Instituto Superior Técnico

MEAer

Controlo por Computador

Relatório: Identificação do modelo

Trabalho realizado por:	*Número:*
Alice Lourenço	86606
Diogo Janeiro	86623
Mariana Tavares	86664

Turno 2ª feira

2019/2020

Conteúdo

1 Introdução

O trabalho laboratorial da cadeira de Controlo por Computador tem como objetivo a identificação de um modelo e o desenho de um controlador para a posição da ponta de uma barra flexível. O controlo é feito a partir de um computador que altera o comportamento da barra consoante o algoritmo implementado. Neste relatório é feita a análise da primeira parte do trabalho laboratorial, que corresponde à identificação do modelo.

Esta tarefa não é trivial (**Q1**), como explicaremos na subsecção 2.1. Começaremos então por calibrar os sensores, por forma a conseguir recolher os dados da posição da ponta da barra flexível α e θ (secção 3, **Q2**). No sentido de desenhar um controlador, é primeiro necessário identificar um modelo do sistema a controlar (secção 4, **Q3**).

No final deste trabalho, a informação dada pelos sensores será convertida de sinal analógico para digital e fornecida ao computador. Este, por cada período de amostragem, aplicará o algoritmo desenvolvido para calcular a tensão a aplicar ao motor. O resultado calculado pelo controlador passará então pelo conversor digital-analógico e esse sinal, por sua vez, é amplificado pelo amplificador de potência para comandar o motor.

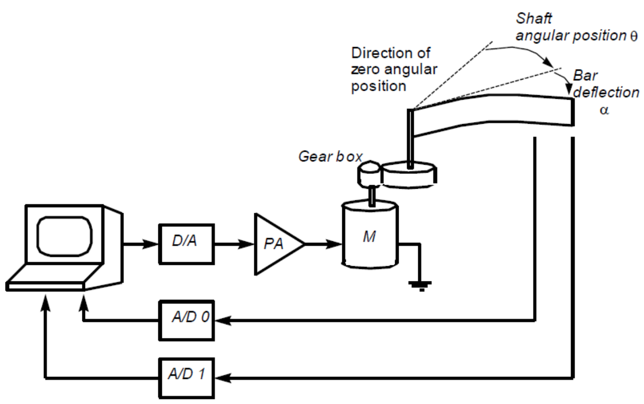

Figura 1: *Desenho esquemático da ligação do hardware ao computador; o computador está ligado à instalação através de conversores AD e DA.*

Ao longo deste trabalho, sempre que forem referidos ficheiros do *Matlab*, o respetivo *sketch* encontra-se nos anexos, secção 6.

2 Instalação

A instalação a controlar (figura 2) é constituída por uma barra flexível ligada ao veio de um motor DC, que está por sua vez ligado a um amplificador de tensão. Esta instalação possui ainda um extensómetro (figura 3) para medir o ângulo de deflexão da barra (α) e um potenciómetro, que se encontra fixo ao veio do motor, para medir o ângulo de rotação do veio (θ). A posição da ponta da barra (y), que é o que se pretende controlar, é dada pela soma destes dois ângulos:

$$y = \alpha + \theta \tag{1}$$

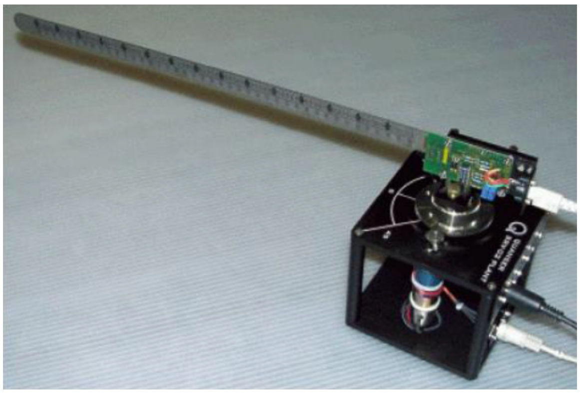

Figura 2: *Instalação a controlar.*

Figura 3: *Localização do extensómetro.*

2.1 Q1

Existem várias razões que fazem com que este controlo não seja trivial. Em primeiro lugar, não é possível criar uma tabela de tensões que se apliquem ao motor de modo a obter um ângulo específico. Isto acontece porque ao aplicar uma voltagem constante ao motor isto traduz-se numa velocidade angular constante e não numa posição angular constante, uma vez que esta corresponde à integração da velocidade angular. Aplicar um sinal constante à entrada, o que se traduz numa velocidade constante, durante um período finito de tempo faria o y ir para a posição desejada. No entanto, esta opção de controlo não é viável porque a barra é flexível, pelo que a posição da ponta da barra não coincide com a posição do veio do motor. O facto da barra ser flexível é também a explicação para um simples controlo por *feedback* não funcionar.

Tem-se assim o segundo problema, o facto da barra ser flexível. Isto implica que existam fenómenos como o efeito chicotada e o efeito de arrastamento que precisam de ser considerados. O efeito chicotada, amplamente debatido ao longo do relatório, é o nome dado quando a rotação do veio do motor muda de sentido e a ponta da barra não reage instantaneamente, tendo tendência a continuar o movimento que tinha anteriormente.

Para estimar aquilo que se passará com o modelo é necessário ter em conta a dinâmica do motor e da barra. Simular a dinâmica do motor é relativamente fácil, uma vez que bastará adicionar um integrador, ou seja, terá que haver um pólo na origem. Adicionar a dinâmica da barra é mais difícil uma vez que, para além de não ser conhecida, engloba mais fatores.

Para representar o comportamento oscilatório da ponta da barra, teremos de inserir pares de pólos conjugados. Por uma questão de simplificação, assume-se apenas um par. Terá ainda de ser considerado o efeito de chicotada, introduzindo um zero no semiplano complexo direito, já que é um elemento que provoca instabilidade no sistema.

$$H(s) = \frac{s - c_1}{s(s^2 + c_2 s + c_3)} \tag{2}$$

3 Calibração dos sensores

Como já referido na secção anterior, a instalação possui dois sensores, um extensómetro e um potenciómetro. Nesta secção pretende-se descobrir as constantes que estabelecem a relação entre o valor de tensão debitado por estes sensores e a grandeza que eles estão a medir. Assim, pretende-se relacionar a tensão medida pelo extensómetro com o ângulo da ponta da barra em relação à sua posição em repouso (α) e a tensão medida pelo potenciómetro com o ângulo do veio do motor (θ). A representação esquemática destes ângulos está representada na figura 4.

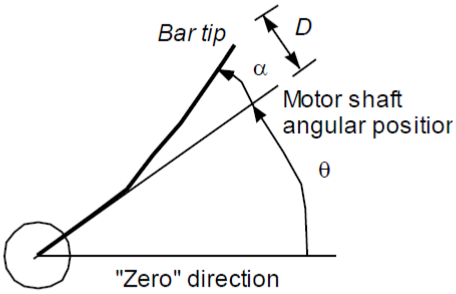

Figura 4: *Esquema da deflexão da barra flexível.*

De modo a conseguir obter estas constantes foi necessário criar um modelo em *simulink* para poder aplicar estímulos à barra e ler os valores dos sensores correspondentes.

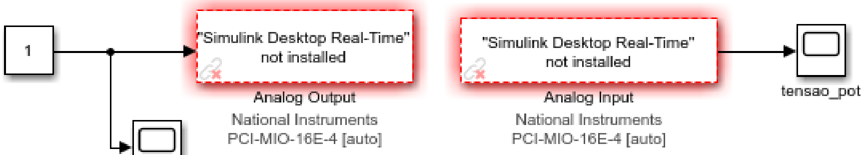

Figura 5: *Diagrama de blocos do Simulink utilizado para a calibração dos sensores.*

Para a realização do ensaio do potenciómetro, foi utilizado o diagrama tal como se encontra representado na figura 5. Para a calibração do extensómetro foi utilizada unicamente a parte correspondente à leitura de dados (direita), uma vez que o input era realizado manualmente, tal explicado na secção 3.1. De salientar que para a realização deste ensaio o input foi posto a 0 (em vez do 1 que aparece no diagrama) para que o motor a rodar não afetasse a leitura do extensómetro (como se verificou que acontecia experimentalmente).

Ao longo de toda esta secção responde-se à pergunta **Q2** do guia laboratorial.

3.1 Medição de α

O **extensómetro** é constituído por uma resistência elétrica cujo valor varia consoante o seu comprimento, pelo que se houver deformações na barra este valor vai variar. A tensão debitada pelo extensómetro é assim proporcional à sua deformação e portanto à deformação da barra, uma vez que este se encontra colado à sua superfície. Isto implica que a relação entre α e a tensão dada pelo sensor seja dada pela equação 3, em que $\mathrm{K_b}$ é o valor da constante que se pretende descobrir e α_e é o valor de tensão medido pelo extensómetro.

$$\alpha = K_b \alpha_e \tag{3}$$

Para obter o valor de $\mathrm{K_b}$ foi necessário defletir a barra com um conjunto de ângulos determinados.

Isto foi feito com a ajuda de um aparelho próprio para esse fim, que tem uma série de dentes separados por 0.64cm que formam uma espécie de pente (figura 6).

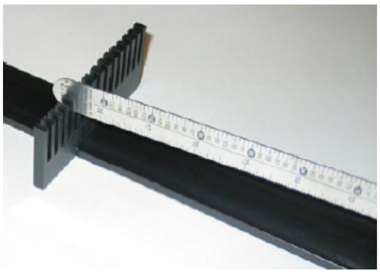

Figura 6: *Aparelho utilizado para calibrar o extensómetro.*

Uma vez que o que é conhecido é o valor da distância entre os dentes e não o ângulo, torna-se necessário descobrir qual é o ângulo associado a cada posição. Observando a figura 4, e chamando L ao comprimento da barra, torna-se fácil chegar à seguinte relação:

$$tg(\alpha) = \frac{D}{L}$$

Utilizando esta relação obtém-se o valor em radianos do ângulo pretendido, mas como é preferível trabalhar com valores em graus é necessário fazer ainda a seguinte conversão:

$$\alpha_{[graus]} = \frac{180°}{\pi}\alpha_{[rad]}$$

Em laboratório, foi medido o comprimento da barra (L = 37.5 cm) e foram medidos os valores de tensão para as várias posições da barra ao longo do "pente", tendo sido obtidos os seguintes resultados:

Tabela 1: *Resultados experimentais.*

Valor tensão (V)	D (cm)	α (graus)
1.71	-5.7150	-8.6652
1.38	-5.0800	-7.7147
1.14	-4.4450	-6.7599
0.84	-3.8100	-5.8013
0.56	-3.1750	-4.8395
0.28	-2.5400	-3.8749
-0.01	-1.9050	-2.9081
-0.33	-1.2700	-1.9397
-0.66	-0.6350	-0.9701
-0.97	0	0
-1.20	0.6350	0.9701
-1.51	1.2700	1.9397
-1.76	1.9050	2.9081
-2.05	2.5400	3.8749
-2.32	3.1750	4.8395
-2.62	3.8100	5.8013
-2.91	4.4450	6.7599

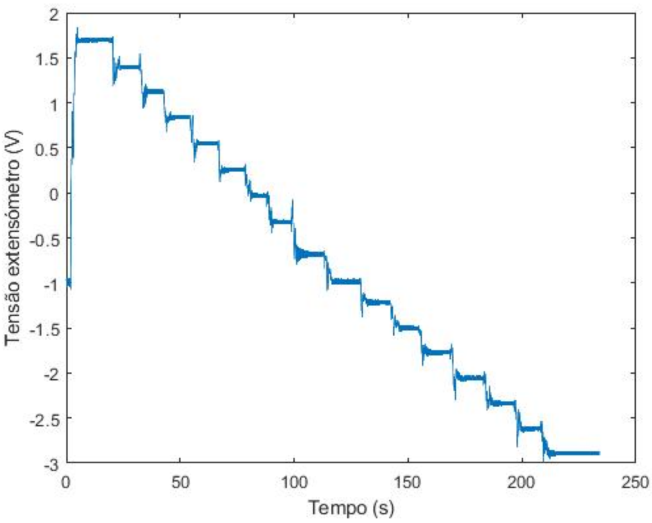

Figura 7: *Plot do valor de tensão debitado pelo extensómetro em função do tempo aquando da realização do ensaio para obtenção de K_b; a barra flexível foi deformada de acordo com os valores da tabela 1.*

Estes resultados experimentais foram obtidos retirando, em cada patamar, o ponto para o maior valor de tempo, para o qual se admite que a leitura já estabilizou o suficiente para se poder fazer esta leitura sem afetar a exatidão do resultado.

Numa situação ideal a relação 3 corresponderia à realidade, no entanto, isto não é verdade pois a relação não é exatamente linear. Assim, K_b corresponde a um comportamento médio e é obtido aplicando o método dos mínimos quadrados aos dados obtidos segundo o procedimento descrito em cima.

Executando o ficheiro de código de Matlab "extensometro.m"(secção 6.1), obteve-se que K_b = -3.3434 $°/V$. O ficheiro calcula o α para todas as medições feitas e depois utiliza a função *polyfit()*, que devolve a melhor relação linear (grau 1, terceira entrada da função), usando o método dos mínimos quadrados. Na figura 8 pode observar-se a relação entre os dados experimentais e a relação linear obtida com a função *polyfit()*.

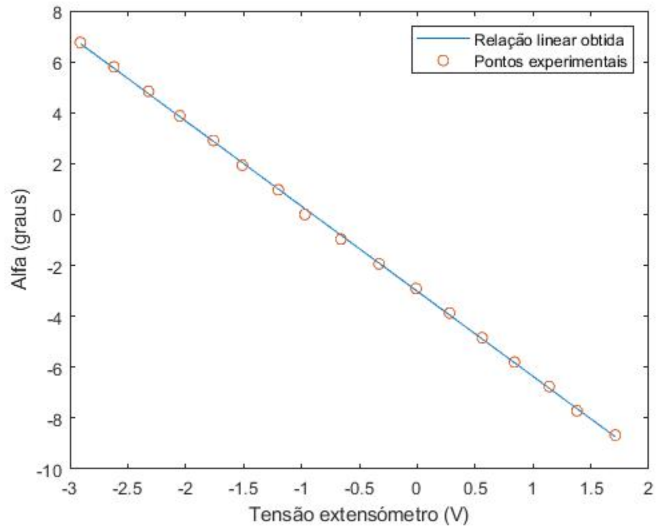

Figura 8: *Comparação entre os dados experimentais obtidos para a calibração do extensómetro e a relação linear obtida para esses mesmos dados usando a função polyfit() do Matlab.*

3.2 Medição de θ

O **potenciómetro** é constituído por um cursor que roda sobre uma ferradura, como na figura 9, de forma a alterar a resistência aos seus terminais. Este cursor está solidário com o veio do motor, sendo a tensão debitada θ_e proporcional ao ângulo θ admitido (equação 4).

$$\theta = K_p\theta_e \tag{4}$$

Para determinar a constante K_p, aplica-se um escalão de 1V ao motor e analisa-se o declive do gráfico resultante, notando que é fácil identificar os intervalos de tempo correspondentes a 360°através das descontinuidades observadas (figura 10).

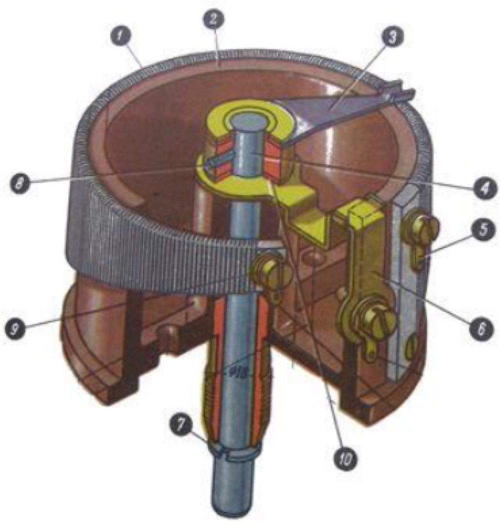

Figura 9: *Ilustração do funcionamento interno de um potenciómetro.*

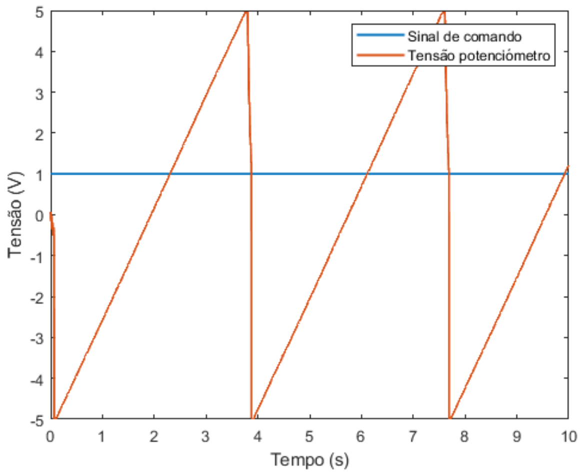

Figura 10: *Plot do sinal de entrada e de saída do potenciómetro aquando da realização do ensaio para obtenção da contante K_p.*

Estas decontinuidades devem-se ao facto de não haver um obstáculo à passagem entre o fim e o início da ferradura, provocando uma queda abrupta da tensão debitada θ_e aproximadamente de +5V para -5V.

Tem-se assim $K_p = \frac{360}{4.956-(-4.985)} = 36.2137°/V$ (anexo da secção 6.2).

As descontinuidades mostraram-se um problema no tratamento dos dados adiante. Neste sentido, experimentou-se ainda outro método de medição do ângulo θ: **fotodetetores**, que consiste na contagem da passagem por cada dente da engrenagem (1024 no total). Desta forma, já não é necessário ter o cuidado de garantir que não se passa a fronteira 0°/360°. Para este método tem-se $K_p = -(\frac{360}{1024})°$.

4 Obtenção do modelo

Passa-se agora à identificação do modelo a ser usado posteriormente para o desenho do controlador. Procede-se portanto à:

- Recolha dos dados
- Processamento dos dados
- Seleção do melhor modelo

4.1 Recolha dos dados Experimentais

Um dos cuidados a ter na recolha de dados é a escolha de sinais de entrada adequados. Optou-se pela utilização de sinais de onda quadrada e sinais PRBS (pseudorandom binary sequence).

É necessário ter em atenção o valor da **amplitude** do sinal injetado. Se o valor imposto à amplitude for demasiado baixo, a folga entre os dentes das engrenagens da caixa de desmultiplicação não permitirá o movimento do veio de acordo com o comando dado ao motor. Se o valor for muito alto, verificar-se-á um comportamento não linear da barra, que não é pretendido pois não está a ser estudado como tal. Neste sentido aplicámos em todas as experiências 1V à amplitude do sinal de entrada do motor.

Quanto à **frequência**, se o sinal for demasiado rápido, de forma a que o sistema não se encontre na gama de frequências em que a sua dinâmica é dominante, este vai apenas filtrar o sinal de entrada, não apresentando o comportamento pretendido. Por outro lado, se o sinal for muito lento, não permitirá que os transientes sejam excitados. Neste sentido, optou-se por valores 0.5 Hz para a frequência da onda quadrada, e 0.10Hz e 0.11Hz para a largura de banda do sinal PRBS.

O primeiros 10 segundos de cada experiência foram excluídos, de forma a garantir que só se consideram dados após estar estabelecido o comportamento do sistema a estudar. Tenta-se ainda obter experiências com uma **duração** mínima de 60s.

Admitiu-se o valor de 0.02s para o **período de amostragem**. Se este valor tomasse valores demasiado grandes, não seriam recolhidos dados suficientes para retratar o comportamento do sistema. Se, pelo contrário, tomar valores demasiado pequenos, os dados excessivos não seriam representativos porque o sistema não teria tempo de responder ao estímulo dado, pelo que se estaría a tentar modelar o que, no fundo, seria apenas ruído.

Tabela 2: *Conjutos de dados experimentais utilizados.*

Sinal	Frequência (Hz)	Largura de Banda (Hz)
prbs3	—	0.10
prbs4	—	0.10
prbs6	—	0.10
prbs8dentes	—	0.10
prbs9dentes	—	0.11
prbs11dentes	—	0.10
square3	0.5	—

Os nomes dos sinais na tabela 4.1 referem-se aos conjuntos de dados utilizados no código

4.2 Processamento dos Dados

Em laboratório, foram realizadas várias experiências com as características definidas acima sendo que os dados recolhidos correspondentes à resposta do sistema foram guardados em vários ficheiros do *Matlab*. São estes ficheiros que são analisados nesta secção.

Os dados recolhidos estão guardados na variável *tensao_pot.signals.values* da seguinte forma:

- 1ª coluna: output potenciómetro θ_e
- 2ª coluna: output extensómetro α_e

Só a meio do processo de recolha de dados foi disponibilizada a ferramenta de contagem através dos fotodetetores, pelo que, desde então, passou-se a guardar também os dados recolhidos por este contador numa coluna adicional:

- 1ª coluna: output contador θ_e
- 2ª coluna: output potenciómetro θ_e
- 3ª coluna: output extensómetro α_e

Para concretizar o processamento dos dados, seguimos as linhas de código sugeridas no enunciado.

Devido à relação velocidade - posição, verifica-se um efeito integrador no motor (**pólo na origem**), que deve ser retirado dos dados, diferenciando. Só posteriormente é que se pode identificar o modelo que relaciona a excitação elétrica imposta ao motor com o ângulo admitido pelo veio. Depois de identificar o modelo, teré de se voltar a adicionar ao mesmo um pólo em 1.

Ao diferenciar estes dados, amplia-se o ruído de altas frequências (já que derivar corresponde a multiplicar por s, pensando no caso de tempo contínuo). Por esta razão, deve-se aplicar um **filtro passa-baixo**. Deve-se, no entanto, ter em atenção que um valor muito elevado de λ, pode resultar na eliminação de efeitos que são necessários considerar para a obtenção do modelo, como por exemplo a chicotada.

Opta-se então por aplicar a função de transferência:

$$G(z) = (1 - z^{-1})(\frac{1-\lambda}{1-\lambda z^{-1}}) \tag{5}$$

De notar que se impõe assim um pólo em λ.

Escolheu-se primeiramente o valor do parâmetro λ sugerido de 0.8. No entanto, este valor foi alterado para 0.75, como explicado na próxima subsecção (4.3).

Removeu-se ainda o valor do *offset* do sinal de entrada do motor u, através da função *detrend()*. É importante forçar a que *detrend()* subtraia uma constante (ordem 0), invocando a função da seguinte forma *detrend(utrend,0)*.

4.3 Seleção do melhor modelo

Para a criação do modelo e dos conjuntos de teste, foram utilizados seis conjuntos de dados diferentes de sinais PRBS e um de onda quadrada. De modo a escolher o modelo a utilizar, no código *ciclo_principal* (secção 6.3), foi criado um modelo para cada conjunto de dados (função *model()* - secção 6.5) e testado com todos os conjuntos de dados (função *testSet()* - secção 6.4), para todas as combinações possíveis de $[n_a,n_b]$, sendo $n_a > n_b$.

Depois de recolhidos os valores de *fit* obtidos da comparação de todos os modelos com todos os conjuntos de teste (função *compare()*), foi escolhido o menor *fit* de cada modelo para cada combinação $[n_a, n_b]$, sendo este um dos fatores utilizados para escolher o modelo a utilizar.

Foi escolhido o menor fit e não o fit médio entre conjuntos de teste, pois este valor é mais representativo, visto que um modelo pode ter um fit relativamente baixo num conjunto de testes e mesmo assim ter um fit médio aceitável, o que não é o ideal sendo o objetivo criar um modelo capaz simular qualquer conjunto de dados.

Tabela 3: *Melhor conjunto de fits, para cada valor de n_a, obtido para λ = 0.8.*

n_a	n_b	fit	sinal utilizado para o modelo
3	2	84.5345	square3
4	3	86.5982	prbs9dentes
5	4	85.5042	prbs9dentes
6	4	85.8121	prbs9dentes
7	2	85.9994	prbs9dentes
8	6	86.6496	prbs9dentes
9	8	86.6037	prbs9dentes

Apesar de um dos critérios ser o *fit*, este não pode ser o único considerado. Outro dos critérios é a ordem do sistema obtido.

O sistema é composto por um motor, que tem um efeito integrador (pólo na origem), e uma barra flexível, que se sabe ter um comportamento oscilatório (par de complexos conjugados). Além disso, ao fazer a filtragem, adiciona-se um pólo em λ. No entanto, uma vez que o efeito integrador foi removido, este pólo não deve ser tido em consideração nesta fase. Assim, procura-se no mínimo um modelo de terceira ordem. Também se deve optar pela a menor ordem possível sem comprometer a exatidão do modelo, de modo a facilitar a implementação de um controlador.

Observando a tabela 3 verifica-se que o *fit* não varia significativamente aumentando a ordem do sistema. Optou-se assim pelo modelo de ordem inferior, ou seja, $n_a = 3$.

É ainda necessário perceber se este modelo simula efeitos como a chicotada. Procedeu-se assim à representação no mesmo gráfico da resposta real e da resposta dada pelo modelo para dois ensaios diferente. De salientar que nestas comparações foram utilizados sinais que não pertenciam nem ao conjunto de treino nem ao conjunto de teste.

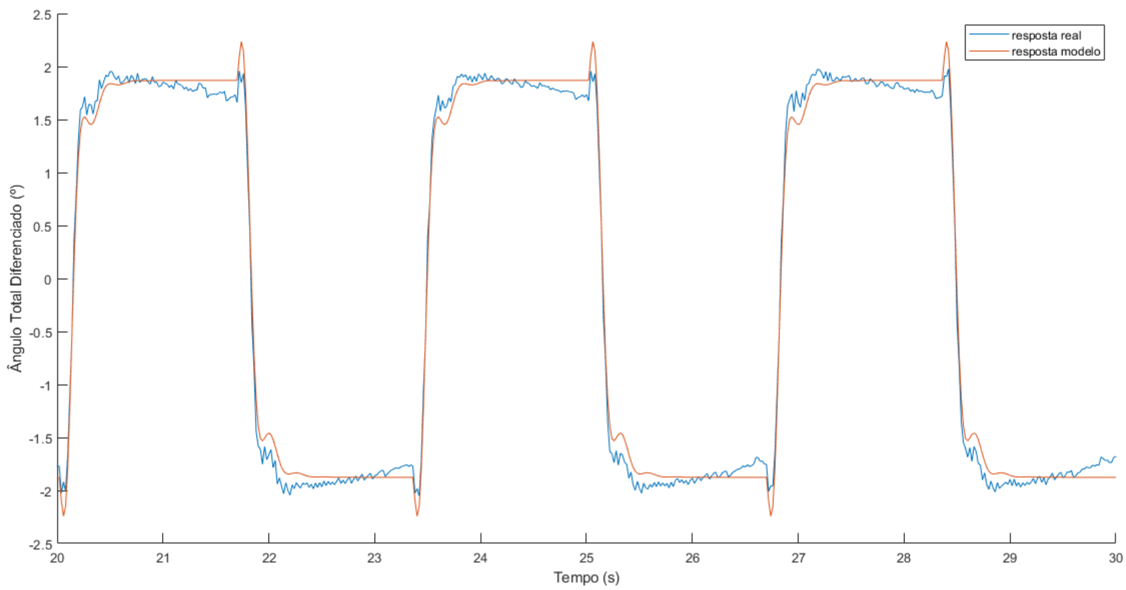

Figura 11: *Comparação entre a resposta real e a resposta dada pelo modelo escolhido para λ= 0.8, usando como conjunto de teste uma onda quadrada.*

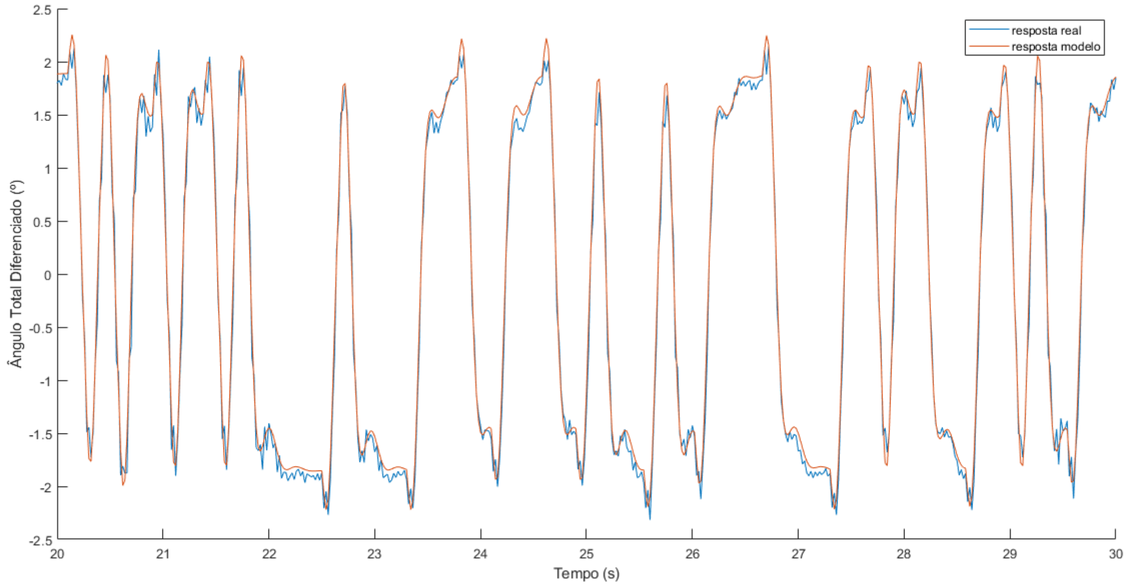

Figura 12: *Comparação entre a resposta real e a resposta dada pelo modelo escolhido para $\lambda = 0.8$ usando como conjunto de teste um sinal PRBS.*

Analisando as figuras 11 e 12 percebe-se que o modelo segue de forma satisfatória o sinal PRBS mas não o sinal de onda quadrada uma vez que apesar de simular a chicotada (segundo pico em cada patamar) não se verifica o pico da oscilação no início do patamar, o que deveria ocorrer.

Decidiu-se então reduzir o parâmetro λ do filtro uma vez que se considerou que o problema estava numa filtragem demasiado grande atenuando o sinal e por isso o modelo não estava a simular a oscilação inicial.

Aplicando $\lambda = 0.75$ obtém-se os seguintes resultados:

Tabela 4: *Melhor conjunto de fits, para cada valor de n_a, obtido para $\lambda = 0.75$.*

n_a	n_b	fit	sinal utilizado para o modelo
3	2	85.0901	square3
4	3	85.9209	prbs9dentes
5	4	85.1094	prbs9dentes
6	4	85.1057	prbs9dentes
7	6	85.8107	prbs9dentes
8	7	85.7579	prbs9dentes
9	8	85.7270	prbs9dentes

Aplicando o mesmo raciocínio aplicado ao set de resultados anterior, escolhemos mais uma vez o modelo com $n_a = 3$. Analisando as figuras 13 e 14, verifica-se que o modelo cumpre os requisitos necessários para à sua validação.

Em síntese, escolheu-se o sinal square3 como conjunto de treino, aplicou-se o filtro com o parâmetro $\lambda = 0.75$, n_a=3, n_b=2 e período de amostragem de 0.02s.

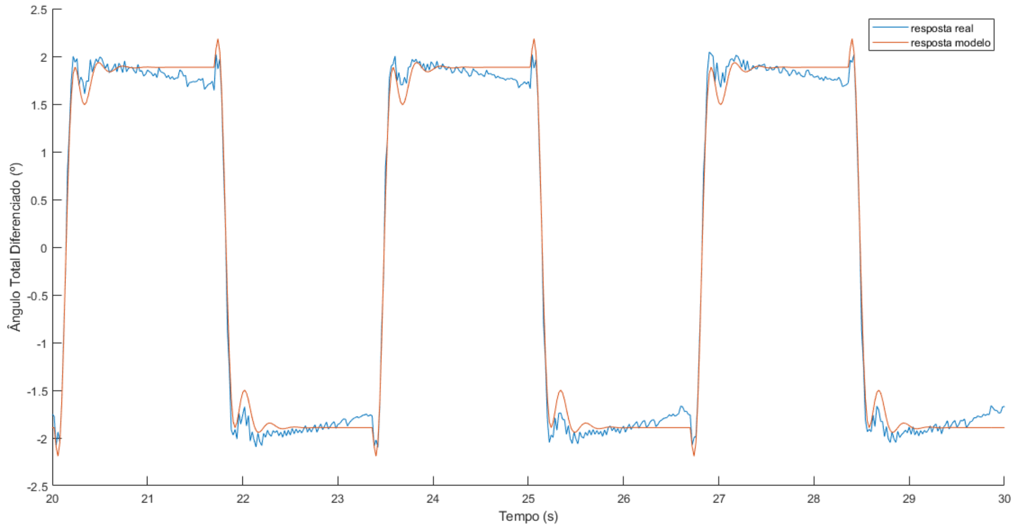

Figura 13: *Comparação entre a resposta real e a resposta dada pelo modelo escolhido para λ = 0.75 usando como conjunto de teste a onda quadrada square3.*

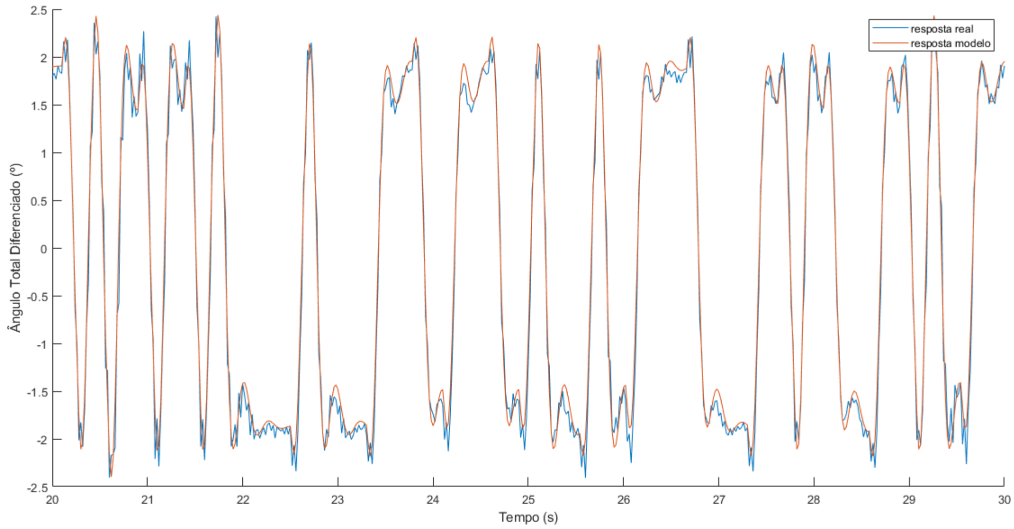

Figura 14: *Comparação entre a resposta real e a resposta dada pelo modelo escolhido para λ = 0.75 usando como conjunto de teste o conjunto prbs11dentes.*

4.4 Conversão para um Modelo em Espaço de Estados

Antes de converter o modelo criado para o espaço de estados, é necessário voltar a adicionar o pólo em z=1 que foi retirado quando se diferenciaram os dados. Para isso foi utilizada a função *conv()* do *Matlab*.

O espaço de estados tem a seguinte representação (onde y é a saída a controlar e u é a entrada do sistema). As matrizes A, B, C e D foram as obtidas para o modelo escolhido na secção anterior.

$$x(k+1) = Ax(k) + Bu(k)$$
$$y(k) = Cx(k) + Du(k)$$

$$A = \begin{bmatrix} 3.3090 & -4.2447 & 2.5238 & -0.5882 \\ 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 & 0.0 \end{bmatrix} \quad , \quad B = \begin{bmatrix} 1.0 \\ 0.0 \\ 0.0 \\ 0.0 \end{bmatrix} \quad , \quad C = \begin{bmatrix} -0.1104 \\ 0.1823 \\ 0.0 \\ 0.0 \end{bmatrix} \quad , \quad D = 0 \tag{6}$$

Foi obtida a função de transferência associada ao modelo, após ser adicionado o efeito integrador.

$$G(z) = \frac{-0.09726z^3 + 0.1756z^2}{z^4 - 3.346z^3 + 4.347z^2 - 2.614z + 0.613} \tag{7}$$

Através da função de transferência foi obtido o diagrama de bode associado.

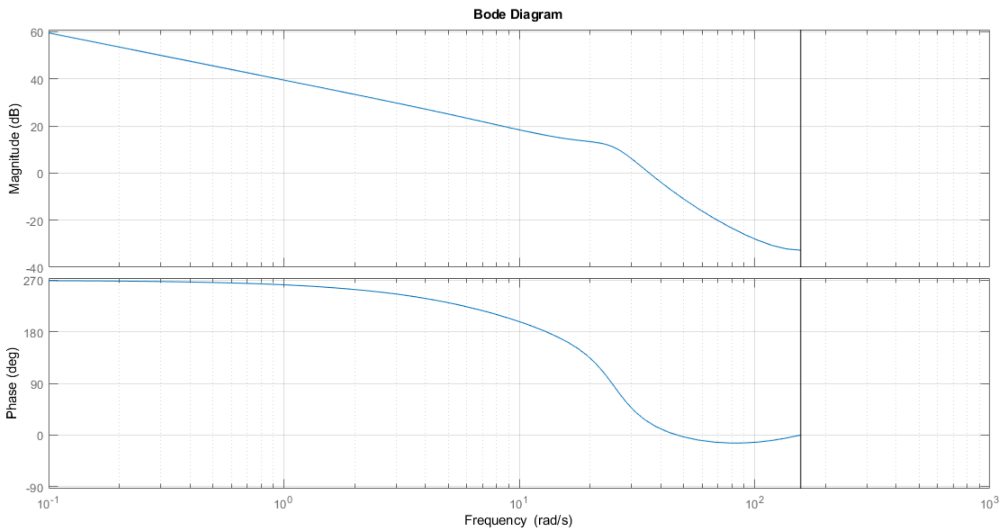

Figura 15: *Diagrama de Bode.*

Utizando a função *zplane()* é possível determinar a possição dos zeros e pólos do função de transferência no plano Z - figura 16. Também é possível através da função *rlocus()* obter o Root Locus - figura 17.

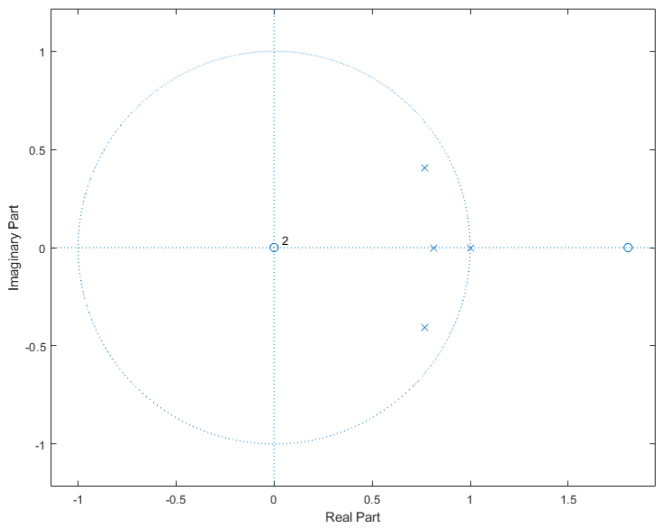

Figura 16: *Plano Z.*

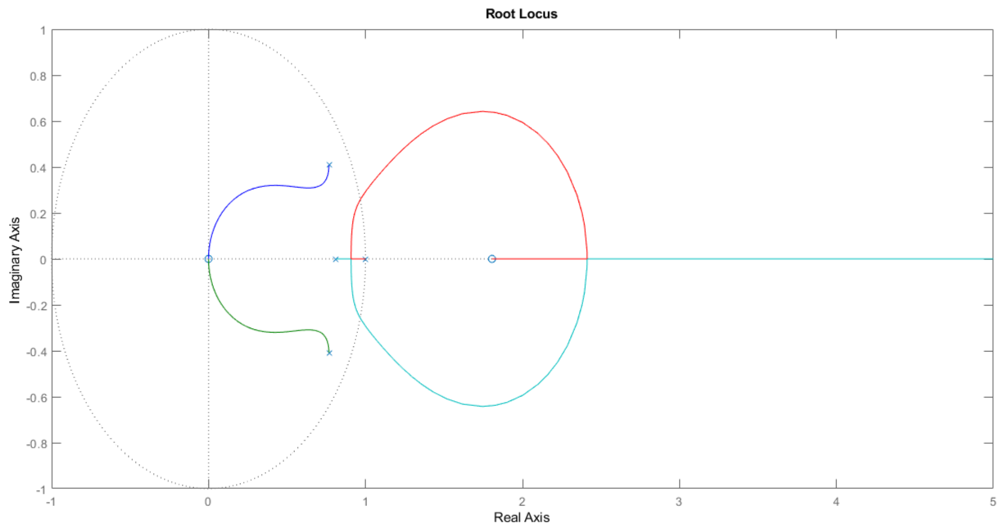

Figura 17: *Root Locus.*

5 Conclusão

Os objetivos definidos para esta parte do laboratório foram cumpridos. Efetuou-se a calibração dos sensores, obtendo-se as constantes K_b e K_p. Isto permitiu recolher dados e efetuar o seu tratamento tendo-se chegado a um modelo que mimica de forma satisfatória o sistema estudado: verificou-se uma boa simulação tanto do efeito de oscilação como de chicotada.

O modelo obtido é apenas de quarta ordem, o que será conveniente para o passo seguinte do projeto: o desenho do controlador. Durante a elaboração deste controlador, que vai ser desenvolvido na segunda parte deste laboratório, reconhece-se que é possível que, apesar de já se ter escolhido um modelo, este possa sofrer alterações.

Uma das fontes de erro que mais afeta a obtenção de dados e consequentemente a obtenção do modelo, trata-se da presença e manipulação do fio que permite obter os dados do extensómetro, visto que é necessário evitar que este ofereça resistência à rotação do veio.

6 Anexos

6.1 extensometro.m

```
%valores calibracao extensometro
load('lab1_dadosextensometro.mat');

%Valores experimentais

for i=9:1:7
    D(i+10,1) = 0.635*i; %0.635cm = 1/4 inch
end

L=37.5; %cm

alpha_E=[1.71
         1.38
         1.14
         0.84
         0.56
         0.28
         0.01
         0.33
         0.66
         0.97
         1.20
         1.51
         1.76
         2.05
         2.32
         2.62
         2.91];

%Calculos para obtencao de alfa
alpha =zeros(17,1);
for R = 1:1:17
    alpha(R,1) = 180/pi*atan(D(R,1)/L);
end

%Metodo dos minimos quadrados para calculo do kb
p=polyfit(alpha_E,alpha,1);
kb=p(1,1)

%Plot dos dados experimetais
plot(tensao_pot.time, tensao_pot.signals.values(:,2));
xlabel('Tempo (s)')
ylabel('Tensao extensometro (V)')
savefig('extensometro_exp.fig')

%Plot comparacao do fit com os dados experimentais
y2=polyval(p,alpha_E);
plot(alpha_E,y2, 'DisplayName', 'Relacao linear obtida')
hold on
scatter(alpha_E,alpha,'DisplayName', 'Pontos experimentais');
xlabel('Tensao extensometro (V)')
ylabel('Alfa (graus)')
savefig('extensometro_comparacao.fig')
```

```

legend
hold off
```

6.2 potenciometro.m

```
%valores calibracao extensometro
load('lab1_dadospotenciometro.mat');

%Calculo do kp
v_max = 4.956;
v_min =  4.985;

kp=360/(v_max (v_min))

%Plot
%Plot do sinal de comando e dos valores do sensor
plot(tensao_pot.time, input.signals.values, 'DisplayName', 'Sinal de comando', ...
    'LineWidth', 1.5);
hold on
plot(tensao_pot.time, tensao_pot.signals.values(:,1), 'DisplayName', 'Tensao ...
    potenciometro', 'LineWidth', 1.5);
xlabel('Tempo (s)')
ylabel('Tensao (V)')
savefig('potenciometro_exp.fig')

legend
hold off
```

6.3 ciclo_principal.m

```
%ciclo_principal   cria um modelo para cada conjunto de dados (model) e
%                  testa o com todos os conjuntos de dados (testSet), para
%                  todas as combinacoes de na, nb; no final cria uma matriz
%                  com o menor fit de cada modelo para todos as combinacoes
%                  de na, nb.

data = [struct('ficheiro','resposta_prbs3','t_elim',29,'∆_t',58,'dentes',0)
        struct('ficheiro','resposta_prbs4','t_elim',50,'∆_t',70,'dentes',0)
        struct('ficheiro','resposta_prbs6','t_elim',10,'∆_t',62,'dentes',0)
        struct('ficheiro','resposta_prbs8dentes','t_elim',10,'∆_t',110,'dentes',1)
        struct('ficheiro','resposta_prbs9dentes','t_elim',30,'∆_t',90,'dentes',1)
        struct('ficheiro','resposta_prbs11dentes','t_elim',10,'∆_t',110,'dentes',2)];
        %data.dentes = 0   apenas potenciometro
        %             1   ambos, utilizando potenciometro
        %             2   ambos, utilizando numero de dentes
N=6;

fitMatrix = zeros(N,N);

for na = 1: 1: 9
    for nb = 1: 1: na
```

```
        nc = na;
        nk = 1;
        nn = [na nb nc nk];
        for T = 1: 1: N
            %Test set
            z_test = testSet(data(T));
            for M = 1: 1: N
                %Modelo
                fit = model(data(M),z_test,nn);
                fitMatrix(M,T,na,nb)=fit;
            end
        end
    end
end

menor = menor_fit(fitMatrix,N);
```

6.4 testSet.m

```
function [z_test] = testSet(data)
%testSet   funcao recebe os dados a utilizar para criar o Test Set; devolve
%          a matriz z_test para testar os modelos criados.

    load(data.ficheiro);
    Kp = 36.2137;
    Ke =  3.3434;

    af = 0.8;
    Afilt = [1  af];
    Bfilt = (1 af)*[1  1];
    i = data.t_elim/0.02;
    j = (data.t_elim + data.Δ_t)/0.02;
    aux_matriz = 1;

    thetae_test = zeros((j i+1),1);
    alphae_test = zeros((j i+1),1);
    utrend_test = zeros((j i+1),1);
    t_aux2 = zeros((j i+1),1);

    col_thetae=1;
    col_alphae=2;
    if data.dentes == 1
        col_thetae=2;
        col_alphae=3;
    end
    if data.dentes == 2
        col_alphae=3;
        Kp = 360/1024;
    end

    for S = i : 1: j
        thetae_test(aux_matriz,1) = tensao_pot.signals.values(S,col_thetae);
        alphae_test(aux_matriz,1) = tensao_pot.signals.values(S,col_alphae);
        utrend_test(aux_matriz,1) = u(S,1);
        t_aux2(aux_matriz,1) = t(aux_matriz);
        aux_matriz = aux_matriz + 1;
```

```
    end

    ytrend_test = thetae_test*Kp + alphae_test*Ke;
    yf_test = filter(Bfilt,Afilt,ytrend_test);
    u_test = detrend(utrend_test,0);
    z_test = [yf_test u_test];
end
```

6.5 model.m

```
function [fit] = model(data,z_test,nn)
%model   funcao recebe os dados a utilizar para criar o modelo, a matriz
%       z_test, resultante da funcao testSet, e o vetor nn; apos criar o
%       modelo, este e testado com Test Set recebido e e devolvido o fit.

    load(data.ficheiro);
    Kp = 36.2137;
    Ke =  3.3434;

    af = 0.8;
    Afilt = [1  af];
    Bfilt = (1 af)*[1  1];

    i = data.t_elim/0.02;
    j = (data.t_elim + data.Δ_t)/0.02;
    aux_matriz = 1;

    thetae_test = zeros((j i+1),1);
    alphae_test = zeros((j i+1),1);
    utrend_test = zeros((j i+1),1);
    t_aux2 = zeros((j i+1),1);

    col_thetae=1;
    col_alphae=2;
    if data.dentes == 1
        col_thetae=2;
        col_alphae=3;
    end
    if data.dentes == 2
        col_alphae=3;
        Kp = 360/1024;
    end

    for S = i : 1: j
        thetae(aux_matriz,1) = tensao_pot.signals.values(S,col_thetae);
        alphae(aux_matriz,1) = tensao_pot.signals.values(S,col_alphae);
        utrend(aux_matriz,1) = u(S,1);
        t_aux2(aux_matriz,1) = t(aux_matriz);
        aux_matriz = aux_matriz + 1;
    end
    ytrend = thetae*Kp + alphae*Ke;
    yf = filter(Bfilt,Afilt,ytrend);
    u_detrend = detrend(utrend,0);
    z = [yf u_detrend];
    th = armax(z,nn);
    [den1,num1] = polydata(th);
```

```
    [¬, fit] = compare(z_test,th);
end
```

6.6 menor_fit.m

```
function [menor] = menor_fit(fitMatrix,N)
%menor_fit   recebe a matriz 'fitMatrix' do 'ciclo_principal' e o numero de
%           conjuntos de dados utilizados, N; devolve a matriz 'menor', com
%           o menor fit de cada modelo para todas as combinacoes de na, nb.

menor = zeros(9,9,N);
for na = 1: 1: 9
    for nb = 1: 1: na
        for M = 1: 1: N
            menor(na,nb,M) = min(fitMatrix(M,:,na,nb));
        end
    end
end
end
```