

## Ipari robotok

*Mobil* robotok egyelőre nem játszanak alapvető szerepet átfogó ipari alkalmazások terén. Jelenleg meglepő tulajdonságokkal rendelkező, leginkább játékokra, házi-robotokként alkalmazható laboratóriumi modellek vannak - a japánok ebben úttörők - (fonó, kétlábos járó, orgonán játszó robotok, stb).

A VDI 2860 irányelv (1981) szerint: „az ipari robot univerzálisan állítható többtengelyű mozgó automata, amelynek mozgásegymásutánisága (utak és szögek) szabadon – mechanikus beavatkozás nélkül – programozható és adott esetben szenzorral vezetett, megfogóval, szerszámmal vagy más gyártóeszközzel felszerelhető, anyagkezelési és technológiai feladatra felhasználható.”

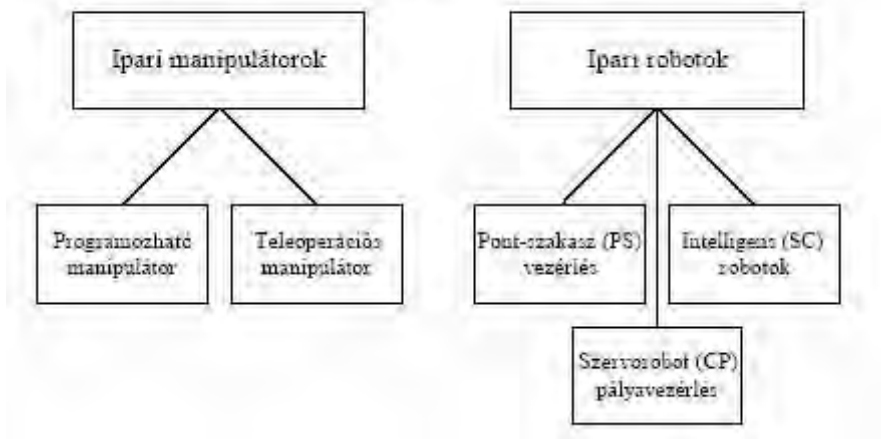
A másik megfogalmazásunk ennél tömörebb:

A robot olyan, nyílt kinematikai láncú mechanizmus, amely:

- irányított mozgásokra képes,
- megfogója *rugalmasan* programozható,
- mesterséges intelligencia funkciókra képes

A robot mechatronikai egység, szerkezet, azaz *mechanikai, elektromechanikai* és *elektronikai* szerkezeti elemek integrálásából származó berendezés.

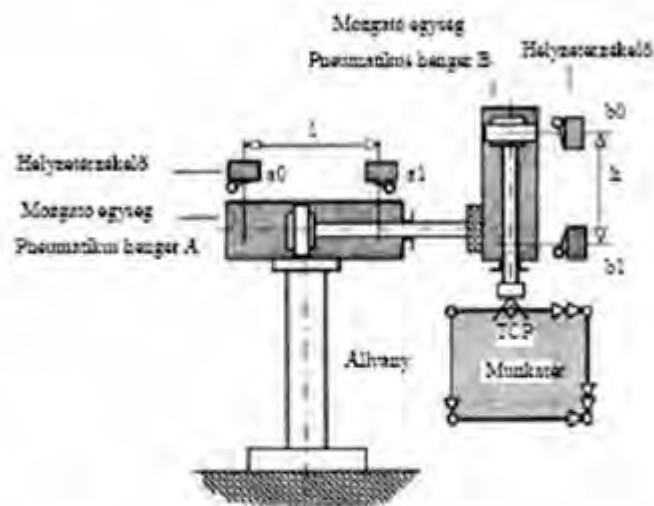
### Robotok csoportosítása irányításuk szerint



Az ipari robotok két fő csoportba sorolhatók: **"standing"** (alaphoz rögzített) és **"portal"** (három további független tengely körül forgatható keretbe, "portál"-ba szerelt, ezáltal lényegesen megnövelt munkaterületű) robotok.

### Programozható manipulátor:

Tulajdonképpen helyező berendezés. Általában megfogó-szerkezettel van felszerelve. Programja egyszerű, fix program, a vezérlés egyetlen mozgásciklus végrehajtására alkalmas. A mozgások egymásutánisága merev program szerint fut le és mechanikus behatás nélkül nem változtatható meg. Ilyen irányítórendszerek a pneumatikus logikákkal realizált vezérlések és a huzalozott relés logikára épülő vezérlések. Programozásuk tehát meglehetősen kényelmetlen.



Ipari manipulátor



### **Teleoperációs manipulátor:**

Tulajdonképpen kézi irányítású manipulátor, melynek távirányítását rudazattal, vagy joystick rendszerű erőátviteli rendszerrel végzik. Főleg ott alkalmazzák, ahol az anyagmozgatási, vagy technológiai munkatér az ember számára veszélyes. A kezelő ekkor figyelőablakkal ellátott, szeparált térben dolgozik, ahonnan a munkatér belátható. Segítségével bonyolult mozgások valósíthatók meg emberi irányítással.

### **Pont-szakasz vezérlésű robot:**

A célpontok adottak, a pályákat a robot felépítése határozza meg. Nincs útmérés, a koordináták mentén diszkrét helyzetérzékelés van. Nincs helyzet-szabályozóköros pozicionálás, csak útvezérlés. Programozása mozgássorrend előírásból és helyzetkapcsolók beállításából áll. Vezérlése PLC szintű. A robotkéz helyzetét csak kevés pontban definiálja.

### **Pályavezérlésű, vagy szervrobot**

A programozott pont pályáját a program meghatározza. Koordinátáknként külön útmérővel rendelkezik és helyzet-szabályozókörrrel pozicionál. A robotfunkciók és pályák számjegyesen programozhatók (CNC típusú vezérlő). Általában van betanítási üzemmód.

### **Intelligens robot**

Bizonyos pályaelemeket a feladat jellege határoz meg a robot szenzorinak segítségével. Pályavezérlésű, magas szinten programozott robotok.

## **Robotok csoportosítása az ellátandó feladat szerint**

### **Anyagkezelő robotok**

Feladatuk munkadarabok és műveleti eszközök manipulálása (általában a műveletek szünetében).

### Jellegzetes feladatok:

- adagolás megmunkálógépre,
- megtartás, forgatás (operációhoz, vagy operáció alatt),
- áthelyezés program szerint,
- válogatás, rendezés,
- szerszámozás, szerszámcsere,
- mérőeszköz csere,
- ...



Anyagmozgató robot

### Műveletvégző robotok

Feladatuk műveleti eszközök operatív mozgatása.

### Jellegzetes alkalmazások:

- festés, tisztítás, sorjázás,
- pont- és vonalhegesztés, varratlerakó hegesztés,
- fúrási, ujjmarási műveletek,
- lézeres megmunkálási műveletek,
- láng- és plazmasugaras vágás,
- mérőeszköz mozgatás, mérés kiszolgálás



Hegesztő robot

### **Szerelő robotok**

Feladatuk munkadarabok és műveleti eszközök manipulálása, mozgatása.

Általában szenzorokkal rendelkeznek.

Jellegzetes feladatok:

- alkatrészek kiválasztása,
- alkatrészek orientálása,
- alkatrészek összeillesztése, helyezése,
- kötések létesítése.



Szerelő robot

## Alapvető fogalmak és definíciók

### Mechanika

A klasszikus ipari robot **teste** a környezethez rögzített tengely (test-tengely) körül szabadon elforgatható - általában  $360^\circ$ -nál kisebb szöggel. A robottesthez a **felsőkar** a **vállízületen**, az **alsókar** a felsőkarhoz a **könyökízületen**, a **kéz** az alsókarhoz a **csuklóízületen** keresztül kapcsolódik. A kézhez különböző **effektorok** (fogó, hegesztő- vagy szórópisztoly, stb.) rögzíthető. A robottestet, a felső- és alsókart külön motorok mozgatják; a kéz, ezáltal az effektor térbeli **orientációját** pedig 3 motorral lehet beállítani. A robotnak tehát legalább 6 tengelye (ízülete), s mindegyik tengelyhez külön motorja van. A konstrukcióra jellemző határokon belül mindegyik tengely a többihez **függetlenül** mozgatható. Bár a tengelyek általában forgatást tesznek lehetővé, egyes esetekben beszélhetünk **transzlációs** tengelyekről is (Scara-elven működő **hajlókarú robotok**, mozgatható robotkeretek esetén).

A kéz és ezáltal az effektor helyzetét az ún. **TCP**-vel (tool center point) jellemezzük; ez a pont az effektor típusától függ: hegesztőpisztoly esetében a TCP az elektróda-csúcsok közötti felezőpont, egy fúró esetében a fúró hegye, stb. A TCP-t a kéz síkjának középpontjához viszonyítjuk.

A **munkaterület, WS** (working space) azon pontok összessége a térben, amelyet a TCP elérhet. A WS-t a konstrukció geometriája határozza meg, határoló felületei általában síkok, henger-, kúp- és gömbfelületek.

Ha  $P \in WS$  a TCP számára elérhető pont, akkor annak ismételt megközelítése esetén a TCP a  $P$  körüli *variációs gömb* belsejébe esik. Az **ismétlési pontosság** a legnagyobb variációs gömb sugara, ha  $P$  bejárja WS-t. Az ismétlési pontosság a robot alkatrészeinek precizitásától és merevségétől függ.

Ha  $Q \in WS$  tetszőlegesen kijelölt pont (függetlenül attól, hogy a TCP elérheti), akkor az ismételt megközelítések esetén a TCP és  $Q$  között adódó legnagyobb eltérést **pozicionálási pontosságnak** nevezzük. A pozicionálási pontosság egyrészt az ismétlési pontosságtól, másrészt a pozíciómérő rendszer felbontásától függ.

### A robotok felhasználása

Az USA-ban először 1965-ben, Európában a hetvenes évek elején jelentek meg az ipari robotok, elsősorban az autóiparban. Elterjedésükre jellemző példa, hogy míg 1980-ban egy BMW-14 előállítására 192, 1985-ben már csak 89 órára volt szükség.

80-as évek végén Németországban körülbelül 14.000 ipari robot működött szerelési (ponthegeztés, vonalhegeztés, összeszerelés, zománcozás, polírozás, stb.) és termelési (szerszámgyártás, hengerelés, kovácsolás, fröccsöntés, stb.) folyamatokban, viszonylag kis százalékban (2,8 %) kutatási és oktatási célokra is.

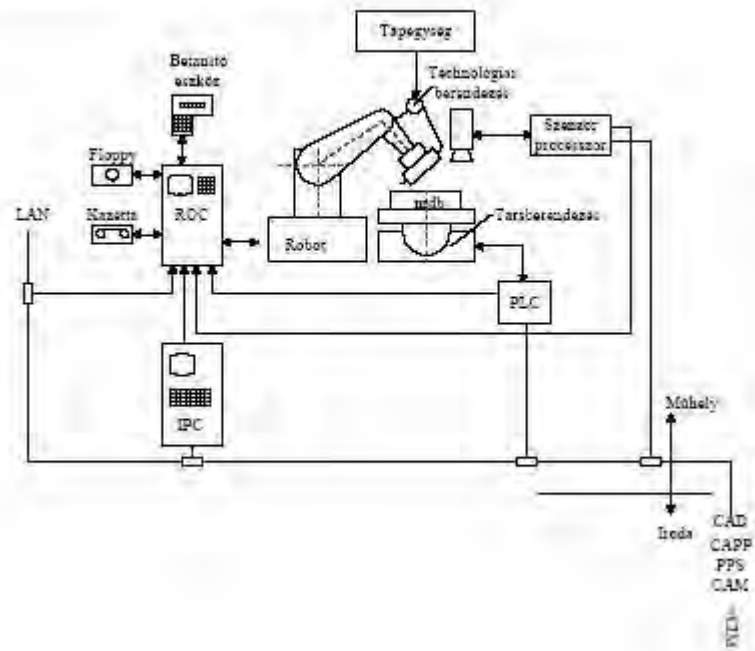
Terjedésük elsősorban azokban az országokban gyors, ahol a bérek magasak (Németország, Svédország). Itt több-száz ezer munkahely szűnt-szűnik meg a robotok alkalmazása miatt.

A belátható jövőben elsősorban

- a mezőgazdaságban (birkanyírás, fejés),
- a mély tengerekben (pl. a transzatlanti kábel karbantartása, elsüllyedt kincsek keresése, stb.),
- az űrkutatásban (űrállomások összeszerelése),
- a gépipari tervezésben,
- a hadiiparban (automata tankok és fegyverrendszerek, majd intelligens harci robotok) is várható az elterjedésük.
- A távolabbi jövőben a determinisztikus alapon működő robotokat felválthatják a feladat-orientált, szenzorvezérlésű, automatikus adaptálódásra képes robotok.
- A különböző **CAP** (Computer Aided Planning), **CAD** (Computer Aided Design), **CAM** (Computer Aided Manufacturing) és **CIM** (Computer Integrated Manufacturing) folyamatok teljesen integrálhatóak lehetnének segítségükkel.
- A szenzorok tökéletesítése által az emberi érzékelés utánzása, sőt, némely területen meghaladása is elképzelhető.
- Integrált szakértői rendszerek kifejlesztése (a robotok független, racionális és célszerű döntések meghozatalára képesek).

A további fejlődés már a fantasztikus regényekben megjósolt lehetőségek megvalósítása lenne: intelligens, független mozgásra és cselekvésre képes robotrendszerek, emberhez hasonló (antropomorf) androidok.

## Robotikai rendszer felépítése



## Robotikai rendszer felépítése

### A rendszer hardver elemei:

Az elvégzendő technológia megvalósításához szükséges elemek általában a következők:

- robot,
- technológiai berendezés,
- segéd-, vagy társberendezések,
- szenzorok



Az irányítás hardver elemei:

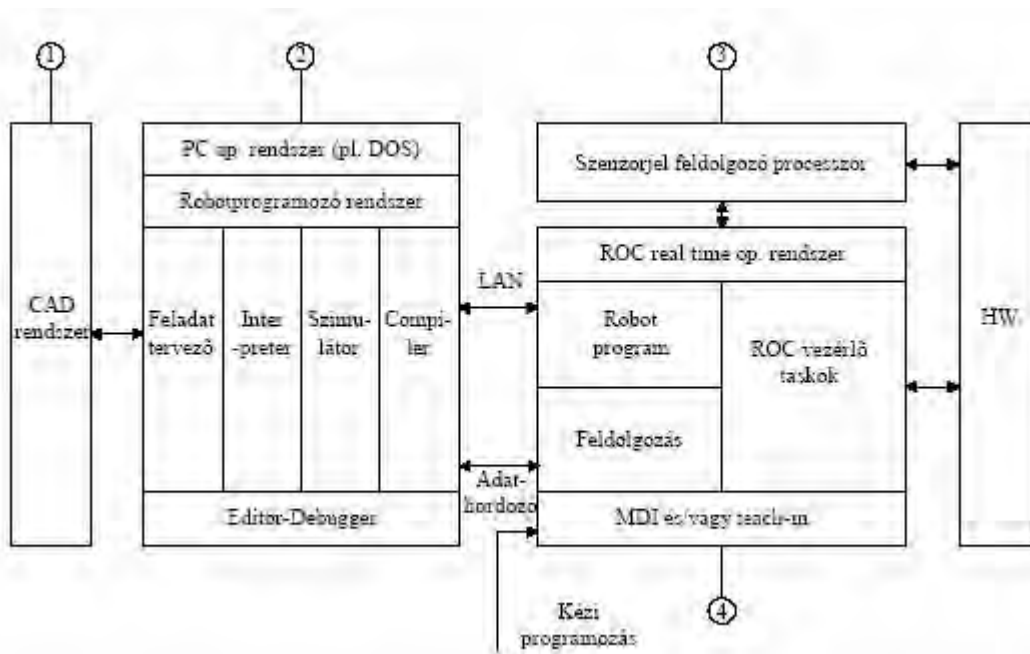
Alapkiépítés:

- robotvezérlo (ROC),
- szenzor processzor (SPS),
- segédberendezés vezérlo PLC

Alternatív elemek:

- cellavezérlo (CAM modul),
- helyi hálózat (LAN),
- tervező munkahelyek (PC, CAD/CAPP... ).

**A rendszer szoftver elemei.**



- CAD rendszer, azaz tervező számítógép szoftverekkel (AutoCAD, CADKEY, FEM, ...),
- Számítógéppel segített robotprogramozó rendszer,
- Szenzorjel feldolgozás,
- Robotvezérlo (real time op. rendszer).
- A programok átküldését az egyik egységtől a másikhoz kommunikációs szoftverek támogatják.

**Vezérlési módok**

A legalapvetőbb feladatok egyike a TCP mozgatása egy adott indulási helyről az adott végpozícióba. Különbséget teszünk egyszerű **mozgásvezérlés** és **összetett szabályozófunkciók** között. Az első vezérlési mód mindazon funkciókat foglalja magában, amikor a pozíció-mérőrendszer adatait visszacsatolva szabályozzuk a motorokat. Az összetett

szabályozófunkciók magasabbrendűek az előzőnél és általában indítják azokat. Mindezeket egy példával világítjuk meg:

A mozgás szabályozását a ROB3 belsejében elhelyezett elektronika végzi. Ez az elektronikus szabályozó rendszer a parancsokat ASCII szekvenciák (American Standard Code for Information Interchange) formájában kapja a vezérlő számítógéptől az RS232 *interface*-en keresztül, és elvégzi a kiindulási beállítást. Ezzel egyidőben folyamatosan összehasonlítja a pillanatnyi pozíciót a beállított pozícióval **minden egyes** motornál. Ha ez a két változó megegyezik, a mozgás leáll. Zavarok esetén rezgések keletkezhetnek (*hardware* hiba). Minden magasabb szabályozó funkció a ROB3 vezérlő számítógépében helyezkedik el.

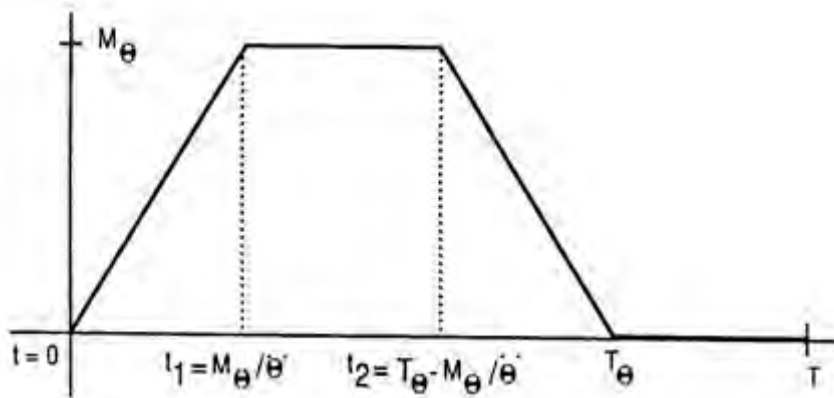
Az ipari robotok motorjának **vezérlő egysége** a következő részekből áll:

- egy rendkívül precíz szögmérő-kódolóval megvalósított pozíció-mérőrendszer. Ez lehet, pl. egy üveggörög, amelynek peremén elhelyezett rovátkák (> 2000) forgás közben impulzust keltenek egy foto-elektromos kapun. Így a forgássebességtől függően maximum 800kHz-es helyzetjelző jelek keletkeznek a vezérlő egység számára. Az érvényes beállított helyzettel (tehát nem a véghelyzettel) való összehasonlítás egy ún. **követő hibajel** generál. A követő hibajelből (a **drift kompenzáció** hatásának figyelembe vételével) képződik a sebesség-beállító jel a motorvezérlés részére. Ez utóbbi biztosítja, hogy a TCP ne végezzen a beállított helyzet körül rezgéseket. Ez az elem azért szükséges, mert az ipari robotok motorjai, egyenáramú motorok, vagy újabban háromfázisú motorok. A probléma nem létezik léptető motorok esetében (itt egyéb nehézségek vannak).

A mozgásvezérlésnek általában két alapvető módját különböztetjük meg: a **PTP-vezérlést** (Point-to-point) és az **útvonalvezérlést**. Az előbbi esetén csak a mozgás kiinduló- és végpontja alapvető; a **TCP** által bejárt útvonal másodrendű. Az utóbbi esetén viszont elsőrendű, hogy a TCP, amennyire csak lehet, az előre meghatározott útvonalon mozogjon.

## **PTP vezérlés**

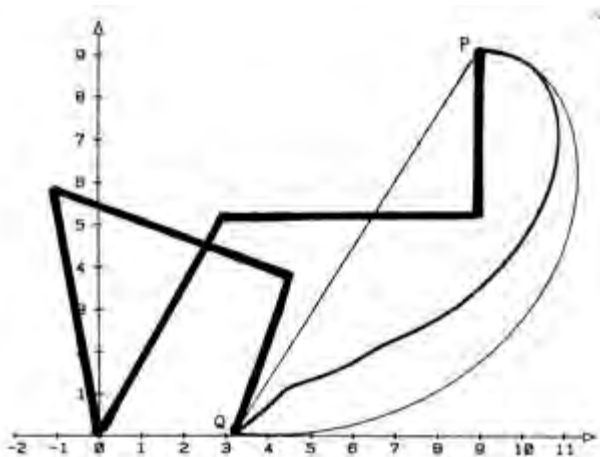
A PTP vezérlés esetén a mozgás kiinduló- (1) ill. véghelyzetét (2) a robottengelyek körüli forgásszögek értékeinek megadásával jellemezzük. A pillanatnyi helyzetben minden egyes tengelyre nézve kiszámoljuk a megfelelő szögmértékeket. Tiszta PTP vezérlés esetén az egyes motorok egymástól függetlenül, a lehető leggyorsabban végrehajtják a hozzájuk tartozó tengely körüli forgatást. Minthogy a szögmértékek a mozgás folyamán változnak, a mozgást kiváltó motorok különböző időpontokban működnek. Ennek következtében a mozgás szaggatottá válhat.



Ha a mozgás részleteit vizsgáljuk, azt látjuk, hogy a forgatás **álló** szöggyorsulással indul, ezután a maximum szögsebességet elérve állandó szögsebességgel halad egy ideig, majd állandó szöggyorsulással fékezik. A teljes  $\Delta\Theta$  szögelforduláshoz szükséges  $T$  időt az elemi kinematikából ismert módon, integrálással számíthatjuk ki ( $M\dot{\theta}$  a szögsebesség):

$$T_{\theta} = \Delta\Theta / M_{\theta} + M_{\theta} / \ddot{\Theta}_{T_1}$$

A teljes PTP elmozdulás ideje az egyes tengelyek körüli elfordulási idők közül a legnagyobb. Az eredmény azért fontos számunkra, mert főleg nehéz terhek mozgása során a PTP mozgás szaggatottsága a robot ízületeinek és áttételeinek gyors kopásához vezethet. Ipari robotok esetében ezt célszerű elkerülni, az ún. **szinkron PTP-vezérlés**re való áttéréssel. A mi esetünkben ennek a neve **tengely-interpoláció**. A mozgásban résztvevő összes tengely azonos időpontban kezdi és végzi mozgását. Ennek megvalósításához a gyorsabb mozgásra képes tengelyeknek a lelassabbhoz, kell igazodniuk. Az egyszerűség kedvéért a gyorsulásokat nem változtatjuk, csak a maximális szögsebességek csökkennek le a megfelelő értékre.



Az ábrán láthatjuk a tiszta és szinkron PTP útvonalak közötti különbséget: a felső  $P$  pont, függőlegesen felfelé állított fogóval, a kiinduló helyzet. A  $Q$  végpontba való mozgás a felsőkar - alsókar síkban jön létre az A1 test-tengely részvétele nélkül.

Látható, hogy a szinkron PTP- mozgás egyenletes ívű, bár lényegesen eltér a  $PQ$  egyenes szakasztól. Ilyen eltérések váratlan mellékhatásokkal járhatnak: pl. akadályba ütközés, ami nem fordulhat elő tiszta PTP mozgás esetén.

Hosszú útvonalak esetén mind a tiszta, mind a szinkron PTP-mozgatásnál célszerű a kezdő- és a végpont közé **köztes pontokat** beiktatni. Ezt az eljárást **interpolációnak** nevezzük (lásd az említett angol nyelvű irodalmat).

## Útvonalvezérlés

Sokszor szükség van egy szerszám, vagy akár egy munkadarab a lehető legnagyobb pontossággal - egy adott térbeli görbe mentén - előre megadott **útvonal-sebességgel** való mozgatására. (Ilyen feladat lehet a vonalhegesztés, ragasztás, egy fúró adott helyre való merőleges beállítása, stb.). Ezekben az esetekben nem fogadhatók el a PTP-mozgatásra jellemző rángások és éles irányváltoztatások. Ekkor célszerű az útvonalvezérlés vagy CP - vezérlés (Continuous Path control) alkalmazása. A PTP-mozgatással ellentétben a CP-mozgatás alkalmazásával az útvonal pillanatnyi kinematikai jellemzői - helyzet, vonalmenti sebesség és gyorsulás - határozzák meg minden egyes motor működését. A funkcionális függés transzformációs egyenletek útján adható meg.

Ebben az üzemmódban a vezérlés kemény követelményeket támaszt a mozgó- és vezérlőegységekkel szemben. Különösen nagy feladat hárul a vezérlő processzorra, hogy az a számítási műveleteket elfogadható időn belül végezze el.

## Programozás

Programozás alatt a továbbiakban nem az egyszerű mozgás-vezérlés gépi rutinjait, hanem az egyes modulok, eljárások és függvények alkalmazását fogjuk érteni.

## Program-eljárások

### Elektromechanikus programozás:

Az előre huzalozott programot csak *hardware úton* lehet változtatni. Egyszerű feladatokra használható robotok, úgymint adagolóeszközök és hengerművek esetén használatos; ez mára már jelentőségét veszítette.

Modern ipari robotok esetében jóval inkább a **tanulás előjátszás alapján** módszer kerül előtérbe. Ennek több, a felhasználói területtől függő változata létezik.

### Mozgás programozása visszajátszással (Play-back):

A "tanító" végigjátssza a TCP mozgását, amelyet egy adott időfelbontásban rögzítünk. A program futása során a mozgás minden fázisa PTP-vezérlés útján ismétlődik. A play-back módszer még mindig használatos, pl. polírozó-robotok programozásához.

### Master-slave programozás:

A programozó a robotot egy áttétel segítségével (pl. egy kis modell-robot megfelelően arányos vezérlése útján) mozgatja. A "master" minden tevékenységét (nemcsak a mozdulatokat!) tárolja, és azonos időfelbontásban, azonnal le is játssza a "slave" számára.

Mivel ez a módszer viszonylag bonyolult, elsősorban ott használják, ahol az előző, betanító módszer nem lehetséges (pl. atomerőművek aktív zónája). Abban az esetben, ha a tárolási folyamat kimarad, **teleoperációról**, vagy **telemanipulációról** beszélünk.

### Betanító (Teach-in) programozás:

Ezt a módszert jelenleg is használják majdnem minden ipari robotnál. A programozó a robotot egy mozgatón keresztül vezérli - vagy a programozási helyek durva beállításával vagy a tanítás-vezérlő programozó eszköz (teach control device) manuális használatával. Evvel a nagyobb zsebszámológéphez hasonlatos eszközzel a robot, biztonsági okokból, csak erősen csökkentett sebességgel működtethető. Nem csak a helyzetek, de az interpolációk, gyorsulások, sebességek, késleltetési idők és szenzor-utasítások is a programba kerülnek.

Megjegyzendő, hogy a modern rendszerekben általában csak a program-keret tanítása történik evvel a módszerrel. A mozgás részletei **on-line** módon kerülnek a programba.

### Off-line programozás:

A programozás szöveges formában írt program segítségével történik. Az on-line programozással ellentétben, amely rendkívül mozdulat-orientált, az off-line programozás inkább **feladat-orientált**. Ahelyett, hogy a robot minden mozdulatát külön beprogramoznánk (mint pl. a teach-in módszer esetén), a feladat absztrakt megfogalmazására helyezzük a hangsúlyt. A programozás hatékonyságát egyes rendszerek felhasználóbarát szolgáltatásokkal segítik (pl. az Apple **Mackintosh** a grafikus tervezési elemekkel). A módszernek, sok előnye mellett (a programozás már a tervezési fázisban lehetséges, stb.), alapvető hibája többek között, hogy a programozás hibái sokszor csak az első próbefutás alatt derülnek ki. A hibakeresés és -javítás csak ismételt futások során végezhető, ami növeli a robotok improduktív idejét.

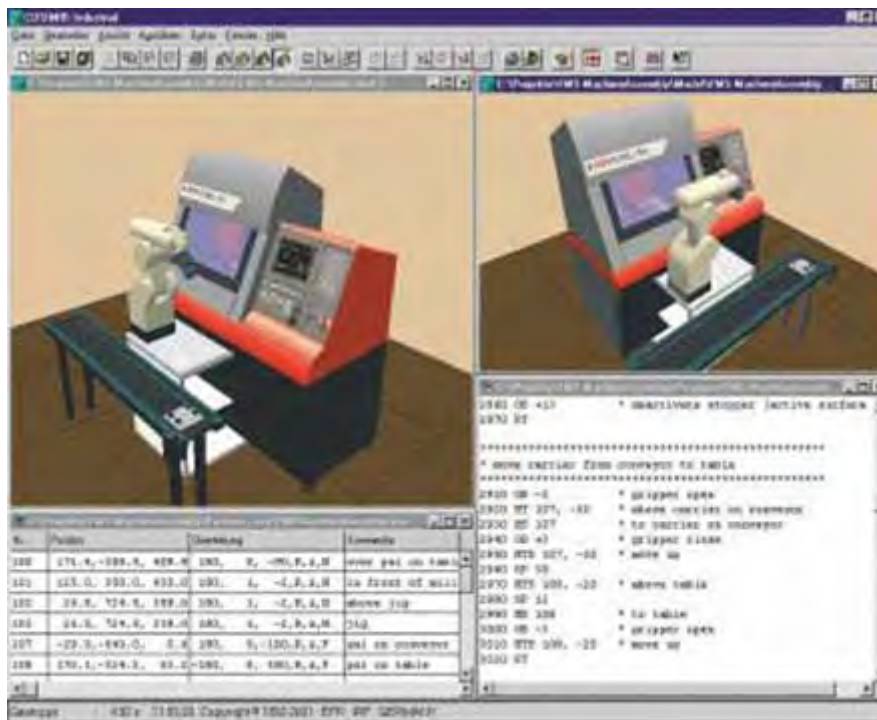
A másik hátrány a programozás absztraktságának következménye, nevezetesen az, hogy a robot egyes funkciókat ellátó részeit és azok kapcsolatát ideálisnak tételezik fel, így nem tud előre számolni a gyakorlatban előforduló pontatlanságokkal és egyéb zavaró körülmények (hőmérséklet, páratartalom, rezgések, stb.) hatásával.

Azt mondhatjuk, hogy a **tisztán** off-line módszer nem jó megoldás. A fenti hibák viszont elkerülhetők, ha a próbefutás előtt az új programot először a terminálon szimuláljuk. Ehhez természetesen szükség van egy teljes **geometriai** és **technológiai adatbankra**, amely egyfelől az összes alkalmazni kívánt robot alkatrészeinek és perifériáinak (forgó asztalok, futószalagok, stb.) geometriai paramétereit, másfelől a munkafolyamatban szereplő egyéb adatokat (motorok és szenzorok karakterisztikái, teher-válaszfüggvények, szerszámok és munkadarabok fizikai jellemzői) tartalmazza. A szimuláció során a kimenet **real-time** módon 3-dimenziós grafika formájában képernyőre kerül, ahol diagnosztizálhatók és kijavíthatók a hibák - szinkronizációs problémák, ütközések, túlterhelés, stb.

A precíz TCP pozícionálási hibáit a **szenzortechnológia** alkalmazásával próbálják kiküszöbölni. Hagyományosan **tapintó** és **vizuális** elven működő szenzorokat használnak a robotok **adaptív vezérlése** céljából. Itt röviden olyan, **nem-determinisztikus** módszereket mutatunk be, ahol a TCP által bejárni tervezett útvonal a programozás alatt még nem minden részletében ismert. A program futása alatt a szenzorok tudósítják a központi műveleti egységet a robot belső "világát" érintő külső eseményekről, amelynek következtében a robot "modell-világában" megfelelő belső események keletkeznek. Ezáltal a szenzor által küldött információkra a robot adekvát módon képes reagálni. Ezen az elven működik például a **varrat-követő szenzor (seam trace sensor)**. A hegesztendő felületen a kiinduló hegesztési pontot teach-in módszerrel adjuk meg a robotnak. A varrat útvonala a munkadarabtól függ, a szenzor a hegesztés alatt folytonosan követi a hegesztő elektróda útját. A munkafolyamat egyes beprogramozandó lépéseinek száma és bonyolultsága lényegesen csökkenthető. Természetesen ennek a módszernek megvannak a korlátjai (alakfelismerési problémák, **reach-in-the-box** feladat). Ha egy dobozban, amelybe további feldolgozásra váró munkadarabok esnek, elég nagy a "rendetlenség", és a kiszemelt munkadarab kellően bonyolult alakú, akkor annak azonosítása, helyzetének meghatározása és a manipulátorral való megközelítése túlságosan is komplexnek bizonyulhat a robot alakfelismerő programja számára.

## Programozási nyelvek

Elvben minden univerzális programnyelv alkalmas robotok programozására. A feladat speciális természetéből következően mégis célszerű volt olyan programnyelvek kifejlesztése, amelyeken a tipikus feladatelemek, mint koordináta-transzformációk, effektor- és szenzorvezérlés könnyen programozhatók. Ilyenek például az **SRL** (Structured Robot Language), a **PASRO** (PAScal for RObots), **ROBEX** (ROBot EXapt), **ARLA** (ASEA Robot Language), **BAPS** ( Bosch Advanced Programming System), **SRCL** (Siemens Robot Control Language), **ROLF** (Robot Language Formula), **ROBOTstar**, **AL** (Assembly Language), **AML** (A Manufacturing Language), **HELP**, **VAL** (Variable Assembly Language), **SIGLA** (SIGma LAnguage).



Cosimir

## Felhasznált irodalom:

1. Dr. Csáki Tibor ROBOTTECHNIKA.
2. Dr. Kulcsár Béla: Robottechnika
3. Makó I. Robottechnika előadásvázlat
4. <http://www.szgt.uni-miskolc.hu/~mako/robel1.pdf>
5. <http://www.szgt.unimiskolc.hu/~mako/robel2.pdf>
6. <http://www.szgt.uni-miskolc.hu/~mako/robel3.pdf>
7. <http://www.szgt.uni-miskolc.hu/~mako/robot.pdf>
8. <http://www.muszeroldal.hu>
9. <http://www.axicont.hu/robot.htm>