



*Mestrado Integrado em
Engenharia Aeroespacial
2009/2010 - Semestre de Inverno*

Controlo Ótimo e Adaptativo

TRABALHO DE LABORATÓRIO
*Identificação e Controlo Digital do Sistema de Posicionamento
de uma Barra Flexível*

Parte 1 – Controlo PID Digital dum motor DC



Preparado por

***Alexandre Bernardino, Margarida Silveira,
J. Miranda Lemos***

IST, Área Científica de Sistemas, Decisão e Controlo

Parte 1 – Controlo PID Digital de um motor DC

Objectivos: Utilizar o “Simulink” para efectuar o interface entre o computador e o processo, utilizar técnicas simples de identificação de parâmetros, implementar controladores em tempo real para um motor DC, projectar controladores PID contínuos e obter as suas versões discretas.

Planeamento: Esta parte decorre em três sessões:

1 – Na primeira sessão os alunos deverão utilizar o Simulink para efectuar o interface com a carta de I/O existente em cada computador. Pretende-se que sejam capazes de ler os sinais dos sensores, enviar comandos para o motor, visualizar os dados em tempo real e guardar dados em ficheiro. Deverão recolher dados suficientes para efectuar a calibração dos sensores, ou seja, obter os factores de conversão de volt para graus (potenciómetro) ou grau/seg (taquímetro).

2 – Na segunda sessão os alunos irão relembrar alguns conceitos de Controlo para implementar controladores proporcionais de modo a controlar o motor em posição angular. Simultaneamente serão relembrados os índices de desempenho para sistemas de primeira e segunda ordem, em termos da resposta ao escalão e resposta na frequência. A partir de experiências devidamente planeadas os alunos deverão conseguir obter os parâmetros estáticos e dinâmicos mais importantes do sistema

3 – Na terceira sessão, os alunos terão que implementar um controlador PID contínuo para controlar o motor DC. Posteriormente, farão a emulação deste controlador para o domínio discreto utilizando técnicas aprendidas nas aulas teóricas.

Não se esqueça de trazer uma pen-drive para guardar os resultados obtidos entre sessões.

Deverá trabalhar num directório em que tenha permissões de escrita. Pode ser a sua *pen*.

Parte 1 – Sessão 1 – Interface de Controlo por Computador

Introdução

Nesta primeira fase do trabalho pretende-se desenvolver um controlador simples para o posicionamento do motor. Na primeira sessão iremos aprender a utilizar o software existente para adquirir e enviar sinais para o processo, assim como métodos para gravar sinais, visualizá-los e exportá-los para o workspace do Matlab. Serão ainda obtidos sinais do processo que irão permitir efectuar a calibração dos sensores.

O Processo

O sistema a utilizar consiste num motor DC da Quanser, contendo um sensor de posição angular (potenciómetro) e um sensor de velocidade angular (taquímetro). Ambos os sensores fornecem à sua saída um valor de tensão proporcional respectivamente à posição e velocidade angular do motor (as constantes de proporcionalidade são desconhecidas à partida). O motor DC é actuado por um comando de tensão.

O Interface

O sistema é ligado a um computador equipado com uma placa I/O composta de conversores analógicos/digitais e digitais/análogos com resolução de 12 bits. A fig. 6 mostra um diagrama global de interligação do sistema.

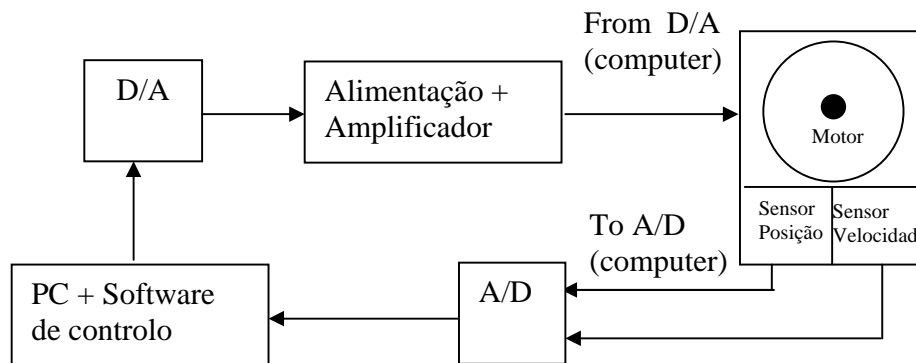


Fig. 6 – Interligação entre as diversas partes do sistema.

O sinal de comando para o motor encontra-se ligado ao canal 1 do conversor D/A. O potenciómetro e o taquímetro estão ligados, respectivamente, aos canais 1 e 3 do conversor A/D.

O Software de aquisição de dados e controlo

Para efectuar a aquisição de sinais e o controlo do processo irá ser utilizado o software “Matlab”+”Simulink” com as *toolboxes* ”Real-Time Workshop”+”Real-Time Windows Target”. Com o “Simulink” é possível projectar sistemas de aquisição de sinais e de controlo através de um interface gráfico simples e com bastantes funcionalidades. O

“Real-Time Workshop” permite exportar para C e compilar os modelos criados em “Simulink”. Finalmente, o “Real-Time Windows Target” permite correr os modelos em tempo-real, no sistema operativo Windows. Embora não seja necessário para a execução do trabalho, é aconselhada ao aluno a consulta dos manuais destas ferramentas de software, disponibilizadas na página web da disciplina. Nas seguintes linhas, procede-se a uma breve descrição do software a utilizar.

Simulink:

O “Simulink” é um ambiente gráfico de simulação que trabalha em conjunção com o Matlab e permite efectuar simulações de sistemas dinâmicos em tempo não real (simulado). Existem diversas bibliotecas de componentes (blocksets) que podem ser interligados de uma forma fácil e intuitiva. Em particular, existem componentes que permitem simular sistemas dinâmicos, geradores de sinais, visualizadores de sinal e comunicar com diversas placa de I/O, entre as quais se encontra a placa a ser utilizada neste trabalho (indica-se mais abaixo o procedimento para identificar o modelo da placa utilizada). No entanto, o Simulink por si só não permite efectuar as simulações sincronizadamente com um relógio real, pelo que é necessário recorrer às ferramentas “Real-Time Workshop” e “Real-Time Windows Target”. Consulte o manual de utilizador do Simulink, disponível na página web da disciplina.

Real-Time Workshop:

Esta ferramenta trabalha sobre o Simulink e permite exportar os modelos de Simulink para código C adaptado a diversas plataformas de hardware e software. O código, depois de compilado, gera um programa executável que corre o modelo simulink num processo separado, sem interface gráfico. É possível, no entanto guardar e visualizar os sinais disponíveis no modelo, assim como alterar as propriedades dos blocos, utilizando o Simulink em modo externo. Neste modo, o Simulink pode comunicar com o processo executável para obter sinais e reconfigurar parâmetros da simulação.

Real-Time Windows Target:

Com esta ferramenta, o código C gerado pelo Real-Time Workshop pode ser compilado para a plataforma Windows e executado em tempo real. Para uma boa compreensão do processo de criação de código em tempo real, utilizando todas as ferramentas atrás descritas, é essencial a consulta do manual de utilização do Real-Time Windows Target, disponível na página web da disciplina. Aconselha-se a leitura do capítulo “Basic Procedures”.

Tarefas a efectuar no laboratório

Esta sessão introdutória está formatada de um modo tutorial, pelo que poucos conhecimentos prévios de Matlab/Simulink serão necessários. Todos os passos necessários para configurar as simulações estão bastante detalhados.

Uma vez que alguns dos procedimentos serão repetidos várias vezes ao longo do trabalho, aconselha-se vivamente aos alunos que elabore, uma “check-list” com base nas tarefas efectuadas, sintetizando os procedimentos necessários para configurar uma simulação em tempo-real, configurar a gravação de dados no workspace, configurar os elementos visualizadores, etc. Com estes procedimentos sistematizados, a eficiência do trabalho será muito maior.

A – Geração, visualização e gravação de sinais.

1. Abrir o SIMULINK. Abra o MATLAB (Versão 7) e execute o comando `<<simulink>>`. Será aberta a janela “Simulink Library Browser”
2. Criar um novo modelo. No menu do “Simulink Library Browser”, escolher File-New-Model, ou premir CTRL+N. Abre-se uma janela gráfica onde se vão inserir e interligar os diversos blocos operacionais.
3. Inserir e configurar um bloco gerador de sinais. Na árvore de selecção existente à esquerda da janela do “Simulink Library Browser” escolher “Simulink-Sources”. Arrastar um elemento “Signal Generator” para a janela do modelo. Abra o elemento (duplo clique) e configure-o para gerar uma onda quadrada de amplitude 2 volt e frequência 2 Hz.
4. Inserir e configurar um bloco visualizador de sinais. Na árvore de selecção escolha “Simulink-Sinks” e arraste um elemento “Scope” para a janela do modelo. Abra o elemento e clique no elemento da *toolbar* “Parameters” (segundo a contar da esquerda). Na *tab* “General” configure “Tick Labels-All”, “Sampling-Sample Time” e escolha como período de amostragem 1 milisegundo (0.001). Na *tab* “Data History” remova a selecção “Limit data points to last” e seleccione “Save data to workspace”. Os dados apresentados neste elemento serão exportados para o workspace do matlab no fim da simulação. Escolha o nome da variável “input” e o formato de dados “Structure with time”.
5. Configurar e correr a simulação. Ligue o elemento “Signal Generator” ao elemento “Scope”. Inicie a simulação (menu “Simulation-Start” ou CTRL-T). Observe o resultado da simulação no elemento visualizador “Scope”.
6. Visualizar sinais no workspace. Confirme que, no workspace do matlab, tem a variável “input”. Visualize o sinal com o comando “plot” e confirme que é idêntico ao observado no elemento “Scope”: `>>plot(input.time,input.signals.values)`
7. Gravar dados do workspace. Guarde os dados em ficheiro (ver comando “save”) para poder gerar o gráfico “off-line”.

B – Envio de comandos para o motor

Atenção: Ao inserir os cabos de ligação, faça-o de modo a que estes fiquem completamente encaixados, estabelecendo uma ligação eléctrica perfeita.

8. Inserir um bloco D/A para enviar comandos para o motor. Na árvore de selecção seleccionar “Real-Time Windows Target” e arrastar o elemento “Analog Output” para a janela do modelo criado.
9. Configurar o bloco D/A. Para ver a placa de aquisição de dados em uso na sua bancada de laboratório, corra o utilitário “**Measurement & Automation**” que pode encontrar no *Desktop*. Na árvore de directórios que aparece no lado esquerdo da janela deste utilitário escolher

Devices and interfaces

NI-DAQmx Devices

Aqui aparece o tipo da placa (ex. NI PCI-6221)

Tome nota do tipo da placa. Em seguida, abrir o bloco D/A e escolher a carta identificada por este procedimento. Definir o período de amostragem (sample time) em 1 milissegundo (0.001). Definir o canal de saída a 1. As gamas de tensão deverão estar configuradas entre -10V e 10V e o tipo de dados como “tensão”. Os valores iniciais e finais deverão ser definidos a 0. Ligar o gerador de sinais a este bloco.

10. Configurar a simulação para funcionar em tempo-real. No menu “Simulation” seleccionar “Configuration Parameters” ou CTRL-E. Abre-se uma janela de diálogo. Na opção “Solver -> Type” escolher “Fixed-step”. Na opção “Hardware Implementation -> Device Type” escolher “32-bit Real-Time Windows Target”. Na opção “Real-Time Workshop -> RTW system target file” escolher “rtwin.tlc”.
11. Configurar Simulink em modo externo. No menu “Tools” escolher “External Mode Control Panel...”. Abre-se uma janela de diálogo. Seleccionar “Signal & Triggering”. Abre-se uma nova janela. Na opção “Duration” introduzir o numero de amostras total da simulação, neste caso 10 seg X 1000 amostras/seg = 10000 amostras. Voltar à janela do modelo e no menu “Simulation” escolher “External”.
12. Compilar o modelo. No menu “Tools” escolher “Real-time Workshop – Build Model” ou CTRL-B. Espere que a compilação termine e verifique se não há erros.
13. Correr o modelo. No menu “Simulation” escolher “Connect to target”. Pedir ao docente do laboratório para verificar a operação do sistema e iniciar a operação.
14. Observe a operação do motor e anote o seu comportamento.

C – Leitura de sinais dos sensores.

15. Inserir bloco A/D para ler os valores dos sensores. Na árvore de selecção seleccionar “Real-Time Windows Target” e levar dois elementos “Analog Input” para o modelo.
16. Configurar o bloco A/D. Abrir o bloco e escolher a carta de aquisição adequada. Definir o período de amostragem (sample time) em 1 milissegundo (0.001). Definir o canal de entrada 1 (potenciómetro) para um dos blocos e 3 (taquímetro) para o outro. As gamas de tensão deverão estar configuradas entre -10V e 10V e o tipo de dados como “tensão”.

17. Inserir dois blocos “Scope” para visualizar os sinais dos sensores. Configurá-los tal como no ponto 4. Dar-lhes os nomes “tensao_pot” e “tensao_taq”, respectivamente.
18. Recompilar o modelo.
19. Correr o modelo. Confirme com o docente se está tudo bem, antes de dar início à simulação. Observe os gráficos relativos à tensão do potenciómetro e do taquímetro.
20. Gravar dados. Certifique-se que as variáveis “tensao_pot” e “tensao_taq” estão disponíveis no workspace do matlab.

D – Calibração dos sensores.

21. Aplique um escalão unitário (1 volt) à entrada do motor. Adquira e grave as tensões obtidas à saída do potenciómetro e do taquímetro. A partir dos dados obtidos, calcule os factores de calibração Kp [grau volt⁻¹] e Kt [grau seg⁻¹ volt⁻¹], que permitam converter as medições dos sensores em volt para graus e grau/seg, respectivamente. Pode observar na figura seguinte (fig. 7) o funcionamento do potenciómetro. O veio do motor está rigidamente ligado ao veio do potenciómetro (ponto 4). O elemento resistivo é alimentado por uma tensão de +10V (ponto 5) e -10V (ponto 9). O cursor é construído de um material condutor e desloca-se sobre o elemento resistivo, podendo rodar continuamente uma vez que não tem batentes mecânicos (fim de curso). Em função da posição do cursor, a tensão de saída (ponto 6) tem um valor entre -10V e + 10V. Note que a tensão medida repete-se a cada rotação inteira do cursor.

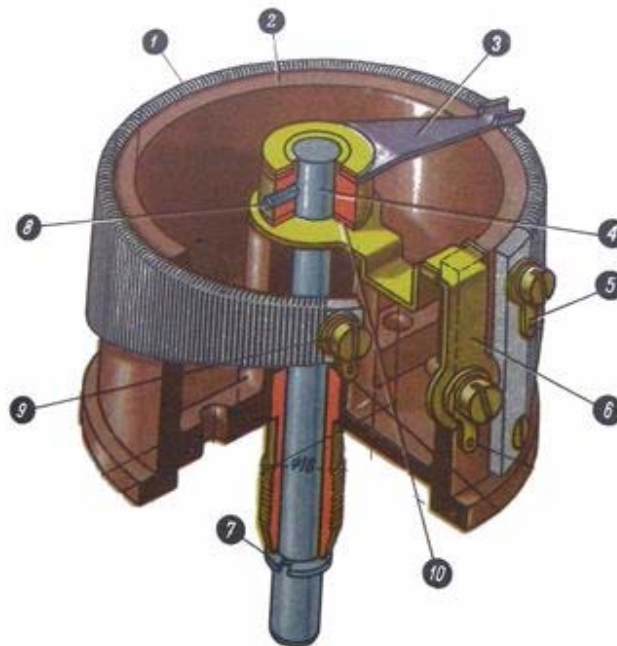


Figura 7. Esquema interno de um potenciómetro.

Nota final: Grave e guarde o modelo criado. Este poderá ser utilizado como “template” para as próximas sessões de laboratório.

Incluir no Relatório

- Comente o funcionamento do motor. Nomeadamente explique porque é que a sua posição angular nunca estabiliza, apesar do seu sinal de comando ser constante.
- Explique o procedimento efectuado para efectuar a calibração dos sensores. Apresente:
 - Diagrama de blocos Simulink utilizado.
 - Gráficos dos sinais de comando, e dos sensores.
 - Cálculos e gráficos auxiliares para obter os valores dos factores de conversão Kp [grau volt⁻¹] e Kt [grau seg⁻¹ volt⁻¹].
- Explique a relação existente entre os sinais lidos do potenciómetro e do taquímetro. Será que um deles se poderia obter a partir do outro? Justifique.

Nota: Não se esqueça de apresentar as unidades em ambos os eixos dos gráficos e legendar devidamente quer os gráficos quer os diagramas.

Parte 1 – Sessão 2 – Controlo Proporcional

Estudo Prévio

O modelo de um motor DC (contínuo).

Um motor DC contínuo, considerando a sua saída em velocidade angular e sem variações de carga, pode ser aproximado por um sistema linear e invariante no tempo (SLIT), de primeira ordem, com a seguinte função de transferência:

$$G_{\omega}(s) = \frac{G_0}{1 + s\tau}$$

Esta função de transferência está escrita na forma das “constantes de tempo”. O parâmetro G_0 é designado por “ganho estático” e o parâmetro “ τ ” é designado “constante de tempo”.

A função de transferência aparece por vezes escrita na forma “polos-zeros”:

$$G_{\omega}(s) = \frac{G}{s + p}$$

Nesta forma o parâmetro p é o simétrico do pólo e o parâmetro G é um ganho (diferente do ganho estático no caso geral). Têm-se as seguintes igualdades:

$$G = G_0 p, \quad p = \frac{1}{\tau}$$

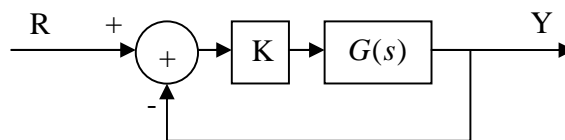
Quando se considera a saída do motor como a posição angular, inclui-se um modo integrador no modelo. A função de transferência neste caso é dada por:

$$G_{\theta}(s) = \frac{G_0}{s(1 + s\tau)} = \frac{G}{s(s + p)}$$

Os parâmetros G e G_0 não são necessariamente os mesmos do caso anterior.

Controlo proporcional.

Um dos métodos mais simples de controlo é por realimentação proporcional. O seguinte diagrama de blocos ilustra este tipo de controlo:



A variável Y é a saída do sistema e a variável R é a referência para a qual se pretende controlar o sistema.

Sob este esquema de realimentação, a função de transferência de R para Y (função de transferência em malha fechada) é dada por:

$$H(s) = \frac{KG(s)}{1 + KG(s)}$$

Erro estático de seguimento.

Um dos indicadores importantes num sistema de controlo é o erro de seguimento em regime estacionário. Este erro é medido como a diferença relativa entre o valor de uma referência constante e o valor para o qual estabiliza a resposta do sistema. Em percentagem pode ser calculado como:

$$E_{ss}(\%) = \frac{r - y_{\infty}}{r} \times 100\%$$

Controlo de velocidade.

Quando a variável realimentada é a velocidade angular do motor, a função de transferência em cadeia fechada, na forma das constantes de tempo, é:

$$H_{\omega}(s) = \frac{\frac{KG_0}{1 + KG_0}}{1 + s \frac{\tau}{1 + KG_0}}$$

ou, na forma pólos-zeros:

$$H_{\omega}(s) = \frac{KG}{s + KG + p}$$

Controlo de posição:

Se a variável realimentada corresponde à posição angular do motor, a função de transferência em malha fechada virá dada por:

$$H_{\theta}(s) = \frac{\frac{KG_0}{\tau}}{s^2 + \frac{s}{\tau} + \frac{KG_0}{\tau}} = \frac{KG}{s^2 + ps + KG}$$

Índices de desempenho

No projecto de sistemas de controlo, alguns índices permitem avaliar da qualidade do projecto. As figuras seguintes mostram a resposta ao escalão unitário e o diagrama de bode de sistemas lineares e invariantes no tempo. As figuras 8 e 9 ilustram alguns destes índices de desempenho no domínio do tempo e da frequência.

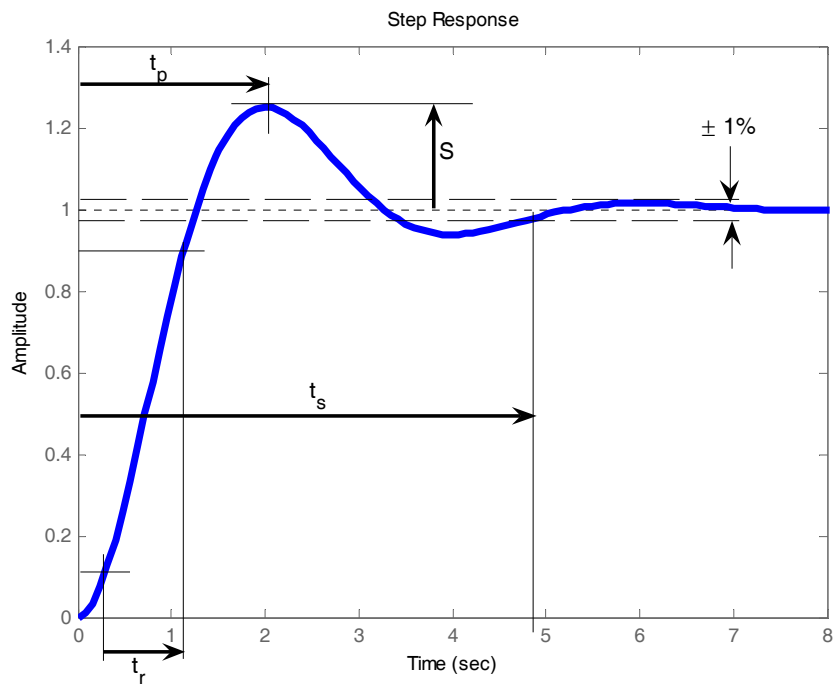


Fig. 8 – Índices de desempenho na resposta no tempo.

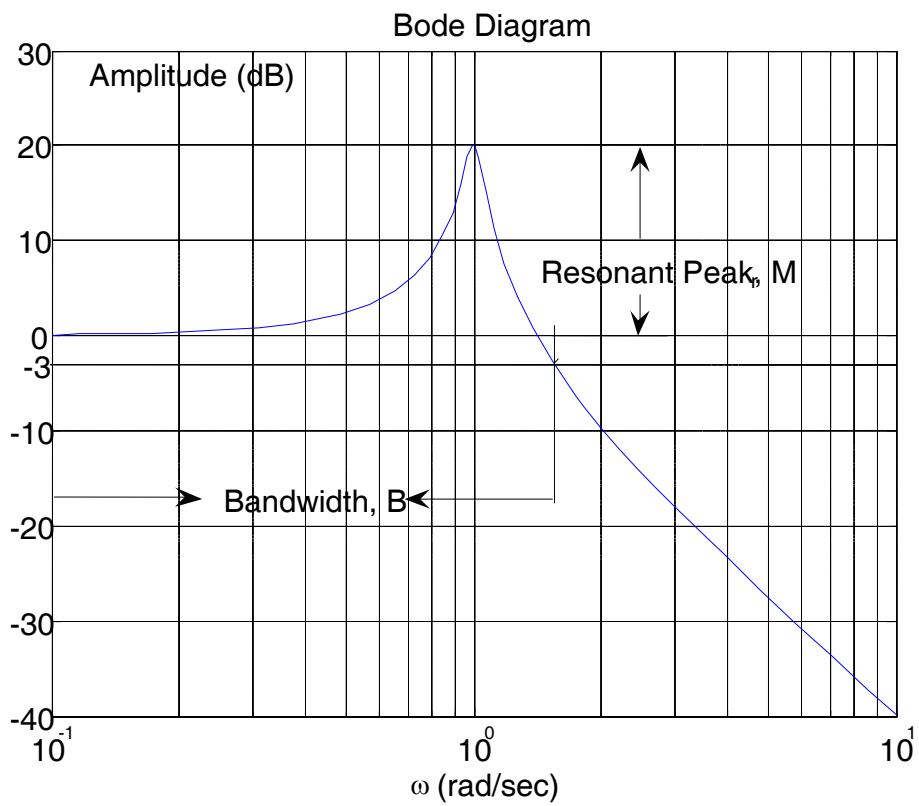


Fig. 9 – Índices de desempenho da resposta em frequência.

Iremos considerar os seguintes índices relativos à resposta a um escalão unitário:

- Tempo de estabelecimento a 1% (t_s) - é o tempo que o sistema leva a estabilizar em torno de 1% do valor final.
- Tempo de crescimento (t_r) - é o tempo que demora a resposta a passar de 10% para 90% do valor final.
- Sobrelevação (S) - é a quantidade em que a resposta excede, no máximo, o valor final. É dado em percentagem.
- Tempo de pico (t_p) - é o instante de tempo em que sucede o máximo da função.

No diagrama de bode, consideram-se os seguintes valores notáveis:

- Frequência de Ressonância (ω_r) - é o valor da frequência para a qual se dá o valor máximo da amplitude.
- Pico de ressonância (M_p) - é o valor pelo qual o máximo excede o valor de patamar (em dB).
- Largura de Banda de -3dB (B) - é a frequência a partir da qual a amplitude se torna inferior em 3 dB ao valor de patamar.

Sistemas de 1ª ordem sem zeros

Estes sistemas têm função de transferência do tipo:

$$G(s) = \frac{G}{s + p}$$

Para estes sistemas, os índices de desempenho tomam os seguintes valores:

- $t_r \approx \frac{2.2}{p}$
- $t_s \approx \frac{4.6}{p}$
- $B = p$

Os outros índices não estão definidos.

Sistemas de 2ª ordem com pólos complexos conjugados (e sem zeros)

Considere-se a função de transferência:

$$G(s) = \frac{G}{s^2 + 2\xi\omega_n s + \omega_n^2}$$

O parâmetro ξ denomina-se “factor de amortecimento” e ω_n denomina-se “frequência natural”. Estes parâmetros estão relacionados com os pólos do sistema de acordo com o diagrama da fig. 10.

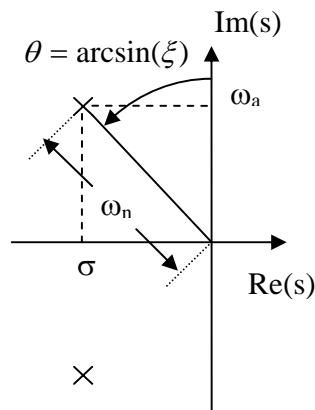


Fig. 10 – Relação entre a posição dos pólos e os parâmetros de um sistema linear de 2ª ordem sem zeros.

O factor de amortecimento determina o tipo de comportamento do sistema em termos de resposta ao escalão:

$\xi = 0$

Regime criticamente estável. O sistema tem dois pólos complexos conjugados sobre o eixo imaginário. O sistema apresenta oscilações não amortecidas e pode tornar-se instável se excitado com entradas sinusoidais de frequência ω_n .

$0 < \xi < 1$

Regime sub-amortecido. O sistema tem pólos complexos conjugados e apresenta sobrelevação e oscilações amortecidas.

$\xi = 1$

Regime criticamente amortecido. O sistema tem dois pólos reais idênticos. Não tem sobrelevação nem oscilações. Converte “rapidamente” para o valor final.

$\xi > 1$

Regime sub-amortecido. O sistema tem dois pólos reais distintos, um deles cada vez mais lento à medida que ξ aumenta, e que determina a dinâmica dominante do sistema.

$\xi < 0$

Sistema instável.

Nos dois primeiros casos $0 \leq \xi < 1$, os índices de desempenho têm os seguintes valores:

- $t_r \approx \frac{1.8}{\omega_n}$
- $t_s \approx \frac{4.6}{\sigma}$
- $B \approx \omega_n$

- $t_p = \frac{\pi}{\omega_a}$
- $\omega_r = \omega_n \sqrt{1 - 2\xi^2}$
- $M_r = \frac{1}{2\xi \sqrt{1 - \xi^2}}$
- $S = e^{-\frac{\xi\pi}{\sqrt{1-\xi^2}}} * 100$

Nos dois casos seguintes $\xi \geq 1$, o sistema aproxima-se de um comportamento de primeira ordem, considerando o pólo mais lento.

Tarefas Preparatórias

Para os modelos genéricos de posição e velocidade de um motor DC, calcule o valor do erro de seguimento em regime estacionário com um controlador proporcional. Apresente o resultado em função dos parâmetros do sistema e do ganho proporcional de controlo.

Tarefas a efectuar no laboratório

NOTA IMPORTANTE: Utilize sempre valores calibrados para as medições dos sensores, utilizando os ganhos de conversão obtidos na última sessão.

A – Identificação de Parâmetros pela Resposta no Tempo

1. Utilizando o ficheiro Simulink que desenvolveu na sessão de laboratório anterior, aplique um escalão ao sistema e obtenha a resposta ao escalão do taquímetro. Garanta que a região do período transitório se encontra bem definida.
2. A partir da resposta ao escalão obtenha os valores para o ganho estático (G_0) e constante de tempo do motor (τ). Repare que deve exprimir o valor de G_0 em unidades de [grau].

B – Controlo de Velocidade Angular

3. Modifique o ficheiro para efectuar um sistema de controlo proporcional com retroacção da velocidade angular. Irá necessitar de blocos de ganho e blocos somadores que poderá encontrar na árvore de selecção do “Simulink Library Browser” em “Simulink->Math Operations”.
4. Como referência utilize uma onda quadrada de amplitude 200 grau s⁻¹ e frequência 0.2 Hz.
5. Coloque elementos visualizadores (Scope) de modo observar e guardar os sinais de erro e de comando do motor (para além dos valores da referência e da saída).
6. Compile e corra o modelo com os seguintes valores para o ganho proporcional de controlo: $K = 0.01$; $K = 0.04$. Observe e guarde os resultados.

7. Estabeleça o valor de K para que o erro de seguimento seja de 10%. Designe este valor por K_{10} .
8. Compile e corra o modelo com esse valor de K. Observe e guarde os resultados.
9. Verifique que todos os resultados estão devidamente guardados porque serão necessários para a elaboração do relatório.

C – Controlo de Posição angular

10. Repita os passos 3 a 5 para uma realimentação de posição. Utilize referências de amplitude de 45 graus.
11. Compile e corra o modelo com os seguintes valores para o ganho proporcional de controlo: $K = 0.05$; $K = 0.5$. Observe e guarde os resultados.
12. Repita a experiência com $K = 0.5$, alterando a amplitude da referência para 5° . Observe as diferenças e guarde os dados.
13. Utilize agora um sinal “Chirp” de amplitude 5° como referência. Este bloco encontra-se sob “Simulink->Sources”. Configure o bloco com “Initial Frequency” a 1 Hz, “Target Time” a 40 seg, e “Frequency at target time” a 40Hz. Terá que colocar em série um bloco de ganho para estabelecer a amplitude do “chirp”.
14. Altere o tempo da simulação para 40 seg e a duração do buffer de dados em “Tools->External Mode Control Panel -> Signals & Triggering -> Duration” para um valor adequado. Compile e corra o modelo, guarde os resultados.
15. Verifique que todos os resultados estão devidamente guardados porque serão necessários para a elaboração do relatório.

Incluir no Relatório

- Tarefas preparatórias
- Tarefa A:
 - Gráfico da resposta ao escalão.
 - Procedimento para a obtenção dos parâmetros do motor.
 - Valores obtidos para G_0 e τ . Não esquecer as unidades.
- Tarefa B:
 - Diagrama de blocos Simulink.
 - Ponto 6 - Para cada valor de K testado, apresente os gráficos com os sinais de referência, taquímetro e sinal de comando. Obtenha estimativas dos indicadores de desempenho relativos à resposta ao escalão, e do erro de seguimento em regime estacionário. Preencha a seguinte tabela:

	$E_{ss} [\%]$	$t_r [ms]$	$t_s [ms]$
$K = 0.01$			
$K = K_{10} =$			
$K = 0.04$			

Compare os valores estimados com os teóricos.

- Ponto 7 - Explique o procedimento efectuado para determinar o valor de K.

- Ponto 8 – Apresente os gráficos com os sinais de referência, taquímetro e sinal de comando. Comente os resultados.
- Tarefa C:
 - Diagrama de blocos Simulink.
 - Pontos 11 e 12 - Para cada valor de K testado, apresente os gráficos com os sinais de referência, potenciômetro e sinal de comando. Obtenha estimativas dos indicadores de desempenho relativos à resposta ao escalão e do erro de seguimento em regime estacionário. Indique os casos em que estas estimativas não sejam fiáveis. Sumarize os seus resultados na tabela seguinte:

	$E_{ss} [\%]$	$t_r [ms]$	$t_s [ms]$	$t_p [ms]$	$S [\%]$
$K = 0.05$					
$K = 0.5$					

Compare com os valores teóricos. Compare os resultados obtidos com os dois valores do ganho

- Pontos 13,14 - Apresente os gráficos com os sinais de referência, potenciômetro e sinal de comando. Sabendo que a variação da frequência no sinal “Chirp” é linear, esboce com base na resposta ao sinal chirp o diagrama de resposta em frequência de amplitude do sistema e obtenha estimativas para os índices de desempenho na frequência. Compare com os valores teóricos obtidos usando a função *bode*.

Parte 1 – Sessão 3 – Controle PID Digital

Estudo Prévio

Filtros de redução de ruído e filtros derivadores.

Muitas vezes os sinais obtidos dos sensores vêm contaminados de ruído de alta frequência que pode provocar efeitos indesejados no sinal de controle. No caso do motor utilizado nos laboratórios é perigosa a introdução de sinais com frequências superiores a 50Hz porque provoca correntes de alta frequência nos enrolamentos e aquecimento indesejável. Para evitar estes problemas, deverão ser utilizados filtros passa baixo de largura de banda suficientemente baixa para obter sinais de comando pouco ruidosos, mas que não introduzam demasiado atraso no sistema.

Nos casos mais comuns utilizam-se filtros de primeira ordem, que são simples de dimensionar e implementar. No caso contínuo, para redução de ruído poderemos utilizar os filtros da forma:

$$F_{lp}(s) = \frac{a}{s + a}$$

Estes filtros têm largura de banda $B = a$ rad/sec e não introduzem ganho estático no sistema.

Nalgumas situações é útil considerar filtros derivadores. Por exemplo, para implementar um controlador PID é necessário calcular a derivada do sinal. No entanto uma derivada pura não é causal e amplifica muito o ruído de alta frequência. Por isso, inclui-se também um pólo conjuntamente com o derivador para permitir a implementação causal e reduzir o efeito do ruído:

$$F_{bp}(s) = \frac{as}{s + a}$$

Discretização de Sistemas Contínuos por ZOH

A maioria dos dispositivos conversores D/A funcionam por manutenção dos valores dos sinais durante um período de amostragem (Retentores de Ordem Zero – *Zero-Order-Hold*). Assim, os sinais enviados para o sistema contínuos são seccionalmente constantes e podem ser vistos como “degraus” entre os instantes de amostragem. Nestas condições, o método de discretização de sistemas “Escalaço invariante” permite obter modelos discretos de sistemas contínuos cuja resposta equivale exactamente ao valor da resposta do sistema contínuo nos instantes de amostragem

Método do escalaço invariante:

$$G_d(z) = (1 - z^{-1})Z \left\{ L^{-1} \left[\frac{G_c(s)}{s} \right]_{t=kh} \right\}$$

Embora saibamos exactamente o que sucede nos instantes múltiplos do período de amostragem ($t = kh$), o comportamento do sistema contínuo entre os instantes de amostragem não é definido à partida. Para que o comportamento seja suave, então a frequência de amostragem deverá ser suficientemente alta em relação ao comportamento dinâmico do sistema na frequência. Como regra empírica, o período de amostragem deverá ser tal que se obtenham 4 a 10 amostras do sinal durante o seu tempo de crescimento:

$$N_r = \frac{t_r}{h} \approx 4 \text{ a } 10$$

(Regra retirada de “Computer Controlled Systems (3rd ed)”, Astrom and Wittenmark, pag. 66-67).

Tarefas Preparatórias

A -Simulação do controlador PID contínuo

1. Em primeiro lugar iremos criar um bloco Simulink para implementar um controlador PID. Apesar de existirem já blocos de controlo PID no *Simulink*, o bloco que iremos desenvolver incluíra o termo de separação da resposta a referências e a perturbações (*b*). Não iremos considerar nesta sessão a utilização de um *anti-windup*. Implemente um diagrama de blocos Simulink de acordo com a fig. 11 (ver o ponto 2 a seguir sobre como criar um bloco no SIMULINK).

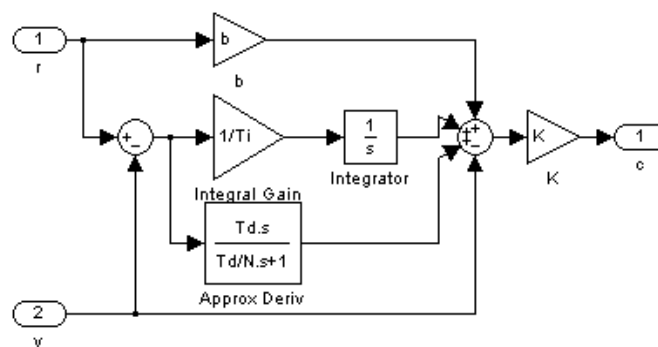


Fig. 11 – Diagrama SIMULINK de um PID contínuo.

Este controlador tem a seguinte função de transferência:

$$U(s) = K \left(bU_c(s) - Y(s) + \frac{1}{T_i s} E(s) + \frac{sT_d}{1 + sT_d / N} E(s) \right)$$

Deverá configurar cada um dos blocos de ganho e de função de transferência com as variáveis indicadas acima (*b*, *T_i*, *T_d*, *N*, *K*), que correspondem aos parâmetros do controlador PID pretendido.

2. No editor Simulink, seleccione todos os elementos do diagrama anterior e faça *Edit – Create Subsystem*. Este comando irá criar um bloco único contendo a função de transferência dada.

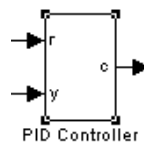


Fig. 12 – Criação de um bloco PID no SIMULINK.

3. Seleccione este bloco e faça *Edit – Edit Mask*. Surgirá uma janela de diálogo na qual deverá escolher a sub-janela *Parameters*. Aqui deverá adicionar linhas para permitir a

edição dos parâmetros K , T_d , T_i , N , e b . Daqui em diante, abrindo o bloco do controlador, surgirá uma janela de diálogo que permitirá alterar os parâmetros.

4. Construa um simulador para o Motor DC controlado com o controlador PID implementado, atrás de acordo com a seguinte fig. 13:

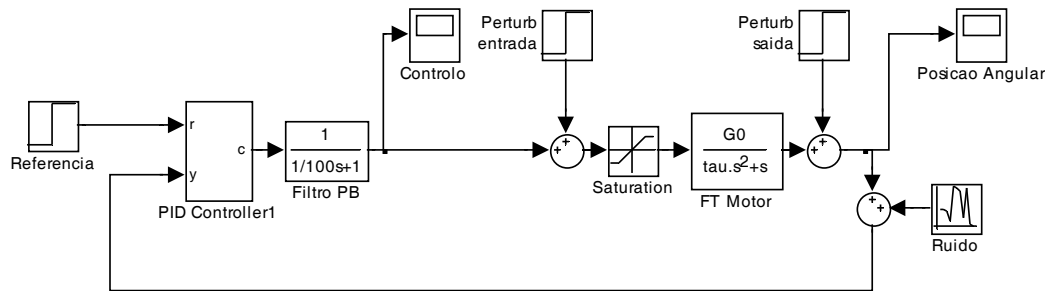


Fig. 13 – Simulação do controlo do motor com o PID.

Os blocos têm o seguinte significado:

Referência – escalão unitário ocorrendo no instante 1 seg.

Perturb saída – escalão de amplitude 0.1 ocorrendo no instante 3 seg, correspondendo, por exemplo a uma perturbação de carga no motor.

Perturb entrada – escalão de amplitude 0.1 ocorrendo no instante 2 seg, correspondendo, por exemplo, a uma variação de tensão de alimentação.

Filtro PB – filtro de largura de banda 100 rad/s (~15 Hz) que reduz o nível de ruído à entrada do motor. O fabricante do motor recomenda a não introdução de componentes de frequência superiores a 50Hz para não danificar o sistema. Um filtro deste tipo deverá ser sempre utilizado à entrada do motor nas restantes sessões de laboratório.

PID Controller – Controlador com a função de transferência pretendida. Deveremos fixar o valor N a 10, para limitar a 10 o ganho do derivador nas altas frequências.

Saturation – bloco que implementa a limitação da tensão de entrada do motor entre -10V e +10 V.

FT Motor – Modelo do motor. Deverá utilizar os parâmetros do ganho estático e constante de tempo identificados nas sessões anteriores.

Ruído - ruído aditivo Gaussiano, modelando o ruído do sensor, de desvio padrão 0.01 (usar o bloco Simulink “*Random Number*”).

5. Utilize o método de Ziegler-Nichols da sensibilidade última para obter valores iniciais para os parâmetros K , T_i e T_d . Este método consiste em utilizar um ganho proporcional e aumentá-lo sucessivamente até que se obtenha uma oscilação de amplitude constante na saída. Repare que, neste caso, como a fase de alta frequência é -180° , esta situação

nunca é obtida exactamente. Pode no entanto aproximá-la para ganho elevado, por forma a obter uma oscilação com muito pouco amortecimento. Uma vez determinado este ganho, designado K_u , e medido o período das oscilações da saída, designado T_u , os ganhos do PID podem ser obtidos por

$$K = 0.6K_u$$

$$T_i = T_u / 2$$

$$T_D = T_u / 8$$

6. Deverá obter uma resposta pouco amortecida. Ajuste manualmente os ganhos do controlador para obter uma resposta com uma sobrelevação inferior a 5% e um tempo de estabelecimento inferior a 200 ms.

Sugestão: aumente a pouco e pouco o tempo derivativo e reduza o ganho proporcional, para reduzir as oscilações da saída. Baixe o termo b para reduzir a sobrelevação.

B – Simulação do controlador PID Discreto

Nesta tarefa iremos implementar um controlador PID discreto para efectuar o controlo digital a 1KHz do motor DC.

1 – É-lhe fornecido um bloco Simulink que implementa o controlador PID Discreto que se obtém utilizando a regra de discretização *ad hoc*:

Parte proporcional:

$$P(kh) = K(bu_c(kh) - y(kh))$$

Parte integral:

$$I(kh) = I(kh - h) + \frac{Kh}{T_i} e(kh - h)$$

Parte derivativa:

$$D(kh) = \frac{T_d}{T_d + Nh} D(kh - h) - \frac{KT_d N}{T_d + Nh} (y(kh) - y(kh - h))$$

2 – Use este PID discreto para controlar a posição do motor. Para isso deverá colocar um bloco retentor de ordem zero com período de amostragem de 1 ms à entrada do motor. Deverá também converter o filtro passa-baixo para o domínio discreto, utilizando o método escalão invariante, para o que pode usar a função do MATLAB *c2d*. Faça ensaios e compare com o PID contínuo.

Tarefas a efectuar no laboratório

Aplique ao sistema real os controladores PID contínuos e discretos projectados anteriormente. Afine o valor dos ganhos até obter as respostas desejadas.

Incluir no Relatório

- Tarefa preparatória - A.
 - Apresente os valores obtidos para os parâmetros de Ziegler e Nichols e ganhos iniciais do controlador.
 - Explique o procedimento efectuado para afinar manualmente os ganhos do controlador.
 - Apresente os valores obtidos para os parâmetros do controlador.
 - Apresente um gráfico com a resposta do sistema, indicando quais a sobrelevação e o tempo de estabelecimento obtidos.

- Tarefa preparatória - B.
 - Apresente o diagrama de blocos do simulador do sistema de controlo, indicando os valores de todos os blocos.
 - Explique o procedimento efectuado para afinar manualmente os ganhos do controlador.
 - Apresente os valores obtidos para os parâmetros do controlador.
 - Apresente um gráfico com a resposta do sistema, indicando quais a sobrelevação e o tempo de estabelecimento obtidos.

- Tarefas do laboratório:
 - Apresente os valores obtidos para os parâmetros do controlador PUID digital.
 - Apresente gráficos com a resposta do sistema, quer com o controlador contínuo, quer com o discreto, indicando quais a sobrelevação e tempo de estabelecimento obtidos.