Laborator 8-9: Cinematica Robotilor

Obiective:

Cunoasterea metodelor cinematice asociate obiectelor SeriaLink: Definirea unui lant cinematic ; Cinematica directa; Cinematica Inversa;Definirea traiectoriilor dorite si simularea acestora; Stabilirea pozitiei de zero.

Utilizarea toolboxului PC in construirea unei structuri personalizate.

Elemente teoretice: Studii de caz

Studiul de caz: Definirea unui robot RR (plan)

(Peter Corke toolbox) Se realizeaza prin utilizarea obiectului Link si a constructorului SerialLink

Definirea elementelor

se definesc utilizand parametrii [tetha,d,a,alpha,sigma]

```
L(1)=Link([0,0,1,0,0]);
L(2)=Link([0,0,1,0,0]);
```

```
L =
 theta=q1, d=0, a=1, alpha=0 (R,stdDH)
 theta=q2, d=0, a=1, alpha=0 (R,stdDH)
```

Construriea lantului cinematic

Lantul cinematic se construieste prin inserierea celor 2 elemente

```
two_link=SerialLink(L,'name','john');
two_link
```

```
two_link =
john (2 axis, RR, stdDH)

+---+-----------+-----------+-----------+-----------+
| j |     theta |         d |         a |     alpha |
+---+-----------+-----------+-----------+-----------+
|  1|         q1|          0|          1|          0|
|  2|         q2|          0|          1|          0|
+---+-----------+-----------+-----------+-----------+

grav =    0  base = 1  0  0  0   tool =  1  0  0  0
          0         0  1  0  0           0  1  0  0
       9.81         0  0  1  0           0  0  1  0
                    0  0  0  1           0  0  0  1
```

Cinematica directa

```
two_link.fkine([0,0])
```

```
ans =
     1     0     0     2
     0     1     0     0
     0     0     1     0
     0     0     0     1
```

Vizualizarea robotului

```
two_link.plot([pi/4,-pi/4])
```

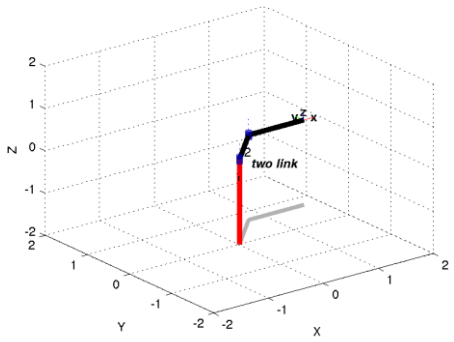


Figura 8.1. Configuratia robotului RR

Studiul de caz: Cinematica inversa

(Peter Corke toolbox) Problema de cinematica inversa este ingreunata de obtinerea solutiilor multipe sau de lipsa solutiilor. Exemplul de caz prezinta cele doua aspecte mentionate pentru un robot binecunoscut Puma 560.

Solutii multiple

```
Pentru robotul Puma 560 se impune configuratia nominala qn (v si exemplul
de caz anterior). Cu ajutorul problemei de cinematic adirecta se determina
postura sculei. Aceasta va fi data de intrare pentru problema de cinemtica
inversa. Toolboxul permite utilizarea metodei analitice de calcul si
obtinerea a 4 solutii in functie de impunerea orientarii "umarului",
cotului" si a articulatiei mainii "carpul" adica a articulatiilor 1,3,5 in
sensul r (spre dreapta), l (spre stanga); u(deasupra), d (dedesupt); f
(rasturnat), n (nerasturnat)
mdl_puma560 % dedinirea obiectului robot
P=p560.fkine(qz);
%cinematica inversa a aceleiasi posturi, impunand cerintele mentionate
q1=p560.ikine6s(P,'run');
q2=p560.ikine6s(P,'rdn');
q3=p560.ikine6s(P,'luf');
% Vizualizarea robotului in cele trei configuratii calculate
figure;p560.plot(q1);title 'RUN'
```

```
figure;p560.plot(q2);title 'RDN'
figure;p560.plot(q3);title 'LUF'
```

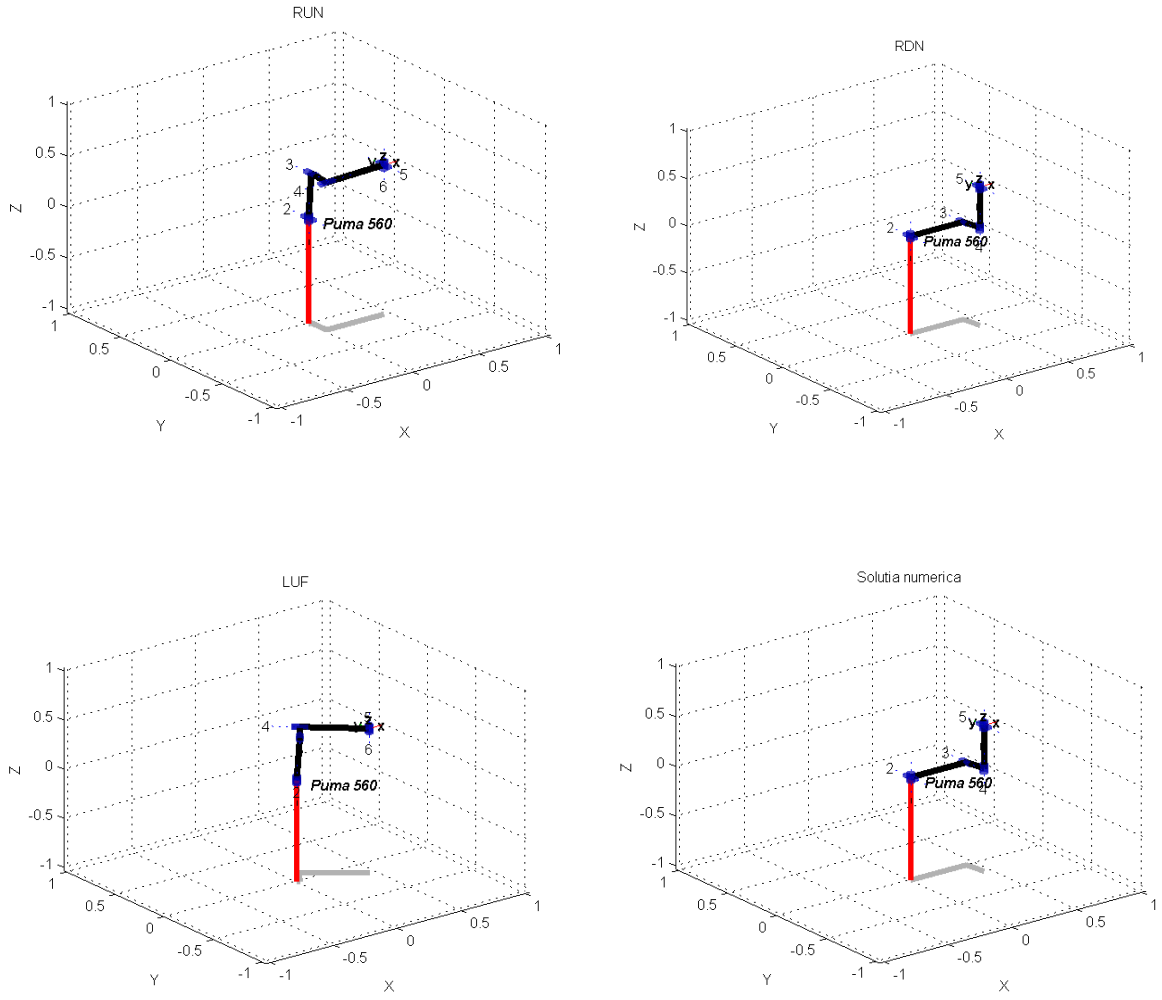


Figura 8.2. Patru configuratii posibile pentru o aceeasi postura a sculei

Solutii inexistente

Posturile care nu se regasesc in volumul de lucru al robotului nu pot fi realizate

```
q4=p560.ikine6s(transl(0,0,1))
```

```
Warning: point not reachable
q4 =
   NaN   NaN   NaN   NaN   NaN   NaN
```

Solutii numerice

Solutiile numerice sunt calculate prin iteratii succesive care pornesc de la o postura initiala (0,0,0,0,0,0) si se indreapta catre postura impusa Din aceat motiv ele nu pot fi controlate in modul in care au fost controlate solutiile analitice.

```
q5=p560.ikine(P);
figure;p560.plot(q5);title 'Solutia numerica'
```

Roboti cu grade de libertate mai mici ca 6 (under-actuated)

Atunci cand numarul gradelor de libertate este mai mic ca 6 (de exemplu robotul SCARA) cinematica inversa a unei posturi generale (cu 6 parametri) nu poate rezolva problema. Pentru a rezolva aceasta problema se precizeaza doar directiile importante pentru problema de cinematica inversa. Mai precis directiile importante sunt acelea care trebuie realizate (X sau Y sau...etc). Numarul directiilor importante trebuie sa fie egal cu numarul gradelor de libertate ale robotului

```
mdl_twolink % construirea robotului twolink
T=transl(0.4,0.5,0.6); %Precizarea posturii
q6=twolink.ikine(T,[0,0],[1,1,0,0,0,0])% ultimul vector precizeaza care sunt directiile importante
% 1 inseamna important; 0 inseamna ne important In consecinta aici
% important este X,Y
```

```
q6 =
   -0.3488    2.4898
```

Roboti redundanti

Atunci cand robotii au mai mult de 6 grade de libertate, apar inca si mai multe configuratii posibile, care conduc la tealizarea unei posturi date. In exemplu de caz se construieste un astfel de robot dintr-o masa X,Y peste care se aseaza robotul Puma 560

```
% Definirea mesei: sunt doua cuple de translatie X,Y (reciproc perpendiculare)
masa=SerialLink([0,0,0,-pi/2,1;-pi/2,0,0,pi/2,1],'base',troty(pi/2),'name','masa');

% rotatia cu pi/2 in jurul lui Y este necesara ptc formalismul D-H impune
% translatiile pe Z; masa insa este in planul XY

%Pregatirea robotului Puma: fata de configuratia implicita se doreste ca
%robotul Puma sa se inalte (pe un piedestal) d=0.3 m
p560.links(1).d =0.1;

%Asamblarea celor doi roboti
roby=SerialLink([masa, p560],'name','roby')

%Stabilirea posturii impuse
T=transl(0.5,1,0.7)*rpy2tr(0,3*pi/4,0);
q7=roby.ikine(T)

% vizualizarea robotului
roby.plot(q7)
```

```
roby =
roby (8 axis, PPRRRRRR, stdDH)

+---+-----------+-----------+-----------+-----------+
| j |     theta |         d |         a |     alpha |
+---+-----------+-----------+-----------+-----------+
|  1|         0|         q1|         0|    -1.571|
|  2|    -1.571|         q2|         0|     1.571|
|  3|        q3|        0.1|         0|     1.571|
|  4|        q4|          0|    0.4318|         0|
|  5|        q5|       0.15|    0.0203|    -1.571|
|  6|        q6|     0.4318|         0|     1.571|
|  7|        q7|          0|         0|    -1.571|
|  8|        q8|          0|         0|         0|
+---+-----------+-----------+-----------+-----------+
```

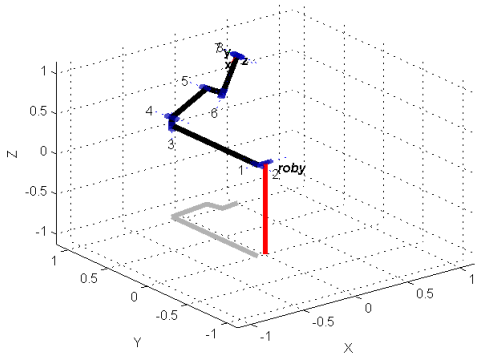


Figura 8.3. Robotul „roby” compus prin inserierea unei mese X,Y cu un robot Puma560

Studiul de caz: Traiectorii

(Peter Corke toolbox) Determinarea unei traiectorii presupune mentionarea (ca date initiale) a punctului de start si a celui de tinta si calcularea (ca date rezultante) a pozitiilor intermediare mentionate intr-o rezolutie de coordonate impusa. Traiectoria poate fi precizata in sistemele de coordonate articulare: atunci cand datele de intrare si cele rezultante sunt pozitiile generalizate; sau in coodonate Carteziene, atunci cand datele mentionate sunt pozitii, unghiuri in spatiul cartezian

```
%Construirea robotului
mdl_puma560
```

Traiectorii in spatiul articular

Pentru a mentiona o traiectorie in spatiu articular este necesara determinarea pozitiilor, vitezelor si acceleratiilor celor N cuple ale robotului. Se porneste de la precizarea posturilor initiale si finale din spatiul cartezian:

```
T1=transl([0.4 0.3 0])*trotx(pi); % o matrice omogena care are componenta de translatie si de
rotatie
T2=transl([0.4 -0.3 0])*trotx(pi/2); %postura finala
% Se determina variabilele unghiulare pentru cele doua posturi (problema de
% cinematica inversa)
q1=p560.ikine6s(T1);
q2=p560.ikine6s(T2);
% Se impune vectorul de timp (timpul in care se doreste realizarea
% traiectorie
t=[0:0.05:2]';
%Determinarea punctelor intermediare
[q,dq,ddq]=jtraj(q1,q2,t);
% Animatia
p560.plot(q)
```

Prezentarea datelor

Prezentarea variatiilor de pozitie, viteza si acceleratie in functie de timp

```
figure;qplot(t,q); title 'Pozitiile';
figure;qplot(t,dq); title 'Vitezele';
figure;qplot(t,ddq); title 'Acceleratiile';
```

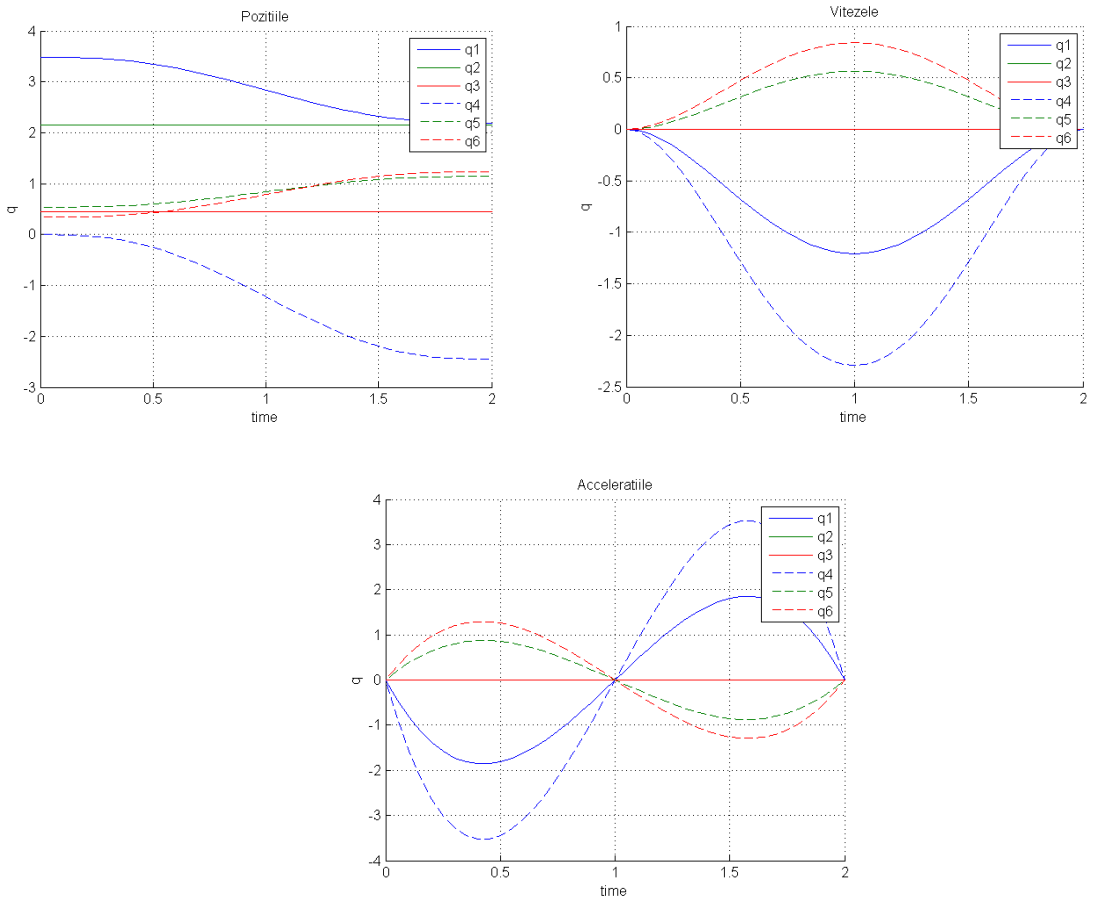


Figura 8.4 Traiectoriile din spatiul articular, calculate prin cinematia inversa a spatiului articular

Traiectoria robotului in XYZ

Este interesant de vazut care e traiectoria sculei atunci cand robotul se de plaseaza utilizand pe coordonatele unghiulare calculate

```
T=p560.fkine(q);
% din matricea omogena a posturilor se extrage vectorul de translatie
P=transl(T); figure;plot(t,P); title 'Transltiile in functie de timp' ; grid
% din matricea omogena a posturilor se extrage vectorul de rotatii
R=tr2rpy(T); figure;plot(t,R); title 'Rotatiile in functie de timp'; grid
% vizualizarea translatiilor
% traiectoria in XOY
figure;plot(P(:,1),P(:,2)); title 'Traiectoria in planul XOY'; axis 'equal'; grid
% se observa ca traiectoria in planul XOY NU ESTE una rectilinie dar ca
% robotul porneste din si ajunge in punctele impuse
```

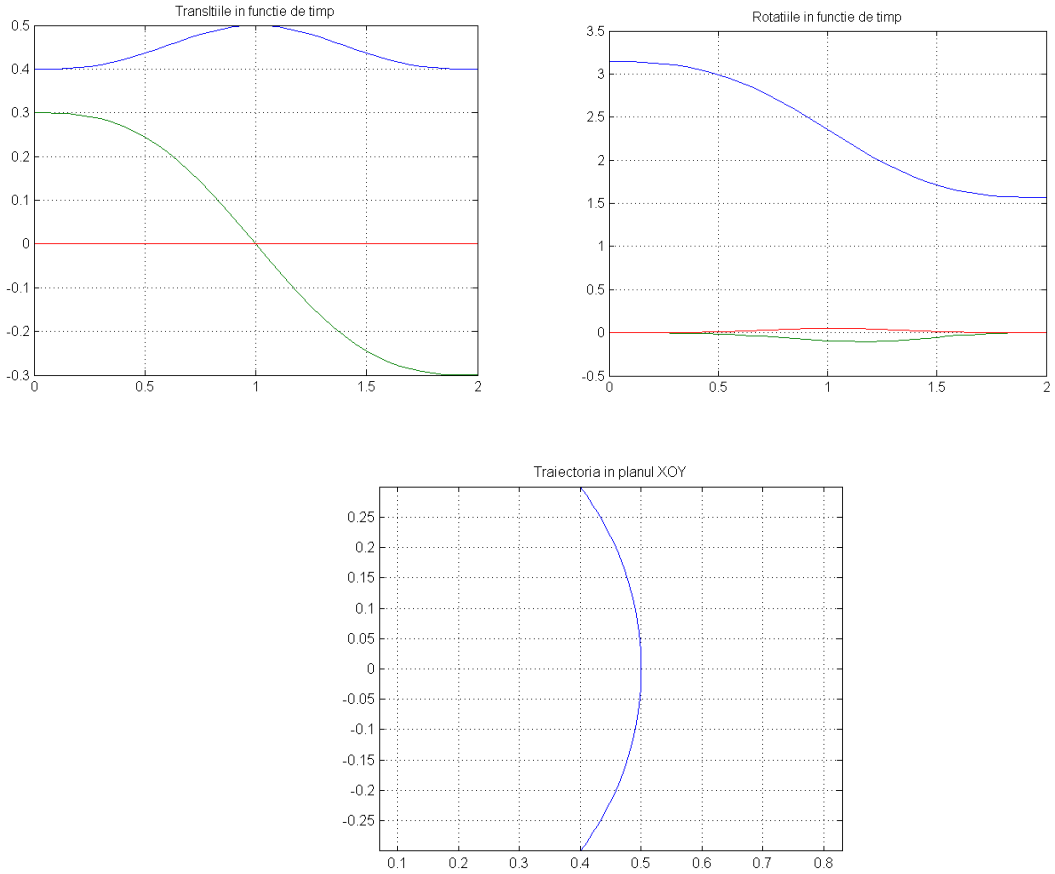


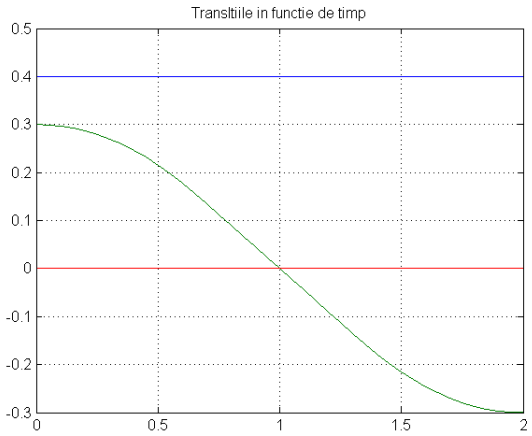
Figura 8.5 Traiectoriile in spatiu Cartezian calculate cu cinematica inversa a spatiului articular

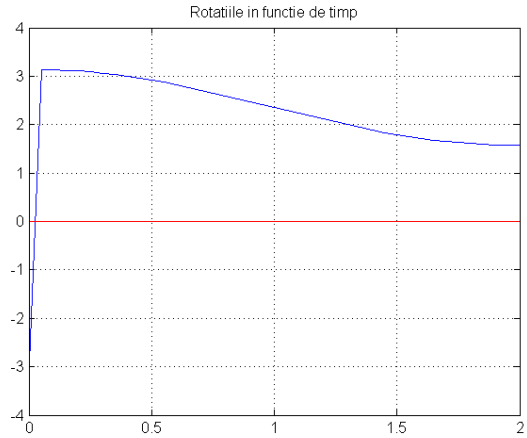
Miscarea Carteziana

Se vor utiliza celeasi puncte de start si de tinta

```
Ts=ctraj(T1,T2,length(t))% determinarea punctelor intermediare in XYZ
% extragerea vectorului de translatie si al celui de rotatie
```

```
figure; plot(t,transl(Ts)); title 'Translatiile'; grid; axis 'equal'
figure; plot(t,tr2rpy(Ts)); title 'Rotatiile'; grid;
% Calculul coordonatelor unghiulare corespunzatoare traiectoriilor prezentate
qc=p560.ikine6s(Ts);
% Utilizarea valorilor determinate in stabilirea posturilor (cinematica
% directa)
Tc=p560.fkine(qc);
% din matricea omogena a posturilor se extrage vectorul de translatie
Pc=transl(Tc); figure;plot(t,Pc); title 'Transltiile in functie de timp' ; grid
% din matricea omogena a posturilor se extrage vectorul de rotatii
Rc=tr2rpy(Tc); figure;plot(t,Rc); title 'Rotatiile in functie de timp'; grid
% vizualizarea translatiilor
% traiectoria in XOY
figure; plot(Pc(:,1),Pc(:,2)); title 'Traiectoria in planul XOY'; axis 'equal'; grid
% se observa ca traiectoria in planul XOY ESTE una rectilinie dar ca
% robotul porneste din si ajunge in punctele impuse
```





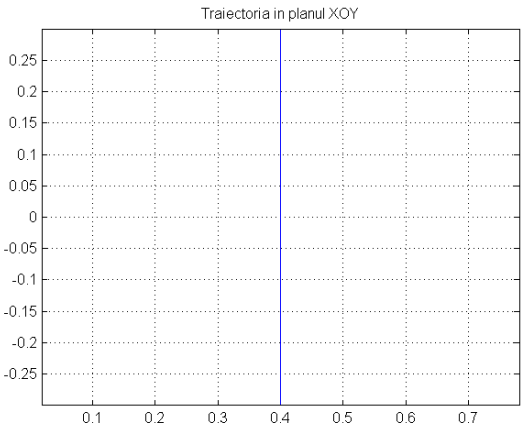


Figura 8.6 Traiectoriile in spatiul cartezian calculate cu cinematica inversa a spatiului cartezian

Trecerea prin singularitati

Singularitatea reprezinta pierderea unuia sa a mai multor grade de mobilitate din cauza suprapunerii unor axe de rotatie Pentru exemplificare se va vrea situatia in care q4 si q6 sunt aliniate Se impun urmatoarele posturi

```
T1=transl([0.5 0.3 0.44])*troty(pi/2);
T2=transl([0.5 -0.3 0.44])*troty(pi/2);
% Se construiesc traiectoriile carteziene
Ts=ctraj(T1,T2,length(t));
% Se determina traiectoriile unghiulare prin metodele analitice
qc=p560.ikine6s(Ts);
% Prezentarea variatiilor de pozitie, viteza si acceleratie in functie de timp
figure;qplot(t,qc); title 'Atentie la saltul din secunda 0.6';
```

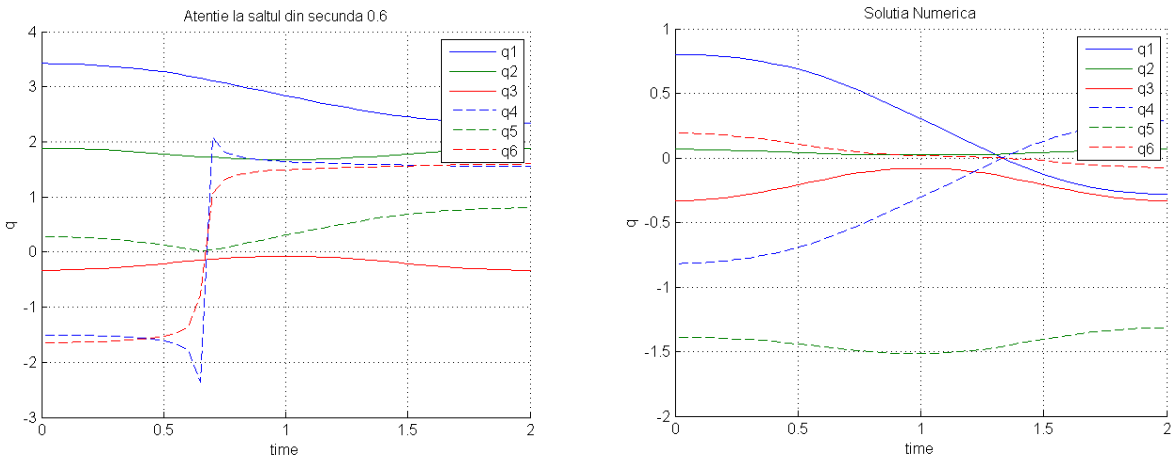


Figura 8.7. Singularitatea obtinuta in spatiu prin cinematica inversa analitica, evitata prin cinematica inversa numerica

Solutia care evita efectul singularitatii

Solutia este cinematica inversa prin metode numerice

```
qcn=p560.ikine(Ts);
figure;qplot(t,qcn); title 'Solutia Numerica';
% Utilizarea metodei numerice trece insa din spatiul cartezian in cel
% articular. Pierderea se manifesta apare la rectilinitatea traiectoriei
```

Studiul de caz: Stabilirea pozitiei de zero

(Peter Corke toolbox) Prin redefinirea structurii se au in vedere urmatoarele subiecte: Stabilirea pozitiei de zero a robotului (prin offset) Determinarea parametrilor DH ai unei structuri care a fost definita cu ajutorul reperului calator; Utilizarea formalismului D-H modificat;

```
%Construirea robotului
mdl_puma560
```

Se realizeaza inca de la nivelul obiectului Link sau la nivelul obiect robot

```
L=Link([0 0 1 0]);
L.offset=pi/4
figure
p560.plot(qz)% reprezentarea robotului nemodificat
title 'configuratia nemodificata'
p560.links(2).offset=pi/2
figure
p560.plot(qz)% reprezentarea robotului nemodificat
title 'configuratia modificata'
```

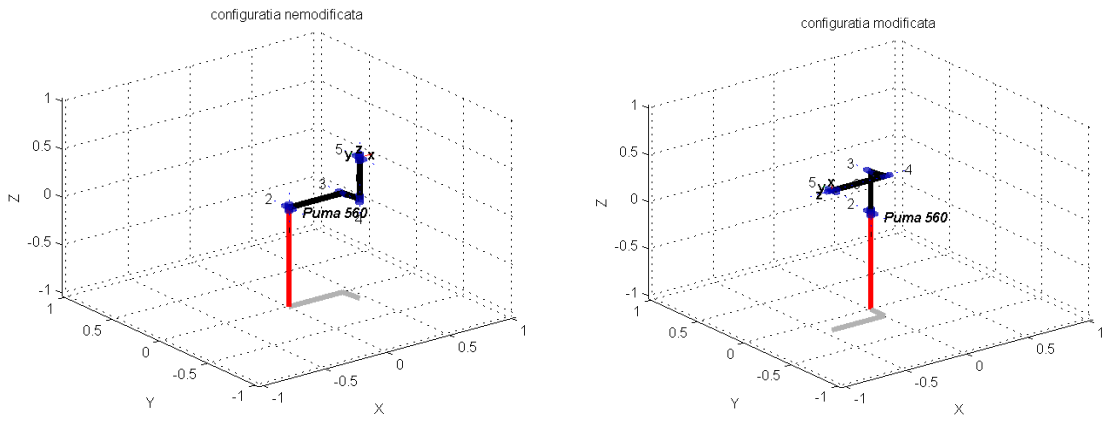


Figura 8.8. Cele doua configuratii obtinute prin modificarea punctului de zero (offset)

Problema propusa 1.

1. Sa se identifice functiile utilizate din pachetul [PC.T] si sa se mentioneze functionalitatea acestora
2. Se cere construirea unei structuri de tipul celei prezentate in figura

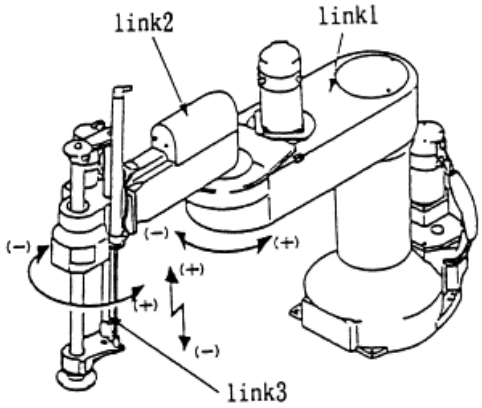


Figura 8.9. Structura propusa [Z.B. Kang[a], T.Y. Chai[a], K. Oshima[b], J.M. Yang[c], S. Fujii[b] **ROBUST VIBRATION CONTROL FOR SCARA-TYPE ROBOTMANIPULATORS, Elvesier Control Engineering Practice** Volume 5, Issue 7, July 1997, Pages 907–917]

Se urmaresc urmatoarele etape: Parametrii necesari rezolvarii acestei probleme sunt definiti de fiecare student in parte; robotu este unul personalizat

- Definirea lantului cinematic:;
- Impunerea unei posturi a efectorului prin Cinematica directa:;
- Determinarea unghiurilor din cuplele robotului atunci cand se cunoaste postura sculei (Cinematica Inversa);
- Definirea traiectoriilor dorite si simularea acestora;