Facultad de Ciencias de la Electrónica

Benemérita Universidad Autónoma de Puebla

Licenciatura en Ingeniería Mecatrónica

Dr. Fernando Reyes Cortés

Robótica

ftp://ece.buap.mx/pub/profesor/FernandoReyes/Robotica/

Primavera 2020

Parte VIII

Filtros pasa bajas y estimación numérica

Contenido

Robótica FCE

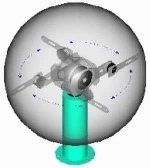
BUAP

Filtro pasa bajas

Función de transferencia de un filtro pasa bajas:

$$\frac{F_u}{u} = \frac{\lambda}{s+\lambda} \Longrightarrow F_u = \frac{\lambda}{s+\lambda}u$$

$$\dot{F}_u(t) = -\lambda F_u(t) + \lambda u(t)$$



Filtro pasa bajas

- $\lambda \in \mathbb{R}_+$ es la frecuencia de corte.
- $u(t) \in \mathbb{R}$ es la entrada.
- $F_u(t) \in \mathbb{R}$ significa la señal filtrada de $u(t)$.
- $\dot{F}_u(t) \in \mathbb{R}$ representa la estimación de la derivada de la entrada $u(t)$.
- $\dot{F}_u(t) \simeq \frac{d}{dt}u(t)$.
- Desventaja: el filtro pasa bajas $\frac{\lambda}{s+\lambda}$ distorsiona la fase, significa que la estimación de la derivada $\dot{F}_u(t)$ tendrá un corrimiento en el tiempo.
- Para frecuencias w dentro del ancho de banda $w \in [0,\ \lambda]$, la magnitud de la función de transferencia $\frac{\lambda}{s+\lambda}$ corresponde a 0 dB (recuerde que $s = jw$).
- Para frecuencias en DC $w = 0$, la magnitud de $\frac{\lambda}{s+\lambda}$ es unitaria, donde $s = jw$.

Robótica FCE

BUAP

Dada una función de transferencia $F_u(s)$:

$$F_u(s) = \frac{\lambda}{s+\lambda}u$$

la correspondiente ecuación diferencial ordinaria de primer orden (ODE) está dada por:

$$\dot{F}_u = -\lambda F_u + \lambda u$$

La solución $F_u(t)$ de la ODE adquiere la siguiente forma:

$$F_u(t) = e^{-\lambda\left(t - t_0\right)}F_u(t_0) + \int_{t_0}^{t} \lambda e^{-\lambda\left(t-\sigma\right)}u(\sigma)d\sigma$$

Filtro pasa bajas discreto

La forma recursiva del filtro pasa bajas

$$\dot{F}_u(t) = -\lambda F_u(t) + \lambda u(t)$$

es la siguiente:

$$F_u(t_k) = e^{-h\lambda} F_u(t_{k-1}) + (1 - e^{-h\lambda}) u(t_{k-1})$$

Por lo tanto, la estimación de la derivada de $u(t)$ es representada por $\dot{u}(t) \simeq \dot{F}_u(t_k)$, y se encuentra dada por:

$$\dot{F}_u(t_k) = -\lambda F_u(t_k) + \lambda u(t_k)$$

Algoritmo de estimación de la derivada de $u(t)$ por filtrado

Notación: $\dot{u}(t) \simeq \dot{F}_u(t_k)$, además $t_k = kh$, es el tiempo discreto; donde h es el periodo de muestreo, y $k = 1, 2, 3, 4, \cdots, N$:

$$\begin{aligned} F_u(t_k) &= e^{-h\lambda} F_u(t_{k-1}) + (1 - e^{-h\lambda}) u(t_{k-1}), \quad F_u(t_k) \text{ es la señal filtrada de } u(t_k). \\ \dot{F}_u(t_k) &= -\lambda F_u(t_k) + \lambda u(t_k), \quad \dot{F}_u(t_k) \text{ es la derivada de la señal de entrada } u(t_k). \end{aligned}$$

Reducción del orden de sistemas dinámicos por medio de filtrado $\frac{\lambda}{s+\lambda}$:

$$\ddot{y}(t) + \alpha_1 \dot{y}(t) + \alpha_0 y(t) = \beta_0 u(t)$$

$$\frac{\lambda}{\lambda+s}\ddot{y}(t) + \frac{\lambda}{\lambda+s}\alpha_1 \dot{y}(t) + \frac{\lambda}{\lambda+s}\alpha_0 y(t) = \frac{\lambda}{\lambda+s}\beta_0 u(t)$$

$$s\frac{\lambda}{\lambda+s}\dot{y}(t) + s\frac{\lambda}{\lambda+s}\alpha_1 y(t) + \frac{\lambda}{\lambda+s}\alpha_0 y(t) = \frac{\lambda}{\lambda+s}\beta_0 u(t) \quad \text{donde } s = \frac{d}{dt}.$$

$$\dot{F}_{\dot{y}(t)}(t) + \alpha_1 \dot{F}_{y(t)}(t) + \alpha_0 F_{y(t)}(t) = \beta_0 F_{u(t)}(t)$$

$$\dot{F}_{\dot{F}_{y(t)}}(t) + \alpha_1 \dot{F}_{y(t)}(t) + \alpha_0 F_{y(t)}(t) = \beta_0 F_{u(t)}(t)$$

Estructura de los filtros: $\dot{F}_{y(t)}(t)$, $\dot{F}_{\dot{y}(t)}(t)$ y $\dot{F}_{u(t)}(t)$.

$$\dot{F}_{y(t)}(t) = -\lambda F_{y(t)}(t) + \lambda y(t), \qquad \dot{F}_{y(t)}(t) \simeq \dot{y}(t)$$

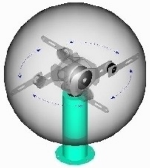
$$\dot{F}_{\dot{y}(t)}(t) = -\lambda F_{\dot{y}(t)}(t) + \lambda \underbrace{\dot{y}(t)}_{\dot{F}_{y(t)}(t)}, \qquad \dot{F}_{\dot{y}(t)}(t) = \dot{F}_{\dot{F}_{y(t)}}(t) \simeq \ddot{y}(t)$$

$$\dot{F}_{u(t)}(t) = -\lambda F_{u(t)}(t) + \lambda u(t)$$

Ventajas de los sistemas dinámicos filtrados

Ventajas de los sistemas dinámicos filtrados

- Cualquier ecuación diferencial (lineal y no lineal) de orden mayor o igual a 2, puede ser reducida a una ecuación diferencial ordinaria de primer orden (ODE).
- Usando filtrado consecutivo (concatenado) a partir de la variable física $y(t)$, se puede estimar (construir) las derivadas de orden alto.
- La ecuación dinámica resultante (ODE) se denomina modelo dinámico filtrado, quedando expresado en variables de estado de los filtros.
- Una característica importante del modelo dinámico filtrado, es que mantiene los mismos parámetros que la ecuación diferencial original.



Robótica FCE
BUAP

Código MATLAB 8: simufiltroA

Robótica, Período de primavera (24 de enero de 2020). Licenciatura en Ingeniería Mecatrónica.
Dr. Fernando Reyes Cortés. Facultad de Ciencias de la Electrónica, BUAP.

simufiltroA.m — MATLAB 2019a

```matlab
clc;
clear all;
close all;
format short
ti=0; h=0.001;  tf = 10;  %tiempo de simulación (segundos).
t=(ti:h:tf)';  %vector columna tiempo
 %Señal con ruido: u(t_k) = 5 sen(2πt_k) + random('Normal' ,t_k, t_k)
u=5*sin(2*pi*t)+random('Normal' ,t,t); %señal con ruido.
[renglones, columnas]=size(t);
Fu=zeros(renglones, columnas); %estado del filtro F_u
lambda=3; %frecuencia de corte λ.
for k=2:renglones
    % F_u(t_k) = e^(-λh) F_u(t_{k-1}) + (1 - e^(-λh)) u(t_{k-1}).
    Fu(k)=exp(-h*lambda)*Fu(k-1)+(1-exp(-h*lambda))*u(k-1);
end
subplot(2,1,1); plot(t,u)  % señal con ruido.
subplot(2,1,2); plot(t,Fu) %señal filtrada.
```

Robótica FCE
BUAP

Señal filtrada $F_u = \frac{\lambda}{\lambda+s}u$

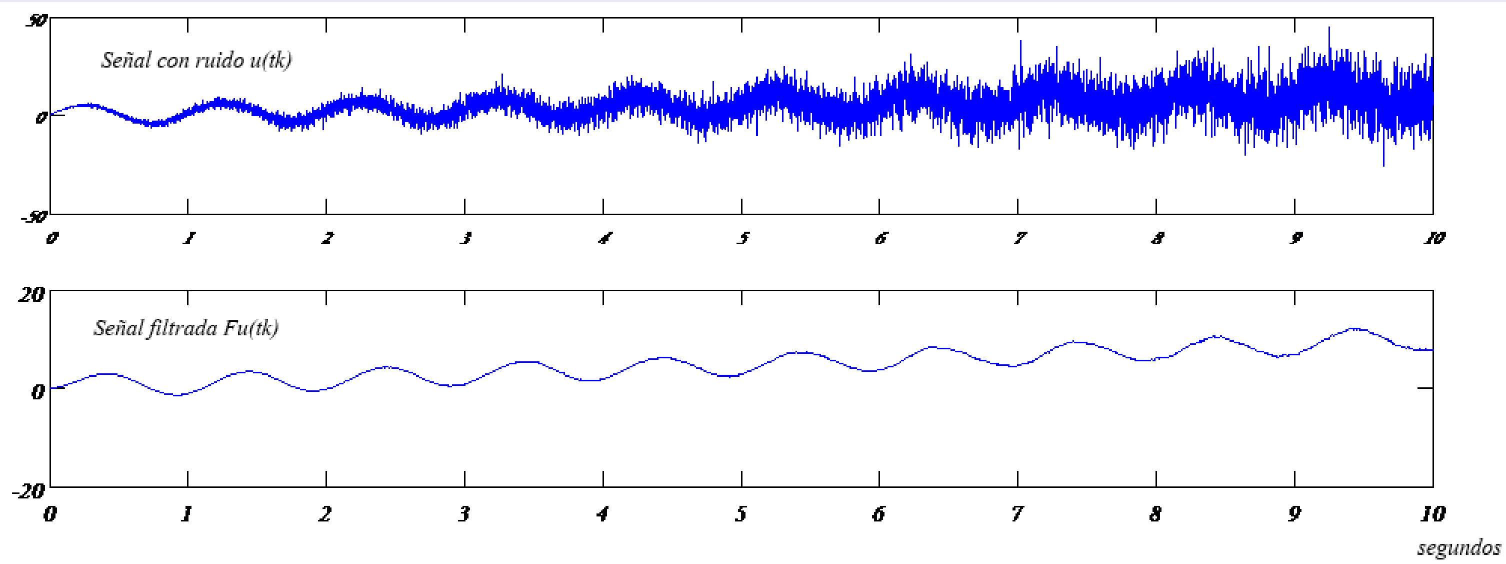


Figura 1: Señal con ruido $u(t_k)$ y señal filtrada $F_u(t_k)$.

Robótica FCE
BUAP

Bode de la función de transferencia $G(s) = \frac{\lambda}{s+\lambda}$

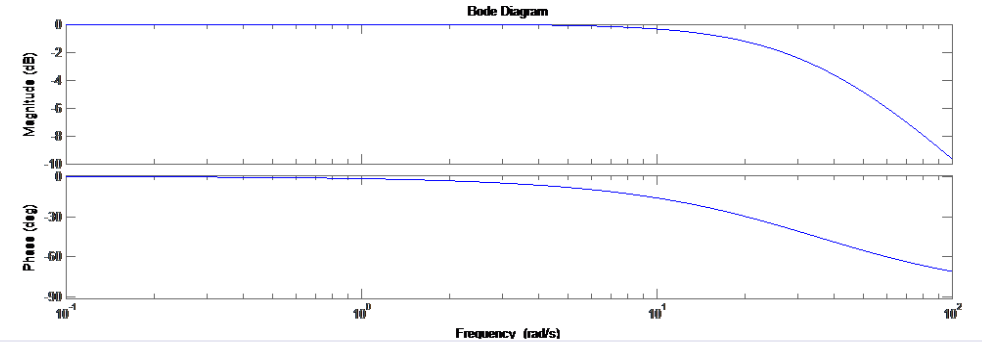


Figura 2: Magnitud y fase de la función de transferencia $G(s) = \frac{\lambda}{s+\lambda}$.

Características importantes del filtro: $\dot{F}_u(t) = -\lambda F_u(t) + \lambda u(t)$

- La figura 2 muestra la magnitud (en decibeles) y fase (grados) de la función de transferencia $G(s) = \frac{\lambda}{s+\lambda}$.
- Observe que la fase presenta una distorsión no lineal, lo que significa que el estado $F_u(t)$ exhibirá un corrimiento en el tiempo, con respecto a la señal de entrada $u(t)$.
- El corrimiento (desfase) representa un inconveniente o desventaja del método de filtrado para obtener la estimación de la derivada de la entrada $u(t)$.

Robótica FCE
BUAP

Obtención de la gráfica de Bode de $G(s) = \frac{\lambda}{s+\lambda}$

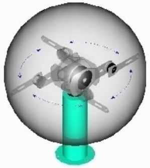
Para obtener el Bode de la función de transferencia $G(s) = \frac{\lambda}{s+\lambda}$ (figura 2) se puede ingresar el siguiente código **MATLAB** directamente en la ventana de comandos:

- *fx* >> `w=(0.1:0.01:100)';` ↩
- *fx* >> `num=[35];` ↩
- *fx* >> `den=[1 35];` ↩
- *fx* >> `G=tf(num,den)` ↩
- *fx* >> `bode(G,w)` ↩

Filtro pasa bajas

El método tradicional para obtener la derivada de una señal $u(t)$ es por diferenciación numérica (método de Euler): por ejemplo, $\dot{u}(t) \simeq \frac{u(t_k)-u(t_{k-1})}{t_k-t_{k-1}}$.

Otra forma de estimar la derivada de una señal $u(t)$ es por medio de filtros, por ejemplo utilizando la siguiente función de transferencia:



$$\frac{F_u}{u} = \frac{\lambda}{s+\lambda} \Longrightarrow F_u = \frac{\lambda}{s+\lambda}u$$

$$\dot{F}_u(t) = -\lambda F_u(t) + \lambda u(t)$$

Filtro pasa bajas

- $\lambda \in \mathbb{R}_+$ es la frecuencia de corte.
- El ancho de banda del filtro representa las frecuencias contenidas entre 0 y λ.
- Para sistemas causales, se considera $t \geq 0$.
- $u(t) \in \mathbb{R}$ es la entrada.
- $F_u(t) \in \mathbb{R}$ es el estado del filtro y representa la señal filtrada de $u(t)$.
- $\dot{F}_u(t) \in \mathbb{R}$ representa la estimación de la derivada de la señal filtrada de $u(t)$, es decir: $\dot{F}_u(t) \simeq \dot{u}(t) = \frac{d}{dt}u(t)$.

La forma recursiva del filtro pasa bajas

$$\dot{F}_u(t) = -\lambda F_u(t) + \lambda u(t)$$

es la siguiente:

$$F_u(t_k) = e^{-h\lambda} F_u(t_{k-1}) + (1 - e^{-h\lambda}) u(t_{k-1})$$

Por lo tanto, la estimación de la derivada de $u(t)$ es representada por $\dot{u}(t) \simeq F_p(t_k)$, y se encuentra dada por:

$$F_p(t_k) = -\lambda F_u(t_k) + \lambda u(t_k)$$

Algoritmo de estimación de la derivada de $u(t)$ por filtrado

Notación: $\dot{u}(t) \simeq F_p(t_k)$, además $t_k = kh$, es el tiempo discreto; donde h es el periodo de muestreo, y $k = 1, 2, 3, 4, \cdots, N$:

$$\begin{aligned} F_u(t_k) &= e^{-h\lambda} F_u(t_{k-1}) + (1 - e^{-h\lambda}) u(t_{k-1}), \quad F_u(t_k) \text{ es la señal filtrada de } u(t_k). \\ F_p(t_k) &= -\lambda F_u(t_k) + \lambda u(t_k), \quad F_p(t_k) \text{ es la derivada de la señal de entrada } u(t_k). \end{aligned}$$

Robótica FCE
BUAP

Código MATLAB 3: FiltroC

Robótica, Período de primavera (24 de enero de 2020). Licenciatura en Ingeniería Mecatrónica.
Dr. Fernando Reyes Cortés. Facultad de Ciencias de la Electrónica, BUAP.

FiltroC.m — MATLAB 2019a

```
function xp=filtroC(t,x)
    u=cos(t);
    a=3;
    b=3;
    xp=-a*x+b*u;
end
```

Estimación de la derivada de $\frac{d}{dt}\cos(t) = -\operatorname{sen}(t)$.

Código MATLAB 3: simuFiltroC

Robótica, Período de primavera (24 de enero de 2020). Licenciatura en Ingeniería Mecatrónica.
Dr. Fernando Reyes Cortés. Facultad de Ciencias de la Electrónica, BUAP.

simuFiltroC.m — MATLAB 2019a

```
clc;
clear all;
close all;
format short
ti=0; h=0.001;  tf = 10;  %tiempo de simulación (segundos).
ts=(ti:h:tf); %vector tiempo.
[renglones, columnas]=size(ts);
opciones=odeset('RelTol',1e-06, 'AbsTol',1e-06,'InitialStep',h,'MaxStep',h);
cond_iniciales=1;
[t, F]=ode45('filtroC',ts, cond_iniciales,opciones);
Fp=filtroC(t,F);
plot(t,Fp, t, -sin(t))
```

Aproximación de la derivada de la función $\cos(t)$

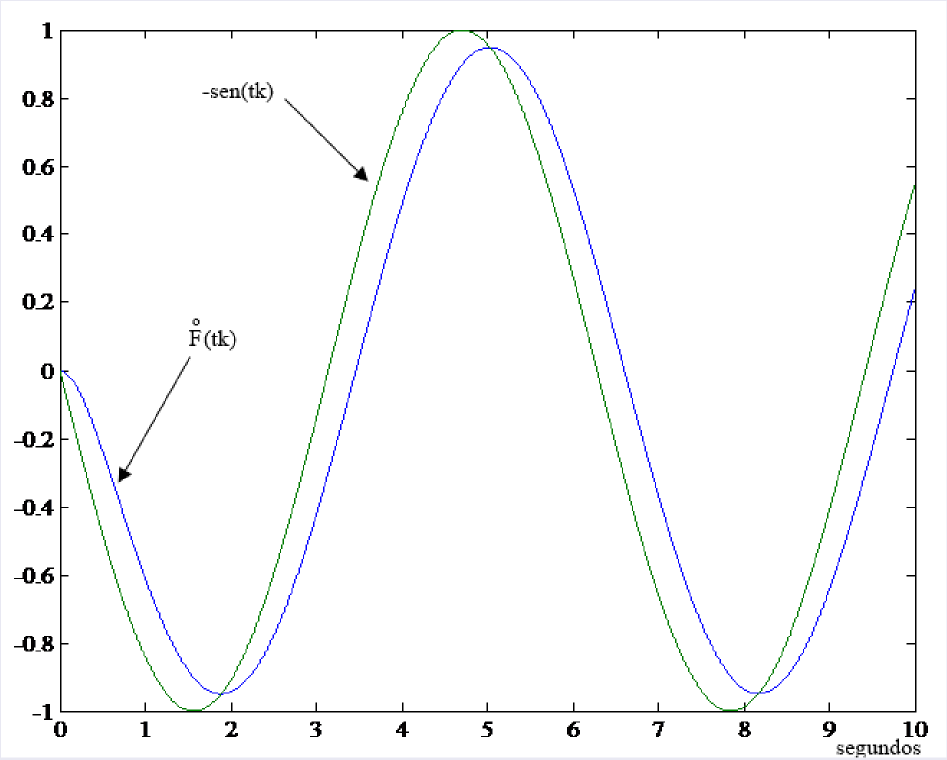


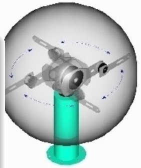
Figura 3: Gráfica comparativa de $\frac{d}{dt}\cos(t) = -\operatorname{sen}(t)$ y $\dot{F}(t) \simeq \frac{d}{dt}\cos(t)$.

Características importantes de la estimación de la derivada: $\dot{F}_u(t) \simeq \frac{d}{dt}u(t)$

- La figura 3 muestra la derivada $\frac{d}{dt}\cos(t) = -\operatorname{sen}(t)$ y la estimación de la deriva por medio de filtrado:

$$\dot{F}(t) \simeq -\operatorname{sen}(t) = \frac{d}{dt}\cos(t)$$

- La respuesta de la estimación de la derivada $\dot{F}_u(t)$ está formada por una etapa transitoria y estado estacionario; sólo en el estado estacionario es donde se da la convergencia hacia la estimación de la derivada $\frac{d}{dt}u(t)$.
- Observe la distorsión de la fase en la señal de estimación de la derivada $\dot{F}_u(t)$, repercute como un corrimiento en el tiempo.



Robótica FCE
BUAP

Código MATLAB 5: FiltroC

Robótica, Período de primavera (24 de enero de 2020). Licenciatura en Ingeniería Mecatrónica.
Dr. Fernando Reyes Cortés. Facultad de Ciencias de la Electrónica, BUAP.

FiltroC.m	MATLAB 2019a

```
function xp=filtroC(t,x)
    u=t.*t;
    a=3;
    b=3;
    xp=-a*x+b*u;
end
```

Robótica FCE
BUAP

Estimación de la derivada de $\frac{d}{dt}t^2 = 2t$.

Código MATLAB 5: simuFiltroC

Robótica, Período de primavera (24 de enero de 2020). Licenciatura en Ingeniería Mecatrónica.
Dr. Fernando Reyes Cortés. Facultad de Ciencias de la Electrónica, BUAP.

simuFiltroC.m — MATLAB 2019a

```
clc;
clear all;
close all;
format short
ti=0; h=0.001;  tf = 10;  %tiempo de simulación (segundos).
ts=(ti:h:tf); %vector tiempo.
[renglones, columnas]=size(ts);
opciones=odeset('RelTol',1e-06,'AbsTol',1e-06,'InitialStep',h,'MaxStep',h);
cond_iniciales=0;
[t, F]=ode45('filtroC',ts, cond_iniciales,opciones);
Fp=filtroC(t,F);
plot(t,Fp, t, 2*t)
```

Robótica FCE BUAP

Aproximación de la derivada de la función t^2

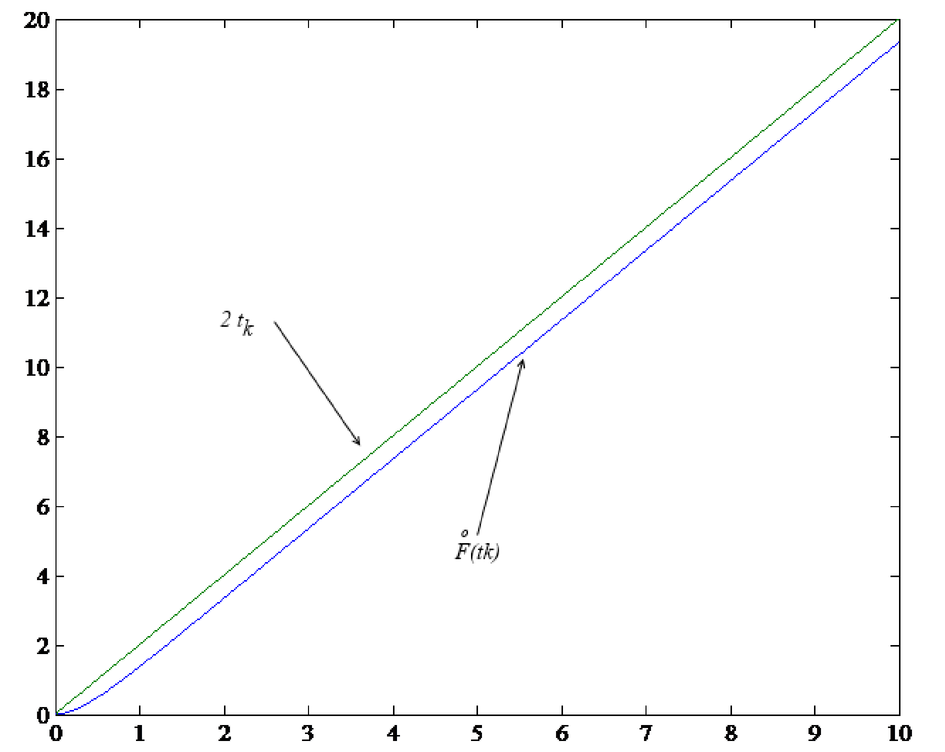


Figura 4: $\frac{d}{dt}t^2 = 2t$ y $\dot{F}(t) \simeq \frac{d}{dt}t^2$.

Robótica FCE
BUAP

Código MATLAB 7: FiltroC

Robótica, Período de primavera (24 de enero de 2020). Licenciatura en Ingeniería Mecatrónica.
Dr. Fernando Reyes Cortés. Facultad de Ciencias de la Electrónica, BUAP.

FiltroC.m — MATLAB 2019a

```
function xp=filtroC(t,x)
    u=1;
    a=3;
    b=3;
    xp=-a*x+b*u;
end
```

Estimación de la derivada de una constante: $\frac{d}{dt}1 = 0$.

Código MATLAB 7: simuFiltroC

Robótica, Período de primavera (24 de enero de 2020). Licenciatura en Ingeniería Mecatrónica.
Dr. Fernando Reyes Cortés. Facultad de Ciencias de la Electrónica, BUAP.

simuFiltroC.m — MATLAB 2019a

```
clc;
clear all;
close all;
format short
ti=0; h=0.001;  tf = 10;  %tiempo de simulación (segundos).
ts=(ti:h:tf); %vector tiempo.
[renglones, columnas]=size(ts);
opciones=odeset('RelTol' ,1e-06, 'AbsTol' ,1e-06,'InitialStep' ,h,'MaxStep' ,h);
cond_iniciales=0;
[t, F]=ode45('filtroC' ,ts, cond_iniciales,opciones);
Fp=filtroC(t,F);
plot(t,Fp, t, 0)
```

Aproximación de la derivada de una constante

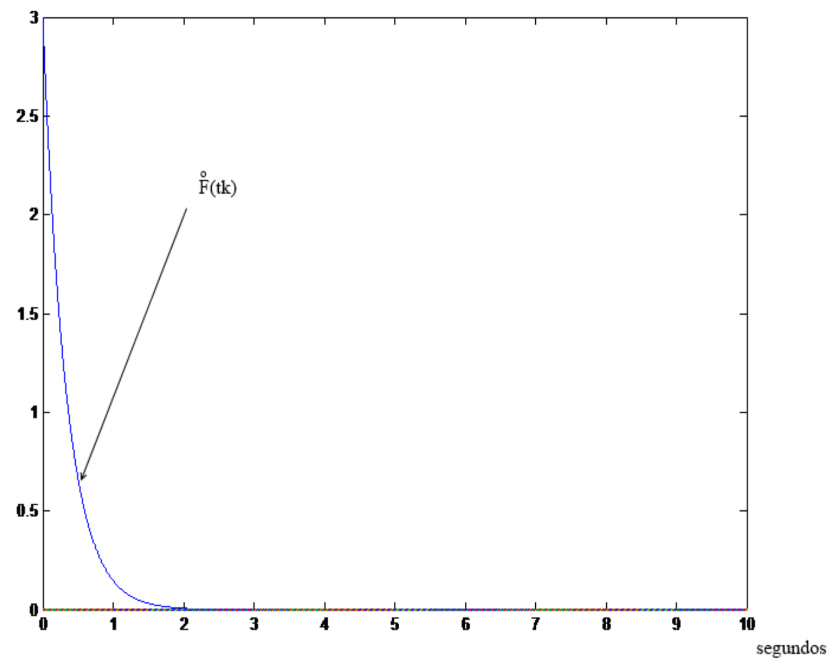


Figura 5: $\frac{d}{dt}1 = 0$ y $\dot{F}(t) \simeq \frac{d}{dt}1 = 0$.

Después de la fase transitoria de $\dot{F}(t)$, en el estado estacionario se da la convergencia; $\dot{F}(t) \simeq \frac{d}{dt}1 = 0$.

Código MATLAB 8: preliminares18c

Robótica, Período de primavera (24 de enero de 2020). Licenciatura en Ingeniería Mecatrónica.
Dr. Fernando Reyes Cortés. Facultad de Ciencias de la Electrónica, BUAP.

preliminares18c.m MATLAB 2019a

```
clc;  clear all;  close all; format short
ti=0; tf = 10; h=0.001; % parámetros de simulación
ts=ti:h:tf; % intervalo de simulación.
cond_iniciales=[0;0];
opciones=odeset('RelTol' ,1e-06, 'AbsTol' ,1e-06,'InitialStep' ,h,'MaxStep' ,h);
disp('Simulación de un sistema lineal de segundo orden' )
[t, x]=ode45('preliminares17' ,ts,cond_iniciales,opciones);
[n, m]=size(x(:,1)); y=zeros(n,m); F=zeros(n,m); Fp=zeros(n,m); u=ones(n,m);
wn=1; rho=0.8;
for k=3:n %simulación del sistema discreto de segundo orden.
    y(k)=(1/(wn^2+(2*rho*wn)/h+1/h^2))*(wn^2*u(k)+(2/h^2+(2*rho*wn)/h)*y(k-1)-1/h^2*y(k-2));
end
lambda=35;
fork=2:n %estimación de la velocidad por filtrado.
    F(k)=exp(-h*lambda)*F(k-1)+(1-exp(-h*lambda))*x(k-1,1); %F(t_k) es el filtro de x_1(t_k).
    Fp(k)=-lambda*F(k)+lambda*x(k,1); % Fp(t_k) es la aproximación de la derivada de x_1(t_k), es decir x_2(t_k).
end
figure
plot(t, x(:,2),t,Fp) %comparación entre la derivada ẋ_1(t_k) = x_2(t_k) y la derivada estimada Fp(t_k) ≃ x_2(t_k).
```

Estimación de la derivada de la posición $x_1(t)$ por filtrado.

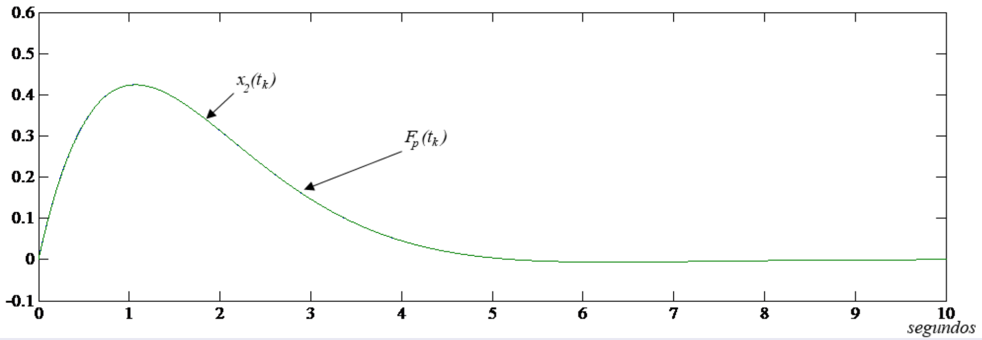


Figura 6: Estimación de la derivada de $x_1(t_k)$ por filtrado.

Resumen

Aplicaciones del sistema: $\dot{F}(t) = -\lambda F(t) + \lambda u(t);\ F(s) = \frac{\lambda}{s+\lambda} u(s)$

- Filtrado de señales.
- Obtiene en forma aproxima la derivada de la señal filtrada de la entrada.

Forma recursiva:

$$\begin{aligned} F(t_k) &= e^{-\lambda h} F(t_{k-1}) + \left(1 - e^{-\lambda h}\right) u(t_{k-1}) \\ \dot{F}(t) \approx \text{Derivada}_u(t_k) &= -\lambda F(t_k) + \lambda u(t_k) \end{aligned}$$