



Instituto Superior Técnico

Universidade de Lisboa

Mestrado Integrado em Engenharia Aeroespacial

Controlo por Computador

*Identificação e controlo por computador
de um braço robótico flexível*
Parte 1: Identificação e validação do modelo

Autores:

86635, Graça São Pedro

86661, Maria Carvalho

86668, Miguel Morgado

Docentes:

Prof. João Pedro Santos Gomes

Prof. João Miranda Lemos

Novembro de 2019

Conteúdo

1	Introdução	1
2	Descrição do sistema físico	2
2.1	Não trivialidade do objetivo de controlo	3
3	Interação entre o sistema físico e o computador	5
3.1	Calibração dos sensores	5
4	Identificação e validação do modelo do sistema	9
4.1	Descrição dos ensaios realizados	9
4.2	Processamento dos dados obtidos	11
4.3	Determinação do modelo ARMAX	13
4.4	Validação do modelo obtido	14
5	Conclusão	19
A	Código MATLAB utilizado	20
A.1	<i>lab1_kp_kb.m</i>	20
A.2	<i>cria_entradas.m</i>	20
A.3	<i>compacta_dados.m</i>	20
A.4	<i>testes_polo_LPF.m</i>	21
A.5	<i>select_model.m</i>	24
A.6	<i>statespace_final.m</i>	25

Lista de Figuras

1.1	Diagrama de blocos representativo do sistema que se pretende obter. Ref - sinal de referência; e - erro; u - entrada do sistema; y - saída do sistema; θ - posição angular do eixo do motor em relação a uma posição de referência; α - desvio angular da ponta da barra em relação à direção do eixo do motor.	1
2.1	Montagem do sistema físico a controlar.	2
2.2	Vista esquemática da montagem utilizada. M - motor; PA - amplificador; D/A, A/D - conversores D/A e A/D.	2
2.3	<i>Root Locus</i> aproximado do sistema físico, para ganhos positivos.	4
2.4	<i>Root Locus</i> aproximado do sistema físico, para ganhos negativos.	4
3.1	Funcionamento do modelo.	5
3.2	Modelo <i>Simulink</i> utilizado.	5
3.3	Variação de θ_e no tempo para efeitos de calibração.	6
3.4	Geometria da deflexão da barra flexível.	7
3.5	Tensão observada no extensómetro para efeitos de calibração.	8
3.6	Relação linear entre α e α_e . Linha contínua - regressão linear; pontos - dados experimentais.	8
4.1	Modelo de <i>simulink</i> para obtenção dos resultados dos ensaios.	10
4.2	Resposta do potenciómetro a uma onda PRBS com largura de banda 0.08Hz. . .	10
4.3	$y(z)$ antes e depois da filtragem, com $\lambda = 0.4Hz$	11
4.4	$y(z)$ antes e depois da filtragem, com $\lambda = 0.7Hz$	12
4.5	$y(z)$ antes e depois da filtragem, com $\lambda = 0.8Hz$	12
4.6	$y(z)$ antes e depois da filtragem, com $\lambda = 0.9Hz$	12
4.7	Filtragem com $\lambda = 0.7$ para um sinal PRBS com largura de banda 0.08Hz. . . .	13
4.8	Detalhes do seguimento dos dados experimentais	14
4.9	Onda quadrada - $f=0.3Hz$	15
4.10	Onda quadrada - $f=0.4Hz$	15
4.11	PRBS - $B=0.09Hz$	16
4.12	PRBS - $B=0.11Hz$	16
4.13	Ângulo total do ensaio e do modelo	17
4.14	Mapa de pólos e zeros	18
4.15	<i>Root-locus</i> associado ao modelo	18
4.16	Resposta em frequência do modelo escolhido	18

1. Introdução

O presente relatório foi produzido no âmbito da unidade curricular de Controlo por Computador, na sequência das três primeiras sessões laboratoriais respeitantes ao projeto *Identificação e Controlo por Computador de um braço robótico flexível*.

O principal objetivo deste trabalho é identificar e desenhar um sistema para controlar a posição do braço robótico flexível. Ao fornecer uma posição de referência para a extremidade do braço, pretende-se que esta seja seguida o mais fielmente possível. Assim, estuda-se a interação entre um sistema físico - o braço - e a componente computacional - o algoritmo. Na figura 1.1, representa-se aquilo que se pretende obter no final de toda a atividade.

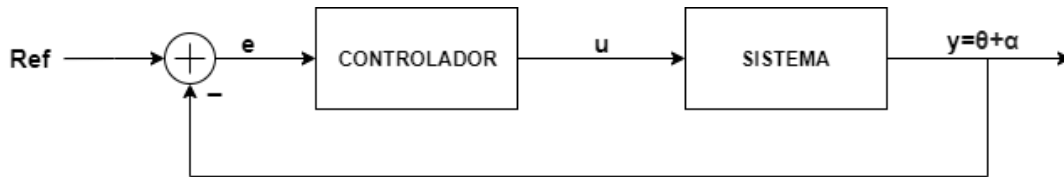


Figura 1.1: Diagrama de blocos representativo do sistema que se pretende obter. Ref - sinal de referência; e - erro; u - entrada do sistema; y - saída do sistema; θ - posição angular do eixo do motor em relação a uma posição de referência; α - desvio angular da ponta da barra em relação à direção do eixo do motor.

O projeto encontra-se dividido em duas partes. A primeira, a que este relatório se refere, corresponde à identificação e validação de um modelo matemático para o sistema físico em estudo. A segunda parte compreenderá o desenho de controladores e respetivos testes.

Inicia-se o relatório com uma breve descrição do sistema físico em estudo e dos sensores utilizados para medir as grandezas de interesse. O capítulo 3 corresponde ao trabalho realizado na primeira das três sessões de laboratório, onde se trabalhou com a geração, visualização, gravação e leitura de sinais, com o envio de estímulos para o motor e com a calibração dos sensores acoplados ao sistema. No capítulo 4, expõem-se os procedimentos utilizados para adquirir, identificar e validar o modelo do sistema mais adequado. Tendo-se identificado vários modelos do sistema consoante o estímulo enviado para o motor, o objetivo desta secção é concluir acerca do modelo mais adequado para descrever numericamente o sistema físico em uso.

2. Descrição do sistema físico

Antes de se proceder à realização da atividade, é importante conhecer o sistema físico que se pretende controlar. Neste capítulo, far-se-á uma breve descrição dos componentes do sistema, bem como dos sensores a ele ligados. A montagem completa está representada na figura 2.1.



Figura 2.1: Montagem do sistema físico a controlar.

O sistema é, portanto, constituído por um motor DC cujo eixo é solidário com uma das extremidades da barra flexível. O motor encontra-se ligado a um amplificador que, por sua vez, está conectado ao computador através de um *multiplexer*, cuja função é definir qual o estímulo que o sistema deve receber, já que cada montagem é partilhada entre dois grupos. Existem ainda no circuito conversores analógico-digitais (A/D) e digital-analógicos (D/A), que, como os nomes indicam, transformam o sinal analógico num sinal digital e vice-versa, respetivamente. Na figura 2.2, representa-se simbolicamente a montagem realizada.

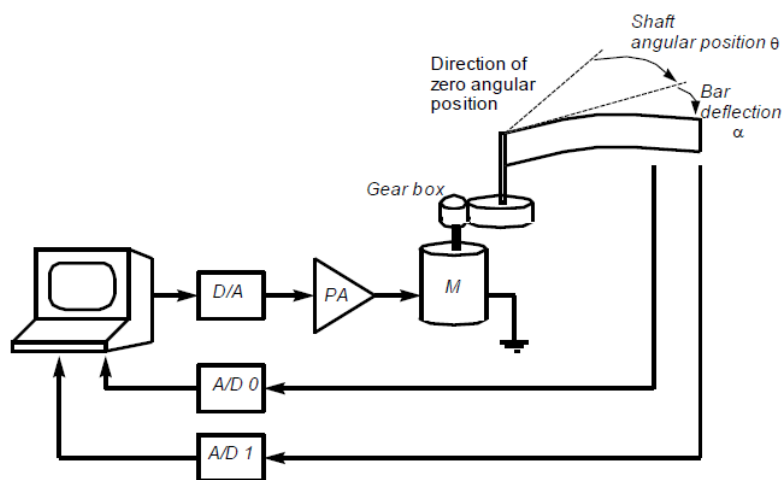


Figura 2.2: Vista esquemática da montagem utilizada. M - motor; PA - amplificador; D/A, A/D - conversores D/A e A/D.

2.1 Não trivialidade do objetivo de controlo

Matematicamente, uma solução diz-se não trivial quando apresenta uma estrutura com algum grau de complexidade e não necessariamente óbvia. No sistema em estudo, existem algumas propriedades que tornam o objetivo de controlo não trivial.

Por um lado, a barra não é rígida, o que gera movimentos oscilatórios em torno da posição de equilíbrio da barra sempre que esta se desloca. Estes movimentos observam-se facilmente nos instantes em que o motor para ou inverte o sentido de rotação - visto de um referencial solidário com o motor, a barra inicialmente desloca-se com velocidade angular no sentido oposto ao da rotação pretendida e de seguida oscila de forma elástica, evidenciando um efeito denominado chicotada, que advém da inércia da barra. Tal comportamento origina uma relação entre a tensão aplicada ao motor e a posição da ponta da barra que não é linear, logo, é não trivial.

Por outro lado, a tensão aplicada ao motor DC é proporcional à velocidade de rotação deste e não à sua posição angular. Logo, não é possível definir uma tabela de tensões elétricas às quais corresponda univocamente uma e uma só posição da ponta da barra. Pela mesma razão, aplicar uma tensão constante no motor resulta apenas numa velocidade angular constante do mesmo em torno do seu eixo.

Estas duas propriedades podem ser descritas através da função de transferência em anel aberto do sistema. O efeito de chicotada é consequência da existência de um zero de fase não mínima no sistema. Além disso, uma vez que se pretende controlar a posição angular da barra, θ , e não a sua velocidade angular - correspondente à primeira derivada de θ -, é necessário incluir um pólo na origem, que constitui um integrador.

Realimentar o sistema com um ganho simples não é, contudo, uma solução útil para o estabilizar. Mesmo que o ganho seja pequeno, o pólo na origem tenderá a deslocar-se na direção do zero de fase não mínima, tornando desta forma o sistema instável. De forma a confirmá-lo, pode-se descrever o sistema arbitrando os seus pólos e zero expectáveis de forma a definir uma função de transferência e desenhando um *Root Locus*.

Aproxime-se o movimento da barra por dois movimentos oscilatórios independentes e complementares: um mais rápido e amortecido e outro mais lento e menos amortecido. Tal traduz-se por dois pares de pólos conjugados no semiplano complexo esquerdo, não muito distantes da origem (ou seja, frequências naturais $\omega_n < 1$). Deste modo, o denominador da função de transferência correspondente ao sistema é constituído pelo produto de dois polinómios completos de segundo grau e o pólo na origem, correspondente ao integrador. O numerador, por outro lado, consiste no zero de fase não mínima. Em resumo, a função de transferência do sistema seria algo semelhante a:

$$FT = \frac{s - a}{s(s^2 + bs + c)(s^2 + ds + e)}. \quad (2.1.1)$$

Apresenta-se, nas figuras 2.3 e 2.4 a distribuição espacial aproximada dos zeros e pólos do sistema em cadeia aberta sujeito a um ganho amplificador do erro de seguimento positivo e negativo, respetivamente.

Na figura 2.3, verifica-se que, em malha fechada e com ganho positivo, o pólo que se encontra na origem se desloca para o semiplano complexo direito, instabilizando o sistema. Para ganhos negativos, de acordo com a figura 2.4, o controlo proporcional com ganho negativo oferece uma margem de ganho reduzida, uma vez que os dois pólos conjugados mais próximos do eixo imaginário se deslocam para o semiplano complexo direito para valores do ganho cada vez mais negativos. Apesar de existir esse intervalo de valores do ganho em que o sistema é estável, tratam-se de ganhos tão pequenos que tornam o sistema demasiado lento a seguir uma entrada de referência, o que não é razoável, tendo em conta o objetivo deste projeto.

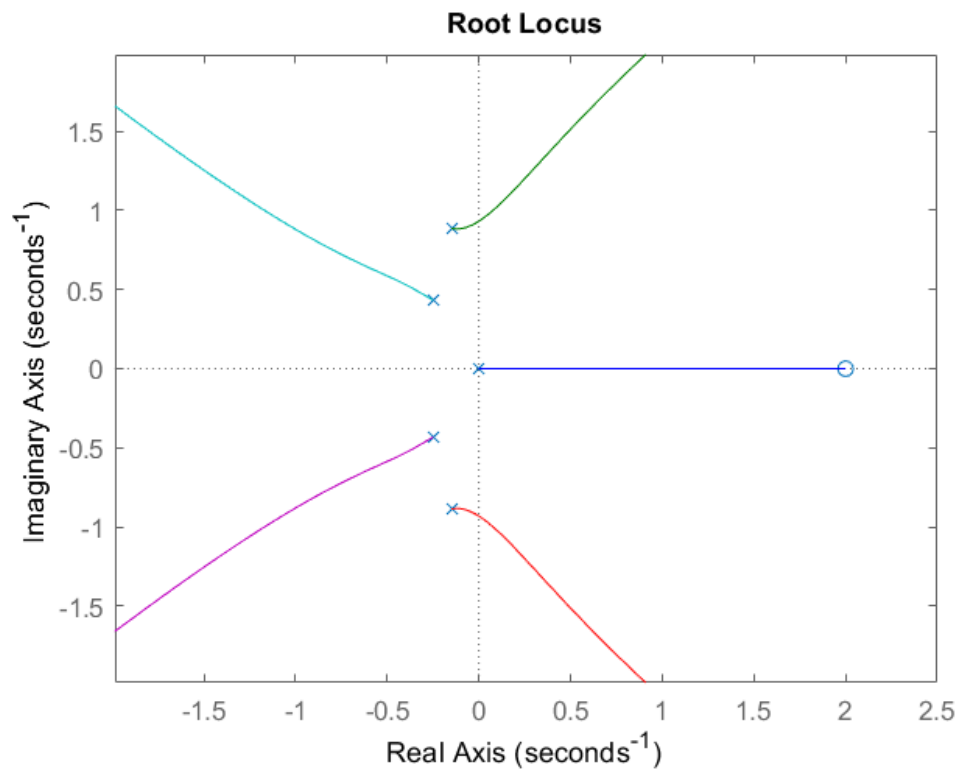


Figura 2.3: *Root Locus* aproximado do sistema físico, para ganhos positivos.

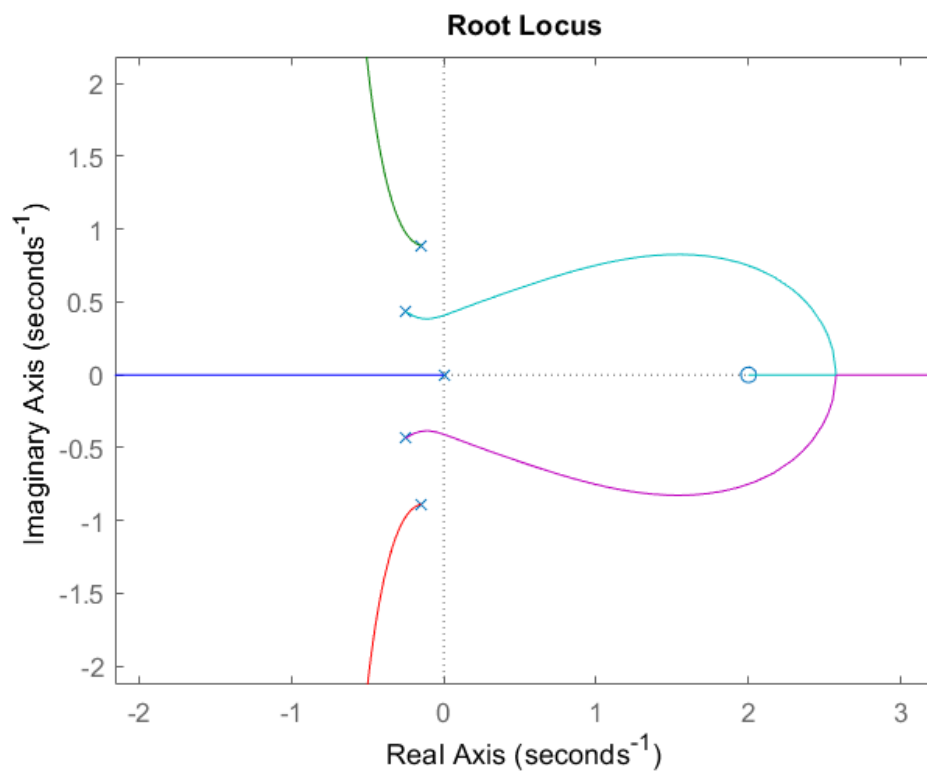


Figura 2.4: *Root Locus* aproximado do sistema físico, para ganhos negativos.

3. Interação entre o sistema físico e o computador

3.1 Calibração dos sensores

A calibração dos sensores é de vital importância, uma vez que permite a confiança nos resultados adquiridos para a construção de um modelo do sistema, permitindo adaptar o *software* às não idealidades do sistema físico. Para além disso, a calibração permite determinar as constantes que relacionam as tensões de saída dos sensores com os valores reais dos ângulos que estes medem.

Sendo o objetivo deste projeto controlar a posição angular da barra flexível, y , é necessário determinar o comportamento das grandezas que a compõem. Como facilmente se observa, quando a barra se desloca para uma determinada posição θ , imposta pelo motor, existe um desvio da posição desejada, α , denominada em diante de deflexão da barra flexível. Assim, tal como descrito na figura 3.1, vem que:

$$y = \theta + \alpha. \quad (3.1.1)$$

As unidades utilizadas para estas grandezas ao longo deste documento serão graus ($^\circ$).

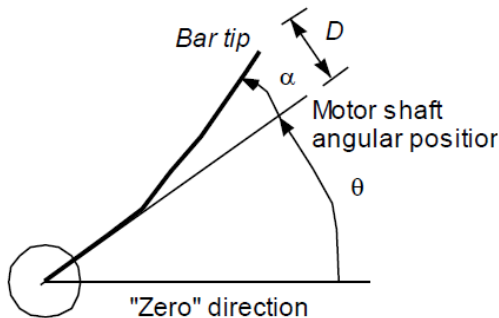


Figura 3.1: Funcionamento do modelo.

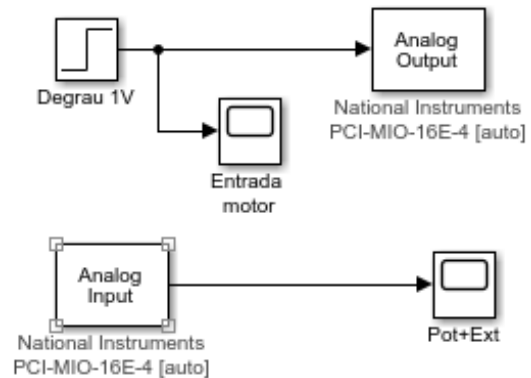


Figura 3.2: Modelo *Simulink* utilizado.

No caso em apreço, por forma a obterem-se os dados necessários para a caracterização do movimento da barra flexível, optou-se pela utilização de um potenciómetro, que mede a tensão θ_e que depois é convertida na posição angular do eixo do motor, e um extensómetro, na extremidade fixa da barra, por forma a medir a tensão α_e que depois se converte na sua deflexão. Uma vez que se trabalham com sistemas elétricos para a medição de grandezas físicas, e estes medem efetivamente tensões, existe uma correspondência entre o valor do ângulo observado, em grau, e o valor que é medido pelos sensores, em volt. Determinam-se agora as constantes que permitem converter as medições das tensões θ_e e α_e nos ângulos θ e α .

Modelo *Simulink*

Tendo em conta a constituição do sistema físico utilizado para a obtenção de dados, descrito no capítulo 2, pode-se construir um modelo *Simulink* que permite observar a evolução dos valores lidos pelos sensores. Utiliza-se um bloco, *Analog Output*, correspondente ao envio de um estímulo para o motor, ao qual é associado o *hardware* utilizado, sendo o *Scope* utilizado para verificar esta entrada. Depois, tem-se um bloco *Analog Input* que recebe os sinais lidos pelo potenciómetro e pelo extensómetro, servindo o *Scope* para verificar a sua evolução.

O código utilizado no tratamento dos dados obtidos por forma a obter as constantes K_p e K_b encontra-se na secção A.1.

Potenciómetro

Num primeiro momento, estudou-se o comportamento do potenciómetro, quando aplicado um sinal de entrada com a forma de um degrau de amplitude 1V. O resultado obtido está expresso graficamente na figura 3.3.

Quando este sinal de entrada foi imposto, o utilizador poderia esperar que o motor estabilizasse numa posição, uma vez que se está a aplicar uma tensão constante, contudo tal não acontece, tal como explicado na secção 2.1. Veja-se que a tensão aplicada irá acionar a rotação do motor, conferindo-lhe uma velocidade angular constante e ficando o motor a rodar indefinidamente, não estabilizando numa posição.

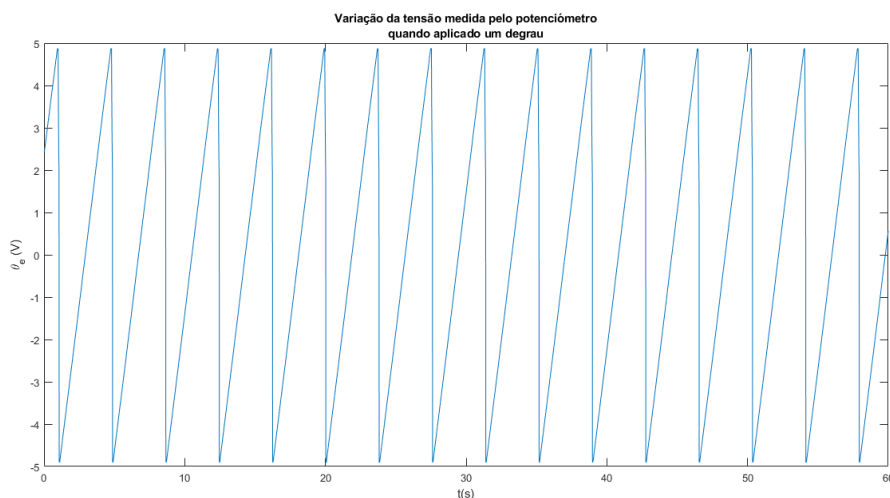


Figura 3.3: Variação de θ_e no tempo para efeitos de calibração.

As linhas verticais que se encontram neste gráfico indicam a existência de uma descontinuidade, a *death line*, previamente conhecida, sendo que o elemento resistivo do potenciómetro é alimentado, nos extremos, correspondentes a 0° e a 360° , por tensões de 5V e -5V, teoricamente. A variação de tensão encontrada corresponde a uma volta completa. Assim sendo, conclui-se que

$$K_p = \frac{\Delta\theta}{\Delta\theta_e} = \frac{360}{4.917 - (-4.897)} = 36.6823^\circ/V, \quad (3.1.2)$$

obtendo-se a seguinte relação linear:

$$\theta = K_p\theta_e = 36.6823\theta_e. \quad (3.1.3)$$

O valor calculado é próximo do valor que se determinaria para um sistema ideal, $K_p = 36^\circ/V$, com $\Delta\theta_e = 10^\circ$.

Extensómetro

Para a determinação da constante que regula as medições do extensómetro, foi medido o valor de α_e em posições onde a deflexão da barra podia ser calculada, por forma a recolher estes dados e avaliar a sua relação.

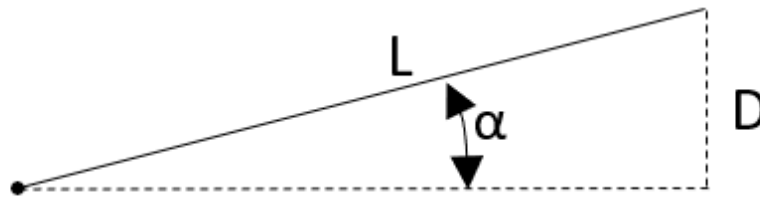


Figura 3.4: Geometria da deflexão da barra flexível.

Atente-se à geometria que se considerou para determinação de α . Sendo α muito pequeno, pode-se aproximar $L \cdot \cos(\alpha) \approx L$, com $L = 37.5\text{cm}$. Utilizando um pente com dentes separados por um quarto de polegada (0.635 cm), fez-se variar D por posições conhecidas e através da relação

$$\tan(\alpha) = \frac{D}{L} \Leftrightarrow \arctan\left(\frac{D}{L}\right) = \alpha, \quad (3.1.4)$$

que vem da geometria da figura 3.4, calculam-se os valores de α correspondentes às diferentes posições. Os dados obtidos encontram-se discriminados nas colunas 1 e 2 da tabela 3.1.

Para a determinação dos valores de α_e correspondentes, estudou-se a saída do extensómetro, tendo a barra sido mantida na mesma posição durante tempo suficiente para os valores estabilizarem. A este procedimento estão inerentes diversas fontes de erro, nomeadamente a dificuldade de manter a barra imobilizada e medir essa posição e a necessidade de interferência humana para mover a barra entre as várias posições, que inevitavelmente adicionava instabilidade ao sistema.

Aquando da análise dos dados, determinou-se o valor estabilizado para cada posição da barra, através do gráfico da figura 3.5. Apresentam-se estes dados na terceira coluna da tabela 3.1.

Tabela 3.1: Dados utilizados para determinar α_e .

D (cm)	$\alpha(^{\circ})$	$\alpha_e(V)$
-5.080	-7.7147	1.680
-4.445	-6.7599	1.411
-3.810	-5.8013	1.128
-3.175	-4.8395	0.8545
-2.540	-3.8749	0.5664
-1.905	-2.9081	0.2587
-1.270	-1.9397	-0.3418
-0.635	-0.9701	-0.3125
0	0	-0.6738
0.635	0.9701	-0.9668
1.270	1.9397	-1.211
1.905	2.9081	-1.494
2.540	3.8749	-1.748
3.175	4.8395	-2.051
3.810	5.8013	-2.324
4.445	6.7599	-2.617
5.080	7.7147	-2.8769

De forma a encontrar a curva que melhor se ajusta aos dados, foram utilizadas duas funções do MATLAB:

- *polyfit* - retorna os coeficientes para um polinómio, no caso, grau 1, que melhor se ajusta (através de uma aproximação de mínimos quadrados) aos dados adquiridos;

- *polyval* - cria uma curva ajustada ao polinómio cujos coeficientes foram devolvidos pela função *polyfit*.

Na figura 3.6, verifica-se que a estimativa de mínimos quadrados é uma boa forma de avaliar a evolução de α , dada a semelhança entre a curva obtida e os pontos medidos. É de salientar que o ponto mais afastado do *plot* do polinómio de primeiro grau corresponde a um dos pontos no meio do pente utilizado para estabilizar a barra flexível, no qual se verificou uma maior dificuldade de obtenção de valores estáveis, uma vez que a barra não ficou totalmente imobilizada.

Da análise, vem que o extensómetro está afeto a um *offset*, que, ao contrário do potenciómetro, não pode ser desprezado, evidente pelo facto de os zeros não corresponderem. Assim, vem que:

$$\alpha = K_b \alpha_e + offset \quad (3.1.5)$$

concluindo-se do resultado da função *polyfit* que $K_b = -3.3652^\circ/V$, com um *offset* = -2.1217° .

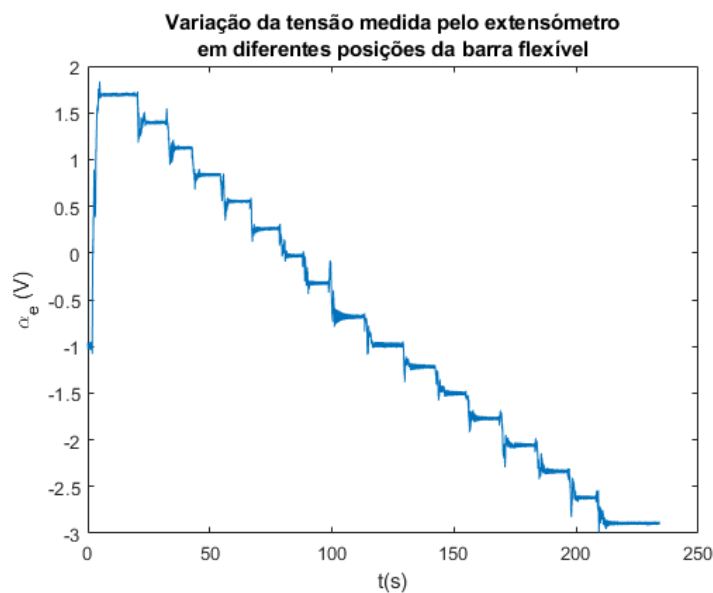


Figura 3.5: Tensão observada no extensómetro para efeitos de calibração.

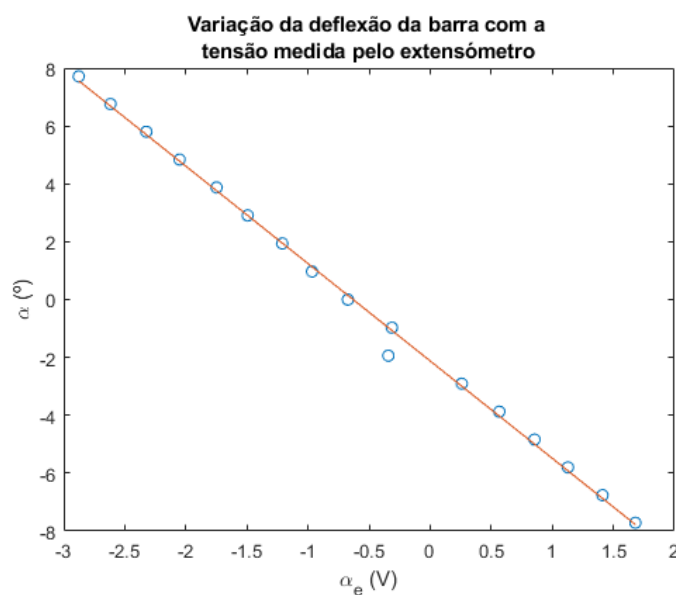


Figura 3.6: Relação linear entre α e α_e . Linha contínua - regressão linear; pontos - dados experimentais.

4. Identificação e validação do modelo do sistema

Neste capítulo, descrevem-se os procedimentos utilizados de modo a identificar o melhor modelo para o sistema a controlar. Este processo é composto pelos seguintes passos:

1. Gerar uma onda da tensão a aplicar ao motor, excitar o sistema com a mesma e recolher os dados da resposta deste;
2. Processar os dados recolhidos para eliminar componentes indesejadas;
3. Identificar um modelo sob a forma de uma equação de diferenças - modelo ARMAX (*Autoregressive-moving-average model with exogenous inputs*):
 - (a) Selecionar as melhores ordens do modelo;
 - (b) Estimar os parâmetros do modelo;
4. Converter o modelo ARMAX obtido num modelo em espaço de estados.

4.1 Descrição dos ensaios realizados

Estando a calibração dos sensores concluída, está-se em condições de proceder à obtenção de dados para análise e identificação do modelo matemático mais adequado para representar o sistema físico.

Para compreender o funcionamento do sistema em situações diversas, optou-se por excitá-lo de diferentes formas, tendo em conta, para o sinal de entrada, características como forma da onda, amplitude, frequência e ainda duração da experiência. As ondas utilizadas foram geradas com recurso ao script da secção A.2.

Assim, foram realizados os ensaios que constam da tabela 4.1.

Tabela 4.1: Sinais utilizados para excitar o sistema e suas características.

Tipo de onda	Frequência (Hz)	Largura de Banda (Hz)	Amplitude (V)
Quadrada	0.3	-	1
Quadrada	0.4	-	1
Quadrada	0.5	-	1
Quadrada	0.5	-	0.5
PRBS	-	0.08	1
PRBS	-	0.09	1
PRBS	-	0.10	1
PRBS	-	0.11	1

A onda quadrada é um sinal simples, e útil no sentido de que é fácil prever a resposta do sistema a esta onda. Contrariamente, um sinal PRBS, *Pseudorandom Binary Sequence*, impõe ao sistema um sinal de entrada com comportamentos pseudo-aleatórios, sendo impossível estimar qual a posição seguinte e, conseqüentemente, envolvendo mais cuidados por parte dos utilizadores.

As frequências e larguras de banda utilizadas para gerar os sinais de entrada correspondem a valores que se encontram na vizinhança dos sugeridos no guia de laboratório (nomeadamente 0.3, 0.4 e 0.5 Hz), por forma a obterem-se diferentes respostas plausíveis. Se estas frequências de excitação do sistema forem demasiado elevadas, o sistema físico não é capaz de reagir rápido o suficiente e acaba por filtrar o *input*. No entanto, pretende-se utilizar uma frequência grande

o suficiente para excitar os modos transientes de resposta do sistema e não apenas os estados estacionários. Todos os sinais de entrada tinham o período de amostragem sugerido, 0.020s, ou seja, uma frequência de amostragem de 50Hz.

Já para a amplitude, tendo sido também usado principalmente o valor sugerido de 1V, optou-se por verificar qual a reação do sistema ao valor de 0.5V, não tendo sido registadas variações significativas. Entendeu-se que, para amplitudes maiores e consequentes movimentos mais rápidos, os comportamentos não lineares do sistema poderiam sobrepor-se ao comportamento ideal que se pretende estudar, sendo esta situação algo a evitar. Amplitudes de valores menores também poderiam conduzir a maus resultados, uma vez que a folga das engrenagens da caixa de velocidades daria origem a uma "zona morta" considerável em que a barra não registaria qualquer movimento, afetando assim a resposta obtida.

Para se recolher os dados, utilizou-se o modelo *simulink*, na figura 4.1 que tem na entrada do sistema um bloco que importa do *Workspace* as características da onda gerada para excitar o sistema e recorria ao bloco *Scope* para a visualização dos resultados no ecrã. À posteriori, estes dados era também guardados no *Workspace* e analisados, da forma explicitada de seguida.

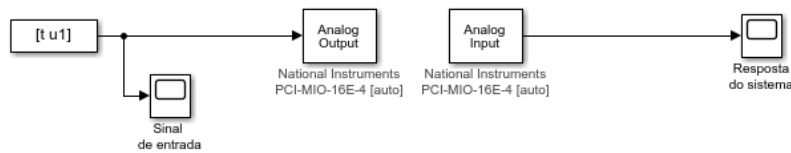


Figura 4.1: Modelo de *simulink* para obtenção dos resultados dos ensaios.

Por forma a fazer ensaios seguros, foram sempre necessárias três pessoas, uma para manusear o amplificador, outra para trabalhar com o *software* e outra para assegurar que não se conspurcava a montagem com os fios ligados à montagem.

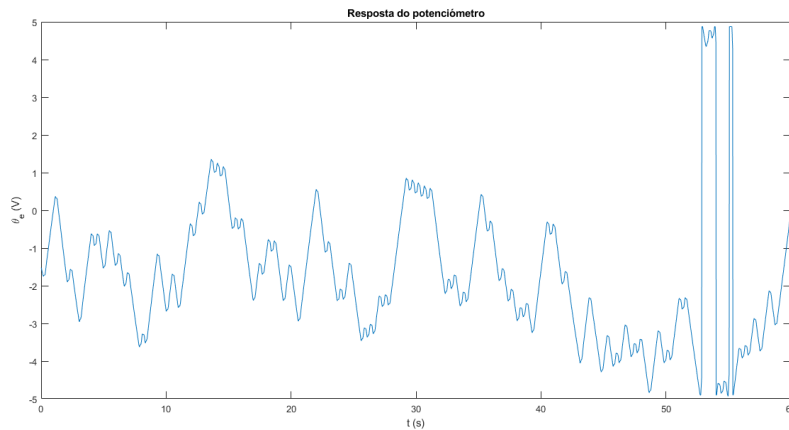


Figura 4.2: Resposta do potenciômetro a uma onda PRBS com largura de banda 0.08Hz.

Um dos principais problemas encontrados ao longo do procedimento experimental foi a existência de descontinuidades em θ_e quando a barra flexível ultrapassava o limite físico entre 0° e 360° , já discutido em 3.1. Por forma a obter resultados coerentes, tem de se analisar intervalos de tempo que não estejam afetados por este problema, tendo por isso, já aquando da análise, sido feita a seleção de dados por forma a evitar isto.

Assim, a duração dos ensaios foi variada, tendo-se optado por utilizar 50 segundos de cada um deles, um período razoável para verificação dos resultados e que não ultrapassa os limites do sistema físico.

Um exemplo dos resultados obtidos, ainda sem avaliação dos dados e seu pós-processamento, encontra-se na figura 4.2. Como se pode observar, entre 50 segundos e 60 segundos, o sistema ultrapassou por várias vezes o limite referido anteriormente, apresentando descontinuidades. Por forma a utilizar este ensaio para identificação do modelo, procedeu-se à truncação dos dados, para não se incluírem as descontinuidades do sinal.

É ainda importante referir que estamos a usar um tempo de amostragem de 0.02s, tal como indicado. É importante referir este valor na medida em que se fosse demasiado grande não seria possível obter dados plausíveis e se fosse demasiado pequeno iria registar-se maioritariamente ruído, impossibilitando de qualquer das formas uma caracterização viável do sistema.

4.2 Processamento dos dados obtidos

Como já foi dito, os dados recolhidos em tempo real, aquando da realização das experiências, necessitam de ser tratados antes de se poder proceder à identificação de um modelo. O processamento dos dados obtidos será feito de acordo com o que é sugerido no guia de laboratório e sobre os intervalos de dados selecionados.

Existe no sistema um efeito integrador, pela relação entre a velocidade angular e a posição angular do eixo do motor, que necessita de ser removido antes do restante processamento, sendo isto feito através da diferenciação dos dados obtidos para a variável y . Ao diferenciar, está-se também a criar maior susceptibilidade a frequências elevadas, pelo que será necessário filtrá-las por forma a não afetar os dados obtidos, com recurso a um filtro passa-baixo. Ao mesmo tempo, o ruído introduzido pelos sensores é igualmente filtrado.

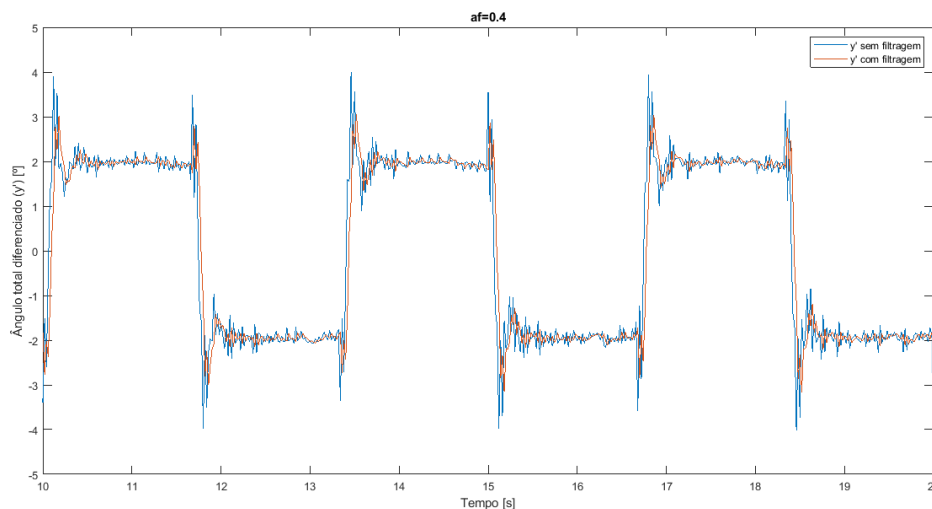


Figura 4.3: $y(z)$ antes e depois da filtragem, com $\lambda = 0.4Hz$.

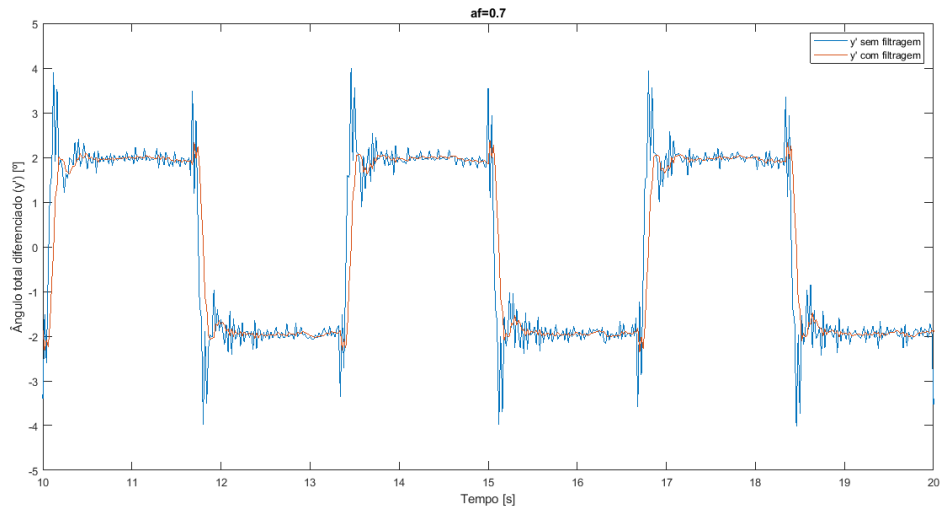


Figura 4.4: $y(z)$ antes e depois da filtragem, com $\lambda = 0.7Hz$.

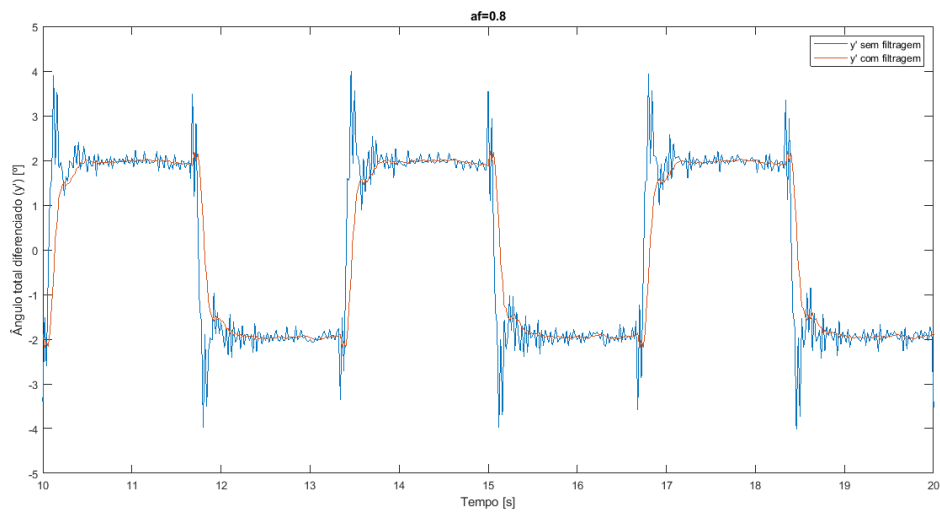


Figura 4.5: $y(z)$ antes e depois da filtragem, com $\lambda = 0.8Hz$.

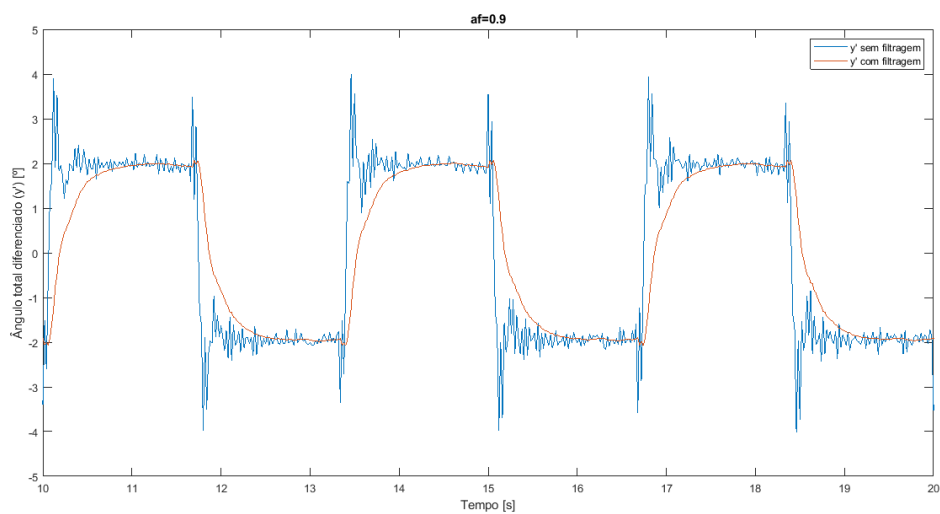


Figura 4.6: $y(z)$ antes e depois da filtragem, com $\lambda = 0.9Hz$.

As duas operações (filtragem e derivação) podem ser concretizadas através da função de transferência G indicada no guia do laboratório:

$$G(z) = (1 - z^{-1}) \left(\frac{1 - \lambda}{1 - \lambda z^{-1}} \right)$$

cujo primeiro fator corresponde à derivação e o segundo fator depende do parâmetro λ que varia entre zero e um. Um valor mais próximo de zero confere ao filtro um maior poder de atenuação, pelo que será mais adequado para ensaios muito perturbados por ruído.

Deste modo, de forma a escolher o valor mais indicado, o desempenho do filtro foi testado e analisado para diferentes valores de λ (figuras 4.3 a 4.6), tendo-se chegado à conclusão de que $\lambda = 0.7Hz$ produzia uma curva melhor ajustada aos dados experimentais, sem ainda assim ignorar ou filtrar o efeito de *chicotada* presente sempre que há uma variação brusca da derivada de $y(z)$ e deveras relevante para a correta determinação do modelo. Por forma a comprovar que esta era de facto uma boa escolha, optou-se também por aplicar o mesmo processo a um sinal PRBS como demonstrado na figura 4.7. Verificou-se que eram obtidos bons resultados

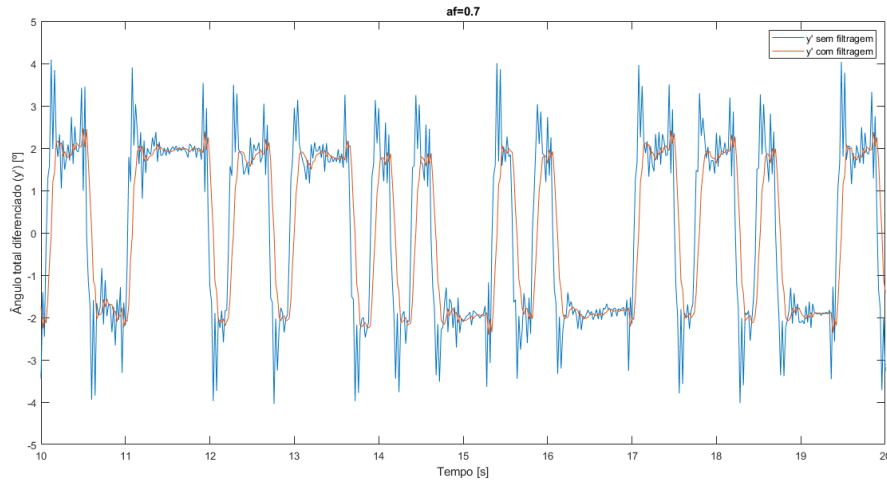


Figura 4.7: Filtragem com $\lambda = 0.7$ para um sinal PRBS com largura de banda 0.08Hz.

4.3 Determinação do modelo ARMAX

Após se obter o melhor valor do pólo do filtro, procedeu-se à determinação do modelo matemático que descreve o sistema. Para tal, recorreu-se ao script *select_model.m*.

Começou-se por determinar o vetor correspondente a y , que foi depois filtrado como descrito anteriormente. Depois, definindo o primeiro dos sete conjuntos de dados da tabela 3.1 como o conjunto de treino do ARMAX, geraram-se vários modelos variando os parâmetros n_a e n_b . Para cada modelo, avaliou-se ainda a qualidade, ou o *fit*, deste face aos restantes conjuntos de dados, ditos conjuntos de teste, obtendo-se um *fit* médio para cada conjunto de n_a , n_b e experiência de treino. Repetiu-se o procedimento, fixando como experiência de treino cada uma das restantes seis experiências.

Importa referir que se impuseram algumas restrições na determinação do melhor modelo matemático. O modelo ARMAX depende de quatro constantes: n_a , n_b , n_c e n_k . Optou-se por variar apenas as duas primeiras, sempre com $n_b \leq n_a - n_k$, mantendo-se n_c e n_k fixos. O valor de n_c é desconhecido, pelo que se assumiu, de acordo com o indicado no guia da atividade, que $n_c = n_a$. Por outro lado, n_k representa o atraso puro do sistema e assumiu-se, sem perda de generalidade, constante e igual a 1.

Atendendo a que o efeito integrador do motor é anulado, que se adicionou um pólo associado ao filtro passa-baixo e que temos pólos conjugados associados às oscilações da barra, conclui-se que o modelo a obter não poderá ser de uma ordem menor que 3.

Assim, uma vez que o objetivo é determinar um modelo com uma ordem relativamente baixa e adequado aos dados medidos, construiu-se a tabela 4.2, onde se discriminam os melhores resultados do *fit* médio para cada ordem do sistema, correspondente a n_a , bem como o valor de n_b e o conjunto de treino utilizado para obter o modelo.

Tabela 4.2: Melhores *fits* médios encontrados para cada n_a .

n_a	n_b	Fit médio (%)	Ensaio utilizado
2	1	76.2627	PRBS, B=0.11Hz
3	2	89.9724	PRBS, B=0.1Hz
4	2	90.7945	PRBS, B=0.08Hz
5	4	91.0453	PRBS, B=0.1Hz
6	5	91.8619	PRBS, B=0.11Hz
7	6	93.7994	PRBS, B=0.11Hz

Desta tabela, constata-se que, quanto maior a ordem do modelo, correspondente a n_a , melhor é a qualidade deste face aos dados experimentais adquiridos. É ainda de referir que os melhores valores do *fit* ocorrem quando o conjunto de treino utilizado está associado a uma onda PRBS e não a uma onda quadrada.

Todavia, não interessa determinar um modelo de sexta ou sétima ordem, uma vez que isso dificulta o desenho do controlador na segunda parte deste projeto. Para além disso, os modelos de ordens mais baixas apresentam *fits* muito próximos daqueles oferecidos pelos modelos de ordens mais elevadas, variando apenas em 3% – 4%. Assim, consideram-se doravante os dois modelos dos indicados na tabela 4.2 correspondentes a $n_a = 3$ e $n_a = 4$, que constituem um bom compromisso entre ordem do modelo e *fit* médio calculado.

4.4 Validação do modelo obtido

Tendo sido escolhidos os modelos com o melhor *fit* médio, há que verificar qual deles deve ser utilizado através do ajuste aos dados experimentais. Para tal, foram desenhados, no mesmo *plot* de MATLAB, representado na figura 4.8, as linhas correspondentes à evolução experimental, representada a azul, à evolução do modelo com $n_a = 3$, a vermelho, e à evolução com $n_a = 4$, a amarelo.

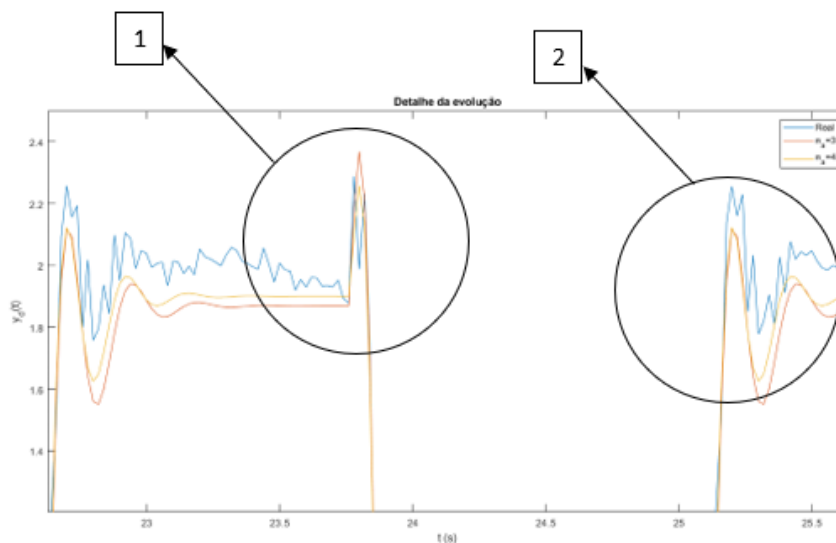


Figura 4.8: Detalhes do seguimento dos dados experimentais

O seguimento de ambos os modelos é bastante razoável, tendo o efeito da chicotada sido distintamente verificado; veja-se o detalhe 1 da figura 4.8, bem como a oscilação inicial na mudança de estado, detalhada no ponto 2 também da figura 4.8.

Nas figuras 4.9, 4.10, 4.11 e 4.12 estão representados os seguimentos de ambos os modelos considerados bem como a evolução experimental, tendo sido feita a validação para parte dos ensaios experimentais que não foram utilizados como *training sets*. Verifica-se que ambos os modelos apresentam resultados muito semelhantes aos àqueles efetivamente observados em laboratório, sendo as maiores oscilações claramente visíveis, tanto para as ondas quadradas como para os sinais PRBS. Existe ainda algum ruído nas fases mais estáveis, consequência da ordem do filtro utilizado.

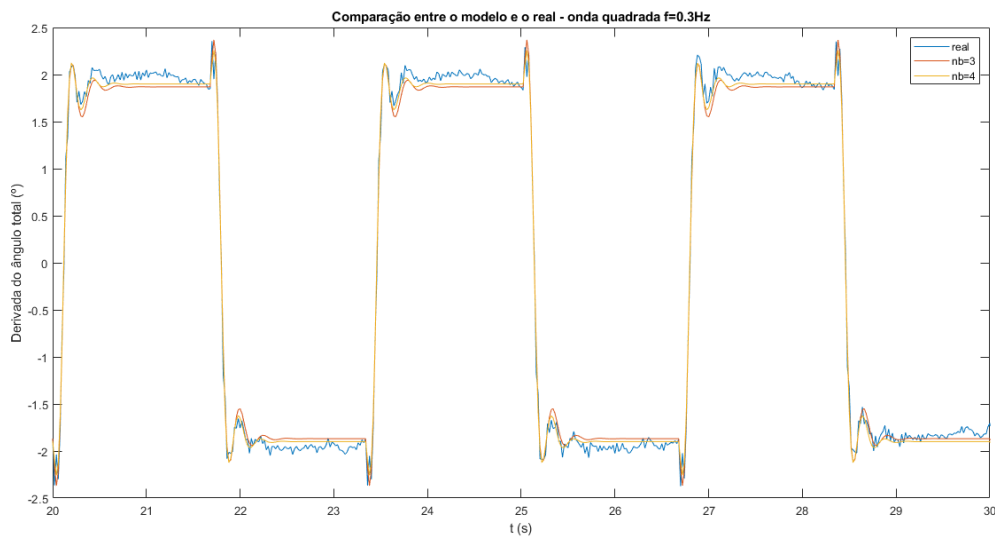


Figura 4.9: Onda quadrada - $f=0.3\text{Hz}$

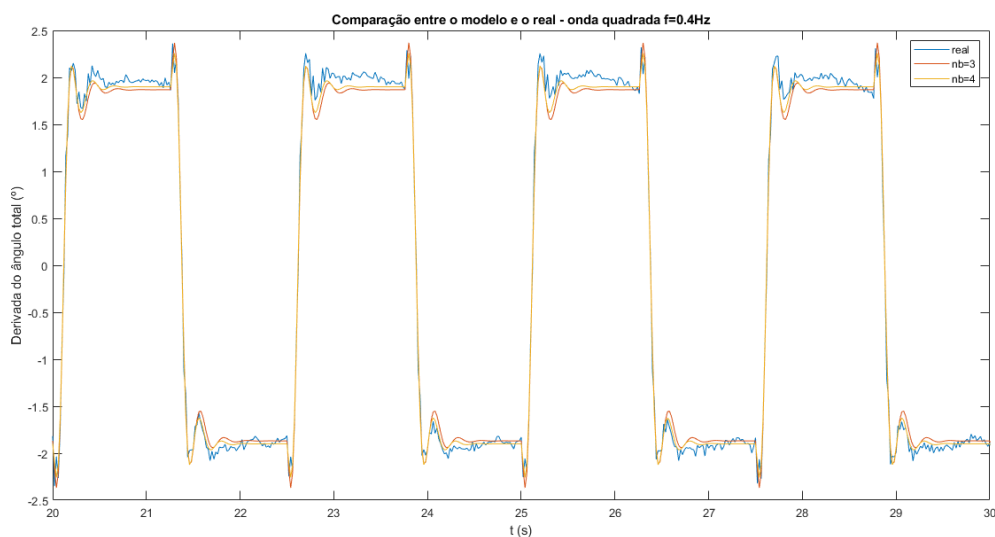


Figura 4.10: Onda quadrada - $f=0.4\text{Hz}$

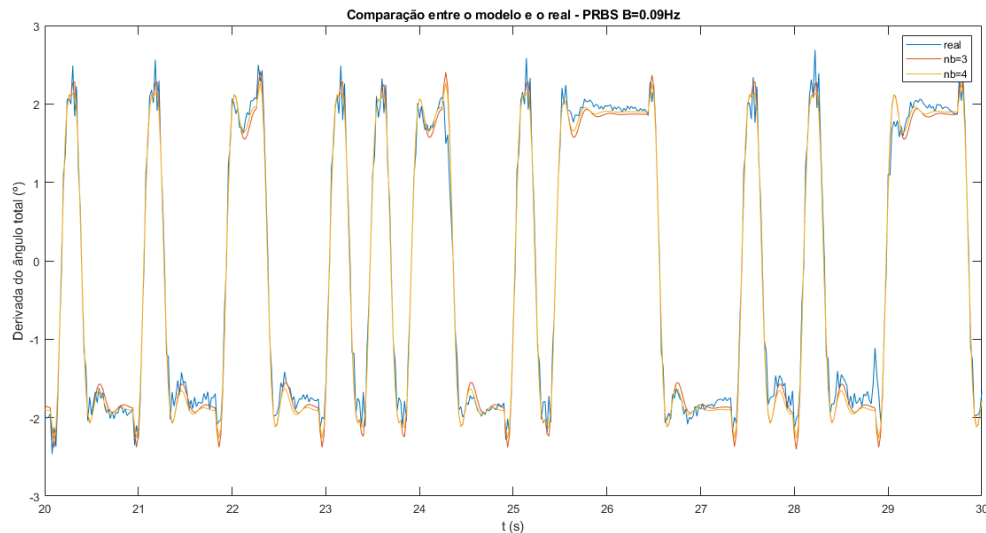


Figura 4.11: PRBS - B=0.09Hz

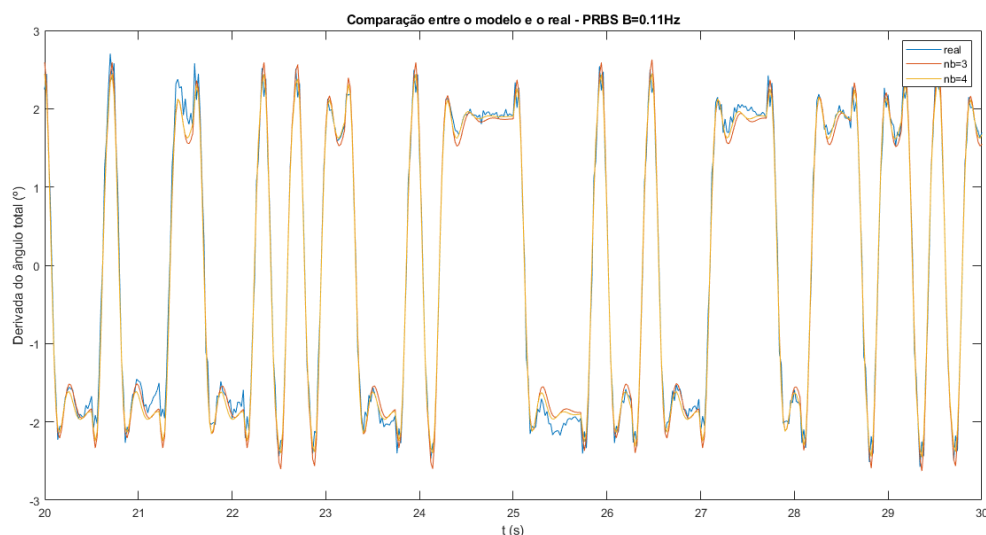


Figura 4.12: PRBS - B=0.11Hz

A evolução do modelo de ordem maior não se revela suficientemente vantajosa, ainda que mais próxima dos valores finais, para se optar pela sua utilização, uma vez que os dados para o modelo de ordem menor não se desviam significativamente dos valores esperados e opta-se por não comprometer a projeção do controlador, que poderia vir a acontecer caso se escolhesse uma ordem de modelo maior ao qual este não conseguiria responder e cuja análise fosse impraticável.

Assim, conclui-se que o modelo que mais se adapta aos objetivos deste projeto é um modelo com $n_a = n_c = 3$, $n_b = 2$ e $n_k = 1$.

Já depois de adicionar um integrador ao modelo (pólo em 1), fizeram-se novas comparações e comprova-se que continuam a ser seguidas as tendências dos ensaios, à parte de um *offset* relacionado com o facto de se ter analisado os dados tendo em conta que a barra começava na posição zero mas tal não ter acontecido no procedimento experimental. Tal está demonstrado na figura 4.13.

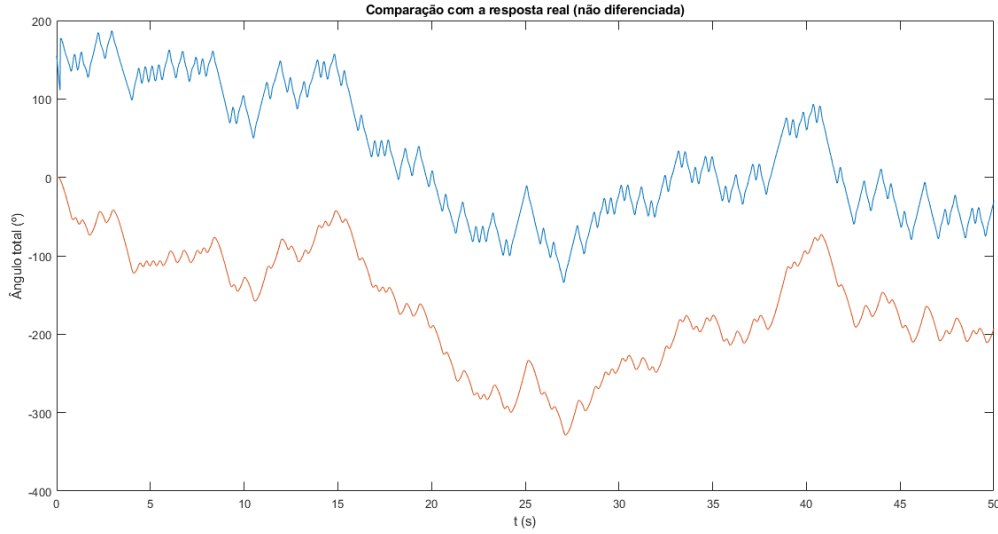


Figura 4.13: Ângulo total do ensaio e do modelo

O modelo em espaço de estados é dado por

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k), \end{aligned} \quad (4.4.1)$$

com y como o ângulo total, em grau, e u a entrada do sistema, sendo x o vetor de estado, com as matrizes

$$A = \begin{bmatrix} 3,2607 & -4,1348 & 2,4332 & -0,5591 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$C = \begin{bmatrix} -0,1554 \\ 0,2569 \\ 0 \\ 0 \end{bmatrix}, D = 0$$

A função de transferência digital associada ao modelo é, então

$$G_d(z) = \frac{-0.1554z^3 + 0.2569z^2}{z^4 - 3.261z^3 + 4.135z^2 - 2.433z + 0.5591} \quad (4.4.2)$$

cujos zeros e pólos se encontram representados no plano Z da Figura 4.14. É possível confirmar a existência de um pólo em $z = 1$, cujo efeito integrador se deve ao motor do sistema, e de um zero fora do círculo unitário. Este zero de fase não mínima está na origem do efeito de chicotada, discutido anteriormente. O comportamento oscilatório da barra é modelado pela presença do par de pólos conjugados no interior do círculo unitário.

Repare-se ainda no *root locus* da Figura 4.15, no qual se pode constatar que as margens de estabilidade do sistema são muito reduzidas, sendo, por isso, recomendável a criação de um controlador que será desenvolvido na segunda parte deste projeto laboratorial.

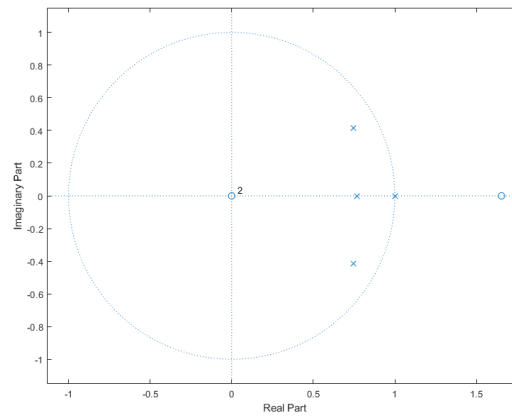
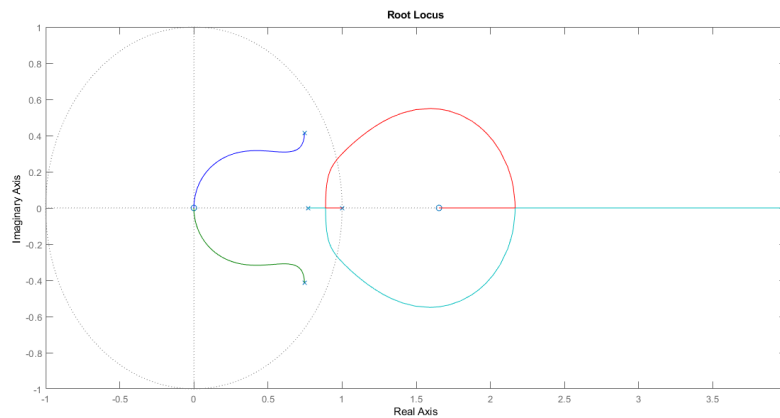


Figura 4.14: Mapa de pólos e zeros


 Figura 4.15: *Root-locus* associado ao modelo

Por fim, os diagramas de Bode do modelo revelam um comportamento quase semelhante a um filtro passa-baixo, na medida em que o ganho às baixas frequências é superior ao ganho às altas frequências. Tal semelhança é expectável dada a inclusão de um filtro passa-baixo simples no modelo, como mencionado na secção 4.2. Note-se o declive inicial de cerca de -20 dB por década no diagrama de Bode da amplitude, indicativo da presença do pólo integrador, e um ligeiro efeito de sobrelevação.

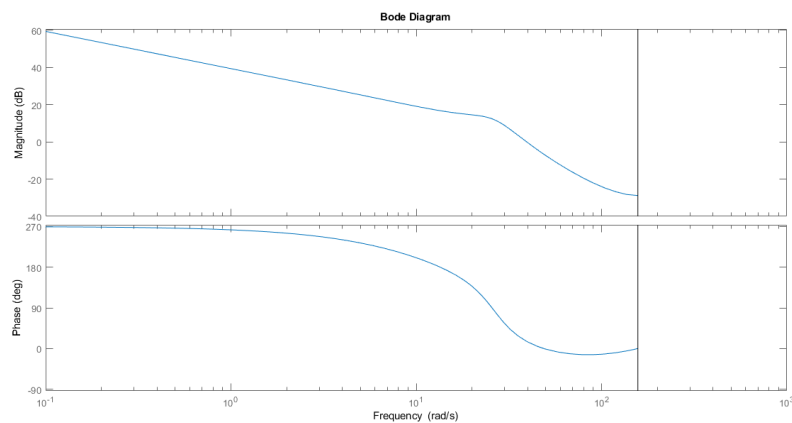


Figura 4.16: Resposta em frequência do modelo escolhido

5. Conclusão

Na primeira parte do projeto laboratorial, foram adquiridos dados do sistema físico em experiências de excitação deste com sinais de entrada distintos. Foi possível aplicar ferramentas já exploradas teoricamente para a determinação de parâmetros e identificação de modelos.

Nestes testes, revelou-se imprescindível uma calibração adequada dos sensores de forma a garantir a fiabilidade dos dados recolhidos. Aquando deste procedimento, foram utilizados métodos para a obtenção de dados que não são os mais precisos, nomeadamente a utilização do pente para a fixação da barra. No entanto, os dados recolhidos permitiram uma calibração suficientemente boa dos sensores, o que se revelou na qualidade do modelo matemático determinado para o sistema.

Tendo estes parâmetros estabelecidos, foi analisada a melhor forma para integrar as reações do sistema no modelo, adicionando-se um diferenciador e um filtro passa-baixo para a filtração dos dados das amostras e a eliminação do efeito integrador do ruído. Ao filtrar os dados, perde-se alguma informação, mas tal é necessário para realizar aproximações matemáticas. Antes disso, foram ainda truncados os dados a utilizar, por forma a não serem consideradas as descontinuidades inerentes à medições.

De seguida, para a conclusão de qual o melhor modelo, testaram-se vários parâmetros do ARMAX, tendo sido estudada qual a combinação para gerar um modelo adequado que não pusesse em causa a utilização futura deste modelo para o desenho do controlador do sistema.

Considera-se então que o modelo encontrado, com $n_a = 3$, $n_b = 2$ e utilizando o conjunto de teste correspondente a um sinal de entrada PRBS com largura de banda igual a 0.1Hz, é uma boa aproximação da dinâmica do sistema, constituindo um bom compromisso entre a ordem do modelo e a viabilidade da sua utilização. Apesar de se terem determinado *fits* médios melhores do que aquele correspondente a este modelo, não se considerou vantajoso utilizá-los, já que tal implicaria aumentar a ordem do sistema para ganhar apenas alguns pontos percentuais no *fit* do modelo.

Refere-se ainda que esta seja uma boa aproximação, há a consciência de que vários fatores podem ter conspurcado os resultados, nomeadamente as incertezas inerentes aos movimentos do *setup* aquando da realização dos ensaios, à utilização de conversores analógicos-digitais e vice versa para a excitação do sistema e visualização dos dados e até ao desgaste do conjunto utilizado.

O próximo passo consiste na aplicação ao modelo determinado de um controlador adequado e, durante as próximas sessões laboratoriais, no seu teste no sistema real. Nesta fase, poder-se-á aplicar conceitos que estão a ser explorados na segunda parte da unidade curricular.

A. Código MATLAB utilizado

A.1 *lab1_kp_kb.m*

```
1 %computing Kp and Kb
2
3 %Kp=constant with units degree/V, that relates theta and thetai
4 %theta=Kp*thetae
5 thetaemin = -4.897;
6 thetaemax = 4.917;
7
8 Kp = 360/(thetaemax-thetaemin);
9
10 %Kb=constant with units ddegree/V, that relates alpha and alphae
11 %alpha=Kb*alphae
12 alphae = [1.68 1.411 1.128 0.8545 0.5664 0.2588 -0.3418 -0.3125 -0.6738 ...
            -0.9668 -1.211 -1.494 -1.748 -2.051 -2.324 -2.617 -2.8769];
13 l = -0.6350*[8 7 6 5 4 3 2 1 0 -1 -2 -3 -4 -5 -6 -7 -8]; %0.6350 = 1/4 inch ...
            (separacao dos dentes)
14 alpha = atan(l/37.5)*180/pi;
15
16 P=polyfit(alphae,alpha,1);
17
18 Kb = P(1);
19
20 y1 = polyval(P,alphae);
21 % figure
22 % plot(alphae,alpha,'o')
23 % hold on
24 % plot(alphae,y1)
25 % hold off
26
27 clearvars -except Kb Kp
```

A.2 *cria_entradas.m*

```
1 %criacao dos estímulos a aplicar ao sistema
2 %ondas quadradas: mudar frecuencia para 0.3Hz,0.4Hz,0.5Hz
3 %PRBS: mudar largura de banda para 0.1Hz, 0.09Hz, 0.08Hz, 0.11Hz
4 t = (0:1/50:120)'; %fs=50Hz; T=120s
5 u1 = square(2*pi*0.4*t); %f=0.4Hz
6 u2 = idinput(length(t)+100, 'prbs', [0 0.1]); %b=0.1
7 u2(1:100) = []; %descartar os primeiros 2 segundos
```

A.3 *compacta_dados.m*

```
1 %gravacao dos dados de todas as experiencias numa matriz tridimensional
2 %i - numero da experiencia
3 %i=1: onda quadrada, f=0.3Hz
4 %i=2: onda quadrada, f=0.4Hz
5 %i=3: onda quadrada, f=0.5Hz
6 %i=4: PRBS, B=0.1Hz
7 %i=5: PRBS, B=0.09Hz
8 %i=6: PRBS, B=0.08Hz
```

```

9      %i=7: PRBS, B=0.11Hz
10     %j – variaveis de cada experiencia
11     %j=1: tempo (t)
12     %j=2: entrada (u)
13     %j=3: saída do potenciometro (theta)
14     %j=4: saída do extensometro (alpha)
15     %k – numero da amostra
16
17     dados=zeros(7,4,2501);
18
19     load('square_wave_03.mat', 'input_sw120_03', 'output_sw120_03');
20     dados(1,1,:) = output_sw120_03.time(1:2501);
21     dados(1,2,:) = output_sw120_03.signals.values(1:2501);
22     dados(1,3,:) = input_sw120_03.signals.values(1:2501,1);
23     dados(1,4,:) = input_sw120_03.signals.values(1:2501,2);
24
25     load('square_wave_04.mat', 'input_sw120_04', 'output_sw120_04');
26     dados(2,1,:) = output_sw120_04.time(1:2501);
27     dados(2,2,:) = output_sw120_04.signals.values(1:2501);
28     dados(2,3,:) = input_sw120_04.signals.values(1:2501,1);
29     dados(2,4,:) = input_sw120_04.signals.values(1:2501,2);
30
31     load('square_wave_05.mat', 'input_sw120_05', 'output_sw120_05');
32     dados(3,1,:) = output_sw120_05.time(1:2501);
33     dados(3,2,:) = output_sw120_05.signals.values(1:2501);
34     dados(3,3,:) = input_sw120_05.signals.values(1:2501,1);
35     dados(3,4,:) = input_sw120_05.signals.values(1:2501,2);
36
37     load('rand_wave_008.mat', 'input_rand60_008', 'output_rand60_008');
38     dados(4,1,:) = output_rand60_008.time(1:2501);
39     dados(4,2,:) = output_rand60_008.signals.values(1:2501);
40     dados(4,3,:) = input_rand60_008.signals.values(1:2501,1);
41     dados(4,4,:) = input_rand60_008.signals.values(1:2501,2);
42
43     load('rand_wave_009.mat', 'input_rand120_009', 'output_rand120_009');
44     dados(5,1,:) = output_rand120_009.time(1450:3950);
45     dados(5,2,:) = output_rand120_009.signals.values(1450:3950);
46     dados(5,3,:) = input_rand120_009.signals.values(1450:3950,1);
47     dados(5,4,:) = input_rand120_009.signals.values(1450:3950,2);
48
49     load('rand_wave_010.mat', 'input_rand120_010', 'output_rand120_010');
50     dados(6,1,:) = output_rand120_010.time(2040:4540);
51     dados(6,2,:) = output_rand120_010.signals.values(2040:4540);
52     dados(6,3,:) = input_rand120_010.signals.values(2040:4540,1);
53     dados(6,4,:) = input_rand120_010.signals.values(2040:4540,2);
54
55     load('rand_wave_011.mat', 'input_rand120_011', 'output_rand120_011');
56     dados(7,1,:) = output_rand120_011.time(1000:3500);
57     dados(7,2,:) = output_rand120_011.signals.values(1000:3500);
58     dados(7,3,:) = input_rand120_011.signals.values(1000:3500,1);
59     dados(7,4,:) = input_rand120_011.signals.values(1000:3500,2);

```

A.4 testes_polo_LPF.m

```

1  %nestes testes, utilizaram-se os dados referentes a onda quadrada com
2  %frequencia igual a 0.3Hz
3
4  t=squeeze(dados(1,1,:));
5  y=transpose(ytrend(1,:));
6  dy=diff(y);
7

```



```

8  %af=0.4
9  af = 0.4;
10 Afilt = [1 -af];
11 Bfilt = (1-af)*[1 -1];
12
13 for i=1:size(ytrend,1)
14     yf(i,:)=filter(Bfilt, Afilt, ytrend(i,:));
15 end
16
17 figure(1)
18 plot(t(1:2500),dy)
19 hold on
20 plot(t,yf(1,:))
21 xlim([10 20])
22 ylim([-5 5])
23 xlabel('Tempo [s]')
24 ylabel('Angulo total diferenciado (y') [deg]')
25 legend('y' sem filtragem', 'y' com filtragem')
26 title('af=0.4')
27 hold off
28
29 %af=0.7
30 af = 0.7;
31 Afilt = [1 -af];
32 Bfilt = (1-af)*[1 -1];
33
34 for i=1:size(ytrend,1)
35     yf(i,:)=filter(Bfilt, Afilt, ytrend(i,:));
36 end
37
38 figure(2)
39 plot(t(1:2500),dy)
40 hold on
41 plot(t,yf(1,:))
42 xlim([10 20])
43 ylim([-5 5])
44 xlabel('Tempo [s]')
45 ylabel('Angulo total diferenciado (y') [deg]')
46 legend('y' sem filtragem', 'y' com filtragem')
47 title('af=0.7')
48 hold off
49
50
51 %af=0.75
52 af = 0.75;
53 Afilt = [1 -af];
54 Bfilt = (1-af)*[1 -1];
55
56 for i=1:size(ytrend,1)
57     yf(i,:)=filter(Bfilt, Afilt, ytrend(i,:));
58 end
59
60 figure(3)
61 plot(t(1:2500),dy)
62 hold on
63 plot(t,yf(1,:))
64 xlim([10 20])
65 ylim([-5 5])
66 xlabel('Tempo [s]')
67 ylabel('Angulo total diferenciado (y') [deg]')
68 legend('y' sem filtragem', 'y' com filtragem')
69 title('af=0.75')
70 hold off

```

```

71
72
73 %af=0.8
74 af = 0.8;
75 Afilt = [1 -af];
76 Bfilt = (1-af)*[1 -1];
77
78 for i=1:size(ytrend,1)
79     yf(i,:)=filter(Bfilt, Afilt, ytrend(i,:));
80 end
81
82 figure(4)
83 plot(t(1:2500),dy)
84 hold on
85 plot(t,yf(1,:))
86 xlim([10 20])
87 ylim([-5 5])
88 xlabel('Tempo [s]')
89 ylabel('Angulo total diferenciado (y') [deg]')
90 legend('y' sem filtragem', 'y' com filtragem')
91 title('af=0.8')
92 hold off
93
94
95 %af=0.85
96 af = 0.85;
97 Afilt = [1 -af];
98 Bfilt = (1-af)*[1 -1];
99
100 for i=1:size(ytrend,1)
101     yf(i,:)=filter(Bfilt, Afilt, ytrend(i,:));
102 end
103
104 figure(5)
105 plot(t(1:2500),dy)
106 hold on
107 plot(t,yf(1,:))
108 xlim([10 20])
109 ylim([-5 5])
110 xlabel('Tempo [s]')
111 ylabel('Angulo total diferenciado (y') [deg]')
112 legend('y' sem filtragem', 'y' com filtragem')
113 title('af=0.85')
114 hold off
115
116
117 %af=0.9
118 af = 0.9;
119 Afilt = [1 -af];
120 Bfilt = (1-af)*[1 -1];
121
122 for i=1:size(ytrend,1)
123     yf(i,:)=filter(Bfilt, Afilt, ytrend(i,:));
124 end
125
126 figure(6)
127 plot(t(1:2500),dy)
128 hold on
129 plot(t,yf(1,:))
130 xlim([10 20])
131 ylim([-5 5])
132 xlabel('Tempo [s]')
133 ylabel('Angulo total diferenciado (y') [deg]')

```

```

134 legend('y' sem filtragem', 'y' com filtragem')
135 title('af=0.9')
136 hold off

```

A.5 *select_model.m*

```

1  %determinacao e validacao do melhor modelo ARMAX para identificacao do sistema
2  lab1_kp_kb
3  compacta_dados
4  clearvars -except dados Kb Kp
5
6  ytrend = zeros(7,2501);
7  yf = zeros(7,2501);
8  u = zeros(7,2501);
9  nk = 1;
10 fit_mat = zeros(7,7,6); %training set, na, nb
11 fit_aux = 0;
12
13 %calculculo de y=theta+alpha
14 for i=1:7
15     for k=1:size(dados,3)
16         ytrend(i,k)=dados(i,3,k)*Kp+dados(i,4,k)*Kb;
17     end
18 end
19
20 u=dados(:,2,:);
21
22 %definicao do filtro passa-baixo
23 af = 0.7;
24 Afilt = [1 -af];
25 Bfilt = (1-af)*[1 -1];
26
27 for i=1:size(ytrend,1)
28     yf(i,:)=filter(Bfilt, Afilt, ytrend(i,:));
29 end
30
31 for i=1:7
32     z=[yf(i,:).' u(i,:).']; %training set
33     for na=1:7
34         nc=na;
35         for nb=1:(na-nk)
36             nn=[na nb nc nk];
37             th = armax(z,nn);
38             fit_aux = 0;
39             for j=1:7
40                 if j==i
41                     continue
42                 end
43                 dados_valid = [yf(j,:).' u(j,:).']; %test set
44                 [r, fit] = compare(dados_valid,th);
45                 fit_aux = fit_aux+fit;
46             end
47             fit_mat(i,na,nb) = fit_aux/6;
48         end
49     end
50 end

```

A.6 *statespace_final.m*

```

1  %determinacao do modelo a utilizar, tendo em conta o melhor modelo
2  %identificado em select_model.m
3  %TESTE 1 na=3 nb=2 - training set 4
4  z=[yf(4,:).' u(4,:).']; %training set
5  na=3;
6  nb=2;
7  nc=na;
8  nk=1;
9  nn=[na nb nc nk];
10 th = armax(z,nn);
11
12 [den1, num1]=polydata(th);
13 [num_int1,den_int1]=eqtflength(num1,conv(den1,[1 -1]));
14
15 [A1,B1,C1,D1]=tf2ss(num_int1, den_int1);
16
17 h1=tf(num_int1, den_int1, 0.02);
18 rlocus(h1);
19 bode(h1);
20 zplane(num_int1, den_int1);
21
22 %TESTE 1 na=4 nb=2 - training set 6
23 z=[yf(6,:).' u(6,:).']; %training set
24 na=4;
25 nb=2;
26 nc=na;
27 nk=1;
28 nn=[na nb nc nk];
29 th = armax(z,nn);
30
31 [den2, num2]=polydata(th);
32 [num_int2,den_int2]=eqtflength(num2,conv(den2,[1 -1]));
33
34 [A2,B2,C2,D2]=tf2ss(num_int2, den_int1);
35
36 h2=tf(num_int2, den_int2, 0.02);
37 rlocus(h2);
38 bode(h2);
39 zplane(num_int2, den_int2);
40
41 %Plot dos graficos que comparam o modelo com o real
42 %Para fazer varias formas diferentes, tem de alterar o indice de u e yf
43 %para a respetiva
44 yfsim1=filter(num1,den1,u(2,:));
45 yfsim2=filter(num2,den2,u(2,:));
46 plot(squeeze(dados(1,1,:)),yf(2,:));
47 hold on
48 xlim([20, 30])
49 plot(squeeze(dados(1,1,:)),yfsim1)
50 hold on
51 plot(squeeze(dados(1,1,:)),yfsim2)
52 hold off

```