Laboratorio Nº2

Señales

Martín Josemaría Vuelta Rojas

Problema 1

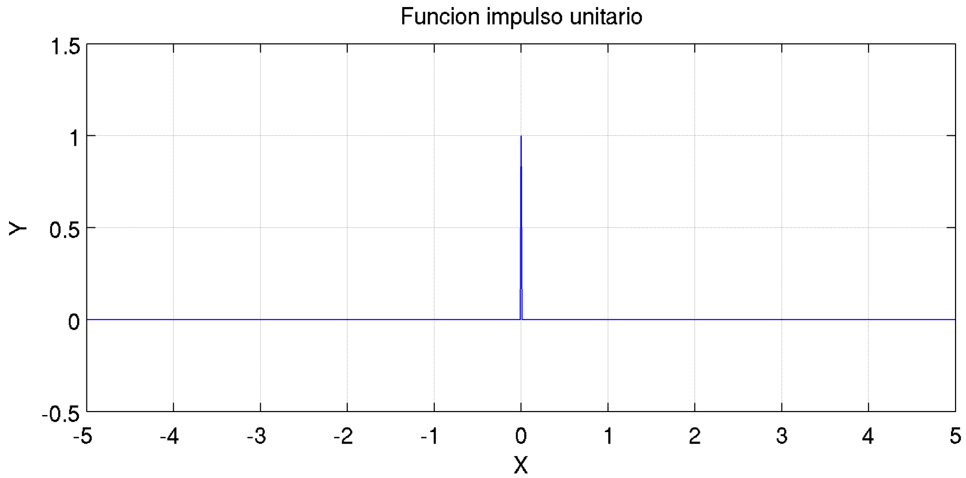
Utilizando MATLAB, haga un programa (`function`) que evalúe las funciones singulares: impulso unitario, escalón unitario y función rampa. Debe graficar cada función singular.

Solución

Script 1 Función impulso unitario

```
function f = impulso(x,y,z)
    switch (nargin)
        case 1, f = 1.*(x==0);
        case 2, f = 1.*(x==y);
        case 3, f = z.*(x==y);
        otherwise
            fprintf('Error: Revise los argumentos de entrada')
end
```

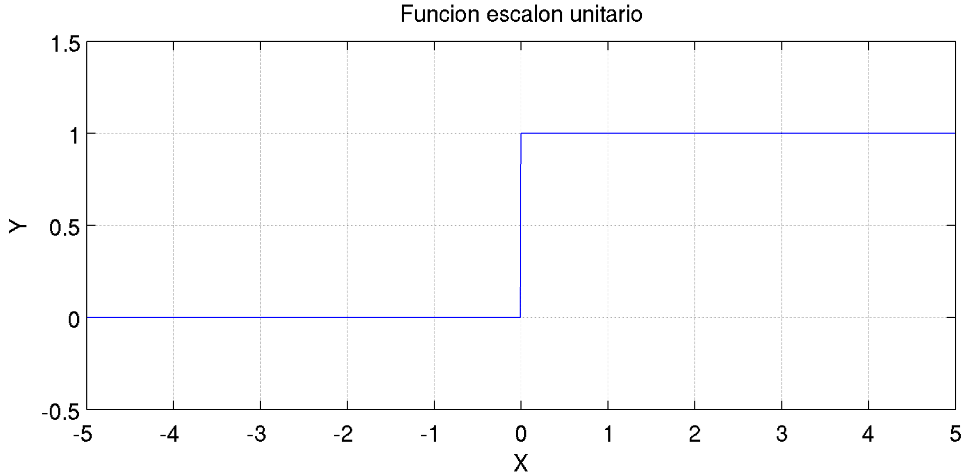
Figura 1 Gráfico de la función impulso unitario del *script* 1.



Script 2 Función escalón unitario

```
function f = escalon(x,y,z)
    switch (nargin)
        case 1, f = 1.*(x>=0);
        case 2, f = 1.*(x>=y);
        case 3, f = z.*(x>=y);
        otherwise
            fprintf('Error: Revise los argumentos de entrada')
end
```

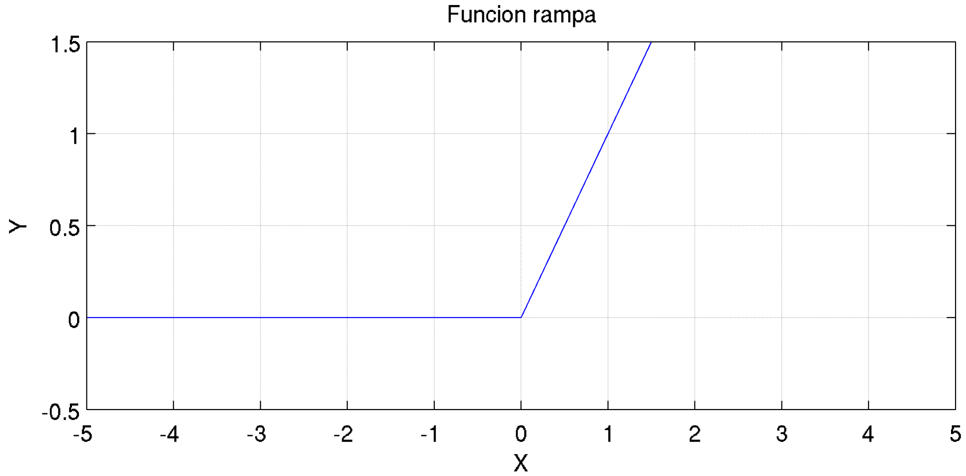
Figura 2 Gráfico de la función escalón unitario del *script* 2.



Script 3 Función rampa

```
function f = rampa(x,y,z)
    switch (nargin)
        case 1, f = x.*(x>=0);
        case 2, f = (x-y).*(x>=y);
        case 3, f = z*(x-y).*(x>=y);
        otherwise
            fprintf('Error: Revise los argumentos de entrada')
end
```

Figura 3 Gráfico de la función rampa del *script* 3.



Problema 2

Haga un programa para visualizar la función compuerta unitaria de

1. Utilizar los comandos `zeros` y `ones`.
2. Utilizar la función desarrollada en el problema 1.

Solución

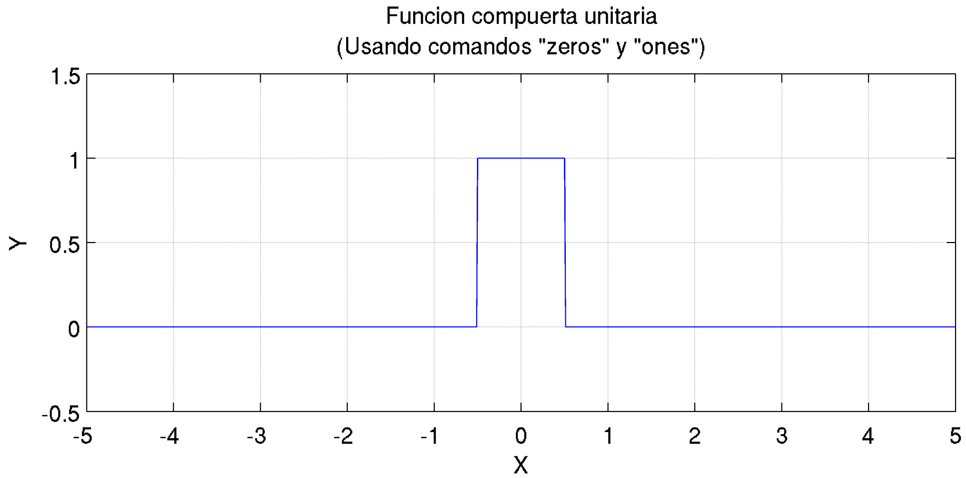
Script 4 Compuerta unitaria empleando `zeros` y `ones`

```
L = 1;

X = -5:0.01:5;
Y = ones(size(X)).*(-0.5*L*ones(size(X)) <= X & X <= 0.5*L*ones(size(X)));

plot(X,Y)
ylim([-0.5, 1.5])
xlabel('X');
ylabel('Y');
title(sprintf('Funcion compuerta unitaria\n(Usando comandos "zeros" y "ones")'));
grid on
```

Figura 4 Resultado de ejecutar el *script* 4.



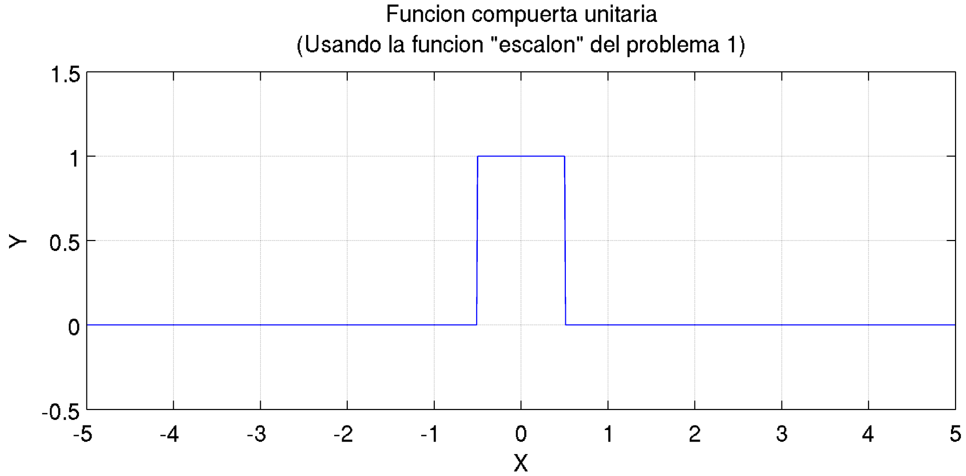
Script 5 Compuerta unitaria empleando la función `escalon` desarrollada en el problema 1

```
L = 1;

X = -5:0.01:5;
Y = escalon(X,-0.5*L).*escalon(-X,-0.5*L);

plot(X,Y)
ylim([-0.5, 1.5])
xlabel('X');
ylabel('Y');
title(sprintf('Funcion compuerta unitaria\n(Usando la funcion "escalon" del
    problema 1)'))
grid on
```

Figura 5 Resultado de ejecutar el *script* 5.



Problema 3

Desarrollar un conjunto de comandos MATLAB para aproximar las siguientes señales periódicas en tiempo continuo, dibujando 5 ciclos de cada una:

1. Onda Cuadrada, de amplitud 5 Volts, frecuencia fundamental 20 Hz.

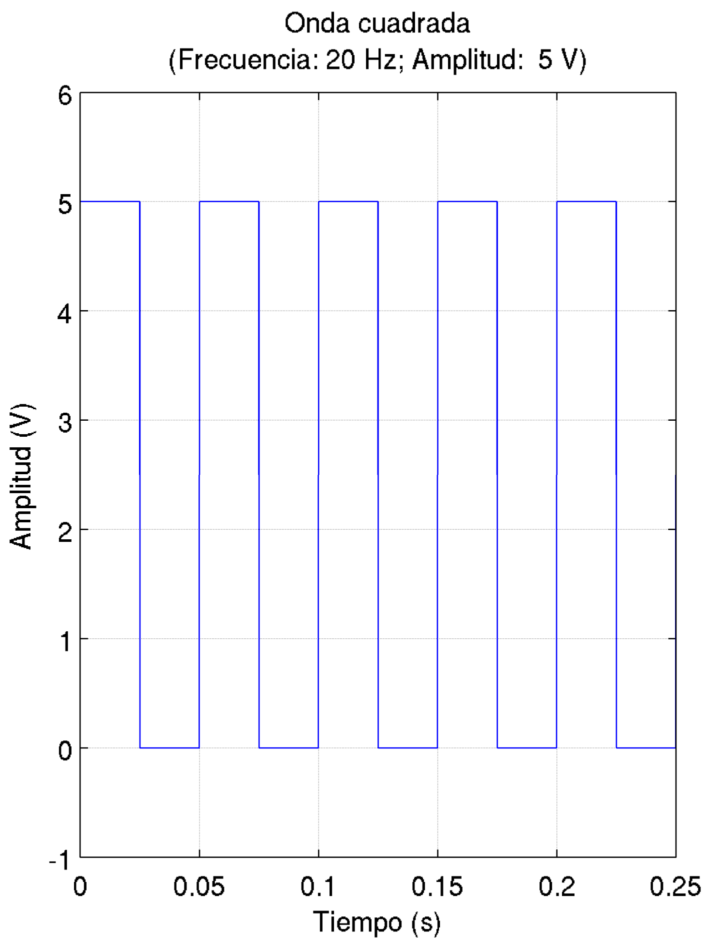
2. Señal diente de sierra, amplitud 5 Volts y frecuencia fundamental 20Hz

Solución

Script 6 Función de onda cuadrada

```
function f = squarew(x,y,z,w)
    switch (nargin)
        case 1, f = 1.*(mod(floor(2*x),2) == 0);
        case 2, f = 1.*(mod(floor(2*y*x),2) == 0);
        case 3, f = z.*(mod(floor(2*y*x),2) == 0);
        case 4, f = z.*(mod(floor(2*y*(x + w)),2) == 0);
        otherwise
            fprintf('Error: Revise los argumentos de entrada\n')
end
```

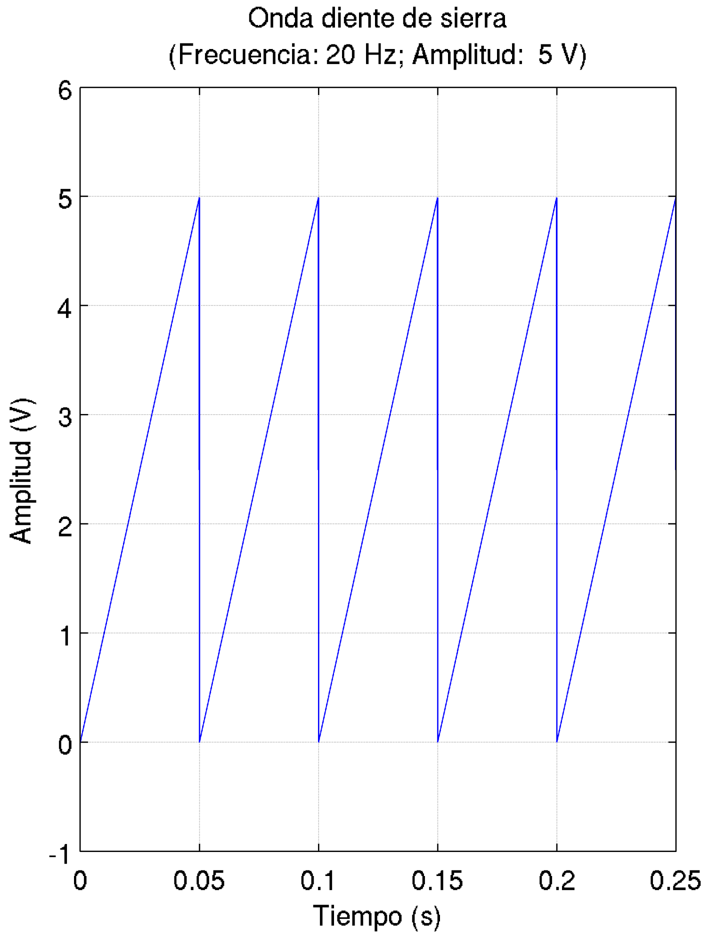
Figura 6 Gráfico de la función rampa del *script* 6.



Script 7 Función de onda diente de sierra

```
function f = saww(x,y,z,w)
    switch (nargin)
        case 1, f = x - floor(x);
        case 2, f = y*x - floor(y*x);
        case 3, f = z*(y*x - floor(y*x));
        case 4, f = z*(y*(x + w) - floor(y*(x + w)));
        otherwise
            fprintf('Error: Revise los argumentos de entrada')
end
```

Figura 7 Gráfico de la función rampa del *script* 7.



Problema 4

La solución a una ecuación diferencial está dada por:

$$x\left(x\right) = 10\mathrm{e}^{-t} - 5\mathrm{e}^{-0{,}5t}$$

Usando MATLAB, grafique la solución de la ecuación en el intervalo $I = [0, 5]$ con una frecuencia de muestreo de 100 Hz.

Solución

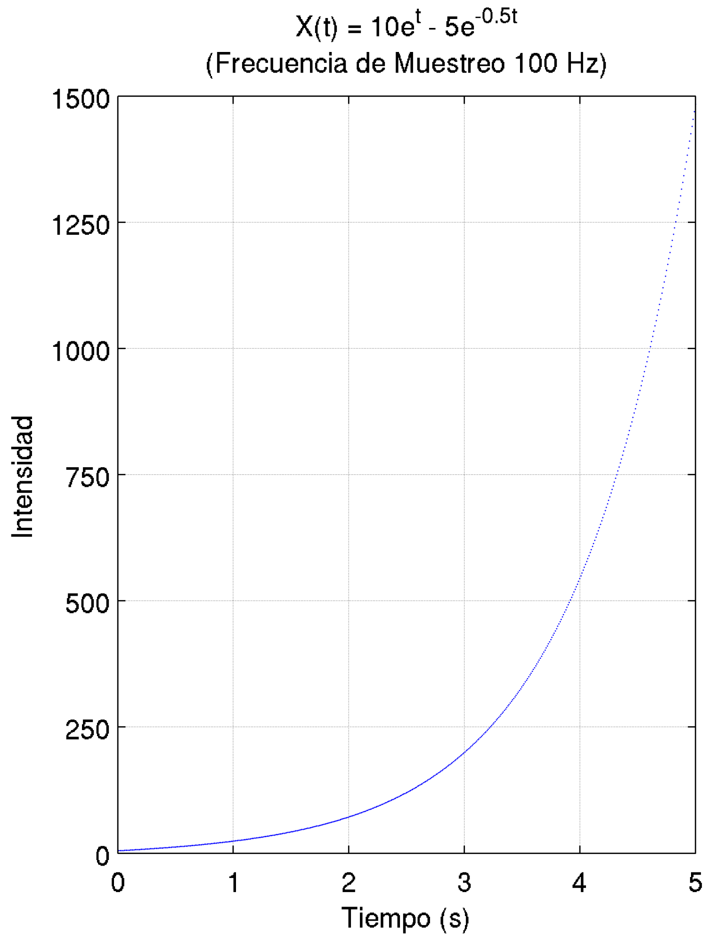
Script 8

```
Fs = 100;

t  = linspace(0,5,5*Fs);
X  = 10*exp(1*t) - 5*exp(-0.5*t);

plot(t,X);
xlabel('Tiempo (s)')
ylabel('Intensidad')
title(sprintf('X(t) = 10e^{t} - 5e^{-0.5t}\nFrecuencia de Muestreo 100 Hz'))
grid on
```

Figura 8 Resultado de ejecutar el *script* 8.



Problema 5

Repita el problema anterior para la siguiente expresión:

$$x(x) = 10\mathrm{e}^{-t} + 5\mathrm{e}^{-0{,}5t}$$

Solución

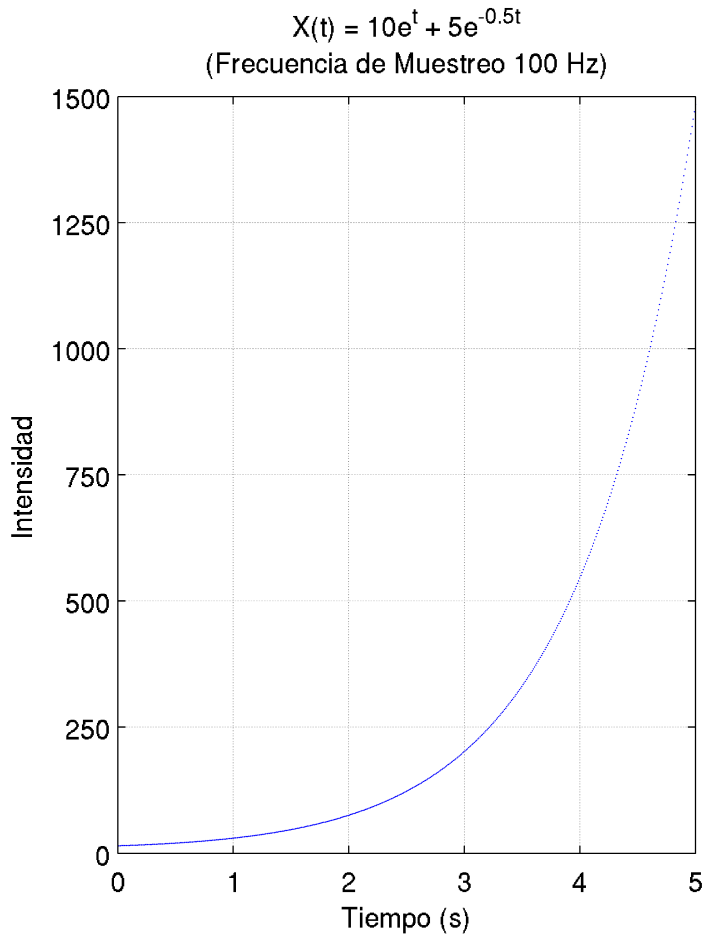
Script 9

```
Fs = 100;

t  = linspace(0,5,5*Fs);
X  = 10*exp(1*A) + 5*exp(-0.5*A);

plot(t,X);
xlabel('Tiempo (s)')
ylabel('Intensidad')
title(sprintf('X(t) = 10e^{t} + 5e^{-0.5t} \n Frecuencia de Muestreo 100 Hz'))
grid on
```

Figura 9 Resultado de ejecutar el *script* 9.



Problema 6

Una señal sinusoidal con amortiguación exponencial está definida por la siguiente expresión:

$$x(x) = \mathrm{e}^{-at} \cos(2\pi f t)$$

donde $f = 1$ Hz y el parámetro a es variable y toma valores sobre el siguiente conjunto: 1, 5, 20. Usando MATLAB, investigar el efecto de variar dicho parámetro en la señal en el intervalo $[0, 5]$. Utilice una frecuencia de muestreo de 20 Hz. Calcule el valor de a para el caso de amortiguamiento crítico. Haga una gráfica para cada caso.

Solución

Script 10

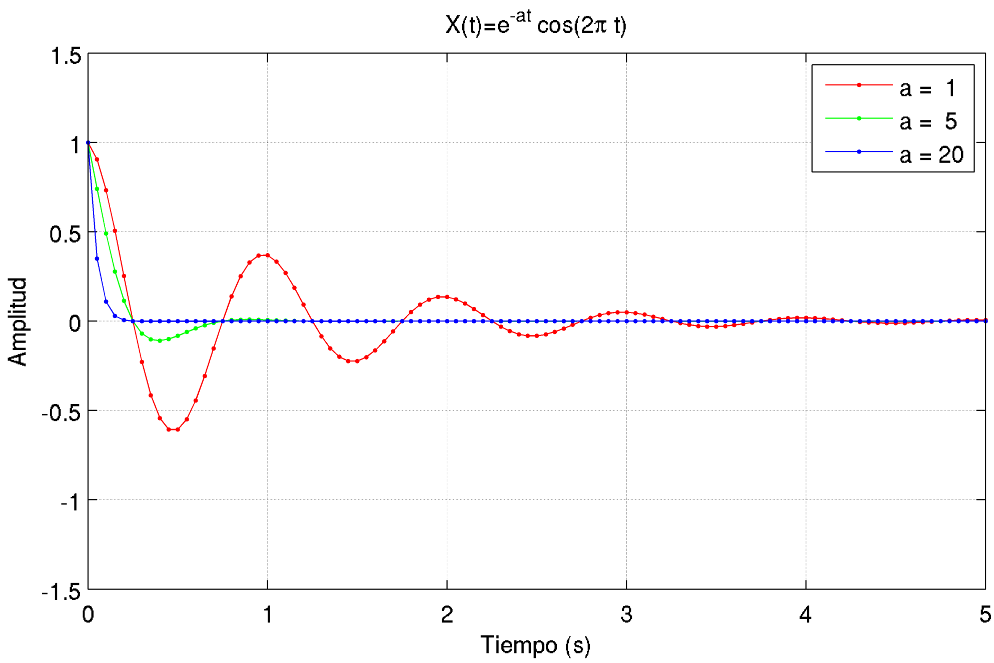
```
Fs = 20;
f  = 1;

t  = linspace(0,5,5*Fs);
X  = zeros(3,length(t));

X(1,:) = exp( -1*t).*cos(2*pi*f.*t);
X(2,:) = exp( -5*t).*cos(2*pi*f.*t);
X(3,:) = exp(-20*t).*cos(2*pi*f.*t);

hold on
plot(t,X(1,:),'r','DisplayName','a =  1')
plot(t,X(2,:),'g','DisplayName','a =  5')
plot(t,X(3,:),'b','DisplayName','a = 20')
xlabel('Tiempo (s)')
ylabel('Amplitud')
title('X(t)=e^{-at} cos(2\pi t)')
legend('show')
grid on
```

Figura 10 Resultado de ejecutar el *script* 10.



De la ecuacion del oscilador amortiguado:

$$m\frac{\mathrm{d}^2x}{\mathrm{d}t^2}+b\frac{\mathrm{d}^2x}{\mathrm{d}t^2}+kx=0.$$

Para el caso con amortiguamiento débil, $b^2<4km$, la solución es de la forma

$$x\left(t\right)=x_0\mathrm{e}^{-\frac{b}{2m}}\cos\left(\omega t\right).$$

Comparando esta última expresión con la ecuación dada en el enunciado del problema, podemos identificar los términos:

$$a=\frac{b}{2m}$$

y

$$\omega=\sqrt{\frac{k}{m}-\left(\frac{b}{2m}\right)^2}=2\pi.$$

Para el caso de amortiguamiento crítico se tiene que cumplir la condición $b=4km$ de modo que

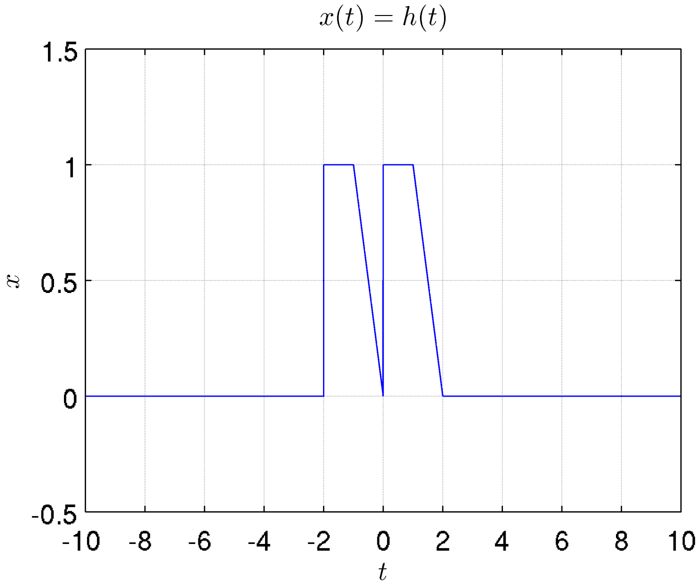
$$a_c^2=\frac{k}{m}$$

Despejando de ω, para un valor dado de a se tiene

$$a_c=\sqrt{4\pi^2-a^2}$$

Problema 7

Para la gráfica mostrada, haga una función en Matlab que visualice $h(t)$. Use el comando `function`.



Graficar:

1. $h\left(t+1\right)$
2. $h\left(\frac{t}{2}-2\right)$
3. $h\left(1-2t\right)$
4. $4h\left(\frac{t}{4}\right)$
5. $\frac{1}{2}h\left(t\right)u\left(t\right)+h\left(-t\right)u\left(t\right)$
6. $h\left(\frac{t}{2}\right)\delta\left(t+1\right)$
7. $h\left(t\right)\left(u\left(t+1\right)-u\left(t-1\right)\right)$

Solución

Script 11 Función que replica la gráfica mostrada en el enunciado.

```
function f = hfun(x)
    f = ((mod(floor(x),2) == 0).*1 + ...
        (mod(floor(x),2) == 1).*(floor(x)-x + 1)).*...
        (-2<x & x<2);
end
```

Script 12 Script para realizar los gráficos solicitados.

```
t = -10:0.0001:10;

figure(1)
x = hfun(t);
plot(t,x)
ylim([min(x)-0.5,max(x) + 0.5])
title('$$h(t)$$','interpreter','latex')
grid on

figure(2)
x = hfun(t+1);
plot(t,x)
ylim([min(x)-0.5,max(x) + 0.5])
title('$$h(t+1)$$','interpreter','latex')
grid on

figure(3)
x = hfun(0.5*t - 2);
plot(t,x)
ylim([min(x)-0.5,max(x) + 0.5])
title('$$h({1 \over 2}t-2)$$','interpreter','latex')
grid on

figure(4)
x = hfun(1 - 2*t);
plot(t,x)
ylim([min(x)-0.5,max(x) + 0.5])
title('$$h(1-2t)$$','interpreter','latex')
grid on

figure(5)
x = 4*hfun(0.25*t);
plot(t,x)
ylim([min(x)-0.5,max(x) + 0.5])
title('$$4h({1\over 4}t)$$','interpreter','latex')
grid on

figure(6)
x = 0.5.*hfun(t).*escalon(t) + hfun(-t).*escalon(t);
plot(t,x)
ylim([min(x)-0.5,max(x) + 0.5])
title('$${1\over 2}h(t)u(t) + h(-t)u(t)$$','interpreter','latex')
grid on

figure(7)
x = hfun(0.5*t).*impulso(t+1);
plot(t,x)
ylim([min(x)-0.5,max(x) + 0.5])
title('$$h({1\over 2}t)\delta(t)$$','interpreter','latex')
grid on

figure(8)
x = hfun(t).*(escalon(t+1)-escalon(t-1));
plot(t,x)
ylim([min(x)-0.5,max(x) + 0.5])
title('$$h(t)(u(t+1)-u(t-1))$$','interpreter','latex')
grid on
```

Figura 11 Resultados de la ejecucion del *script* 12.