

# Especificação, Modelação e Projecto de Sistemas Embutidos

## Simulink - introdução **HandsOn Session**

Paulo Pedreiras  
pbrp@ua.pt



Departamento de Electrónica, Telecomunicações e Informática  
Universidade de Aveiro

**V1.2 Outubro/2009**

# Simulink

**Simulink** – o que é e para que serve:

- Extensão gráfica do Matlab
  - **Modelação, simulação e análise** de sistemas
- **Diversas toolboxes** disponíveis para diversas áreas:
  - Processamento de sinal,
  - controlo,
  - comunicação, etc
- **Extensível**: ferramentas para desenvolvimento de blocos/funções pelo utilizador
- Possibilidade de **síntese automática de código**
  - Real-Time Workshop
- Possibilidade de efectuar **simulação** com o “**hardware na malha**” (*hardware in the loop*)

# Simulink

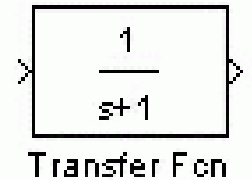
- Em Simulink os **sistemas** são **representados** graficamente por **diagramas de blocos**
- **Diversos tipos de blocos** estão disponíveis:
  - Funções de transferência, operações aritméticas, somas de sinais, multiplexers e de-multiplexers, dispositivos de IO (virtuais), geradores de funções, osciloscópios, ...
- O Simulink encontra-se **bem integrado** no Matlab, sendo fácil transferir dados entre ambos os ambientes
- **Usado em diversas áreas**: controlo de processos industriais, sistemas de controlo de máquinas, sistemas automotivos, industria aeroespacial, ...

# Modelação de sistemas

## Modelação de sistemas

- Em Simulink um **modelo** é uma **coleção de blocos** que **comunicam** uns com os outros por meio de sinais.
- Há **duas classes** essenciais de elementos
  - **Blocos**: são usados para gerar, modificar, combinar ou mostrar sinais
  - **Linhas**: usadas para transferir sinais entre blocos distintos

# Modelação de sistemas



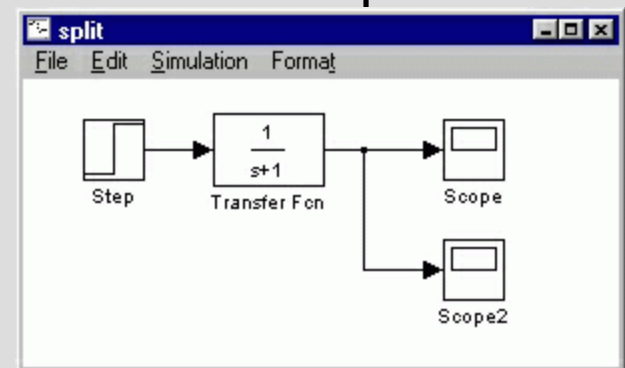
## Blocos (tipos básicos)

- **Fontes (Sources):** Usados para gerar diversos tipos de sinais
- **Consumidores (Sinks):** Usados para mostrar ou efectuar a saída (*output*) de sinais
- **Discretos:** elementos de sistema lineares, de tempo discreto (funções de transferência, estado de espaços, etc.)
- **Lineares:** elementos de sistema lineares, de tempo-contínuo e ligações (junções de soma (algébrica), blocos de ganho, ...)
- **Não-Linear (Non-Linear):** operadores não lineares (funções arbitrárias, saturação, atraso, ...)
- **Conexões:** Multiplex, demultiplex, barramentos, etc.
- Cada bloco pode ter **zero ou vários terminais de entrada** (input) e zero ou vários **terminais de saída** (output). Os terminais são denotados por pequenos triângulos

# Modelação de sistemas

## Linhas

- **Transmitem sinais** na **direcção** indicada pela seta, ligando obrigatoriamente um terminal da saída a um terminal de entrada
  - Exceptua-se a derivação (*tap*) de uma linha, por forma a **dividir um sinal** que será **fornecido** a **dois** blocos receptores.
- Todavia **as linhas não podem injectar um sinal noutra linha.** Nestes casos tem de usar um bloco, e.g. soma.
- Os sinais podem ser **escalares** ou **vectores**. Para sistemas SISO (*Single Input Single Output*) são usados habitualmente sinais escalares. Em sistemas MIMO (*Multiple Input Multiple Output*) são usados frequentemente sinais vectoriais

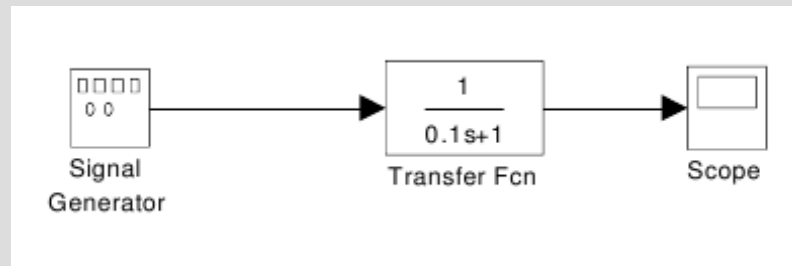


# Criação de um modelo passo-a-passo

- **Apresentam-se agora os passos para construir o modelo e analisar um sistema simples: análise de um filtro passa-baixo.**
- **Iniciar o Simulink**
  - No prompt do Matlab escrever “simulink”
  - No menu “Start” escolher a opção “Simulink”->”Library Browser”
  - Em ambos os casos aparece a janela “Simulink Library Browser” que contem os blocos pré-definidos do Simulink
- **Criar o ficheiro para conter o modelo**
  - File->New->Model
- **Adicionar blocos ao modelo**
  - Adicionar um gerador de impulsos: *Simulink->Sources->Signal Generator*
  - Adicionar um osciloscópio: *Simulink->Sinks->Scope*
  - Adicionar o filtro: *Simulink->-Continuous>Transfer Function*

# Criação de um modelo passo-a-passo

- **Interligar** os diversos blocos

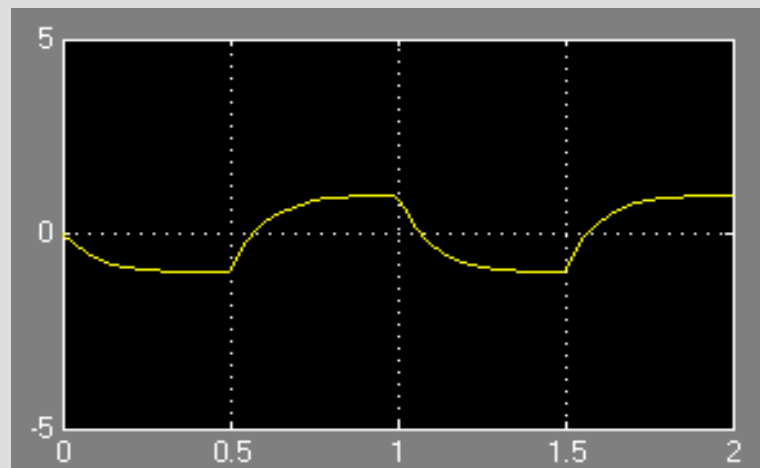


- **Configurar** os diversos componentes
  - As propriedades dos blocos podem ser configuradas com um “duplo click”
    - **Gerador de sinais**: forma da onda, amplitude e frequência
    - **Função de transferência**: especificar os coeficientes da função de transferência
    - **Scope**: abre o osciloscópio



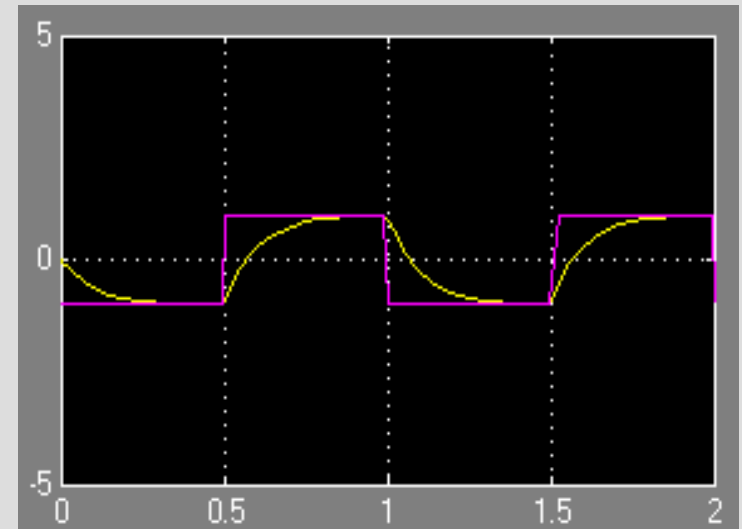
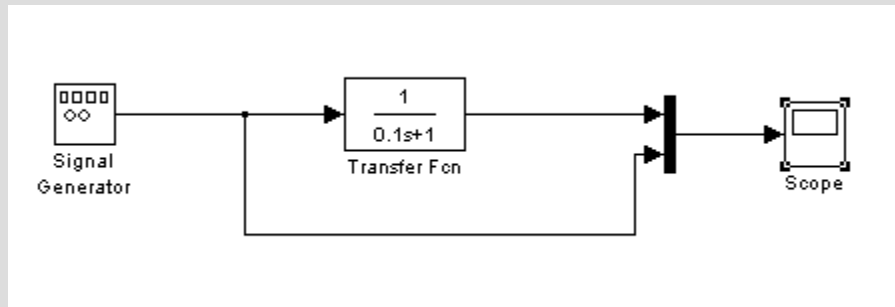
# Criação de um modelo passo-a-passo

- Configurar o gerador de sinais para **onda quadrada, amplitude unitária e frequência de 1Hz**
- Configurar a função de transferência para o polinómio  $1/(s/10+1)$
- Filtro passa-baixo de primeira ordem, com uma constante de tempo  $\tau$  de 0.1s. A frequência de corte é  $f_c = 1/(2*\pi*\tau)$ , a que corresponde uma amplitude de  $1/\text{SQR}(2)$  da amplitude dos sinal de entrada.
- Abrir o osciloscópio
- **Simulation -> Start ou botão “play”**
- Observar a forma e amplitude do sinal de saída



# Criação de um modelo passo-a-passo

- Comparação de sinais
  - Frequentemente é interessante ter o **display simultâneo** de ambos os sinais
    - Deriva-se o sinal da saída do gerador e aplica-se num multiplexer (*Simulink*->*Signal Routing*->*Mux*)



# Exercícios

## Exercício 1 - Análise do comportamento na frequência do filtro

- Efectuar simulações para as seguintes frequências:  
{0.1, 0.5, 1, 2, 5, 10} HZ
- Para cada uma anotar o valor de pico do sinal de saída
- Nota alguma outra alteração?
- Comentar ...

# Exercícios

- Exercício 2 - Modelação de um sistema físico: queda livre de uma bola elástica

