

Estudio comparativo del costo computacional del cálculo cinemático de trayectorias para manipuladores de tres grados de libertad (3 DOF)

Docente

Ing. Hernán Gianetta

Jefe TP

Ing. Damián Granzella

Alumnos

Halamaj, Pablo - Legajo: 113233-7

Campanini, Héctor Manuel - Legajo: 136897-7

Elementos del TP

Para el presente Trabajo Practico se utilizaron 2 arquitecturas de Robot de 3 grados de libertad. Las mismas están basadas en las definiciones del Paper “A comparative Study for Computational Cost of Fundamental Robot Manipulators”.

Para este TP se utilizo el Toolbox “Robotics Toolbox for Matlab (release 9.10)” de Peter Corke.

El área de trabajo elegida corresponde al tipo Esférica y las clases de manipuladores elegidos fueron el SN (estructura de cuerpo del tipo PRR) y el RR (estructura de cuerpo del tipo RPR).

Primer Manipulador (RR)

A continuación se muestran las características del manipulador usado:

```
>> L(1) = Link([180, 0, 1, 0, 0]);
```

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

```
>>L(3) = Link([180, 0, 1, 0, 0]);
```

```
>> treslink = SerialLink(L, 'name', '3link')
```

```
treslink = 3link (3 axis, RPR, stdDH, slowRNE)
```

+---+-----+-----+-----+-----+-----+					
j	theta	d	a	alpha	offset
+---+-----+-----+-----+-----+-----+					
1	q1	0	1	0	0
2	0	q2	1	0	0
3	q3	0	1	0	0
+---+-----+-----+-----+-----+-----+					

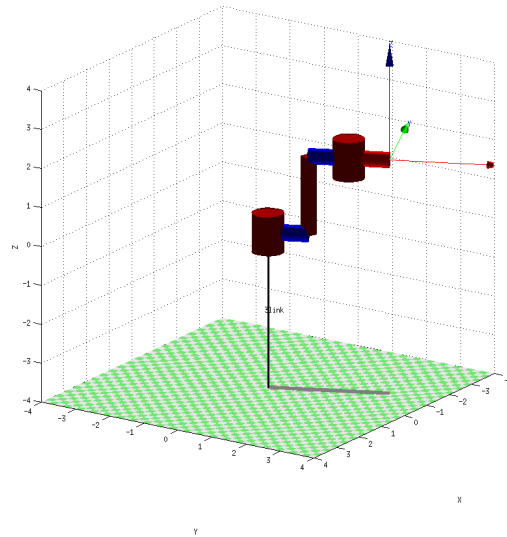
```

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

```

>>

A continuación podemos ver una imagen del Manipulador



Segundo Manipulador (SN)

A continuación se muestran las características del manipulador usado:

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

```
>> L(2) = Link([0, 0, 1, 90, 0]);
```

```
>> L(3) = Link([-90, 0, 1, 0, 0]);
```

```
>> treslink = SerialLink(L, 'name', '3link')
```

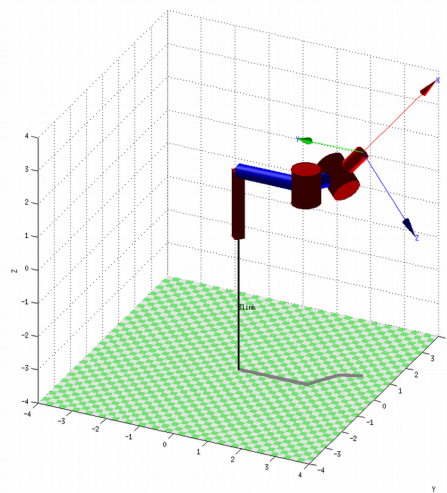
```
treslink = 3link (3 axis, PRR, stdDH, slowRNE)
```

j	theta	d	a	alpha	offset
1	0	q1	2	0	0
2	q2	0	1	90	0
3	q3	0	1	0	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
```

A continuación podemos ver una imagen del Manipulador



Trayectoria Definida

En ambos manipuladores se busco que persigan la misma trayectoria, la cual fue definida con las siguientes instrucciones de Matlab:

```
>>Ts = ctraj(transl(0,0,0), transl(2,2,2),100)
```

Resultados

En ambos casos se utilizaron las funciones “tic” y “toc” de Matlab para obtener una medición mas precisa sobre los tiempos utilizados para el calculo de la cinemática inversa de la trayectoria.

Las instrucciones utilizadas para el calculo fueron:

```
>>tic  
>>qc = treslink.ikine(Ts, [], [1 1 1 0 0 0])  
>>tiempo = toc
```

Los resultados obtenidos fueron:

Tipo de Manipulador	Tiempo Necesario para el Cálculo
SN	2.5 segundos
RR	92.8 segundos

Conclusiones

En el caso del manipulador RR, el toolbox generó múltiples advertencias acerca de que se sobrepasaban las 1000 iteraciones limite para el calculo de los puntos intermedios de la trayectoria.

Esto dió como resultado una trayectoria menos precisa, la cual se pudo ver al utilizar la función “plot3d” sobre el robot y la matriz “qc”, en dicha representación del movimiento se notan cambios abruptos de las posiciones del brazo.

Esto es lo que genera que para la trayectoria presentada el manipulador RR necesita mayor tiempo de calculo.

A continuación se transcriben los archivos de Matlab, como referencia y para reproducir las pruebas aquí escritas.

Primer Manipulador:

```
%Levanto el Toolbox
%startup_rvc
clear
% theta = rotacion entre uniones
% d = distancia entre Uniones
% a = longitud del brazo entre uniones
% alpha = rotacion que tiene el brazo
% sigma = si es union por revolucion o prismatica
% Creo las uniones
L(1) = Link([180, 0, 1, 0, 0])
L(2) = Link([0, 0, 1, 0, 1])
L(3) = Link([180, 0, 1, 0, 0])
% creo el Brazo
treslink = SerialLink(L, 'name', '3link')

% defino el workspace para el ploteo
treslink.plotopt={'workspace',[-4 4 -4 4 -4 4]}
treslink.plot([2 0 3])

% creo la trayectoria entre 0,0,0 y 2,2,2 con 100 Puntos
Ts = ctraj(transl(0,0,0), transl(2,2,2),100);
% le calculo la cinematica inversa
tic
qc = treslink.ikine(Ts, [], [1 1 1 0 0 0])
tiempo = toc
treslink.plot(qc)
```

Segundo Manipulador:

```
% levanto el Toolbox
%startup_rvc
```

```

clear
% theta = rotacion entre uniones
% d = distancia entre Uniones
% a = longitud del brazo entre uniones
% alpha = rotacion que tiene el brazo
% sigma = si es union por revolucion o prismatica
% Creo las uniones
L(1) = Link([0, 0, 2, 0,1]);
L(2) = Link([0, 0, 1, 90,0]);
L(3) = Link([-90, 0, 1, 0,0]);
% creo el Brazo
treslink = SerialLink(L, 'name', '3link')
% defino el workspace para el ploteo
treslink.plotopt={'workspace',[-4 4 -4 4 -4 4]}
treslink.plot([0 0 0])

% creo la trayectoria entre 0,0,0 y 2,2,2 con 100 Puntos
Ts = ctraj(transl(0,0,0), transl(2,2,2),100)
% le calculo la cinemática inversa
tic
qc = treslink.ikine(Ts, [], [1 1 1 0 0 0])
tiempo = toc
treslink.plot(qc)

```
