



**Universidade Federal de Campina Grande**

**Centro de Engenharia Elétrica e Informática**

Curso de Graduação em Engenharia Elétrica

RIENZY ARAÚJO DE AZEVEDO

**ELABORAÇÃO DE MATERIAL DIDÁTICO PARA ESTUDO DE  
SISTEMAS DINÂMICOS NOS AMBIENTES MATLAB E  
SIMULINK**

Campina Grande, Paraíba  
Outubro de 2013

RIENZY ARAÚJO DE AZEVEDO

ELABORAÇÃO DE MATERIAL DIDÁTICO PARA ESTUDO DE  
SISTEMAS DINÂMICOS NOS AMBIENTES MATLAB E  
SIMULINK

*Trabalho de Conclusão de Curso submetido à  
Unidade Acadêmica de Engenharia Elétrica da  
Universidade Federal de Campina Grande  
como parte dos requisitos necessários para a  
obtenção do grau de Bacharel em Ciências no  
Domínio da Engenharia Elétrica.*

Área de Concentração: Sistemas de Controle

Orientador:  
George Acioli Júnior

Campina Grande, Paraíba  
Outubro de 2013

RIENZY ARAÚJO DE AZEVEDO

ELABORAÇÃO DE MATERIAL DIDÁTICO PARA ESTUDO DE  
SISTEMAS DINÂMICOS NO AMBIENTE MATLAB E  
SIMULINK

*Trabalho de Conclusão de Curso submetido à Unidade  
Acadêmica de Engenharia Elétrica da Universidade  
Federal de Campina Grande como parte dos requisitos  
necessários para a obtenção do grau de Bacharel em  
Ciências no Domínio da Engenharia Elétrica.*

Área de Concentração: Sistemas de Controle

Aprovado em        /        /

**Professor Avaliador**  
Universidade Federal de Campina Grande  
Avaliador

**Professor George Acioli Júnior**  
Universidade Federal de Campina Grande  
Orientador, UFCG

## AGRADECIMENTOS

Agradeço aos meus pais por acreditarem desde sempre na minha capacidade e não medirem esforços para apoiar a busca de meus objetivos.

Aos meus irmãos pela fraternidade e exemplos constantes de sucesso.

A George Acioli Júnior pelo conhecimento compartilhado e pelas oportunidades concedidas.

À UFCG e todos os professores e funcionários que fizeram parte da minha graduação.

Aos colegas de curso pelo companheirismo e solidariedade nas horas mais difíceis, e aos amigos que compartilharam minhas alegrias e conquistas como se fossem seus.

Por fim, agradeço a todos que, como esses, de alguma forma colaboraram para o meu crescimento pessoal e profissional, me ajudando a chegar até aqui.

*“Tornamos nosso mundo significativo pela coragem de nossas perguntas e pela profundidade de nossas respostas.”*

Carl Sagan

## LISTA DE ILUSTRAÇÕES

Figura 1: Exemplo de sistema de controle em malha fechada .....	12
Figura 2: Exemplos de respostas de diferentes sistemas à mesma excitação de amplitude limitada .....	12
Figura 3: Sistema em malha fechada.....	13
Figura 4: Gráfico de Lugar das Raízes traçado no MATLAB. ....	14
Figura 5: Efeitos da adição de um zero ao gráfico do lugar das raízes. ....	15
Figura 6: Diagramas de Bode traçados no MATLAB.....	16
Figura 7: Diagrama de Nyquist plotado no MATLAB.. ....	17
Figura 8: (a) Sistema em malha aberta. (b) Sistema em malha fechada com a lei de controle $u = -Kx$ .....	18
Figura 9: Sistema em malha fechada usando controle digital .....	19
Figura 10: <i>Software</i> MATLAB R2013b.....	20
Figura 11: Navegador de bibliotecas do Simulink. ....	21
Figura 12: Código para definição de modelo de sistema massa-mola no MATLAB.....	24
Figura 13: Inserção do modelo do sistema massa-mola sem os polinômios de numerador e denominador. ....	24
Figura 14: Determinação dos pólos de um sistema e plotagem do diagrama de pólos e zeros no MATLAB. ....	25
Figura 15: Diagrama de pólos e zeros plotado no MATLAB. ....	25
Figura 16: Resposta ao degrau de um modelo de motor plotada no LTI Viewer.....	26
Figura 17: Uso do comando <i>feedback</i> para definir um sistema em malha fechada com controlador PID. ....	26
Figura 18: Uso de <i>pidtune</i> para sintonizar automaticamente um controlador.....	27
Figura 19: GUI da ferramenta <i>pidtool</i> .....	27
Figura 20: Código para traçar o lugar das raízes de um modelo cuja saída é a posição de um motor. ....	28
Figura 21: Gráfico de lugar das raízes plotado no MATLAB.....	28
Figura 22: Gráfico do lugar das raízes plotado na ferramenta SISO.....	29
Figura 23: Uso dos comando <i>margin</i> para obter o diagrama de Bode. ....	30
Figura 24: Diagrama de Bode gerado no MATLAB.....	30
Figura 25: Código para determinar a matriz de ganho do controlador para alocar os pólos de malha fechada em uma posição desejada.....	31
Figura 26: Código para definir a matriz de ganho através da técnica LQR. ....	32
Figura 27: Função definida diretamente no plano z. ....	32
Figura 28: Resposta ao degrau do sistema de controle de direção plotada no MATLAB. ....	33
Figura 29: Lugar das raízes do sistema de controle de direção plotado no MATLAB. .	33
Figura 30: Diagrama de pólos e zeros do sistema de controle de direção plotado no MATLAB. ....	34
Figura 31: Resposta ao degrau de um sistema com duas variáveis de saída usando observador. ....	35
Figura 32: Modelo de motor DC com a velocidade como saída construído no Simulink. ....	36

Figura 33: Modelo de motor DC com a velocidade como saída construído com blocos da biblioteca Simscape .....	37
Figura 34: Especificação do bloco que será sintonizado na ferramenta SISO. ....	38
Figura 35: Resposta ao degrau de um sistema de pêndulo invertido plotada através do Simulink. ....	38

# SUMÁRIO

Agradecimentos .....	iv
Lista de Ilustrações .....	vi
Sumário.....	viii
1 Introdução .....	9
1.1 Objetivos .....	9
1.2 Motivação .....	10
1.3 Organização do Texto .....	10
2 Revisão Bibliográfica .....	11
2.1 Sistemas Dinâmicos .....	11
2.2 Técnicas de Projeto de Controladores .....	14
2.3 Controle Digital .....	19
2.4 MATLAB.....	20
2.5 Simulink.....	21
3 Procedimentos Metodológicos.....	23
3.1 Modelagem .....	23
3.2 Análise .....	24
3.3 PID .....	26
3.4 Lugar das Raízes .....	28
3.5 Resposta em Frequência .....	30
3.6 Espaço de Estados.....	31
3.7 Controle Digital .....	32
3.8 Modelagem Usando Simulink .....	35
3.9 Projeto de Controladores Usando Simulink.....	37
4 Resultados.....	39
5 Considerações Finais .....	40
Bibliografia.....	41



# 1 INTRODUÇÃO

Dentre as disciplinas obrigatórias no curso de Graduação em Engenharia Elétrica do Departamento de Engenharia Elétrica (DEE) da Universidade Federal de Campina Grande (UFCG), as disciplinas Controle Analógico e Laboratório de Controle Analógico visam o estudo, análise e controle de sistemas dinâmicos no tempo contínuo.

Conhecendo a grande capacidade de simulação de tais sistemas no *software* MATLAB, a apostila “Sistemas Dinâmicos nos Ambientes MATLAB e Simulink” foi desenvolvida para servir como material de referência para alunos do DEE e estudantes de engenharia de controle em geral.

Todas as funções e ferramentas abordadas na apostila foram testadas e são padrão na versão de estudante do MATLAB R2013.

## 1.1 OBJETIVOS

O objetivo deste projeto é elaborar um material didático que facilite o estudo de controle de sistemas dinâmicos através dos programas MATLAB e Simulink. Tal material deve cobrir a modelagem e a análise de processos dinâmicos, além do uso de métodos de desenvolvimento de controles destes processos como controlador proporcional integral derivativo (PID), lugar das raízes, resposta em frequência e espaço de estados. Também serão introduzidos métodos de conversão de modelos no tempo contínuo para o tempo discreto e como usá-los para desenvolver controladores que atendam aos critérios de projeto dos processos.

Cada capítulo da apostila é dedicado ao estudo no MATLAB de um processo físico diferente, como sistema de controle de direção, velocidade de motor, posição de motor, sistema de suspensão automotiva, controle de *pitch* de aeronave e pêndulo invertido. Com a prática nestes processos, o estudante deve ser capaz de aplicar seus conhecimentos na modelagem, análise e controle de sistemas reais.

Para garantir a confiabilidade do material, todos os códigos foram testados e os resultados foram publicados na apostila, de modo que o aluno possa conferir se a simulação feita no *software* é condizente com a esperada.

## 1.2 MOTIVAÇÃO

A principal motivação deste projeto foi elaborar um material didático atualizado que auxilie o estudo de sistemas de controle usando MATLAB e Simulink na disciplina Laboratório de Controle Analógico.

Sabendo que o MATLAB possui funções, bibliotecas e ferramentas padrões dedicadas à simulação de sistemas dinâmicos, vislumbrou-se na utilização deste *software* a oportunidade de incentivar o aluno a analisar, simular e projetar controladores para modelos de sistemas semelhantes aos que ele encontrará na prática ao longo de sua carreira profissional.

## 1.3 ORGANIZAÇÃO DO TEXTO

No capítulo 2 deste relatório são apresentados os fundamentos teóricos em que esse estudo se baseia. O capítulo 3 é dedicado à descrição dos procedimentos metodológicos para a escrita de cada seção na apostila, desde a modelagem até os métodos de projeto de controle para os sistemas sugeridos. No capítulo 4, os resultados obtidos no projeto de elaboração da apostila são apresentados e, finalmente, no capítulo 5 são feitas as considerações finais do trabalho.

## 2 REVISÃO BIBLIOGRÁFICA

Neste capítulo são apresentadas as bases teóricas que fundamentam este estudo. São tratados os conceitos de estabilidade e representações de modelos de sistemas dinâmicos, além de serem introduzidas noções de projeto de controle pelos métodos de Lugar das Raízes, Resposta em Frequência, Espaço de Estados e uma breve introdução sobre sistemas de controle digitais.

### 2.1 SISTEMAS DINÂMICOS

São considerados sistemas dinâmicos aqueles que mudam ou evoluem com o tempo obedecendo uma determinada função. Para muitos sistemas físicos, esta função pode ser representada por um conjunto de equações diferenciais.

Uma equação diferencial é linear se seus coeficientes forem constantes ou funções apenas da variável independente. Sistemas dinâmicos que podem ser descritos por equações diferenciais lineares e invariantes no tempo são classificados como lineares invariantes no tempo (OGATA, 1997). Para simplificar as análises, consideraremos neste trabalho todos os sistemas como lineares invariantes no tempo (LTI).

Segundo Ogata (1997), os avanços na teoria e na prática do controle automático propiciam meios para se atingir desempenho ótimo de sistemas dinâmicos, melhoria da produtividade e alívio no trabalho enfadonho de muitas operações manuais repetitivas de rotina. Por isso, os engenheiros e cientistas, em sua maioria, devem possuir agora um bom conhecimento deste campo.

Os sistemas de controle de malha fechada, aqueles que, na presença de perturbações, reduzem a diferença entre o sinal de saída de um sistema e o sinal de referência, e que operam com base nesta diferença, tendem a ser mais eficientes do que sistemas de malha aberta e, por isso, receberão maior atenção neste trabalho. A meta global de um sistema de controle de malha fechada é usar o princípio de realimentação para forçar a variável de saída de um processo dinâmico a seguir um valor de referência com precisão, independente do caminho da referência e qualquer perturbação externa

(FRANKLIN; POWELL; EMAMI-NAEINI, 2002). Um exemplo de sistema de controle em malha fechada de temperatura é apresentado na Figura 1.

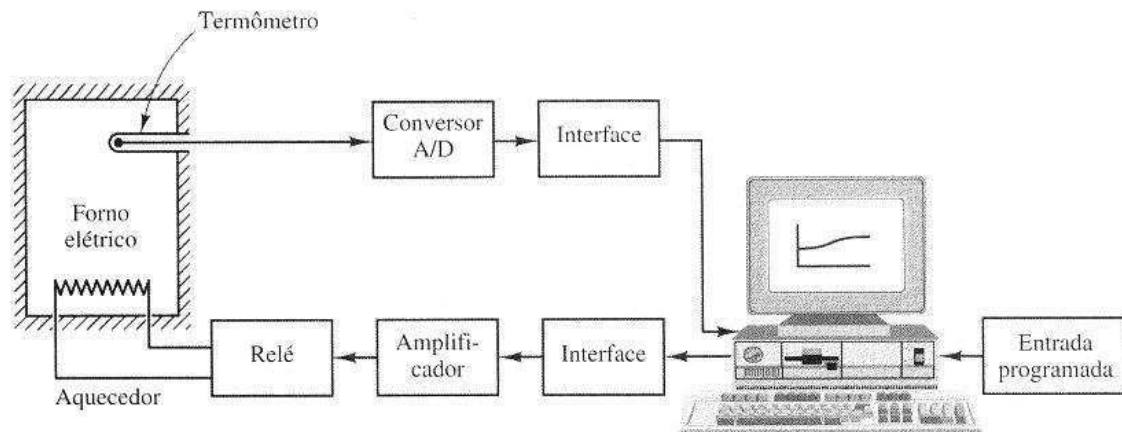


Figura 1: Exemplo de sistema de controle em malha fechada. Fonte: Ogata (1997).

De acordo com Dorf e Bishop (1998), ao se considerar o projeto e a análise de sistemas de controle com retroação, a estabilidade é da maior importância. De um ponto de vista prático, um sistema de controle a malha fechada que seja instável é de pouco valor. Por definição, um sistema é considerado estável se for submetido a uma excitação limitada ou a uma perturbação e a resposta for limitada em magnitude. Exemplos de respostas de diferentes sistemas a uma mesma excitação limitada são apresentadas na Figura 2.

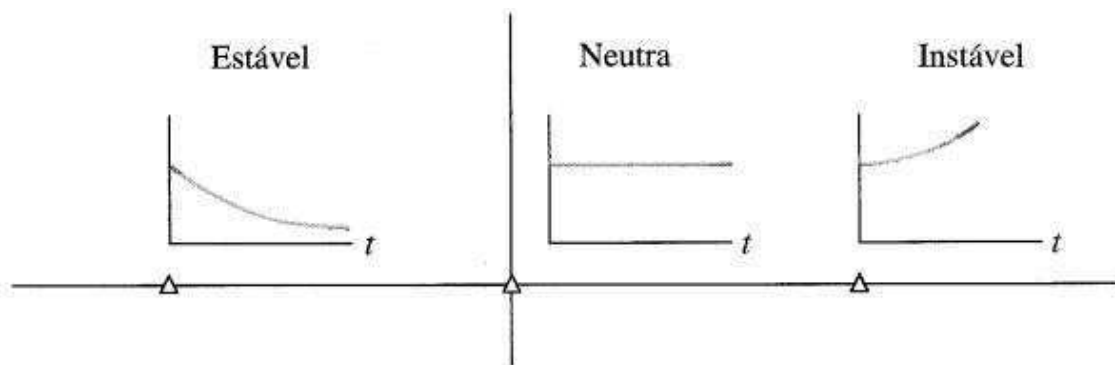


Figura 2: Exemplos de respostas de diferentes sistemas à mesma excitação de amplitude limitada. Fonte: Dorf e Bishop (1998)

O estado de um sistema dinâmico em um dado instante de tempo é caracterizado pelo valor instantâneo de seus atributos de interesse, que podem ser grandezas tais quais potencial elétrico, temperatura e posição. Para que os estados futuros do sistema possam ser determinados através das condições iniciais e entradas futuras, o comportamento dinâmico de cada variável de estado deve ser uma função conhecida dos valores

instantâneos de outras variáveis e entradas do sistema. O modelo em espaço de estados representa estas funções de dependências em termos de equações diferenciais de primeira ordem (GREWAL; ANDREWS, 2001). Para sistemas contínuos LTI, a representação por espaço de estados padrão é dada pelas Equações 1 e 2, onde  $x$  é um vetor de variáveis de estado,  $\dot{x}$  é a derivada no tempo do vetor de estado,  $u$  é a entrada ou vetor de controle,  $y$  é a saída do sistema,  $A$  é a matriz de estado,  $B$  é a matriz de entrada,  $C$  é a matriz de saída e  $D$  é a matriz de transformação direta.

$$\dot{x} = Ax + Bu \quad (1)$$

$$y = Cx + Du \quad (2)$$

E a função de transferência do sistema pode ser obtida através da Equação 3.

$$G(s) = C(sI - A)^{-1}B + D \quad (3)$$

Em casos de sistemas em malha fechada, a relação da saída  $C(s)$  com a entrada de referência  $R(s)$  da Figura 3 é fornecida pela Equação 4. Neste exemplo,  $H(s)$  trata-se da função de transferência de um controlador, que compara o valor real da grandeza de saída do processo com a referência (valor desejado), determina o desvio e produz um sinal de controle que reduzirá o desvio a zero ou a um valor pequeno.

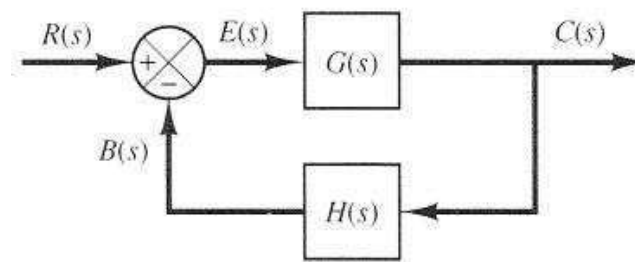


Figura 3: Sistema em malha fechada. Fonte: Ogata (1997).

$$\frac{C(s)}{R(s)} = \frac{G(s)}{1 + G(s)H(s)} \quad (3)$$

Um exemplo de função de transferência de controlador automático é dado na Equação 4, que descreve o comportamento de um controlador Proporcional Integral Derivativo (PID).

$$H(s) = K_p + K_d s + \frac{K_i}{s} \quad (4)$$

## 2.2 TÉCNICAS DE PROJETO DE CONTROLADORES

A estabilidade relativa e o desempenho de um sistema de controle a malha fechada estão diretamente relacionados com a localização das raízes da equação característica no plano  $s$ . Por isso, é muito útil determinar como as raízes se deslocam ao longo deste plano à medida que se varia o valor de um parâmetro (DORF; BISHOP, 1998). O gráfico do lugar das raízes, exemplificado na Figura 4, é uma ferramenta poderosa para se analisar e projetar sistemas de controle com retroação e pode ser facilmente traçado no MATLAB.

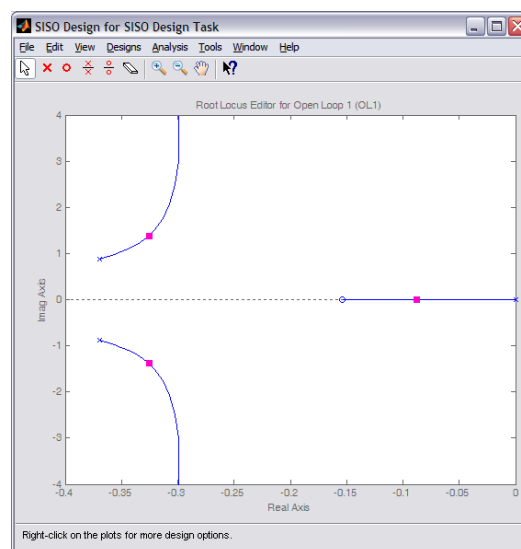


Figura 4: Gráfico de Lugar das Raízes traçado no MATLAB. Fonte: Messner e Tilbury (2012).

Segundo Ogata (1997), na prática, o lugar das raízes de um sistema pode indicar que não se pode alcançar o desempenho desejado unicamente pelo ajuste no valor do ganho. Com efeito, em alguns casos, o sistema pode não ser estável para todos os valores de ganho. Torna-se necessário, então, dar uma nova forma aos lugares das raízes para se atender às especificações de desempenho.

No uso do gráfico do lugar das raízes para projetar um sistema de controle, caso se faça necessário um compensador mais robusto que o simples ajuste do ganho, deve-se compreender plenamente os efeitos que a adição de pólos e zeros trarão ao gráfico.

Com isso, é possível determinar as localizações dos pólos e zeros do compensador que irão gerar a forma do lugar das raízes para que o sistema atenda aos critérios de desempenho.

Em essência, neste método, os lugares das raízes do sistema são submetidos a uma mudança de forma através do uso de um compensador de modo que um par de pólos a malha fechada dominantes possa ser alocado na posição desejada (OGATA, 1997). Na Figura 4 são exemplificados os efeitos da adição de um zero ao gráfico do lugar das raízes de um sistema com três pólos.

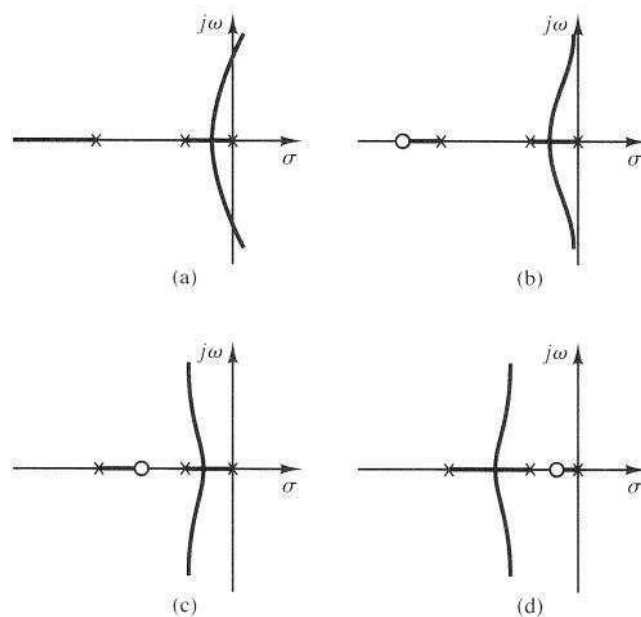


Figura 5: Gráfico do lugar das raízes de (a) Sistema com três pólos (b), (c) e (d) Sistema com três pólos com a adição de um zero à sua função de transferência. Fonte: Ogata (1997).

De acordo com Franklin, Powell e Emami-Neini (2002), o método de projeto de sistemas de controle na indústria mais popular é o de resposta em frequência, pois ele fornece bons controladores mesmo na presença de incertezas no modelo do sistema. Outra vantagem no uso deste método é a facilidade com que cada dado experimental pode ser usado. Simples medições de amplitude e fase da planta recebendo excitação senoidal são suficientes para desenvolver um controlador apropriado.

A resposta de frequência de um sistema é definida como a resposta de estado estacionário do sistema a um sinal de entrada. A senóide é um sinal de entrada peculiar, e o sinal de saída resultante em um sistema linear, bem como os sinais ao longo deste, é senoidal em regime permanente e difere da forma de onda do sinal de entrada somente no que diz respeito à amplitude e ângulo de fase (DORF; BISHOP, 1998). Os diagramas de Bode constituem-se, portanto, em uma das ferramentas gráficas mais potentes para

analisar e projetar sistemas de controle, e podem ser facilmente plotados usando MATLAB, como apresentado na Figura 6.

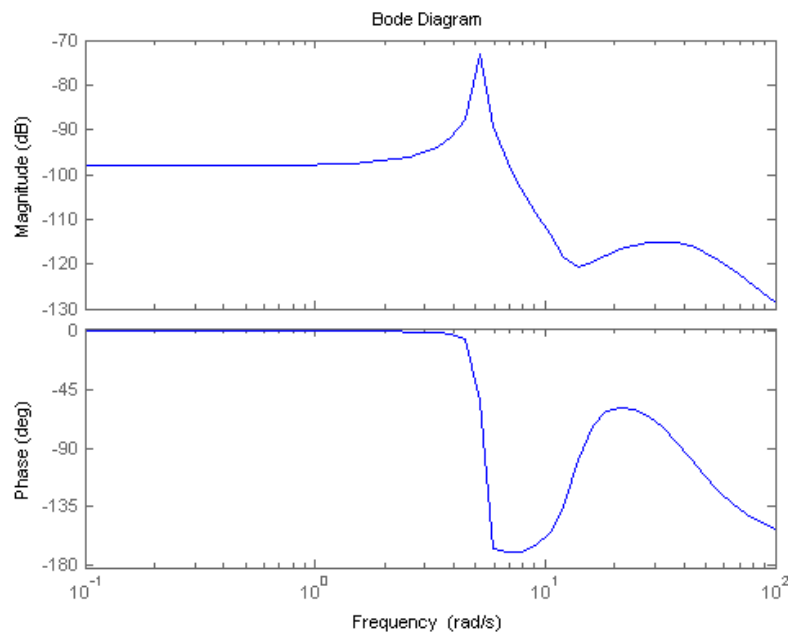


Figura 6: Diagramas de Bode traçados no MATLAB.

Quando se utiliza o método de resposta em frequência é importante ficar atento à estabilidade do sistema. Para isso, se utilizam os conceitos de margem de ganho e margem de fase. Franklin, Powell e Emami-Neini (2002) definem margem de ganho como o fator que o ganho pode ser elevado antes de tornar o sistema instável. Já a margem de fase é o valor que a fase excede  $-180^\circ$  no ponto  $|K G(j\omega) = 1|$ .

Os diagramas de Nyquist, do mesmo modo que os diagramas de Bode, são comumente usados para a representação de sistemas de controle com retroações lineares e invariantes no tempo, no domínio da frequência. Os diagramas de Nyquist são polares, enquanto os diagramas de Bode são retangulares. Um dos diagramas pode ser mais conveniente para uma operação em particular, porém qualquer operação pode ser conduzida por meio de ambos os diagramas (OGATA, 1997).

O diagrama de Nyquist, apresentado na Figura 7, pode fornecer informações a respeito da estabilidade relativa do sistema em malha fechada. Segundo Dorf e Bishop (1998), o critério de Nyquist diz respeito ao mapeamento da Equação 5 e o número de circunSCRIÇÕES da origem no plano  $F(s)$ .

$$F(s) = 1 + L(s) \quad (5)$$



O critério de Nyquist define que um sistema de controle com retroação é estável se, e somente se, o número de circunscrições no sentido anti-horário no ponto  $(-1,0)$  do diagrama de Nyquist for igual ao número de pólos de  $L(s)$  com parte real positiva.

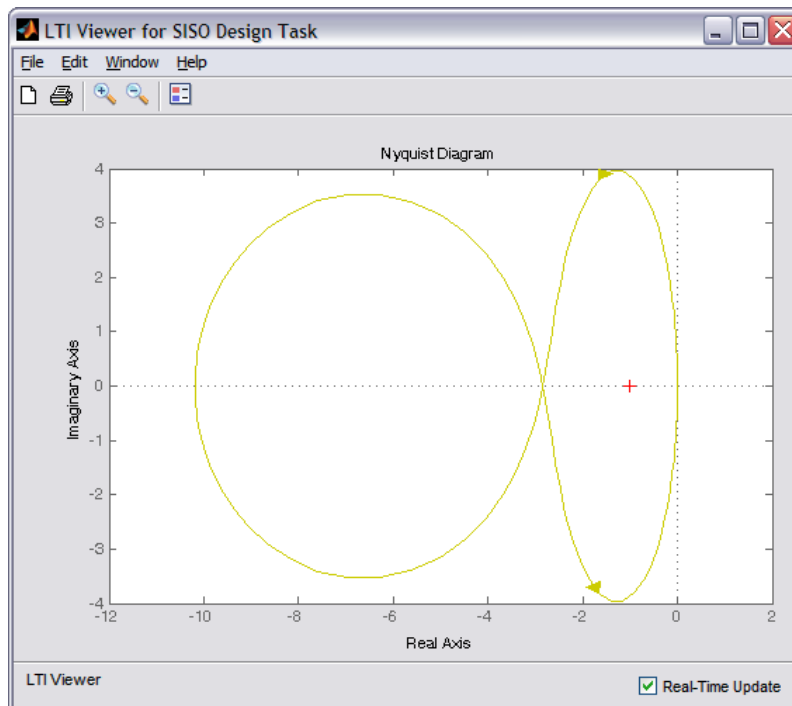


Figura 7: Diagrama de Nyquist plotado no MATLAB. Fonte: Messner e Tilbury (2012).

O método no domínio do tempo, expresso em termos de variáveis de estado, também pode ser usado para projetar estruturas de compensação adequadas. O interesse recai, fundamentalmente, em controlar o sistema com um sinal de controle,  $u(t)$ , que seja uma função das diversas variáveis de estado medidas. Desenvolve-se então um controlador com variáveis de estado que opera sobre as informações disponíveis sob a forma de medida (DORF; BISHOP, 1998).

Segundo Franklin, Powell e Emami-Naeini (2002), as principais vantagens no uso desta técnica são possibilitar o estudo de modelos mais gerais (as equações diferenciais não precisam ser lineares e o método se estende facilmente para sistemas com múltiplas entradas/múltiplas saídas), introduzir as ideias de geometria em equações diferenciais e conectar descrições internas e externas (o estado de um sistema dinâmico frequentemente descreve a distribuição de energia internamente no sistema). O uso do método de projeto de controle por variáveis de espaço de estados é particularmente vantajoso quando o processo possui mais de uma entrada de controle ou mais de uma saída medida.

De acordo com Dorf e Bishop (1998), a retroação de estado usualmente não é prática por duas razões. A primeira é que ela conduz a compensadores do tipo PD ou PID, com banda passante infinita, quando se sabe que componentes reais e compensadores sempre apresentam banda passante finita. A segunda razão é que, simplesmente, não é possível ou prático medir todos os estados e fazer sua retroação.

Diferentemente de se especificar somente os pólos a malha fechada dominantes (abordagem de projeto convencional), o presente enfoque de alocação de pólos especifica todos os pólos a malha fechada. Há, contudo, um custo associado com o posicionamento de todos os pólos a malha fechada, porque a alocação de todos requer a medida, com sucesso, de todas as variáveis; ou, além disto, requer a inclusão de um observador no sistema. Há também um requisito sobre que parte do sistema deve ser escolhida para a alocação arbitrária de pólos. O requisito é que o sistema seja a estados completamente controláveis (OGATA, 1997).

$$u = -Kx \quad (5)$$

Considerando que a lei de controle da Equação 5 é escolhida para compensar o sistema da Figura 8.a, a arquitetura do sistema em malha fechada se tornará semelhante à apresentada na Figura 8.b.

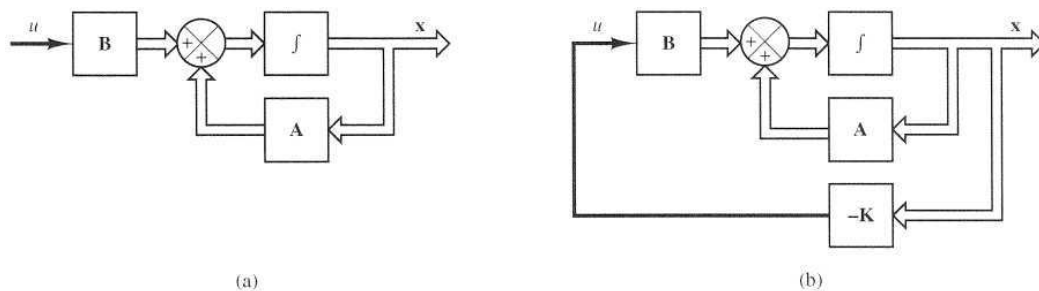


Figura 8: (a) Sistema em malha aberta. (b) Sistema em malha fechada com a lei de controle  $u = -Kx$ .  
Fonte: Ogata (1997).

Sabendo que uma condição necessária para a alocação arbitrária de pólos é que o sistema seja a estados completamente controláveis, deve-se entender plenamente as noções de controlabilidade e observabilidade. De acordo com Dorf e Bishop (1998), um sistema pode ser dito controlável se existir um controle  $u$  sem restrições que possa transferir qualquer estado inicial  $x(0)$  para qualquer outra localização desejada  $x(t)$ . Em outras palavras, caso o posto da matriz de controlabilidade, apresentada na Equação 6, for igual à ordem do sistema, ele é considerado controlável.

$$P_c = [B \ AB \ A^2B \ \dots \ A^{n-1}B \ B] \quad (6)$$

Já a observabilidade se refere à capacidade de se estimar uma variável de estado. Um sistema é observável se, e somente se, existir um tempo  $T$  finito tal que o estado inicial  $x(0)$  possa ser determinado a partir do histórico de  $y(t)$ , dado o controle  $u(t)$  (DORF; BISHOP, 1998). Em outras palavras, o sistema pode ser classificado como observável caso o determinante da matriz de observabilidade, expressa na Equação 7, for diferente de zero.

$$Q = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} \quad (7)$$

## 2.3 CONTROLE DIGITAL

Segundo Aström e Wittenmark (1996), praticamente todos os sistemas de controle desenvolvidos hoje são baseados em controle com computadores. É, portanto, importante compreender sistemas de controle digitais, que evitam muitas das dificuldades em implementações analógicas. Por exemplo, não há problemas com acurácia de componentes, é muito simples ter cálculos sofisticados na lei de controle, e é fácil incluir funções lógicas e não lineares. Além disso, bancos de dados podem ser mantidos de maneira a acumular conhecimento a respeito das propriedades do sistema.

O exemplo de um sistema em malha fechada com controle digital é apresentado na Figura 9.

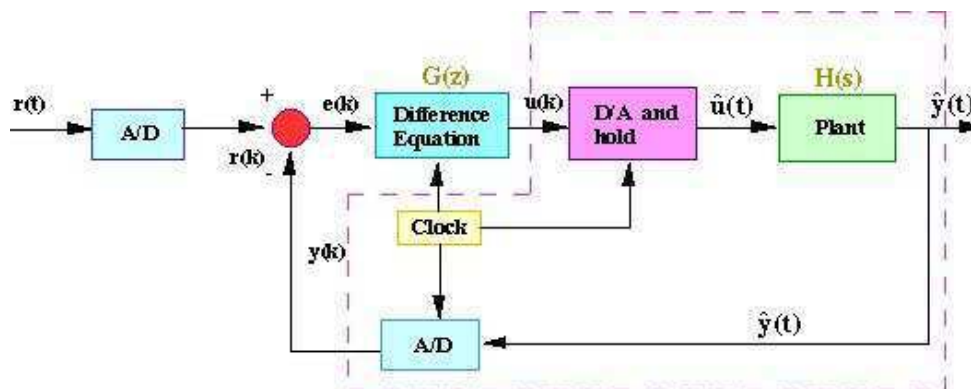


Figura 9: Sistema em malha fechada usando controle digital. Fonte: National Instruments (2013).

Para que o sistema de controle digital seja implementado, é necessário fazer a conversão do sinal de saída analógico para um sinal discreto. Isto é feito através de um conversor A/D. Além disso, o sinal de controle,  $u(k)$ , é convertido para o analógico usando um conversor D/A e um segurador de ordem zero.

## 2.4 MATLAB

O MATLAB é um ambiente interativo de linguagem de alto nível para computação numérica, visualização e programação desenvolvido pela *MathWorks, Inc.* De acordo com o site do fornecedor, é possível usar este software para analisar dados, desenvolver algoritmos, criar modelos e aplicações. A linguagem, ferramentas e funções matemáticas embarcadas permitem explorar múltiplas abordagens e chegar a soluções mais rapidamente do que usando *spreadsheets* ou linguagens de programação tradicionais como C/C++ ou Java.

Nos últimos anos, o MATLAB tem se estabelecido como referência em computação numérica, processamento de sinais e desenvolvimento de algoritmos. Por muito tempo, a versão profissional do *software* era a única disponível. A chegada da versão mais barata para estudantes possibilitou seu uso em sala de aula.

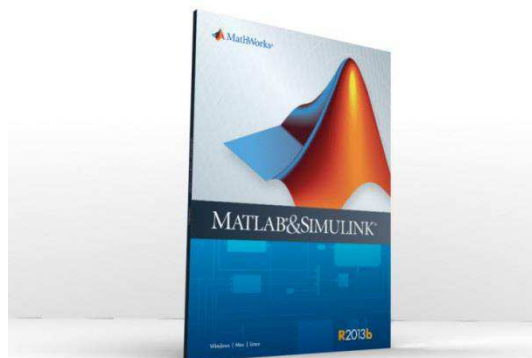


Figura 10: *Software* MATLAB R2013b. Fonte: MathWorks, Inc.

Pode-se pensar no MATLAB (uma abreviatura de *Matrix Laboratory* – Laboratório de Matrizes) como uma espécie de linguagem projetada exclusivamente para realizar manipulações com matrizes. Todas as variáveis manipuladas no MATLAB são matrizes, isto é, neste ambiente só existe um tipo de dado, uma matriz ou um arranjo retangular de números. Além de possuir um conjunto extenso de rotinas para a obtenção de saídas gráficas, o MATLAB possui muitas funções predefinidas que podem ser

chamadas pelo usuário para resolver muitos e diferentes tipos de problemas de controle (OGATA, 1997).

A família de programas MATLAB inclui o programa principal (base) e mais uma variedade de *toolboxes*, uma coleção de arquivos especiais chamados arquivo-m (*m-files*) que estendem a funcionalidade do programa principal. Juntos, o programa principal e o *Control System Toolbox* propiciam a capacidade para projetar e analisar sistemas de controle. A maioria das sentenças, funções e comandos são independentes da plataforma computacional. Qualquer que seja o sistema operacional particular que estiver sendo utilizado, a interação com o MATLAB será basicamente a mesma (DORF e BISHOP, 1998). Neste trabalho será tratada a interação do usuário com o MATLAB independente da plataforma.

## 2.5 SIMULINK

Segundo seu guia do usuário, o Simulink é um ambiente de diagramas de blocos para simulação de projetos baseados em modelos. Ele suporta projeto a nível do sistema, simulação, geração automática de código e verificação contínua de sistemas embarcados. O Simulink fornece um editor gráfico, bibliotecas de blocos configuráveis, e solucionador para modelar e simular sistemas dinâmicos. O Simulink é integrado com MATLAB, sendo capaz de incorporar algoritmos do programa principal em seus modelos e exportar resultados de simulações para o MATLAB, possibilitando análises posteriores.

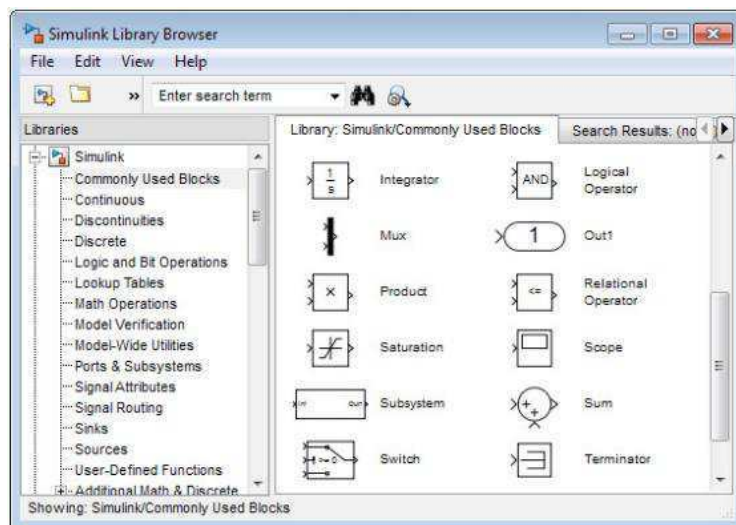


Figura 11: Navegador de bibliotecas do Simulink. Fonte: MathWorks, Inc. (2012).

Engenheiros de automação têm descoberto cada vez mais que a simulação no desenvolvimento de sistemas de controle avançados é uma ferramenta vital para otimizar tempo e custo do projeto. Como uma ferramenta de desenvolvimento, o Simulink tem se tornado padrão de excelência pela flexibilidade e acurácia em sua capacidade de simulação e modelagem. Como resultado de sua arquitetura aberta, o Simulink permite que engenheiros criem bibliotecas de blocos customizadas para que possam apoiar trabalhos científicos futuros.

## 3 PROCEDIMENTOS METODOLÓGICOS

Neste capítulo serão descritos os procedimentos adotados para a escrita de cada seção da apostila “Sistemas Dinâmicos nos Ambientes MATLAB e Simulink”, objeto de estudo deste trabalho.

Foram seis sistemas físicos tratados na apostila, tendo um capítulo dedicado a cada um deles. Em cada capítulo foram escritas uma seção específica para o estudo da modelagem, análise, projeto de controladores PID, e os métodos de desenvolvimento Lugar das Raízes, Resposta em Frequência, Espaço de Estados e Controle Digital dos sistemas. Foram elaboradas ainda duas seções dedicadas ao uso do Simulink para modelagem e controle.

Todos os modelos, bem como os códigos de MATLAB deste trabalho foram retirados do site do departamento de engenharia da Universidade de Michigan. Todos os códigos foram testados para garantir a confiabilidade dos resultados esperados pelo aluno no estudo da apostila.

### 3.1 MODELAGEM

O primeiro passo no projeto de controladores é o desenvolvimento de modelos matemáticos apropriados derivados de leis físicas ou de dados experimentais. Nesta seção de cada capítulo são apresentadas algumas abordagens básicas para modelar os processos e como inseri-los no MATLAB para análises futuras.

Inicialmente, os parâmetros físicos do sistema são definidos na apostila para que seja possível determinar suas representações em função de transferência e por espaço de estados. Logo, definem-se os critérios de projeto, em geral, *overshoot*, tempo de estabelecimento, tempo de subida e erro de regime permanente.

Com os parâmetros físicos definidos, o modelo do processo em função de transferência pode ser inserido no MATLAB através dos comando *tf*. Já a representação em espaço de estados é inserida com o uso da função *ss*. Um exemplo de um sistema massa-mola inserido no MATLAB é exibido na Figura 12.

```

1      m = 1; %Definição dos parâmetros físicos
2      k = 1;
3      b = 0.2;
4
5      A = [0 1; -k/m -b/m];
6      B = [0 1/m];
7      C = [1 0];
8      D = [0];
9      sys_ss = ss(A,B,C,D) %Modelo em espaço de estados
10
11     num = [1];
12     den = [m b k];
13     sys_tf = tf(num,den) %Modelo em função de transferência

```

Figura 12: Código para definição de modelo de sistema massa-mola no MATLAB.

Uma maneira alternativa de modelagem é imputar a função de transferência diretamente, sem utilizar os polinômios de numerador e denominador, como exemplificado na Figura 13.

```

1      m = 1; %Definição dos parâmetros físicos
2      k = 1;
3      b = 0.2;
4
5      s = tf('s');
6      sys_tf = 1/(m*s^2+b*s+k) %Modelo em função de transferência

```

Figura 13: Inserção do modelo do sistema massa-mola sem os polinômios de numerador e denominador.

## 3.2 ANÁLISE

Uma vez que modelos matemáticos do sistema foram obtidos, seja na forma de espaço de estados ou função de transferência, é possível então analisá-los e prever como serão as respostas do sistema, tanto no domínio do tempo quanto no da frequência. Nesta seção dos capítulos, são apresentadas maneiras para determinar propriedades dinâmicas como estabilidade, velocidade e erro de regime permanente.

A representação em função de transferência é particularmente prática quando se analisa estabilidade. Se todos os pólos da função possuem a parte real negativa, o sistema é dito estável. Os pólos de um sistema podem ser facilmente encontrados no ambiente MATLAB usando o comando `pole`, como exemplificado na Figura 14.



```

5     s = tf('s');
6     sys_tf = 1/(m*s^2+b*s+k) %Modelo em função de transferência
7     pole(sys_tf) %Determinação dos pólos do sistema
8     pzmap(sys_tf) %Plotagem do diagrama de pólos e zeros

```

Figura 14: Determinação dos pólos de um sistema e plotagem do diagrama de pólos e zeros no MATLAB.

A execução do código-texto exibido na Figura 14 gera o diagrama de pólos e zeros exibido na Figura 15, que facilita a visualização da localização dos pólos do sistema.

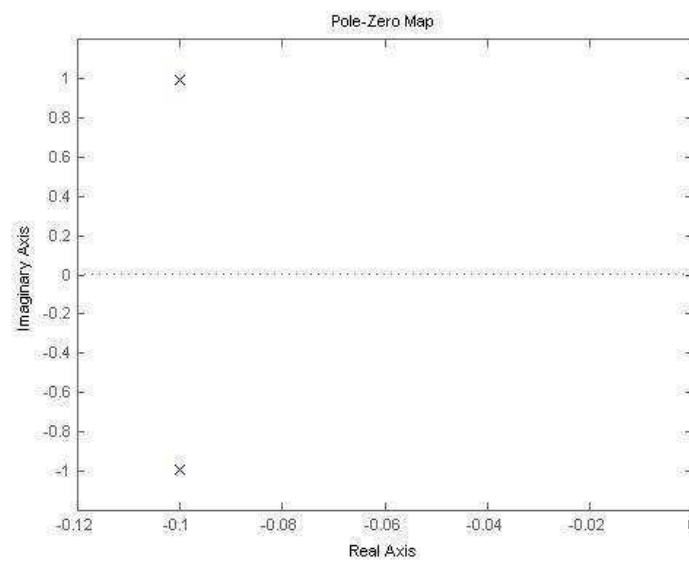


Figura 15: Diagrama de pólos e zeros plotado no MATLAB.

Gráficos importantes para análise do sistema como diagramas de Bode, resposta ao degrau e resposta ao impulso podem ser facilmente plotados no MATLAB através dos comandos *bode*, *pole* e *impulse*, respectivamente. No entanto, através da ferramenta LTI viewer, é possível traçar a resposta do sistema para qualquer tipo de sinal. No exemplo da Figura 16, a resposta ao degrau de um modelo para a posição de um motor é plotada no LTI viewer.

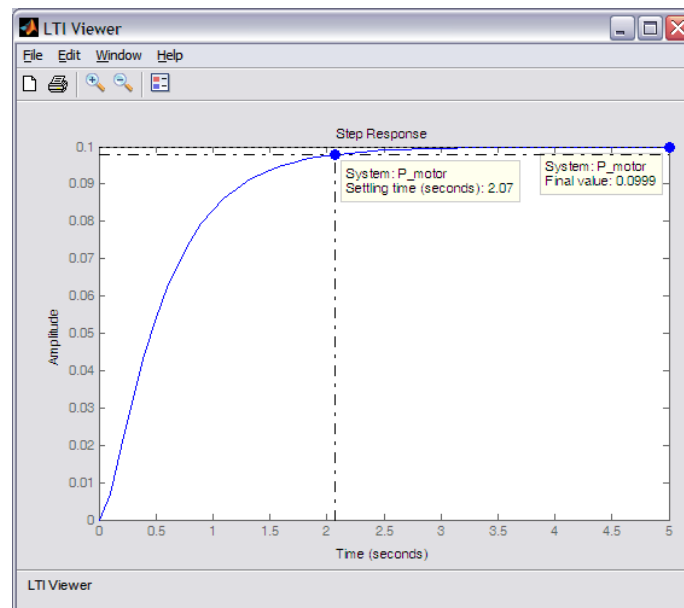


Figura 16: Resposta ao degrau de um modelo de motor plotada no LTI Viewer.

Outros comandos importantes apresentados nesta seção para análise do sistema são *isstable* (para avaliar a estabilidade do sistema), *damp* (determina a frequência natural e razão de amortecimento) e *zpk* (converte o modelo para a forma pólos-zeros-ganho).

### 3.3 PID

Esta seção dos capítulos foi utilizada para observar a resposta da planta em malha fechada com controladores PID de diferentes ganhos. Na Figura 17, isto é ilustrado para um modelo de velocidade do motor, onde o comando *feedback* é usado para fechar a malha do sistema.

```

7      s = tf('s')
8      motor = K / ((J*s+b) * (L*s+R) + K^2); %Definição do modelo
9
10     Kp = 75; %Definição do controlador
11     Ki = 1;
12     Kd = 1;
13     C = Kp + Kd*s + Ki/s;
14
15     sys_mf = feedback(C*motor,1); %Sistema em malha fechada
16     step(sys_mf,[0:1:200]); %Gráfico da resposta ao degrau

```

Figura 17: Uso do comando *feedback* para definir um sistema em malha fechada com controlador PID.

Uma maneira de aperfeiçoar o controlador PID demonstrada na apostila é através do comando *pidtune* na linha de comando. Nos argumentos deste comando, é possível determinar, por exemplo, a frequência de cruzamento de ganho e a margem de fase desejadas no sistema, e o controlador é gerado automaticamente por meio destes parâmetros pela função *pidtune*. A otimização do controlador PID de um sistema em malha fechada é demonstrado na Figura 18.

```

1  s = tf('s');
2  P = 1/(s^2 + 10*s + 20); %Definição do sistema
3
4  opcs = pidtuneOptions('CrossoverFrequency',32,'PhaseMargin',90);
5  [C,info] = pidtune(P,'pid',opcs) %Chamada da função pidtune

```

Figura 18: Uso de *pidtune* para sintonizar automaticamente um controlador.

Outra maneira de otimizar o controle utilizada nesta seção é a ferramenta PID Tuner, fornecida pelo MATLAB r2013, que sintoniza automaticamente o controlador para que o sistema atenda às exigências de projeto. Ela é chamada pelo comando *pidtool*, e, através da interface gráfico do usuário (GUI) apresentada na Figura 19, o controlador pode ser ajustado de acordo com parâmetros tais quais tempo de estabelecimento e tempo de subida.

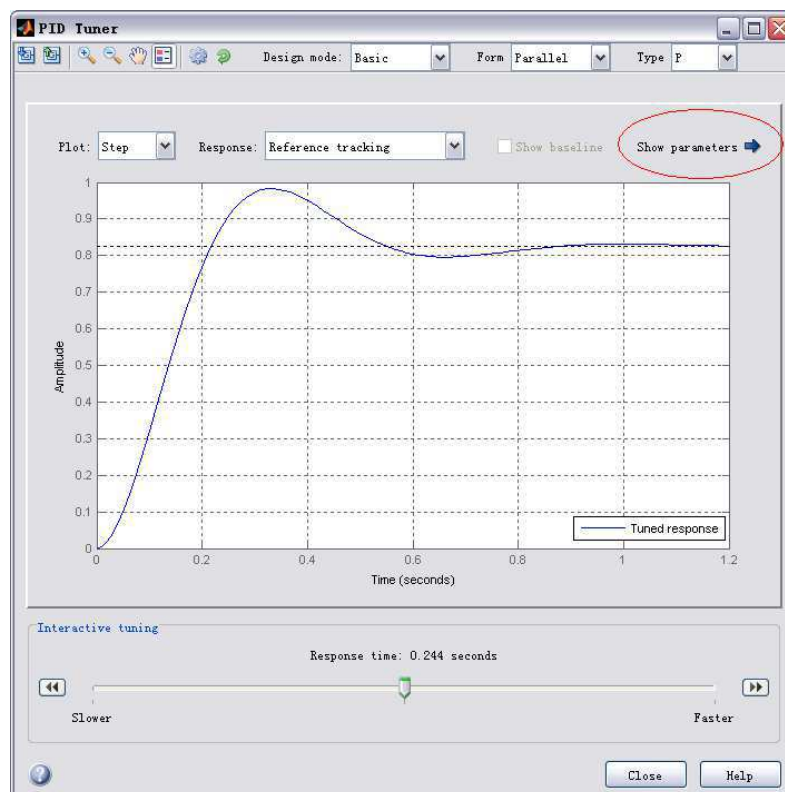


Figura 19: GUI da ferramenta *pidtool*. Fonte: Messner e Tilbury (2012).

### 3.4 LUGAR DAS RAÍZES

Esta seção de cada capítulo da apostila foi aproveitada para descrever como plotar gráficos do lugar das raízes no MATLAB e usá-los para projetar controladores que satisfazem determinados critérios de desempenho.

O gráfico do lugar das raízes pode ser facilmente plotado no MATLAB através do comando `rlocus`. Na Figura 20, é apresentado o código que deve gerar o lugar das raízes de um sistema para modelar a posição de um motor.

```

1      J = 3.2284E-6; %Definição do modelo
2      b = 3.5077E-6;
3      K = 0.0274;
4      R = 4;
5      L = 2.75E-6;
6      s = tf('s');
7      P_motor = K/(s*((J*s+b)*(L*s+R)+K^2));
8
9      rlocus(P_motor) %Plota o gráfico do lugar das raízes
10     sgrid(.5, 0) %Plota as retas referentes aos critérios de projeto

```

Figura 20: Código para traçar o lugar das raízes de um modelo cuja saída é a posição de um motor.

O gráfico do lugar das raízes gerado a partir da execução do código da Figura 20 é exibido na Figura 21.

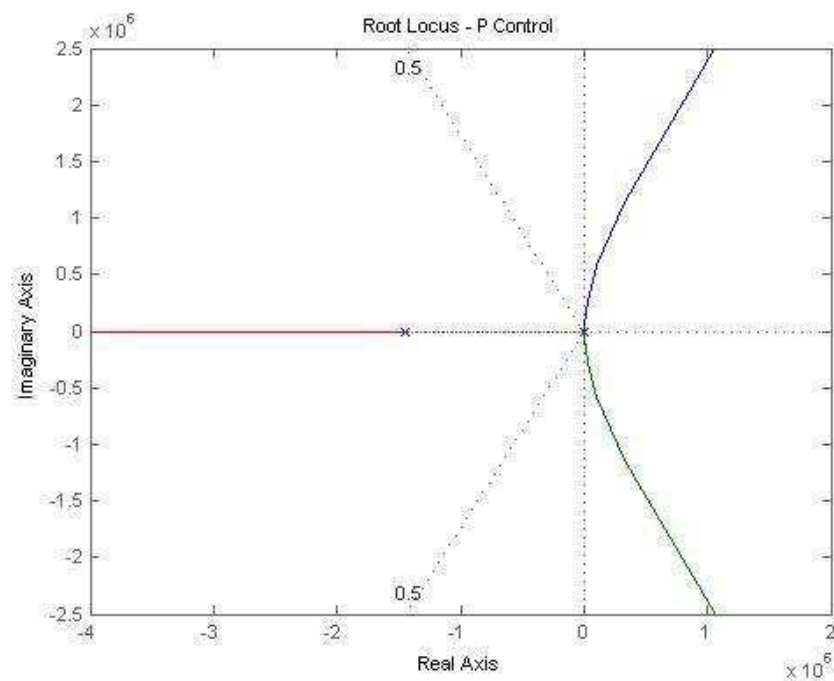


Figura 21: Gráfico de lugar das raízes plotado no MATLAB.

Além de *rlocus*, o comando *sgrid* pode ser usado para que seja possível visualizar as regiões do gráfico que atendem aos critérios de projeto definidos. A partir destas regiões, como descrito na apostila, o estudante pode usar a função *rlocfind* para encontrar o ganho de malha que irá posicionar os pólos do sistema na localização desejada.

Um artifício importante na elaboração desta seção foi o uso da ferramenta de projeto SISO para desenvolver controladores a partir do gráfico do lugar das raízes. Nela, é possível modificar a arquitetura do sistema, plotar rapidamente diferentes gráficos para análise e verificar interativamente a resposta do sistema, além de ser capaz de sintonizar automaticamente o controlador. No exemplo da Figura 22, o gráfico do lugar das raízes de um modelo de velocidade de motor já exibe as regiões que atendem aos critérios de projeto.

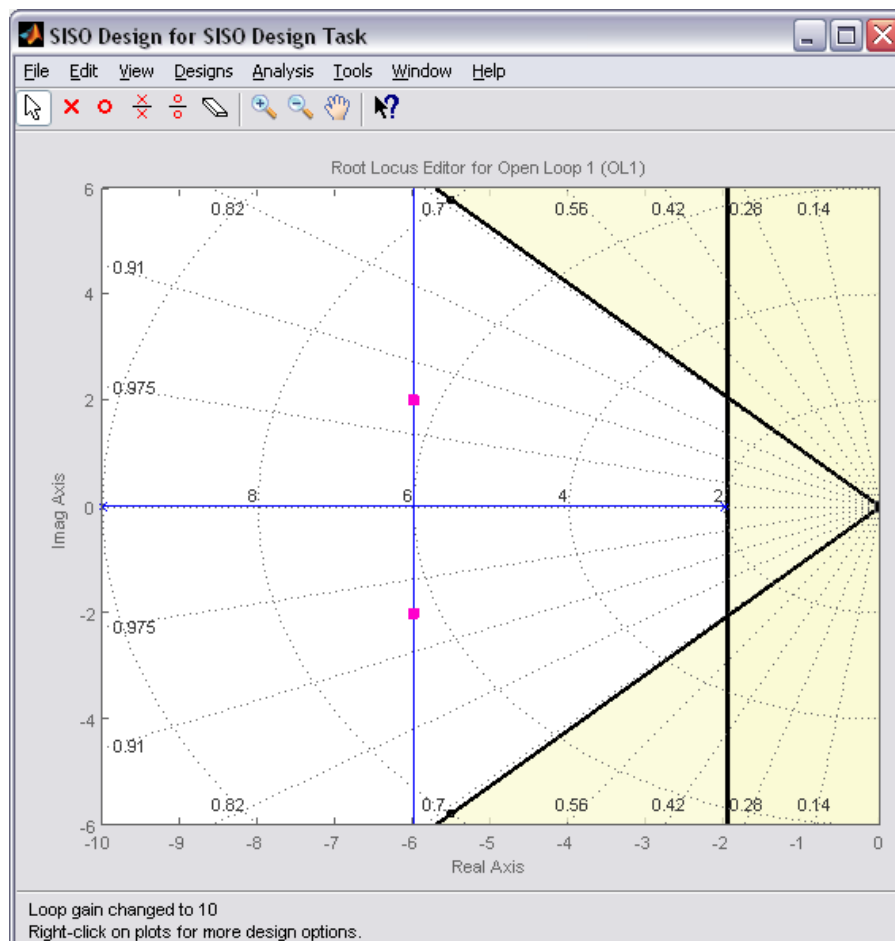


Figura 22: Gráfico do lugar das raízes plotado na ferramenta SISO.

Outro comando útil apresentado nesta seção da apostila foi o *minreal*, que pode ser usado para reduzir os modelos dos sistemas através do cancelamento de pólos e zeros.

### 3.5 RESPOSTA EM FREQUÊNCIA

Esta seção de cada capítulo da apostila foi dedicada a demonstrar como usar a resposta em frequência dos sistemas em malha aberta para prever seus comportamentos em malha fechada, e como utilizar esta informação para projetar os controladores.

Além de plotar os diagramas de Bode, a função *bode* do MATLAB pode ser usada para determinar os valores exatos de magnitude e fase em uma frequência exata. No exemplo da Figura 23, desejava-se conhecê-las na frequência 10 rad/s. O comando *margin* foi usado para demarcar as margens de ganho e fase no diagrama.

```

7 - P_motor = K/(s*((J*s+b)*(L*s+R)+K^2));
8
9 - C = 72 %Definição do controlador proporcional
10
11 - [mag,phase,w] = bode(P_motor,10) %Magnitude e fase na frequência 10 rad/s
12 - margin(C*P_motor)

```

Figura 23: Uso dos comando *margin* para obter o diagrama de Bode.

Os diagramas de Bode gerados da execução do código da Figura 23 são apresentados na Figura 24.

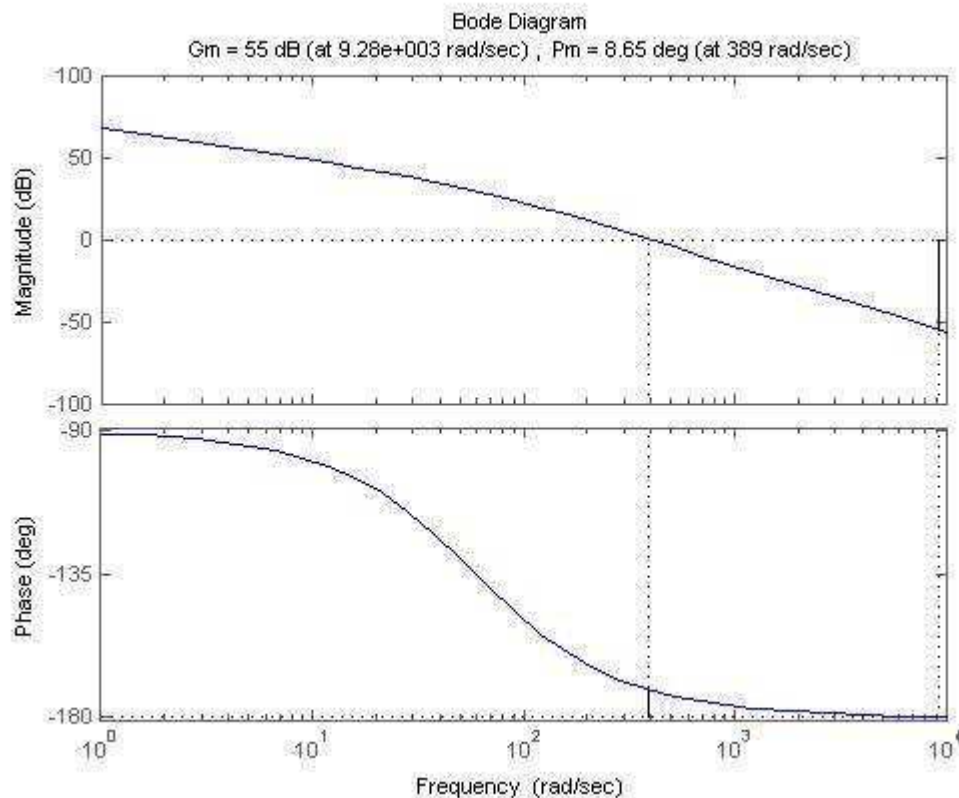


Figura 24: Diagrama de Bode gerado no MATLAB.

Uma maneira de analisar a estabilidade do sistema através da resposta em frequência apresentada na apostila é usando o diagrama de Nyquist, que pode ser facilmente plotado através do comando do MATLAB *nyquist*. Uma outra função que pode ser interessante no uso da resposta em frequência para projetos de controladores é *lsim*, que é capaz de traçar a resposta do sistema a qualquer tipo de sinal.

### 3.6 ESPAÇO DE ESTADOS

Nesta seção de cada capítulo da apostila foi abordado o método de projeto de controladores através do espaço de estados. Para isso, é necessário que se defina a localização desejada para os pólos do sistema e, então, a função *place* do MATLAB é utilizada para definir a matriz de ganho. No exemplo da Figura 25, este procedimento é descrito para alocar em -1,5 o polo de um sistema de controle de direção.

```

11 - sys = ss(A,B,C,D); %Definição do modelo em espaço de estados
12 - x0 = [0]; %Condição inicial nula
13 - p1 = -1.5; %Polo do sistema em -1,5
14
15 - K = place(A,B,[p1]) %Determinação da matriz de ganho
16 - sys_cl = ss(A-B*K,B,C,D); %Sistema em malha fechada com os novos ganhos
17 - lsim(sys_cl,u,t,x0); %Plotagem da resposta ao degrau do sistema
18 - axis([0 10 0 0.35])

```

Figura 25: Código para determinar a matriz de ganho do controlador para alocar os pólos de malha fechada em uma posição desejada.

Como descrito no material produzido, o MATLAB é capaz de criar as matrizes de controlabilidade e observabilidade através dos comandos *ctrb* e *obsv*, respectivamente. A partir delas, é possível classificar se os sistemas são controláveis e observáveis usando as funções *rank*, para determinar o posto das matrizes, e *det* para calcular seus determinantes.

Em alguns capítulos da apostila, fez-se uso da técnica Regulador Linear Quadrático (LQR) para gerar uma matriz de ganho ótima sem escolher explicitamente os pólos de malha fechada do sistema. Para utilizar o método LQR, é necessário definir as matrizes ponderadas de custo (Q) e de controle (R), como exemplificado na Figura 26.

```

1 -   A = [-0.313 56.7 0; -0.0139 -0.426 0; 0 56.7 0];
2 -   B = [0.232; 0.0203; 0];
3 -   C = [0 0 1];
4 -   D = [0]; %Definição do modelo em espaço de estados
5
6 -   p = 2;
7 -   Q = p*C'*C; %Definição de matriz de custo
8 -   R = 1; %Definição de matriz de controle
9
10 -  [K] = lqr(A,B,Q,R); %Determinação da matriz de ganho
11 -  sys_cl = ss(A-B*K,B,C,D); %Sistema em malha fechada

```

Figura 26: Código para definir a matriz de ganho através da técnica LQR.

### 3.7 CONTROLE DIGITAL

Nesta seção de cada capítulo são descritos procedimentos para conversão de modelos no tempo contínuo para o tempo discreto, além de ser demonstrado o uso de tais modelos para projetar controladores digitais.

No MATLAB, a conversão de sistemas do tempo contínuo para o discreto é feito através da função *c2d*, que utiliza um tempo de amostragem definido pelo usuário para digitalizar o modelo. Uma função de transferência também pode ser definida diretamente no plano *z* como exemplificado na Figura 27.

```

1 -   Ts = 1/50; %Definição do tempo de amostragem
2 -   z = tf('z',Ts);
3 -   G = 0.2*(z-0.999)/(z-0.9998); %Definição da função de transferência

```

Figura 27: Função definida diretamente no plano *z*.

A partir do modelo no tempo discreto, são descritos na apostila procedimentos para projetar controladores através das técnicas tratadas até aqui utilizando comandos como *step*, *feedback*, *pzmap*, *rlocus*, *zgrid* e *rlocfind*. Gráficos de resposta ao degrau, lugar das raízes e diagrama de pólos e zeros plotados através destes comandos são apresentados, respectivamente, nas Figuras Figura 28, Figura 29 e Figura 30.



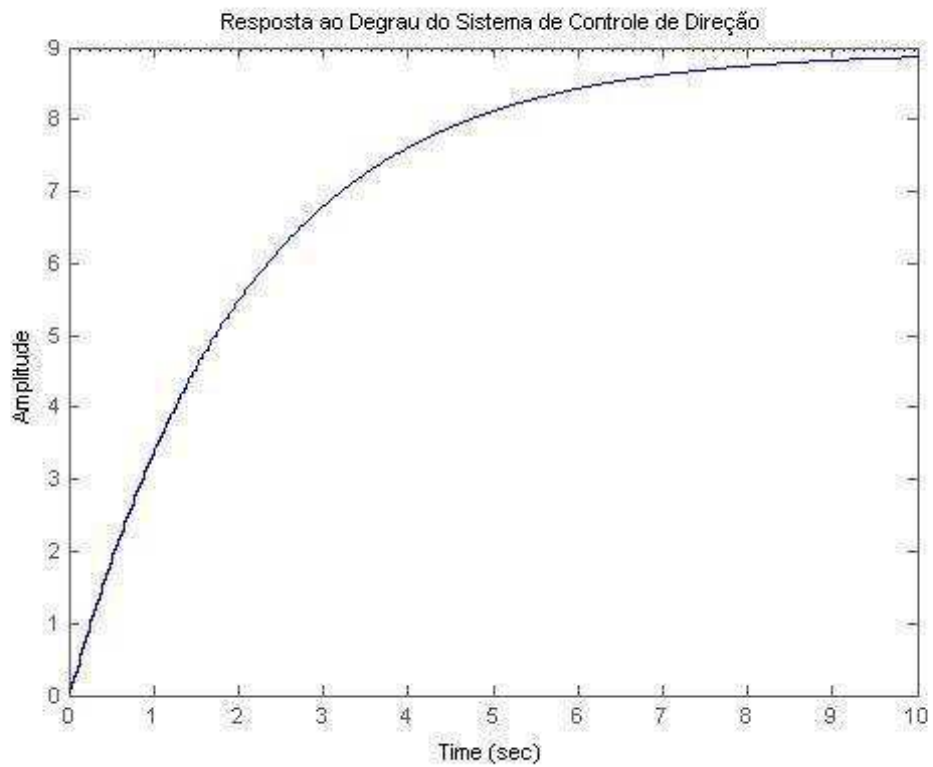


Figura 28: Resposta ao degrau do sistema de controle de direção plotada no MATLAB.

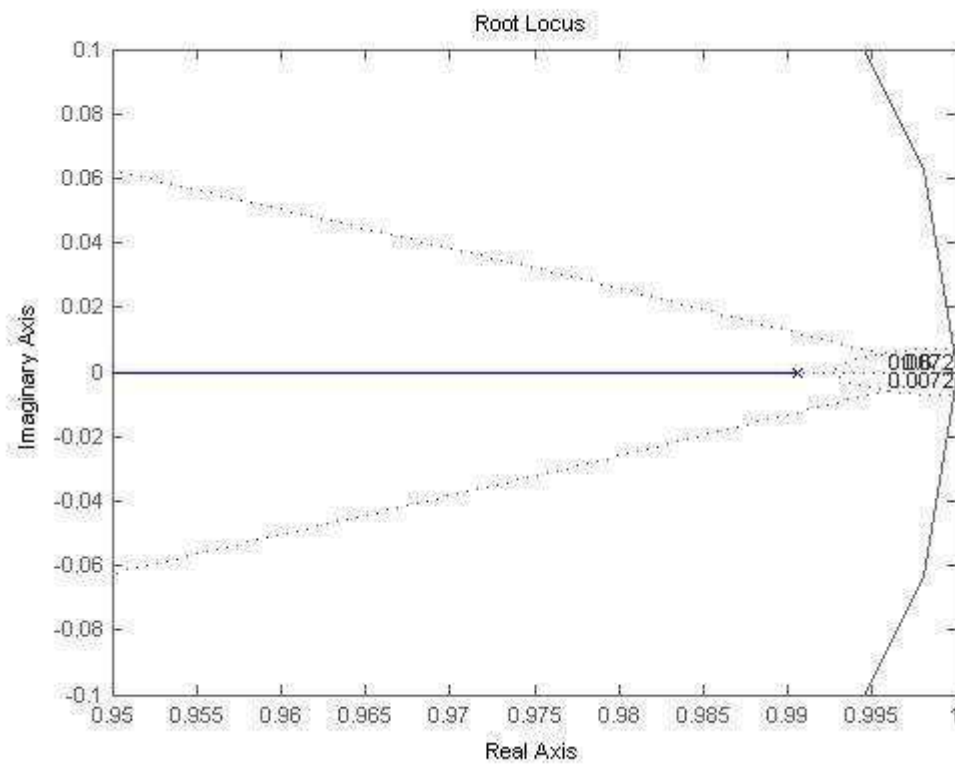


Figura 29: Lugar das raízes do sistema de controle de direção plotado no MATLAB.

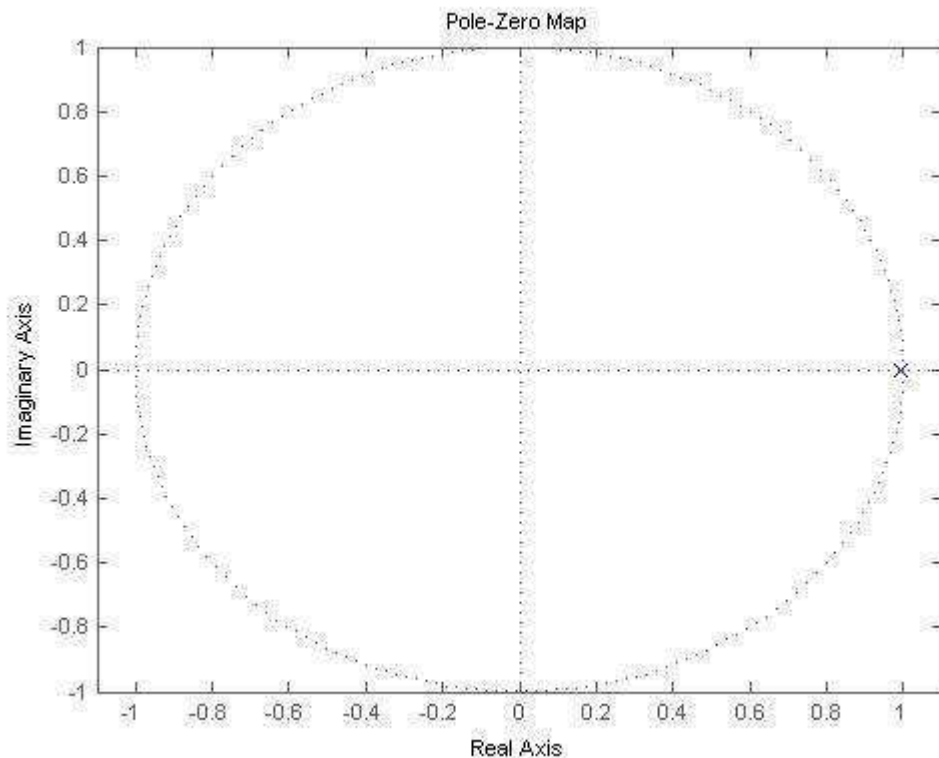


Figura 30: Diagrama de pólos e zeros do sistema de controle de direção plotado no MATLAB.

Nas técnicas demonstradas até este ponto, foi assumido que todas as variáveis de estado do sistema são mensuráveis. Esta suposição pode não ser válida em todos os casos. Na apostila é descrito um método para projetar um observador de ordem completa para estimar o estado do sistema com base nas variáveis medidas e no modelo da planta.

Uma vez que o observador irá estimar os valores das variáveis de estado que estão em constante mudança, é desejado que a dinâmica do observador seja significativamente mais rápida do que a dinâmica do sistema de circuito fechado sem o observador. Na Figura 31 é ilustrada a resposta ao degrau de um sistema de pêndulo invertido com duas variáveis, posição do carrinho e desvio angular do pêndulo, que utiliza um observador para estimar suas variáveis de estado.

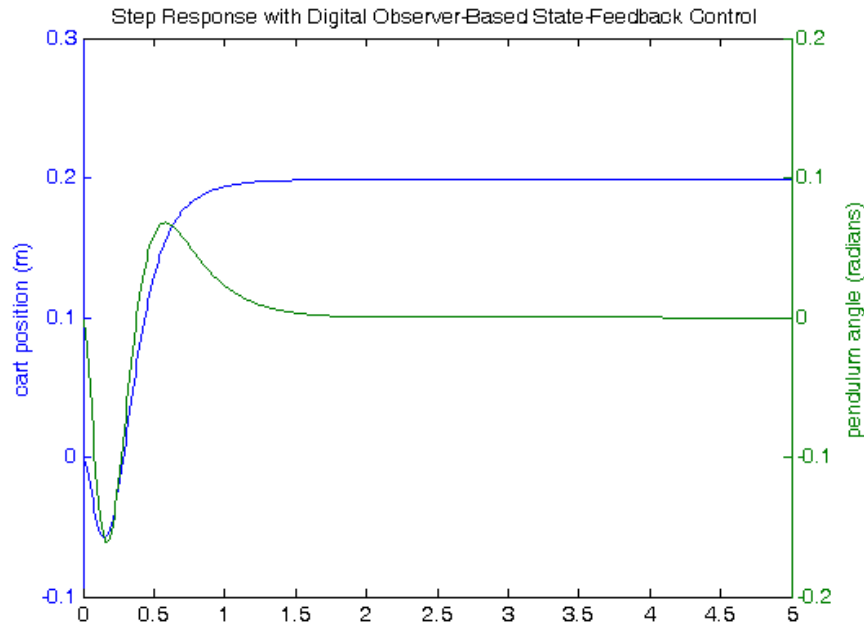


Figura 31: Resposta ao degrau de um sistema com duas variáveis de saída usando observador.

### 3.8 MODELAGEM USANDO SIMULINK

É muito simples simular modelos matemáticos de sistemas físicos representados graficamente através de diagramas de blocos no Simulink. Uma ampla gama de blocos são disponibilizados ao usuário em bibliotecas fornecidas para representar vários fenômenos e modelos em uma variedade de formatos. Na apostila “Sistemas Dinâmicos nos Ambientes MATLAB e Simulink”, são apresentadas maneiras de analisar rapidamente a resposta de sistemas complexos e aproximar soluções numéricas para estes modelos.

Em cada capítulo da apostila são construídos modelos dos sistemas utilizando blocos das bibliotecas padrão do Simulink. Na Figura 32 é ilustrado um modelo para o motor DC cuja variável de saída é a velocidade. Na apostila também é descrito como criar subsistemas a partir do modelo criado e como extraí-los para o MATLAB possibilitando análises posteriores.

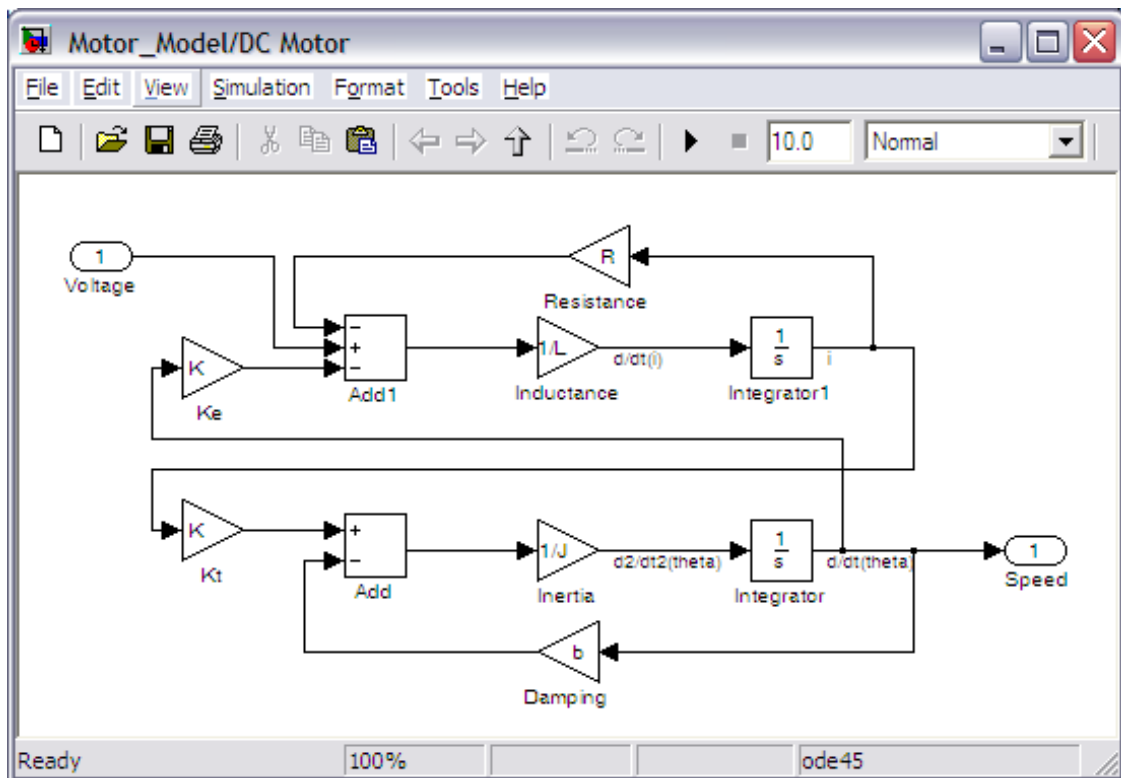


Figura 32: Modelo de motor DC com a velocidade como saída construído no Simulink. Fonte: Messner e Tilbury (2012).

Alternativamente, é possível construir o modelo do Motor DC usando os blocos de extensão do Simscape no Simulink. Como os blocos na biblioteca Simscape representam componentes físicos reais, os modelos multi-domínio complexos podem ser construídos sem a necessidade de formular equações matemáticas a partir de princípios físicos. O modelo construído com blocos desta biblioteca, exibido na Figura 33, é equivalente ao do motor DC da Figura 32.

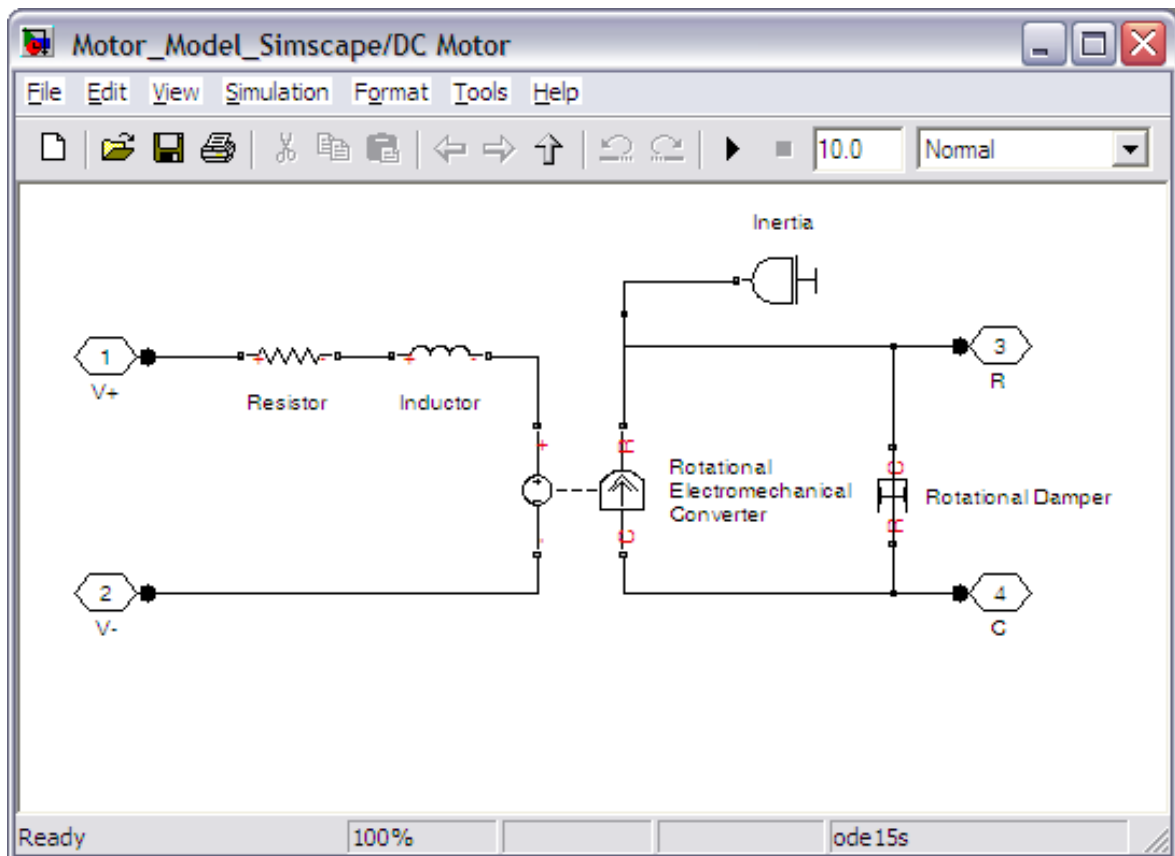


Figura 33: Modelo de motor DC com a velocidade como saída construído com blocos da biblioteca Simscape. Fonte: Messner e Tilbury (2012).

### 3.9 PROJETO DE CONTROLADORES USANDO SIMULINK

Na apostila “Sistemas Dinâmicos nos Ambientes MATLAB e Simulink” é demonstrado como é possível usar a ferramenta SISO para sintonizar controladores adicionados através de diagramas de blocos. A ferramenta funciona da mesma maneira, tanto para modelos construídos no MATLAB, quanto para aqueles construídos no Simulink. Para isso, basta especificar o bloco controlador que será sintonizado na tela apresentada na Figura 34.

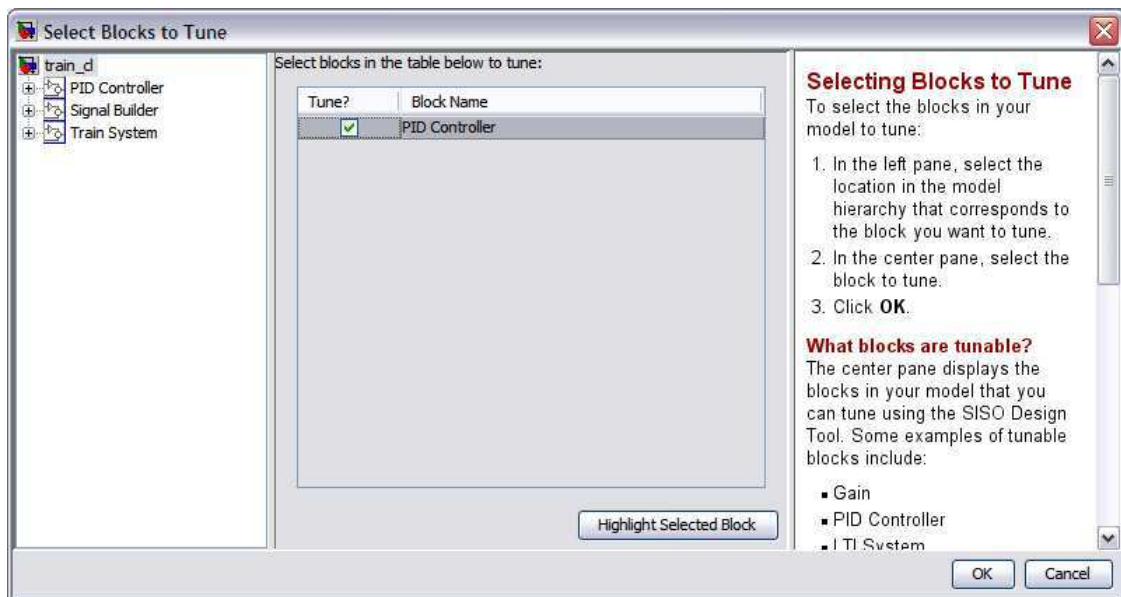


Figura 34: Especificação do bloco que será sintonizado na ferramenta SISO.

Além da ferramenta SISO, o controlador pode ser sintonizado automaticamente por uma configuração sugerida pelo Simulink. Isto é feito diretamente no diagrama de blocos, como descrito em procedimento na apostila.

Outra aplicação interessante do Simulink é poder simular os sistemas de controle projetados no MATLAB e poder verificar graficamente com facilidade todas as variáveis de estado, e traçar a resposta do sistema a uma grande variedade de sinais. A resposta ao degrau de um sistema de pêndulo invertido traçada no Simulink é ilustrada na Figura 35.

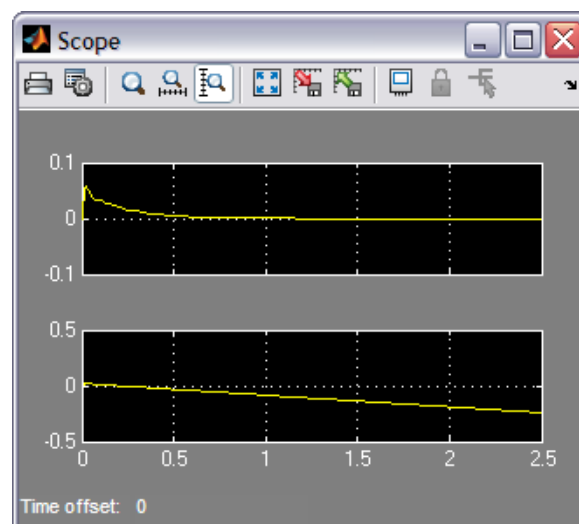


Figura 35: Resposta ao degrau de um sistema de pêndulo invertido plotada através do Simulink.

## 4 RESULTADOS

O resultado deste trabalho pode ser conferido diretamente na apostila “Sistemas Dinâmicos nos Ambientes MATLAB e Simulink”, que está disponível no apêndice A deste relatório. Ela é constituída por sete capítulos (Introdução, Controle de Direção, Velocidade de Motor, Posição de Motor, Suspensão, Controle de Aeronaves e Pêndulo Invertido), e cada capítulo possui nove seções (Modelagem, Análise, PID, Lugar das Raízes, Resposta em Frequência, Espaço de Estados, Controle Digital, Simulink: Modelagem e Simulink: Controle).

O material faz uso de códigos-texto, gráficos, telas das ferramentas e diagramas de blocos para ilustrar de maneira direta como o estudante pode chegar aos mesmos resultados apresentados na apostila. Desta forma, além de abranger seu conhecimento na simulação de processos, o aluno deve ser capaz de escolher qual técnica usar para projetar sistemas de controle e simular sua aplicação à planta de seu estudo.

## 5 CONSIDERAÇÕES FINAIS

Este projeto deve ser de grande valia para os alunos das disciplinas Controle Analógico e Laboratório de Controle Analógico, que agora têm à sua disposição um material didático atualizado e suficiente para integrar seus conhecimentos em sistemas dinâmicos à capacidade computacional do MATLAB.

Todas as ferramentas e funções utilizadas no material são padrão no MATLAB R2013, desta forma o estudante pode fazer uso dos códigos presentes na apostila sem preocupar-se em instalar bibliotecas adicionais para possibilitar seu funcionamento.

Futuramente, projeta-se que a apostila seja dividida e organizada em forma de guias de experimentos, que possam ser seguidos pelos alunos do Laboratório de Controle Analógico durante as aulas. Com isso, o texto resultante deste projeto deverá ser o material de referência principal da disciplina.



## BIBLIOGRAFIA

DORF, R.; BISHOP, R. **Sistemas de Controle Moderno**. 8va. ed. [S.l.]: LTC, 1998.

FRANKLIN, G.; POWELL, J. D.; EMAMI-NAEINI, A. **Feedback Control of Dynamic Systems**. 4ta. ed. [S.l.]: Prentice-Hall, 2002.

GREWAL, M. S.; ANDREWS, A. P. **Kalman Filtering: Theory and Practice Using MATLAB**. 2da. ed. [S.l.]: John Wiley & Sons, 2001.

MATHWORKS, INC. Guia do Usuário do Simulink, 2012. Disponível em: <<http://www.mathworks.com/products/datasheets/pdf/simulink.pdf>>. Acesso em: 07 Outubro 2013.

MATHWORKS, INC. Mathworks - MATLAB and Simulink for Technical Computing, 2013. Disponível em: <<http://www.mathworks.com/>>. Acesso em: 07 Outubro 2013.

MESSNER, B.; TILBURY, D. Control Tutorials for MATLAB and Simulink. **University of Michigan**, 2012. Disponível em: <<http://ctms.engin.umich.edu/>>. Acesso em: 30 Agosto 2013.

NATIONAL INSTRUMENTS. National Instruments Brasil, 2013. Disponível em: <<http://www.ni.com/cms/images/devzone/tut/4block3.gif>>. Acesso em: 05 Outubro 2013.

OGATA, K. **Engenharia de Controle Moderno**. 3ra. ed. [S.l.]: Prentice-Hall, 1997.

THE MATHWORKS, INC. **Using Simulink and Stateflow in Automotive Applications**. [S.l.]: The MathWorks, Inc., 1998.