%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% %

% PARCIAL : II - 50% %

% NOMBRE : RANGEL ALVARADO %

% CÉDULA : 08-734-1444 %

% FECHA : 08.JUL.12 %

% MATERIA : TÓPICOS ESPECIALES DE ROBÓTICA MÓVIL %

% PROFESOR: DR. RONY CABALLERO %

% %

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% %

% BREVE LEYENDA %

% %

% PHI -> matriz de transición %

% GAMMA -> matriz de entrada %

% LAMBDA -> matriz de perturbación %

% Ck | H -> matriz de salida %

% Dp -> matriz de prealimentación %

% u -> vector de entrada %

% L | d -> vector de perturbación %

% w -> ruido blanco de proceso %

% v | n -> ruido blanco de medición %

% I -> matriz identidad %

% T -> Período de muestreo

% P -> covarianza

% y -> vector de salida sin perturbación

% z -> vector de salida con perturbación

% ^

% x -> predicción del siguiente estado

% %

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% %

% MODELO DISCRETO: %

% x(k + 1) = PHI(k)\*x(k) + GAMMA(k)\*u(k) + LAMBDA\*w(k) %

% z(k) = y(k) + n(k) %

% = C(k)\*x(k) + Dp(k)u(k) + n(k) %

% %

% MODELO EN VARIABLES DE ESTADO: %

% . %

% x = A\*x + B\*u + delta %

% y = C\*x + D\*u %

% %

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% %

% ANTES DE INICIAR EL ALGORITMO %

% PHI = e^aT %

% = I + A\*T + A^2\*T^2/2! + A^3\*T^3/3! + ... + A^n\*T^n/n! %

% GAMMA = integral(e^at, t, 0, T) %

% = I\*T + A\*(T^2/2) + (A^2/2!)\*(T^3/3) + ... %

% + (A^(n-1)/(n-1)!)\*(T^n/n) %

% LAMBDA = L, es un vector [m, 1] %

% %

% ALGORITMO DE KALMAN: %

% 1 - Establecer condiciones iniciales para xh, Ph %

% xh es de dimensión [1 m] %

% Ph es de dimensión [m m] %

% ^ %

% xh = [ 0, 0] (por ejemplo) %

% Ph = [1 0; 0 1] (por ejemplo) %

% 2 - Calcular la ganancia de Kalman %

% K[k] = P[k/k-1]\*[H\*P[k/k-1]\*H' + R]^-1 %

% 3 - Actualizar estimación del vector de estado con las mediciones z(k) %

% ^ ^ %

% x[k/k-1] = PHI\*x[k-1] + GAMMA\*u[k-1] %

% 4 - Calcular la covarianza del vector estado actualizado %

% P[k] = (I - K(k)\*H)\*P[k/k-1] %

% 5 - Predecir el vector de estado futuro y su covarianza %

% x[k+1/k] = PHI\*x[k] + GAMMA\*u[k-1] %

% P[k/k+1] = PHI\*P[k-1]\*PHI' + LAMBDA\*Q\*LAMBDA' %

% %

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% %

% DATOS DEL PARCIAL... %

% %

% Aplique un filtro de Kalman al siguiente sistema %

% %

% . %

% x = [-2, 1; -1, -2]\*x + [0; 1]\*u + [1; -1]\*v %

% %

% y = [1, 0]\*x + w %

% %

% v ~ N(0, 0.09) -> Q = 0.09 %

% w ~ N(0, 0.05) -> R = 0.05 %

% %

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Datos ingresados por el estudiante para ejecutar el filtro

T = 0.05; % Período de muestreo

A = [-2, 1; -1, -2]; % Matriz de sistema

B = [0; 1]; % Matriz de entrada

L = [1; -1]; % Matriz de error

H = [1, 0]; % Matriz de salida

I = eye(2,2); % Matriz identidad

% Ruido en el sistema

Q = 0.090; % ruido de proceso

R = 0.05; % ruido de medición

% Condiciones Iniciales del Sistema (para simular)

t = 0:T:10; % vector de tiempo

u = cos(t); % vector de enrada

x0 = [0; 0]; % estado inicial

x = x0; % estado inicial

y = H\*x0; % estado inicial

% Conjetura de condiciones iniciales para el filtro de Kalman

xh = [0.5;-0.5]; % xh(0)

Ph = eye(2,2); % P0

% Simulacion

% Cálculo de variables de tiempo discreto

PHI = (I + A\*T + ... % Matriz de transición

A^2\*T^2/factorial(2) + ...

A^3\*T^3/factorial(3) + A^4\*T^4/factorial(4) + ...

A^5\*T^5/factorial(5) + A^6\*T^6/factorial(6));

N = (I\*T + ... % Integral de la serie de Taylor

A\*(T^2/2) + (A^2/factorial(2))\*(T^3/3) + ...

(A^3/factorial(3))\*(T^4/4) + (A^4/factorial(4))\*(T^5/5) + ...

(A^5/factorial(5))\*(T^6/6) + (A^6/factorial(6))\*(T^7/7));

GAMMA = N\*B; % Matriz de Entrada

LAMBDA = N\*L; % Matriz de Ruido

% Matrices de almacenamiento de estimados y covarianzas

% Llenamos las matrices de ceros de dimension 1 x longitud\_de\_vector\_tiempo

% Predictivos

xh1 = zeros(1, length(t));

xh2 = zeros(1, length(t));

% Covarianza

Ph1 = zeros(1, length(t));

Ph2 = zeros(1, length(t));

% Estados Iniciales guardados

xh1(1) = xh(1);

xh2(2) = xh(2);

Ph1(1) = Ph(1);

Ph2(2) = Ph(2);

for k = 1:length(t) - 1

v = sqrt(Q)\*randn; % ruido blanco de proceso

w = sqrt(R)\*randn; % ruido blanco de medición

% sistema

x(:,k + 1) = PHI\*x(:,k) + GAMMA\*u(k) + LAMBDA\*v; % ecuación discreta

y(k + 1) = H\*x(:,k + 1) + w; % salida

% Actualización de la medición en tiempo discreto

Kgain = Ph\*H'\*(H\*Ph\*H' + R)^-1; % ganancia de kalman

xh = xh + Kgain\*(y(k) - H\*xh); % predicción de la medición

Ph = (I - Kgain\*H)\*Ph; % actualización de error covarianza

% Actualización en tiempo discreto

xh = PHI\*xh + GAMMA\*u(k); % proyectar el estado siguiente

Ph = PHI\*Ph\*PHI' + LAMBDA\*Q\*LAMBDA'; % proyectar error de covarianza

% Almacenamiento para gráfico posterior

xh1(k + 1) = xh(1); % señal filtrada Xh1

xh2(k + 1) = xh(2); % señal filtrada Xh2

Ph1(k + 1) = Ph(1); % señal de covarianza Ph1 en Xh1

Ph2(k + 1) = Ph(2); % señal de covarianza Ph2 en Xh2

end

% Parte I.

% Entonces grafique y compare la salida filtrada (y = X1) proveniente del

% filtro Kalman con la real.

% Tratamos estos datos como figura - Graficar Salida filtrada

figure('name', 'RANGEL ALVARADO - 8-734-1444 - Y vs X');

set(0,'DefaultAxesColorOrder', ... % Ajustamos colores Red, Cyan, Blue

[1 0 0; 0 1 1;0 0 1 ]);

plot(t, y(1,:), t, xh1, t, x(1,:)); % y(t) vs xh(1,t), x(1, t)

xlabel('Time (secs)'); % Etiqueta eje X

ylabel('Xn, Yn'); % Etiqueta eje Y

title('Y(t) vs [Xh(t), X(t)]'); % Título de la figura 1

legend('y(t) - salida con error', ...% Leyenda de la figura 1

'xh(1,t) - pronóstico variable 1',...

'x (1,t) - variable real');

% Parte II.

% Grafique los elementos de la matriz de covarianza de estimación en

% función del tiempo.

% Tratamos estos datos como figura - Graficar Covarianza

figure('name', 'RANGEL ALVARADO - 8-734-1444 - P vs t');

set(0,'DefaultAxesColorOrder', ...

[1 0 0; 0 0 1 ]); % Ajustamos colores Red, Cyan, Blue

plot(t, Ph1, t, Ph2); % Ph(1,t) vs Ph(2,t)

xlabel('Time (secs)'); % Etiqueta eje X

ylabel('Salida de Covarianza'); % Etiqueta eje Y

title('P (Covarianza) vs t'); % Título del gráfico 2

legend('Ph(1,t) - elem. 1 de matriz covarianza',...% Leyenda de la figura 1

'Ph(2,t) - elem. 2 de matriz covarianza');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



