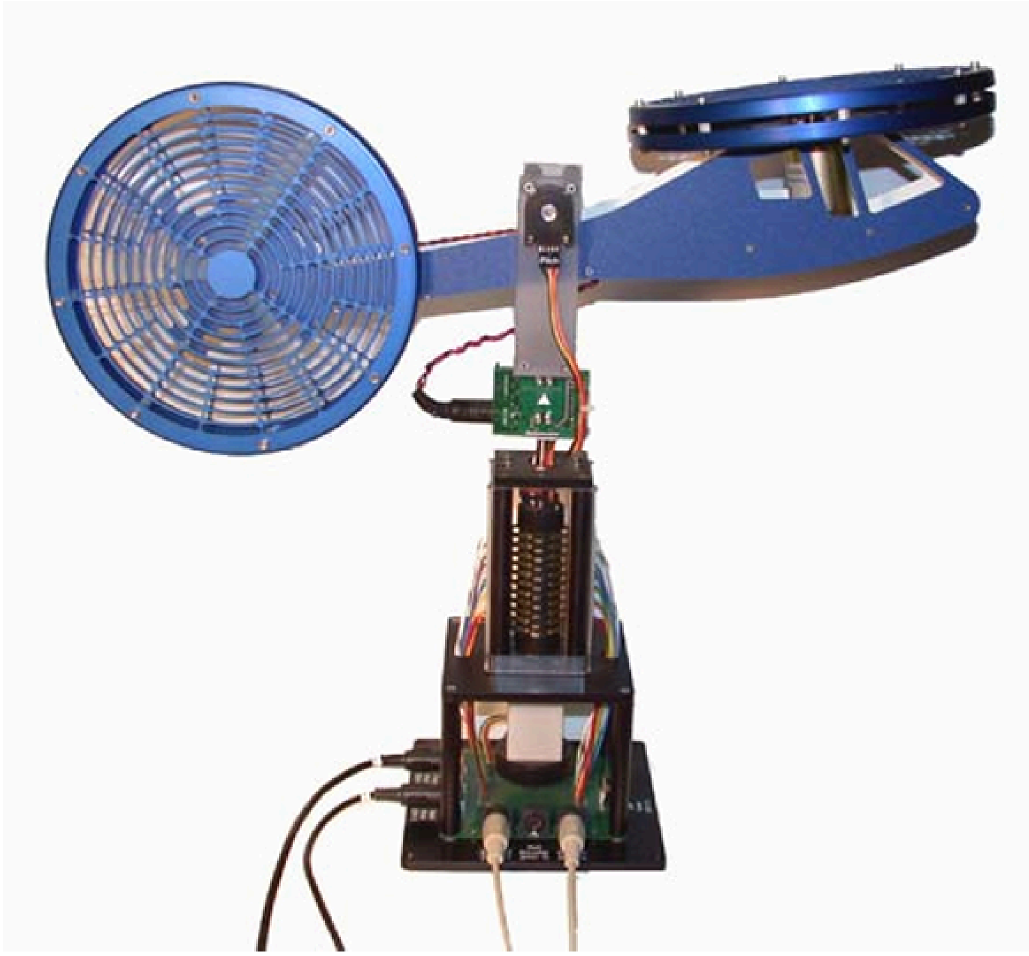
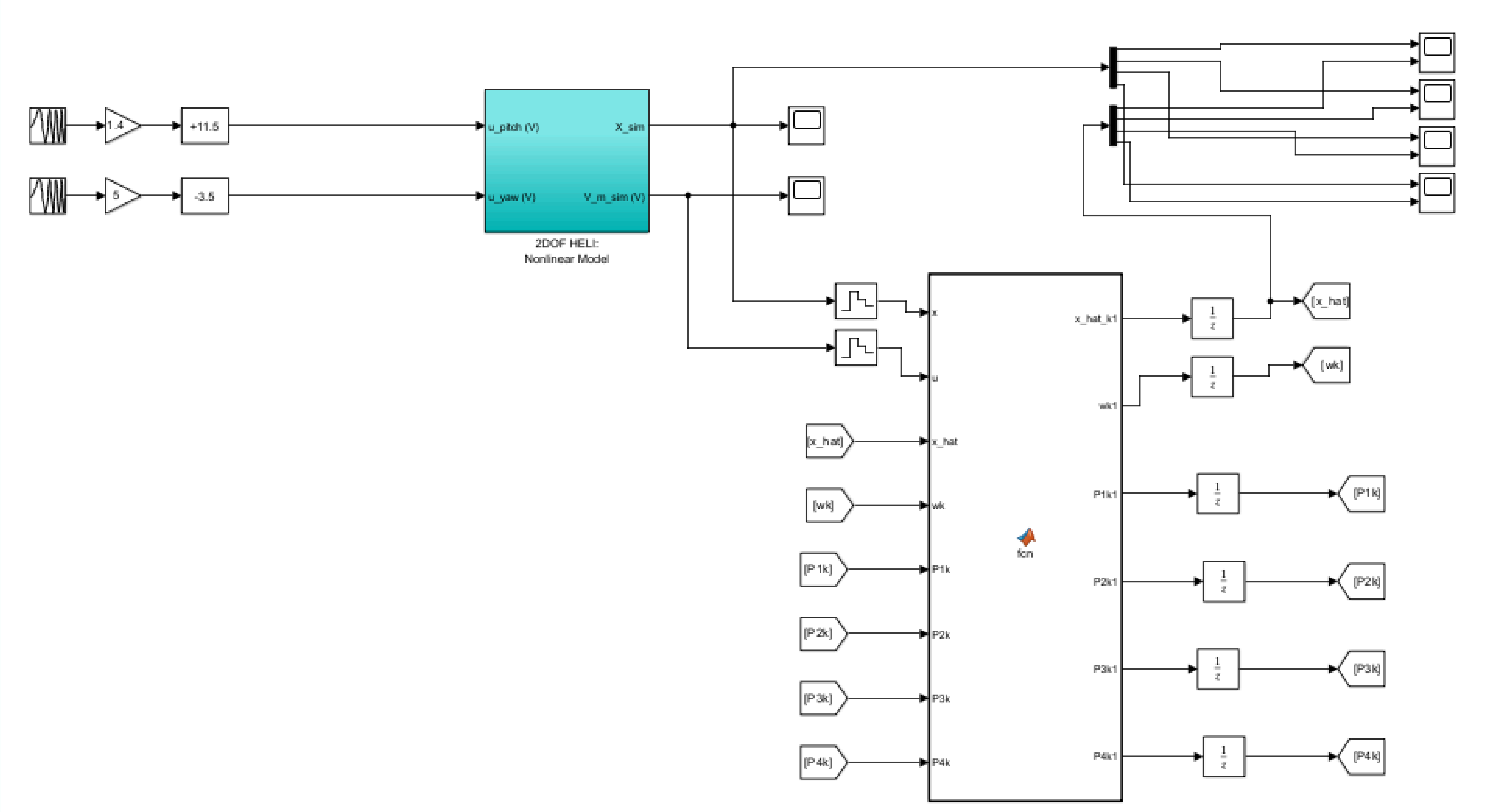
HELICOPTERO_2DOF_EKF

Implementar un identificador neuronal para el modelo del helicóptero Quanzer



Zhivago Ramos 0246042
Juan Pablo Larios 0244215
CONTROL DIFUSO

MODELO



1.4
+11.5
5
-3.5
u_pitch (V)
u_yaw (V)
X_sim
V_m_sim (V)
2DOF HELI:
Nonlinear Model
x
u
x_hat
wk
P1k
P2k
P3k
P4k
fcn
x_hat_k1
wk1
P1k1
P2k1
P3k1
P4k1
[x_hat]
[wk]
[P1k]
[P2k]
[P3k]
[P4k]

$$(J_{eq,p} + m_{heli} l_{cm}^2)\,\ddot{\theta} = K_{pp} V_{m,p} + K_{py} V_{m,y} - B_p \dot{\theta} - m_{heli} g l_{cm}$$
$$(J_{eq,y} + m_{heli} l_{cm}^2)\,\ddot{\psi} = K_{yy} V_{m,y} + K_{yp} V_{m,p} - B_y \dot{\psi} + 2 m_{heli} l_{cm}^2 \theta \dot{\psi} \dot{\theta}.$$

cross-coupling

P L A N T A

$$X = [\theta,\ \psi,\ \dot{\theta},\ \dot{\psi}]$$

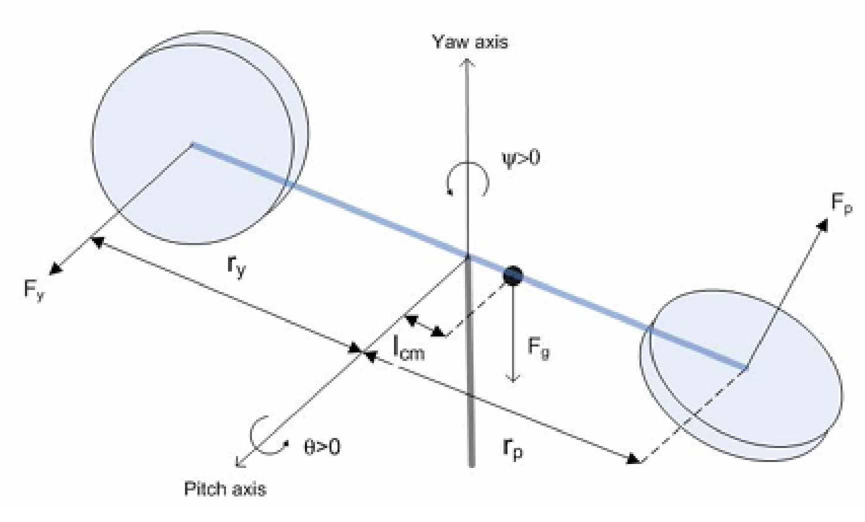


Figure 12: Dynamics of 2 DOF Helicopter.

Entradas principales
u_pitch (V): Voltaje, responsable del movimiento de pitch. (inclinación vertical)
u_yaw (V): Voltaje,genera el movimiento de yaw. (rotación horizontal)

Pitch: movimiento vertical del brazo del helicóptero
Inercia del eje (Jp)
Torque principal generado por el motor de pitch (Kpp)
Torque cruzado inducido por el motor de yaw (Kyp).
Par gravitacional, que depende del ángulo de inclinación
Amortiguamiento viscoso; fricción (Bp).
Sus salidas son el ángulo de pitch (θ) y su velocidad angular ($\dot{\theta}$).

Yaw: rotación horizontal del helicóptero
Inercia del eje (Jy)
Torque del motor lateral (Kyy)
Efecto cruzado del motor de pitch (Kpy).
Amortiguamiento; fricción (By)
Sus salidas son el ángulo de yaw (ψ) y su velocidad angular ($\dot{\psi}$).

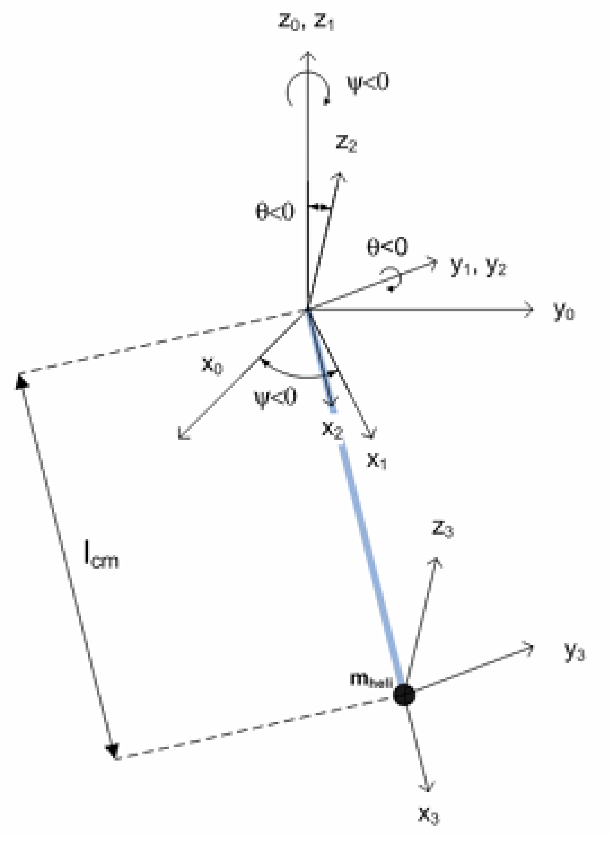


Figure 13: Kinematics of 2 DOF Helicopter.

MODELO NEURONAL

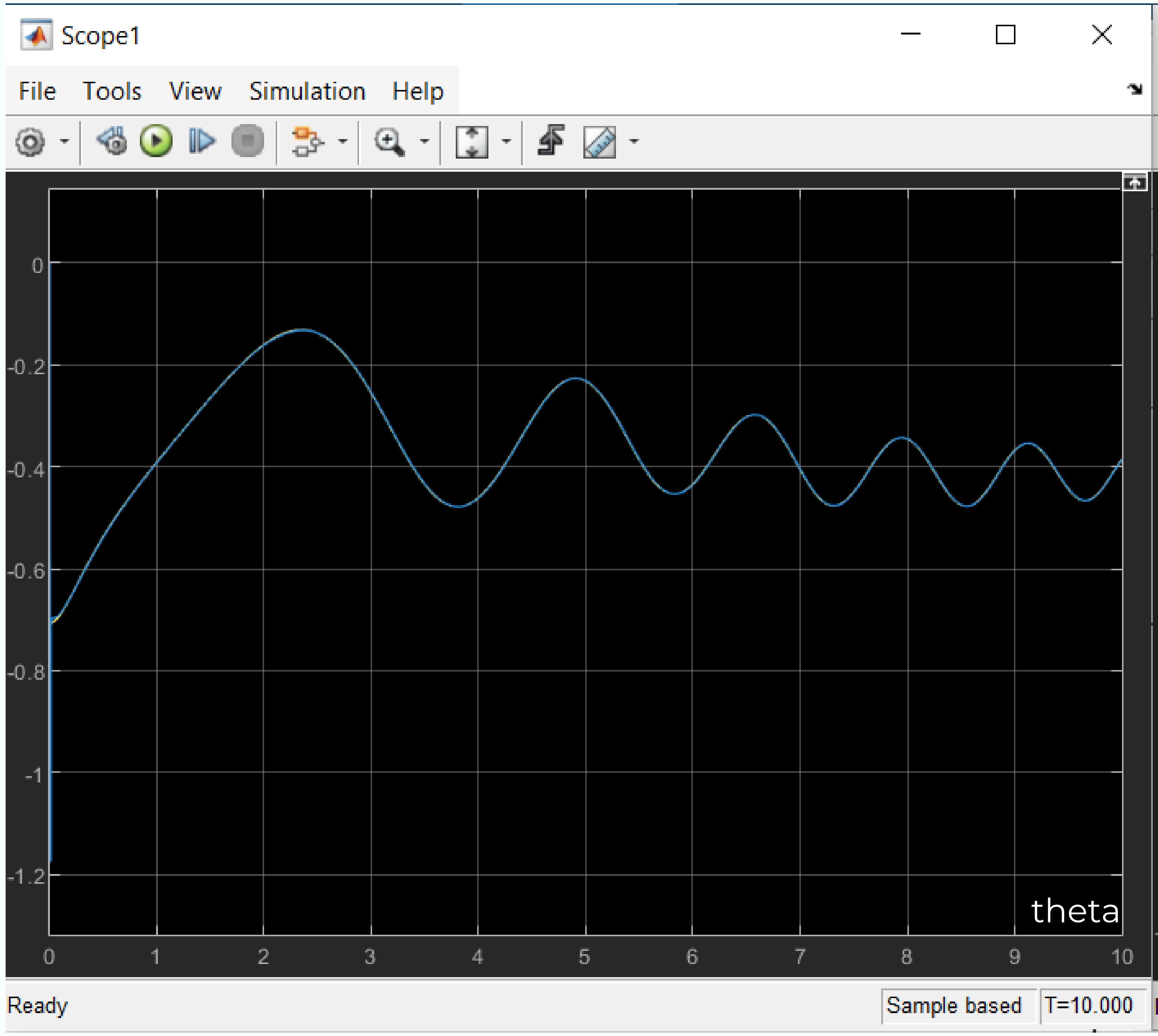
Propuesta 1

```
%% Neural states

x_hat_k_1 = wk(1)*tanh(x(3)) +  wk(2)*tanh(x(1));
x_hat_k_2 = wk(3)*tanh(x(4)) +  wk(4)*tanh(x(2));

x_hat_k_3 = wk(5)*tanh(x(3)) +  wk(6)*tanh(x(1)) + w_1*u(1);
x_hat_k_4 = wk(7)*tanh(x(4)) +  wk(8)*tanh(x(2)) + w_1*u(2);


x_hat_k1 = [x_hat_k_1;x_hat_k_2;x_hat_k_3;x_hat_k_4];
```

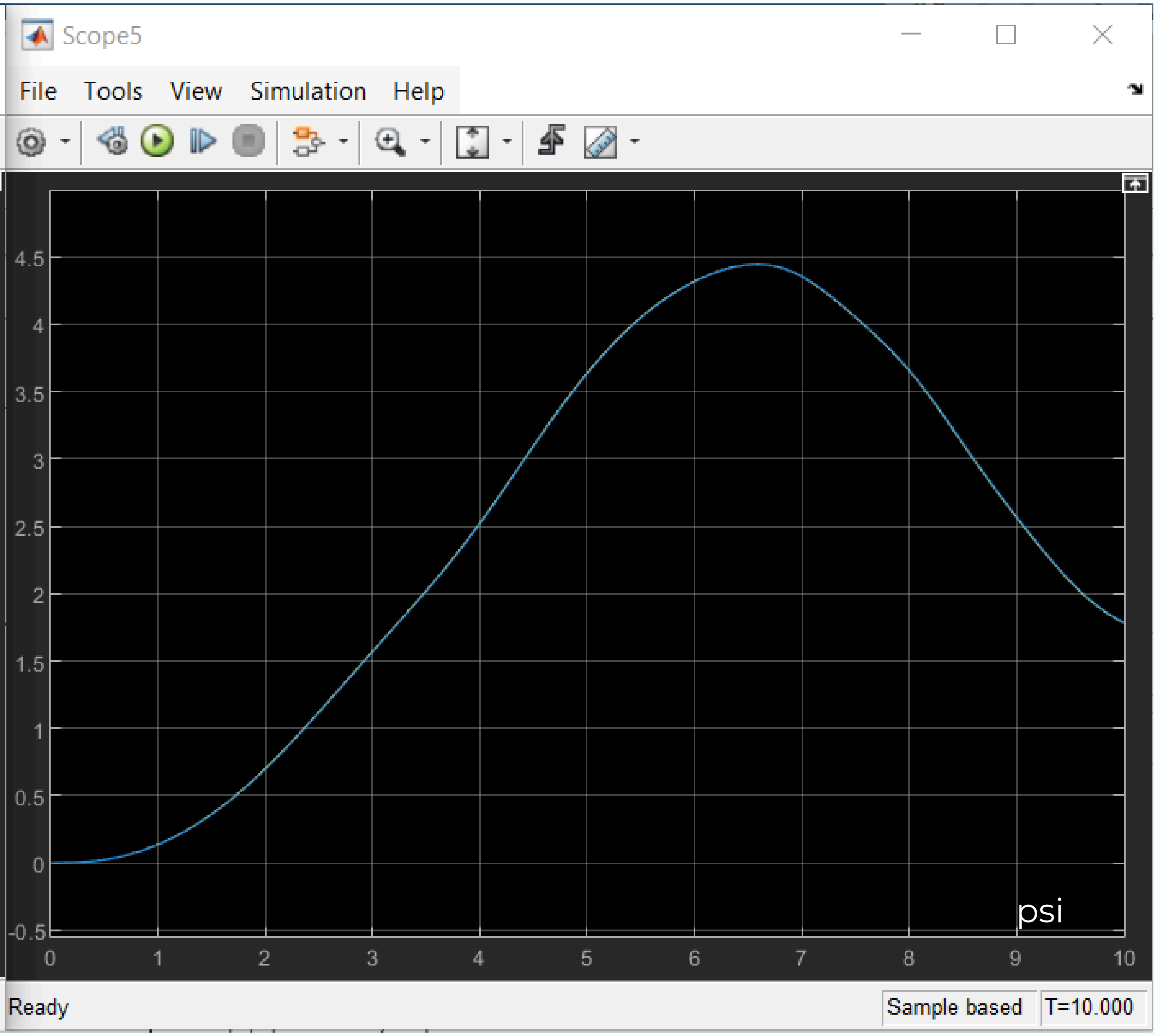
```
%% H matrix coeff

H1=[tanh(x(3)) tanh(x(1))]';
H2=[tanh(x(4)) tanh(x(2))]';
H3=[tanh(x(3)) tanh(x(1))]';
H4=[tanh(x(4)) tanh(x(2))]';
```

```
%error
error=x-x_hat;


% First filter

K1=(P1k*H1)/(R1+H1'*P1k*H1);
w1k1=[wk(1);wk(2)]+K1*error(1);
P1k1=P1k-K1*H1'*P1k+Q1;


% Second filter


K2=P2k*H2/(R2+H2'*P2k*H2);
w2k1=[wk(3);wk(4)]+K2*error(2);
P2k1=P2k-K2*H2'*P2k+Q2;


% Third filter

K3=(P3k*H3)/(R3+H3'*P3k*H3);
w3k1=[wk(5);wk(6)]+K3*error(3);
P3k1=P3k-K3*H3'*P3k+Q3;


% Fourth filter

K4=(P4k*H4)/(R4+H4'*P4k*H4);
w4k1=[wk(7);wk(8)]+K4*error(4);
P4k1=P4k-K4*H4'*P4k+Q4;


wk1=[w1k1;w2k1;w3k1;w4k1];
```

RESULTADOS

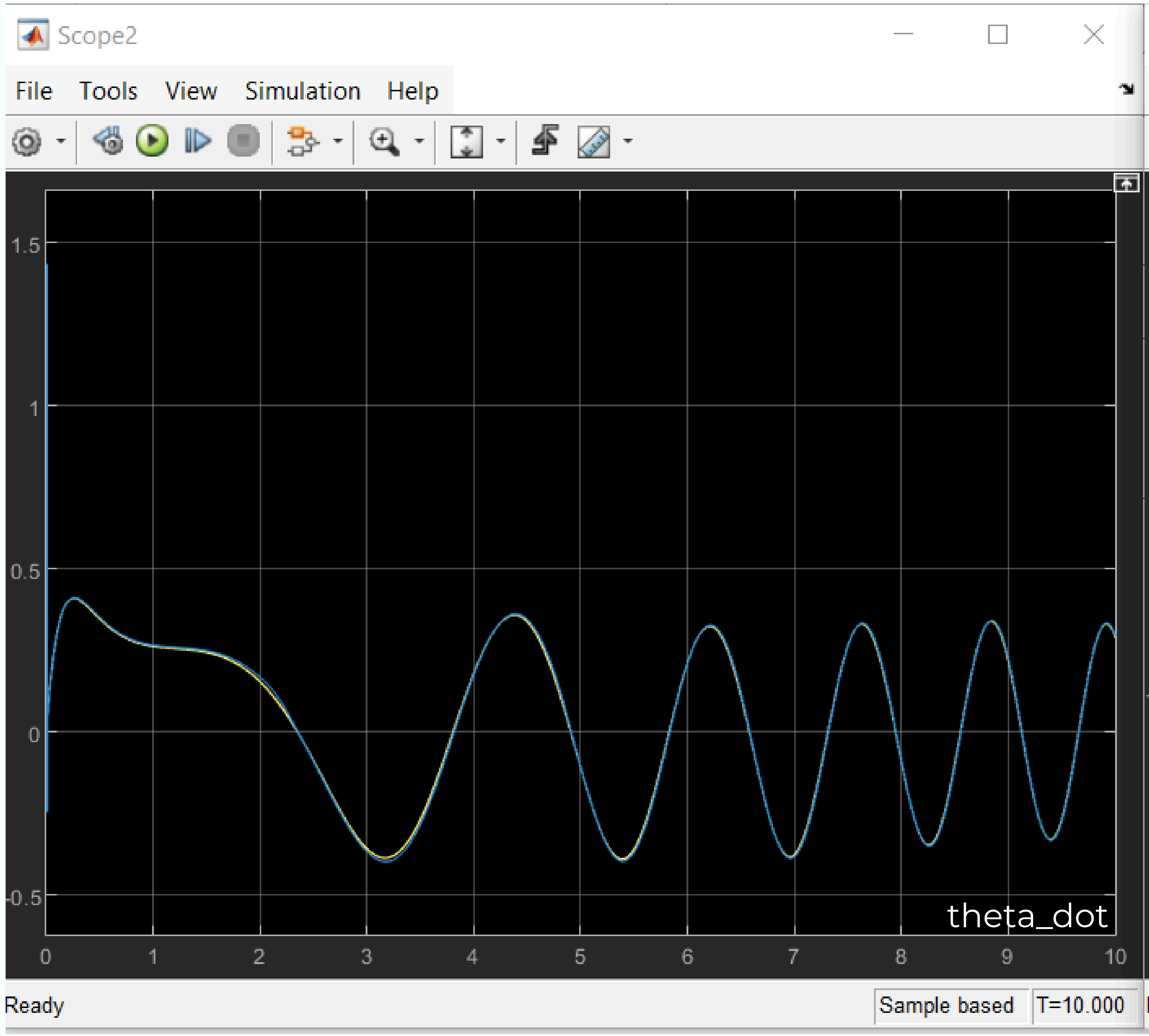


Scope1
File Tools View Simulation Help
0
-0.2
-0.4
-0.6
-0.8
-1
-1.2
0 1 2 3 4 5 6 7 8 9 10
theta
Ready
Sample based
T=10.000

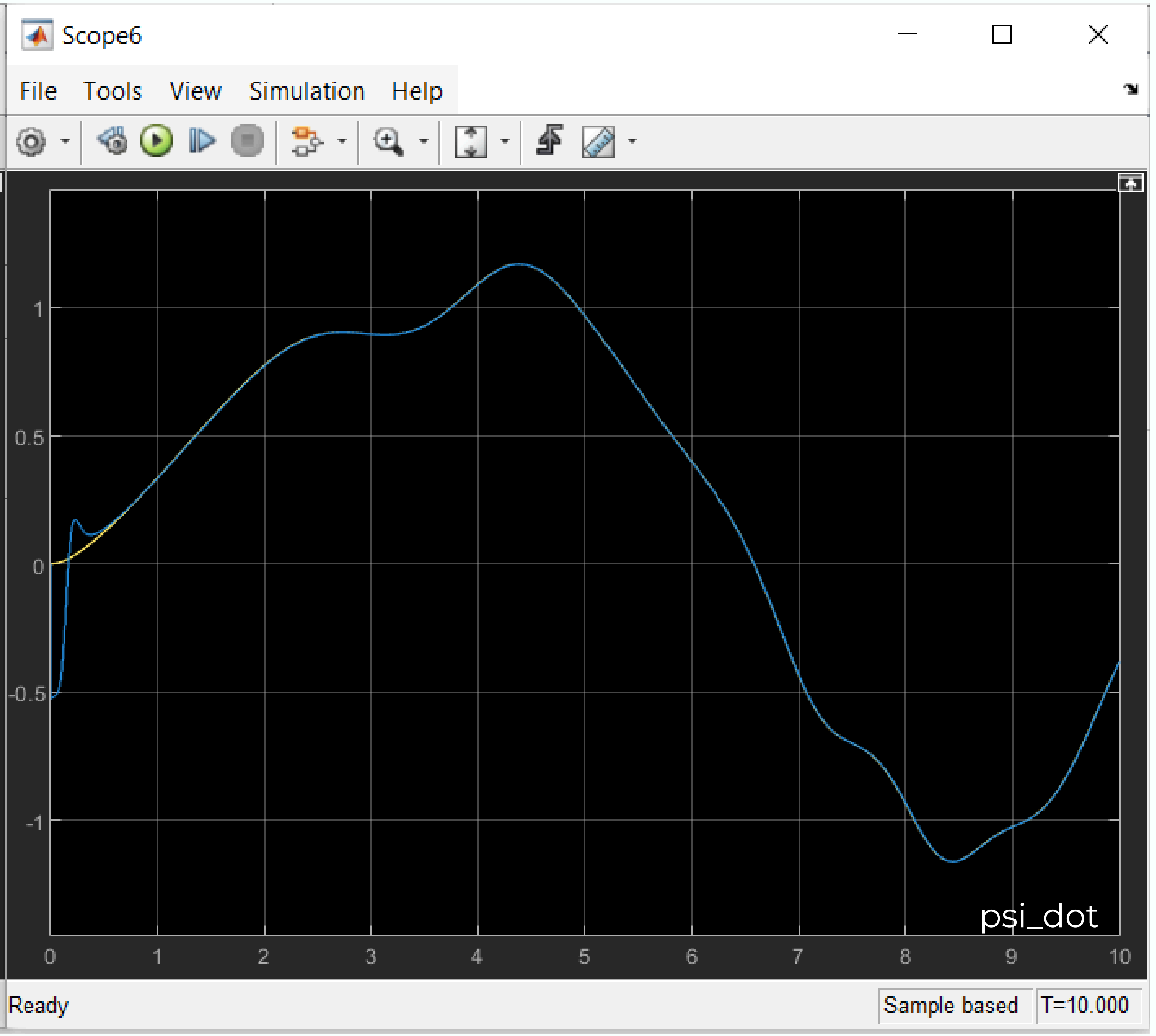


Scope5
File Tools View Simulation Help
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
-0.5
0 1 2 3 4 5 6 7 8 9 10
psi
Ready
Sample based
T=10.000

RESULTADOS



Scope2
File Tools View Simulation Help
1.5
1
0.5
0
-0.5
0 1 2 3 4 5 6 7 8 9 10
theta_dot
Ready
Sample based
T=10.000



Scope6
File Tools View Simulation Help
1
0.5
0
-0.5
-1
0 1 2 3 4 5 6 7 8 9 10
psi_dot
Ready
Sample based
T=10.000

MODELO NEURONAL

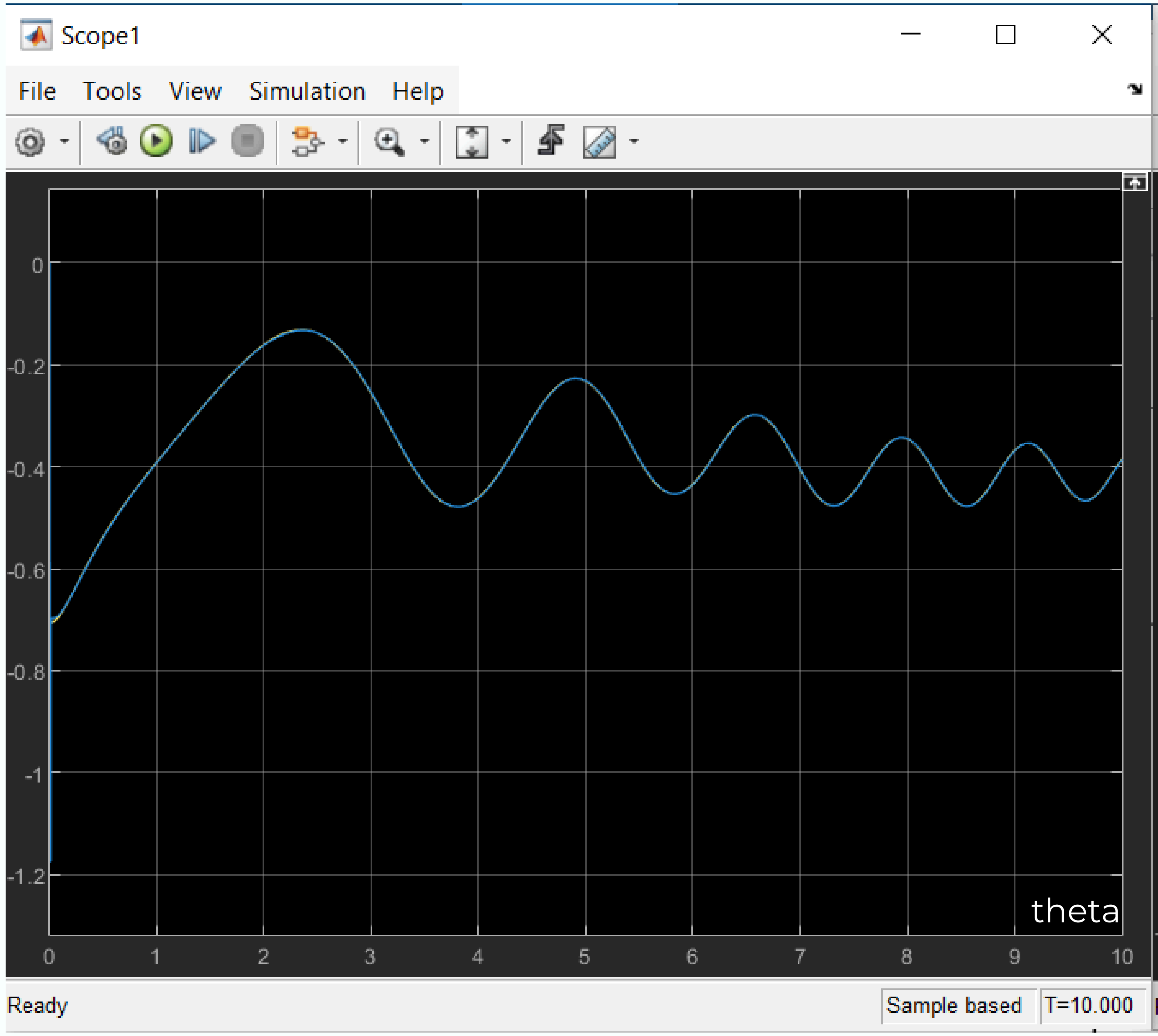
Propuesta 2

```
%% Neural states

x_hat_k_1 = wk(1)*tanh(x(3)) +  wk(2)*tanh(x(1));
x_hat_k_2 = wk(3)*tanh(x(4)) +  wk(4)*tanh(x(2));

x_hat_k_3 = wk(5)*tanh(x(3)) +  wk(6)*tanh(x(1))*cos(x(1))^2 + w_1*u(1);
x_hat_k_4 = wk(7)*tanh(x(4)) +  wk(8)*tanh(x(2))*cos(x(2)) + w_1*u(2);


x_hat_k1 = [x_hat_k_1;x_hat_k_2;x_hat_k_3;x_hat_k_4];
```

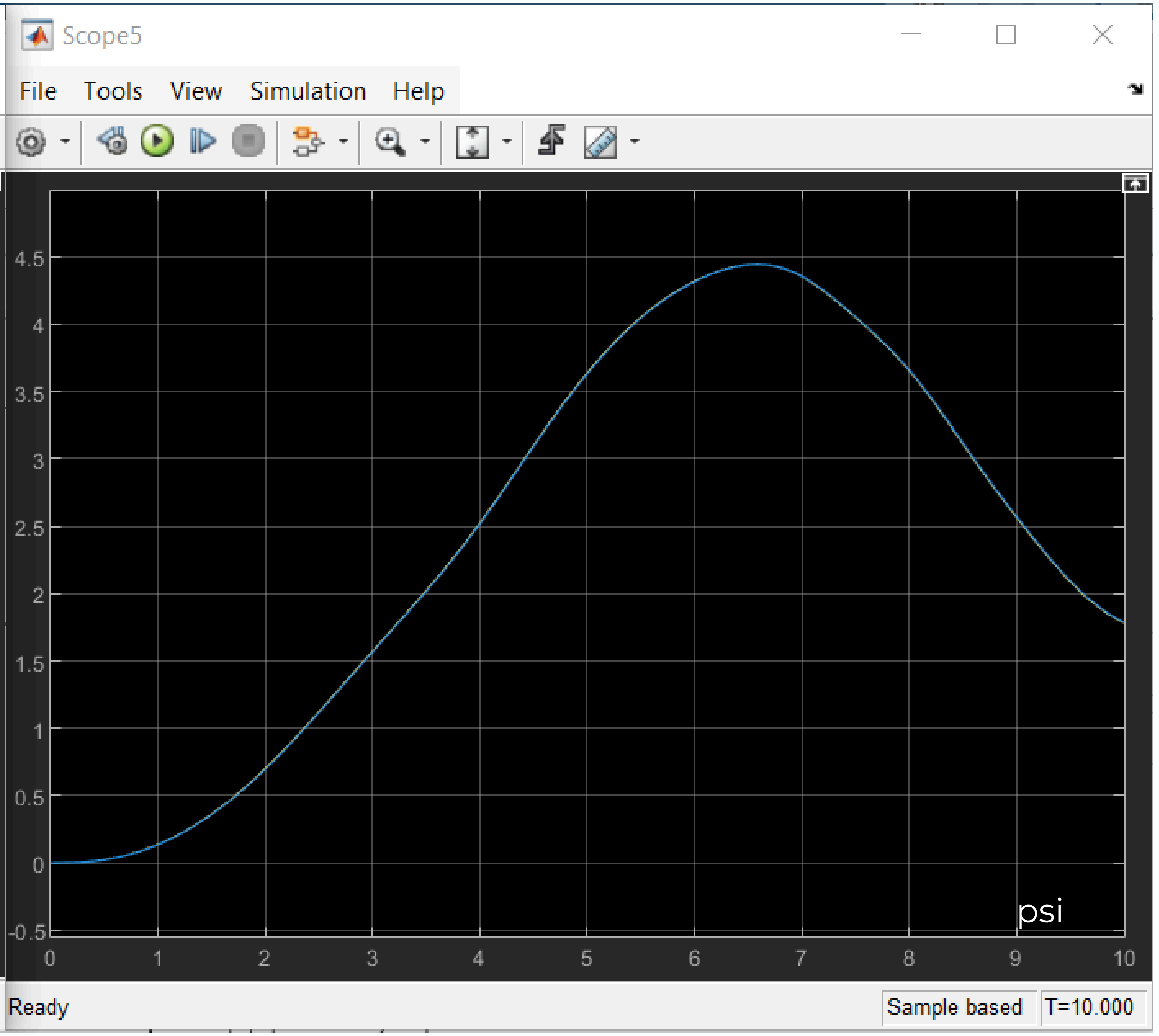
```
%% H matrix coeff

H1=[tanh(x(3)) tanh(x(1))]';
H2=[tanh(x(4)) tanh(x(2))]';
H3=[tanh(x(3)) tanh(x(1))*cos(x(1))^2]';
H4=[tanh(x(4)) tanh(x(2))*cos(x(2))]';
```

```
%error
error=x-x_hat;


% First filter

K1=(P1k*H1)/(R1+H1'*P1k*H1);
w1k1=[wk(1);wk(2)]+K1*error(1);
P1k1=P1k-K1*H1'*P1k+Q1;


% Second filter


K2=P2k*H2/(R2+H2'*P2k*H2);
w2k1=[wk(3);wk(4)]+K2*error(2);
P2k1=P2k-K2*H2'*P2k+Q2;


% Third filter

K3=(P3k*H3)/(R3+H3'*P3k*H3);
w3k1=[wk(5);wk(6)]+K3*error(3);
P3k1=P3k-K3*H3'*P3k+Q3;


% Fourth filter

K4=(P4k*H4)/(R4+H4'*P4k*H4);
w4k1=[wk(7);wk(8)]+K4*error(4);
P4k1=P4k-K4*H4'*P4k+Q4;


wk1=[w1k1;w2k1;w3k1;w4k1];
```

RESULTADOS

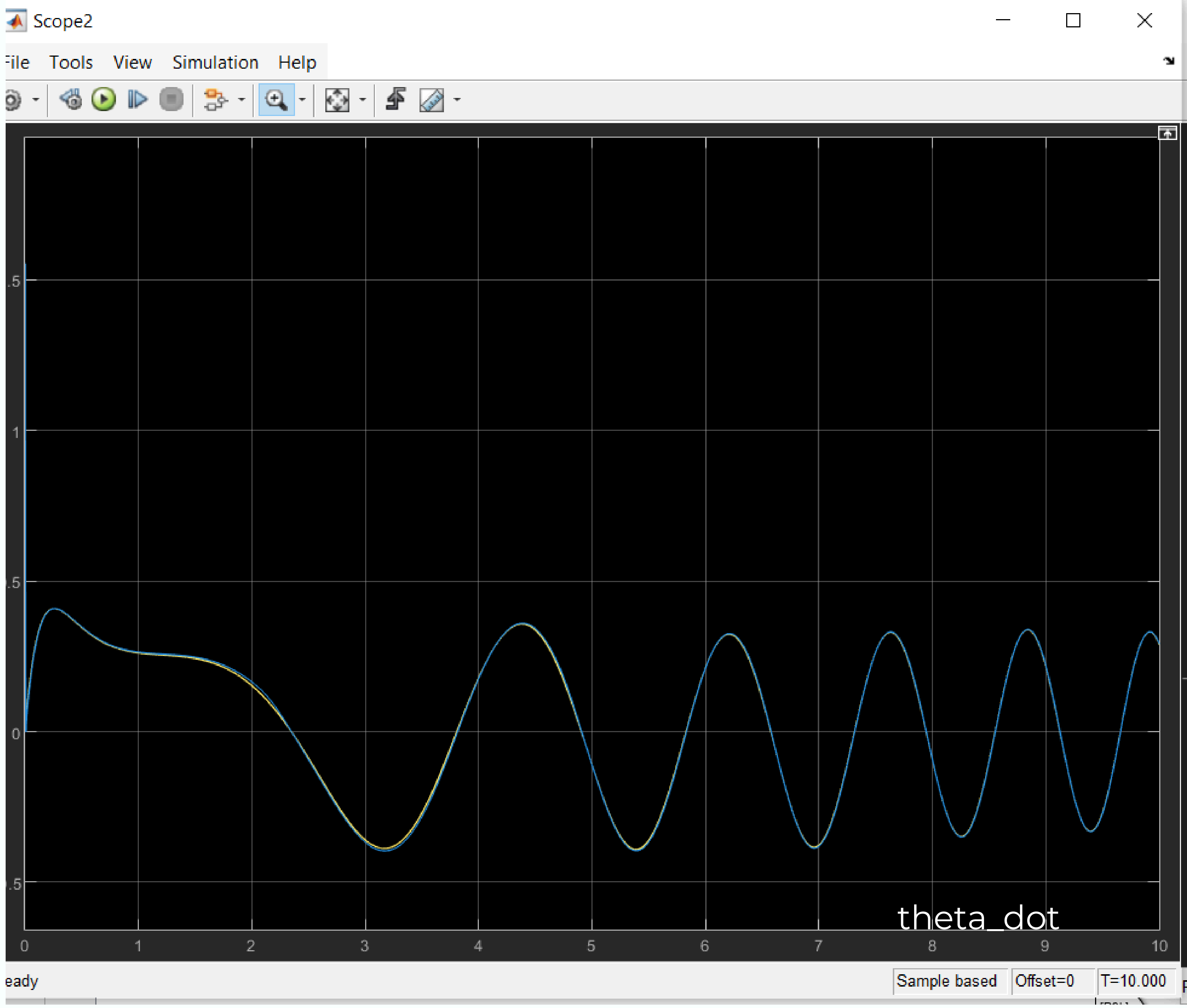


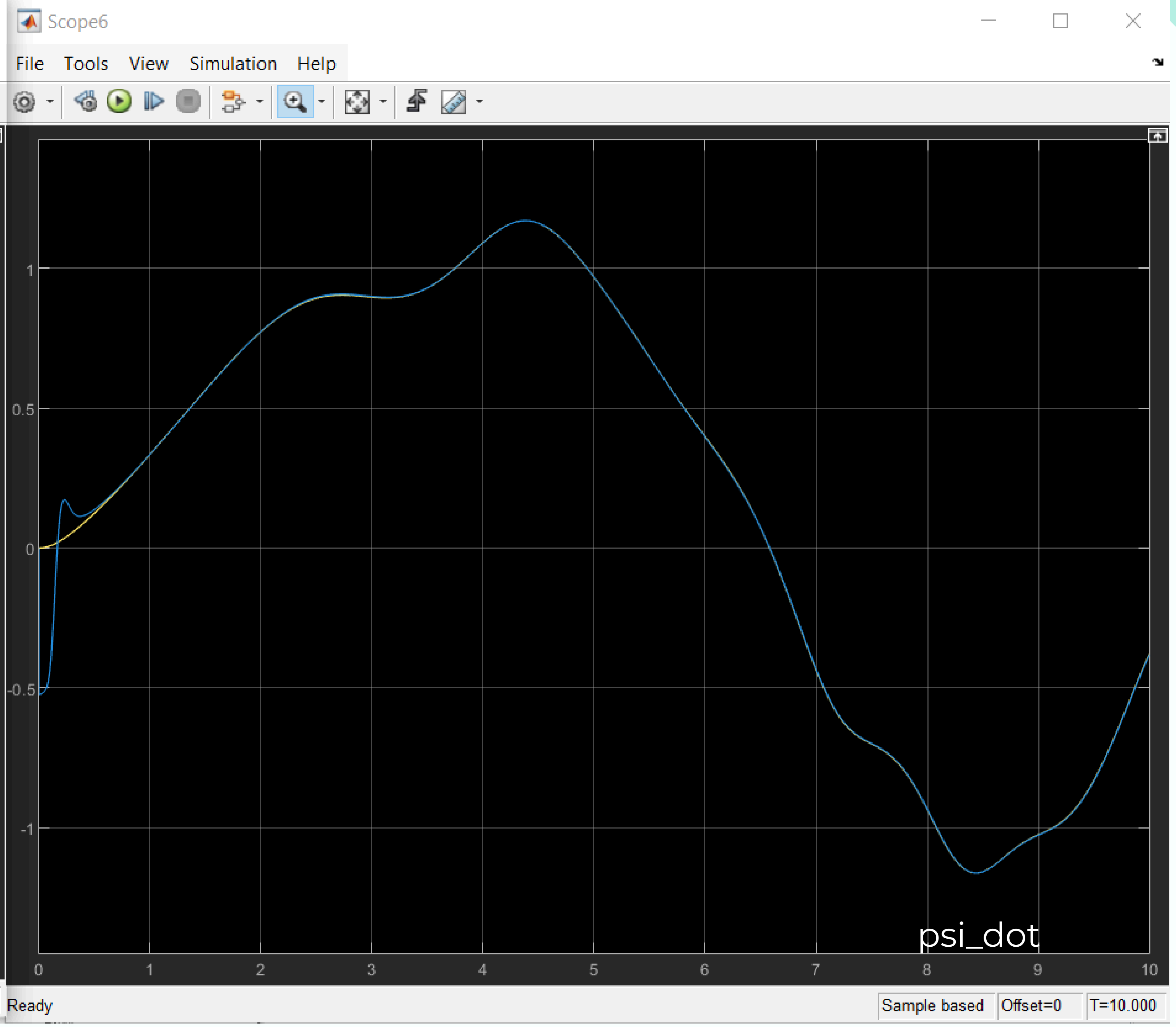
Scope1
File Tools View Simulation Help
0
-0.2
-0.4
-0.6
-0.8
-1
-1.2
0 1 2 3 4 5 6 7 8 9 10
theta
Ready
Sample based
T=10.000



Scope5
File Tools View Simulation Help
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
-0.5
0 1 2 3 4 5 6 7 8 9 10
psi
Ready
Sample based
T=10.000

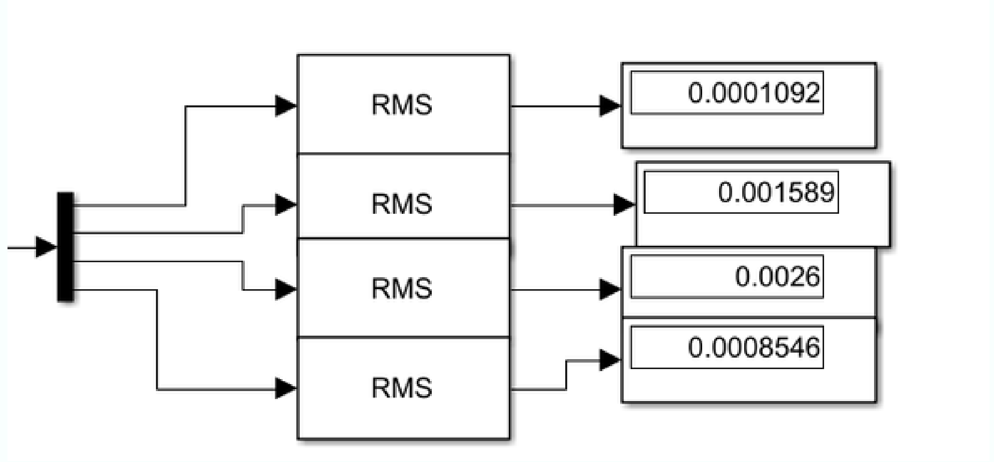
RESULTADOS


Scope2
File Tools View Simulation Help
theta_dot
Sample based
Offset=0
T=10.000

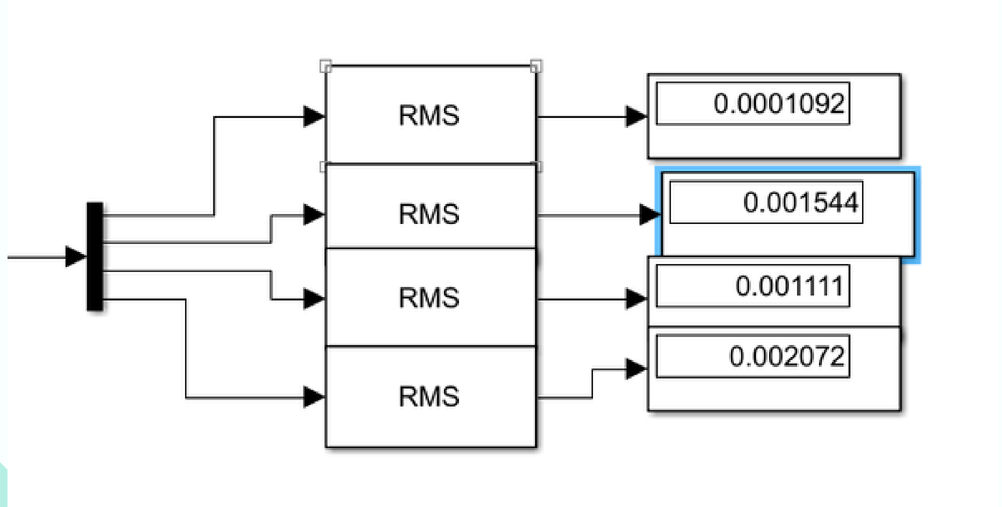

Scope6
File Tools View Simulation Help
psi_dot
Ready
Sample based
Offset=0
T=10.000

CONCLUSIONES

RHONN 1



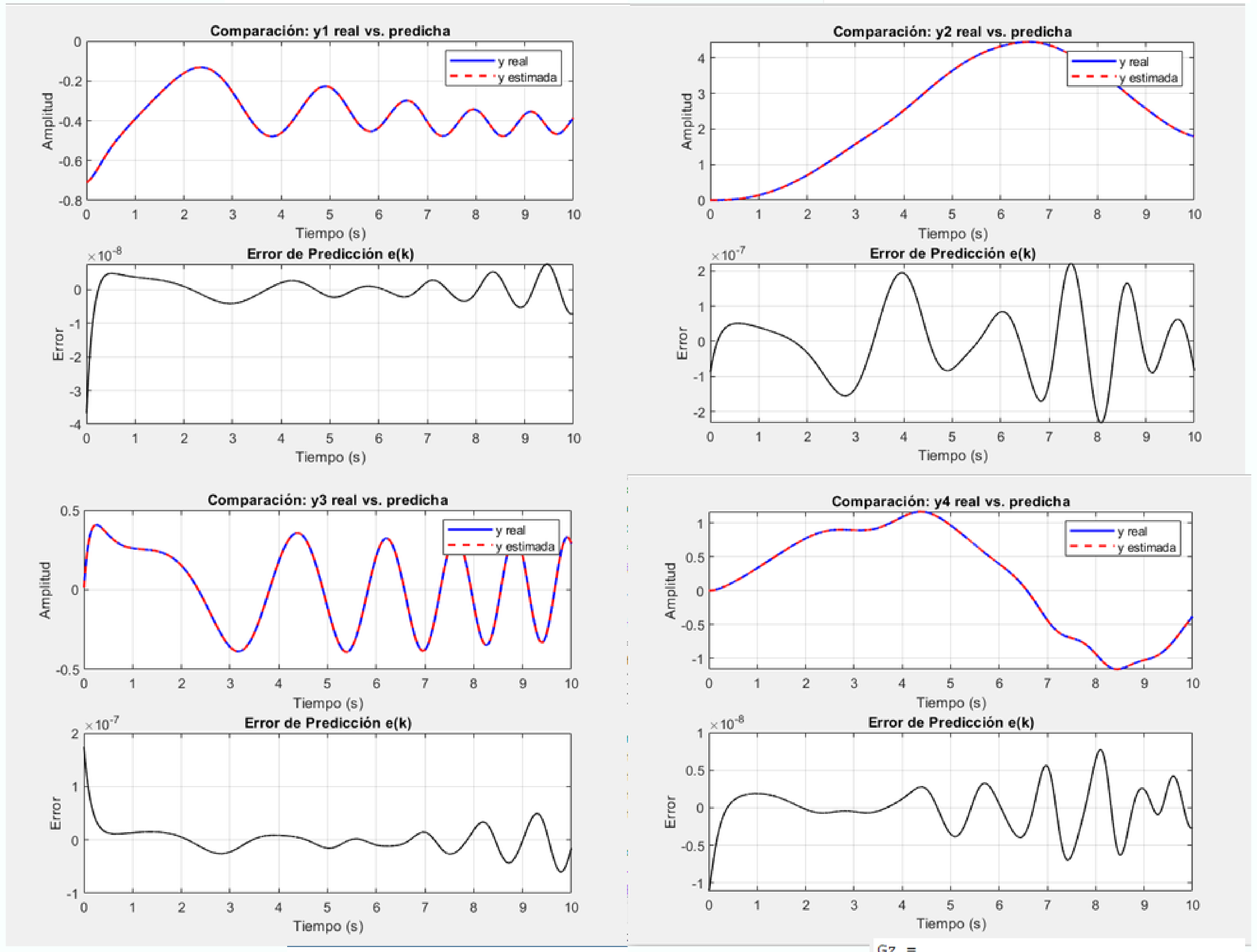
RHONN 2



EKF

```
Gz =

  1.441e-05 z^2 - 1.441e-05 z
  ----------------------------------
  z^3 - 2.989 z^2 + 2.978 z - 0.9892

Sample time: 0.001 seconds
Discrete-time transfer function.
```

```
Gz =

  9.204e-05 z^2 - 9.194e-05 z
  ----------------------------------
  z^3 - 2.458 z^2 + 1.916 z - 0.4578

Sample time: 0.001 seconds
Discrete-time transfer function.
```




```
Gz =

  5.057e-07 z^2 - 5.058e-07 z
  ----------------------------------
  z^3 - 2.999 z^2 + 2.998 z - 0.9991

Sample time: 0.001 seconds
Discrete-time transfer function.
```

```
Gz =

  -8.781e-07 z^2 + 8.78e-07 z
  ---------------------------------
  z^3 - 3.003 z^2 + 3.005 z - 1.003

Sample time: 0.001 seconds
Discrete-time transfer function.
```