

Señales en MATLAB y SIMULINK

Práctica 1

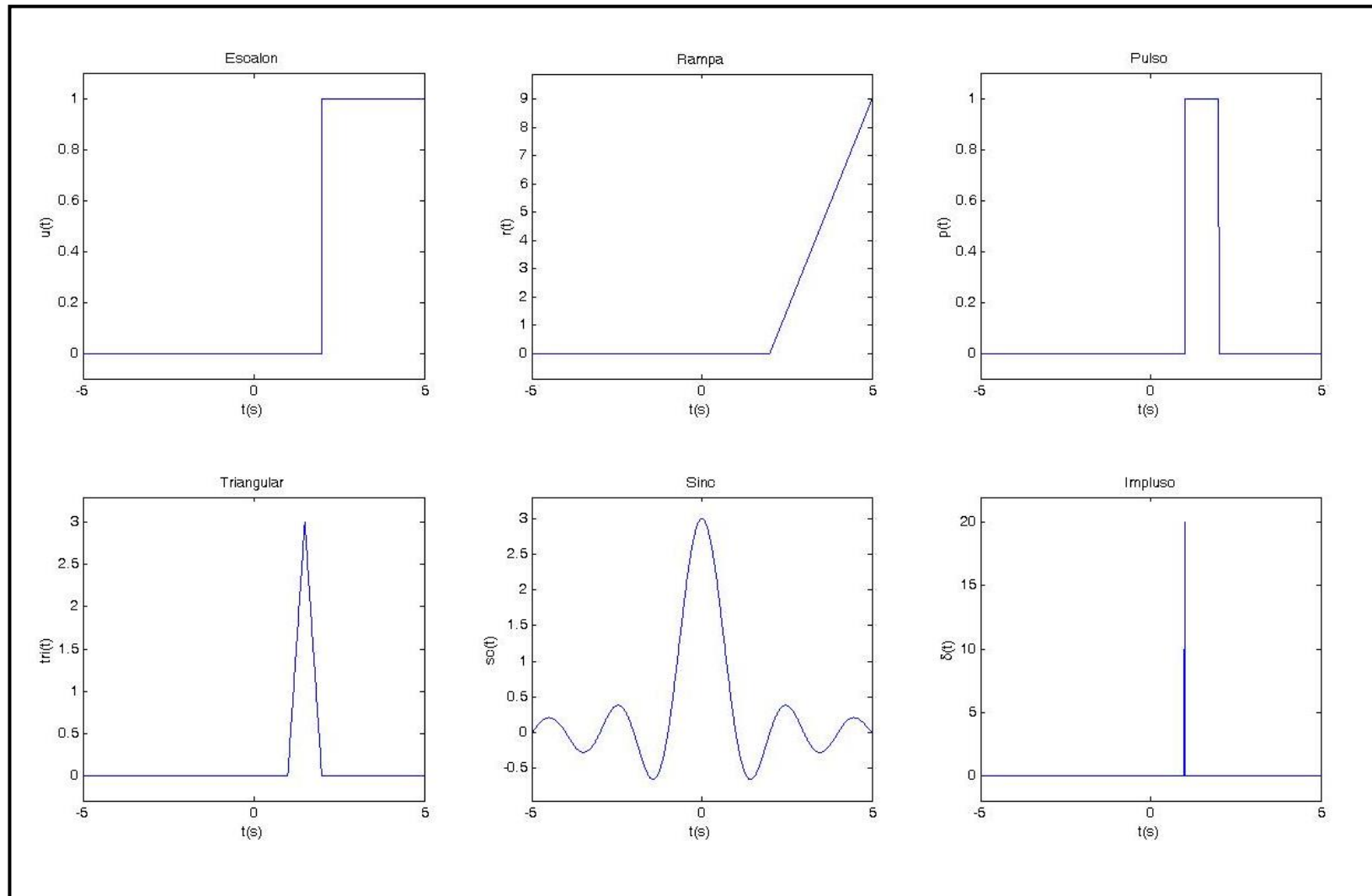
Contenidos de la práctica

1. Catálogo de señales de entrada
 1. Funciones de MATLAB
 2. Integración en SIMULINK
 3. Catálogo de señales de entrada discretas
2. Manipulación de señales discretas
3. Señales periódicas
 1. Tren de pulsos
 2. Dientes de sierra
 3. Seno rectificado de media onda
4. Convolución discreta

1. Catálogo de señales de entrada

- Escalón:
$$e(t) = A \cdot u(t - t_0) = \begin{cases} A & t \geq t_0 \\ 0 & t < t_0 \end{cases}$$
- Rampa:
$$r(t) = m \cdot (t - t_0) \cdot u(t - t_0)$$
- Pulso:
$$p(t) = A \cdot (u(t - t_0) - u(t - t_1))$$
- Triangular:
$$tr(t) = m(t - t_0) \cdot u(t - t_0) - 2m \left(t - \left(\frac{t_0 + t_1}{2} \right) \right) \cdot u \left(t - \frac{t_0 + t_1}{2} \right) + m(t - t_1) \cdot u(t - t_1)$$
- Seno cardinal:
$$sinc(t) = A \frac{\sin\left(\frac{2\pi}{T}t\right)}{t}$$
- Impulso:
$$\delta(t - t_0) = 0, t \neq t_0 \quad \int_{-\infty}^{+\infty} \delta(\tau) d\tau = 1$$

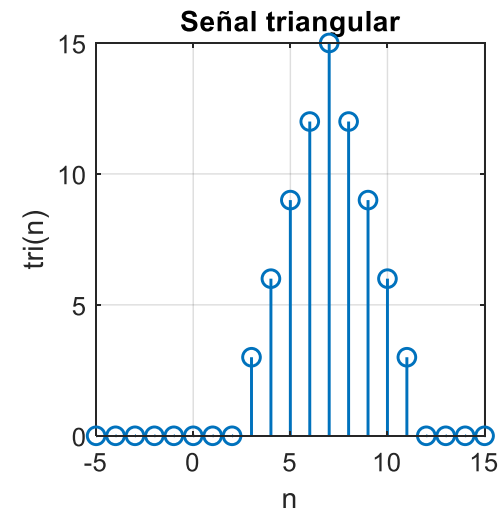
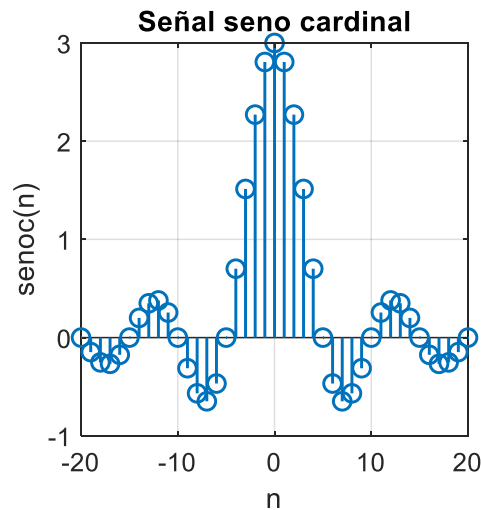
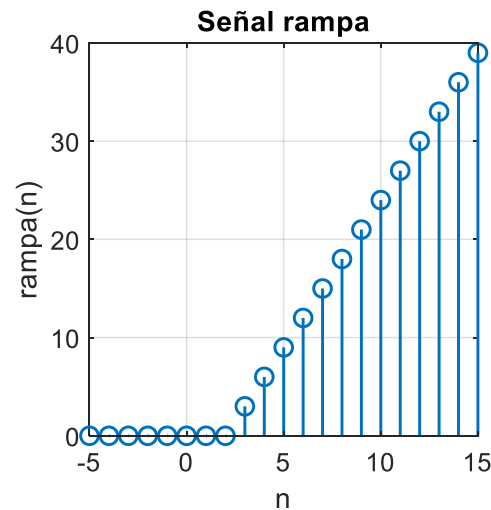
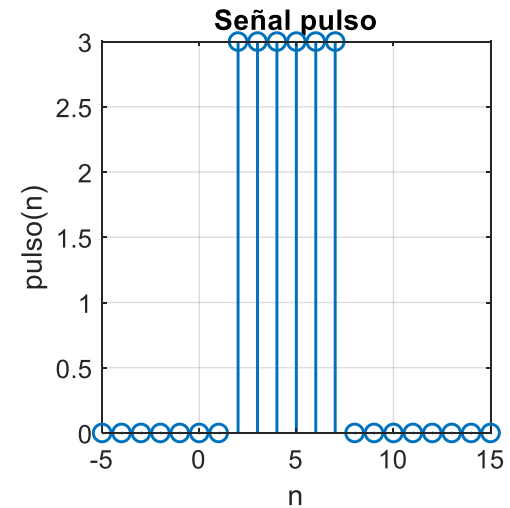
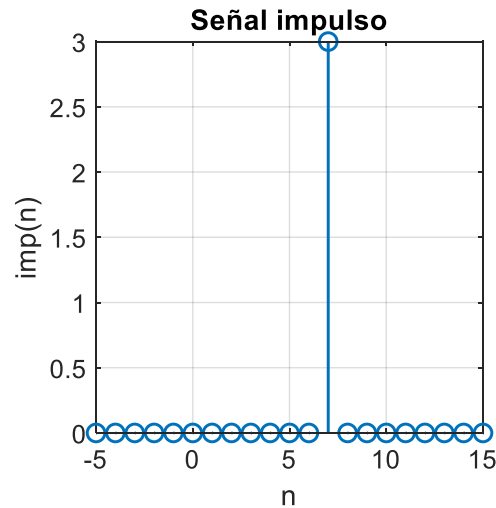
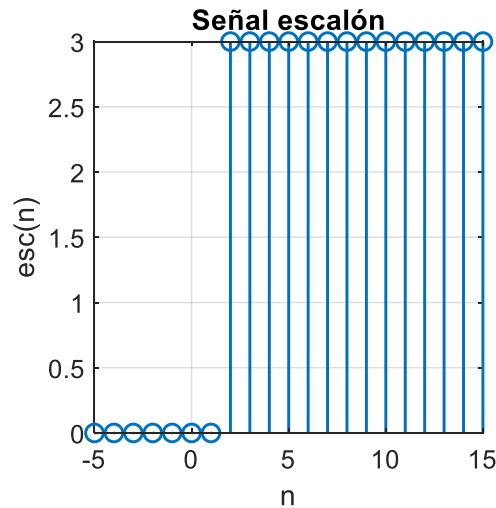
1. Catálogo de señales de entrada



1. Catálogo de señales de entrada discretas

- Escalón: $esc(n) = A \cdot u(n - n_0) = \begin{cases} A & n \geq n_0 \\ 0 & n < n_0 \end{cases}$
- Impulso: $imp(n) = A \cdot \delta(n - n_0) = \begin{cases} A & n \geq n_0 \\ 0 & n < n_0 \end{cases}$
- Pulso: $rect(n) = A \cdot (u(n - n_0) - u(n - n_1))$
- Rampa: $rampa(n) = m \cdot (n - n_0) \cdot u(n - n_0)$
- Seno cardinal: $senoc(n) = A \frac{\sin\left(\frac{\pi}{N}n\right)}{\pi n}$
- Triangular: $tri(n) = ???$

1. Catálogo de señales de entrada



1.2 Funciones de MATLAB

Notación	Parámetro
<code>ni</code>	Muestra de inicio de la señal
<code>nf</code>	Muestra de fin de la señal
<code>no</code>	Muestra de cambio
<code>n1</code>	2ª muestra de cambio
<code>A</code>	Amplitud
<code>m</code>	Pendiente
<code>N</code>	Periodo seno cardinal
<code>p</code>	Número de periodos

- `[n,u] = escalon(ni,nf,no,A)`
- `[n,imp] = impulso(ni,nf,no, A)`
- `[n,p] = pulso(ni,nf,no,n1,A)`
- `[n,r] = rampa(ni,nf,no,m)`
- `[n,sc] = senoc(N,p,A)`
- `[n,tri] = triangular(ni,nf,no,n1,m)`

1.2 Funciones de MATLAB

- Ejemplo: Entrada escalón

```
function [n,u] = escalon(ni, nf, n0, A)
% function [n,u] = escalon(ni, nf, n0, A)
% Arg. entrada:
% * ni : muestra inicial de la senal
% * nf : muestra final de la senal
% * n0 : muestra de comienzo del escalon
% * A : amplitud del escalon

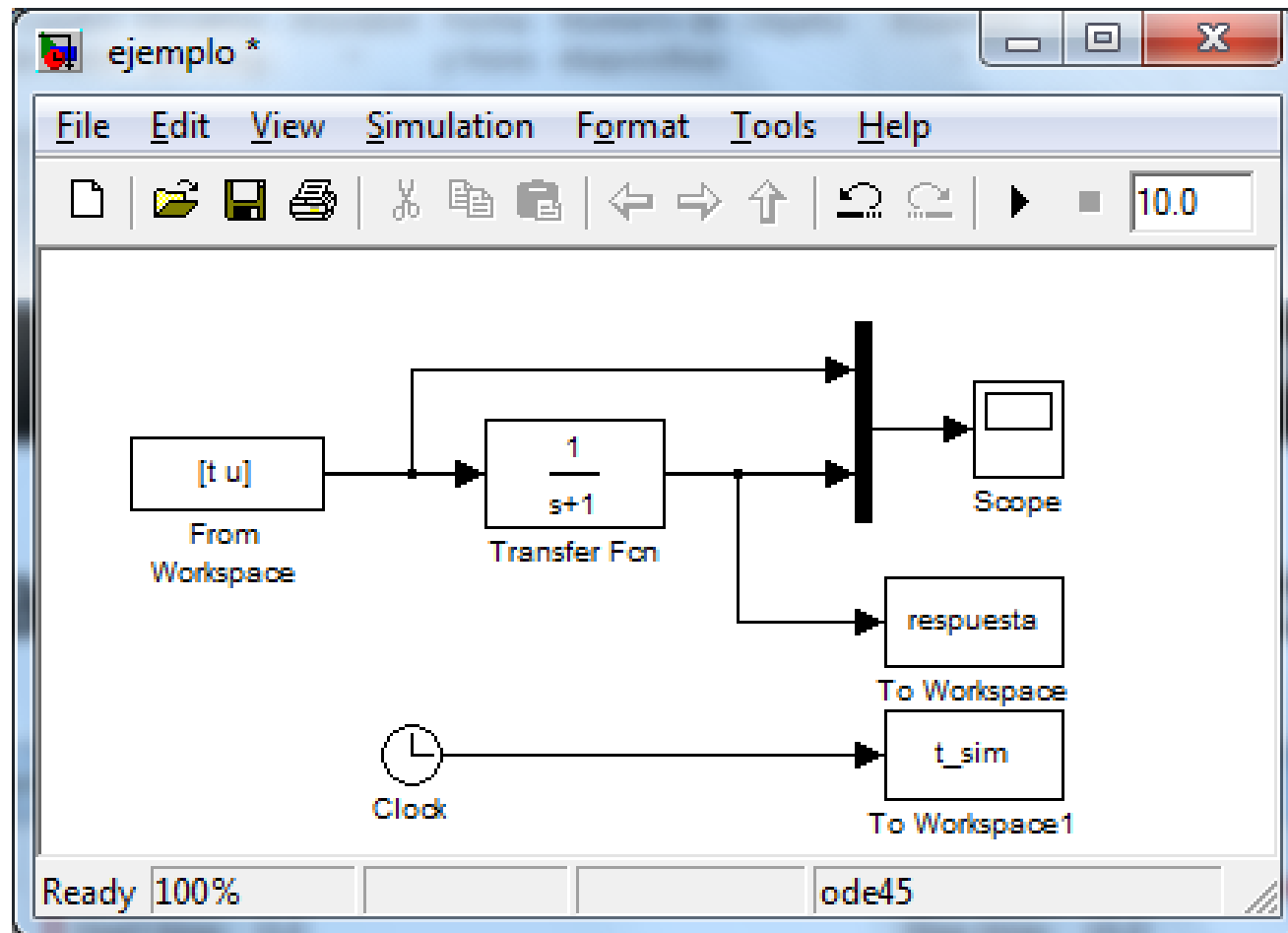
n = ni:nf;
n = n';
u = A*(n>=n0);

end
```

- Crear funciones para las demás entradas del catálogo

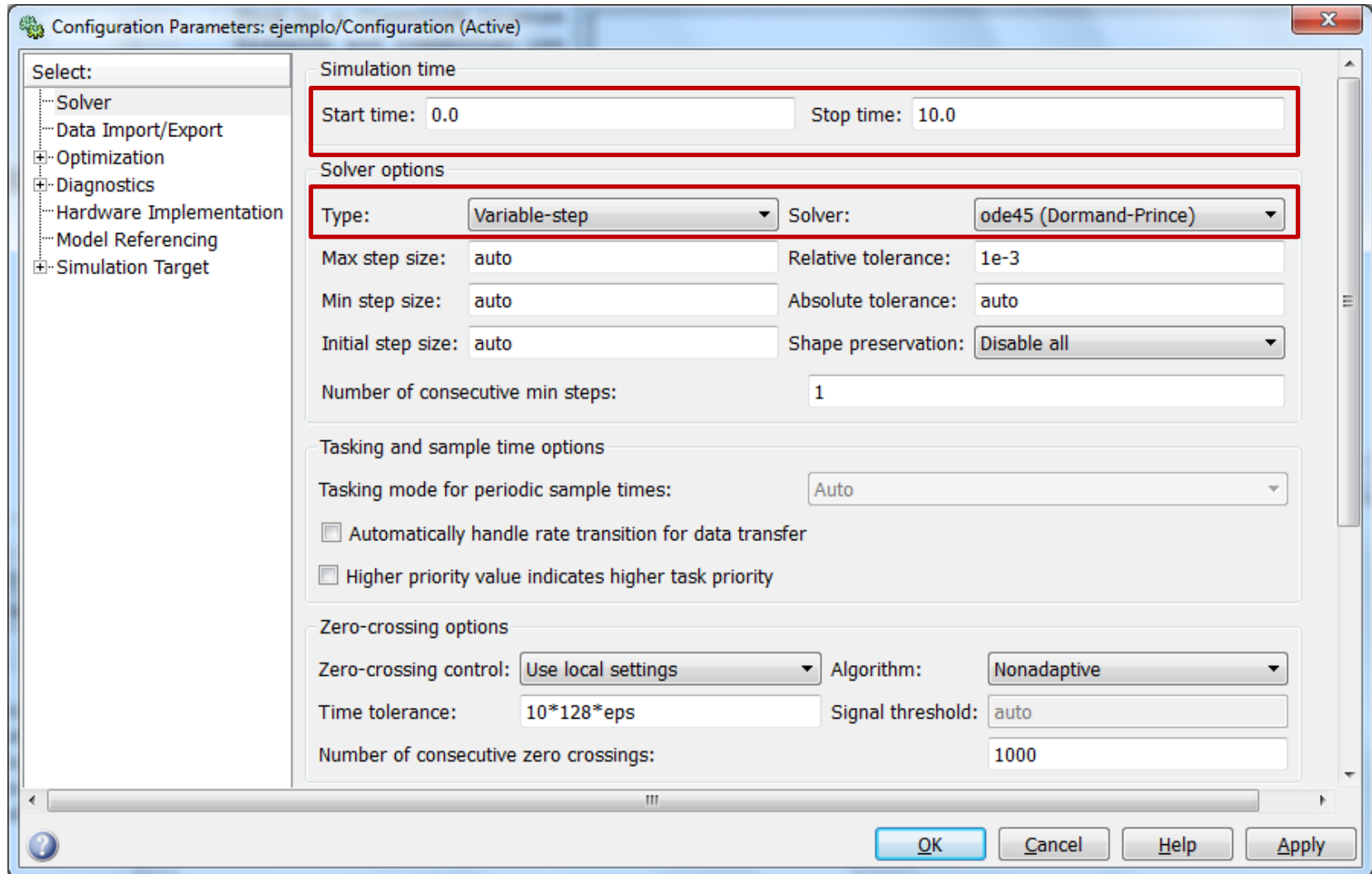
1.3 Integración en SIMULINK

- SIMULINK



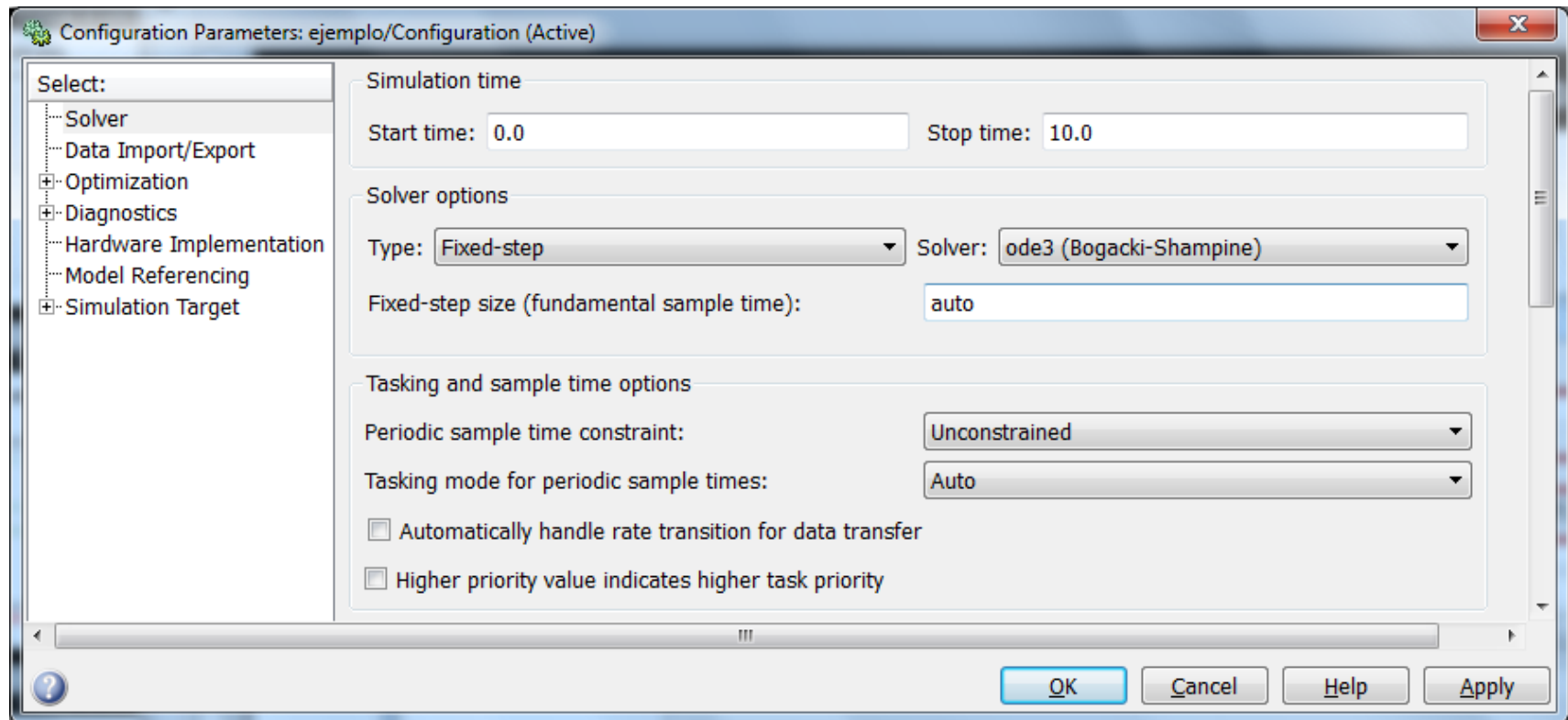
1.3 Integración en SIMULINK

Opciones de simulación (tiempo continuo)



1.3 Integración en SIMULINK

- Opciones de simulación (**tiempo discreto**)

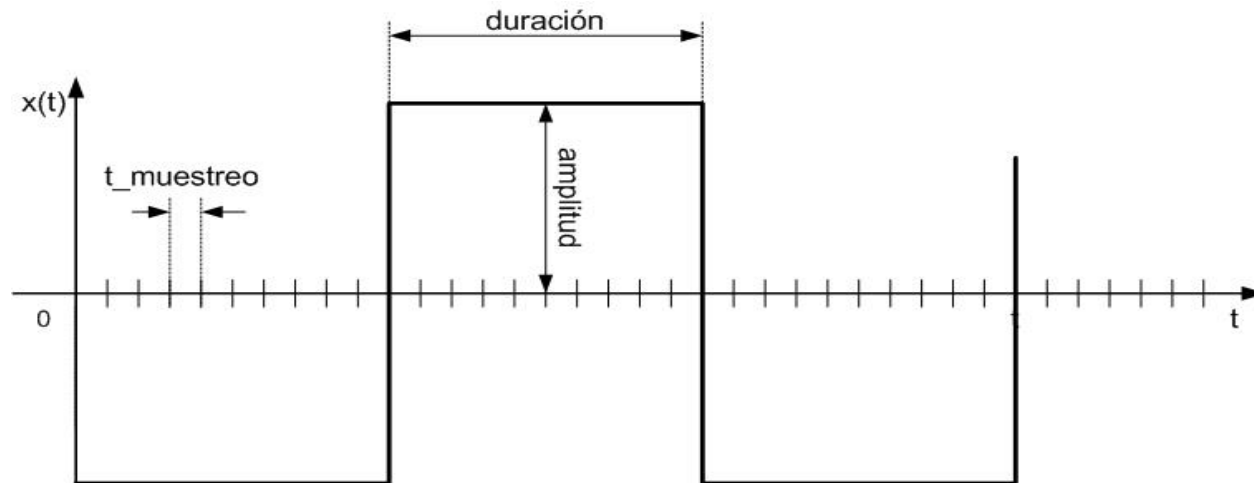


2. Manipulación de señales discretas

- Manipulaciones sobre la variable independiente (n)
 - Desplazamiento $x(n - n_0), x(n + n_0)$
 - Operaciones siguientes respecto de cero!!!!!!
 - Compresión $x(M \cdot n) \rightarrow$ Submuestreo
 - Dilatación $x\left(\frac{n}{N}\right) \rightarrow$ Sobremuestreo
 - ¿Qué hacer con las muestras intermedias?
 - Reflexión $x(-n)$
- Crear funciones en MATLAB para cada operación
 - Las funciones deben mantener los índices de entrada

3. Señales periódicas

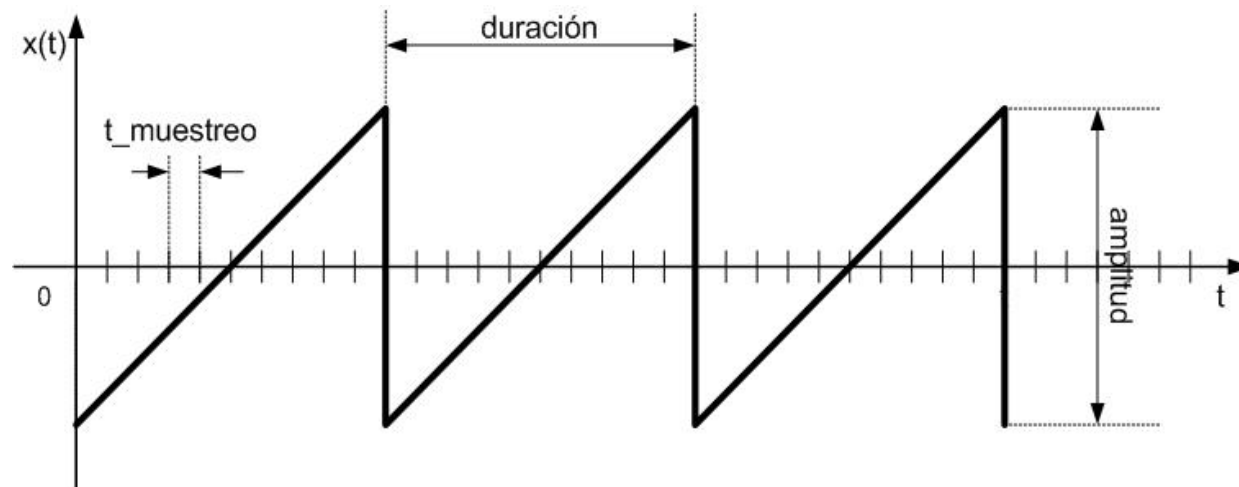
- Tren de pulsos
 - **function** $[t,x] = \text{pulsos}(A,w,n,Ts)$



- **A**: amplitud de los pulsos
- **w**: ancho del pulso
- **n**: número de pulsos
- **Ts**: tiempo de muestreo

3. Señales periódicas

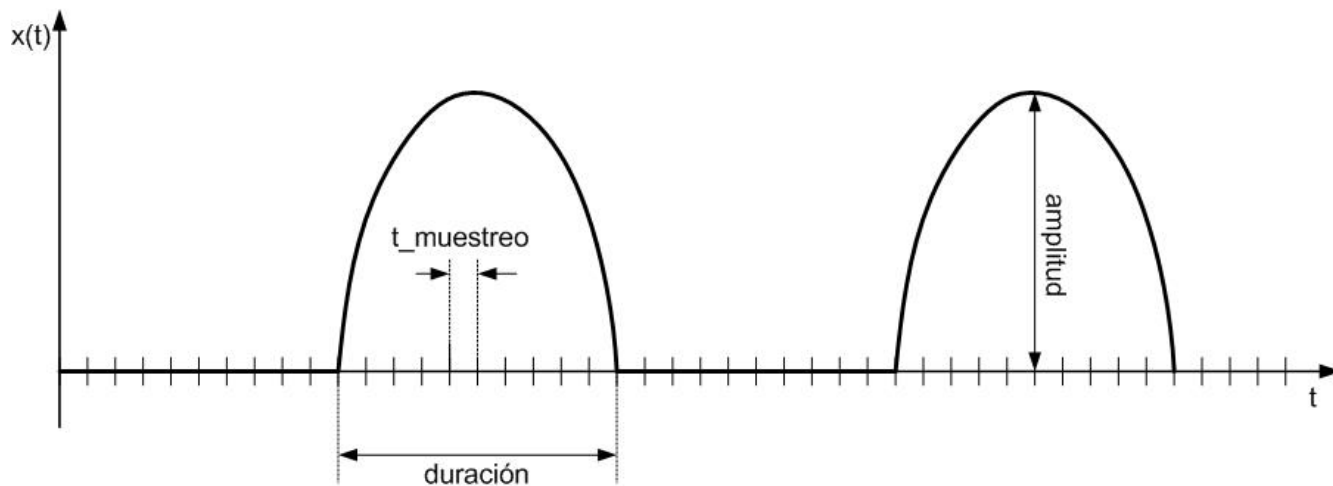
- Dientes de sierra
 - **function** `[t,x] = dientes(A, w, n, Ts)`



- **A**: amplitud de los dientes
- **w**: ancho del diente
- **n**: número de dientes
- **Ts**: tiempo de muestreo

3. Señales periódicas

- Onda senoidal rectificada (media onda)
 - **function** `[t,x] = rectificada(A, w, n, Ts)`



- **A**: amplitud de los pulsos
- **w**: ancho del pulso
- **n**: número de pulsos
- **Ts**: tiempo de muestreo

4. Convolución discreta

- Expresión analítica:

$$y(n) = x(n) * h(n) = \sum_{k=-\infty}^{\infty} x(k) \cdot h(n - k)$$

- Algoritmo
 - Paso 1. Recibir valores de secuencias $x(n)$ y $h(n)$
 - Paso 2. Realizar reflexión sobre la secuencia $h(k) \rightarrow h(-k)$
 - Paso 3. Realizar desplazamiento de n muestras sobre la secuencia reflejada
 - Paso 4. Multiplicar las dos secuencias $x(n)$ y $h(n-k)$
 - Paso 5. Realizar el sumatorio de los productos
 - Paso 6. Incrementar n y volver al paso 3.

4. Convolución discreta

- Implementar una función con la siguiente cabecera:
 - `function cn = convolucion(x, h, n0, n1)`
 - x: secuencia de entrada
 - h: secuencia de entrada
 - n0: primera componente de la convolución
 - n1: última componente de la convolución
- Ejemplo:
 - $x(n)=[0\ 0\ 0\ 0\ 0\ 0\ 1\ 2\ 3\ 2\ 1\ 0\ 0\ 0\ 0\ 0\ 0]$
 - $h(n)=[0\ 0\ 0\ 0\ 0\ 0\ 1\ 2\ 3\ 4\ 5\ 0\ 0\ 0\ 0\ 0\ 0]$
 - $n0=0; n1 = 16$

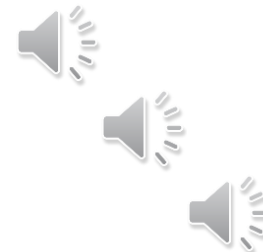
4.1 Auralización

- Modelar y replicar la acústica que la forma de una habitación/espacio provoca sobre un sonido
 - Aplicado a entornos de realidad virtual
 - Efectos de sonido
 - Diseño de auditorios o salas de conciertos
- Cualquier sonido grabado “en seco” (cámara sin ecos) puede ser reproducido como si se hubiese generado en otro lugar mediante la siguiente convolución discreta

$$a(n) = h(n) * s(n)$$

donde

- $s(n)$ un sonido sin eco
- $h(n)$ la respuesta a impulso de la habitación
- $a(n)$ es la secuencia auralizada



5 Algoritmo de Karplus-Strong

- Sintetizamos sonidos mediante la reproducción de señales periódicas que oscilan a una determinada frecuencia
- Generalmente esas señales son sinusoidales, pero podemos utilizar cualquiera
- El algoritmo de Karplus-Strong (KS) propone utilizar señales aleatorias que decaen con el tiempo:

