



# 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  $\lambda$  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 de Kalman  $K_1(t), K_2(t), K_3(t)$  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  $v$ .

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

$$\begin{aligned}\dot{\hat{x}}_1 &= \sigma(\hat{x}_2 - \hat{x}_1) + k_1(t)e_1 \\ \dot{\hat{x}}_2 &= r\hat{x}_1 - \hat{x}_2\hat{x}_1\hat{x}_3 + k_2(t)e_1 \\ \dot{\hat{x}}_3 &= -b\hat{x}_3 + \hat{x}_1\hat{x}_2 + k_3(t)e_1 \\ e_1 &= y - \hat{x}_1\end{aligned}\tag{2}$$

El filtro de Kalman extendido discreto se plantea como:

$$\begin{aligned}
\hat{x}(k+1) &= \hat{x}(k+1|k) + K(k)(y(k) - H(k)\hat{x}(k+1|k)) \\
\hat{x}(k+1|k) &= f(\hat{x}(k), u(k), 0) \\
K(k) &= P(k+1|k)H(k)^T(H(k)P(k+1|k)H(k)^T + R)^{-1} \\
P(k+1|k) &= AP(k)A^T + Q(k) \\
P(k) &= (I + K(k)H(k))P(k+1|k)
\end{aligned} \tag{3}$$

En donde se observa que  $A = (\partial/\partial x)f(x)$ ,  $H = (\partial/\partial x)h(x)$  con matriz de covarianza  $Q(k)$  y covarianza de la salida  $R$ . La matriz de covarianza  $Q(k)$  tiene forma diagonal debido a que los ruidos son independientes y se calcula de la siguiente forma:

$$Q(k) = \begin{Bmatrix} \text{cov}(\epsilon_1) & 0 & 0 \\ 0 & \text{cov}(\epsilon_2) & 0 \\ 0 & 0 & \text{cov}(\epsilon_3) \end{Bmatrix} \tag{4}$$

$$R(k) = \text{cov}(v)$$

De la misma manera, se puede aplicar el filtro de Kalman de manera discreta por medio de la ecuación diferencial de Ricatti de la siguiente manera:

$$\begin{aligned}
\dot{P} &= A(t)P + PA(t)^T - PH(t)^T R^{-1} H(t)P + Q \\
K(t) &= P(t) \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} R^{-1}
\end{aligned} \tag{5}$$

En donde se debe definir  $P(0) > 0$  para su funcionamiento.

## Resultados

La simulación del ejercicio se realizó por medio de Matlab y se utilizaron los dos métodos del filtro extendido de Kalman mostrados en (3) y (5). En el caso del filtro discreto se observa el siguiente retrato fase:

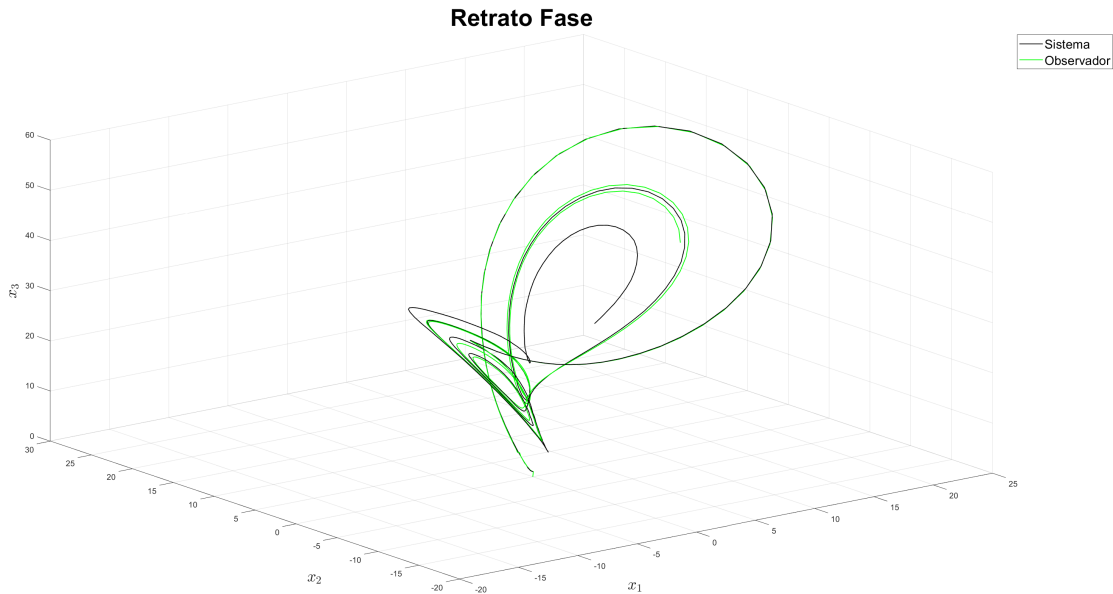


Figura 1: Se observa el sistema de Lorenz y su observador

El sistema de Lorenz se simuló con los siguientes parámetros:  $\sigma = 10$ ,  $r = 28$ ,  $b = 8/3$ . La progresión de los estados en el tiempo se observan en la siguiente figura:

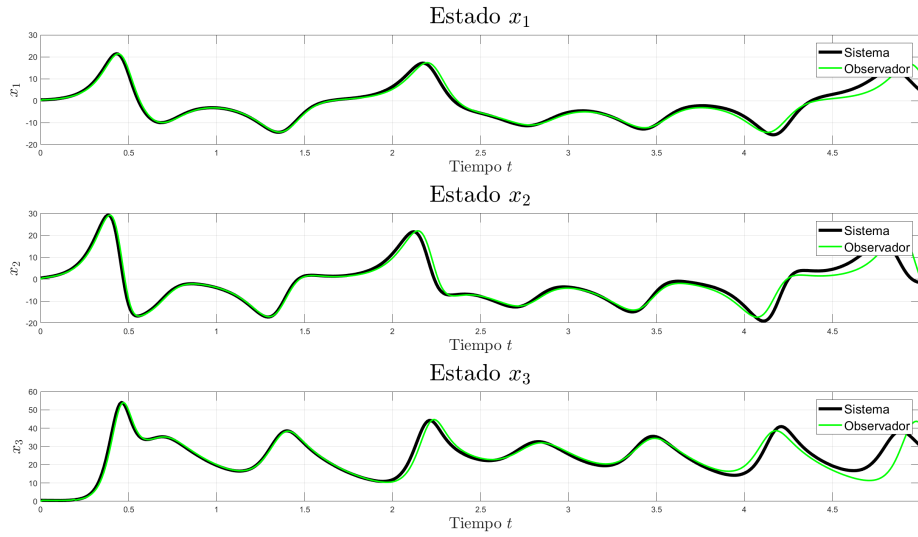


Figura 2: Progresión de los estados en el tiempo

En base a la figura anterior (2), se obtuvieron los errores del sistema original con el observador.

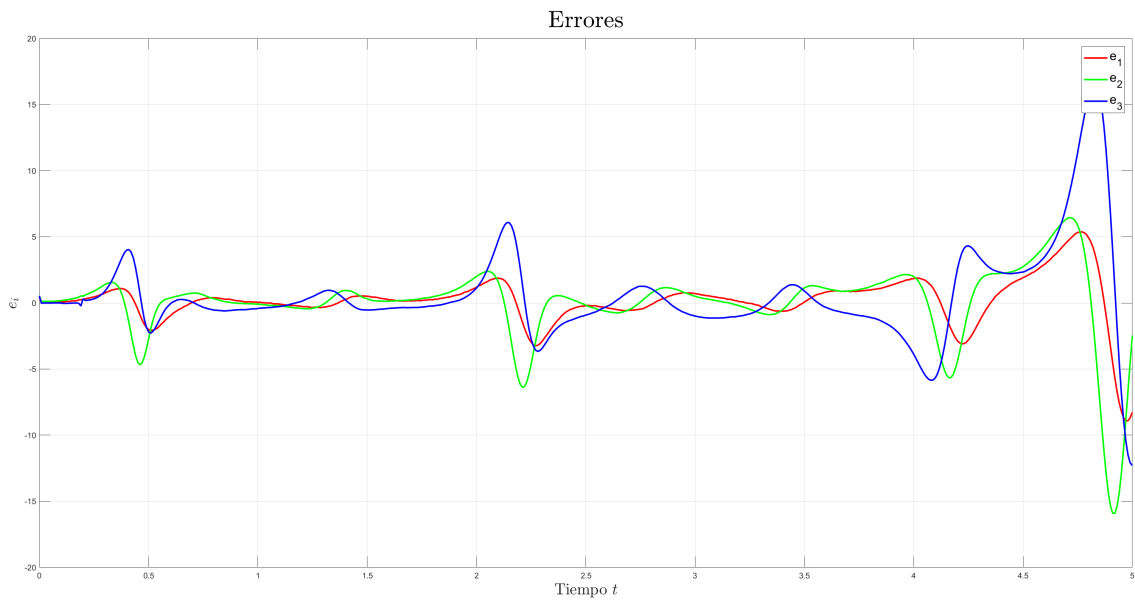


Figura 3: Error del sistema de Lorenz y su observador

Con la simulación de forma continua se obtuvo el siguiente retrato fase:

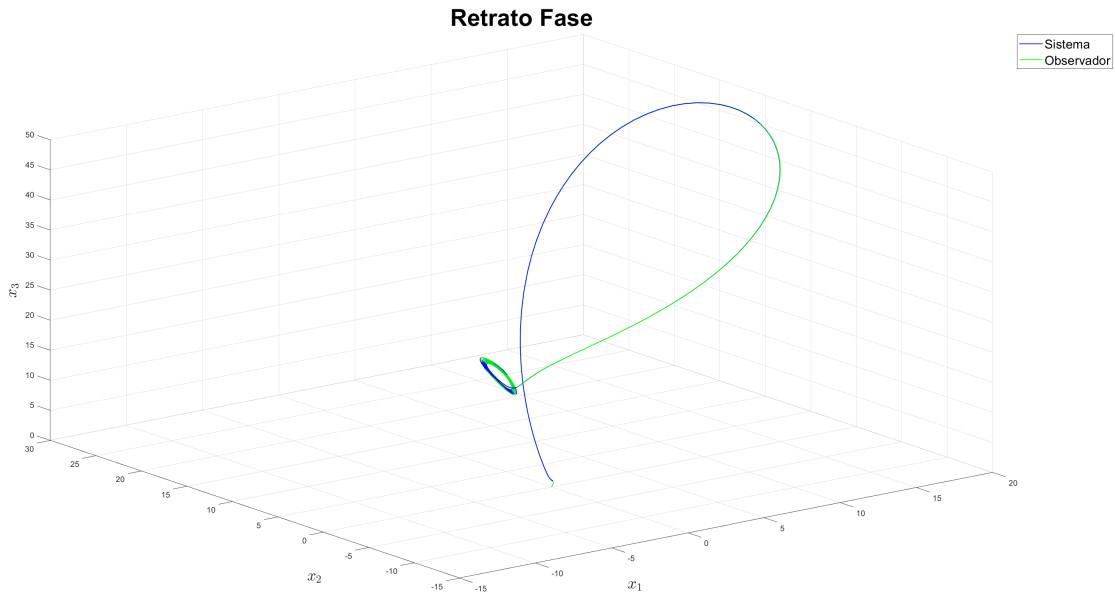


Figura 4: Se observa el sistema de Lorenz y su observador

El sistema fue simulado con los mismos parámetros que la simulación discreta. La progresión de los estados en el tiempo se observan en la siguiente figura:

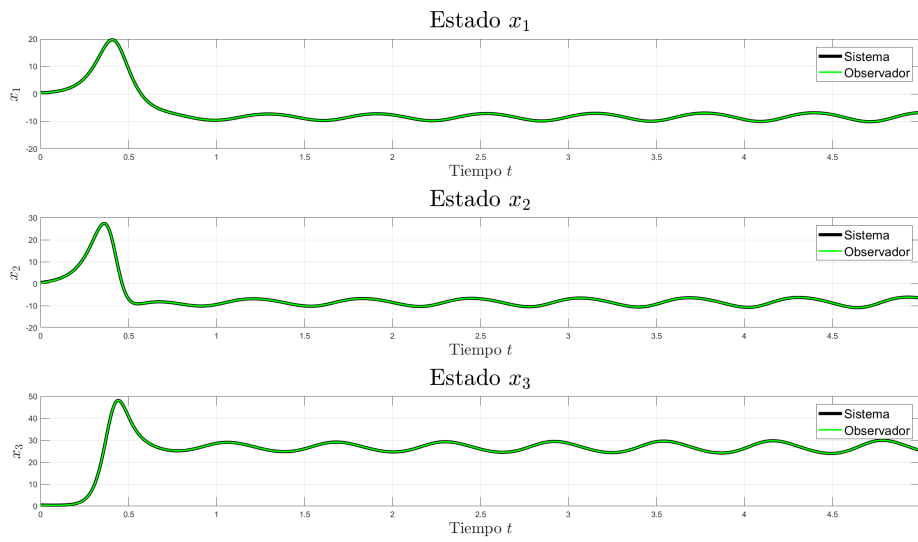


Figura 5: Progresión de los estados en el tiempo

En base a la figura anterior (5), se obtuvieron los errores del sistema original con el observador.

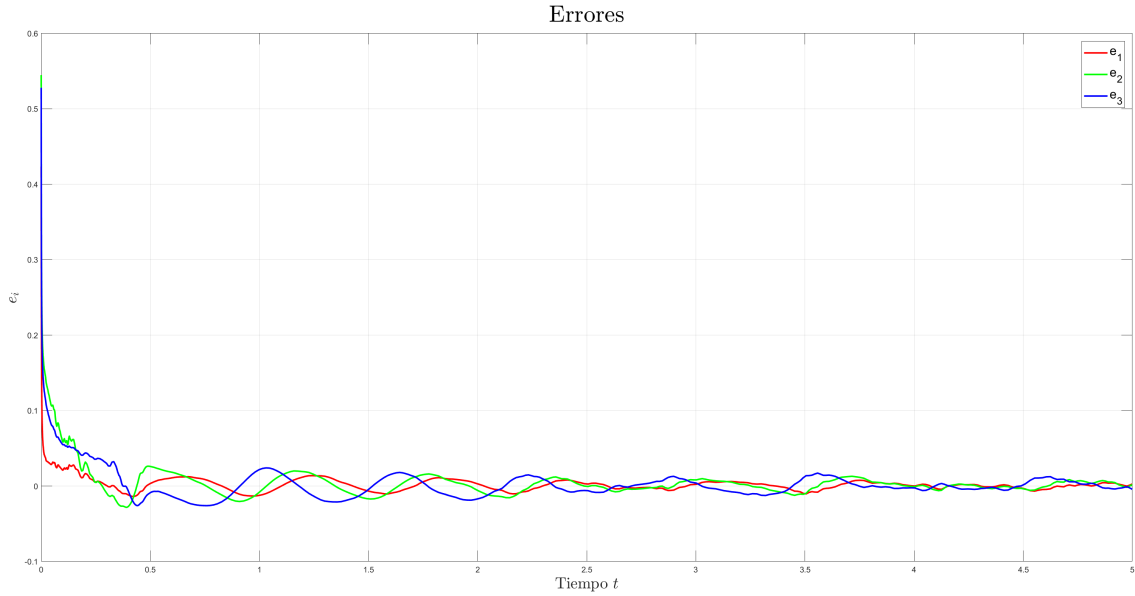


Figura 6: Error del sistema de Lorenz y su observador

## Ejercicio 2

En este ejercicio se utiliza el sistema de Rossler (ecuación 6) al cual se le aplica una señal constante y/o de variación lenta en el tiempo en el valor de  $\lambda$ . Esta señal es enviada de manera encriptada dentro del sistema por medio de la salida  $y$ .

$$\begin{aligned}\dot{x}_1 &= -x_2 - x_3 \\ \dot{x}_2 &= x_1 + \lambda x_2 \\ \dot{x}_3 &= 2 + (x_1 - 4)x_3 \\ y &= x_3\end{aligned}\tag{6}$$

Para recrear la señal  $\lambda$  es necesario que se pueda hacer un observador el cual se pueda sincronizar con el sistema. Sin embargo un observador del sistema en 6 mostrado en 7 no se sincroniza con el sistema. Por lo tanto se realiza el cambio de coordenada siguiente  $\xi_1 = x_1$ ,  $\xi_2 = x_2$ ,  $\xi_3 = \log(x_3)$  y se obtiene el sistema

$$\begin{aligned}\dot{\hat{x}}_1 &= -\hat{x}_2 - \hat{x}_3 \\ \dot{\hat{x}}_2 &= \hat{x}_1 + \lambda \hat{x}_2\end{aligned}\tag{7}$$

$$\begin{aligned}\dot{\xi}_1 &= -\xi_2 - \xi_3 \\ \dot{\xi}_2 &= \xi_1 + \lambda \xi_2 \\ \dot{\xi}_3 &= \xi_1 + 2e^{-\xi_3} - 4 \\ \bar{y} &= \xi_3\end{aligned}\tag{8}$$

Con este cambio de coordenadas el observador puede sincronizarse y por lo tanto se puede utilizar en el método de identificación lineal de parámetros para reconstruir la señal  $\lambda$ . Para realizar este proceso se deben de cumplir las siguientes ecuaciones:

$$\begin{aligned}\dot{w}_i &= Kw_i + Lu_i \\ \hat{y} &= \phi_0(w) + \hat{\lambda}\phi_1(w) \\ \hat{\lambda} &= -v\phi_1(w)P(\hat{y} - \bar{y}) \\ \dot{P} &= -v(\phi_1(w)^2 P^2 - \gamma P)\end{aligned}\tag{9}$$

Y utilizando el sistema de Rossler en las ecuaciones de 9 el cual utiliza un estimador de mínimos cuadrados con factor de olvido se obtienen las siguientes ecuaciones diferenciales a resolver.

$$\begin{aligned}
\dot{w}_{01} &= w_{02} \\
\dot{w}_{02} &= w_{03} \\
\dot{w}_{03} &= -k_0 w_{01} - k_1 w_{02} - k_2 w_{03} + \bar{y} \\
\dot{w}_{11} &= w_{12} \\
\dot{w}_{12} &= w_{13} \\
\dot{w}_{13} &= -k_0 w_{11} - k_1 w_{12} - k_2 w_{13} + u_1 \\
\dot{w}_{21} &= w_{22} \\
\dot{w}_{22} &= w_{23} \\
\dot{w}_{23} &= -k_0 w_{21} - k_1 w_{22} - k_2 w_{23} + u_2 \\
\dot{\hat{\lambda}} &= -v\phi_1(w)P(\hat{y} - \bar{y}) \\
\dot{P} &= -v(\phi_1(w)^2 P^2 - \gamma P)
\end{aligned} \tag{10}$$

En donde se define  $\phi_0(w) = k_0 w_{01} + (k_1 - 1)w_{02} + k_2 w_{03} + w_{12} + w_{21} + w_{23}$ ,  $\phi_1(w) = w_{03} - w_{11} - w_{22}$  y  $u_0 = \xi_3$ ,  $u_1 = -x_3$ ,  $u_2 = (2/x_3) - 4$ . Y el resultado de la integración de  $\hat{\lambda}$  es la aproximación a la señal de  $\lambda$ .

## Resultados

En la figura 7 se observa el retrato fase del sistema. Donde se observa el movimiento caótico del sistema y en la figura 8 se observa la progresión de los estados en el tiempo y hay que notar que la señal enviada del sistema es el tercer estado con la cual se va a aproximar el valor de  $\lambda$ .

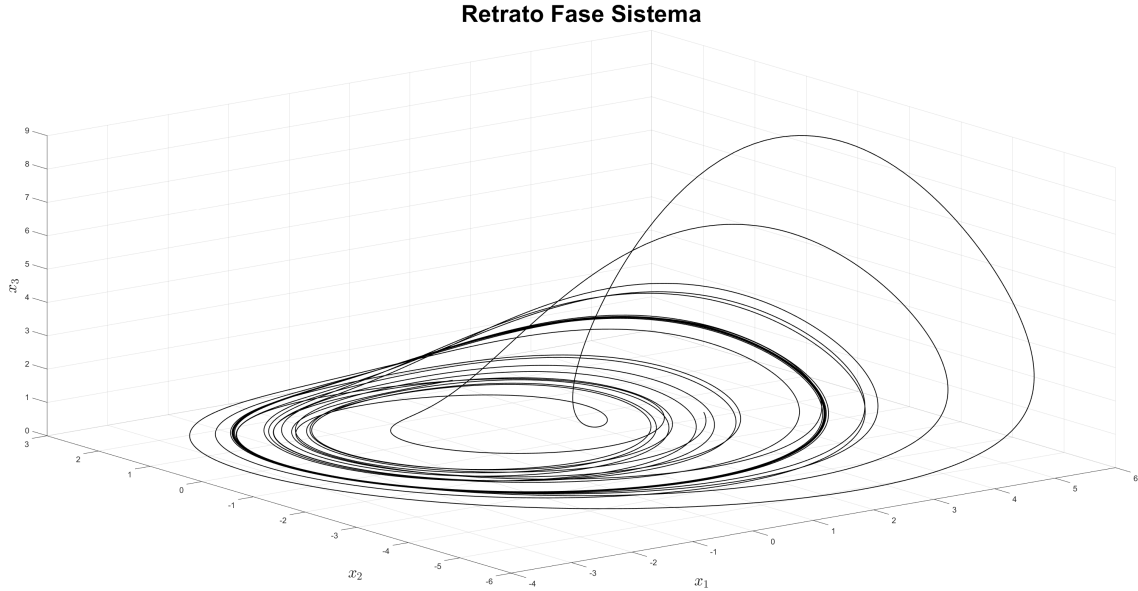


Figura 7: Retrato Fase del sistema de Rossler

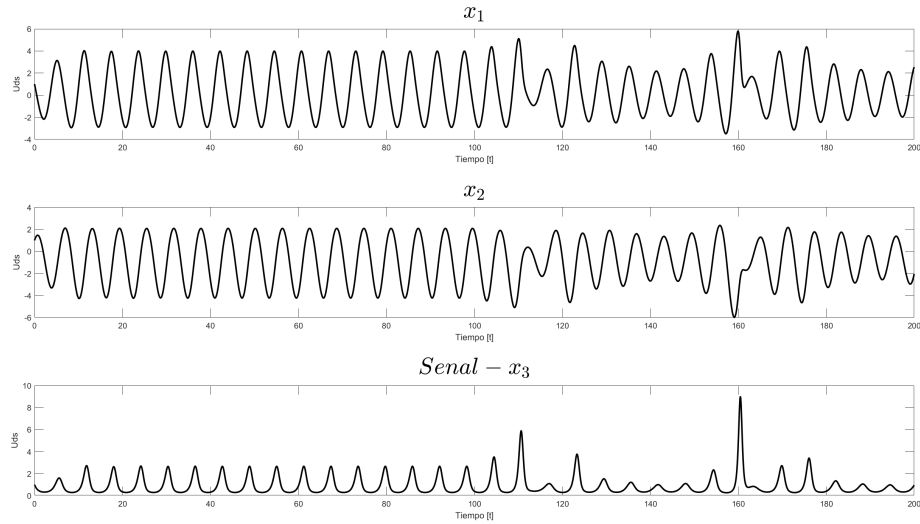


Figura 8: Progresión de los estados en el tiempo

En la figura 9 se observa como la señal recuperada  $\hat{\lambda}$  se acerca a la señal  $\lambda$  original. Cuando la señal es constante la señal se recupera en poco tiempo (aproximadamente 5s) y la señal con cambios lentos mantiene un cierto nivel de error, se puede observar en la figure 10, pero mantiene la forma general de la señal  $\lambda$ .

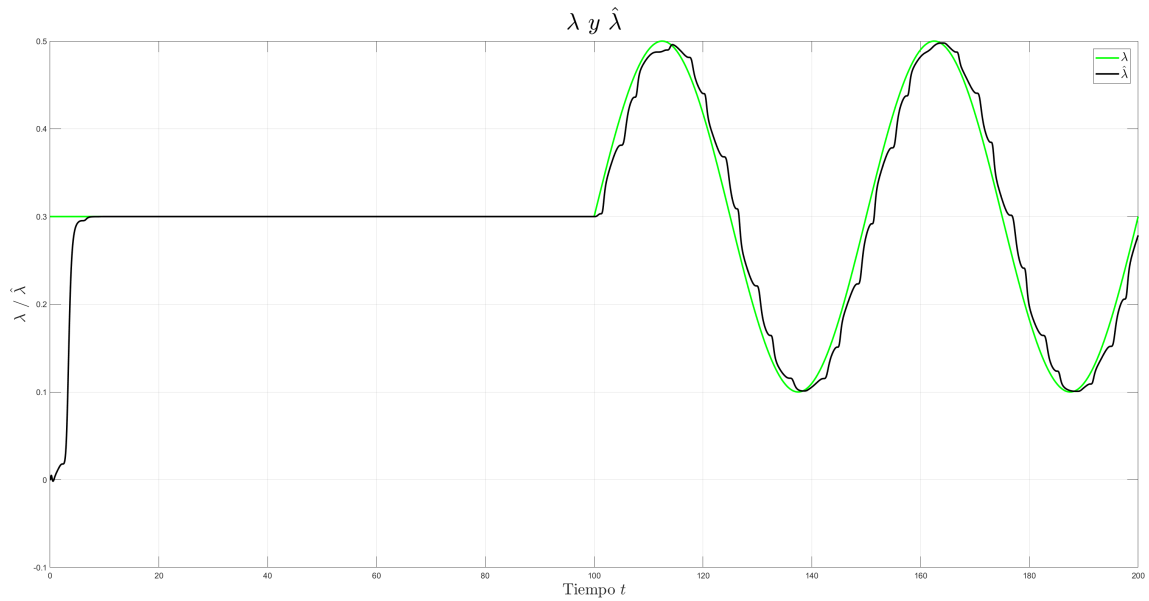


Figura 9:  $\lambda$  y  $\hat{\lambda}$  en el tiempo



Figura 10: Error de  $\lambda$

### Ejercicio 3

En el ejercicio tres se utiliza el mismo sistema de Rossler de la ecuación 6 al cual se le va a aplicar un observador adaptable para la sincronización en comunicación. Este observador funciona solamente si el sistema transmisor cuenta con la siguiente forma:

$$\begin{aligned} \dot{x}_d &= Ax_d + \phi_0(y_d) + B \sum_{i=1}^m \theta_i \phi_i(y_d) \\ y_d &= Cx_d \end{aligned} \quad (11)$$

El observador propuesto para la forma anterior se define como:

$$\begin{aligned} \dot{x} &= Ax + \phi_0(y_d) + B \left[ \sum_{i=1}^m \hat{\theta}_i \phi_i(y_d) + \hat{\theta}_0 G(y_d - y) \right] \\ y &= Cx \\ \dot{\hat{\theta}}_i &= \chi_i(y_d, y), i = 1, 2, \dots, m \end{aligned} \quad (12)$$

Y observando la ecuación anterior, el sistema de Rossler no cumple con las características. Para esto se realiza un tercer cambio de coordenadas a los resultados de la ecuación 8 de la siguiente forma:

$$\begin{aligned} z &= Q(\lambda)\xi \\ q(\lambda) &= \begin{Bmatrix} \lambda & -1 & 1 \\ 1 & 0 & \lambda \\ 0 & 0 & 1 \end{Bmatrix} \end{aligned} \quad (13)$$

Con este cambio de coordenadas al ser derivado respecto al tiempo se obtiene el siguiente sistema el cual ya cumple con la forma de la ecuación 11 y se tiene:

$$\begin{aligned} \dot{z}_1 &= 4e^{-\bar{y}} - 2 + \lambda e^{\bar{y}} \\ \dot{z}_2 &= z_1 - z_3 - e^{\bar{y}} + \lambda(2 - 4e^{-\bar{y}}) \\ \dot{z}_3 &= z_2 + 4e^{-\bar{y}} - 2 + \lambda \bar{y} y_z = [001]z \end{aligned} \quad (14)$$

Para poder encontrar de manera efectiva la señal  $\lambda$  transmitida, se filtran las señales de  $\xi_1$  y



$\xi_2$  en la ecuación 15.

$$\begin{aligned}\dot{\hat{\xi}}_1 &= k_1 \hat{\xi}_2 + k_1 \bar{y} + e^{\bar{y}} \\ \dot{\hat{\xi}}_2 &= \hat{\xi}_1 + k_2 \hat{\xi}_2 + k_2 \bar{y} - 4e^{\bar{y}} + 2\end{aligned}\quad (15)$$

Lo cual nos permite reescribir el sistema en términos de  $\mathbf{z}$  y de las nuevas coordenadas  $\xi$  filtradas obteniendo el siguiente sistema:

$$\begin{aligned}\dot{\eta}_1 &= k_1 \eta_2 - k_1 k_2 y_\eta + (k_1 + 1)(4e^{-y_\eta} - 2) \\ \dot{\eta}_2 &= \eta_1 + k_2 \eta_2 - (k_1 + K_2^2 + 1)y_\eta + k_2(4e^{-y_\eta} - 2) - e^{y_\eta} \\ \dot{y}_\eta &= \eta_2 + k_2 y_\eta + 4e^{-y_\eta} - 2 + \lambda(\hat{\xi}_2 + y_\eta)\end{aligned}\quad (16)$$

El cual tiene la forma de la ecuación 12 y ademas permite definir el observador adaptable de la ecuación 12 como:

$$\begin{aligned}\dot{\hat{\eta}}_1 &= k_1 \hat{\eta}_2 - k_1 k_2 \hat{y}_\eta + (k_1 + 1)(4e^{-\hat{y}_\eta} - 2) + l_1(\hat{y}_\eta - y_\eta) \\ \dot{\hat{\eta}}_2 &= \hat{\eta}_1 + k_2 \hat{\eta}_2 - (k_1 + K_2^2 + 1)\hat{y}_\eta + k_2(4e^{-\hat{y}_\eta} - 2) - e^{\hat{y}_\eta} + l_2(\hat{y}_\eta - y_\eta) \\ \dot{\hat{y}}_\eta &= \hat{\eta}_2 + k_2 \hat{y}_\eta + 4e^{-\hat{y}_\eta} - 2 + \hat{\lambda}(\hat{\xi}_2 + \hat{y}_\eta) + l_3(\hat{y}_\eta - y_\eta) \\ \dot{\hat{\lambda}} &= \hat{\theta} \\ \dot{\hat{\theta}} &= -\gamma(\hat{\xi}_2 + y_\eta)(\hat{y}_\eta - y_\eta)\end{aligned}\quad (17)$$

Con el sistema de la ecuación 16 y su observador 17 es posible recrear la señal  $\lambda$  de inicio.

## Resultados

Observando los estados en el tiempo, el observador sigue al sistema. Sin embargo las ganancias de  $k_1$  y  $k_2$  así como las de  $l_1$ ,  $l_2$  y  $l_3$  influyen en el estado de  $y_\eta$  el cual se puede observar tiene variaciones a lo largo del tiempo. Se observa un rizo alrededor de la señal base.

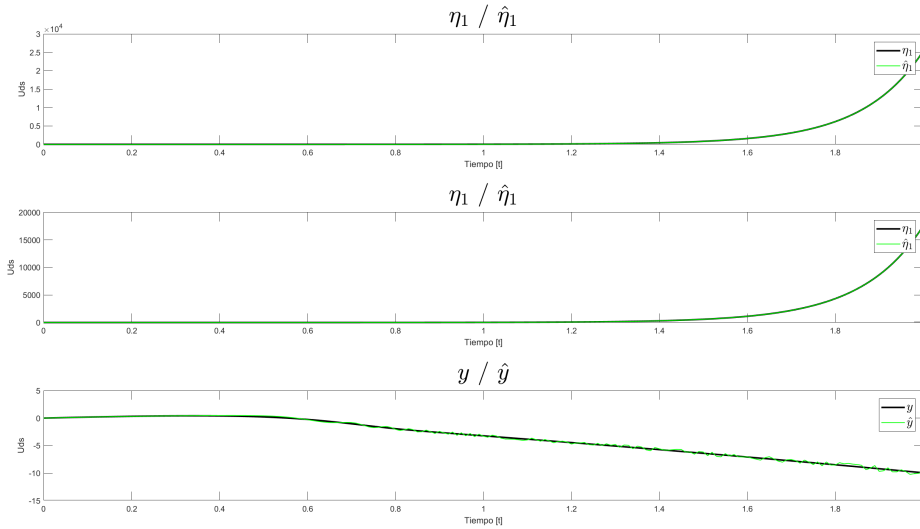


Figura 11: Progresión de los estados en el tiempo

En la reconstrucción de  $\lambda$  se observa que debido al ruido del observador en el estado de  $y_\eta$  genera un ruido alrededor de la señal de referencia.

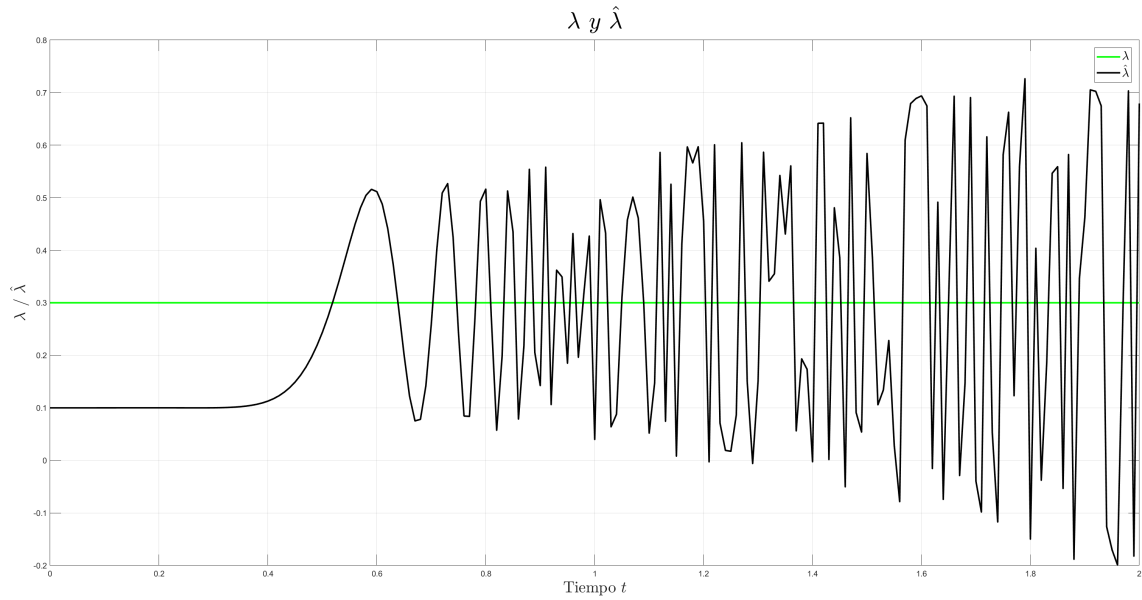


Figura 12:  $\lambda$  y  $\hat{\lambda}$  en el tiempo

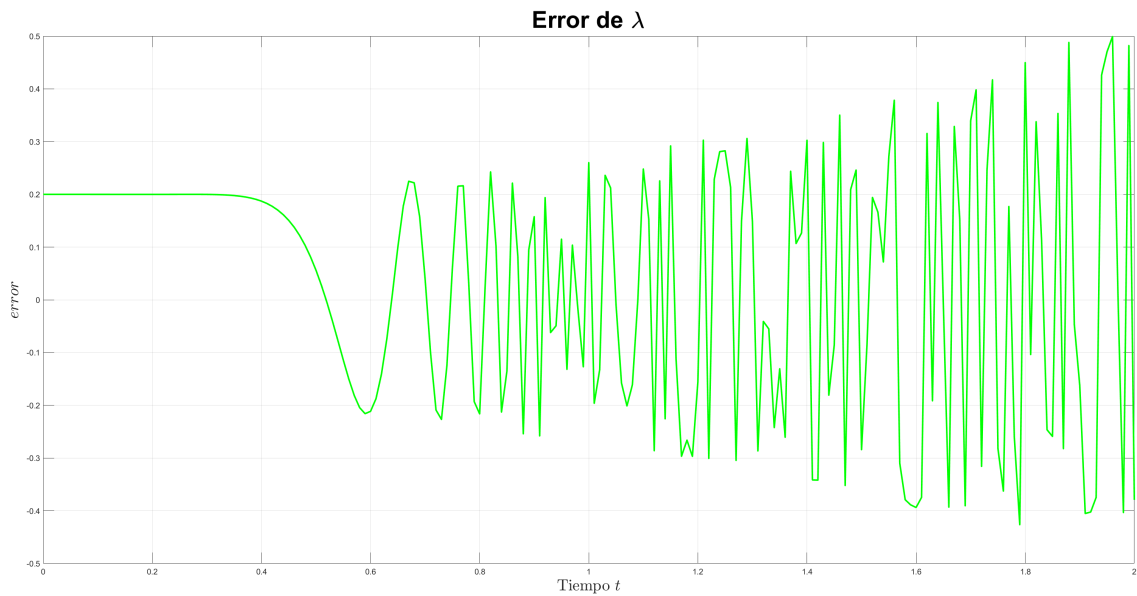


Figura 13: Error de  $\lambda$

## Conclusión

En el primer ejercicio del sistema de Lorenz, se nota de manera inicial que al comparar el sistema discreto y continuo existe una diferencia del comportamiento en el mismo tiempo con las mismas condiciones iniciales. Al comprar las figuras 1 con 4 y 2 con 5 se observa que el sistema tiene comportamiento diferente, debido a la naturaleza caótica del sistema.

En el caso del filtro discreto, debido a que el integrador es un integrador de paso simple de Euler existen errores en cada paso de integración lo cual hace que el sistema avance de manera muy rápida y el calculo de las ganancias de Kalman  $K(k)$  puede no ser suficiente para que converjan en todo momento. Esto sucede mas que nada en los momentos donde existen cambios grandes dentro de los estados. Se puede observar en la figura 2 cuando esta por el tiempo de 2s

y 4.5s que es donde el error mostrado en la figura 3 tambien sufre picos.

En el caso continuo, el movimiento del sistema es mas suave y por lo tanto se observan cambio menos abruptos en el tiempo. El integrador en este caso es un Runge-Kutta de 4<sup>to</sup> orden. Y el calculo de las ganancias de Kalman  $K(t)$  por medio de la ecuación de Ricatti permite que el observador converja con el sistema volviendo los errores del sistema muy pequeños tendiendo a cero con el paso del tiempo, esto se observa en la figura 6.

En el segundo ejercicio se utilizo el sistema de Rossler para enviar una señal  $\lambda$  por medio de la salida del sistema. Utilizando un identificador lineal de parámetros con un estimador de mínimos cuadrados con factor de olvido se recrea la señal enviada. Esta señal se puede observar en la figura 9 en color verde y en color negro la señal recuperada  $\hat{\lambda}$ .

En el tercer ejercicio al hacer los diferentes cambios de coordenadas se logra dar forma al sistema Rossler a una Forma de tipo Lure y después a una forma que permita tener un observador adaptable por medio de un parámetro  $\theta$  en donde se logra la sincronización variando este valor. Si la forma del sistema como en la ecuación 16 permite separar el valor adaptable, el sistema puede recrearla y recuperar un mensaje del sistema original.

## Referencias

- [1] Alexander Fradkov and Henk Nijmeijer. Adaptive observer-based synchronization for communication. *International Journal of Bifurcation and Chaos*, 10:2807–2813, 12 2000.
- [2] H. Huijberts, H. Nijmeijer, and R. Willems. System identification in communication with chaotic systems. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 47(6):800–808, 2000.
- [3] Henk Nijmeijer. A dynamical control view on synchronization. *Physica D: Nonlinear Phenomena*, 154:219–228, 06 2001.

# 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);
xg = zeros(3,n);
y = zeros(1,n);
xg_k = zeros(3,n);
K = zeros(3,n);
P = zeros(3,3,n);
Q = zeros(3,3,n);
R = zeros(1,n);
eps = zeros(3,n);
v = zeros(1,n);
e1 = zeros(1,n);

% Ruido del proceso y lectura
mu = 0;
sig = 0.01;
% 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,.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)

```

```

xlabel({'$x_3$'}, '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),0,0,0,0,0,0,0,0,0,0,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_nc) + l2*(y_ncg - y_nc);
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

```