

# **Planeación de trayectoria de un robot serial**

Robótica

Marcoc-rasi

## Apéndice

Contenido del reporte .....	1
Objetivo .....	1
Planteamiento del modelo.....	1
Modelo de la trayectoria .....	2
Modelo del robot .....	5
Establecimiento de una prueba.....	9
Simulación .....	10
Resultado.....	20
Conclusiones.....	25

## Contenido del reporte

### Objetivo

- Comparar el comportamiento cinemático de las velocidades y de los pares de las juntas de un robot RRR en el plano, al implementar una simulación de una planeación de movimientos en espacio de las juntas del robot.
- Cambiar los archivos de la simulación a través de Simulink, para meter un diseño propio, diferente al propuesto por el profesor.

### Planteamiento del modelo

## Modelo de la trayectoria

### ***Planeación de movimientos en el espacio de trabajo de un robot***

La planeación del espacio de las juntas consiste en determinar las posturas que debe adoptar un robot para seguir una trayectoria determinada, para ello se parte del modelo cinemático.

$$\mathbf{x} = \mathbf{f}(q) \quad \mathbf{q} = \mathbf{g}(x)$$

No siempre es posible establecer el conjunto de funciones anterior, por lo que el planteamiento del cálculo de la solución se establece mediante en establecer un modelo en función de las restricciones cinemáticas de la pose de la tarea.

$$\mathbf{F}(X, q) = \mathbf{x} - \mathbf{f}(q) = 0$$

$$\mathbf{F}(X, q) = \mathbf{f}(q) - \mathbf{x} = 0$$

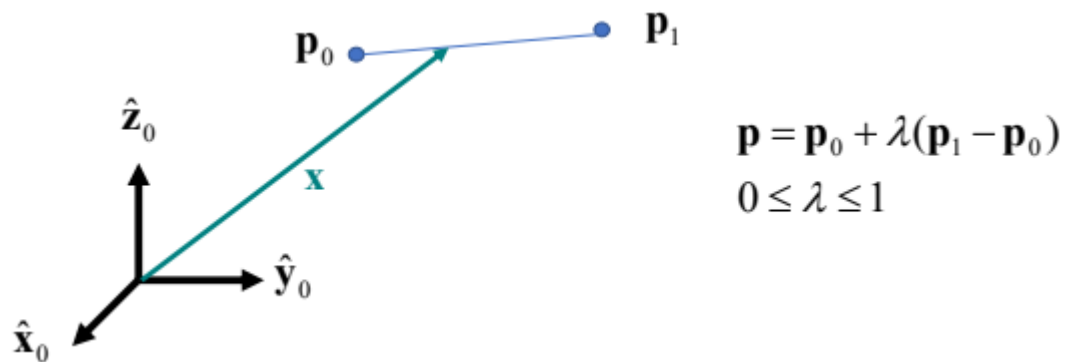
El vector de pose  $\mathbf{x}$  del robot consiste en expresar los requerimientos de la tarea en una trayectoria.

$$\mathbf{x} = \begin{pmatrix} x_1(t) \\ \vdots \\ x_m(t) \end{pmatrix}$$

Una trayectoria es una sucesión por los cuales se desplazar un cuerpo en un periodo de tiempo determinado.

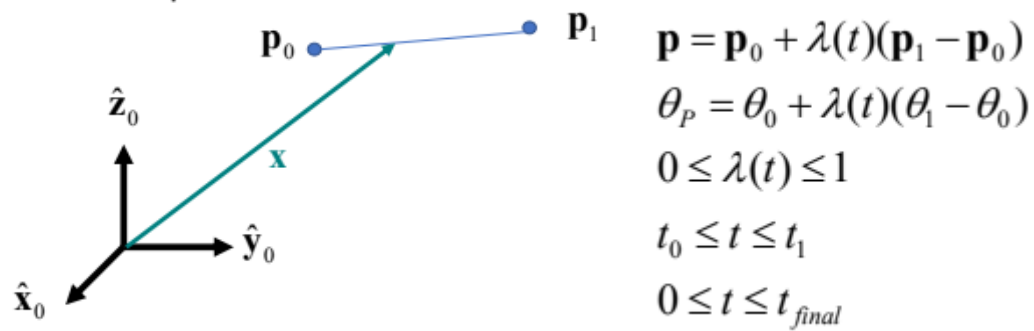
- Lugar geométrico.

Es una ecuación que describe una la relación continua de una serie de puntos en el espacio.



- Perfil de trayectoria.

Es una ecuación paramétrica en términos del tiempo la cual nos indica la evolución del vector  $\mathbf{x}$  con respecto al tiempo de ejecución de una tarea.



En esta ocasión la solución para el perfil de trayectoria fue otorgada por el profesor.

$$\mathbf{q} = \begin{pmatrix} q_{l_0} + \lambda(t) \cdot (q_{l_1} - q_{l_0}) \\ \vdots \\ q_{n_0} + \lambda(t) \cdot (q_{n_1} - q_{n_0}) \end{pmatrix}$$

$$\begin{aligned} \lambda(t) & \quad \lambda(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \\ 0 \leq \lambda(t) \leq 1 & \quad \dot{\lambda}(t) = a_1 + 2a_2 t + 3a_3 t^2 + 4a_4 t^3 + 5a_5 t^4 \\ & \quad \ddot{\lambda}(t) = 2a_2 + 6a_3 t + 12a_4 t^2 + 20a_5 t^3 \end{aligned}$$

$t = 0$	$t = t_f$
$\lambda(0) = 0 = a_0$	$\lambda(t_f) = 1$
$\dot{\lambda}(0) = 0 = a_1$	$\dot{\lambda}(t_f) = 0$
$\ddot{\lambda}(0) = 0 = a_2$	$\ddot{\lambda}(t_f) = 0$

En este caso como vemos planteamos nuestra ecuación, el perfil de trayectoria  $\lambda$ , es una función en la que su solución siempre va a valer entre 0 y 1, esto nosotros lo establecemos, así sabremos si el robot es capaz de llegar al lugar propuesto o no, como sabemos las condiciones finales e iniciales ( $a_1, a_2, a_3$ ), en este caso de acuerdo a nuestro robot, planteamos la solución al sistema, donde se obtuvo el valor de  $a_3, a_4$  y  $a_5$ , para plantear la solución se encontró la siguiente matriz a partir de las condiciones finales e iniciales

$$\begin{aligned} t &= t_f \\ \lambda(t_f) &= 1 = a_3 t_f^3 + a_4 t_f^4 + a_5 t_f^5 \\ \dot{\lambda}(t_f) &= 0 = 3a_3 t_f^2 + 4a_4 t_f^3 + 5a_5 t_f^4 \\ \ddot{\lambda}(t_f) &= 0 = 6a_3 t_f + 12a_4 t_f^2 + 20a_5 t_f^3 \\ \begin{pmatrix} t_f^3 & t_f^4 & t_f^5 \\ 3t_f^2 & 4t_f^3 & 5t_f^4 \\ 6t_f & 12t_f^2 & 20t_f^3 \end{pmatrix} \begin{pmatrix} a_3 \\ a_4 \\ a_5 \end{pmatrix} &= \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \end{aligned}$$

Se resolvió usando mathematica, en donde obtuvimos que  $a_1, a_2$  y  $a_3$  tienen este valor.

$$\text{Out}[3]=\left\{\frac{10}{t_f^3}, -\frac{15}{t_f^4}, \frac{6}{t_f^5}\right\}$$

Al sustituir este valor dentro nuestra ecuación obtenemos.

$$t = t_f$$

$$\lambda(t) = \frac{10}{t_f^3}t^3 - \frac{15}{t_f^4}t^4 + \frac{6}{t_f^5}t^5$$

$$\dot{\lambda}(t) = \frac{30}{t_f^3}t^2 - \frac{60}{t_f^4}t^3 + \frac{30}{t_f^5}t^4$$

$$\ddot{\lambda}(t) = \frac{60}{t_f^3}t - \frac{180}{t_f^4}t^2 + \frac{120}{t_f^5}t^3$$

De esta manera obtenemos nuestra ecuación de perfil de trayectoria hecha con un perfil quintico.

#### Modelo del robot

- Modelo de la postura.

Los robots seriales tienen como objetivo mover un objeto o herramienta de un lugar a otro dentro de su espacio de trabajo para realizar alguna tarea mediante una serie de movimientos programados. El modelo de un robot serial de la configuración consiste en establecer una ecuación que describa la manera en que sus eslabones están dispuestos en función de la postura de cada una de sus juntas.

El modelo cinemático de la postura del robot describe la posición y la orientación del efector final (EF) relacionado con el sistema de referencia  $\{P\}$ , con respecto a las relaciones de posición y orientación presentes entre los eslabones con respecto a un sistema de referencia fijo  $\{0\}$ .

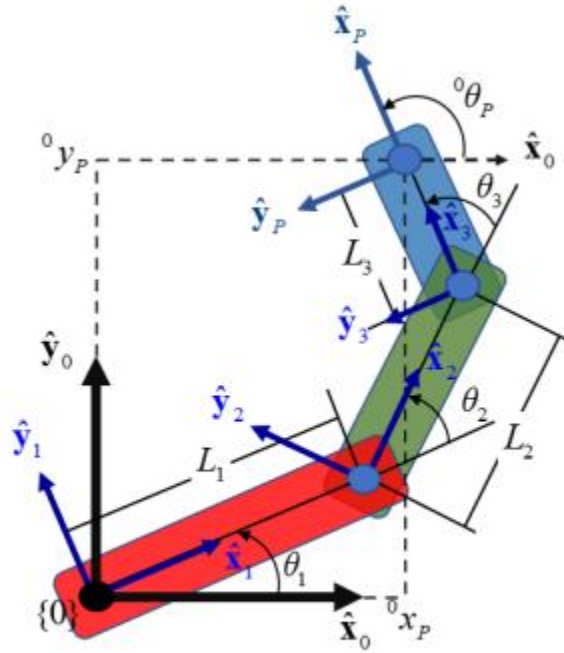


Figura 3-0-1. Planteamiento de un robot serial RRR en un plano  $x_0y_0$ .

El planteamiento de la ecuación se establece al asociar sistemas de referencias relativos asociados los cuales servirán para establecer las relaciones de posición y de movimiento de los eslabones cuando el robot ejecute una tarea mover un material o herramienta de un lugar a otro al posicionar su efector final. Para establecer el modelo de la postura del efector final en función de las relaciones de posición y orientación presentes entre los eslabones que conforman a un robot en un plano se utiliza la siguiente matriz de transformación homogénea

$${}^i\mathbf{T}_j = \begin{pmatrix} {}^i\mathbf{R}_j & {}^i\mathbf{p}_j \\ \mathbf{0}^T & 1 \end{pmatrix}$$

La matriz de transformación, está compuesta por la matriz de rotación  ${}^i\mathbf{R}_j$  que proyecta las direcciones unitarias de un sistema de referencia cartesiano relativo  $\{j\}$  sobre un sistema de referencia cartesiano anterior  $\{i\}$ , un vector  ${}^i\mathbf{p}_j$  que describe la posición del origen del sistema de referencia  $\{j\}$  con respecto al origen de un sistema de referencia anterior  $\{i\}$ , además de un vector de perspectiva, que

para este caso es un vector nulo,  $\mathbf{0}^T$  y de un factor de escala homogéneo. Para este caso, debido a que todos los eslabones están en sobre un plano, la transformación que se utilizará para describir las relaciones de posición de las juntas del robot y la orientación de los eslabones será la siguiente:

$${}^i\mathbf{T}_j = \begin{pmatrix} \cos({}^i\theta_j) & -\sin({}^i\theta_j) & 0 & {}^ix_j \\ \sin({}^i\theta_j) & \cos({}^i\theta_j) & 0 & {}^iy_j \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

El modelo de la postura se establece a partir de los elementos que componen a la transformación  ${}^0\mathbf{T}_P$ , que describe las relaciones de posición y orientación del EF con respecto al sistema inercial {0}.

El planteamiento de la transformación  ${}^0\mathbf{T}_P$  está compuesta por un conjunto de transformaciones homogéneas que describen las relaciones de posición y orientación entre cada una de las juntas asociadas a los eslabones.

$${}^0\mathbf{T}_P = {}^0\mathbf{T}_1 {}^1\mathbf{T}_2 {}^2\mathbf{T}_3 {}^3\mathbf{T}_P$$

La transformación  ${}^0\mathbf{T}_1$ , describe las relaciones de posición y orientación del eslabón L1 unida a una junta rotacional R las cuales se establecen entre el sistema de referencia {1} y el sistema de referencia {0}. Esta transformación se compone de una matriz  ${}^0\mathbf{R}_1$  que se compone de las proyecciones de los vectores directores homogéneos que componen al sistema {1} sobre los vectores homogéneos que componen al sistema de referencia {0}, estas proyecciones se establecen mediante el ángulo  $\Theta_1$  la entrada del actuador relacionada con esta junta. En el caso del vector de posición  ${}^0\mathbf{p}_1$  describe la ubicación del origen del sistema {1} con respecto al origen del sistema {0}, que para este caso es un vector nulo ya que comparten el mismo punto en el plano.

Básicamente esta de ficción se repite entre todos los sistemas de referencia de donde obtenemos las siguientes ecuaciones.

$$\begin{aligned}
 {}^0\mathbf{R}_1 &= \begin{pmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 \\ 0 & 0 & 1 \end{pmatrix} & {}^1\mathbf{p}_2 &= \begin{pmatrix} L_1 \\ 0 \\ 0 \end{pmatrix} & (3.7) & {}^2\mathbf{R}_3 &= \begin{pmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 \\ \sin(\theta_3) & \cos(\theta_3) & 0 \\ 0 & 0 & 1 \end{pmatrix} \\
 {}^0\mathbf{p}_1 &= \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} & {}^1\mathbf{R}_2 &= \begin{pmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 \\ \sin(\theta_2) & \cos(\theta_2) & 0 \\ 0 & 0 & 1 \end{pmatrix} & {}^2\mathbf{p}_3 &= \begin{pmatrix} L_2 \\ 0 \\ 0 \end{pmatrix} \\
 {}^0\mathbf{T}_1 &= \begin{pmatrix} {}^0\mathbf{R}_1 & {}^0\mathbf{p}_1 \\ \mathbf{0}^T & 1 \end{pmatrix} = \begin{pmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} & {}^1\mathbf{T}_2 &= \begin{pmatrix} {}^1\mathbf{R}_2 & {}^1\mathbf{p}_2 \\ \mathbf{0}^T & 1 \end{pmatrix} = \begin{pmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & L_1 \\ \sin(\theta_2) & \cos(\theta_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} & {}^2\mathbf{T}_3 &= \begin{pmatrix} {}^2\mathbf{R}_3 & {}^2\mathbf{p}_3 \\ \mathbf{0}^T & 1 \end{pmatrix} = \begin{pmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & L_2 \\ \sin(\theta_3) & \cos(\theta_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
 {}^3\mathbf{R}_P &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} & {}^3\mathbf{p}_P &= \begin{pmatrix} L_3 \\ 0 \\ 0 \end{pmatrix} \\
 {}^3\mathbf{T}_P &= \begin{pmatrix} {}^3\mathbf{R}_P & {}^3\mathbf{p}_P \\ \mathbf{0}^T & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & L_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}
 \end{aligned}$$

Por lo que si resolvemos nuestra ecuación que relaciona estas ocasiones obtenemos

$$\begin{aligned}
 {}^0\mathbf{T}_P &= {}^0\mathbf{T}_1 {}^1\mathbf{T}_2 {}^2\mathbf{T}_3 {}^3\mathbf{T}_P = \\
 &= \begin{pmatrix} \cos(\theta_1 + \theta_2 + \theta_3) & -\sin(\theta_1 + \theta_2 + \theta_3) & 0 & L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) + L_3 \cos(\theta_1 + \theta_2 + \theta_3) \\ \sin(\theta_1 + \theta_2 + \theta_3) & \cos(\theta_1 + \theta_2 + \theta_3) & 0 & L_1 \sin(\theta_1) + L_2 \sin(\theta_1 + \theta_2) + L_3 \sin(\theta_1 + \theta_2 + \theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}
 \end{aligned}$$

A partir de la transformación  ${}^0\mathbf{T}_P$  se establece el vector de la postura  ${}^0\xi_P$  del efector final, el cual se compone del vector de posición  ${}^0\mathbf{p}_P$  y el vector orientación  ${}^0\boldsymbol{\theta}_P$  del sistema {P} con respecto al sistema inercial {0}.

$${}^0\xi_P = \begin{pmatrix} {}^0\mathbf{p}_P \\ {}^0\boldsymbol{\theta}_P \end{pmatrix}$$



En la Figura 3-0-2, muestra los elementos que componen al vector de postura del efector final, donde se muestra al vector  ${}^0\mathbf{p}_P^T = (x_P \ y_P)$  y el ángulo de orientación  $\theta_P$ .

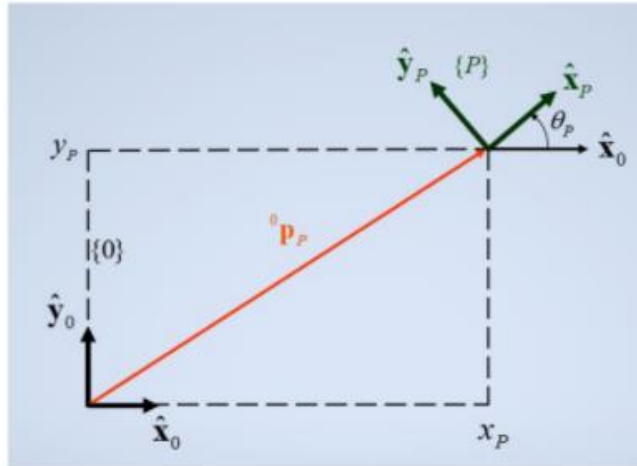


Figura 3-0-2. Postura del sistema  $\{P\}$ , compuesta por el vector de posición  ${}^0\mathbf{p}_P$  y el ángulo de orientación  $\theta_P$ .

Por lo que el vector de postura  ${}^0\boldsymbol{\xi}_P$  de la cadena cinemática RRR se establece de la siguiente forma:

$${}^0\boldsymbol{\xi}_P = \begin{pmatrix} x_P \\ y_P \\ \theta_P \end{pmatrix} = \begin{pmatrix} L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) + L_3 \cos(\theta_1 + \theta_2 + \theta_3) \\ L_1 \sin(\theta_1) + L_2 \sin(\theta_1 + \theta_2) + L_3 \sin(\theta_1 + \theta_2 + \theta_3) \\ \theta_1 + \theta_2 + \theta_3 \end{pmatrix}$$

### Establecimiento de una prueba

En nuestro programa una simulación de una planeación de trayectorias en el espacio de juntas de un robot RRR, en la cual moverá su efector final de un punto a otro dentro de su espacio de trabajo con una determinada orientación.

De la simulación de la planeación de movimientos en el espacio de las juntas el alumno obtendremos las gráficas del comportamiento cinemático de las velocidades y de los pares obtenidos en cada una de las juntas.

Programaremos una simulación de una trayectoria en el espacio de trabajo de un robot RRR siguiendo una trayectoria de una línea recta moviendo y orientando su efector final, bajo las mismas condiciones de la simulación anterior.

De la simulación de la trayectoria en el espacio de trabajo del robot obtendremos las gráficas del comportamiento cinemático de las velocidades y de los pares obtenidos en cada una de las juntas. Y al final compararemos el comportamiento de los dos tipos de simulaciones.

## Simulación

Para la simulación se introdujeron las ecuaciones dentro de 4 archivos, el primero en el cual se muestra la solución a la simulación de las juntas “Sim\_juntas\_1”, el cual contiene el siguiente código

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%Resolución de la configuración inicial de un robot
```

```
%Parámetros del robot
```

```
L1 = 0.12; %m
```

```
L2 = 0.12; %m
```

```
L3 = 0.071213; %m
```

```
x0 = 0.055; %m
```

```
y0 = 0.0; %m
```

```
%Posición y orientación inicial
```

```
xp_inicial = 0.1; %m
```

```
yp_inicial = 0.1; %m
```

```
thetap_inicial = pi/2; %rad
```

```
%Posición y orientación final
```

```
xp_final = 0.2; %m
```

```
yp_final = -0.05; %m
```

```
thetap_final = 0; %rad
```

```
%Determinación de la coordenada x3 y y3 en la configuración inicial
```

```

x3_inicial = xp_inicial-L3*cos(thetap_inicial);
y3_inicial = yp_inicial-L3*sin(thetap_inicial);

%Determinación de la coordenada x3 y y3 en la configuración inicial

x3_final = xp_final-L3*cos(thetap_final);
y3_final = yp_final-L3*sin(thetap_final);

%Cálculo de la postura inicial

%Definición de las variables
syms theta1_in theta2_in theta1_fin theta2_fin

%Cálculo de la postura inicial
[theta1_in,theta2_in] =
solve(x0+L1*cos(theta1_in)+L2*cos(theta1_in+theta2_in)==x3_inicial,y0+L1*
sin(theta1_in)+L2*sin(theta1_in+theta2_in)==y3_inicial,[theta1_in,theta2_
in]);

%Cálculo de la postura final
[theta1_fin,theta2_fin] =
solve(x0+L1*cos(theta1_fin)+L2*cos(theta1_fin+theta2_fin)==x3_final,y0+L1*
sin(theta1_fin)+L2*sin(theta1_fin+theta2_fin)==y3_final,[theta1_fin,thet
a2_fin]);

%%%%%Simulación numerica %%%%%%%%%%%

%Duración del tiempo de simulación

T = 10; %seg tiempor total de la tarea
muestreo = 20;
intiempo = 10/muestreo;

%segmento del tiempo
tin = 0:intiempo:T;

%Cálculo del ángulo theta3 solución 1
theta3in_1 = thetap_inicial-theta1_in(1)-theta2_in(1);
theta3fin_1 = thetap_final-theta1_fin(1)-theta2_fin(1);

%Cálculo del ángulo theta3 solución 2
theta3in_2 = thetap_inicial-theta1_in(2)-theta2_in(2);
theta3fin_2 = thetap_final-theta1_fin(2)-theta2_fin(2);

%%%%Corrección del cálculo de última postura

theta1_finc = 0.2575;
theta2_finc = -0.9935;
theta3_finc = 0.7360;

```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%Duración del tiempo de simulación
```

```
T = 10; %seg tiempo total de la tarea  
muestreo = 20;  
intiempo = 10/muestreo;
```

```
%segmento del tiempo
```

```
tin = 0:intiempo:T;
```

```
Vtheta1=((30/T^3)*tin.^2-(60/T^4)*tin.^3+(30/T^5)*tin.^4)*(theta1_finc-  
theta1_in(1));  
Vtheta2=((30/T^3)*tin.^2-(60/T^4)*tin.^3+(30/T^5)*tin.^4)*(theta2_finc-  
theta2_in(1));  
Vtheta3=((30/T^3)*tin.^2-(60/T^4)*tin.^3+(30/T^5)*tin.^4)*(theta3_finc-  
theta3in_1);
```

```
figure(1)  
hold on  
plot(tin,Vtheta1,'Color','red');  
plot(tin,Vtheta2,'Color','green');  
plot(tin,Vtheta3,'Color','blue');  
title('Velocidades angulares de las juntas');  
xlabel('t');  
ylabel('1/s');  
legend('\theta1','\theta2','\theta3');  
hold off
```

```
%Simulación 1
```

```
figure(2)  
for i = 1:1:muestreo+1
```

```
t = intiempo*(i-1); %Se toma en cuenta 0 como inicio de la acción
```

```
theta1sim1 = theta1_in(1)+((10/T^3)*t^3-  
(15/T^4)*t^4+(6/T^5)*t^5)*(theta1_finc-theta1_in(1));  
theta2sim1 = theta2_in(1)+((10/T^3)*t^3-  
(15/T^4)*t^4+(6/T^5)*t^5)*(theta2_finc-theta2_in(1));  
theta3sim1 = theta3in_1+((10/T^3)*t^3-  
(15/T^4)*t^4+(6/T^5)*t^5)*(theta3_finc-theta3in_1);
```

```
x1L1 = [0 x0+L1*cos(theta1sim1)];  
y1L1 = [0 y0+L1*sin(theta1sim1)];
```

```
x2L1 = [x0+L1*cos(theta1sim1)  
x0+L1*cos(theta1sim1)+L2*cos(theta1sim1+theta2sim1)];  
y2L1 = [y0+L1*sin(theta1sim1)  
y0+L1*sin(theta1sim1)+L2*sin(theta1sim1+theta2sim1)];
```

```

x3L1 = [x0+L1*cos(theta1sim1)+L2*cos(theta1sim1+theta2sim1)
x0+L1*cos(theta1sim1)+L2*cos(theta1sim1+theta2sim1)+L3*cos(theta1sim1+the
ta2sim1+theta3sim1)];
y3L1 = [y0+L1*sin(theta1sim1)+L2*sin(theta1sim1+theta2sim1)
y0+L1*sin(theta1sim1)+L2*sin(theta1sim1+theta2sim1)+L3*sin(theta1sim1+the
ta2sim1+theta3sim1)];

```

```

%plot(xp_inicial,yp_inicial,'k*');
%plot(xp_final,yp_final,'k*');
%figure
line(x1L1,y1L1,'Color','red','LineStyle','-')
line(x2L1,y2L1,'Color','green','LineStyle','-')
line(x3L1,y3L1,'Color','blue','LineStyle','-')

```

```

F(i) = getframe;

```

```

end

```

```

movie(F)

```

```

%%%%Estudio de manipulabilidad

```

```

wmax = L1*L2*sin(pi/2);

```

```

theta1_est = double(theta1_in(1)+((10/T^3)*tin.^3-
(15/T^4)*tin.^4+(6/T^5)*tin.^5)*(theta1_finc-theta1_in(1)));
theta2_est = double(theta2_in(1)+((10/T^3)*tin.^3-
(15/T^4)*tin.^4+(6/T^5)*tin.^5)*(theta2_finc-theta2_in(1)));
theta3_est = double(theta3in_1+((10/T^3)*tin.^3-
(15/T^4)*tin.^4+(6/T^5)*tin.^5)*(theta3_finc-theta3in_1));

```

```

figure
w_est= abs(L1*L2*sin(theta2_est)/wmax);
plot(tin,w_est,'Color','red');
title('Manipulabilidad');
xlabel('t');
ylabel('w');

```

```

%%%%%Generación de señales

```

```

t_sig = transpose(tin);

```

```

theta1_tran = transpose(-theta1_est);
theta2_tran = transpose(-theta2_est);
theta3_tran = transpose(-theta3_est);

```

```

signal_theta1 = [t_sig theta1_tran];
signal_theta2 = [t_sig theta2_tran];
signal_theta3 = [t_sig theta3_tran];
%%%%%%%%%%%%%%

```

El segundo archivo contiene la solución para la trayectoria “Tray\_clas2”, el cual contiene el siguiente código

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Definición de trayectoria

%Posición y orientación inicial

xp_inicial = 0.1; %m
yp_inicial = 0.1; %m
thetap_inicial = pi/2; %rad

%Posición y orientación final
xp_final = 0.32; %m
yp_final = -0.05; %m
thetap_final = 0; %rad

%%Parametro
L3 = 0.071213; %m

%Trayectoria

t_total = 10; %s
t_muestreo = 0.1; %s
t_sim = 0:t_muestreo:t_total; %arreglo del tiempo

%Puntos de la trayectoria
xp = xp_inicial+((10/t_total^3)*t_sim.^3-
(15/t_total^4)*t_sim.^4+(6/t_total^5)*t_sim.^5)*(xp_final-xp_inicial);
yp = yp_inicial+((10/t_total^3)*t_sim.^3-
(15/t_total^4)*t_sim.^4+(6/t_total^5)*t_sim.^5)*(yp_final-yp_inicial);
thetap = thetap_inicial+((10/t_total^3)*t_sim.^3-
(15/t_total^4)*t_sim.^4+(6/t_total^5)*t_sim.^5)*(thetap_final-
thetap_inicial);

plot(xp,yp)

figure
plot(t_sim,thetap)

%%Cálculo de las xy del sistema 3

x3 = xp-L3*cos(thetap);
y3 = yp-L3*sin(thetap);

figure
plot(x3,y3)

%%Fin del diseño de trayectorias
```

%%%

El tercer archivo contiene la solución para la información de la postura del robot “postura2”, el cual contiene el siguiente código

%%%

```
%Función de la postura del robot 3R

function F = postura2(q,x3,y3)

%El cambio de variables de la postura
theta1 = q(1);
theta2 = q(2);

%Parámetros del robot
L1 = 0.12; %m
L2 = 0.12; %m

x0 = 0.055; %m
y0 = 0.0; %m

%Definición de la función
F(1) = x0+L1*cos(theta1)+L2*cos(theta1+theta2)-x3;
F(2) = y0+L1*sin(theta1)+L2*sin(theta1+theta2)-y3;

end
```

%%%

El cuarto archivo contiene la solución para solucionar la postura del robot “Sol\_clas2”, el cual contiene el siguiente código

%%%

```
waprox = [pi/2 -pi/2]; %Vector de proximación de la configuración

for i=1:length(t_sim)

    q_cal(i,:) = fsolve(@(q) postura2(q,x3(i),y3(i)),waprox);
    %planteamiento del cálculo de la solución de la postura

    theta1_cal(i) = q_cal(i,1); %Solicitud el ángulo Theta 1
    theta2_cal(i) = q_cal(i,2); %Solicitud el ángulo Theta 2

    waprox = [theta1_cal(i) theta2_cal(i)]; %Vector de aproximación
```

```

end

%Cálculo del ángulo theta3

for i=1:length(t_sim)

theta3_cal(i) = thetap(i)-theta1_cal(i)-theta2_cal(i);%Cálculo del ángulo
theta 3

end

%Parámetros del tiempo

t_total = 10; %s Tiempo total de la simulación
t_muestreo = 0.1; %s Incremento del tiempo para la simulación
t_sim = 0:t_muestreo:t_total; %arreglo del tiempo

%creación de la señal de salida para la simualción
%de la planeación en el espacio de trabajo del robot
t = transpose(t_sim);

theta1_t = transpose(-theta1_cal);
theta2_t = transpose(-theta2_cal);
theta3_t = transpose(-theta3_cal);

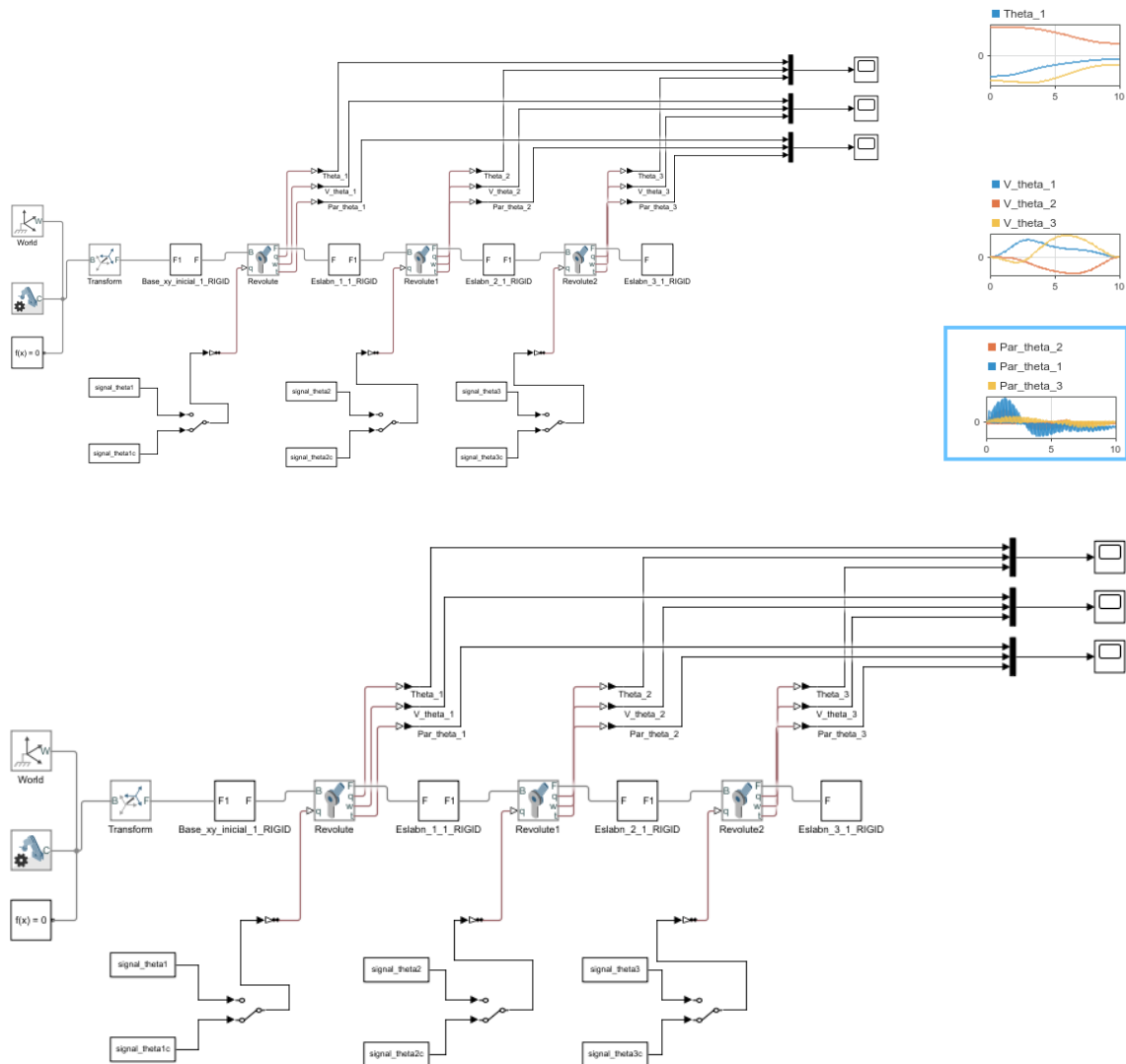
signal_theta1c = [t theta1_t];
signal_theta2c = [t theta2_t];
signal_theta3c = [t theta3_t];

%Fin del programa
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

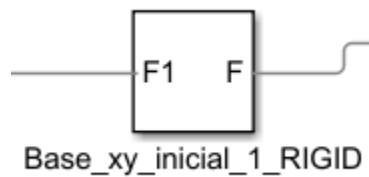
```

Después de insertar estos archivos dentro de un proyecto en Matlab, usamos Simulink para poder ligarlos con los eslabones dentro de un ensamble CAD, el esquema usado en Simulink es el siguiente.

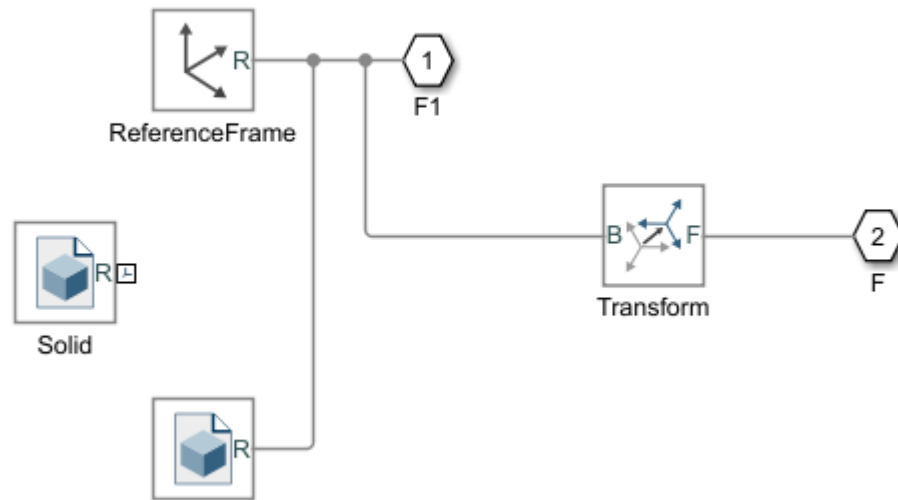




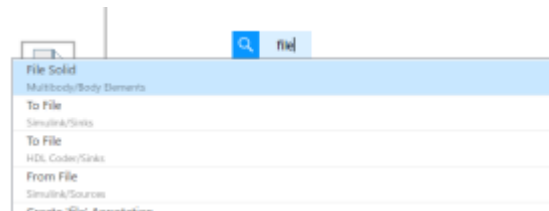
En nuestro caso el profesor nos brindó de un primer ensamble, el cual modificamos, de manera que podamos introducir un archivo cada diferente y poder cambiar cada eslabón sin necesidad de volver a introducir un ensamble nuevo. Para hacer eso necesitamos entrar en cualquier recuadro donde se encuentra la información de cada eslabón dentro de simulink, que tienen el siguiente símbolo



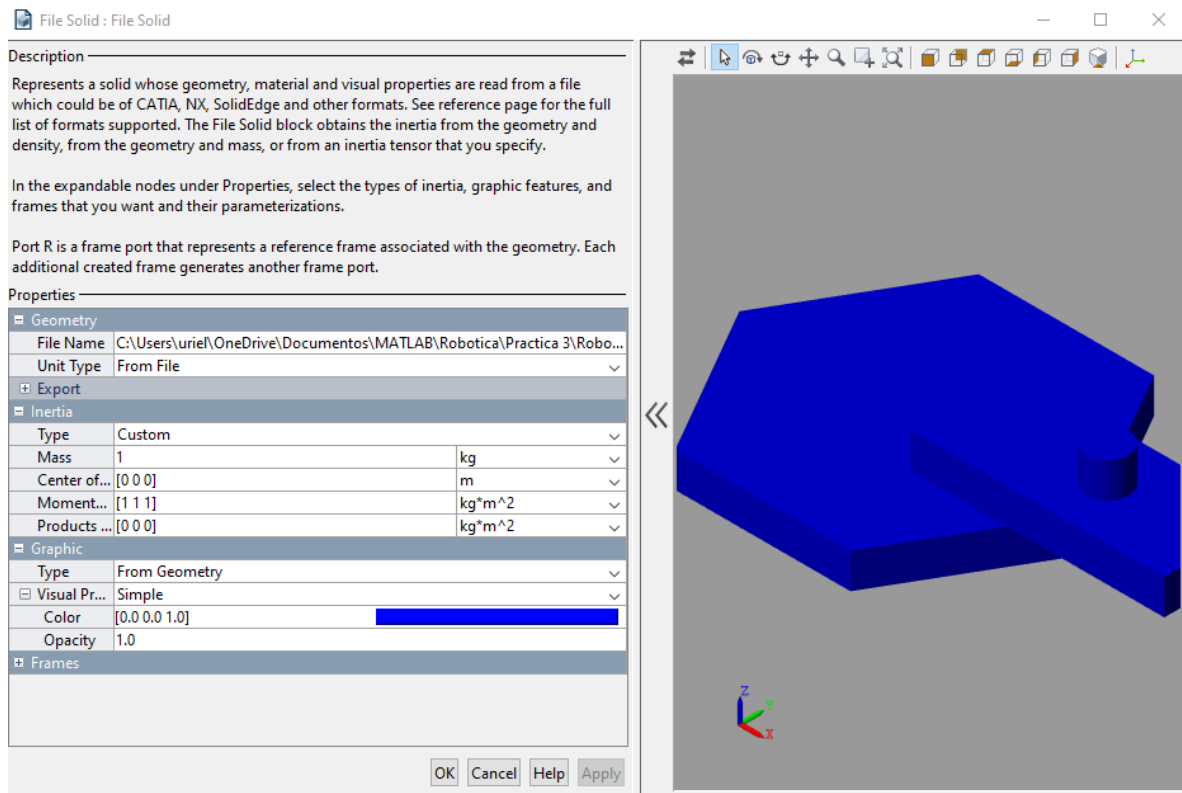
En este caso es la base, una vez dentro se muestra el siguiente esquema



Como se observa tenemos dos cuadros similares donde uno no está conectado, un nuestro eslabón y otro el del profe para introducir este eslabón debes introducir “file solid” dentro del espacio de trabajo.

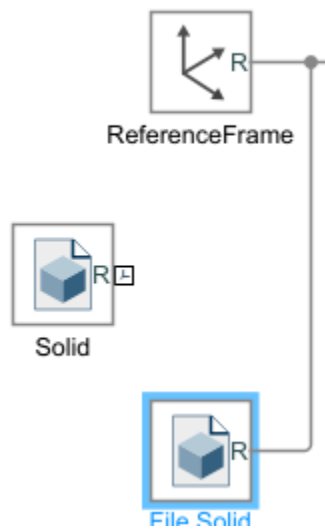


Ya que tengamos el “file solid” damos dobles click para editarlo en donde nos aparecerá lo siguiente.



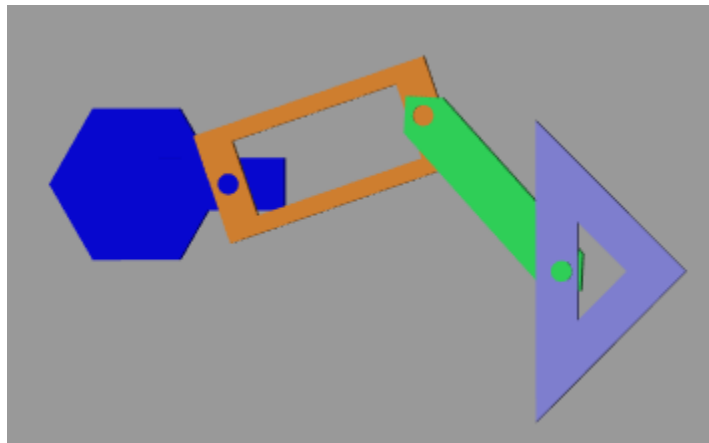
En el apartado “file name”, buscaremos nuestra pieza, la cual tiene que estar guardada en un archivo de tipo step. Lo segundo a hacer es poner en el apartado de “inertia”, en el subapartado type, custom, el resto de parámetros dentro de inercia, dejarlos como están. Y por último podemos en el apartado “graphic” cambiar el color si usamos la configuración visual básica, ya que el archivo step puede no guardar el color original de la pieza.

Y listo solo cerramos la ventana y conectamos el file solid nuevo al archivo simulink.



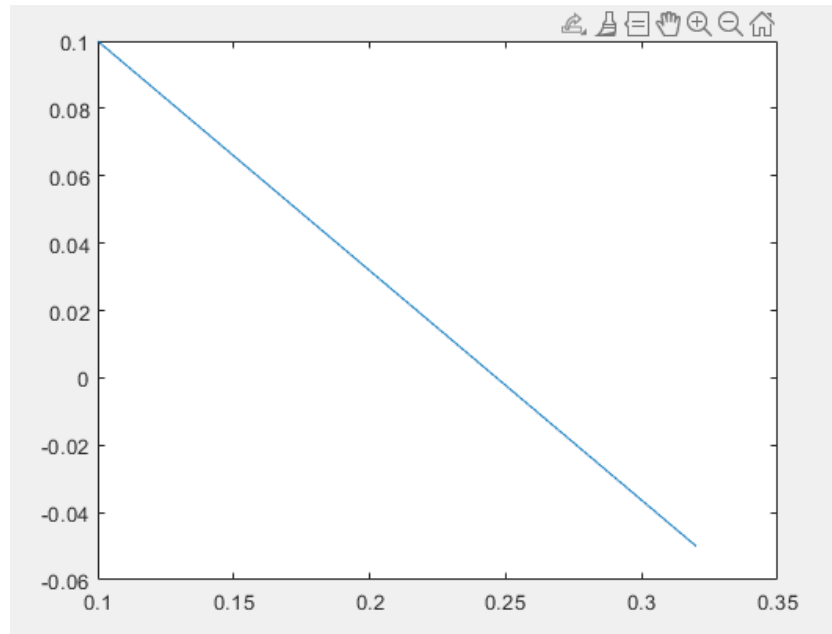
## Resultado

Empezaremos hablando del primer caso en el que nosotros calculamos la trayectoria, La simulación del robot Tuvo resultados éxitos, el siguiente video muestra la simulación completa

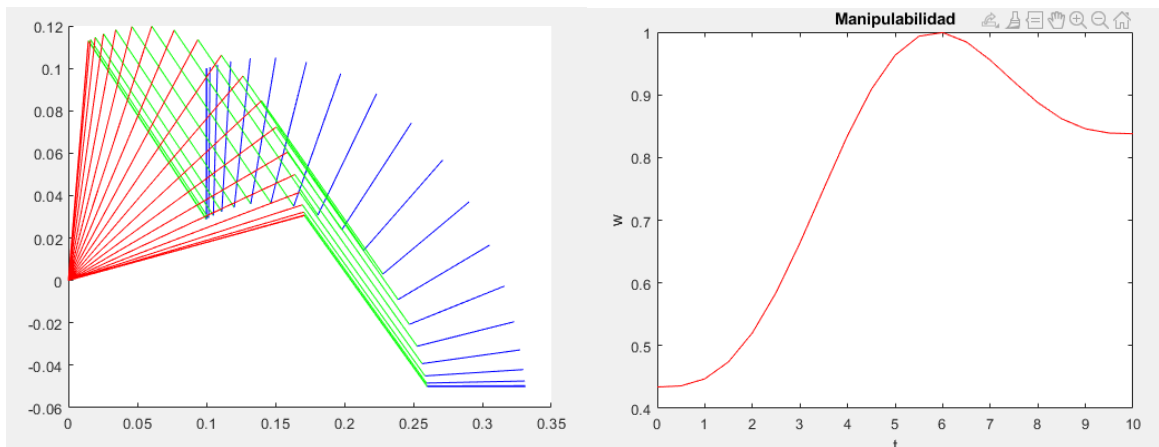


<https://www.youtube.com/watch?v=H62ig98LHUM>

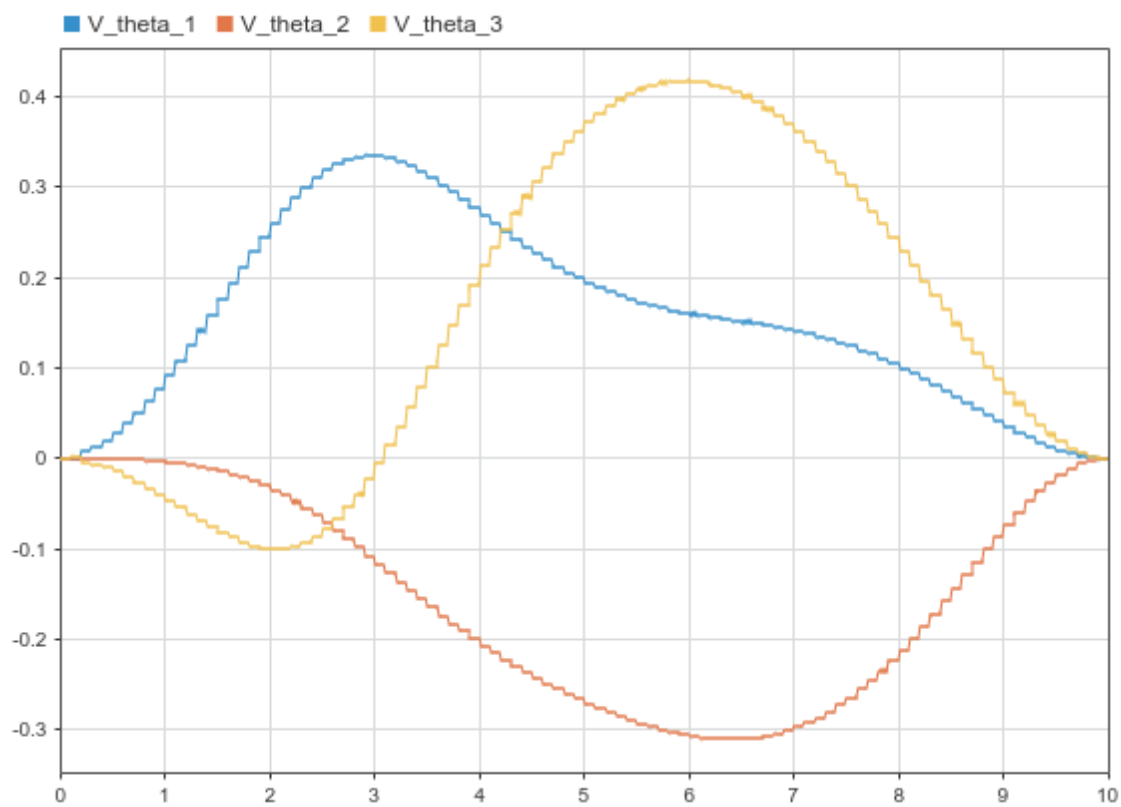
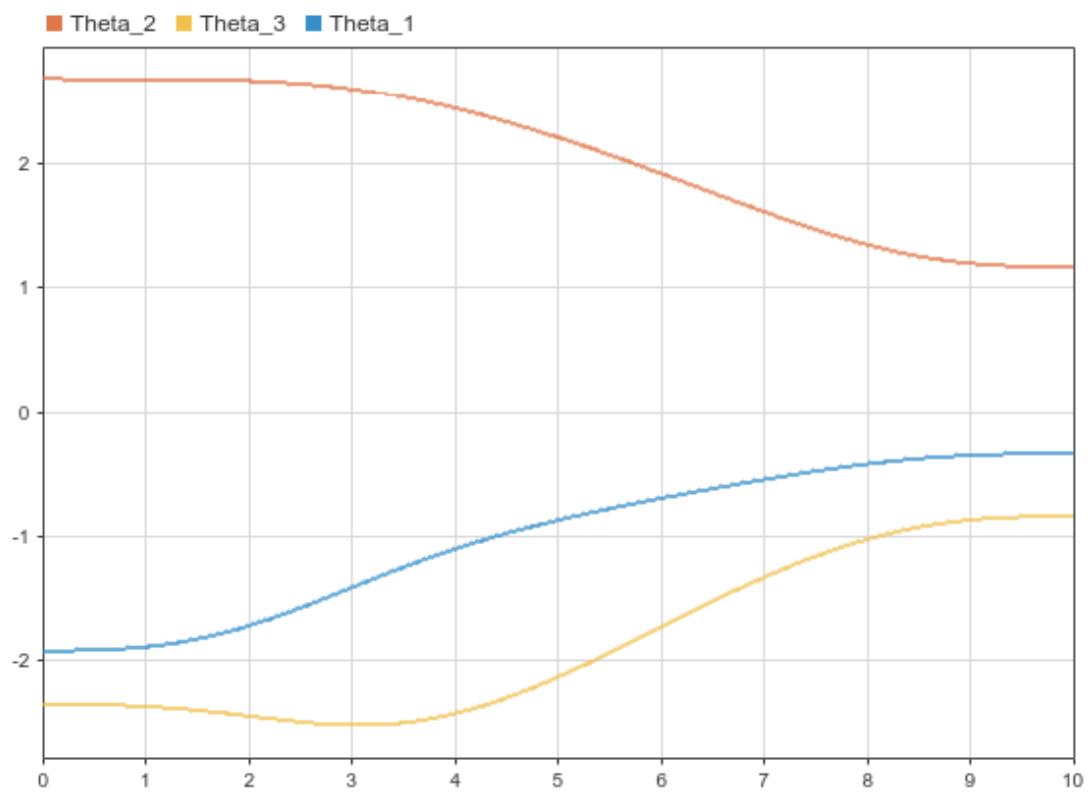
la trayectoria propuesta fue la siguiente

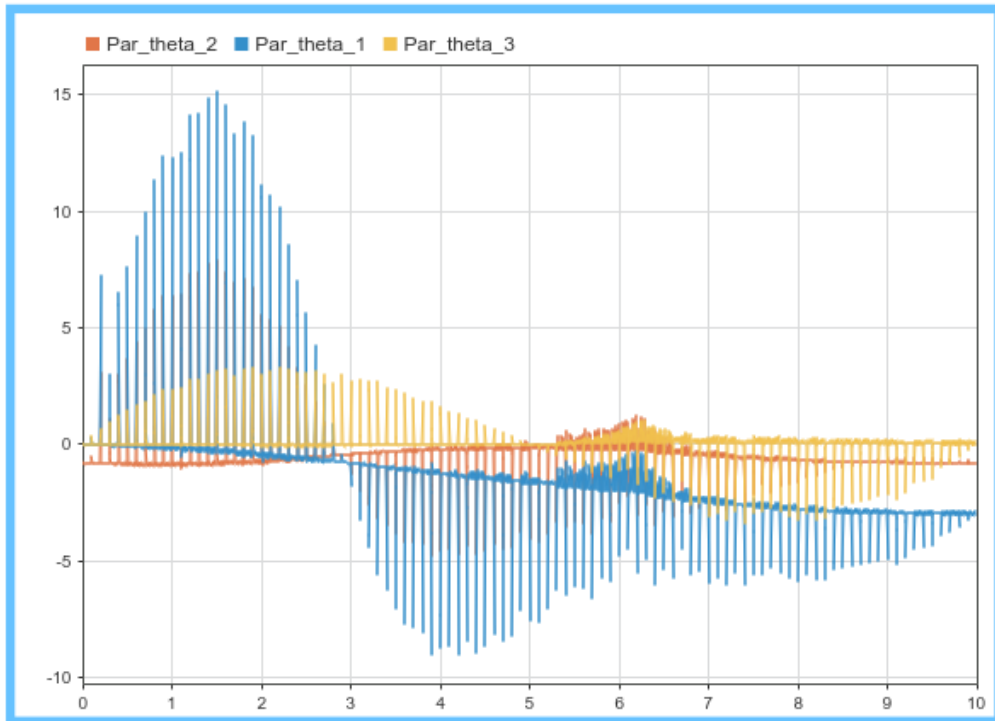


En la siguiente imagen se muestra la comprobación de que el robot puede realizar los movimientos, y esto lo comprobamos con la gráfica de manipulabilidad, que no varía demasiado

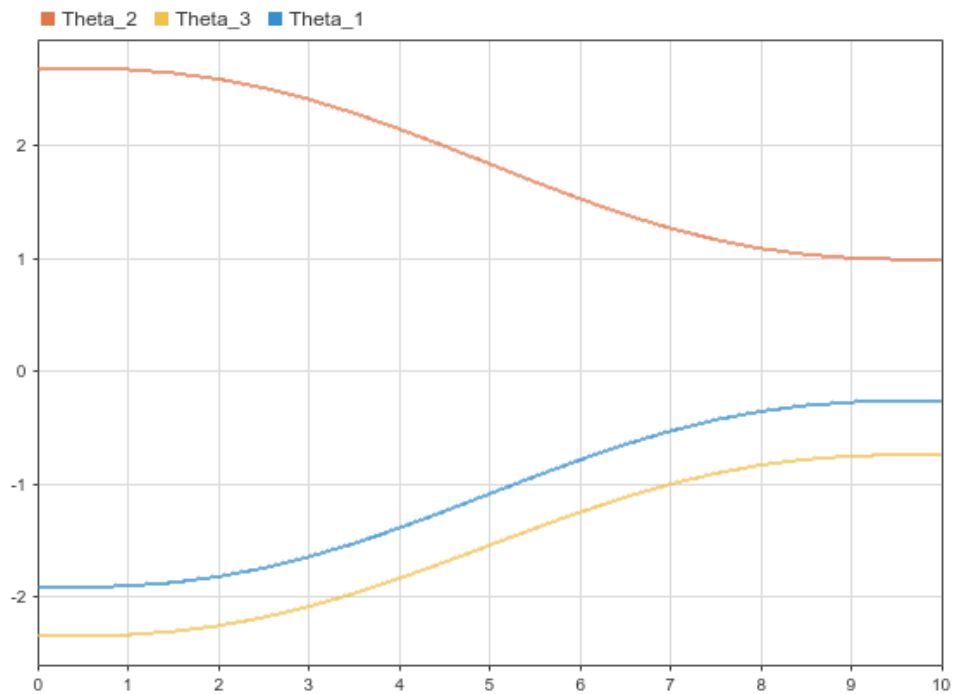


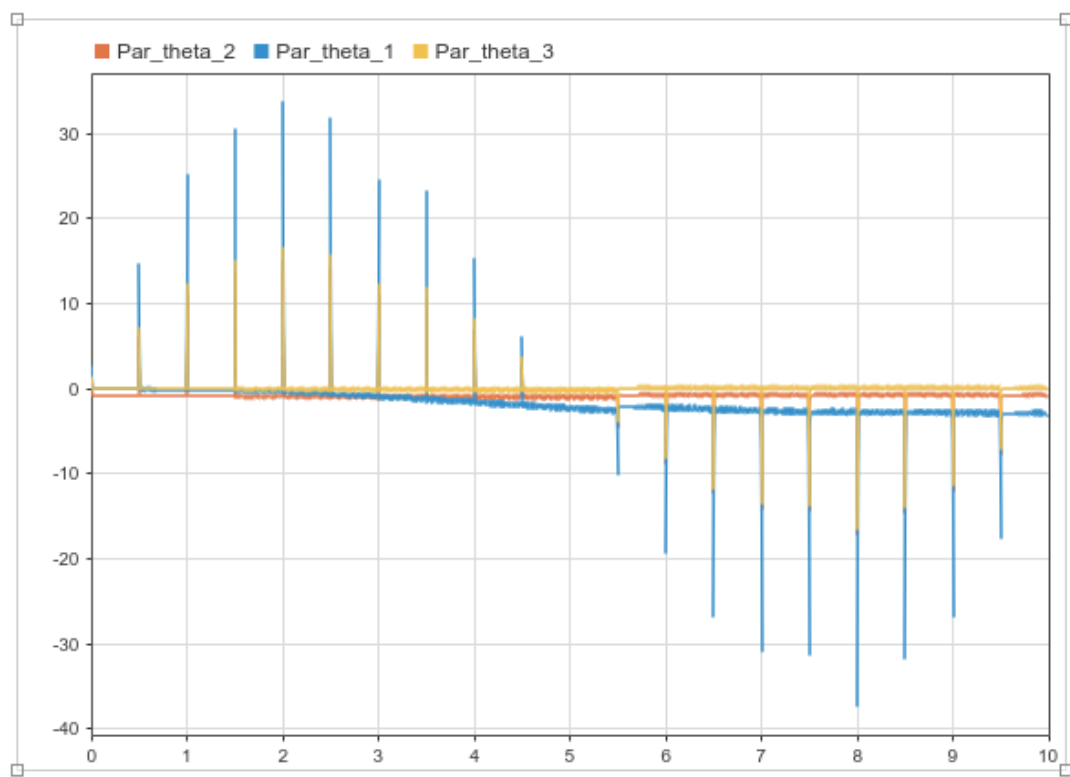
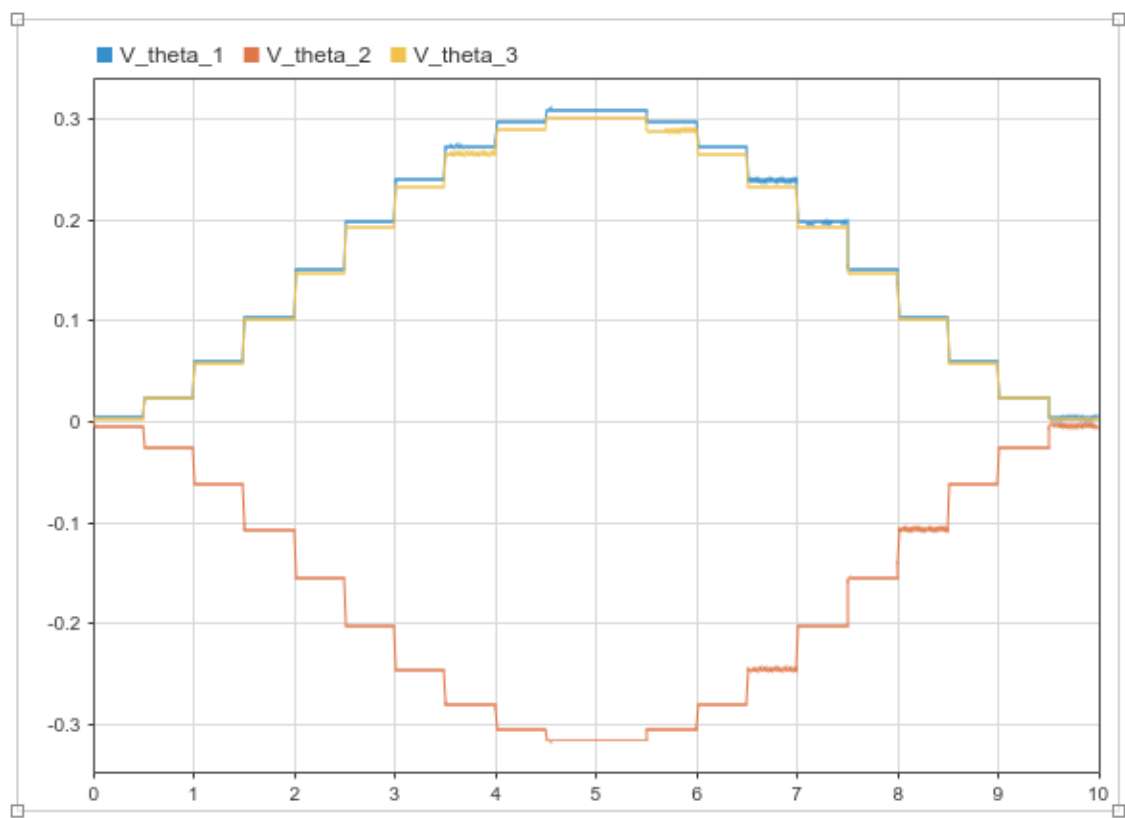
Las siguientes imágenes muestran las graficas calculadas, del movimiento del robot, que nos muestran como cambian, los ángulos, las velocidades, y la pared, en cada junta.





En la siguiente simulación fue solamente introduciendo los puntos a los que queríamos que llegara el efector final. Obteniendo las siguientes graficas



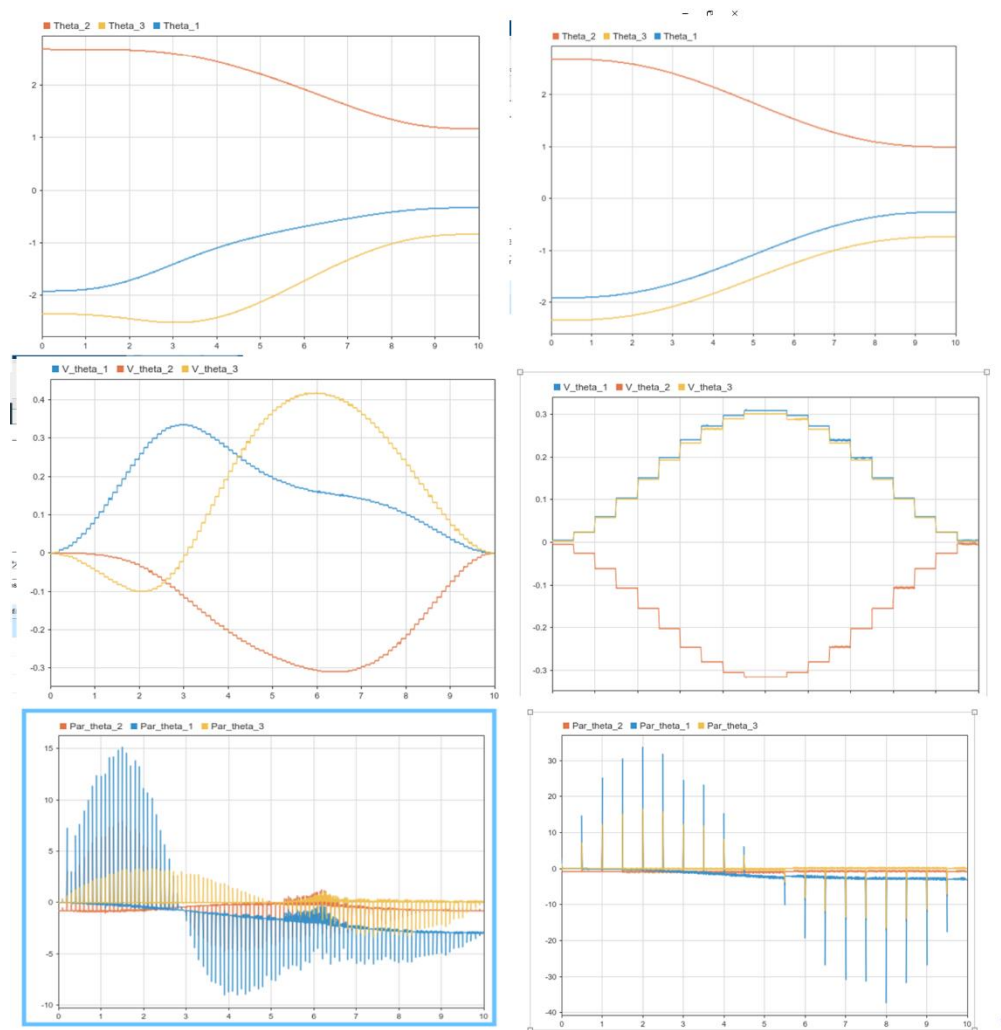




## Conclusiones

Se cumple parcialmente (parcialmente)

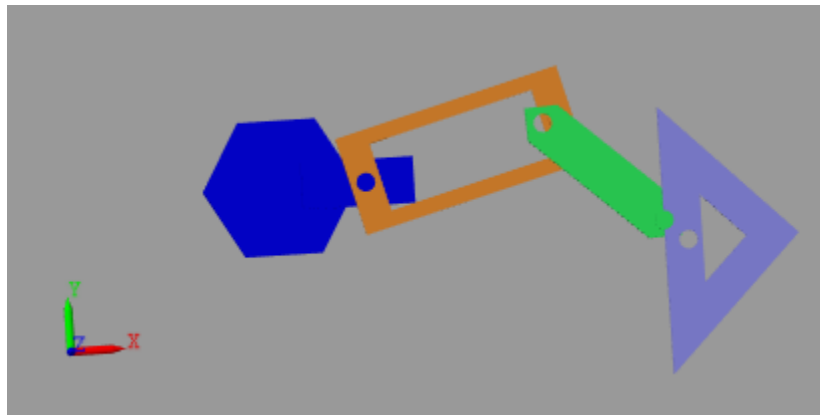
En nuestro caso teníamos dos objetivos, uno la simulación, y otro la importación de eslabones nuevos desde cad. Para el primer objetivo lo cumplimos completamente obteniendo la simulación y los diagramas de comportamiento.



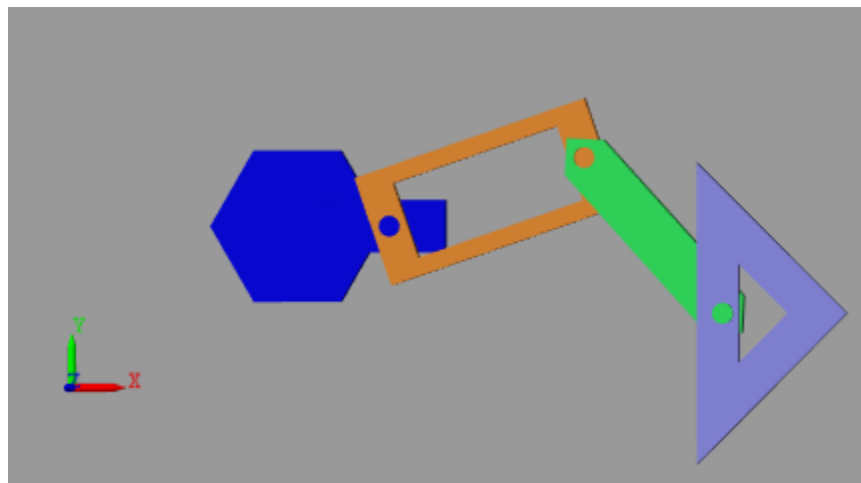
A la izquierda la trayectoria calculada y en la derecha solo calculando la posición del EF, se puede observar que la principal diferencia es el esfuerzo al que

someten las juntas, ya que cuando lo calculamos nosotros, le cuesta mucho mas realizar los movimientos, y gasta más energía, mientras que si no lo hacemos las juntas muestran un comportamiento más armónico, en donde sus velocidades cambian menos de valor y los pares son mucho menores.

El segundo objetivo fue cubierto parcialmente ya que si bien se observa un cambio con respecto a las piezas originales, solo pudieron ser reemplazadas por piezas que respeten las dimensiones del ensamble original, estas dimensiones son en específico, las posiciones originales, en donde se acoplan los motores, ya que si cambiaba la dimensión de las piezas me aparecían desfasadas como en la siguiente imagen



Esto me obligo a usar las dimensiones originales, y dejando la pieza bien ensamblada



Esto por su puesto, es un error, y para resolverlo investigue en todo simulink y no encontré nada, el error pienso está en el ensamble original ya que este jala los sistemas de origen iniciales, y por lo tanto a la hora de importar un eslabón, no modificamos los datos del ensamble, solo de ese eslabón en específico, si corrigiéramos el ensamble completo estoy seguro el problema se resolvería.

El video de el mal funcionamiento se puede encontrar en la siguiente liga

<https://www.youtube.com/watch?v=KicELYRbVz8>