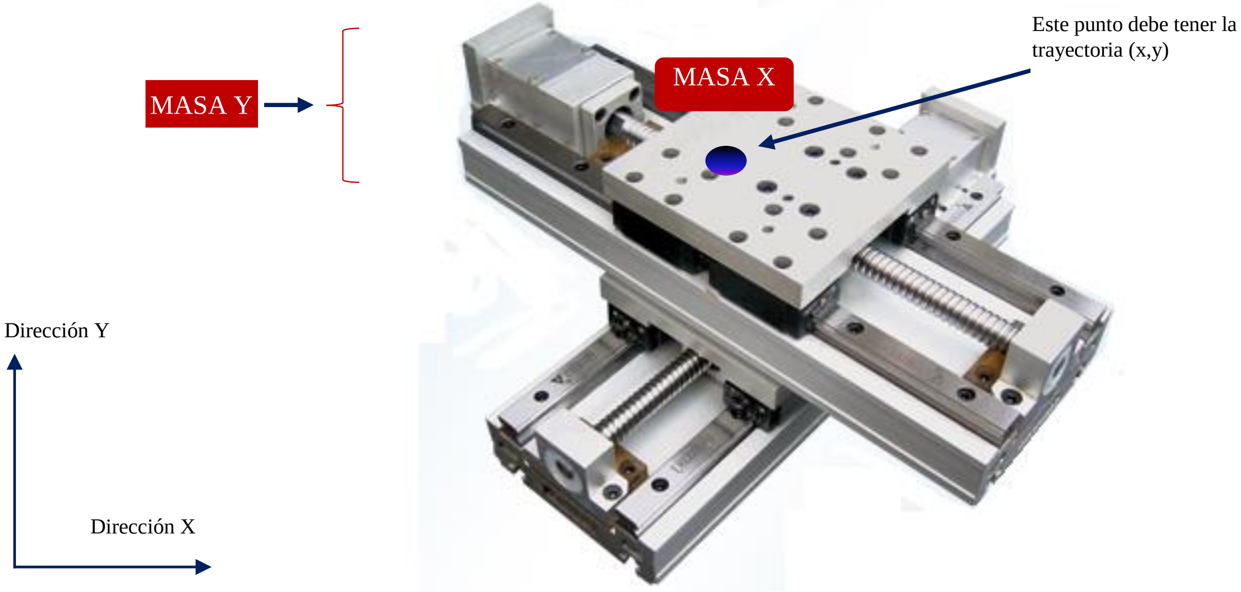
PROBLEMA

Se quiere diseñar el controlador de un torno numérico para posicionamiento y tracking en dos dimensiones. El sistema de posicionamiento de la cuchilla (circulo) se muestra en la siguiente figura

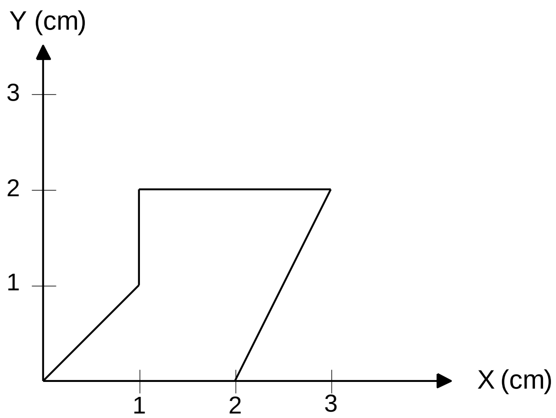


El mejor diseño es el que cumple trayectoria en menos tiempo posible sin que pase el voltaje y potencia

Control Optimo de un Torno de Control Numérico

Se quiere diseñar el controlador de un torno de control numérico para posicionamiento y tracking en dos dimensiones.

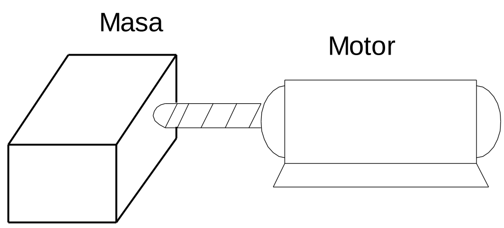
La forma de la pieza a tornear tiene la siguiente forma geométrica:



1. Forma de la pieza

1. Proceso de Modelamiento.

Si analizamos el sistema, nos damos cuenta que podemos modelar el sistema que acciona el motor X y el sistema que acciona el motor Y de manera similar; es por ello que modelamos el sistema que mostramos a continuación en forma general. Luego podremos adaptarlos a cada sistema según sus parámetros eléctricos y mecánicos propios de cada sistema, así como su masa.



2.Forma del sistema para motor X y para el motor Y

a. <u>Modelamiento del sistema de la figura 2:</u>

- **Parte eléctrica**:

$$V = Ri + L\frac{di}{dt} + e_b \tag{1}$$

$$e_b = K_b\dot{\theta} \tag{2}$$

$$T_e = K_T i \tag{3}$$

- **Parte mecánica:**

$$\sum torques = I\ddot{\theta} \tag{4}$$

$$T_e - F_t r = I\ddot{\theta} \tag{5}$$

$$F_t = K_S F_L \tag{6}$$

$$F_L - F_f = m\ddot{x} \tag{7}$$

$$\theta = \frac{2\pi x}{p} \tag{8}$$

- **Deducción de la ecuación de estado:**

Como la inductancia es despreciable, y reemplazando la ecuación 2 en 1, tenemos:

$$V = Ri + e_b$$
$$V = Ri + K_b \dot{\theta}$$

reemplazando la ecuación 8 tenemos:

$$V = Ri + K_b \frac{2\pi \dot{x}}{p}$$
$$i = \frac{V}{R} - \frac{2\pi K_b \dot{x}}{Rp} \qquad (9)$$

Reemplazando la ecuación 3 y 6 y de 8 en la ecuación 5:

$$K_T i = K_S F_L r = \frac{2\pi I \ddot{x}}{p}$$

reemplazando la ecuación 7 y 9:

$$K_T i - K_S (m\ddot{x} + F_f) r = \frac{2\pi I \ddot{x}}{p}$$
$$\ddot{x} = \frac{K_t i}{\frac{2\pi I}{p} + K_s mr} - \frac{K_S F_f r}{\frac{2\pi I}{p} + K_s mr}$$

$$\ddot{x} = \frac{K_t \left[\frac{V}{R} - \frac{2\pi K_b \dot{x}}{Rp}\right]}{\frac{2\pi I}{p} + K_S mr} - \frac{K_S F_f r}{\frac{2\pi I}{p} + K_S mr}$$

$$\ddot{x} = \frac{-2\pi K_T K_b \dot{x}}{Rp\left[\frac{2\pi I}{p} + K_S mr\right]} - \frac{K_T V}{R\left[\frac{2\pi I}{p} + K_S mr\right]} - \frac{K_S F_f r}{\left[\frac{2\pi I}{p} + K_S mr\right]}$$

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & \dfrac{-2\pi k_T K_b}{Rp\left[\frac{2\pi I}{p} + K_S mr\right]} \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ \dfrac{k_T}{R\left[\frac{2\pi I}{p} + K_S mr\right]} \end{bmatrix} V + \begin{bmatrix} 0 \\ \dfrac{-k_S r}{\left[\frac{2\pi I}{p} + K_S mr\right]} \end{bmatrix} F_f \qquad (10)$$

Además de la ecuación 9 tenemos:

$$i = \left[\frac{-2K_b}{pR}\right]\dot{x} + \left[\frac{1}{R}\right]V \qquad (11)$$

b. <u>Consideraciones:</u>

La ecuación de estado deducida y dada por la ecuación 10, es valida para ambos sistemas; es decir para el sistema que comprende el motor X y la masa sobre la cual acciona; y para el sistema que comprende el motor Y y la masa sobre la cual acciona. Solamente hay que considerar lo siguiente para cada sistema:

Para el motor X:

Parámetros eléctricos y mecánicos : son datos del problema
Masa : Masa X

Para el motor Y:

Parámetros eléctricos y mecánicos : son datos del problema
Masa : Masa Y + Masa X + Masa del motor X y su tornillo

2. Ecuación de estado con acción Integrativa.

La matriz de estado para ambos sistemas tendrán la forma:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ x - x^* \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & \dfrac{-2\pi k_T K_b}{Rp\left[\dfrac{2\pi l}{p} + K_S mr\right]} & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \int (x - x^*) \end{bmatrix} + \begin{bmatrix} 0 \\ \dfrac{k_T}{R\left[\dfrac{2\pi l}{p} + K_S mr\right]} \\ 0 \end{bmatrix} V + \begin{bmatrix} 0 \\ \dfrac{-k_S r}{\left[\dfrac{2\pi l}{p} + K_S mr\right]} \\ 0 \end{bmatrix} F_f + ..$$

$$.........+ \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} x^*$$

3. Descripción del Programa en Matlab.

Los programas desarrollados son tres los cuales tienen las siguiente objetivo:

a. Velocidad.m

Es una función el cual recibe como parámetro la velocidad resultante en cada tramo que es ingresado desde el programa **trayec.m**
Esta función me genera las velocidades resultantes en cada tramo de la figura a formar, en un tiempo determinado

b. Trayec.m

En base a la velocidad instantánea entregada por la función **velocidad.m,** calculamos aquí las componentes de las velocidades tanto en el eje X como en el eje Y. Es aquí donde integramos las componentes de Vx y Vy para obtener las trayectorias de referencia en el eje X y en el eje Y.

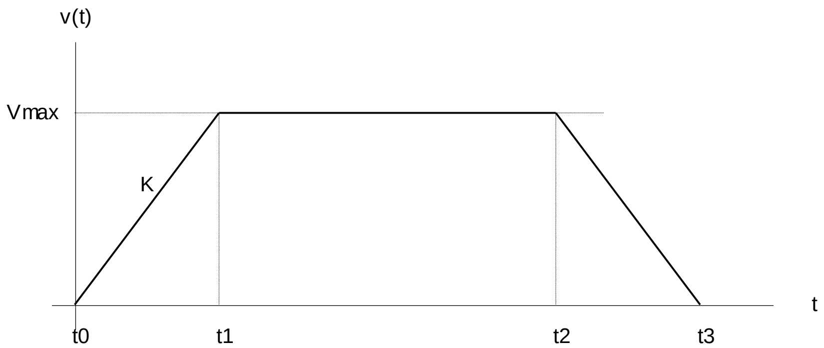
c. Torno.m

Es el programa principal que llama a las funciones **trayec.m** y **velocidad.m.** Desde se realiza el calculo de la ley de control para realizar la trayectoria deseada tanto en el eje X como en el eje Y.

A continuación describiremos cada uno de los programas en forma detallada para su mejor entendimiento.

Velocidad.m

En esta función ingresa la velocidad total máxima, distancia por tramo, resolución del tiempo y porcentaje del tiempo total que demora en pasar desde una velocidad cero hasta la velocidad máxima. Nuestro objetivo de esta función es obtener una velocidad que tenga una forma trapezoidal en cada tramo, es decir se acelera lentamente hasta alcanzar la velocidad máxima, que permanezca constante por un cierto tiempo y luego desacelera hasta llegar nuevamente a la velocidad cero, como observamos en la figura.



3. Forma trapezoidal de la velocidad instantánea total

Primero procedemos a calcular el tiempo que demora en realizar cada tramo a la velocidad dada. Esto lo hallamos integrando la curva de la velocidad para obtener el espacio recorrido (el cual es conocido); o lo que es lo mismo hallamos el área del trapecio, el cual nos dará la distancia; de esta manera obtenemos el tiempo que demora en cada tramo:

```
t_fin   = d/((1-P_rampa)*abs(Velomax));
```

También, calculamos el tiempo que demora en cada rampa de subida, que viene a ser un porcentaje del tiempo total de cada tramo:

```
t_rampa = P_rampa*t_fin;
```

Luego obtenido el tiempo por cada tramo procedemos a calcular la velocidad en cada instante de tiempo, para así formar los cuatro trapecios:

```
V_subida    =  (Velomax/t_rampa)*[0:dt:+t_rampa-dt];
V_constante =  Velomax*ones(1,length([t_rampa:dt:t_fin-t_rampa-dt]));
V_bajada   = -(Velomax/t_rampa)*[t_fin-t_rampa:dt:t_fin]+(Velomax*t_fin/t_rampa);
```

El programa completo e muestra a continuación:

```
%************************** velocidad.m ****************************************

% Esta funcion calcula las velocidades resultantes para cada tramo de la figura %
a formar. Nosotros hemos asumido que la forma de la velocidad sera en forma
% trapezoidal por cada tramo; es decir como tenemos 5 tramos entonces tendremos %
cinco velocidades de forma trapezoidal en todo el recorrido de la figura
%                             *******************
%                             *                   *
%                            *                     *
%                           *                       *
%

function V_tramo = Velocidad(d, Velomax,P_rampa, dt)
% Esta funcion recibe como parametros:
% d          : distancia del tramo a recorrer
% Velomax    : Velocidad maxima que alcanza en cada tramo y el cual permanece
               constante por cierto tiempo
% P_rampa    : Duracion de la rampa de subida y bajada expresado en '%'del tiempo
               total en cada tramo
% dt         : Diferencial de tiempo

t_fin    = d/((1-P_rampa)*abs(Velomax));
t_rampa = P_rampa*t_fin;
V_subida    =  (Velomax/t_rampa)*[0:dt:+t_rampa-dt];
V_constante =  Velomax*ones(1,length([t_rampa:dt:t_fin-t_rampa-dt]));
V_bajada   = -(Velomax/t_rampa)*[t_fin-t_rampa:dt:t_fin]+(Velomax*t_fin/t_rampa);
V_tramo     =  [ V_subida   V_constante    V_bajada ];
```

Trayec.m

Desde este programa llamamos a la función **velocidad.m** el cual nos entrega la velocidad en cada tramo, y a partir del cual puedo obtener las componentes de las velocidades tanto en X como en Y:

```
% Calculo de las componentes de las velocidades Vx, Vy
% -----------------------------------------------------

% Evaluacion de la velocidad Vx y Vy en el tramo a-b
V1 = Velocidad(d1, V, ramp,dt);  %Velocidad resultante en tramo a-b
m=0;                              %pendiente en tramo a-b
Vab(1,:) = V1;                   %Velocidad Vx en tramo a-b
Vab(2,:) = m.*Vab(1,:);          %Velocidad Vy en tramo a-b
```

De la misma manera hacemos para los demás tramos como se muestra en el programa **trayec.m**.

Una vez obtenida todas las componentes de las velocidades en X e Y para cada tramo, los reúno en un vector de la siguiente manera:

```
Vel_total = [Vab Vbc Vcd Vde Vef];  %velocidad total
```

el cual me contiene la componente de Vx y Vy durante la realización de toda todo la figura. Luego obtenemos nuestra trayectoria de referencia tanto en el eje X como el eje Y, y esto lo obtenemos integrando el vector velocidad de componentes en Vx e Vy:

```
for i=1:length(Vel_total)-1
   ruta(:,i+1)=ruta(:,i)+dt.*Vel_total(:,i+1);  %integro la velocidad
end
```

Es este punto donde obtengo mis referencias de trayectoria tanto en el eje X como el eje Y que me servirán para ingresarlo al realizar el control.

```
%************************** trayec.m ************************************

% En este programa calculo las velocidades correspondientes al eje X
% y al eje Y (Vx, Vy) a partir de ingresar la velocidad maxima resultante
%
%************************************************************************

function trayectoria = trayec()

% Ingreso de valores de velocidad
% -------------------------------
disp('   '); disp('   '); disp('   ');

V    =  input('Ingrese velocidad resultante   (m/s) : ');
ramp =  input('Ingrese el porcentaje de rampa   (%) : ');
dt   = input('Ingrese diferencial de tiempo     (ms) : ');
disp('   '); disp('   ');

%V = 0.001;  %V = 0.004;       %rango de velocidades recomendadas
%dt = 0.02;
%ramp =0.2;

% Coordenadas de la trayectoria
% -----------------------------
a = [0,0];
b = [2,0];
c = [3,2];
d = [1,2];
e = [1,1];
f = [0,0];

punto_ini= a;              % punto de partida de la trayectoria
R =0.01*[a;b;c;d;e;f];     % coordenadas en metros
```

```matlab
% Calculo de las distancias a rrecorrer en cada tramo
% ---------------------------------------------------
d1=abs(R(2,1)-R(1,1));                              %longitud tramo a-b
d2 =sqrt((R(3,1)-R(2,1)).^2+(R(3,2)-R(2,2)).^2); %longitud tramo b-c
d3=abs(R(4,1)-R(3,1));                                %longitud tramo c-d
d4=abs(R(5,2)-R(4,2));                              %longitud tramo d-e
d5=sqrt((R(6,1)-R(5,1)).^2+(R(6,2)-R(5,2)).^2);  %longitud tramo e-f

ceros = zeros(2,2);
sizeceros = size(ceros,2);

% Calculo de las componentes de las velocidades Vx, Vy
% -----------------------------------------------------

% Evaluacion de la velocidad Vx y Vy en el tramo a-b
V1 = Velocidad(d1, V, ramp,dt);  %Velocidad resultante en tramo a-b
m=0;                              %pendiente en tramo a-b
Vab(1,:) = V1;                   %Velocidad Vx en tramo a-b
Vab(2,:) = m.*Vab(1,:);          %Velocidad Vy en tramo a-b

% Evaluacion de la velocidad Vx y Vy en el tramo b-c
V2 = Velocidad(d2, V, ramp,dt); %Velocidad resultante en tramo b-c
m=2;                             %pendiente en tramo b-c
Vbc(1,:) = V2./sqrt(1+m.^2);    %Velocidad Vx en tramo b-c
Vbc(2,:) =m.*Vbc(1,:);          %Velocidad Vx en tramo b-c

% Evaluacion de la velocidad Vx y Vy en el tramo c-d
V3 = Velocidad(d3, -V,ramp, dt); %Velocidad resultante en tramo c-d
m=0;                             %pendiente en tramo c-d
Vcd(1,:) = V3;                   %Velocidad Vx en tramo c-d
Vcd(2,:) = m.*Vcd(1,:);          %Velocidad Vy en tramo c-d
```

```matlab
% Evaluacion de la velocidad Vx y Vy en el tramo d-e
V4 = Velocidad(d4, -V, ramp,dt); %Velocidad resultante en tramo d-e
m=0;                             %pendiente en tramo d-e
Vde(2,:) = V4;                   %Velocidad Vx en tramo d-e
Vde(1,:) = m.*Vde(2,:);          %Velocidad Vy en tramo d-e

% Evaluacion de la velocidad Vx y Vy en el tramo e-f

V5 = Velocidad(d5, -V, ramp,dt);  %Velocidad resultante en tramo e-f
m=1;                              %pendiente en tramo e-f
Vef(1,:) = V5./sqrt(1+m.^2);      %Velocidad Vx en tramo e-f
Vef(2,:) =m.* Vef(1,:);           %Velocidad Vy en tramo e-f

Vel_total = [Vab Vbc Vcd Vde Vef];  %velocidad total

% Reconstruyo la trayectoria a partir de las velocidades
% -----------------------------------------------------
ruta(:,1)=punto_ini';                    %inicio punto de partida
for i=1:length(Vel_total)-1
   ruta(:,i+1)=ruta(:,i)+dt.*Vel_total(:,i+1);  %integro la velocidad
end

Vel_total = [Vel_total ceros];
ruta = [ruta ceros];

% Determinacion del vector de tiempos
% ------------------------------------
t = 0:dt:(length(Vel_total)-1)*dt;
sizet = length(t);
```

```
% Ploteo de Graficos
% ------------------
figure(1);
plot(t,Vel_total(1,:));          %ploteo Vx
xlabel('t[seg]');
ylabel('V_x [mt/seg]');
title('Velocidad en el EJE X'); grid;

figure(2);
plot(t,Vel_total(2,:));           %ploteo Vy
xlabel('t[seg]');
ylabel('V_y [mt/seg]');
title('Velocidad en el EJE Y'); grid;

figure(3);
plot(t,ruta(1,:),'b');            %ploteo X reconstruido
xlabel('t[seg]');
ylabel('[mt]');
title('Posicion en el EJE X');
legend('Referencia en X para mi Control');grid;

figure(4);
plot(t,ruta(2,:));                %ploteo Y reconstruido
xlabel('t[seg]');
ylabel('[mt]');
title('Posicion en el EJE Y');
legend('Referencia  en Y para mi control');grid;

figure(5);
plot(ruta(1,:),ruta(2,:));     %ploteo curva reconstruida X-Y
xlabel('t[seg]');
ylabel('[mt]');
title('CURVA RECONSTRUIDA A PARTIR DE LAS VELOCIDADES');grid;

save curvas Vel_total ruta R t dt;
```

Torno.m

Este es el programa principal que llama a la función **trayec.m** el cual a su vez este llama a **velocidad.m.** Como este es el programa principal y donde se realiza el control entonces lo describiremos secuencialmente:
Primero inicializamos los parámetros eléctricos tanto del motor X como del motor Y, así como las características del tornillo sin fin, y las masas involucradas en el sistema, como se muestra a continuación:

```
%******************************** Torno.m *****************************
% Este programa realiza el control del sistema de control numerico

clc, clear all; close all;

% Parametros tenicos del motor que realiza trayectoria en X
% ---------------------------------------------------------
R_X     = 2;         % resistencia interna (ohm)
Kb_X    = 0.025;      % coeficiente de Fuerza Electromotriz (volt-sec/rad)
Kt_X    = 0.0205;     % coeficiente torque corriente (N-m/Amp)
r_X     = 0.02/2;     % radio deltornillo sinfin (m)
p_X     = 0.004;      % paso del tornillo sinfin (m)
Ks_X    = 0.9;        % constante de fuerzas del tornillo sin fin
m_X     = 0.125;      % masa x (kg)
M_X     = 0.22;       % masa del motor x y tornillo sinfin (kg)
I_X     = 0.0000435;   % inercia total del motor x y tornillo sinfin (kg-m2)
```

```
Fric_X = 0.2;        % friccion viscosa (N-seg/m)

% -----------------------------------------------------------------------
Pmax_X = 400;     % potencia maxima del motor x (W)
Vmax_X = 40;        % maximo voltaje aplicable (V)

% Parametros tenicos del motor que realiza trayectoria en Y
% -----------------------------------------------------------------------
R_Y    = 30;         % resistencia interna (ohm)
Kb_Y   = 0.021;      % coeficiente de Fuerza Electromotriz (volt-sec/rad)
Kt_Y   = 0.032;      % coeficiente torque corriente (N-m/Amp)
r_Y    = 0.04/2;        % radio del tornillo sinfin (m)
p_Y    = 0.004;      % paso del tornillo sinfin (m)
Ks_Y   = 0.95;       % constante de fuerzas del tornillo sin fin
m_Y    = 0.35;      % masa y (kg)
M_Y    = 0.55;       % masa del motor y y tornillo sinfin (kg)
I_Y    = 0.0000855; % inercia total del motor y y tornillo sinfin (kg-m2)
Fric_Y = 0.25;       % friccion viscosa (N-seg/m)
% -------------------------------------------------------------------------
Pmax_Y = 600;    % potencia maxima del motor y (W)
Vmax_Y = 60;        % maximo voltaje aplicable (V)
```

Definidos los diversos parámetros, procedemos a inicializar nuestras matrices A, B para ambos sistemas; así como la matriz integrativa aumentada para ambos sistemas. También llamamos a la función **trayec.m** el cual nos entrega las referencias de las trayectorias en el eje X y en el eje Y. Esto lo vemos a continuación:

```
%*********** Ecuaciones de estado del sistema del motor X  **************
% -------------------------------------------------------------------------
% Para que las matrices A y B sean mas comprensible, se reasignan valores
% a las variables estandar :

R=R_X; Kb=Kb_X; Kt=Kt_X; r=r_X; p=p_X; Ks=Ks_X; m=m_X; I=I_X; Fric=Fric_X;

% aqui solo se considera la masa m_X del motor

a22 = -2*pi*Kt*Kb/(p*R*(2*pi*I/p + m*r*Ks));
b21 =   Kt/(R*(2*pi*I/p + m*r*Ks));

w21 =  -Ks*r/(2*pi*I/p + m*r*Ks);

A_X = [ 0       1;     0     a22 ];

B_X = [   0 ;   b21];

Wf_X =[ 0 ;    w21 ];

C_X = [ 1 0 ];
D_X = 0;

% Ecuacion para el calculo de la corriente del motor X
a_x = -2*pi*Kb/(p*R);
b_x = 1/R;
```

```
% Matrices del nuevo sistema con accion integral

Ai_X = [ 0       1    0 ;   0     a22    0;    1       0    0 ];

Bi_X = [ 0
         b21
         0 ];

% **********  Ecuaciones de estado del sistema del motor Y *************
% -----------------------------------------------------------------------

% Para que las matrices A y B sean mas comprensible, se reasignan valores
% a las variables estandar :

R=R_Y; Kb=Kb_Y; Kt=Kt_Y; r=r_Y; p=p_Y; Ks=Ks_Y; m=m_Y+m_X+M_X; I=I_Y; Fric=Fric_Y;

a22 = -2*pi*Kt*Kb/(p*R*(2*pi*I/p + m*r*Ks));
b21 =   Kt/(R*(2*pi*I/p + m*r*Ks));
w21 =  -Ks*r/(2*pi*I/p + m*r*Ks) ;

A_Y = [ 0      1;    0      a22];

B_Y = [ 0 ;  b21];

Wf_Y = [ 0;  w21];

C_Y = [ 1   0 ];

D_Y = 0;

% Ecuacion para el calculo de la corriente del motor Y
a_y = -2*pi*Kb/(p*R);
b_y = 1/R;

% Matrices del nuevo sistema con accion integral

Ai_Y = [ 0       1    0  ;    0     a22    0;     1       0    0 ];

Bi_Y = [ 0
         b21
         0 ];
```

En este punto del programa, ingresamos los pesos para ambos sistemas, los pesos que se obtuvo al realizar las pruebas se muestran en las conclusiones. Una vez ingresado los pesos aquí calculamos la matriz de ganancias Kix, Kiy, los cuales nos servirán en nuestro control.
Luego también discretizamos las matrices A, B y la de Wf.

```
%*************** Aqui comienza el control **************************

trayec;                 %llamada a la funcion que me genera mis
                        %trayectorias a partir de la velocidad dada

load curvas;            %cargo los valores: Vel_total ruta R t dt;
                        %obtenidos en la funcion 'trayec'


% Ingreso de valores de los pesos
% -------------------------------
disp('Matriz Qi para determinar la ganancia del controlador K');
disp('------------------------------------------------------');
disp('    ');

disp('ingrese los pesos del motor X');
disp(' ...........................');
disp('    '); disp('    ');
```

```
q1x = input('Introducir factor de peso q1 [x]     : ');
q2x = input('Introducir factor de peso q2 [dx/dt] : ');
q3x = input('Introducir factor de peso q3 [int]   : ');
disp('   '); disp('   ');

%aqui estan los mejores pesos hallados para el eje Y

%q1x = 0;
%q2x = 0;
%q3x = 1e8;

disp('ingrese los pesos del motor Y');
disp(' ............................');
disp('   '); disp('   ');

q1y = input('Introducir factor de peso q1 [x]     : ');
q2y = input('Introducir factor de peso q2 [dx/dt] : ');
q3y = input('Introducir factor de peso q3 [int]   : ');

%aqui estan los mejores pesos hallados para el eje Y
%q1y = 0;
%q2y = 0;
%q3y = 1e8;

% Determinacion de matriz de realimentacion K
% --------------------------------------------

Qxi=diag([q1x q2x q3x]);                  %matriz de pesos del motor X
Px = are(Ai_X,Bi_X*Bi_X',Qxi);               %calculo de la ecuacion de Riccati
Kix= Bi_X'*Px;                           %ganancia del controlador

Qyi = diag([q1y q2y q3y]);               %matriz de pesos del motor X
Py = are(Ai_Y,Bi_Y*Bi_Y',Qyi);               %calculo de la ecuacion de Riccati
Kiy = Bi_Y'*Py;                          %ganancia del controlador

% Discretizando las Matrices
% ---------------------------

[Ax_d,Bx_d]=c2d(A_X,B_X,dt);
[Ax_d,Wfx_d]=c2d(A_X,Wf_X,dt);

[Ay_d,By_d]=c2d(A_Y,B_Y,dt);
[Ay_d,Wfy_d]=c2d(A_Y,Wf_Y,dt);
```

Definimos las condiciones iniciales para la velocidad y posición en X e Y. Luego realizamos el control dentro del lazo for. También calculamos la corriente para después estimar la potencia instantánea dentro de toda la trayectoria.

```
% Condiciones Iniciales para simulacion
% -------------------------------------
x(:,1) = [0 0]';    %condiciones iniciales  para la velocidad y posicion en X
y(:,1) = [0 0]';    %condiciones iniciales  para la velocidad y posicion en Y
errint_x=0;          %condiciones iniciales para el error
errint_y=0;          %condiciones iniciales para el error


% En la matriz ruta() se encuentran las referencias constantes por tramos
% ----------------------------------------------------------------------
for i=1:length(t)

     errint_x = errint_x + (x(1,i) - ruta(1,i))*dt;    %error de posicion X
     errint_y = errint_y + (y(1,i) - ruta(2,i))*dt;    %error de posicion Y

     ux = -Kix*[x(:,i);errint_x];       %señal de control: u(voltaje)
```

```
     uy = -Kiy*[y(:,i);errint_y];    %señal de control: u(voltaje)
     volt_x(1,i)=ux;                     %almaceno señal de control (voltaje)
     volt_y(1,i)=uy;                     %almaceno señal de control (voltaje)

     Ffx = Fric_X*x(2,i);                %constante friccion por la velocidad
     Ffy = Fric_Y*y(2,i);                %constante friccion por la velocidad

     if i==length(t)
     break;
     end

   x(:,i+1) = Ax_d*x(:,i) + Bx_d*ux + Wfx_d*Ffx; %se almacena los valores de:
                                       posicion y velocidad en X
   y(:,i+1) = Ay_d*y(:,i) + By_d*uy + Wfy_d*Ffy; %se almacena los valores de:
                                       posicion y velocidad en Y

end

% Calculo de la Corriente de los motores
ix = a_x*x(2,:) + b_x*volt_x;
iy = a_y*y(2,:) + b_y*volt_y;

% Calculo de la Potencia consumida en los motores
Potx = ix.*volt_x;
Poty = iy.*volt_y;
```

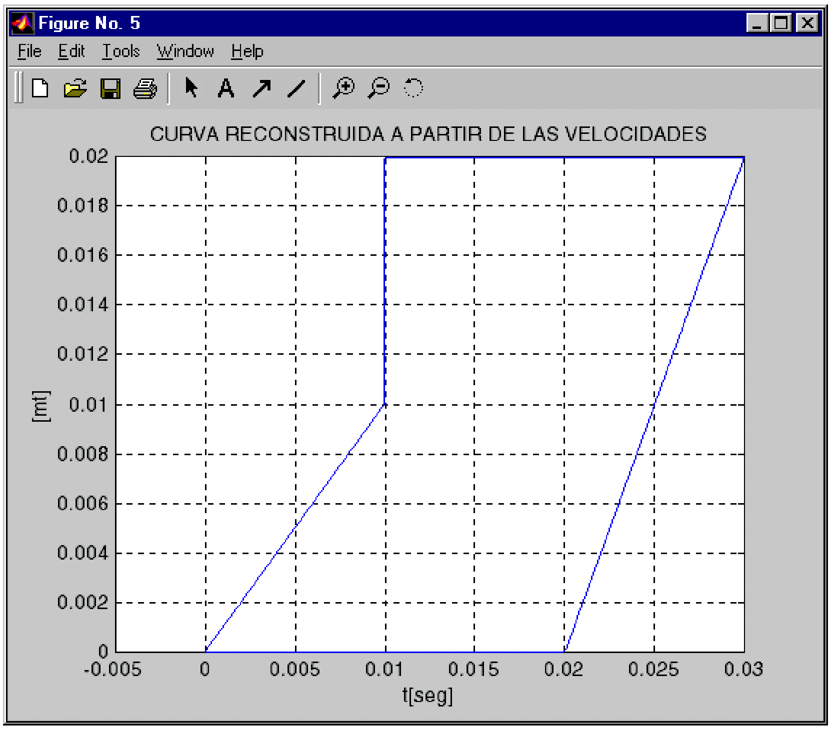
Finalmente ploteamos los gráficos más importantes.

```
figure;
plot(t,x(1,:),t,ruta(1,:),':r');        %trayectoria real vs  trayectoria de
referencia
xlabel('t [seg]'); ylabel('x [mt]');
title('Posicion x');grid;
legend('Posicion obtenida','Referencia');

figure;
plot(t,x(2,:));                        %velocidad X
xlabel('t [seg]'); ylabel('dx/dt [mt/seg]');
title('Velocidad x'); grid;

%-----------------------------------------
figure;
plot(t,y(1,:),t,ruta(2,:),':r');         %trayectoria real vs trayectoria de
refernecia
```

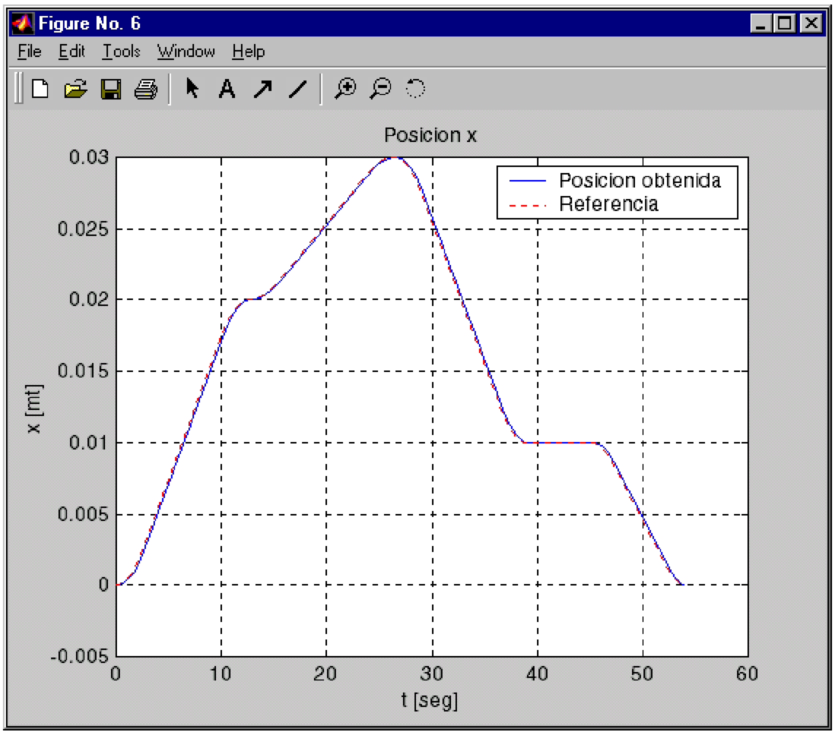
```
xlabel('t [seg]'); ylabel('y [mt]');
title('Posicion y'); grid;
legend('Posicion obtenida','Referencia');

figure;
plot(t,y(2,:));                          %velocidad Y
xlabel('t [seg]');  ylabel('dy/dt [mt/seg]');
title('Velocidad y'); grid;

figure;
plot(t,volt_x,t,volt_y,':b');
xlabel('t [seg]'); ylabel('u [voltios]');
title('Esfuerzo de control'); grid;
legend('Voltaje en X','Voltaje en Y');

figure;
plot(x(1,:),y(1,:));
title('Forma generada con el control');
xlabel('x [mt]'); ylabel('y [mt]');grid;

figure;
plot(t,Potx,':b');
hold on;
plot(t,Poty);
xlabel('t [seg]'); ylabel('potencia]');
title('Potencias instantaneas');
legend('Potencia Px','Potecia Py');
grid;
hold off
```

4. Visualización de las gráficas pedidas.

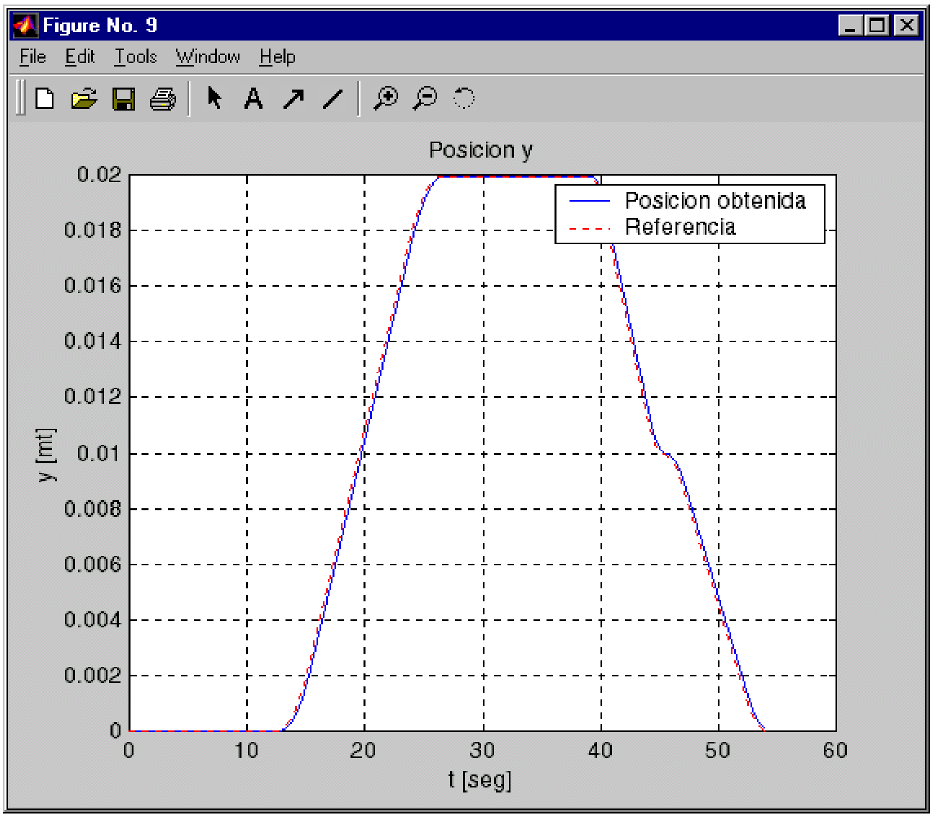
- Aquí observamos la trayectoria deseada X-Y:



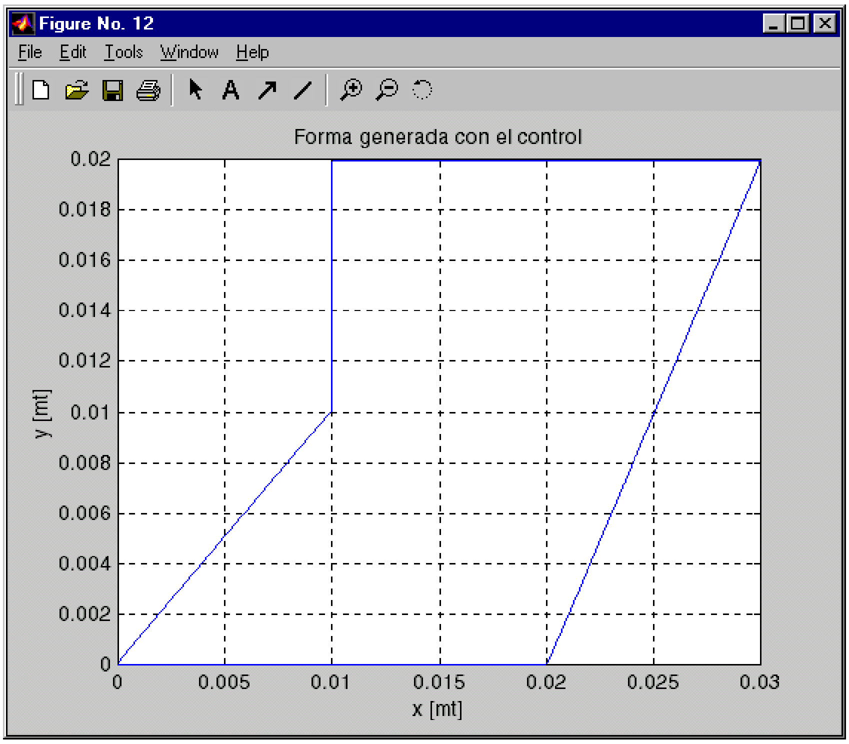
- Aquí observamos la trayectoria en X realizada por la cuchilla y comparada con la deseada. Como se observa no se nota la diferencia:



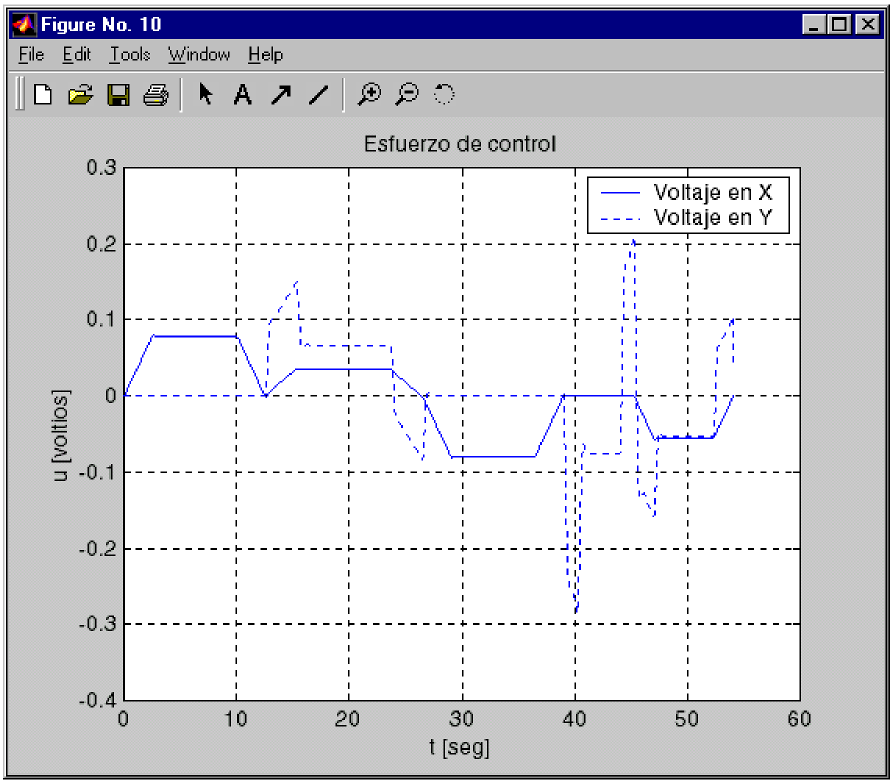
- Aquí observamos la trayectoria en Y realizada por la cuchilla y comparada con la deseada. Como se observa no se nota la diferencia:



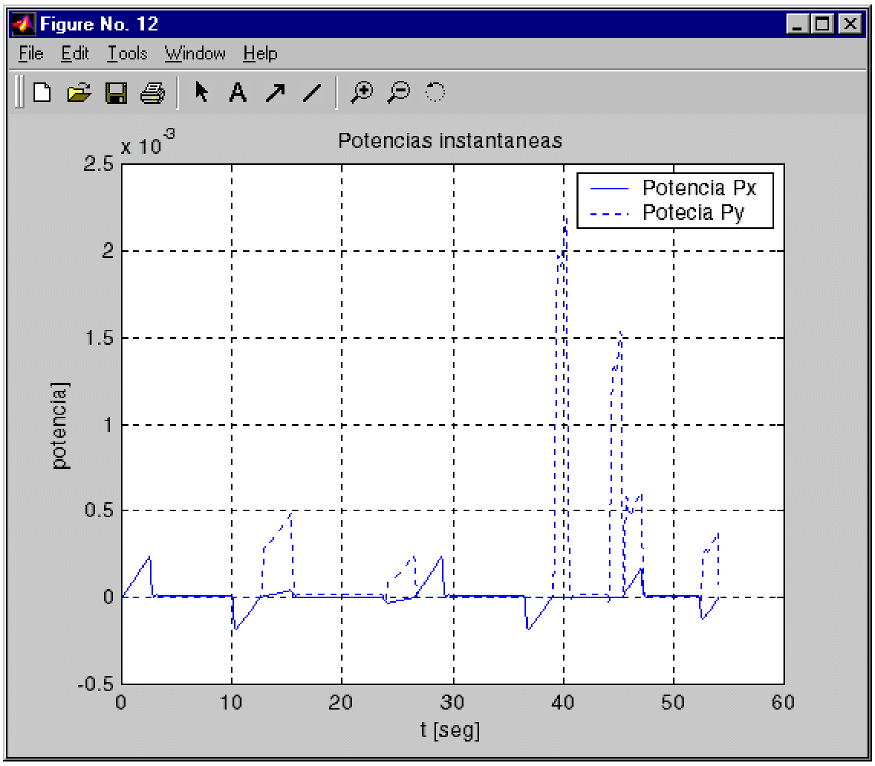
- Aquí observamos la trayectoria en X-Y realizada por la cuchilla, como se aprecia es semejante a la primera figura mostrada:



- Aquí observamos el valor de voltaje aplicado a cada uno de los motores en función del tiempo. Como se aprecia es bastante pequeño:



- Aquí observamos la potencia desarrollada por cada unos de los motores. Como vemos la potencia es mínima.



5. Análisis y Conclusiones

Al ejecutar el programa **torno.m** se presenta los siguientes mensajes para introducir los valores velocidad, porcentaje de rampa y dt. Luego se ingresa los tres pesos para cada motor:

Se realizaron varias pruebas que consistían en la variación de la velocidad así como la variación de los pesos con el objetivo de encontrar la mejor forma de la figura formada por la cuchilla del torno.

- Los valores de las velocidades en los tramos, se hizo variar en el rango de **0.001- 0.004m/s**
- Los valores del porcentaje de tiempo de rampa de la velocidad se fijo en: **0.2%**
- Los valores del diferencial del tiempo se fijo en **0.02**
- También se hizo variar el peso q3 en ambos motores, entre los rangos de **1e6 - 1e11**

Se observa que a medida que se aumenta la velocidad es más difícil que la cuchilla forme la figura deseada; aunque esto se puede realizar aumentando considerablemente el peso q3x,q3y; es decir; si aumento la velocidad de manera que la figure se realice mas rápido, entonces deberé aumentar el peso q3 demasiado.
Sin embargo si la velocidad esta dentro del rango mencionado arriba se logra obtener la figura muy aproximada con pesos relativamente moderados, aunque son algo elevados.

En vista a esto, se recomienda ingresar los siguientes valores:

```
Velocidad:          0.002
% tiempo rampa:  0.2
diferencial dt      : 0.02
q1x  = 0
q2x  = 0
q3x  = 1e8
q1y  = 0
q2y  = 0
q3y  = 1e10
```

»torno

Ingrese velocidad resultante   (m/s) : 0.002
Ingrese el porcentaje de rampa   (%) : 0.2
Ingrese diferencial de tiempo   (ms) : 0.02

Matriz Qi para determinar la ganancia del controlador K
--

ingrese los pesos del motor X
…………...........................

Introducir factor de peso q1 [x]   : 0
Introducir factor de peso q2 [dx/dt] : 0
Introducir factor de peso q3 [int]   : 1e8

ingrese los pesos del motor Y
………….. ....

Introducir factor de peso q1 [x]   : 0
Introducir factor de peso q2 [dx/dt] : 0
Introducir factor de peso q3 [int]   : 1e10
»

Es este valor el que **se recomienda** usar para formar la figura con el torno, a una velocidad aceptable y con pesos relativamente moderados altamente. Si observamos las gráficas la construcción de la figura en este caso se realiza en aproximadamente **54 segundos.**
Los pesos q1, q2 en ambos sistemas no tienen mucho efecto por lo que se mantiene su valor a cero.
Se podría bajar el peso q3 del motor Y **a q3y= 1e8** igual que para el motor X y aun se mantiene la forma deseada de la figura, con una ligera deformación.

Si luego aumento la velocidad, por ejemplo a **0.004m/s** entonces los pesos mínimos adecuados para que me reconstruya bien la forma de la figura son para ambos casos: **q3x= q3y = 1e11**, lo cual es demasiado elevado y tal vez podría ser difícil implementarlo en un sistema con procesador. En este caso el tiempo de fabricación de la pieza se reduciría a **27 segundos** aproximadamente como se muestra en las figuras pero a costa de tener pesos exageradamente elevados.
En este caso hay un compromiso entre reducir el tiempo de la fabricación de la pieza que se logra aumentando la velocidad a costa de aumentar extremadamente los pesos q3x, q3y.
Como observamos en las figuras de potencia y voltaje, las restricciones se mantienen holgadamente.
Además como notamos la potencia y el voltaje desarrollado en el motor Y es mas elevado que el desarrollado en el motor X ,y esto es como se esperaba, pues el motor Y acciona una mayor carga que el motor X.

6. Descripción de las figuras que se observan al ejecutar: torno.m

Figuras generadas en la llamada a la función **trayect.m**:

Figura 1	*Representa la velocidad en el eje X que se realizara en toda la formación de la figura. Notamos que en el cuarto tramo la v=0 y en los demás es de forma trapezoidal, como se esperaba.*
Figura 2	*Representa la velocidad en el eje Y que se realizara en toda la formación de la figura. Notamos que en el primer y tercer tramo V=0 y en otros es de forma trapezoidal, como se esperaba.*
Figura 3	*Describe la trayectoria a realizarse en el eje X en un determinado tiempo.*
Figura 4	*Describe la trayectoria a realizarse en el eje Y en un determinado tiempo.*
Figura 5	***Curva recontruida a partir de las dos trayectorias en X y Y obtenidas de las dos figuras anteriores. Esta es la referencia.***

Figuras generadas desde **torno.m**:

Figura 6	*Comparación entre la referencia de trayectoria X y la trayectoria X realizada por la cuchilla.*
Figura 7	*Velocidad desarrollada en el eje X por la cuchilla*
Figura 8	*Comparación entre la referencia de trayectoria Y y la trayectoria Y realizada por la cuchilla.*
Figura 9	*Velocidad desarrollada en el eje Y por la cuchilla*
Figura 10	*Voltaje generado por el motor X y el voltaje generado por el motor Y*
Figura 11	***Figura X-Y realizado por la cuchilla del torno***
Figura 12	*Potencia desarrollada por el motor X y el motor Y durante toda la formación de la figura.*