

Introdução às Telecomunicações

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

Grupo: ____ nº ____ e ____

1º Trabalho de Laboratório

Objectivo 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 efectuada numa aula.

PARTE I

Ponto 1 - Visualização de um sinal periódico

Objectivo: 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 electrónicos para executar o que se pretenda. O SIMULINK arranca com uma janela onde existem conjuntos de componentes pré-definidos.
3. Nesta nova janela, abra um novo modelo (menu “file”) e guarde-o logo com um nome (por exemplo, *senal.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 efectuada pela cópia (arrastamento com o rato) de componentes pré-definidos a partir dos conjuntos da janela “simulink” para o seu modelo.
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 “senal”.

Em simulação de comportamentos em Telecomunicações é habitual usar-se um sinal sinusoidal, para representar a voz de uma pessoa, ou outro sinal que se pretenda estudar. Um sinal sinusoidal tem uma frequência pura, e é periódico, ao contrário da nossa voz. No entanto, acaba por ser uma boa aproximação pois trabalhamos com sistemas lineares e assim estudamos o comportamento dos circuitos para aquela frequência. Convém nunca esquecer que esse sinal está apenas a representar a nossa voz e não teria de ser sinusoidal, ou periódico.

6. Feche a janela “Sources” para evitar ter um ambiente de trabalho com muitas janelas abertas.
7. Abra o componente “senal” 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 podermos ver a forma de onda do gerador.
9. Mude-lhe o nome para “vis I”.
10. Agora ligue os dois componentes. Para isso ligue com o botão esquerdo premido a saída do “senal” à entrada do “vis I”. Ficou com uma linha a ligá-los.

11. Na escolha “parameters” do menu “Simulation” parametrize a experiência que vai efectuar. 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. É preciso, agora, calibrá-lo. 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 ver o sinal de um modo mais visível.
14. Termine a experiência na escolha “Stop” do menu “simulation”
15. Copie agora 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” com o botão direito do rato e ligue esse ponto à entrada de “vis 2”, sempre com o botão carregado. Tem agora dois visualizadores para o “sinal”.
17. Comece outra vez a experiência. Automaticamente aparece-lhe a janela do “vis 2”. Tem, agora, de calibrá-la. 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.
19. Copie agora o componente “Auto-scale Graph” do conjunto “sink”, mude-lhe o nome para “vis 3” e ligue-o também ao “sinal”.

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

20. Termine a experiência.
21. Carregando em “CTRL” e no botão direito do rato sobre o gerador de sinal “sinal”, copie-o para outro local do modelo (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.
23. Agora copie um somador do conjunto “Linear”. Ligue os dois geradores ao somador. Ligue o somador ao “vis 1”.
24. Seleccione com o rato os dois geradores e o somador. Escolha a selecção “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.
27. Ligue o “vis 2” e o “vis 3”, inicie a experiência e visualize as ondas.
28. Termine a experiência.
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

Objectivo: Utilização de codificação digital.

Comentário: Os sinais do ponto 1 são analógicos. Hoje em dia a comunicação é essencialmente digital. A comunicação digital significa que o valor do sinal só pode tomar um certo número de valores discretos. Para concretizar, vamos imaginar que pode tomar 16 valores possíveis: 0, 1, 2, 3, ... 15. O problema agora é como se vai enviar o valor que se tem para o receptor. Neste ponto vamos ver os casos de comunicação binária, ternária e quaternária.

Se tivermos comunicação binária podemos usar um de dois símbolos (neste caso bits) (“0” ou “1”). Para conseguirmos enviar o valor temos de transmitir esses símbolos em série uns atrás de outros (neste caso, precisamos de 4 bits para poder codificar todos os valores $2^4=16$).

Se tivermos comunicação ternária, quer dizer que temos símbolos ternários (por exemplo, “a”, “b” e “c”). Isto é, o receptor em vez de distinguir entre o 1 e 0, consegue distinguir um de três símbolos. Num conjunto de nove valores do sinal, em vez dos 16 exemplificados acima (0, 1, 2, ... , 8) necessitaríamos de dois símbolos para codificar todos os valores $3^2=9$.

Se tivermos comunicação quaternária, então o receptor consegue distinguir um de quatro símbolos diferentes (“a”, “b”, “c” e “d”). Para os dezasseis níveis de tensão bastariam dois símbolos quaternários $4^2=16$. Outro modo de pensar para este caso é que cada símbolo quaternário pode ser representado por dois símbolos binários (bits) $4^2=2^4=16$ (normalmente até se chama dibits aos símbolos quaternários).

Exercício: Para os seguintes conjuntos de valores possíveis do sinal indique codificações de símbolos para os codificar.

Codificação binária

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

Codificação ternária

0	
1	
2	
3	
4	
5	
6	
7	
8	

Codificação quaternária

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

O problema seguinte é como se representam os símbolos em valores de tensão na linha. Isto é, que tensão na linha deve ter o símbolo “0”? A resposta é complexa e existem várias maneiras. A esta representação chama-se **código de linha**. Iremos estudar isto na disciplina. Por agora, e para símbolos binários, poderemos considerar que o “1” pode ser representado por 5 Volt durante o tempo todo que deve durar o símbolo, e o “0” por 0 Volt durante o tempo todo que deve durar o símbolo. ... E quanto deve durar o símbolo? Se tivermos um ritmo na linha de 1 símbolo por segundo ele deve durar 1 segundo. Se tivermos um ritmo de 50 símbolos por segundo, ele deve durar 20 mseg. Indique a duração de cada símbolo para os seguintes ritmos de linha:

3000 K símbolos por segundo _____ . 512 K símbolos por segundo _____ .
 1000 M símbolos por segundo _____ . 1 G símbolos por segundo _____ .

Ponto 3 – Uso de geradores de ondas específicas

Objectivo: 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. 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 os 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

3. 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:

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

Objectivo: Familiarização com a função “integração” de um sinal e com algumas das facilidades que ela permite. O integrador pode ser muito importante nos receptores de comunicação digital, como se irá ver. Na parte analógica, o integrador tem um efeito de um filtro passa-baixo, 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 receptor recebe da linha. Vamos agora tentar fazer um receptor.
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 receptor é sempre à custa de mais energia. Se fosse um telemóvel as pilhas durarão tão menos tempo quanto maior for a amplificação necessária. 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

Cada vez que há um símbolo, a saída do integrador vai crescendo (calculando a área) e seria bom que em cada símbolo novo a saída do integrador voltasse a zero. Como sabemos, então, no receptor quando um símbolo acaba? Podemos fazer um circuito muito simples que indique isso. Na realidade as coisas são um pouco mais complicadas, mas como estamos no início da disciplina temos de viver com aquilo que sabemos. Para além disso, o interesse deste trabalho é a familiarização com o simulink. Vamos aproveitar o sinal de dois modos: Integrando para saber se foi transmitido um “0” ou um “1”; e usá-lo também para saber os momentos em que os símbolos acabam. A associação de um derivador e de um integrador permite detectar os instantes em que os bits começam e acabam e o tipo de bits transmitidos. Neste caso o derivador vai ter respostas não nulas nos instantes de transição entre bits, informação manifestamente insuficiente para realizar uma decisão correcta quanto ao bit transmitido. A junção de um integrador cujo valor da resposta não é nulo somente na presença de potência, permitirá definir um valor adicional essencial para a decisão de bit. Começemos pelo primeiro modo.

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

Objectivo: 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 mal ele entre no receptor (o sinal fica assim ligado ao integrador, ao diferenciador e ao “*Mux*” para se ver).
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. Seria melhor ter “picos” todos na mesmo sentido. Use um “*Abs*” do conjunto “*nonlinear*” à saída do diferenciador. Visualize os sinais.
6. Também não interessa ter “picos” tão grandes. Aplique o componente “*Saturation*” do conjunto “*Nonlinear*” para limitar o sinal derivado 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. Este atraso não é, de facto, necessário agora. Volte a retirar o atraso.
9. Guarde o modelo

Como vê, só se tem “picos” quando o sinal muda. Seria bom ter picos sempre depois do tempo de um símbolo para forçar a saída do integrador a zero e depois medir o que ele produziu ao fim do novo símbolo. Imagine que o circuito de sincronismo era perfeito e que tinha a indicação de quando o símbolo acabava. Neste caso, diga se mesmo assim se consegue fazer um receptor e invente um “algoritmo” para que o receptor trabalhe com exactidão.

Ponto 6 – Uso dos componentes “Reset Integrator” e “Relay”

Objectivo: 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. Ora a saída do diferenciador tem esse comportamento. Para o valor da terceira entrada use o componente *Constant* do bloco *Sources* com o valor zero. Na prática, em electrónica, ia-se buscar a terra do circuito, por exemplo. 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 para cima do valor de on, ele põe como saída o valor que escolhermos para a tensão em on. Quando passa para baixo do valor de off, ele põe na saída o valor que escolhemos para a tensão de off. Os valores de on e de off não têm necessariamente de ser iguais. Coloque o valor de on em 0.6 e o de off em 0.2 e as tensões de on em 3 e de off 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 detecta todos os símbolos. Invente uma forma de representar um símbolo que faça com que exista sempre sincronismo qualquer que seja a sequência de símbolos.

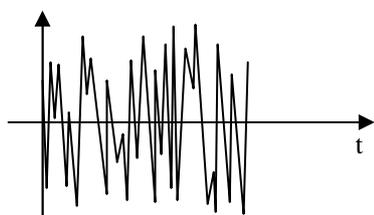
7. Mude o valor de on do *Relay* de 0.6 para 1. Que consequências isso traz para o circuito do receptor?
-
-
-
-
-
-

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

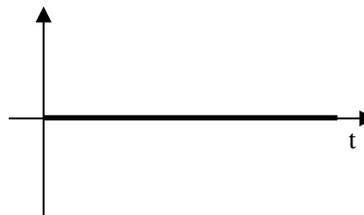
Ponto 7 – Comportamento com ruído

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

Comentário: Tudo isto pode parecer um disparate. Porque é que não medimos simplesmente a onda de entrada para ver se ela tem tensão acima de zero (ou de metade do valor máximo do símbolo) ou não? Ora o que acontece é que existe ruído nas linhas de telecomunicações (ou no espaço no caso de ondas rádio). Na altura da decisão o ruído pode fazer com que a decisão seja mal feita. 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. Para se colocar ruído use o componente “*Random Number*” do componente *Sources*. Para se ter alguma potência, coloque um amplificador à saída deste componente com um ganho de 2.
2. Aumente em um a entrada do “*Mux*” para poder visualizar o ruído e os nossos sinais. Visualize-os.
3. Como vê, ruído deste 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*. Vai-se só fazer um pequeno truque por este ser o primeiro trabalho: adiciona-se depois de se tirar o sinal para o componente diferenciador. A soma do ruído iria estragar os “picos” de sincronização que tanto trabalho nos deu a obter. Na realidade os circuitos de sincronismo não são feitos à custa de diferenciadores.
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
