Lucrarea 5

Proiectarea traiectoriei roboţilor industriali

Lucrarea işi propune sa prezinte principalele metode de proiectare a traiectoriei pentru roboţii industriali.

5.1 Introducere

Locul geometric al poziţiilor succesive pe care un punct material le ocupŭ în spaţiu în timpul mişcării sale se numeşte *traseul punctului*. Dacă aceste poziţii sunt specificate la momente determinate de timp, locul lor geometric defineşte *traiectoria punctului*. Altfel spus, traiectoria se poate defini ca fiind traseul unui punct supus constrângerilor de timp. Traiectoria de mişcare a unui robot este traiectoria descrisă de centrul sistemului de coordonate mobil, ataşat corpului manipulat.

Proiectarea traiectoriei roboţilor industriali se face în general în sistemul de coordonate carteziene pentru că acest sistem de cooronate este aparent utilizatorului, lucru care facilitează proiectarea traiectoriei. În schimb, probramarea mişcării trebuie să se facă pentru fiecare cuplă a robotului (mişcarea se realizează în coordonate robot - operaţionale). Această transformare de coordonate se face apelând la modelul geometric invers. În general, fiecare robot este prevăzut cu un controller dedicat în care modelul geometric direct şi modelul geometric invers sunt preprogramate.

În functie de modul de realizare a treiectoriei roboţilor industriali, acestia se clasifică în roboţi PTP[1]şi roboţi CP[2].

5.2 Traiectoria punct-cu-punct

Traiectoria roboţilor PTP este memorată sub forma unei mulţimi de puncte din spaţiul de lucru parcurs de robot. Aceste puncte vor fi atinse în decursul execuţiei mişcării, însă nu exista nici un control asupra traiectoriei între aceste puncte. Mişcarea între puncte se realizează după una din urmaătoarele tehnici:

- **Mişcarea neconstrânsă** (eng. - unconstrained joint motion) Toate axele robotului se mişcă în paralel, fiecare la parametri maximi (viteză, acceleraţie). În funcţie de

[1] **PTP** - (eng) Point To Point.
[2] **CP** - (eng) Continuous Path.

distanţa pe care trebuie să o parcurgă fiecare cuplă şi de viteza maximă admisibilă, mişcarea poate dura timpi diferiţi pe diferite axe (mişcarea poate să se termine pe una dintre axe înaintea mişcărilor pe alte axe).

- **Mişcarea cuplă după cuplă** (eng. - joint by joint motion) Axele robotului execută mişcările pe rând (câte o cuplă odată). Traiectoria rezultată depinde de ordinea în care se mişcă axele, şi în general este foarte diferită de linia dreaptă între cele două puncte, putând să apară în unele cazuri chiar reversări ale mişcării pe anumite axe.
- **Mişcarea interpolată liniar** (eng. - linear interpolated joint motion) Se realizează prin mişcarea în paralel a axelor robotului cu viteze (şi eventual acceleraţii) precalculate astfel încât să mişcarea să se termine simultan pe toate axele (axele mai rapide sunt încetinite corespunzător). Mişcarea rezultată este optimă: nu prezintă discontinuităţi şi durata sa este minimă (egală cu durata mişcării in cazul neconstrâns).
- **Traiectorii polinomiale** (eng. - polynomial trajectories) Traiectoria se proiectează folosind polinoame ai căror coeficienţi se determină cu ajutorul restricţiilor impuse pentru poziţie, viteză şi acceleraţie.

5.3 Traiectoria continuă

Traiectoria roboţilor CP este memorată ca o funcţie parametrică şi nu ac o mulţime de puncte. Există un control exact al poziţiei pe tot parcursul traiectoriei. Capacitatea de memorie utilizată este mult mai mică decât în cazul traiectoriei PTP, în schimb volumul de calcul în timp real creşte considerabil.

Obţinerea unei traiectorii în linie dreaptă asemănătoare roboţilor CP se poate face si cu controllere PTP folosind algoritmul lui Taylor. Acesta se bazează pe împărţirea succesivă traiectoriei în segmente despărţite de puncte nod între care se foloseşte mişcarea neconstrânsă sau mişcarea interpolată liniar. Împarţirea se continuă pâna când eroarea în poziţie faţă de linia dreaptă măsurată la mijlocul segmentului este sub o valoare impusă iniţial (eroarea admisă).

Algoritmul lui Taylor

1. Se calculează coordonatele robot (operaționale) pentru punctele Start și Țintă.
2. Se calculează punctul median în spațiul coordonatelor robot și coordonatele carteziene pentru acest punct.
3. Dacă eroarea față de linia dreaptă este mai mare decât deviația impusă atunci:

a. Plasează un punct nod la mijlocul traseului cartezian în linie dreaptă.

b. Împarte traseul în două segmente în jurul punctului nod

c. Apelează recursiv acest algoritm pentru segmentul stâng

d. Apelează recursiv acest algoritm pentru segmentul drept

Altfel treci la pasul 4.

4. Returnează secvența de puncte nod

Modelele mai vechi de controllere pentru roboţi nu permiteau controlul vitezei, presupunând că mişcarea se face cu viteză constantă. Această ipoteză presupune însă acceleraţii infinite la pornire şi oprire, acceleraţii ce nu pot fi realizate practic. În consecinţă robotul nu urmărea exact traiectoria programată, aparând întârzieri faţă de aceasta şi în consecinţă erori staţionare la poziţie. Controllerele moderne înlatură aceste neajunsuri permiţând profile de viteză trapezoidale (triunghiulare) sau parabolice.

5.4 Probleme Propuse

Se consideră structura de manipulare cu două grade de libertate prezentată în figura 5.1.

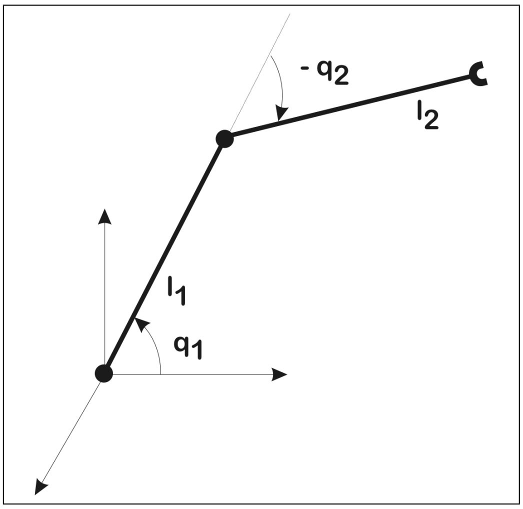

Figura 5.1.

Pentru această structură se determină modelul geometric direct:

$$\begin{cases} X &= 0 \\ Y &= l_2 \cos(q_1 + q_2) + l_1 \cos(q_1) \\ Z &= l_2 \sin(q_1 + q_2) + l_1 \sin(q_1) \end{cases}$$

si modelul geometric invers:

$$\begin{cases} a &= \frac{Y^2+Z^2-l_1^2-l_2^2}{2l_1l_2} \\ q_2 &= \pm \arctan \frac{\sqrt{1-a^2}}{a} \\ q_1 &= \arctan \frac{Z}{Y} - \arctan \frac{l_2 \sin(q_2)}{l_1+l_2\cos(q_2)} \end{cases}$$

1. Să se verifice modelele prezentate.

2. Se consideră următoarele puncte *Start* şi *Ţintă* (în coordonate carteziene):

a) $\begin{cases} S(0, 30, 30\sqrt{3}) \\ T(0, 20\sqrt{3} - 10, 10\sqrt{3} + 20 \end{cases}$

b) $\begin{cases} S(0, 10, 40 + 10\sqrt{3}) \\ T(0, 40, 20\sqrt{3} \end{cases}$

c) $\begin{cases} S(0, 20, 40) \\ T(0, 30\sqrt{3}, 30 \end{cases}$

d) $\begin{cases} S(0, 20 + 20\sqrt{3}, 20) \\ T(0, 20, 20 + 20\sqrt{3} \end{cases}$

Să se scrie un program Matlab pentru reprezentarea grafică a traiectoriei între punctele *Start* şi *Ţintă* considerând cele două mişcări de tip cuplă după cuplă şi apoi o mişcare interpolată liniar.

3. Cunoscând $\begin{cases} \omega_1 = 60°/sec \\ \omega_1 = 60°/sec \\ \varepsilon_1 = 60°/sec^2 \\ \varepsilon_1 = 90°/sec^2 \end{cases}$ vitezele si acceleraţiile unghiulare maxime ale celor două cuple, să se reprezinte grafic viteza unghiulară pe fiecare grad de libertate (pentru traiectoriile de la punctul 2) şi traiectoria rezultată în planul $\mathcal{ZOY}$. Se vor considera următoarele cazuri:

a) Mişcare neconstrânsă

b) Mişcare interpolată liniar

4. Să se scrie un program Matlab care implementează algoritmul lui Taylor. Să se reprezinte grafic în planul $\mathcal{ZOY}$ traiectoriile obţinute între punctele *Start* şi *Ţintă* date.