

Implementación de una matriz cinemática en DSP

Introducción a la cinemática de un robot

La cinemática de un robot se encarga de estudiar el movimiento del mismo con respecto a un sistema de referencia.

Se tienen dos problemas a resolver dentro de la cinemática del robot:

- Problema cinemático directo (PCD).
- Problema cinemático inverso (PCI).

PCD

Al robot lo podemos considerar como una cadena cinemática formada por objetos rígidos o eslabones unidos entre sí mediante articulaciones, por lo tanto se puede establecer un sistema de referencia fijo situado en la base del robot y describir la localización de cada uno de los eslabones con respecto a dicho sistema de referencia. De esta manera encontrando una matriz homogénea de transformación T , la cual relacione la posición y orientación del extremo del robot respecto al sistema de referencia fijo, podremos solucionar este inconveniente.

Básicamente se utilizan tres métodos:

- Relaciones geométricas: utilizado para robots de pocos grados de libertad, método no sistemático
- Matrices de transformación homogénea: método sistemático.
- Cuaternios: se utiliza a éstos de manera equivalente a las matrices.

PCI

El objetivo de este problema es encontrar los valores que deben adoptar las coordenadas articulares del robot para que su extremo se posicione y oriente de acuerdo a una determinada localización espacial, por lo que la obtención de las ecuaciones va a ser fuertemente dependiente de la configuración del robot.

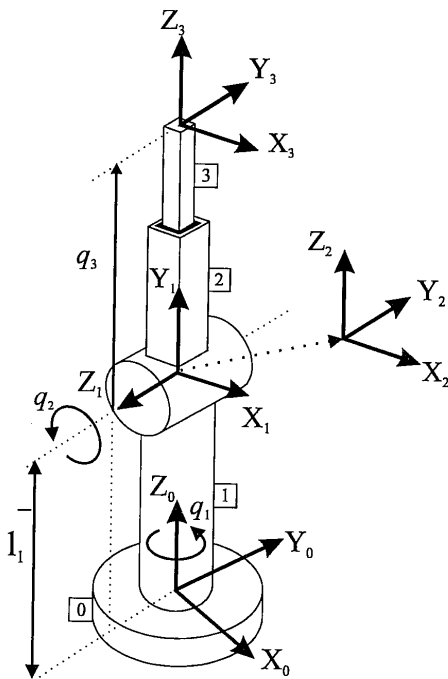
Para este caso se suelen utilizar los siguientes métodos de resolución:

- Métodos geométricos: utilizado para robots de pocos grados de libertad, o para casos donde se consideren sólo los primeros grados de libertad dedicados a posicionar el extremo.
- Matriz de transformación homogénea: consiste en obtener la matriz T , y a partir de ella elegir las ecuaciones correspondientes donde se obtendrán las variables articulares.

Desarrollo e implementación en CW

Implemente el código C en CW para el DSP56800/E de la cadena cinemática directa de la figura, usando la matriz homogénea, y usando como set-point, una trayectoria lineal continua a cada eje. Defina los límites y área de trabajo del manipulador.

Articulación	θ	d	a	α
1	q_1	L_1	0	90
2	q_2	0	0	-90
3	0	q_3	0	0



El sistema de referencia inicial será el (X_0, Y_0, Z_0) , donde se tendrá $q_1 = q_2 = 0$.

A continuación se planteará el método de la matriz homogénea de manera de resolver el problema cinemático directo. Para ello se realizarán las matrices 0A_1 , 1A_2 y 2A_3 que permiten calcular la matriz T.

$$[T] = [{}^0A_1] * [{}^1A_2] * [{}^2A_3]$$

$${}^0A_1 = \begin{bmatrix} C1 & 0 & S1 & 0 \\ S1 & 0 & -C1 & 0 \\ 0 & 1 & 0 & L1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1A_2 = \begin{bmatrix} C2 & 0 & -S2 & 0 \\ S2 & 0 & C2 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^2A_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & q3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\longrightarrow [T] = \begin{bmatrix} C1C2 & -S1 & -C1S2 & -q3C1S2 \\ S1C2 & C1 & -S1S2 & -q3S1S2 \\ S2 & 0 & C2 & q3C2+L1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = [T] \begin{bmatrix} q1 \\ q2 \\ q3 \\ 1 \end{bmatrix} \longrightarrow \text{De aquí determinamos X, Y, y Z}$$

$$X = (\text{Cos}q1\text{Cos}q2)q1 - (\text{Sen}q1)q2 - (\text{Cos}q1\text{Sen}q2)q3 - (q3\text{Cos}q1\text{Sen}q2)$$

$$Y = (\text{Sen}q1\text{Cos}q2)q1 + (\text{Cos}q1)q2 - (\text{Sen}q1\text{Sen}q2)q3 - (q3\text{Sen}q1\text{Sen}q2)$$

$$Z = (\text{Sen}q2)q1 + (\text{Cos}q2)q3 + q3\text{Cos}q2 + l1$$

A continuación se verá el código en "C" que se propuso de manera de obtener la posición final del robot junto con la trayectoria que describe.

```

/** #####
** Filename : TP1_robotica.C
** Project  : TP1_robotica
** Processor : 56F8367
** Version  : Driver 01.13
** Compiler  : Metrowerks DSP C Compiler
** Date/Time : 28/05/2009, 15:50
** Abstract  :
**   Main module.
**   This module contains user's application code.
** Settings  :
** Contents  :
**   No public methods
**
** (c) Copyright UNIS, a.s. 1997-2008
** UNIS, a.s.
** Jundrovská 33
** 624 00 Brno
** Czech Republic
** http    : www.processorexpert.com
** mail    : info@processorexpert.com
**
#####
*/
/* MODULE TP1_robotica */

```

```

/* Including needed modules to compile this module/procedure */
#include "Cpu.h"
#include "Events.h"
#include "MFR1.h"
#include "MEM1.h"
#include "TFR1.h"

```

```

/* Including shared modules, which are used for whole project */

```

```

#include "PE_Types.h"
#include "PE_Error.h"
#include "PE_Const.h"
#include "IO_Map.h"

void main(void)
{
    /* Write your local variable definition here */

    Frac16 q1,q2,q3,ta,tb,Va,Vb,x,y,z;

    /*** Processor Expert internal initialization. DON'T REMOVE THIS CODE!!! ***/
    PE_low_level_init();
    /*** End of Processor Expert internal initialization.          ***/

    /* Write your code here */

    for(;;) {

        q2=0;
        q3=20;
        for(q1=0;q1<20;q1++) {

            q2=q2+1;
            q3=q3+1;

            Va=TFR1_tfr16SinPIx(q1);
            Vb=TFR1_tfr16SinPIx(q2);

            ta=TFR1_tfr16CosPIx(q1);
            tb=TFR1_tfr16CosPIx(q2);

            x=L_sub(L_add(L_add(L_sub(L_mult(L_mult(ta,tb),q1),L_mult(Va,q2),L_mult(ta,q3))),
                L_mult(L_mult(q3,c1),Vb)));

            y=L_sub(L_add(L_mult(L_mult(Va,tb),q1),L_mult(ta,q2)),L_sub(L_mult(Va,q3),
                L_mult(L_mult(q3,Va),Vb)));

            z=L_add(L_sub(L_add(L_mult(Vb,q1),L_mult(tb,q3)),L_mult(q3,tb)),10);

            printf("%d \t %d \t %d",x,y,z);

        }
    }

    /* END TP1_robotica */
    /*

```

**

#####

**

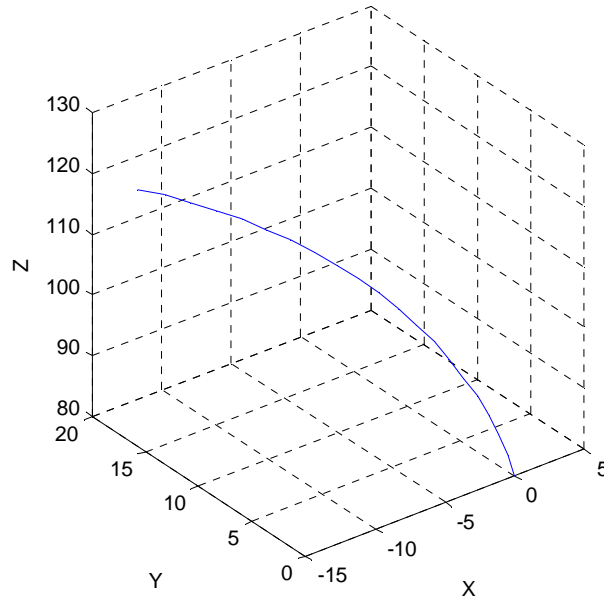
** This file was created by UNIS Processor Expert 2.99 [04.17]
** for the Freescale 56800 series of microcontrollers.
**

** #####
*/

Valores que se obtuvieron:

X	Y	Y
0	0	80
0,249353576	1,004504811	82,0110556
0,393122634	2,014947233	84,04299538
0,430620319	3,026684893	86,09396647
0,361355819	4,035036053	88,16210031
0,185036227	5,03528776	90,24551362
-0,098431939	6,022704064	92,34230934
-0,488941543	6,992534306	94,45057759
-0,986183987	7,940021459	96,56839666
-1,589648699	8,860410513	98,69383394
-2,298622973	9,74895687	100,824947
-3,112192145	10,60093477	102,9597843
-4,029240122	11,41164571	105,0963867
-5,04845025	12,17642682	107,232788
-6,168306523	12,89065926	109,3670159
-7,387095148	13,54977653	111,4970935
-8,702906436	14,14927271	113,6210398
-10,11363704	14,68471072	115,7368709
-11,61699254	15,15173035	117,8426011
-13,21049031	15,54605634	119,9362438
-14,89146282	15,86350624	122,0158125

Por medio de la función `plot3(x,y,z)` de matlab se obtuvo la siguiente gráfica:



Conclusión: La importancia de esta práctica es resaltar como se puede implementar el movimiento de un robot partiendo de su sistema de referencia, con su número de articulaciones y planteando una serie de límites; para obtener una serie de puntos que describen la trayectoria del robot.

De esta misma manera obtenemos con el CW el diseño de cómo será nuestro DSP final.

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES
Departamento de Electrónica

*Implementación de una matriz
cinemática en DSP*

Materia: Robótica

Profesor: Ing. Hernán Giannetta

Cátedra: R6055 Grupo:

Alumnos: Espinoza, Mario

David de Lima, Darío

8 de Junio de 2009