METSZ és a PARLEL (parallel=párhuzamos) helyett PÁRH használható.

A programozó program.

Bármely ember-számológép-szerszámgép rendszer működésének feltétele a számológépre megírt megfelelő program. Tekintve, hogy az ember a munkadarabot valamilyen /szimbolikus/ nyelven írja le és a szerszámgép az utasításokat valamilyen más nyelven "érti meg", a számológépet felfoghatjuk, mint tolmácsot, aki fordít az ember használt nyelvről a szerszámgép "használt" nyelvre. A számológép e feladatát egy program révén képes betölteni, amelyet programozó programnak, vagy fordító programnak nevezhetünk.

A programozó programokat általában funkcionálisan különböző részekre osztják, melyek sok esetben időben is egymás után "futnak" a számológépen és felhasználják egymás eredményeit. Ennek oka többféle:

1) A teljes program általában nagy terjedelmű lenne, mely a számológépek korlátozott operatív memóriája következtében nem

férne el a memóriában.

2) Az igények változatai esetében a program részenként cserélhető, vagy fejleszthető, bővíthető.

3) Programozástechnikai okok:

- a) Részenként általánosabban írható meg és más célra is használható.
- b) Nagyobb a szimbolikus programelvek (ALGOL, FORTRAN) használhatósága.
- c) Könnyebb a megírás, kipróbálás, ellenőrzés és javítás.

Az APT rendszerrel már utaltunk arra, hogy a program fő részét egy "szimulációs APT számológép" képezi, amely tulajdonképpen numerikusan kódolt bonyolult számítási és mozgási utasításokat értelmez, és "bont le" egyszerű mozgási lépéseket leíró utasításokká, valamint néhány egyéb információt épít be megfelelő helyekre a mozgási utasítások közé. Az APT számológép mellett azonban az APT rendszer tartalmaz még egy translatort és preprocessorot (fordító és előfeldolgozó program), amely az APT számológép előtt "működik" és a következőt végzi:

a) A translator a szimbolikus APT-nyelven megírt program utasításait átkódolja numerikus utasításokká és az összetett utasításokat (makrok, ciklusutasítások, stb.) lebontja alaputasítások sorozatára.

b) A preprocessor megoldja a definíciókban előforduló geometriai feladatokat és az előforduló geometriai elemeket kanonikus formában fejezi ki.

Az APT számológép utasításaiban már minden geometriai elemnek kanonikus formában kell előfordulnia. A kanonikus formát tulajdonképpen az definiálja, hogy milyen paraméterekkel kell a szimulációs számológép részére előállítani. Például egy kört az APT nyelvben többféleképpen lehet definiálni:

középpontjával és sugarával
középpontjával és érintő egyenesével, vagy körével
két érintő egyenesével, vagy körével és sugarával, stb.

de az APT számológép számára mindig (X, Y, R) alakban kell megadni, ahol az első két szám a középpont koordinátái, a harmadik a sugár.

Azt is említettük már, hogy az APT számológép "kimenetét" egyetlen konkrét szerszámgép sem tudja megérteni, tehát a kiadott, lefordított és elemi lépésekre bontott program nem alkalmas semmilyen szerszámgép közvetlen vezérlésére. Ugyanakkor a produkált program olyan, hogy viszonylag egyszerűen fordítható bármely konkrét szerszámgép nyelvére (a szerszámgépek meghatározott kategóriájában természetesen). Ezt a fordítást és a vezérlő program elkészítését minden konkrét szerszámgépre, vagy azok megadott választékán belül bármelyikre egy külön program, u.n. postprocessor (utófeldolgozó) végzi. Tehát pl. ha a programozott szerszámgépek listája egy géppel bővül, elég a postprocessort (ha egy öszszevont van) kiegészíteni, vagy azok sorozatát (ha minden géphez másik van) kibővíteni. Az egész programrendszer többi eleme változatlan marad.

Általában a postprocessort megelőző programrészt processornak is nevezik, de néha csak az APT számológépet, vagy azzal analóg feladatu programrészt nevezik processornak. A különféle programrészek rendszerint egymás után működnek, de majdnem kivétel nélkül külön működik a postprocessor rész. Természetesen ismeretek speciális egyedi programok is, amelyek egy adott programozó nyelvről egy adott számológépen egy adott szerszámgépre fordítanak. Ezek talán az adott relációban sokkal gazdaságosabbak (kisebb a fordítóprogram és gyorsabban fordít), ezzel szemben teljesen specifikusak. A törekvés általában eddig az általánosabb programrendszerek készítése volt, sőt azoknak minél inkább univerzális számológép-programozó nyelven való megírása. Pl. az APT rendszer

legfőbb részei teljesen FORTRAN nyelven is meg vannak írva.

Bárhogyan és bármely számológépre történjek is a program megírása, a munka nagysága és bonyolultsága komplikáltabb programozó rendszereknél vetekszik egy ALGOL-fordítóprogram megírásával. A NSz-ek ESz-el történő programozása általában legalább közepes nagyságu számológépet igényel és egy programozó program elkészítése nagy erőfeszítést egy kollektiva részéről, melynek tevékeny tagja kell legyen elengedhetetlenül egy műszaki szakember, vagy számos szerszám-gép-programozó, fordítóprogram készítésben járatos matematikusok és a számológép közvetlen kiszolgálói és használói. A fordítóprogram készítőinek át kell tekinteni a lehetséges előforduló munkadarabok, megmunkálási módok, szerszám-gépfunkciók, szerszám-gép-konstrukciók és szerszám-gépi kódok széles skáláját, hogy egy némiképp általános, megbízható és a befektetett munkával arányos haszonnal használható fordító-program-rendszert alkothassanak meg. Feltétlenül nagy mennyiségű számológépi üzemidőt kell biztosítani a munkához és a valószínűleg nélkülözhetetlen a szerszám-gépek többé-kevésbé széles skálájából a jellegzetes és szélsőséges példányok reprezentálása és kísérleti programokkal azok működtetési lehetőségének biztosítása.

Egy számológép és egy szerszám-gép esetén csupán egy egyedi programot lehet készíteni, amely nem minden körülmények között felel meg a célnak. Hangsúlyozhatjuk azt a tényt is, hogy sokan vitatják univerzális szerszám-gépprogramozó rendszerek kifejlesztésének eredményességét és gazdaságosságát. Helyettük az egyedi programozó rendszereket, minden szerszám-géphez külön megírt programokat és nyelveket tartanak hasznosnak. Szerintük az APT koncepció és a többi univerzális rendszerek nem váltották be a hozzájuk fűzött reményeket és hasznuk nem áll arányban kidolgozási költségeikkel.^{x/}

x/ F.A. Peters: A simple handle for a simple tool. The Computer Bulletin June 1965.

A számológépes programozás perspektívái

Az előzőkben több helyen utaltunk már arra, hogy az ESz felhasználása a NSz-ek üzemeltetésében nem zárul le azzal, hogy a NSz-ek programozásának szimbolizálását, illetve a munkadarabok szerszámgépi programja elkészítésének automatizálását lehetővé teszik.

A számológépek felhasználása a teljes alkatrészgyártás, sőt a tervezéssel járó munkák nagy részének teljes automatizálását is lehetővé teszi.

Ma még nagyrészt a számológépeken olyan programokat használnak szélesebb gyakorlatban, amelyek a kész, végleges megtervezett alkatrész, munkadarab komplett, szimbolikus nyelven leírt és információhordozón kódolt programját követelik meg.

A programoknak tartalmazniuk kell a felhasznált szerszámgépre, a szerszámokra, előtolásokra, egyéb segédfunkciókra való utalást is, megfelelően ütemezve és meglehetősen szigorú szabályok szerint.

Mi az, amelyet a számológép még ebből a munkából az embertől átvehet? Amely révén az embert az APT-vel kapcsolatban már említett b) tevékenységi szinten tételezhetjük fel az ember-számológép-szerszámgép együttesben? Ezt talán a következőkben foglalhatjuk össze:

- 1) Az embert mentesíteni a programirástól és kódolással járó manuális munkától.
- 2) Mentесíteni magától a programnak a megszerkesztésétől a meghatározott műszaki és technológiai adatok alapján (pl. műszaki rajz és technológiai követelmények alapján bizonyos lehetőségek felkínálása mellett) a szerszámgép, szerszám és nyersanyagok tekintetében.
- 3) A különféle megoldási lehetőségek közül az optimális, vagy egyedül helyes megoldások kiválasztása. Pl. a lehetséges szer-

számgépek, szerszámok, hűtőanyagok, stb. közül a megfelelő kiválasztása, a megfelelő követelmények megadása után a geometriai tőrészek, esetleg a munkadarab alapanyagának kiválasztása, a megmunkálások fokozatainak és sorrendjének összeállítása, stb.

- 4) A munkadarab tervezéssel, konstruálással járó számítási és mérlegelési, valamint precizirozási munkák, továbbá átmeneti tervek különféle módokon való szemléltetésének elvégzése.
- 5) Végül "alkotó" közreműködés a tervezésben azáltal, hogy elképzéseket eleméz, interpretál, vagy "kritizál" és "véleményeket" közöl, vagy "ellenjavaslatokat tesz" az ember által szolgáltatott tervvel, vagy elképzéssel kapcsolatban.

Az elmondott funkciók egy tényleges rendszerben fordított sorrendben játszódhatnak le, ha egyáltalán a számológép végzi azokat: először az 5) és 4) funkció, majd a 3) és végül a 2) és 1) funkció. Az 5) és 4) funkciók az ember aktív közreműködése és befolyása alatt folynak, így azokról feltétlenül tájékozott. A 3) funkció eredményéről célszerű, ha tájékozódik és a lehetőséget erre biztosítani kell. A 2) és 1) funkció az embert, a tervezőt tulajdonképpen már ugyanugy nem érdekli, mint ahogy egy mai rendszernél maga a számológépi fordítás lefolyása nem érdekli a munkadarab-programozót.

Természetesen az ellenőrzési lehetőséget az ember számára a fontosabb lépéseknél biztosítani kell a most felvázolt tökéletesebb rendszernél ugyanugy, mint a mai programozó programoknál. Ennek célja főként az esetleges hibák időben történő felderítése és a szükséges közbelépés lehetőségének biztosítása, hogy a hibát, ha már fellépett, minél "olcsóbban fizessük meg", vagy küszöbölhessük ki.

A már ma működő rendszereknél mindennapos a rajzológép vagy fényernyő és megfelelő program, amely a programozott munkadarab kívánt részletének (vagy az egésznek) kívánt méretben és néző-

pontból való megjelenítését biztosítja vizuálisan. Kivánságra a számológép lehetővé teszi a szerszám utjának végigkövetését a munkadarab mentén a megmunkáláskor, mielőtt akár a munkadarab konkrét szerszámgépre szerkesztett programja elkészült volna, de mindenesetre mielőtt a szerszámgépet üzemeltettük volna a szóbanforgó munkadarab elkészítése érdekében. Ez a módszer általában gazdaságosabb, mint a szerszámgép próbaüzemeltetése a kész programmal (pl. poliuretánhab megmunkálásával), mert a számológép perifériális berendezései gyorsabban és olcsóbban üzemelnek.

A ma meglévő programozó programok is tartalmaznak már u.n. diagnosztika részt, amely a program többi részének működésével párhuzamosan állandó ellenőrzéseket végez és "jelentéseket", jelzéseket közöl, szükség esetén vész- vagy hibajelzést ad és le is állítja a program további futását.

Természetes, hogy mind a szemléltető program, mind a diagnosztika program hatáskörét és működését ki kell terjeszteni az egész rendszer fejlesztésével párhuzamosan. A két program szerepe keveredhet és a két program egybe is olvadhat, hiszen fő szerepe tulajdonképpen az ember tájékoztatása: akár az ember közvetlen, vagy közvetett kérdésére válaszolva (értékelések, választások, feltételek közlése), akár pedig rendkívüli helyzetek vagy hibák, tévedések, esetleg üzemzavar vagy matematikai szélsőségek, abszurd helyzetek jelzésével, közlésével. Az utóbbiaknak feltehetően az egész rendszer tökéletesedésével párhuzamosan mind ritkábbaknak kell lenniük, így e szerep is kisebb jelentőségűvé válik.

Szinte felesleges említeni, hogy az ilyen "tökéletes" rendszerek programjai csak nagy teljesítményű ESz-ekre készíthetők el. Azt azonban hangsúlyozni kell, hogy a problémák nagy része nem szorítkozik kizárólag a szerszámgépek programozására és a gyártandó munkadarabok tervezésére és így az egész fejlesztést

nem szabad elszigetelni a számítógépek egyéb területeken való mind hatásosabb és racionálisabb felhasználhatóságának kutatásától. Talán nem is szabad a vázolt fejlesztéssel kapcsolatban egy lépést sem tovább menni anélkül, hogy előbb az elért eredményeket fel ne használjuk másutt, valamint meg ne vizsgáljuk, hogy a más területeken elért eredmények miképpen hasznosíthatók ezen a területen. Gondoljunk itt az IBM számítógépgyár programfejlesztő törekvéseire, amely teljesen hasonló feladatok megoldására tervezte a COGO, a STRESS nyelveket, illetve programrendszereket és ugyancsak érdekelt az IITRI-ben végzett fejlesztésben az APT rendszerrel kapcsolatban.

A numerikus vezérlésű szerszámgépek gazdaságos programozásának igénye az MIT-ben folyt fejlesztésen, az APT rendszer fejlesztésén keresztül kétségtelenül elősegítette általában a problémaorientált számítógépprogramozó nyelvek, sőt a szimbólikus, program-orientált számítógépprogramozó nyelvek fejlődését is, ma azonban inkább a NSz-ek programozásának fejlettsége függ a számítógépprogramozó rendszerek fejlesztése terén elért általános eredményektől. Kétségtelen azonban, hogy az automatizálásban és a számítógépeknek e célra történő felhasználásában az alkatrészgyártásnak, a szerszámgépeknek uttörő szerep jutott.

S u m m a r y

Numerically controlled machine tool programming with a
computer.

Numerically controlled machine tools appeared in recent years in Hungary and their development and organisation of production is now under way. In this connection arose the need for the automation of the programming of numerical machine tools. The examination of the problem was assigned to the Computing Centre of the Hungarian Academy of Sciences.

In the present paper the author gives a short synopsis of the results obtained in different parts of the world in the field of the use of universal digital computers for the automatic programming of numerically controlled machine tools. Arising problems and principles of their solution are discussed. Furthermore, the article surveys the development, the classification and the trends of further developments of symbolic programming systems.