

# GÉPIPARI AUTOMATIZÁLÁS

## Ipari robotok irányítása, programozása

5. előadás

Dr. Pintér József

# TARTALOMJEGYZÉK

1. ROBOTVEZÉRLÉSEK TÍPUSAI
2. ROBOTOK PROGRAMOZÁSA

**Dr. Pintér József**

## A ROBOTOK IRÁNYÍTÁSA

A robotkar egy előre definiált útvonalon, pályán visz végig egy szerszámot, megfogót, szórópisztolyt, stb., és a pálya bizonyos pontjain ezekkel különféle műveleteket végez. Felépítésük mozgásuk meghatározottsága alapján a robotvezérlések típusai:

- Pick and place robot (manipulátor) vezérlések
- Pontvezérlésű robotok
- Pályavezérlésű robotok
- Teach-in play-back (lejátszós) robot vezérlés



**Dr. Pintér József**

- ❖ **Pick and place robot (manipulátor) vezérlések**
  - ✓ Koordinátáinként két (esetleg néhány) diszkrét pozícióba vezérelhetők.
  - ✓ A mozgások sorrendjét (szekvenciáját) egyszerű vezérlő (pl. PLC adja).
  - ✓ A pozíciók kijelölése mechanikus ütközőkkel, vagy más kétállású szenzorral történik. Programozáskor az ütközők állítandók és a vezérlés szekvenciája átprogramozandó.
  - ✓ Egyszerű feladatokhoz (pl. adagolás, rakodás) használhatók.

**Dr. Pintér József**

## ❖ Pontvezérlésű robotok

- ✓ A robot programozott pontja áthalad egy pontsorozattal adott útvonalon, de két – szomszédos – pont között a pálya csak hozzávetőlegesen ismert, mert a csuklók nincsenek egymással szinkronizálva.
- ✓ A vezérlés az adott pontokból – inverz transzformációval – kiszámítja a csukló koordinátákat, és alapjelként átadja a szervohajtásoknak végrehajtásra.
- ✓ Programozásuk – a pontsorozat megadása útján történik betanítással, vagy számítógépes támogatással.

Ezt a vezérlést **PTP (Point to Point)** irányításnak nevezik.

## ❖ Pályavezérlésű robotok CP (Continuous Path)

✓ A (csuklónként elhelyezett) szervohajtások közös órajellel működő alapjelképzőről (interpolátor) kapják alapjelüket, amelyet követnek, ezért a tengelyek sebessége egymáshoz szinkronizált, és a robot követi az előírt pályát.

✓ A legáltalánosabban használható, összetett robotvezérlés.

✓ Szereléshez, megmunkáláshoz, hegesztéshez is alkalmasak.

- ❖ **Teach-in play-back (lejátszós) robot vezérlés**
- ✓ Egy nagy kapacitású információ tároló (pl. mágneses adathordozó), melyen a betanítás szakaszában felvett csukló koordinátákat tárolja a futó idő függvényében.
- ✓ Felépítése egyszerű.
- ✓ Nem tartalmaz aritmetikai egységet és interpolátort.
- ✓ Rendszertechnikailag jelentősen eltér az előző típusoktól.
- ✓ Alkalmazási területe: festés, csiszolás.
- ✓ A program módosítása körülményes.

**Dr. Pintér József**

A következő egységek szükségesek egy robot vezérlési feladatainak elvégzéséhez:

- ❖ CPU és aritmetikai processzor modul
- ❖ Memória modul a program illetve pályaadatok tárolásához
- ❖ Külső adattároló
- ❖ Kezelő egység (terminál)
- ❖ Kézi vezérlő egység
- ❖ I/O
  - Bináris
  - Analóg
  - Digitális
  - Egyéb interface-ek
- ❖ Szervo modul a motorszabályozáshoz
- ❖ Egyéb közvetlen szerszámvezérlő egységek

modul  
I/O  
egység  
soros/párhuzamos egység



# Robotok programozása

A robotok programozásának lehetséges módszerét döntő módon meghatározza

- a vezérlés típusa és
- intelligenciája.

Elterjedt módszerek:

- ❖ Egyszerű betanítás
- ❖ Programnyelvi utasításokkal történő programozás
- ❖ Kombinált módszerek



**IGM-KUKA hegesztőrobot programozó-konzolja**

# Robotok programozása

## ❖Egyszerű betanítás

✓Elsősorban teach-in play-back vezérlésű robotok programozási módja, de lehet ilyen a pont- és a pályavezérlésű robotoknál is.

✓Betanításkor: a robotcsuklókat mozgató szervók ellazíthatók, kézzel mozgathatók, ezalatt a vezérlés a futó idő függvényében „lerakja” a csuklókoordinátákat.

✓A betanítás másik módja: a robot szimulátorral (a robot kicsinyített hasonmása, amelyet kézzel mozgatnak) történő mozgatása. A robot követi a szimulátor csuklóinak mozgását és tárolja a koordinátákat

## Robotok programozása

- ❖ Programnyelvi utasításokkal történő programozás
- ✓ A programnyelv szintaktikai szabályok szerint épül fel.
- ✓ A koordinátákat előzetes számítás útján kell meghatározni és programozni.
- ✓ Ismertebb programozási nyelvek:
  - AL, VAL, ROBEX, stb.
- ✓ A robotprogramozási nyelvek felépítése hasonlít a számítástechnikai programnyelvekhez.
- ✓ Programozási szintjük megfelel a robotvezérlés intelligencia szintjének.

**Dr. Pintér József**

- ❖ **Programnyelvi utasításokkal történő programozás**
- ✓ **Utasítástípusok:**
  - a. **Környezetet definiáló, leíró utasítások (POINT, DEFINE)**
  - b. **mozgást leíró utasítások (MOVE, SPEED..)**
  - c. **Logikai, aritmetikai utasítások (AND, OR, +, -, DISTANCE)**
  - d. **a végrehajtás sorrendjét meghatározó utasítások, szubrutin definiáló és hívó utasítások (JUMP, CALL)**
  - e. **megfogó (effektor) működtetés (OPEN, CLOSE)**
  - f. **Input, output utasítások (INPUT, OUTPUT)**

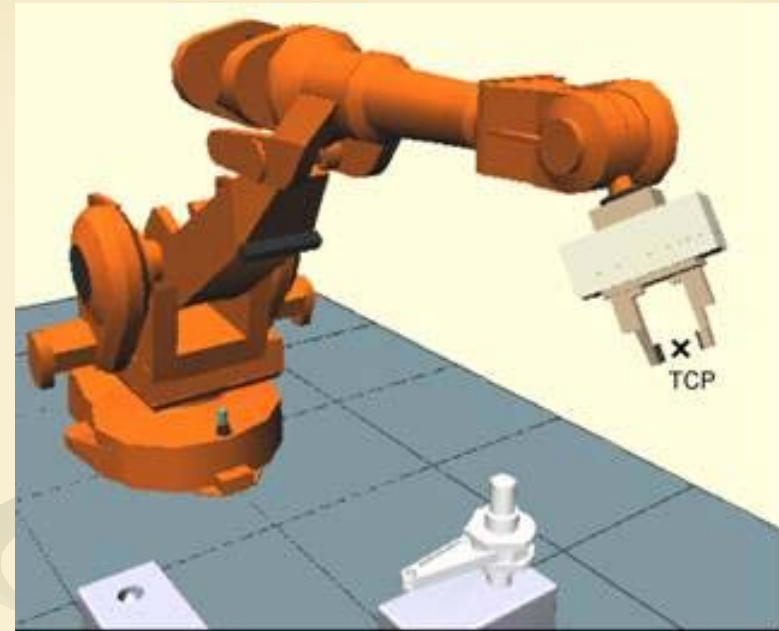
# Robotok programozása

## ❖ Kombinált módszerek

Általában programnyelvi utasításokat használ, de a kritikus koordináta pontok betaníthatók a robot mozgatása útján, és a program hivatkozhat ezekre a pontokra (VAL).

# Robotok programozása

Minden egyes szerszámra vagy megfogóra egy úgynevezett szerszám-középpontot ( **TCP, Tool Center Point**) definiálunk. Ez lehet a festékszórópisztoly szórófeje, vagy az a pont, ahol az ujjak megfogják a munkadarabot. Pályabejárás alatt mindig a TCP adott pályájú mozgását értjük. Ezalatt a TCP pozíciója és orientációja (elfordulása) is a megkívánt módon változik.



# Robotok programozása

A pályabejárás megtervezésénél még egyéb tényezőkre is ügyelnünk kell. Figyelembe kell vennünk a robot munkaterét:


- Milyen messze tud kinyúlni a kar, mit tudunk vele elérni pl. a szerelőasztalon?
- Hol vannak azok a „holt terek” az elvileg ideális munkatéren belül, amelyeket az okoz, hogy a robot ízületei csak bizonyos szögtartományban képesek elfordulni, elcsavarodni?
- Milyen akadályok nehezítik a robot szabad mozgását?

A robotprogramozás két nagy csoportra osztható:

- ON-LINE,
- OFF-LINE.

Az **ON-LINE** programozás: magát a robotot programozzák.

➤ Előnye, hogy a programozó számításba tudja venni a munkaterületen elhelyezkedő tárgyakat és azonnal ellenőrizni tudja a működést.

➤ Hátránya: különösen ipari szempontból jelentős  a programozás idejére a robotot le kell állítani, tehát ezalatt nem dolgozik.



## A robotok programozásának módjai

A sokfajta elvégzendő munkafolyamat, illetve az ezekhez tervezett különféle robotok más és más programozási technikákat igényelnek. Az ún. **ON-LINE** programozás megköveteli a robot jelenlétét: a robotot vagy annak modelljét mozgatva tanítjuk be a bejárando útvonalat.

Az **OFF-LINE** programozási mód alkalmazásakor nincs szükségünk a robotra, egy számítógép mellett ülve, 3 dimenziós objektum-szimuláció segítségével, vagy egyszerű szöveges bevitellel írjuk meg a programot.

**Dr. Pintér József**

## ON-LINE programozás

Az online programozás fogalma különböző technikákat takar:

➤ **Direkt betanítás (Direct Teach-In):** Olyan helyeken alkalmazzák, ahol a robot folytonos pályairányítással vagy sebességvezérléssel mozgatja a szerszámot, viszonylag bonyolult pályát bejárva (például autókarosszériák festésénél). A kezelő végigvezeti a robot karját a kívánt útvonalon, miközben a vezérlőegység folyamatosan feljegyzi a robotkar helyzetét, így később önállóan visszajátszhatja azt. Nagyméretű robotoknál a robotkar könnyített, hajtások nélküli modelljét mozgatja a betanító.

Ezt a módszert angolul **Master-Slave Teach-in (mester-szolga betanítás)** néven említik.

**Dr. Pintér József**

## ON-LINE programozás

Az on-line programozás fogalma különböző technikákat takar:

➤ **Indirekt betanítás (Indirect Teach-In, vagy egyszerűen Teach-In):**

Egy kézi vezérlőberendezés segítségével a robotot a pálya lényeges pontjaiba mozgatjuk, és ezek helyzetét memorizáljuk. A robot feladata lesz a pontok közötti pálya megtervezése és kiszámítása.

# ON-LINE programozás

## Pontvezérlésű robotok programozása

Lényege, hogy a robotot a kezelő számítógép, vezérlőkonzol vagy kézi vezérlőkészülék segítségével tanítja meg a szükséges mozgásokra. A tanítás során a kezelő pontonként halad, és az egyes pontokat külön-külön beírja a robot memóriájába. A vezérlőkonzol a robotra vagy mellé rögzített kezelőpult, míg a kézi vezérlőkészülék egy kisméretű, hordozható billentyűzet. A mozgáspontok rögzítése után valamilyen programnyelven meg kell írni azt a programot, amelyik közli a robottal, hogy mit kezdjen az előzetesen letárolt pontokkal.

**Dr. Pintér József**

# ON-LINE programozás

## Pályavezérlésű robotok programozása

A programozó kézzel végigvezeti a robotkارت az elvégzendő feladat útvonalán. A mozgás paramétereit ennek során a robot vezérlőegysége önállóan rögzíti. Ehhez természetesen szükséges, hogy a programozó az adott feladat elvégzésében járatos legyen (pl. hegesztésnél tudja a hegesztés mozzanatait stb.). Ezzel a programozási móddal igen bonyolult működések is elérhetőek, de óriási memóriakapacitás kell az útvonal paramétereinek rögzítéséhez.

## Az OFF-LINE programozás

Ennek során a programozó egy számítógép termináljánál, a robottól függetlenül fejleszti ki a működést irányító programot.

Ez azt jelenti, hogy nem kell a gép mellett lenni, hanem egy számítógépnél egy konzollal végre lehet hajtani a programozást, le lehet szimulálni a robot és a munkadarab viszonyát 3D-ben.



## Az OFF-LINE programozás

Ezután ezt a programot rögzítik a robot memóriájában. A legnagyobb előnye ennek az eljárásnak, hogy a programozás illetve a programfejlesztés alatt a robotot nem kell kikapcsolni. Ennek az ipari alkalmazásokban van nagy gazdasági jelentősége. Ugyanakkor igen nehéz feladat ilyen programot írni, különösen, ha a működési területen sok tiltott zóna van.

## Az OFF-LINE programozás

Offline programozásnál a robot működésének definiálása általában valamilyen magas szintű nyelven történik. A program megírása előtt minden esetben be kell táplálni a bejárando mozgáspálya kitüntetett pontjait. Ez történhet online betanítással, vagy a robottól teljesen függetlenül, szöveges vagy grafikus adatbevitellel.

Bár a TCP helyzetének és orientációjának leírásához elég a robot ízületeinek aktuális állását rögzíteni, a programozó számára ez nem elég szemléletes, és nehezen kiszámítható. Éppen ezért a robot munkakörnyezetében különböző koordináta-rendszereket definiálunk, és a pontokat ezekre vonatkoztatjuk. A következő ábra néhány általánosan használt koordináta-rendszert mutat be.

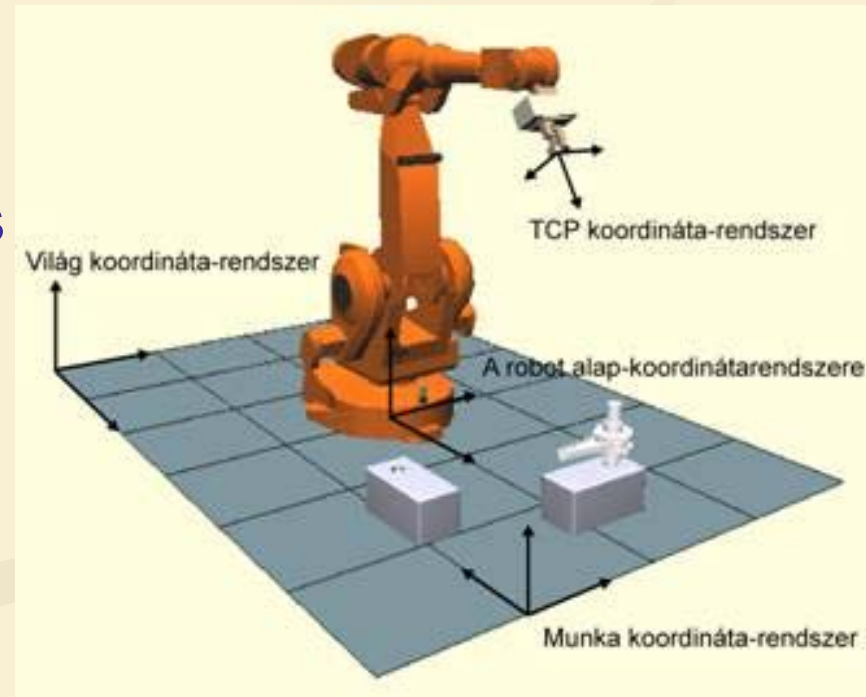
**Dr. Pintér József**



## Az OFF-LINE programozás

# KOORDINÁTARENDSZEREK

- A **világ (vagy bázis) koordinátarendszer** a teljes munkaterület alap-koordinátarendszere. A TCP mozgását az egyszerűbb esetekben erre vonatkoztatjuk.
- A robot **alap-koordinátarendszerét** legtöbbször – a könnyebb számolás kedvéért – a világ koordinátarendszerrel azonosnak tekintjük. Ha több robot dolgozik együtt a munkatérben, akkor ez nem valósítható meg.



## Az OFF-LINE programozás

### KOORDINÁTARENDSZEREK

- A **munka-** vagy más néven **aktuális koordináta-rendszer** az éppen végzett munkafolyamat alapját jelöli. Ehhez a koordináta-rendszerhez rendelhetjük a munkadarabok helyzetét. A munkadarabokhoz további koordináta-rendszereket is rendelhetünk.
- **TCP (szerszám) koordináta-rendszer:** pozíciója és orientációja a világ koordináta-rendszerhez képest egyértelműen definiálja a szerszám helyzetét. A TCP koordináta-rendszerben kiszámíthatjuk a megfogandó munkadarab távolságát és megközelítési irányát.

## Az OFF-LINE programozás

Az off-line programozás illetve a szabadon definiált koordináta-rendszerek használata több olyan lehetőséget nyújt, amelyet az egyszerűbb programozási módok nem biztosítottak:

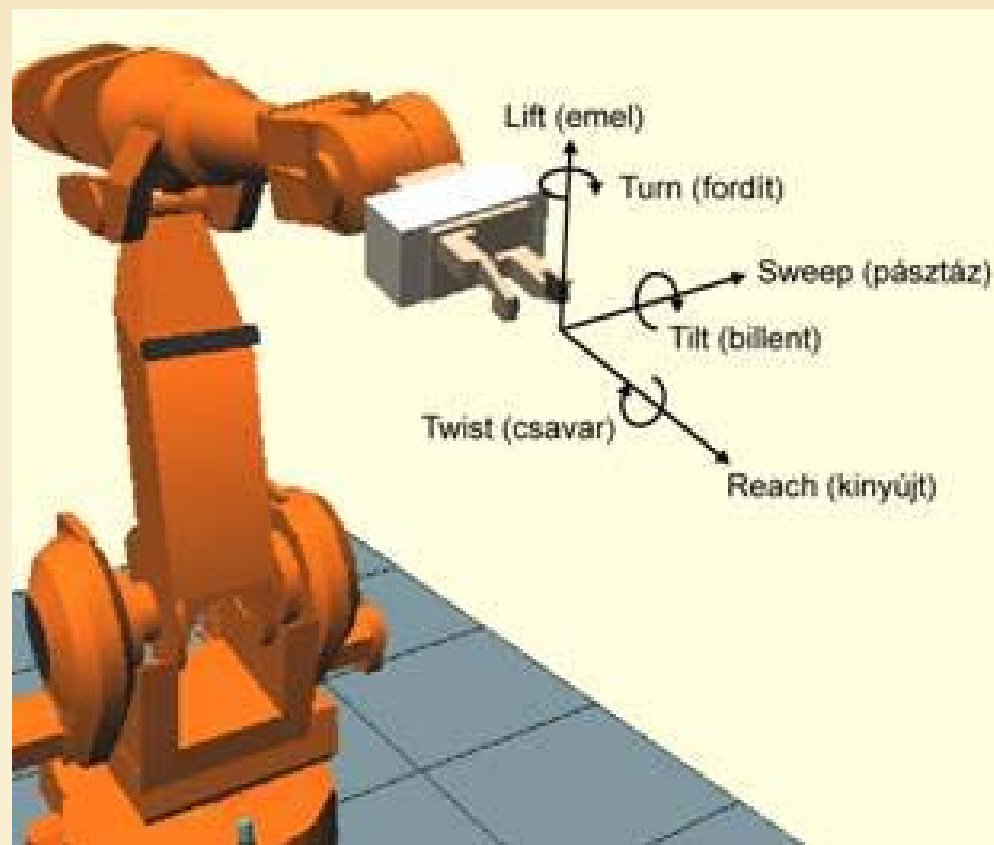
❖ A pálya algoritmikus módszerekkel számítható, így könnyedén beprogramozhatunk olyan ciklikusan változó paraméterű folyamatokat, mint például a rácsszerűen elhelyezkedő furatok egymás utáni elkészítése.

## Az OFF-LINE programozás

❖ A térpontok helyzetét mindig csak abban a koordináta-rendszerben kell megadnunk, amelyikben az a legszemléletesebb, a legkönnyebben számítható. Egy, a munkadarabon levő lyuk helyzetét a leglogikusabb magához a munkadarabhoz – illetve a hozzá rendelt koordináta-rendszerhez – viszonyítani. Ha a munkadarab valamilyen mozgást végez, annak mozgásának alapján – a későbbiekben tárgyalt módszerekkel – a lyuk mozgása is viszonylag könnyedén meghatározható bármely vonatkoztatási rendszerben.

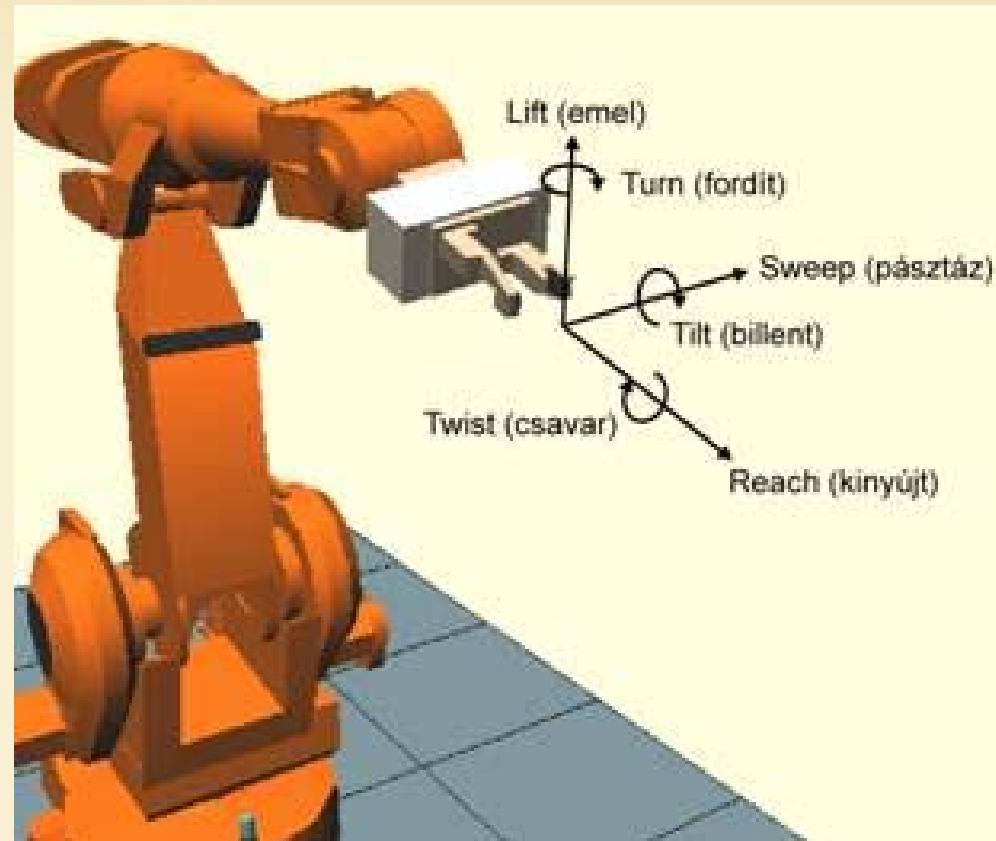
**Dr. Pintér József**

A mozgások definiálása is szemléletesebbé válik, ha a megfelelő koordináta-rendszerre vonatkoztatjuk őket. Az ábra a szerszám szemszögéből elvégezhető mozdulatokat mutatja be: ezek a TCP koordináta-rendszerben történő egyszerű eltolások és elforgatások eredményei.



**Dr. Pintér József**

A koordináta-rendszereknek nem kell feltétlenül derékszögűnek lenniük. Ha a robot geometriája mást kíván meg, alkalmazhatunk henger- vagy gömb koordináta-rendszereket is (RTT illetve RRT karoknál) rendszereket is (RTT illetve RRT karoknál).



**Dr. Pintér József**

## A robot-geometria direkt feladata

A vezérlőegység legalapvetőbb számítási feladata az, hogy az ízületi szögekből és elmozdulásokból meghatározza a TCP helyzetét (a robot alap-koordinátarendszerében). Ezt nevezzük *a robot-geometria direkt feladatának*.

A gyakorlatban megkívánt számítás azonban ennek éppen a fordítottja, vagyis az, hogy a TCP helyzetéből meghatározzuk az ízületi szögeket és elmozdulásokat, s ezáltal a robotot a megfelelő pozícióba állíthassuk. Ez *a robot-geometria inverz feladata*

**Dr. Pintér József**

# A robot-geometria inverz feladata

Többféle módszer létezik az ízületi szögek meghatározásához a TCP helyzetének ismerete alapján (inverz feladat). A számítások sokszor bonyolultak, és a kapott eredmények nem egyértelműek: van, hogy ugyanazt a TCP helyzetet az ízületi szögek más-más kombinációi is előállíthatják, s ezek közül esetleg a munkatér adottságai alapján kell választanunk. Általánosan használható megoldási módok nemigen léteznek, inkább csak sémák, útmutatók. Minden egyes robotra meg kell találni a felépítéséből adódó legegyszerűbb számítást.

**Dr. Pintér József**



**Köszönöm a figyelmet!**

The background features a light beige gradient. In the bottom right corner, there are several overlapping, wavy, light-colored lines that create a sense of movement and depth.