FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Introdução às Telecomunicações

Departamento de Engenharia Eletrotécnica
Secção de Telecomunicações
Mestrado integrado em Engenharia Eletrotécnica e de Computadores
Licenciatura em Engenharia Informática

Grupo: ____ nº _____ e _____

_____

1º Trabalho de Laboratório

Objetivo Geral: Familiarização com o programa SIMULINK, e com o funcionamento dos componentes que vão ser mais usados no laboratório da disciplina.

Este trabalho tem duas partes e dura duas aulas. Cada parte deve ser efetuada numa aula.

PARTE I

Ponto 1 - Visualização de um sinal periódico

Objetivo: Como gerar um sinal? Como podemos ver esse sinal?
Familiarização com o SIMULINK, com os geradores de sinais e com os visualizadores de sinais.

Procedimentos:

1. Abra o MATLAB
2. Escreva "simulink" para começar o programa SIMULINK. Este programa permite ter *modelos* com circuitos eletrónicos para executar o que se pretenda. O SIMULINK arranca com uma janela onde existem conjuntos (pastas) de componentes pré-definidos.
3. Nesta nova janela, abra um novo modelo (menu "*file*") e guarde-o logo com um nome (por exemplo, *sinal.m*). Use uma pendrive para ir guardando o trabalho do seu grupo, evitando encher o disco do computador com os seus trabalhos. Por vezes o MATLAB bloqueia e pode perder tudo. Vá guardando o trabalho frequentemente.
4. O novo modelo tem uma nova janela que é o seu espaço de trabalho. A construção do circuito que desejar é efetuada pela cópia (arrastamento com o rato) dos componentes pré-definidos a partir dos conjuntos (pastas) para esta nova janela.
5. Abra o conjunto "*Sources*" (com um duplo clique) para copiar um gerador de funções. Copie dessa janela para a do modelo (com o botão esquerdo do rato arrastando o ícone) o componente "*Signal Generator*". Mude-lhe o nome para "sinal".

   Em simulação de comportamentos em Telecomunicações é habitual usar-se uma função sinusoidal, para representar o sinal (a voz de uma pessoa, música, ou outro sinal que se pretenda estudar). Uma função sinusoidal tem uma frequência pura, e é periódica, ao contrário da nossa voz. No entanto, acaba por ser útil pois trabalhamos com sistemas lineares e assim estudamos o comportamento do sistema para aquela frequência. Convém nunca esquecer que essa função sinusoidal está a representar o sinal e não confundir com funções sinusoidais que são as portadoras.
6. Vá fechando as janelas dos conjuntos que for abrindo (neste caso a "*Sources*") para evitar ter um ambiente de trabalho com muitas janelas abertas.
7. Abra o componente "sinal" com um duplo clique no botão esquerdo do rato. Escolha a forma de onda sinusoidal, uma frequência de 1000 rad/s (**O MATLAB e o SIMULINK trabalham sempre em radianos por segundo**), e um pico/amplitude de 8. Relembrando, a relação entre radianos e Hertz é $\omega = 2\pi f$.
8. Proceda do mesmo modo que nos pontos 5 e 6 para copiar o componente "*scope*" do conjunto "*sinks*". O componente *scope* é um visualizador para se poder ver a forma de onda do gerador.
9. Mude-lhe o nome para "*vis 1*".
10. Agora ligue os dois componentes. Para isso ligue com o botão esquerdo premido a saída do "*sinal*" à entrada do "*vis 1*". Ficou com uma linha a ligá-los.

11. Na escolha "*parameters*" do menu "Simulation" parametrize a experiência que vai efetuar. Coloque o "Max Step Size" em 0.0001 e o "Min Step Size" em 0.00001. Estes parâmetros controlam a duração dos ciclos de avaliação dos componentes desde que começa a experiência até que termina.
12. Comece a experiência na escolha "Start" do menu "Simulation".
13. Abra o "vis 1" com um duplo clique para o calibrar. No "Vertical Range" mude o número para 10, pois o pico/amplitude do sinal é de oito. Depois de mudar o número, clique com o rato no espaço de "Horizontal Range" para o novo valor de "Vertical Range" fazer efeito. Proceda de um modo idêntico mudando o "Horizontal Range" para 0.1. Agora já pode visualizar melhor o sinal.
14. Termine a experiência na escolha "Stop" do menu "Simulation".
15. Para conhecer outro visualizador, copie o componente "Graph" do "sink", para perto do "vis 1". Mude-lhe o nome para "vis 2".
16. Para poder ver também o sinal no "vis 2" tem de o ligar. Para isso escolha qualquer ponto da linha que liga "sinal" a "vis 1" e, premindo o botão direito do rato, ligue esse ponto à entrada de "vis 2". Tem agora dois visualizadores para o "sinal".
17. Comece outra vez a experiência. A janela do "vis 2" aparece automaticamente, mas tem de a calibrar. Abra o "vis 2" e escolha, por exemplo, 0.01 para "Time Range", -10 para "y-min" e 10 para "y.max".
18. Termine a experiência na escolha "Stop" do menu "Simulation".
19. O terceiro visualizador que vai conhecer, chama-se "Auto-scale Graph" do conjunto "sink". Copie-o, mude-lhe o nome para "vis 3" e ligue o "sinal" também a ele.

Explique sucintamente o que observa no auto-scale e qual a diferença entre os visualizadores

_____

_____

_____

_____

_____

20. Termine a experiência na escolha "Stop" do menu "Simulation".
21. Carregando em "CTRL" e no botão direito do rato sobre um componente, copia esse componente. Copie o gerador de sinal "sinal" para outro local do modelo (por exemplo, por baixo de "sinal"). Mude-lhe a frequência para 500 rad/s e ponha os picos/amplitude dos dois sinais em 4.
22. Destrua todas as ligações que criou antes.
23. Agora copie um somador do conjunto "Linear". Ligue os dois geradores ao somador. Ligue o somador ao "vis 1".
24. Selecione com o rato os dois geradores e o somador. Escolha a seleção "Group" no menu das "options" para criar um bloco e simplificar o modelo. Mude o nome do bloco para "sinal T". Abra o bloco para ver o uso do componente "Out_1". De um modo idêntico poderia haver um componente "In_1" para sinais de entrada para o bloco.
25. Inicie a experiência e visualize a onda em "vis 1". Agora já se tem um sinal com duas frequências.
26. Termine a experiência na escolha "Stop" do menu "Simulation".
27. Ligue o "vis 2" e o "vis 3", inicie a experiência e visualize as ondas.
28. Termine a experiência na escolha "Stop" do menu "Simulation".
29. Guarde o modelo, e feche a janela.

Explique sucintamente porque é que a forma de onda é a que obteve

_____

_____

_____

_____

_____

_____

_____

Ponto 2 – Codificação binária, ternária e quaternária

Objetivo: Compreensão da codificação digital e da velocidade de transferência de informação.

Comentário: Os sinais do ponto 1 são analógicos. Hoje em dia a comunicação é essencialmente digital. A comunicação digital significa que o sinal só pode tomar um valor de um certo conjunto de valores discretos possíveis. Normalmente designa-se esse valor pela palavra **símbolo**.

Por exemplo, se for um sinal binário, existem apenas dois símbolos possíveis ("a" ou "b"). Neste caso podemos até falar de bits para os símbolos ("0" ou "1").
Se for um sinal ternário os símbolos só podem ter um de três valores ("a", "b", ou "c").
Se for um sinal quarternário os símbolos só podem ter um de três valores ("a", "b", "c", ou "d").

O **formato** dos símbolos pode variar: podem ser níveis de tensão, atrasos da portadora, valores de frequência, ou mais do que uma destas grandezas ao mesmo tempo. Durante a disciplina vão-se estudar ainda outros formatos.

A **velocidade de transferência de informação** é medida em símbolos por segundo (**símbolos/segundo**) e obviamente tem a ver com quantos símbolos se enviam por segundo. Um modo fácil de determinar é clacular o inverso do tempo que decorre entre o início de transmissão de um símbolo e o início de transmissão do símbolo seguinte. Por exemplo, se esse tempo for de 10 milissegundo, estamos a transmitir a 10 símbolos por segundo. Se durar 1 milissegundo estamos a transferir a 1000 símbolos por segundo.

Para concretizar, vamos considerar primeiro a codificação binária e quaternária, e deixar para o fim a ternária.

Exercício 1: Vamos querer transmitir mensagens/blocos e cada bloco pode ser um elemento de um conjunto (alfabeto) de 16 blocos possíveis: 0, 1, 2, 3, ... 15. A primeira pergunta é quantos símbolos são necessários para codificar cada bloco se usarmos uma codificação binária? _____.
E quantos símbolos são necessários para codificar cada bloco em quaternário? _____.

Exercício 2: Escreva então esses símbolos nas tabelas em baixo em que aparecem os 16 blocos do alfabeto.

Codificação binária

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	

Codificação quaternária

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	

Exercício 3: A relação entre quaternário e binário é passível de ainda outro pensamento: Cada vez que se transmite um símbolo quarternário quantos bits se estão realmente a transmitir nesse símbolo? _____.

Exercício 4: Considere agora os símbolos quaternários. Vão-se calcular várias durações e ritmos em símbolos por segundo (quaternários) e bits por segundo (os bits correspondentes aos quaternários) para se perceber a diferença. Preencha a seguinte tabela:

Símbolos por segundo	Duração de cada símbolo em segundos	Duração equivalente de cada bit, em segundos	Ritmo (símbolos por segundo)	Ritmo (bits por segundo)
3.000 (3 k)				
512 k				
1 M (Mega)				
1 G (Giga)				

Exercício 6: **Generalização**. Imagine que cada símbolo pode ser um de 1024 valores possíveis (do mesmo modo que um símbolo quaternário pode ser um de quatro valores possíveis). Preencha a tabela seguinte:

Símbolos por segundo	Duração de cada símbolo em segundos	Duração equivalente de cada bit, em segundos	Ritmo (símbolos por segundo)	Ritmo (bits por segundo)
3.000 (3 k)				

Exercício 7: **Caso ternário**. Considere agora que estamos a usar símbolos ternários e que o nosso alfabeto tem 27 blocos diferentes. Quantos símbolos ternários se têm de usar para codificar cada bloco? _____.

Ponto 3 – Uso de geradores de ondas específicas

Objetivo: Familiarização com o uso de geradores de sequências especiais. Vai-se começar com uma sequência estranha e depois com uma sequência binária.

Procedimentos:

1. Abra um novo modelo e guarde-o já na pendrive, seque.m.
2. Parametrize o "*Repeating Sequence*" com os seguintes valores

   [0 0.1 0.1 0.3 0.3 0.4 0.4 0.5 0.5] e
   [0 0 5 5 0 0 5 5 0]

Desenhe a onda que obteve:

3. Use o "*Repeating Sequence*" que produz ondas com quaisquer tipos de formas (oblíquos, ascendentes, planas, etc.). Parametrize-o para obter uma onda periódica, com a seguinte característica: subida de zero a um em 0.2 segundos, depois atingir o dez ao fim de 0.5 segundos e atingir o cinco ao fim de 0.9 segundos. Visualize a onda com um "*Graph*".

Escreva os valores dos parâmetros que escolheu para obter a onda que se pediu

_____

_____

4. Guarde o modelo.
5. Num novo modelo, defina uma sequência binária repetitiva com 5 Volt para o "1" e 0 Volt para o "0", de tal modo que a sequência seja "01001" e que o ritmo de transmissão seja de 5 símbolos por segundo (considere cada bit como um símbolo).
6. Desenhe com indicação da escala a forma de onda obtida para uma sequência

7. Guarde o modelo.

Ponto 4 –Integral de um sinal, e uso do componente de ganho

Objetivo: Familiarização com a função "integração" de um sinal e com algumas das facilidades que ela permite. O integrador pode ser muito importante nos recetores de comunicação digital, como se irá ver. Na parte analógica, o integrador tem um efeito de um filtro passa-baixo (também como se irá ver num trabalho futuro).

Procedimentos:

1. Abra um novo modelo, integ.m, e guarde-o já na pendrive.
2. Copie o último gerador da sequência binária do ponto anterior. Imagine que esse sinal é o sinal que o recetor recebe da linha. Vamos agora tentar fazer um recetor.

   Um recetor recebe o sinal muito atenuado e, para além disso, tem de saber quando começa um símbolo e quando acaba. No momento de acabar tem de decidir que símbolo recebeu. No nosso caso, o sinal que chega ao recetor não está atenuado, e então fica-se "apenas" com o problema de quando um símbolo começa e acaba.
3. Copie o componente "*Integrator*" do conjunto "Linear". Ligue o sinal à entrada do "*Integrator*".
4. Para poder ver os dois sinais ao mesmo tempo tem de os multiplexar. Copie um "*Mux*" de "*connections*" e parametrize-o para ter apenas duas entradas. Ligue a saída do integrador à primeira entrada do "*Mux*" e a saída do sinal à outra entrada. Ligue a sua saída a um "*Graph*" com os valores de 1 para "Time Range" e –5 e 5 para os eixos.
5. Parametrize a experiência com os valores para *Max Step Size* e *Min Step Size* do ponto 1.
6. Inicie a experiência.

Explique sucintamente o que observa

_____

_____

_____

_____

_____

7. Termine a experiência.
8. Realmente o que se observa no ponto anterior é muito ténue. Seria bom realçar um pouco mais. Para isso, poderemos amplificar o sinal. Atenção que amplificação no recetor é sempre à custa de mais energia. Se fosse um telemóvel isso teria implicações na duração da bateria… Copie um amplificador ("*Gain*" do conjunto "*Linear*") e coloque-o entre a saída do integrador e o "*Mux*". Abra o "*Gain*" e ponha-lhe um ganho de 5.

Explique sucintamente o que observa

_____

_____

_____

_____

_____

9. Guarde o modelo.

PARTE II

No ponto anterior, cada vez que há um símbolo a "1" a saída do integrador vai crescendo (calculando a área). Seria bom que no início de um novo símbolo a saída do integrador voltasse a zero.
Mas como se sabe quando um símbolo acaba? O recetor tem apenas o sinal de entrada e não tem outra linha a dar a sincronização… Tem de usar esse sinal de entrada para tudo o que quiser.
Como estamos no início da disciplina vamos fazer uma solução muito simples. Vamos aproveitar o sinal de dois modos: a) Integrando para saber se foi transmitido um "0" ou um "1"; e b) usá-lo também para saber os momentos em que os símbolos acabam ou começam.
Para o primeiro modo usa-se o integrador. Para o segundo modo vamos usar um derivador. Vamos derivar o sinal e a saída do derivador vai ser não nula nos instantes de transição do sinal. Repare que esta solução é manifestamente insuficiente para realizar uma decisão correta. Vai-se começar por este segundo modo e depois vamos alterar o ponto 4 para fazer o primeiro modo.

Ponto 5 – Derivada de um sinal, uso do componente *Saturation*, do componente *Transport Delay* e do componente *Abs*

Objetivo: Familiarização com a função "derivada" de um sinal e com algumas das facilidades que ela permite. Familiarização com o uso dos componentes *Saturation*, *Transport Delay* e *Abs*.

Procedimentos:
1. Abra o modelo integ.m e guarde-o com outro nome, deriv.m.
2. Copie o componente "*Derivative*" de "*Linear*", e ligue à sua entrada o sinal (o sinal fica assim ligado ao integrador, ao diferenciador e ao "*Mux*").
3. Mude o "*Mux*" para três entradas e ligue a saída do diferenciador.
4. Comece a experiência e visualize os três sinais.

Explique sucintamente o que observa

_____

_____

_____

_____

_____

5. Não interessa ter "picos" para cima e para baixo. É melhor ter "picos" todos na mesmo sentido. Use um "*Abs*" do conjunto "*nonlinear*" na saída do diferenciador. Visualize os sinais.
6. Também não interessa ter "picos" tão grandes. Aplique o componente "*Saturation*" do conjunto "*Nonlinear*" para limitar o os picos a 5 e 0. Visualize os sinais.
7. Os "picos" marcam os instantes em que cada símbolo acabou. Poderia ser interessante ter esses picos um pouco depois do final de cada símbolo. Para isso atrase o sinal de 25 mseg. Para isso use o componente "*Transport Delay*" do conjunto "*nonlinear*". Visualize os três sinais ao mesmo tempo.
8. O ponto anterior serviu apenas para mostrar como o componente "*Transport Delay*" funciona. Como o atraso não é realmente necessário volte a retirar o atraso.
9. Guarde o modelo

Como vê, só se tem "picos" quando o sinal muda. Ora, quando são transmitidos dois símbolos iguais não temos um "pico" no início do segundo símbolo.
Considere que tinha um circuito que lhe dava "picos" no início de cada símbolo (em vez deste circuito simples com o derivador que tem esta limitação).
Mesmo com isso, existe ainda o problema da saída do integrador não ir a zero no final de cada símbolo. Mas, mesmo assim, isto é, com este integrador, consegue descrever um "algoritmo" que consiga determinar o símbolo que acabou de ser transmitido?

_____

_____

_____

_____

_____

_____

_____

_____

Ponto 6 – Uso dos componentes "*Reset Integrator*" e "*Relay*"

Objetivo: Familiarização com o funcionamento dos componentes *Reset Integrator* e *Relay*.

Procedimentos:

1. O integrador que usámos vai sempre integrando e não volta a zero. Queremos um que volte a zero no final do símbolo. Para isso substitua o integrador por um "*Reset Integrator*" do bloco "*nonlinear*". Este integrador integra a entrada sempre que a entrada de controlo (a segunda) é zero. Quando a entrada de controlo for diferente de zero ele põe à saída o valor da terceira entrada.
2. Para o sinal de controlo podemos usar a saída do diferenciador. Para o valor da terceira entrada use o componente *Constant* do bloco *Sources* com o valor zero. Na prática, em eletrónica, ia-se buscar a terra do circuito. Visualize os sinais.
3. Os sinais já começam a parecer melhor! Convém agora ter um sinal quadrado e não um sinal triangular. Para isso use o componente *Relay*. O componente *Relay* trabalha do seguinte modo: quando o sinal de entrada passa <u>para cima</u> do valor de <u>on</u>, ele põe como saída o valor que escolhermos para a tensão em <u>on</u>. Quando passa <u>para baixo</u> do valor de <u>off</u>, ele põe na saída o valor que escolhemos para a tensão de <u>off</u>. Os valores de <u>on</u> e de <u>off</u> não têm necessariamente de ser iguais. Coloque o valor de <u>on</u> em 0.6 e o de <u>off</u> em 0.2 e as tensões de <u>on</u> em 3 e de <u>off</u> em 0 Volt.
4. Já temos uns pequenos instantes onde podemos ver que se o sinal for maior do que um certo valor (o valor que decidimos para a saída do *Relay*) temos o símbolo "1" e se for menor temos o símbolo "0".
5. Mude agora a sequência de "01001" para "01010".
6. Visualize as ondas.

Comente o que observa e porque é que neste caso o sincronismo já deteta todos os símbolos. Mas é evidente que as sequências não são sempre "01010"…

Consegue pensar num outro formato de símbolo que ajude no sincronismo, nomeadamente, ajude a gerar "picos", para depois o recetor decidir sobre o que recebeu (isto é, sem estar dependente da sequência da mensagem)?

_____

_____

_____

_____

_____

7. Mude o valor de <u>on</u> do *Relay* de 0.6 para 1. Que consequências isso traz para o circuito do recetor?

_____

_____

_____

_____

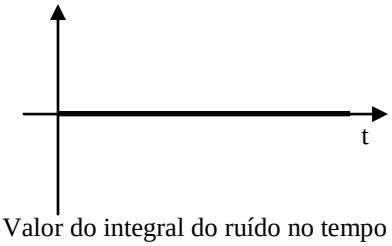
8. Volte a colocar o valor inicial. Guarde o modelo.

Ponto 7 – Comportamento com ruído

Objetivo: Perceber o uso do integrador como elemento redutor do efeito do ruído.

Comentário: O uso do integrador pode parecer uma demasiada sofisticação do recetor para se determinarem os vários símbolos. Porque é que não medimos simplesmente a onda de entrada para ver se ela tem tensão acima ou abaixo do valor médio dos dois valores de cada símbolo?
Uma das razões que impede que isso funcione é a existência de ruído no canal de comunicação. Na altura da decisão o ruído pode fazer com que a decisão seja errada. Normalmente considera-se em telecomunicações que o ruído é aleatório e que tem média nula. Isto é, o seu integral é nulo, como está mostrado na figura abaixo. O integrador tem assim o efeito de anular a perturbação causada pelo ruído.

Valor do ruído no tempo



Valor do integral do ruído no tempo

Procedimentos:
1. Use o componente "*Random Number*" do componente *Sources* para gerar o ruído. Para se ter alguma potência, coloque um amplificador à saída deste componente com um ganho de 2. Note que estamos a exagerar bastante.
2. Aumente em um a entrada do "*Mux*" para poder visualizar o ruído e os nossos sinais. Visualize-os.
3. Como vê, um ruído como este pode estragar qualquer momento de decisão.
4. Volte a desligar o ruído do "*Mux*" pois interessa adicioná-lo ao sinal e não vê-lo sozinho.
5. Adicione então a saída do amplificador com ruído ao sinal com um "*Sum*" do bloco *linear*. Isto é, adicionou-se o ruído ao sinal no canal.
   No entanto, vamos só fazer uma simplificação um pouco exagerada.
   > Vamos usar o sinal ainda sem ruído para a entrada do derivador. Claro que isto é impossível na realidade. Fazemos isto para que o derivador não comece a gerar "picos" por tudo e por nada…

   Na realidade os circuitos de sincronismo não são feitos à custa de derivadores…
6. Visualize os sinais todos e depois retire do "*Mux*" o sinal para poder visualizar os outros com mais clareza.
7. Guarde o modelo

Comente o que obteve

_____

_____

_____

_____

_____

_____