



Cooperación de Sistemas Dinámicos

Tarea 2

CINVESTAV - unidad Saltillo

Enrique Benavides Téllez

26 de junio del 2020

Resumen

El presente documento contiene la segunda tarea del curso de *Cooperación de Sistemas Dinámicos* en la cual se desarrolló un observador del sistema de Lorenz en donde las ganancias para la sincronización del sistema se obtiene por medio del filtro de Kalman como segundo ejercicio utilizando un sistema transmisor se va a recuperar la señal λ como constante y variando lentamente y como tercer ejercicio obtener el mensaje encriptado del sistema en la ecuación (78) de las presentaciones de la clase.

Desarrollo

Ejercicio 1

En este ejercicio, el sistema de Lorenz mostrado en la ecuación 1 se le va a aplicar un filtro de Kalman para calcular las ganancias que permitan al observador de la ecuación 2 sincronizarse con el sistema. El sistema se encuentra alterado por ruido blanco tanto en los estados como en la salida, dichos ruidos se observan como $\epsilon_i \forall i = (1,2,3)$ y v .

$$\begin{aligned}\dot{x}_1 &= \sigma(x_2 - x_1) + \epsilon_1 \\ \dot{x}_2 &= rx_1 - x_2x_1x_3 + \epsilon_2 \\ \dot{x}_3 &= -bx_3 + x_1x_2 + \epsilon_3 \\ y &= x_3 + v\end{aligned}\tag{1}$$

$$\begin{aligned}\dot{\hat{x}}_1 &= \sigma(\hat{x}_2 - \hat{x}_1) + k_1(t)e_3 \\ \dot{\hat{x}}_2 &= r\hat{x}_1 - \hat{x}_2\hat{x}_1\hat{x}_3 + k(t)_2e_3 \\ \dot{\hat{x}}_3 &= -b\hat{x}_3 + \hat{x}_1\hat{x}_2 + k(t)_3e_3\end{aligned}\tag{2}$$

Pseudocódigo

Algorithm 1 Calculate $y = x^n$

Require: $n \geq 0 \vee x \neq 0$

Ensure: $y = x^n$

$y \leftarrow 1$

if $n < 0$ **then**

$X \leftarrow 1/x$

$N \leftarrow -n$

else

$X \leftarrow x$

$N \leftarrow n$

end if

while $N \neq 0$ **do**

if N is even **then**

$X \leftarrow X \times X$

$N \leftarrow N/2$

else $\{N \text{ is odd}\}$

$y \leftarrow y \times X$

$N \leftarrow N - 1$

end if

end while

Ejercicio 2

Ejercicio 3

Conclusión

Apendice

Códigos Matlab

```
% Ejercicio 1 - Tarea 2
close all
clear
clc
```

```
%% Sistema de Lorenz sujeto a ruido blanco en proceso y medición
```

```
% Tiempo de simulación
tf = 5;
dt = 0.01;
tspan = 0:dt:tf;
n = length(tspan);
```

```
%% Método Discreto - FKE general
```

```
% Prealocacion de variables
```

```
x = zeros(3,n); % Sistema
xg = zeros(3,n); % Observador
y = zeros(1,n); % Salida del
xg_k = zeros(3,n); %  $x(k+1|k)$ 
K = zeros(3,n); % Ganancias
P = zeros(3,3,n); % Error de c
Q = zeros(3,3,n); % Matriz de
R = zeros(1,n); % Covarianza
eps = zeros(3,n); % Ruidos del
v = zeros(1,n); % Ruido de l
e1 = zeros(1,n); % Error de v
```

```
% Ruido del proceso y lectura
```

```
mu = 0; % Centro del
sig = 0.01; % Varianza
```

```
% Condiciones iniciales
```

```
x(:,1) = rand(3,1);
xg_k(:,1) = x(:,1);
P(:, :, 1) = rand(3);
```

```
% Parametros del sistema
```

```
sigma = 10;
r = 28;
b = 8/3;
```

```
% Matrices lineales
```

```
A = [-sigma, sigma, 0; r, -1, 0; 0, 0, -b];
```

```
for k = 1:n
```

```
% Ruido del sistema y salida
```

```
eps(:,k) = mu + sig*randn(1,3);
```

```

v(1,k) = mu + sig*randn(1,1);
% Calcular Sistema con ruido
B = [0;-x(1,k)*x(3,k);x(1,k)*x(2,k)];
x(:,k+1) = x(:,k) + ( A*x(:,k) + B )*dt + eps(:,k);
y(1,k) = x(1,k) + v(1,k);

% Obtener el estimado a posteriori del estado
Bg = [0;-xg_k(1,k)*xg_k(3,k);xg_k(1,k)*xg_k(2,k)];
xg_k(:,k+1) = xg_k(:,k) + ( A*xg_k(:,k) + Bg )*dt;

% Obtener error de covarianza estimado
F = [ 1-sigma*dt,sigma*dt,0 ; (r-x(3,k))*dt,1-dt,-x(1,k)*dt ; x(2,k)*dt,x(1,k)*dt,1-b*dt;
c_e1 = covarianza(eps(1,1:k),eps(1,1:k));
c_e2 = covarianza(eps(2,1:k),eps(2,1:k));
c_e3 = covarianza(eps(3,1:k),eps(3,1:k));
% Q(:, :, k) = [c_e1,0,0;0,c_e2,0;0,0,c_e3];
Q(:, :, k) = [sig*sig,0,0;0,sig*sig,0;0,0,sig*sig];
P_k = F*P(:, :, k)*F' + Q(:, :, k);

% Obtener ganancias de Kalman
H = [1,0,0];
% R(1,k) = covarianza(v(1,1:k),v(1,1:k));
R(1,k) = sig*sig;
K(:,k) = P_k*H'*inv( H*P_k*H' + R(1,k) );

% Generar copia del sistema con ganancias de Kalman
e1(1,k) = y(1,k) - xg(1,k);
xg(:,k+1) = xg_k(:,k) + K(:,k)*(e1(1,k))*dt;

% Actualizar error de covarianza
P(:, :, k+1) = (eye(3) - K(:,k)*H)*P_k;
end

% Errores
e2 = x(2,:) - xg(2,:);
e3 = x(3,:) - xg(3,:);

%% Metodo continuo - Paper
tspan_c = [0 tf];

x0_L = [x(1,1),x(2,1),x(3,1),0,0,0,.1,.1,.1,.1,.1,.1,.1,.1];
[t_L,x_L] = ode45(@LORENZ,tspan_c,x0_L);

%% Figuras
figure(1)
set(gcf, 'Position', get(0, 'Screensize'));
% Estado 1
subplot(3,1,1)
plot(tspan,x(1,1:n),'k','linewidth',4);hold on;grid on
plot(tspan,xg(1,1:n),'g','linewidth',2)

```

```

title('Estado  $x_1$ ','interpreter','latex','fontsize',30)
xlabel({'Tiempo  $t$ }','Interpreter','latex','fontsize',20)
ylabel({' $x_1$ }','Interpreter','latex','fontsize',20)
legend({'Sistema','Observador'},','fontsize',16)

% Estado 2
subplot(3,1,2)
plot(tspan,x(2,1:n),'k','linewidth',4);hold on; grid on
plot(tspan,xg(2,1:n),'g','linewidth',2)
title('Estado  $x_2$ ','interpreter','latex','fontsize',30)
xlabel({'Tiempo  $t$ }','Interpreter','latex','fontsize',20)
ylabel({' $x_2$ }','Interpreter','latex','fontsize',20)
legend({'Sistema','Observador'},','fontsize',16)

% Estado 3
subplot(3,1,3)
plot(tspan,x(3,1:n),'k','linewidth',4);hold on; grid on
plot(tspan,xg(3,1:n),'g','linewidth',2)
title('Estado  $x_3$ ','interpreter','latex','fontsize',30)
xlabel({'Tiempo  $t$ }','Interpreter','latex','fontsize',20)
ylabel({' $x_3$ }','Interpreter','latex','fontsize',20)
legend({'Sistema','Observador'},','fontsize',16)

% set(gca,'LooseInset',get(gca,'TightInset'));
% saveas(gcf,'E1_Estados_Disc.png')

% Errores
figure(2)
set(gcf, 'Position', get(0, 'Screensize'));
plot(tspan,e1,'r','linewidth',2); hold on; grid on
plot(tspan,e2(1:n),'g','linewidth',2)
plot(tspan,e3(1:n),'b','linewidth',2)
title('Errores','interpreter','latex','fontsize',30)
xlabel({'Tiempo  $t$ }','Interpreter','latex','fontsize',20)
ylabel({' $e_i$ }','Interpreter','latex','fontsize',20)
legend({'e_1','e_2','e_3'},','fontsize',16)

% set(gca,'LooseInset',get(gca,'TightInset'));
% saveas(gcf,'E1_Errores_Disc.png')

% Retrato Fase
figure(3)
set(gcf, 'Position', get(0, 'Screensize'));
plot3(x(1,:),x(2,:),x(3:),'k','linewidth',1); hold on; grid on
plot3(xg(1,:),xg(2,:),xg(3:),'g','linewidth',1)
title('Retrato Fase','fontsize',30)
xlabel({' $x_1$ }','Interpreter','latex','fontsize',20)
ylabel({' $x_2$ }','Interpreter','latex','fontsize',20)
zlabel({' $x_3$ }','Interpreter','latex','fontsize',20)
legend({'Sistema','Observador'},','fontsize',16)

```

```

% set(gca,'LooseInset',get(gca,'TightInset'));
% saveas(gcf,'E1_RetratoFase_Disc.png')

%% Figuras sistema continuo
figure(4)
set(gcf, 'Position', get(0, 'Screensize'));
% Estado 1
subplot(3,1,1)
plot(t_L,x_L(:,1),'k','linewidth',4);hold on;grid on
plot(t_L,x_L(:,4),'g','linewidth',2);
title('Estado  $x_1$ ','interpreter','latex','fontsize',30)
xlabel({'Tiempo  $t$ ','Interpreter','latex','fontsize',20)
ylabel({' $x_1$ ','Interpreter','latex','fontsize',20)
legend({'Sistema','Observador'},,'fontsize',16)

% Estado 2
subplot(3,1,2)
plot(t_L,x_L(:,2),'k','linewidth',4);hold on;grid on
plot(t_L,x_L(:,5),'g','linewidth',2);
title('Estado  $x_2$ ','interpreter','latex','fontsize',30)
xlabel({'Tiempo  $t$ ','Interpreter','latex','fontsize',20)
ylabel({' $x_2$ ','Interpreter','latex','fontsize',20)
legend({'Sistema','Observador'},,'fontsize',16)

% Estado 3
subplot(3,1,3)
plot(t_L,x_L(:,3),'k','linewidth',4);hold on;grid on
plot(t_L,x_L(:,6),'g','linewidth',2);
title('Estado  $x_3$ ','interpreter','latex','fontsize',30)
xlabel({'Tiempo  $t$ ','Interpreter','latex','fontsize',20)
ylabel({' $x_3$ ','Interpreter','latex','fontsize',20)
legend({'Sistema','Observador'},,'fontsize',16)

% set(gca,'LooseInset',get(gca,'TightInset'));
% saveas(gcf,'E1_Estados_Cont.png')

% Errores
e1L = x_L(:,1) - x_L(:,4);
e2L = x_L(:,2) - x_L(:,5);
e3L = x_L(:,3) - x_L(:,6);

figure(5)
set(gcf, 'Position', get(0, 'Screensize'));
plot(t_L,e1L,'r','linewidth',2); hold on; grid on
plot(t_L,e2L,'g','linewidth',2)
plot(t_L,e3L,'b','linewidth',2)
title('Errores','interpreter','latex','fontsize',30)
xlabel({'Tiempo  $t$ ','Interpreter','latex','fontsize',20)
ylabel({' $e_i$ ','Interpreter','latex','fontsize',20)

```

```

legend({'e_1','e_2','e_3'},'fontsize',16)

% set(gca,'LooseInset',get(gca,'TightInset'));
% saveas(gcf,'E1_Errores_Cont.png')

% Retrato Fase
figure(6)
set(gcf, 'Position', get(0, 'Screensize'));
plot3(x_L(:,1),x_L(:,2),x_L(:,3),'b','linewidth',1); hold on; grid on
plot3(x_L(:,4),x_L(:,5),x_L(:,6),'g','linewidth',1);
title('Retrato Fase','fontsize',30)
xlabel({'$x_1$'},'Interpreter','latex','fontsize',20)
ylabel({'$x_2$'},'Interpreter','latex','fontsize',20)
zlabel({'$x_3$'},'Interpreter','latex','fontsize',20)
legend({'Sistema','Observador'},,'fontsize',16)

% set(gca,'LooseInset',get(gca,'TightInset'));
% saveas(gcf,'E1_RetratoFase_Cont.png')

%% Funciones
% FKE
function c = covarianza(x,y)
n_x = length(x);
n_y = length(y);

if n_x == n_y
% Promedio de variables
x_b = mean(x);
y_b = mean(y);
% Calculo de covarianza
c = sum( (x - x_b).*(y - y_b) )/n_x;
end

end

% Paper
function dx = LORENZ(~,x)

sigma = 10;
r = 28;
b = 8/3;
sig = 0.01;

x1 = x(1);
x2 = x(2);
x3 = x(3);

E1 = sig*randn();
E2 = sig*randn();
E3 = sig*randn();

```

```

dx(1) = sigma*(x2 - x1) + E1;
dx(2) = r*x1 - x2 - x1*x3 + E2;
dx(3) = -b*x3 + x1*x2 + E3;

v = sig*randn();
y = x1 + v;

% Filtro Kalman
p11 = x(7);
p12 = x(8);
p13 = x(9);
p21 = x(10);
p22 = x(11);
p23 = x(12);
p31 = x(13);
p32 = x(14);
p33 = x(15);

P = [p11,p12,p13;p21,p22,p23;p31,p32,p33];
F = [-sigma,sigma,0 ; r-x3,1,-x1 ; x2,x1,-b];
H = [1,0,0];
Q = [sig^2,0,0;0,sig^2,0;0,0,sig^2];
R = sig^2;

Pp = F*P + P*F' - P*H'*(1/R)*H*P + Q;

dx(7) = Pp(1,1);
dx(8) = Pp(1,2);
dx(9) = Pp(1,3);
dx(10) = Pp(2,1);
dx(11) = Pp(2,2);
dx(12) = Pp(2,3);
dx(13) = Pp(3,1);
dx(14) = Pp(3,2);
dx(15) = Pp(3,3);

K = P*[1;0;0]*(1/R);

xg1 = x(4);
xg2 = x(5);
xg3 = x(6);

e1 = y - xg1;

dx(4) = sigma*(xg2 - xg1) + K(1)*e1;
dx(5) = r*xg1 - xg2 - xg1*xg3 + K(2)*e1;
dx(6) = -b*xg3 + xg1*xg2 + K(3)*e1;

dx = dx';

```



```

end

% Ejercicio 2 - Tarea 2
close all
clear
clc

%% Sistema Transmisor de Rossler con método de identificación de parámetros lineal

tspan = [0 200];
dt = 0.1;

n = (tspan(2)/dt) + 1;
tL = zeros(1,n);
lambda = zeros(1,n);
for i = 1:n
    tL(i) = (i-1)*dt;
    lambda(i) = LAMBDA(tL(i));
end

% Sistema de Rossler con cambio de variables
x0_R = [1,1,log(1),0,0,0,0,0,0,0,0,0,rand()];
[t,x] = ode45(@(t,x)lambdaRossler(t,x),tspan,x0_R);
x(:,3) = exp(x(:,3));

% Figuras Rossler con cambio de variables
figure(1)
set(gcf, 'Position', get(0, 'Screensize'));
subplot(3,1,1)
plot(t,x(:,1),'k','linewidth',2);
title('$x_1$', 'Interpreter', 'latex', 'fontsize', 30)
xlabel('Tiempo [t]')
ylabel('Uds')

subplot(3,1,2)
plot(t,x(:,2),'k','linewidth',2);
title('$x_2$', 'Interpreter', 'latex', 'fontsize', 30)
xlabel('Tiempo [t]')
ylabel('Uds')

subplot(3,1,3)
plot(t,x(:,3),'k','linewidth',2);
title({'$Senal - x_3$'}, 'Interpreter', 'latex', 'fontsize', 30)
xlabel('Tiempo [t]')
ylabel('Uds')

set(gca, 'LooseInset', get(gca, 'TightInset'));
saveas(gcf, 'E2_Estados.png')

% Retrato Fase
figure(2)

```

```

set(gcf, 'Position', get(0, 'Screensize'));
plot3(x(:,1),x(:,2),x(:,3),'k','linewidth',1)
grid on
title('Retrato Fase Sistema','fontsize',30)
xlabel({'$x_1$'},'Interpreter','latex','fontsize',20)
ylabel({'$x_2$'},'Interpreter','latex','fontsize',20)
zlabel({'$x_3$'},'Interpreter','latex','fontsize',20)
set(gca,'LooseInset',get(gca,'TightInset'));
saveas(gcf,'E2_RetratoFase.png')

% Lambda
figure(3)
set(gcf, 'Position', get(0, 'Screensize'));
plot(tL,lambda,'g','linewidth',2); hold on; grid on
plot(t,x(:,13),'k','linewidth',2)
title('$\lambda \ y \ \hat{\lambda}$','interpreter','latex','fontsize',30)
xlabel({'Tiempo $t$'},'Interpreter','latex','fontsize',20)
ylabel({'$\lambda \ / \ \hat{\lambda}$'},'Interpreter','latex','fontsize',20)
legend({'$\lambda$', '$\hat{\lambda}$'},'interpreter','latex','fontsize',16)
set(gca,'LooseInset',get(gca,'TightInset'));
saveas(gcf,'E2_Lambda.png')

% Error de Lambda
lambdaI = interp1(t,x(:,13),tL);
eL = lambda - lambdaI;
figure(4)
set(gcf, 'Position', get(0, 'Screensize'));
plot(tL,eL,'g','linewidth',2); grid on
title('Error de $\lambda$','fontsize',30)
xlabel({'Tiempo $t$'},'Interpreter','latex','fontsize',20)
ylabel({'$error$'},'Interpreter','latex','fontsize',20)
set(gca,'LooseInset',get(gca,'TightInset'));
saveas(gcf,'E2_ErrorLambda.png')

%% Funciones
% Obtener lambda de la salida del sistema
function dx = lambdaRossler(t,x)
%% Sistema
x1 = x(1);
x2 = x(2);
x3 = x(3);
lambda = LAMBDA(t);

dx(1) = -x2 - exp(x3);
dx(2) = x1 + lambda*x2;
dx(3) = x1 + 2*exp(-x3) - 4;
y = x3;

%% Reconstruccion de Lambda

```

```

x3p = exp(x3);
u1 = -x3p;
u2 = (2/x3p) - 4;

k0 = 512;
k1 = 192;
k2 = 24;
v = 800;
gamma = 0.002;

w01 = x(4);
w02 = x(5);
w03 = x(6);
w11 = x(7);
w12 = x(8);
w13 = x(9);
w21 = x(10);
w22 = x(11);
w23 = x(12);
lambdag = x(13);
p = x(14);

dx(4) = w02;
dx(5) = w03;
dx(6) = -k0*w01 - k1*w02 - k2*w03 + y;
dx(7) = w12;
dx(8) = w13;
dx(9) = -k0*w11 - k1*w12 - k2*w13 + u1;
dx(10) = w22;
dx(11) = w23;
dx(12) = -k0*w21 - k1*w22 - k2*w23 + u2;

phi0 = k0*w01 + (k1 - 1)*w02 + k2*w03 + w12 + w21 + w23;
phi1 = w03 - w11 - w22;

yg = phi0 + lambdag*phi1;

dx(13) = -v*phi1*p*(yg - y);
dx(14) = -v*( (phi1^2)*(p^2) - gamma*p );

dx = dx';
end

function y = LAMBDA(t)
if t < 100
y = 0.3;
else
y = 0.3 + 0.2*sin((pi/25)*t);
end
end

```

```

% Ejercicio 3 - Tarea 2
close all
clear
clc

%% Sistema Transmisor de Rossler con observador adaptable para comunicaciones
tspan = [0 2];
dt = 0.01;
tspan1 = 0:dt:2;

n = (tspan(2)/dt) +1;
tL = zeros(1,n);
lambda = zeros(1,n);
for i = 1:n
    tL(i) = (i-1)*dt;
    lambda(i) = LAMBDA(tL(i));
end

% Sistema de Rossler con cambio de variables
x0_R = [.1,.1,log(.1),.1,.1,log(.1),.1,.1,0,0,0,0,0,0,0.1];
[t,x] = ode45(@(t,x)lambdaRossler(t,x),tspan1,x0_R);
x(:,3) = exp(x(:,3));

% Figuras Rossler con cambio de variables
% figure(1)
% set(gcf, 'Position', get(0, 'Screensize'));
% subplot(3,1,1)
% plot(t,x(:,1),'k','linewidth',2);hold on
% plot(t,x(:,4),'r','linewidth',2);
% plot(t,x(:,7),'b','linewidth',2);
% title('$x_1$', 'Interpreter', 'latex', 'fontsize',30)
% xlabel('Tiempo [t]')
% ylabel('Uds')
% legend({'$xi$', 'Z', '$xi_{filtrada}$'}, 'interpreter', 'latex', 'Fontsize',16)
%
% subplot(3,1,2)
% plot(t,x(:,2),'k','linewidth',2);hold on
% plot(t,x(:,5),'r','linewidth',2);
% plot(t,x(:,8),'b','linewidth',2);
% title('$x_2$', 'Interpreter', 'latex', 'fontsize',30)
% xlabel('Tiempo [t]')
% ylabel('Uds')
% legend({'$xi$', 'Z', '$xi_{filtrada}$'}, 'interpreter', 'latex', 'Fontsize',16)
%
% subplot(3,1,3)
% plot(t,x(:,3),'k','linewidth',2);hold on
% plot(t,x(:,6),'r','linewidth',2);
% title({'$Senal - x_3$'}, 'Interpreter', 'latex', 'fontsize',30)
% xlabel('Tiempo [t]')
% ylabel('Uds')

```

```

% legend({'$\xi$', 'Z'}, 'interpreter', 'latex', 'FontSize', 16)
%
% % Retrato Fase
% figure(2)
% set(gcf, 'Position', get(0, 'Screensize'));
% plot3(x(:,1),x(:,2),x(:,3),'k','linewidth',1);hold on
% plot3(x(:,4),x(:,5),x(:,6),'r','linewidth',1)
% grid on
% title('Retrato Fase Sistema X y Z','fontsize',30)
% xlabel({'$x_1$'}, 'Interpreter', 'latex', 'fontsize', 20)
% ylabel({'$x_2$'}, 'Interpreter', 'latex', 'fontsize', 20)
% zlabel({'$x_3$'}, 'Interpreter', 'latex', 'fontsize', 20)

figure(3)
set(gcf, 'Position', get(0, 'Screensize'));
subplot(3,1,1)
plot(t,x(:,9),'k','linewidth',2);hold on
plot(t,x(:,12),'r','linewidth',1)
title({'$\eta_1 \ / \ \hat{\eta}_1$'}, 'Interpreter', 'latex', 'fontsize', 30)
xlabel('Tiempo [t]')
ylabel('Uds')
legend({'$\eta_1$', '$\hat{\eta}_1$'}, 'interpreter', 'latex', 'FontSize', 16)

subplot(3,1,2)
plot(t,x(:,10),'k','linewidth',2);hold on
plot(t,x(:,13),'r','linewidth',1);
title({'$\eta_1 \ / \ \hat{\eta}_1$'}, 'Interpreter', 'latex', 'fontsize', 30)
xlabel('Tiempo [t]')
ylabel('Uds')
legend({'$\eta_1$', '$\hat{\eta}_1$'}, 'interpreter', 'latex', 'FontSize', 16)

subplot(3,1,3)
plot(t,x(:,11),'k','linewidth',2);hold on
plot(t,x(:,14),'r','linewidth',1);
title({'$y \ / \ \hat{y}$'}, 'Interpreter', 'latex', 'fontsize', 30)
xlabel('Tiempo [t]')
ylabel('Uds')
legend({'$y$', '$\hat{y}$'}, 'interpreter', 'latex', 'FontSize', 16)
% set(gca, 'LooseInset', get(gca, 'TightInset'));
% saveas(gcf, 'E3_Estados.png')

% Lambda
figure(4)
set(gcf, 'Position', get(0, 'Screensize'));
plot(tL,lambda,'g','linewidth',2); hold on; grid on
plot(t,x(:,15),'k','linewidth',2)
title('$\lambda \ y \ \hat{\lambda}$', 'interpreter', 'latex', 'fontsize', 30)
xlabel({'Tiempo $t$'}, 'Interpreter', 'latex', 'fontsize', 20)
ylabel({'$\lambda \ / \ \hat{\lambda}$'}, 'Interpreter', 'latex', 'fontsize', 20)

```

```

legend({'$\lambda$', '$\hat{\lambda}$'}, 'interpreter', 'latex', 'fontsize', 16)
% set(gca, 'LooseInset', get(gca, 'TightInset'));
% saveas(gcf, 'E3_Lambda.png')

% % Error de Lambda
% set(gcf, 'Position', get(0, 'Screensize'));
% lambdaI = interp1(t, x(:, 15), tL);
% eL = lambda - lambdaI;
% figure(5)
% set(gcf, 'Position', get(0, 'Screensize'));
% plot(tL, eL, 'g', 'linewidth', 2); grid on
% title('Error de \lambda', 'fontsize', 30)
% xlabel({'Tiempo $t$'}, 'Interpreter', 'latex', 'fontsize', 20)
% ylabel({'$error$'}, 'Interpreter', 'latex', 'fontsize', 20)
% % set(gca, 'LooseInset', get(gca, 'TightInset'));
% % saveas(gcf, 'E3_ErrorLambda.png')

%% Funciones
% Obtener lambda de la salida del sistema
function dx = lambdaRossler(t, x)
%% Sistema
E1 = x(1);
E2 = x(2);
E3 = x(3);
lambda = LAMBDA(t);

% Sistema con cambio de variable
dx(1) = -E2 - exp(E3);
dx(2) = E1 + lambda*E2;
dx(3) = E1 + 2*exp(-E3) - 4;
y = E3;

% Cambio de coordenadas a Z
z1 = x(4);
z2 = x(5);
z3 = x(6);

dx(4) = 4*exp(-y) - 2 + lambda*exp(y);
dx(5) = z1 - z3 - exp(y) + lambda*(2 - 4*exp(-y));
dx(6) = z2 + 4*exp(-y) - 2 + lambda*y;
yz = z3;

% Transformacion Filtrada
k1 = 1;
k2 = 1;
E1f = x(7);
E2f = x(8);

dx(7) = k1*E2f + k1*yz + exp(yz);
dx(8) = E1f + k2*E2f + k2*yz - 4*exp(yz) + 2;

```

```

% Sistema con nuevas coordenadas
eta1 = x(9);
eta2 = x(10);
y_nc = x(11); %

dx(9) = k1*eta2 - k1*k2*y_nc + (k1 + 1)*(4*exp(-y_nc) - 2);
dx(10) = eta1 + k2*eta2 - ( k1 + k2*k2 + 1 )*y_nc + k2*(4*exp(-y_nc) - 2) - exp(y_nc);
dx(11) = eta2 + k2*y_nc + 4*exp(-y_nc) - 2 + lambda*(E2f + y_nc); %

% Observador adaptable
l1 = .01;
l2 = .01;
l3 = .01;
gamma = 1;
eta1_g = x(12);
eta2_g = x(13);
y_ncg = x(14);
teta = x(15);

dx(12) = k1*eta2_g - k1*k2*y_ncg + (k1 + 1)*(4*exp(-y_nc) - 2) + l1*(y_ncg - y_nc);
dx(13) = eta1_g + k2*eta2_g - ( k1 + k2*k2 + 1 )*y_ncg + k2*(4*exp(-y_nc) - 2) - exp(y_n
dx(14) = eta2_g + k2*y_ncg + 4*exp(-y_nc) - 2 + teta*(E2f + y_nc) + l3*(y_ncg - y_nc);

dx(15) = -gamma*(E2f + y_nc)*(y_ncg - y_nc);

dx = dx';
end

function y = LAMBDA(t)
if t < 25
y = 0.3;
else
y = 0.3 + 0.2*sin((pi/25)*t - 25);
end
end

```