



UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
UNIDADE ACADÊMICA DE ENGENHARIA ELÉTRICA
LABORATÓRIO DE INSTRUMENTAÇÃO ELETRÔNICA E
CONTROLE

FELIPE LIRA FURTADO

ELABORAÇÃO DE GUIAS DE EXPERIMENTOS
UTILIZANDO AS BANCADAS DO SERVOMOTOR
DE PRECISÃO E DO LEVITADOR MAGNÉTICO

Campina Grande
Maio de 2014

FELIPE LIRA FURTADO

ELABORAÇÃO DE GUIAS DE EXPERIMENTOS
UTILIZANDO AS BANCADAS DO SERVOMOTOR
DE PRECISÃO E DO LEVITADOR MAGNÉTICO

Trabalho de Conclusão de Curso no curso de graduação
em Engenharia Elétrica na Universidade Federal de
Campina Grande

Área de concentração: Controle e Automação

Orientação: Prof. Doutor George Acioli Jr.

Campina Grande

Maio de 2014

Resumo

Este trabalho tem como objetivo a elaboração de guias de experimentos para a disciplina Laboratório de Controle Digital. No decorrer do trabalho foi estudado teoria de identificação de sistemas utilizando a técnica dos Mínimos Quadrados e o método de projeto alocação de polos utilizando o espaço de estados.

Além disso, serão descritos as características relevantes da bancada Servomotor DC de Feedback Instruments. Por fim, serão mostrados os resultados deste trabalho que são os guias de experimentos elaborados.

Palavras-Chave: Servomotor de Precisão, Identificação, Alocação de Polos

Abstract

This work aims the elaboration of experiment guides for the discipline of Digital Control Laboratory. Along the work it was studied the theory of system identification using the Least Squares method and the project method of pole placement using state spaces.

Beyond that, some relevant characteristics of the Precision Modular Servo from Feedback Instruments will be described. At last, the results of this work will be shown, which are the guides for the elaborated experiments.

Keywords: Precision Modular Servo, Identification, Pole-Placement

Sumário

1	Introdução	1
1.1	Revisão Bibliográfica	1
1.2	Introdução Geral	1
1.3	Objetivo Geral	2
1.4	Objetivo Específico	2
1.5	Organização do Texto	2
2	Fundamentação Teórica	4
2.1	Fundamentação Teórica	4
2.2	Alocação de Polos Utilizando Representação em Espaço de Estados	5
2.2.1	Generalidades de Sistemas de Controle	5
2.2.2	Discretização de Sistemas	5
2.2.3	Realimentação de Estados Linear	6
2.2.4	Projeto do Observador de Estados	7
2.2.5	Projeto do Sistema de Controle Completo	9
2.2.6	Exemplo Prático - Alocação de Polos para a planta Integrador Duplo	11
2.3	Identificação de Sistemas	13
2.3.1	Mínimos Quadrados	14
2.3.2	Resposta ao Degrau	16
2.3.3	Modelagem a partir da Resposta ao Degrau	18
2.4	Linearização e Escalonamento	21
2.4.1	Linearização de Pequenos Sinais	21
2.5	Considerações	22
3	Plataformas Didáticas para Realização dos Experimentos	23
3.1	Introdução	23
3.2	Material Utilizado	23
3.3	Plataforma Didática - Servomotor de Precisão	24
3.3.1	Unidade Mecânica	25
3.3.2	Unidade Elétrica	26
3.3.3	Sistema de Controle	28
3.3.4	Modelo do Motor DC	29
3.3.5	Controle por computador	32
3.3.6	Projeto de Controladores	33
3.4	Plataforma Didática - Levitador Magnético	36

3.4.1	Descrição da Unidade Magnética	36
3.4.2	Sistema de Controle	36
3.4.3	Controle do Computador	37
3.4.4	Modelo de Levitação Magnética	38
3.4.5	Linearização do Modelo	41
3.4.6	Identificação de Modelos no Levitador Magnético	42
3.4.7	Implementação de controle do Maglev	44
3.5	Considerações	44
4	Experimentos Propostos	46
4.1	Introdução	46
4.2	Experimento 1 - Servo Motor de Precisão - Alocação de polos utilizando o Espaço de Estados	46
4.2.1	Atividade de Preparação	46
4.2.2	Atividade Experimental	47
4.2.3	Resoluções	48
4.3	Experimento 2 - Servo Motor de Precisão - Identificação de Sistemas	49
4.3.1	Atividade de Preparação	49
4.3.2	Atividade Experimental	51
4.3.3	Resoluções	55
4.4	Experimento 3 - Levitador Magnético - Linearização e Método do Relé	62
4.4.1	Atividade de Preparação	62
4.4.2	Atividade Experimental	62
4.4.3	Resoluções	64
4.5	Considerações	66
5	Conclusão	69
A	Apêndice	72

Lista de Figuras

2.1	Diagrama de blocos do controlador obtido com a realimentação linear de estados	7
2.2	Diagrama de blocos do controlador obtido combinando a realimentação linear de estados com o observador.	9
2.3	Diagrama de blocos introduzindo um sinal de comando no controlador com realimentação linear de estados e um observador	10
2.4	Valores reais e estimados para uma malha de controle completo de um integrador duplo	13
2.5	Resposta ao degrau para processos de característica não-oscilatória.	16
2.6	Resposta ao degrau pra processos de característica oscilatória	17
2.7	Teste de resposta ao degrau	18
2.8	Saída $u(t)$ com Relé com entrada $e(t)$	20
2.9	Processo com realimentação por relé.	20
2.10	Curva de Nyquist de um processo G_p de função descritiva de um relé $N(a)$	21
3.1	Visão geral de um módulo servomotor	25
3.2	Layout da bancada do módulo servomotor	26
3.3	Unidade Mecânica do Módulo Servomotor	27
3.4	Unidade Elétrica do Módulo Servomotor	27
3.5	Sistema de controle do Módulo Servomotor	28
3.6	Modelo elétrico-mecânico do motor DC	29
3.7	Menu para inicialização do Servo Motor	31
3.8	Interface entre o Matlab e os arquivos do Servo Motor	31
3.9	Exemplo de resposta ao modelo do servo motor sintonizado (carga mínima)	32
3.10	Diagrama do sistema de controle por computador	33
3.11	Blocos de equipamentos externos no projeto do Servomotor usando o Matlab	34
3.12	Modelo de identificação	34
3.13	Sinal de Excitação utilizado na modelagem	35
3.14	Resposta ao degrau de um sistema identificado	35
3.15	Unidade Mecânica do Levitador Magnético	37
3.16	Esquemático do Sistema de Controle	37
3.17	Modelo Maglev	37
3.18	Diagrama do sistema de controle computacional	38
3.19	Modelo fenomenológico do Levitador Magnético	39
3.20	Modelo para controle de posição do Maglev	39
3.21	Menu para inicialização	40

3.22	Interface entre o Matlab e os arquivos do Levitador Magnético	40
3.23	Exemplo de resposta ao modelo não-linear do Levitador Magnético (nenhuma tensão de controle aplicada)	41
3.24	Exemplo de resposta ao modelo linear do Levitador Magnético (nenhuma tensão de controle aplicada)	42
3.25	Identificação de sistemas instáveis	43
3.26	Exemplo de Identificação: respostas do modelo e da planta em malha fechada	45
4.1	Modelo para identificação de Velocidade	51
4.2	Sinal de controle implementado durante a identificação de Velocidade	52
4.3	Sinal de controle implementado durante a identificação de Velocidade com Ruído Branco	52
4.4	Modelo para identificação de Posição	53
4.5	Programas das Preparações	56
4.6	Códigos da Atividade Experimental 1	57
4.7	Respostas dos modelos da Atividade Experimental 1	58
4.8	Códigos da Atividade Experimental 2	58
4.9	Respostas dos modelos da Atividade Experimental 2	59
4.10	Código e resposta do modelo da Atividade Experimental 3	59
4.11	Código e resposta do modelo da Atividade Experimental 4	60
4.12	Código e resposta do modelo da Atividade Experimental 5	61
4.13	Programa <i>Maglev_PD</i>	63
4.14	Programa <i>Maglev_PID</i>	64
4.15	Programa <i>Maglev_PD</i> em execução	65
4.16	Sinal desejado e sinal adquirido durante a execução do <i>Maglev_PD</i>	65
4.17	Programa <i>Maglev_PID</i> em execução	66
4.18	Sinal desejado e sinal adquirido durante a execução do <i>Maglev_PID</i>	67
4.19	Programa <i>Maglev_PD_Rele</i> em execução	67
4.20	Sinal desejado e sinal adquirido durante a execução do <i>Maglev_PD_Rele</i>	68
4.21	Sinal desejado e sinal adquirido durante a execução do <i>Maglev_PD_Rele</i>	68
A.1	Diagrama Elétrico do AD1 e AD2 na Interface de Controle Analógica	72
A.2	Diagrama Elétrico do DA1 e DA2 na Interface de Controle Analógica	73
A.3	Diagrama Elétrico do AD3 na Interface de Controle Analógica, AD4, AD5, AD6 e AD7 também são buffers com a mesma estrutura de AD3	73
A.4	Diagrama Elétrico do AD1 e AD2 na Interface de Controle Analógica	74
A.5	Diagrama Elétrico do DA1 e DA2 na Interface de Controle Analógica	74
A.6	Diagrama Elétrico do AD3 na Interface de Controle Analógica, AD4, AD5, AD6 e AD7 também são buffers com a mesma estrutura de AD3	75

Capítulo 1

Introdução

1.1 Revisão Bibliográfica

O aluno Sérgio Paulo Melo de Souza, da Universidade Federal de Campina Grande fez, em Agosto de 2011 e sob orientação do professor Dr. João Batista a elaboração de guias de experimentos utilizando a bancada Servomotor DC, esse trabalho foi o precursor do atual.

O aluno Daniel Vicentini Guimarães, da Escola de Engenharia de São Carlos fez, em 2011 e sob orientação do professor Manoel Luís de Aguiar a elaboração de um Kit Didático para aprendizado de controladores em tempo real utilizando matlab e o toolbox realtime workshop.

O aluno Marwan K. Abbadi, do departamento de Engenharia Elétrica e Computacional da Universidade de Bradley fez, em Maio de 2004 a modelagem e controle do Sistema de Levitação Magnética presentes no Kit Didático de Levitação Magnética da *Feedback Instruments* sob orientação do Professor Doutor Winfred Anakwa utilizando a linguagem Assembly em um microprocessador Intel 8051.

O aluno Nawroul Ibrahim Mahmoud, da divisão Aydelning, da universidade Linköpings fez, em Setembro de 2003 e sob supervisão de Magnus Åkerbald modelou e implementou um sistema de controle do tipo *Backstepping* em um levitador magnético da *Feedback Instruments* (O 33-210).

1.2 Introdução Geral

O estudo dos sistemas de controle é de fundamental importância para os estudantes de Engenharia Elétrica, visto que os conceitos empregados nesse estudo abrangem áreas da Engenharia que estão em constante crescimento como robótica e controle de máquinas.

Na automatização dos ciclos de produção, robôs e máquinas estão sendo cada vez mais utilizados para realizar tarefas do processo produtivo [André L. Maitelli]. Nesse contexto, problemas como a precisão e rapidez na execução das tarefas são solucionados por sistemas de controle. Portanto, o estudo de aplicações práticas desses sistemas se torna importante para estudantes da área. Neste trabalho propõe-se o desenvolvimento de experimentos de modelagem de plantas e projeto de controladores para a disciplina Laboratório

de Controle Digital utilizando a bancada de experimentos da *Feedback Instruments*.

O método de identificação e de projeto de sistemas de controle abordados nesse trabalho são Mínimos Quadrados, Alocação de Pólos e Discretização de Sistemas respectivamente. Todos assuntos são lecionados na disciplina Controle Digital. Com o método dos Mínimos Quadrados é possível obter um modelo do processo a partir de dados experimentais. Esse método é bastante útil quando se torna difícil ou mesmo impossível a obtenção de um modelo do processo a partir de princípios físicos. A alocação de polos visa projetar um controlador de modo que o sistema tenha a característica desejada.

1.3 Objetivo Geral

Embasar e aprofundar a utilização das bancadas de experimentos do servomotor de precisão e do levitador magnético, ambas da *Feedback Instruments*. Em seguida propor experimentos que utilizem a bancada Servomotor de Precisão e desenvolver guias destes para a disciplina Laboratório de Controle Digital.

1.4 Objetivo Específico

Elaboração dos processos:

- Modelagem de plantas e projeto de controladores digitais utilizando Alocação de Polos com realimentação de estados.
- Identificação de Sistemas utilizando o método dos Mínimos Quadráticos.

1.5 Organização do Texto

Este trabalho está dividido da seguinte forma:

- **Fundamentação Teórica:** No capítulo 2 são apresentados todos os conceitos teóricos necessários para o desenvolvimento deste trabalho. Eles se concentram em dois grandes tópicos, que são o de Alocação de Polos utilizando representação em espaço de estados e identificação de sistemas.
- **Plataformas Didáticas para Realização dos Experimentos:** No capítulo 3 são apresentadas as plataformas didáticas nas quais os experimentos foram desenvolvidos, são elas o Servomotor de Precisão e o Levitador Magnético, elas são exploradas tanto em termos de software quanto de hardware.
- **Experimentos Propostos:** No capítulo 4 são apresentados os resultados obtidos por meio de simulações e experimentos durante o desenvolvimento deste trabalho como também é feita uma breve descrição sobre os guias de experimentos que foram elaborados, assim como apresentadas possíveis resoluções dos exercícios.
- **Conclusão**

- **Anexos:** Nesta seção são apresentados os guias de experimentos elaborados, os quais foram um dos objetivos desse trabalho.

Capítulo 2

Fundamentação Teórica

2.1 Fundamentação Teórica

Nesse capítulo são apresentados alguns dos conceitos mais fundamentais para o entendimento dos experimentos propostos nos capítulos seguintes. Na seção 2.2 é apresentado o embasamento teórico referente a Alocação de Polos utilizando representação em espaço de estados, ela apresenta métodos de projeto baseados em modelos internos do sistema que podem ser encarados como soluções para problemas específicos e idealizados de controle. Esse capítulo é subdividido em duas grandes seções.

Na primeira seção se introduz um conjunto de métodos de projetos que são nomeados alocação de polos no ponto de vista de espaços de estados. O nome "alocação de polos" se refere ao fato de que o projeto é formulado visando obter um sistema em malha fechada com os polos especificados. Já a sub-seção 2.2.2 mostra como obter a versão discretizada de um modelo contínuo. Em seguida 2.2.3 mostra como obter a resposta desejada quando já se possui os polos desejados e o sistema discretizado. Na sub-seção 2.2.4 é discutido o problema da reconstrução de estados através de saídas previamente medidas, para isso se faz necessário a introdução do conceito de observadores. Na 2.2.5 é apresentado como adquirir um sistema de controle completo e então discutido o problema servo, ou seja, como obter um sistema que possa também seguir sinais de comando. Por fim, na sub-seção 2.2.6 mostra-se um exemplo que engloba boa parte dos conceitos abordados na seção.

Em seguida a seção 2.3 aborda uma introdução geral sobre o conceito do assunto abordado e sua aplicabilidade. Existem vários métodos de analisar os dados obtidos por experimentos visando adquirir o modelo do sistema, dentre eles, na sub-seção 2.3.1 é apresentado um método básico utilizado nesse trabalho, que é o método dos Mínimos Quadrados. Em seguida, nas sub-seções 2.3.2 e 2.3.3 são apresentadas algumas entradas que podem ser inseridas em uma planta desconhecida para realizar o processo de identificação, como degrau e relé, assim como sua modelagem. Por fim são apresentados na seção 2.4 conceitos básicos sobre linearização, que permitem trabalhar com não-linearidades de modo relativamente preciso e serão úteis quando houver a realização dos experimentos com o kit didático Levitador Magnético.

2.2 Alocação de Polos Utilizando Representação em Espaço de Estados

2.2.1 Generalidades de Sistemas de Controle

Alguns fatores devem ser considerados no projeto de sistemas de controle, por exemplo:

1. Atenuação das perturbações de carga;
2. Redução do efeito do ruído de medição;
3. Rastreamento do Sinal de Controle;
4. Variações e incertezas no comportamento do processo.

Os problemas de controle podem ser classificados em problemas de regulação e em problema servo. No primeiro, a principal questão é o compromisso entre a redução das perturbações de carga e as flutuações criadas pelo ruído de medição que é injetado no sistema devido à realimentação. No segundo, a principal questão é o rastreamento de um sinal de referência.

A maioria dos métodos de projeto dá foco em alguns dos aspectos mencionados acima e o projetista tem que checar se os outros requisitos também são satisfeitos. Neste guia iremos desenvolver um método de projeto baseado em modelos de estado cujo propósito é obter o polinômio característico desejado do sistema em malha fechada. Este método é fácil de ser aplicado em sistemas de ordem baixa, mas pode ser difícil escolher os polos adequados para sistemas de ordem elevada.

Assumiremos inicialmente que o processo a ser controlado é descrito pelo modelo

$$\begin{aligned}\frac{dx}{dt} &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t)\end{aligned}\tag{2.1}$$

Em que u representa a variável de controle, x representa o vetor de estados, y representa o vetor da saída e A , B , C e D são matrizes constantes, A é a matriz de estado, B é a matriz de entrada, C é a matriz de Saída e D é a matriz de transmissão direta. Todas as informações e demonstrações presentes nesse capítulo foram originárias de [7].

2.2.2 Discretização de Sistemas

Considere a versão discreta do modelo apresentado em (2.1):

$$\begin{aligned}x(kh + h) &= \phi x(kh) + \Gamma u(kh) \\ y(kh) &= Cx(kh) + Du(kh)\end{aligned}\tag{2.2}$$

em que

$$\begin{aligned}\phi &= e^{Ah} \\ \Gamma &= \int_0^h e^{As} ds \cdot B\end{aligned}\tag{2.3}$$

Para simplificar, consideramos $h=1$ no modelo (2.2):

$$\begin{aligned} x(k+1) &= \phi x(k) + \Gamma u(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned} \quad (2.4)$$

em que ϕ é o equivalente discreto da matriz de estado e Γ é o equivalente discreto da matriz de entrada

2.2.3 Realimentação de Estados Linear

Como queremos que o sistema em malha fechada tenha o polinômio característico desejado, ou seja, as suas propriedades serão especificadas pelos polos desse sistema, é natural admitirmos que a realimentação de estados seja linear;

$$u(k) = -Lx(k) \quad (2.5)$$

Assumimos que o sistema é descrito pelas equações (2.1) e que o período de amostragem é tal que o processo pode ser descrito pelo sistema discreto das equações (2.2). O objetivo é encontrar uma lei de realimentação linear na forma (2.5) tal que o sistema em malha fechada tenha uma equação característica especificada. Seja o sistema descrito por (2.2) e o polinômio característico da matriz ϕ seja dado por:

$$z^n + a_1 z^{n-1} + \dots + a_n \quad (2.6)$$

Se o sistema representado pela equação (2.2) é alcançável, então podemos transformá-lo para sua forma canônica pela transformação $z = T \cdot x$, resultando em:

$$z(k+1) = \tilde{\phi} \cdot z(k) + \tilde{\Gamma} u(k) \quad (2.7)$$

onde

$$\tilde{\phi} = \begin{vmatrix} -a_1 & -a_2 & \dots & -a_{n-1} & -a_n \\ 1 & 0 & & 0 & 0 \\ 0 & 1 & & 0 & 0 \\ \vdots & \vdots & & 0 & 0 \\ 0 & 0 & \dots & 1 & 0 \end{vmatrix} \quad (2.8)$$

$$\tilde{\Gamma} = \begin{vmatrix} 1 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{vmatrix} \quad (2.9)$$

Os coeficientes do polinômio característico que determinam os polos de malha fechada aparecem explicitamente nessa representação. De (2.5) podemos tirar a lei de realimentação

$$u = \tilde{L}x = - \begin{vmatrix} p_1 - a_1 & p_2 - a_2 & \dots & p_n - a_n \end{vmatrix} z \quad (2.10)$$

dá um sistema em malha fechada com o polinômio característico.

$$P(z) = z^n + p_1 z^{n-1} + \dots + p_n \quad (2.11)$$

Podemos encontrar a solução do problema original transformando para as coordenadas originais.

$$u = -\tilde{L}z = -\tilde{L}Tx = -Lx \quad (2.12)$$

Resta determinar a matriz de transformação T. Uma forma simples de determinar essa matriz é baseada na propriedade das matrizes de alcançabilidade. Seja W_c a matriz de alcançabilidade dada por:

$$W_c = [\Gamma \quad \phi\Gamma \quad \dots \quad \phi_{n-1}\Gamma] \quad (2.13)$$

E seja \tilde{W}_c a matriz alcançabilidade de (2.7). As matrizes são relacionadas por $\tilde{W}_c = TW_c$. Logo,

$$T = \tilde{W}_c \cdot W_c^{-1} \quad (2.14)$$

Portanto, o cálculo da matriz de realimentação linear L é dado por

$$\begin{aligned} L &= (p_1 - a_1 \quad p_2 - a_2 \quad \dots \quad p_n - a_n) \cdot T \\ L &= (p_1 - a_1 \quad p_2 - a_2 \quad \dots \quad p_n - a_n) \cdot P(\phi) \end{aligned} \quad (2.15)$$

Onde p_1, \dots, p_n são os polos de malha fechada do sistema. A equação (2.15) é conhecida como fórmula de *Ackermann*.

O diagrama de blocos de um sistema com realimentação linear é mostrado na figura (2.1).



Figura 2.1: Diagrama de blocos do controlador obtido com a realimentação linear de estados

Especificamente, $P(z)$ é dado por:

$$P(z) = z^n + p_1 \cdot z^{n-1} + \dots + p_n \cdot I \quad (2.16)$$

2.2.4 Projeto do Observador de Estados

Nem sempre é possível medir todos os estados de um sistema. Portanto, é interessante determinarmos os estados de um sistema a partir das medidas disponíveis e de um modelo. Assumindo que o sistema é descrito por (2.17), o problema consiste então em calcular ou reconstruir o estado $x(k)$ das sequências de entradas e saídas $y(k), y(k-1), \dots, u(k), u(k-1), \dots$. Isso é possível se o sistema é observável ([7]).

$$\begin{aligned} x(k+1) &= \phi x(k) + \Gamma u(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned} \quad (2.17)$$

Esse método de reconstrução tem a vantagem de ser pouco sensível ao ruído. Considerando o sistema em (2.4) e assumindo que o estado x deve ser aproximado pelo estado \hat{x} do modelo

$$\hat{x}(k+1) = \phi \hat{x}(k) + \Gamma u(k) \quad (2.18)$$

que tem a mesma entrada do sistema (2.4)

Podemos dizer que este modelo é perfeito no sentido de que os elementos das matrizes ϕ e Γ são idênticos àqueles do sistema (2.4) e se as condições iniciais forem as mesmas, então o estado \hat{x} do modelo (2.18) será igual ao estado x do sistema real em (2.4). Se as condições forem diferentes, então \hat{x} convergirá para x apenas se o sistema tiver estabilidade assintótica. A reconstrução pode ser melhorada usando também as saídas medidas. Isto pode ser feito introduzindo uma realimentação da diferença entre as saídas medidas e estimadas $y - C\hat{x}$. Portanto,

$$\hat{x}(k+1|k) = \phi \hat{x}(k|k-1) + \Gamma u(k) + K(y(k) - C\hat{x}(k|k-1)) \quad (2.19)$$

onde K é uma matriz de ganho. A notação $\hat{x}(k+1|k)$ significa que a estimativa $x(k+1)$ é baseada nas medições disponíveis até o tempo k . Note que o termo $K(y(k) - C\hat{x}(k|k-1))$ não gera contribuição se a saída estimada é igual a medida. Para calcular a matriz K , introduzimos o erro de reconstrução

$$\tilde{x} = x - \hat{x} \quad (2.20)$$

Subtraindo (2.19) de (2.4), temos

$$\begin{aligned} \tilde{x}(k+1|k) &= \phi \tilde{x}(k|k-1) + K(y(k) - C\hat{x}(k|k-1)) \\ &= (\phi - KC) \tilde{x}(k|k-1) \end{aligned} \quad (2.21)$$

Portanto, se K for escolhido tal que o sistema (2.21) seja assintoticamente estável, o erro \hat{x} irá convergir para zero.

O sistema em (2.19) é chamado observador do sistema (2.4) por que ele produz o estado do sistema a partir de medições das entradas e saídas e permite enunciar o Teorema 1. Agora o problema consiste em encontrar uma matriz K tal que $\phi - KC$ tenha autovalores prescritos.

Teorema 1 (Dinâmica do Observador). *Considere um sistema discreto dado por (2.4). Seja $P(z)$ um polinômio de grau n , onde n é a ordem do sistema. Assumindo que o sistema é completamente observável, então existe uma matriz K tal que a matriz $\phi - KC$ do observador (2.19) tenha polinômio característico $P(z)$.*

A determinação da matriz K no observador (2.19) é feita de forma análoga a determinação da matriz L na realimentação de estados. O cálculo da matriz K é dado por:

$$K = P(\phi) \cdot W_C^{-1} \cdot (0 \ \dots \ 0 \ 1)^T \quad (2.22)$$

2.2.5 Projeto do Sistema de Controle Completo

Na sessão 2.2.3 consideramos que todas as variáveis de estados são conhecidas ou medidas diretamente para calcularmos a matriz de ganho de realimentação linear L . Na sessão 2.2.4 resolvemos o problema de calcular os estados a partir das saídas do sistema. Agora iremos combinar essas duas soluções para obtermos a solução desejada para o problema de alocação de polos. Seja o sistema descrito por:

$$\begin{aligned} z(k+1) &= \phi x(k) + \Gamma u(k) \\ y(k) &= Cx(k) \end{aligned} \quad (2.23)$$

Uma lei de realimentação linear que relaciona u para y tal que o sistema em malha fechada tenha os polos desejados, assumindo que todas as variáveis de estados são medidas, é dado pela equação (2.24).

$$u(k) = -Lx(k) \quad (2.24)$$

Porém, considerando que nem todos os estados possam ser medidos, a lei de realimentação linear muda para:

$$u(k) = -L\hat{x}(k|k-1) \quad (2.25)$$

onde \hat{x} é obtido do observador dado pela equação (2.19)

O sistema em malha fechada tem as propriedades desejadas. Para mostrar isso, introduzimos:

$$\tilde{x} = x - \hat{x} \quad (2.26)$$

Das equações (2.23) e (2.25), temos que o sistema é descrito por:

$$\begin{aligned} x(k+1) &= (\phi - \Gamma L).x(k) + \Gamma L.\hat{x}(k|k-1) \\ x(k+1|k) &= (\phi - KC).\tilde{x}(k|k-1) \end{aligned} \quad (2.27)$$

Podemos ver que o sistema tem ordem $2n$ e os autovalores do sistema em malha fechada são os autovalores das matrizes $\phi - \Gamma L$ e $\phi - KC$. Obtemos então a figura (2.2), que relaciona o controlador obtido combinando a realimentação linear de estados com o observador.

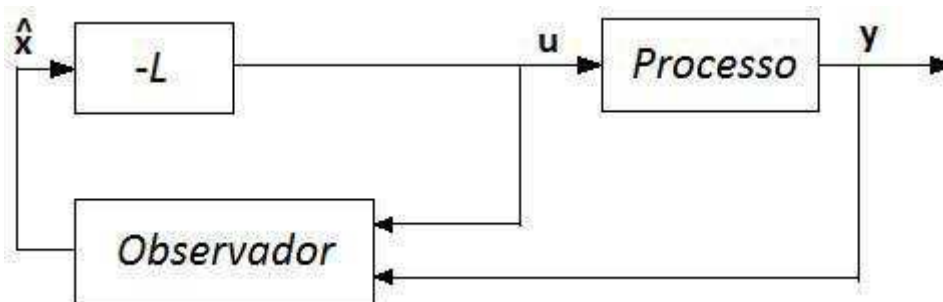


Figura 2.2: Diagrama de blocos do controlador obtido combinando a realimentação linear de estados com o observador.

O controlador pode ser visto como uma caixa preta que gera o sinal de controle a partir da saída do processo. O controlador descrito por (2.28) e (2.25) pode ser representado por uma função de transferência ao pulso da saída medida y para o sinal de controle u dada pela equação (2.29):

$$\hat{x}(k+1|k) = \phi\hat{x}(k|k-1) + \Gamma u(k) + K(y(k) - C\hat{x}(k|k-1)) \quad (2.28)$$

$$H_c(z) = -L.(zI - \phi + \Gamma L + KC)^{-1}.K \quad (2.29)$$

O problema servo

O problema servo tem por objetivo fazer os estados e as saídas do sistema responderem de uma maneira especificada a sinais de comando. Uma forma simples de obtermos a resposta desejada a sinais de comando é substituir a realimentação de estados (2.25) por

$$u(k) = -L\hat{x}(k) + L_c u_c(k) \quad (2.30)$$

em que u_c é o sinal de comando. Agora o sistema em malha fechada pode ser descrito por

$$\begin{aligned} x(k+1) &= (\phi - \Gamma L).x(k) + \Gamma L.\tilde{x}(k) + \Gamma L_c u_c \\ \hat{x}(k+1) &= (\phi - KC).\tilde{x}(k) \\ y(k) &= Cx(k) \end{aligned} \quad (2.31)$$

onde $\tilde{x} = x - \hat{x}$. Obtemos então a figura (2.3), que mostra em diagrama de blocos como o sinal de comando é introduzido no controlador obtido na sessão passada.

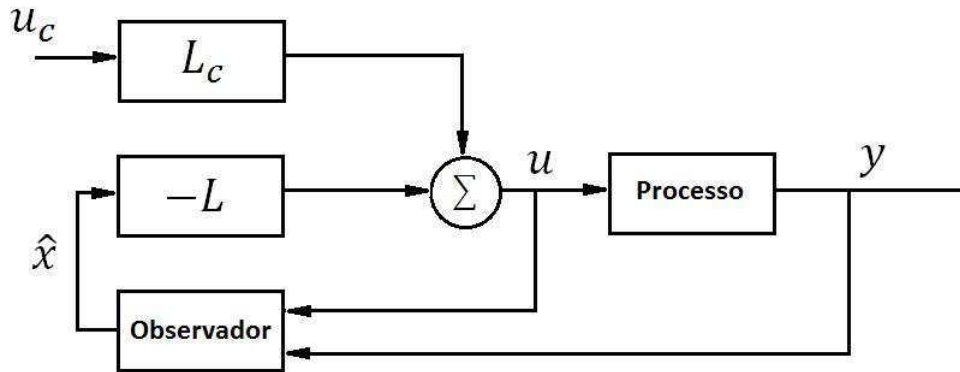


Figura 2.3: Diagrama de blocos introduzindo um sinal de comando no controlador com realimentação linear de estados e um observador

A função de transferência ao pulso do sinal de comando para a saída do processo pode ser encontrada a partir das equações em (2.31), que dá

$$H_{cl}(z) = C.(zI - \phi - \Gamma L)^{-1}.\Gamma L_c = L_c \frac{B(z)}{A_m(z)} \quad (2.32)$$

em que A_m é a matriz de estado para o pulso do sinal de comando.

O sistema em malha fechada obtido pela lei de controle dada em (2.30) tem os mesmos zeros do processo e os polos são os autovalores da matriz $\phi - \Gamma L$.

2.2.6 Exemplo Prático - Alocação de Polos para a planta Integrador Duplo

Utilizando a convenção de amostragem no tempo, a planta do integrador duplo amostrada é descrita por:

$$x(k+1) = \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix} \cdot x(k) + \begin{bmatrix} \frac{h^2}{2} \\ h \end{bmatrix} \cdot u(k)$$

A realimentação linear pode ser descrita de modo generalizado como:

$$u = -l_1 \cdot x_1 - l_2 \cdot x_2$$

O que em malha fechada nos dá:

$$x(k+1) = \begin{bmatrix} 1 - \frac{l_1 \cdot h^2}{2} & h - \frac{l_2 \cdot h^2}{2} \\ -l_1 \cdot h & 1 - l_2 \cdot h \end{bmatrix} \cdot x(k)$$

A equação característica do sistema em malha fechada é:

$$z^2 + \left(\frac{l_1 \cdot h^2}{2} + l_2 \cdot h - 2 \right) \cdot z - \left(\frac{l_1 \cdot h^2}{2} - l_2 \cdot h + 1 \right) = 0$$

Assumindo que a equação característica desejada seja:

$$z^2 + p_1 \cdot z + p_2 = 0$$

Isso nos leva às seguintes equações lineares para l_1 e l_2

$$\begin{aligned} \frac{l_1 \cdot h^2}{2} + l_2 \cdot h - 2 &= p_1 \\ \frac{l_1 \cdot h^2}{2} - l_2 \cdot h + 1 &= p_2 \end{aligned}$$

Equações estas cuja solução é:

$$l_1 = \frac{1}{h^2} \cdot (1 + p_1 + p_2) \quad (2.33)$$

$$l_2 = \frac{1}{2 \cdot h} \cdot (3 + p_1 - p_2) \quad (2.34)$$

No caso do integrador duplo é sempre possível encontrar parâmetros de controle que fornecem uma equação característica arbitrária de um sistema em malha fechada. O sistema linear de equações para l_1 e l_2 possui solução para todos os valores de p_1 e p_2 .

Assume-se agora que a equação característica polinomial desejada é dada por $P(z) = z^2 + p_1 \cdot z + p_2$. Tem-se então.

$$W_c = \left| \begin{array}{cc} \Gamma & \phi \cdot \Gamma \\ \frac{h^2}{h} & \frac{3 \cdot h^2}{h} \end{array} \right|$$

Além disso,

$$W_c^{-1} = \left| \begin{array}{cc} \frac{-1}{h^2} & \frac{1.5}{h} \\ \frac{1}{h^2} & \frac{-0.5}{h} \end{array} \right|$$

O polinômio característico da matriz ϕ é $z^2 - 2 \cdot z + 1$. Logo

$$P(\phi) = \phi^2 + p_1 \cdot \phi + p_2 \cdot 1 = \left| \begin{array}{cc} 1 + p_1 + p_2 & 2 \cdot h + p_1 \cdot h \\ 0 & 1 + p_1 + p_2 \end{array} \right|$$

Pela fórmula de Ackermann (2.15) temos

$$\begin{aligned} L &= \left| \begin{array}{cc} 0 & 1 \end{array} \right| \cdot W_c^{-1} \cdot P(\phi) \\ &= \left| \begin{array}{cc} \frac{1}{h^2} & \frac{-1}{2 \cdot h} \end{array} \right| \cdot P(\phi) \\ &= \left| \begin{array}{cc} \frac{1+p_1+p_2}{h^2} & \frac{3+p_1-p_2}{2 \cdot h} \end{array} \right| \end{aligned} \quad (2.35)$$

Que é o resultado obtido em 2.33 e 2.34. Podemos em seguida encontrar um observador para a planta do integrador duplo. A matriz $\phi - K \cdot C$ é dada por:

$$\phi_0 = \phi - K \cdot C = \left| \begin{array}{cc} 1 & h \\ 0 & 1 \end{array} \right| - \left| \begin{array}{c} k_1 \\ k_2 \end{array} \right| \cdot \left| \begin{array}{cc} 1 & 0 \end{array} \right| = \left| \begin{array}{cc} 1 - k_1 & h \\ -k_2 & 1 \end{array} \right|$$

Logo a equação característica é dada por:

$$z^2 - (2 - k_1) \cdot z + 1 - k_1 + k_2 \cdot h = 0$$

Se a equação característica desejada é $z^2 + p_1 \cdot z + p_2 = 0$, as seguintes equações são obtidas:

$$\begin{aligned} 2 - k_1 &= -p_1 \\ 1 - k_1 + k_2 \cdot h &= p_2 \end{aligned}$$

Equações lineares essas que fornecem:

$$\begin{aligned} k_1 &= 2 + p_1 \\ k_2 &= \frac{1 + p_1 + p_2}{h} \end{aligned}$$

Simplifica-se então o observador para o caso *deadbeat*, que ocorre quando a matriz $\phi - K \cdot C$ possui todos os polos nulos. Esse observador possui a propriedade de que seu erro vai para zero em um tempo finito. Ou seja

$$\begin{aligned} k_1 &= 2 \\ k_2 &= \frac{1}{h} \end{aligned}$$

E o observador se torna

$$\begin{aligned} \hat{x}_1(k+1) &= \hat{x}_1(k) + h \cdot \hat{x}_2(k) + 2 \cdot (y(k) - \hat{x}_1(k)) \\ \hat{x}_2(k+1) &= \hat{x}_2(k) + \frac{1}{h} \cdot (y(k) - \hat{x}_1(k)) \end{aligned}$$

O que fornece, com cálculos diretos

$$\begin{aligned} \hat{x}_1(k+1) &= 2 \cdot y(k) - y(k-1) \\ \hat{x}_2(k+1) &= \frac{y(k) - y(k-1)}{h} \end{aligned}$$

Considera-se agora que o vetor L é desejado para a frequência natural $\omega = 1$, amortecimento $\xi = 0.7$ e $h = 0.44$. Isso significa $L = \begin{bmatrix} 0.73 & 1.21 \end{bmatrix}$. A figura 2.4 mostra o valor real e os estimados quando os valores estimados são usados para serem a lei de controle.

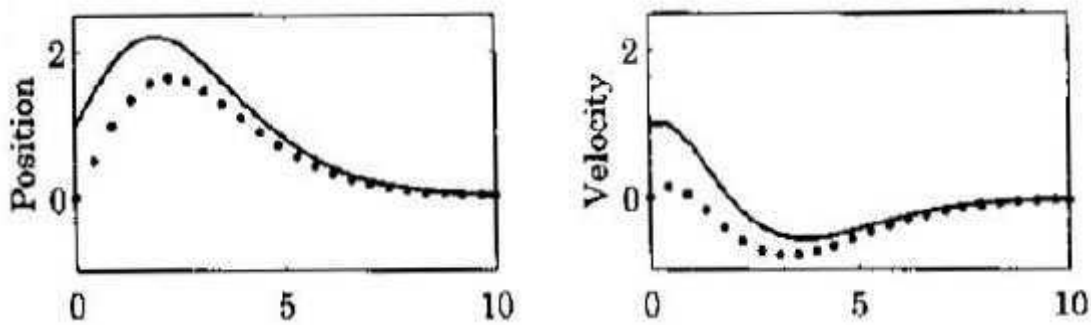


Figura 2.4: Valores reais e estimados para uma malha de controle completo de um integrador duplo

2.3 Identificação de Sistemas

A noção de modelos matemáticos é fundamental para a engenharia. Um modelo é uma forma bastante útil e compacta de representar o conhecimento sobre determinado processo.

Em alguns casos, os modelos podem ser obtidos a partir de princípios físicos. Porém, se torna muito difícil obtermos modelos de distúrbios, que são igualmente importantes, dessa forma. Estes modelos geralmente são obtidos de modo experimental.

Em princípio, existem duas formas diferentes para os quais os modelos podem ser obtidos: a partir de conhecimento prévio (por exemplo, através de leis físicas) ou por experimentações do processo.

Em muitos casos não é possível fazer um modelo completo apenas com conhecimentos específicos. Alguns parâmetros devem ser determinados através de experimentos. Esta abordagem é chamada identificação de sistemas. Existem muitos métodos para a análise de informações obtidas de experimentos. Um método simples, porém muito útil, é o método dos Mínimos Quadrados que será apresentado na próxima sessão

Todas as informações presentes nesse capítulo foram retiradas de [7] e [1], enquanto todas as demonstrações (exceto as de discretização) foram retiradas de [7].

2.3.1 Mínimos Quadrados

O método dos mínimos quadrados é tal que os parâmetros desconhecidos do modelo devem ser escolhidos de tal forma que: *”A soma dos quadrados das diferenças entre os valores observados e os calculados multiplicados por números que medem o grau de precisão seja mínimo.”* Para serem capazes de dar uma solução analítica, os valores calculados devem ser funções lineares dos parâmetros desconhecidos. No método dos mínimos quadrados a variável y é dada por:

$$\hat{y} = \theta_1 \cdot \varphi_1 + \theta_2 \cdot \varphi_2 + \dots + \theta_n \cdot \varphi_n \quad (2.36)$$

Onde $\varphi_1, \varphi_2, \dots, \varphi_n$ são funções conhecidas e $\theta_1, \theta_2, \dots, \theta_n$ são os parâmetros conhecidos. Pares de observações $(x_i, y_i), i = 1, 2, \dots, N$ são obtidos de experimentos. O problema consiste em determinar os parâmetros de tal forma que as variáveis \hat{y} calculados a partir do modelo de (2.36) e os valores experimentais x_i sejam tão próximos quanto possíveis das variáveis medidas y_i .

Assumindo que todas as medições tem a mesma precisão, o método dos mínimos quadrados nos diz que os parâmetros devem ser escolhidos de tal forma que o erro seja pequeno, de acordo com uma função de custo (ou critério de desempenho). Definindo a função custo mínimos quadrados:

$$J(\theta) = \frac{1}{2} \cdot \sum_{t=1}^N \varepsilon^2 \quad (2.37)$$

onde $\varepsilon = y(t) - \hat{y}(t) = y(t) - \phi(t) \cdot \theta$

Para simplificar os cálculos, os seguintes vetores são definidos:

$$\varphi = \begin{bmatrix} \varphi_1 & \varphi_2 & \dots & \varphi_N \end{bmatrix}^T \quad (2.38)$$

$$\theta = \begin{bmatrix} \theta_1 & \theta_2 & \dots & \theta_N \end{bmatrix}^T \quad (2.39)$$

$$y = \begin{bmatrix} y_1 & y_2 & \dots & y_N \end{bmatrix}^T \quad (2.40)$$

$$\varepsilon = \begin{bmatrix} \varepsilon_1 & \varepsilon_2 & \dots & \varepsilon_N \end{bmatrix}^T \quad (2.41)$$

$$\phi = \begin{bmatrix} \varphi^T(1) \\ \varphi^T(2) \\ \vdots \\ \varphi^T(N) \end{bmatrix} \quad (2.42)$$

Assim, a função de custo pode ser reescrita como:

$$J(\theta) = \frac{1}{2} \cdot \varepsilon^T \cdot \varepsilon = \frac{1}{2} \cdot \|\varepsilon\|^2 \quad (2.43)$$

onde $\varepsilon = y - \hat{y}e\hat{y} = \phi \cdot \theta$

Podemos encontrar o valor mínimo derivando a equação (2.43) e igualando a zero.

Logo,

$$\begin{aligned} \frac{dJ}{d\theta} &= \frac{1}{2} \cdot \frac{d\|\varepsilon\|^2}{d\theta} = \frac{1}{2} \cdot 2 \cdot \phi^T \cdot [y - \phi \cdot \theta] = 0 \\ \phi^T \cdot y &= \phi^T \cdot \phi \cdot \theta \end{aligned}$$

Portanto, os parâmetros θ são dados por:

$$\theta = (\phi^T \cdot \phi)^{-1} \cdot \phi^T \cdot y \quad (2.44)$$

Podemos definir então

$$R_{n \times m} = \sum_{t=1}^N \phi^T(t) \cdot \phi(t) \quad (2.45)$$

$$F_{n \times 1} = \sum_{t=1}^N \phi^T(t) \cdot y(t) \quad (2.46)$$

Assim, a estimativa pode ser calculada como:

$$\theta = R^{-1} \cdot f \quad (2.47)$$

onde assume-se que a inversa de $R_{n \times m}$ existe.

A solução dada pela equação (2.47) pode ser escrita de forma compacta. Note que a solução pode ser agrupada da seguinte maneira:

$$\theta = (\phi^T \cdot \phi)^{-1} \cdot \phi^T \cdot Y \quad (2.48)$$

onde

$$\theta = [\theta(1) \quad \theta(2) \quad \dots \quad \theta(N)]^T \quad (2.49)$$

$$Y = [y(1) \quad y(2) \quad \dots \quad y(N)]^T \quad (2.50)$$

$$\phi = \begin{bmatrix} \varphi^T(1) \\ \varphi^T(2) \\ \vdots \\ \varphi^T(N) \end{bmatrix} \quad (2.51)$$

2.3.2 Resposta ao Degrau

Métodos gráfico-temporais são populares entre operadores na área de controle. Tais métodos se baseiam na observação direta da variável de saída do processo. O principal atrativo destes métodos é o apelo gráfico, que torna intuitivo o procedimento de modelagem do processo.

O método da resposta ao degrau, em particular, ainda é bastante utilizado para obter um modelo para o processo. Em regime permanente, é aplicado um degrau na entrada do processo, e a saída é registrada. O modelo é então obtido a partir da curva de reação do processo, observando alguns parâmetros gráficos. Várias classes de modelos podem ser obtidas com esse método.

Sistemas Não-Oscilatórios

A saída típica de um sistema de característica não oscilatória é apresentada na figura (2.5). Note que o procedimento é realizado em torno de um ponto de operação, de modo que o modelo obtido é válido em uma faixa limitada. Os parâmetros que podem ser obtidos diretamente a partir da curva de reação do processo são:

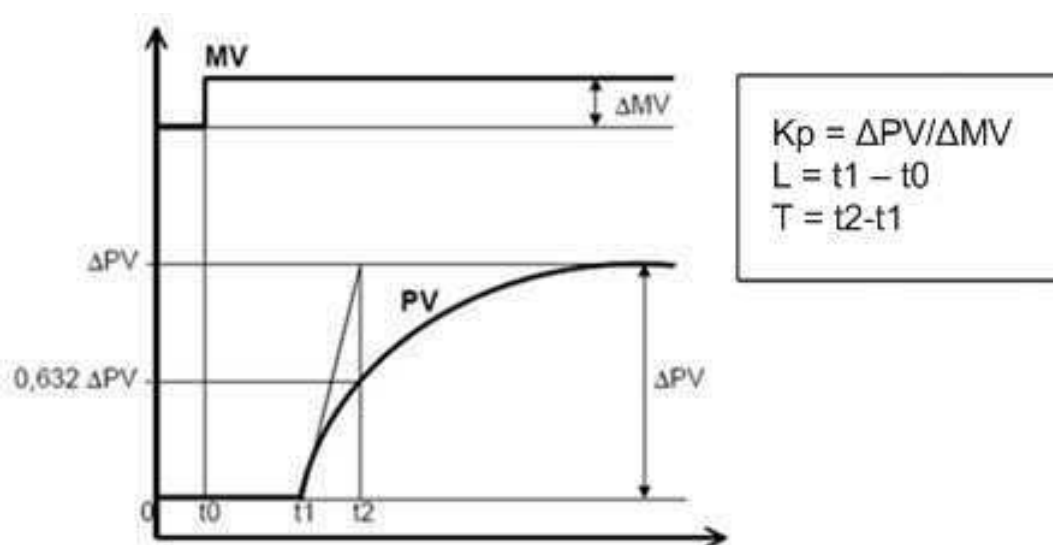


Figura 2.5: Resposta ao degrau para processos de característica não-oscilatória.

em que MV é um degrau, ΔMV é a amplitude do degrau, PV é o sinal medido e ΔPV nos diz o ponto de estabilidade do sistema.

- Ganho, K_p : É obtido da curva de reação pela relação da variação do sinal de saída, δPV , e o sinal de entrada, δMV , i.e., $K_p = \frac{\Delta PV}{\Delta MV}$.
- Atraso de transporte, L : é obtido a partir da reta com maior inclinação possível na curva da figura (2.5). O prolongamento desta reta da forma mostrada leva à estimativa de L .

- Taxa de crescimento, T : É obtida da inclinação da reta usada na determinação do atraso de transporte.
- Constante de tempo de primeira ordem, T_1 : depende do modelo utilizado. Tipicamente, o ponto da curva que atinge 63% de Δy é utilizado na determinação de T_1 , quando um modelo de primeira ordem é utilizado.
- Constante de tempo de segunda ordem, T_2 : assim como T_1 depende do modelo utilizado. Para modelos com mais de uma constante de tempo, outros pontos da curva de reação devem ser utilizados, e, normalmente, soluções por métodos numéricos são necessárias.

Alguns modelos que podem ser obtidos a partir desse procedimento são os seguintes:

$$G(s) = \frac{G_o}{T_1 \cdot s + 1}$$

$$G(s) = \frac{G_o}{(T_1 \cdot s + 1)(T_2 \cdot s + 1)}$$

$$G(s) = \frac{G_o}{T_1 \cdot s + 1} \cdot e^{-Ls}$$

$$G(s) = \frac{G_o}{(T_1 \cdot s + 1)(T_2 \cdot s + 1)} \cdot e^{-Ls}$$

$$G(s) = \frac{G_o}{(T_1 \cdot s + 1)^2} \cdot e^{-Ls}$$

Sistemas Oscilatórios

A saída típica de um sistema de característica oscilatória é apresentada na figura (2.6). Novamente, o procedimento é realizado em torno de um ponto de operação, e o modelo obtido é válido apenas em torno de um ponto de operação. Além dos parâmetros obtidos para o caso de uma resposta não-oscilatória, existem ainda os seguintes parâmetros:

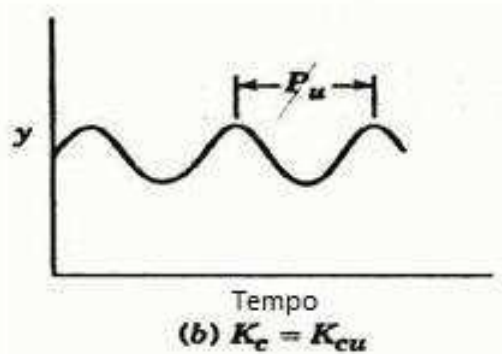


Figura 2.6: Resposta ao degrau pra processos de característica oscilatória

- Período de Oscilação, T_{osc} : Pode ser medido diretamente da curva nos primeiros instantes de oscilação, conforme mostrado na figura (2.6).
- Decaimento do sobressinal, β : É obtido a partir da relação entre os dois primeiros picos da resposta transitória, conforme mostrado na figura (2.6) (representado por P_u na figura).

Os modelos típicos obtidos a partir desse procedimento são:

$$G(s) = \frac{G_o \cdot \omega_n^2}{s^2 + 2 \cdot \zeta \cdot \omega_n s + \omega_n^2}$$

$$G(s) = \frac{G_o \cdot \omega_n^2}{s^2 + 2 \cdot \zeta \cdot \omega_n s + \omega_n^2} \cdot e^{-Ls}$$

2.3.3 Modelagem a partir da Resposta ao Degrau

Suponha que o processo da figura (2.7) está no estado inicial nulo e um degrau de amplitude h é aplicado em $t = 0$ a sua entrada. Ambas, saída e entrada do processo são armazenadas, desde a aplicação do degrau até o tempo em que a saída entra em regime permanente. Suponha que o modelo do processo é descrito pela função de transferência:

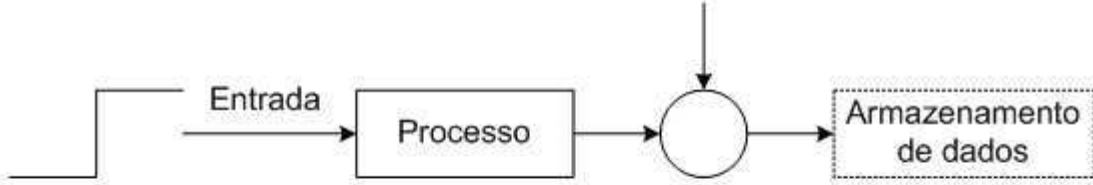


Figura 2.7: Teste de resposta ao degrau

$$G(s) = \frac{Y(s)}{U(s)} \cdot \frac{G_o}{T_1 s + 1} \cdot e^{-Ls}$$

Para essa relação, a saída $y(t)$ para uma entrada em degrau de amplitude h , é dada por:

$$y(t) = hG_o(1 - e^{-\frac{t-L}{T_1}}) + \omega(t), t \geq L$$

onde $\omega(t)$ é o ruído branco presente na medição de $y(t)$. A equação anterior pode ser reescrita como:

$$e^{-\frac{t-L}{T_1}} = 1 - \frac{y(t)}{hG_o} + \frac{\omega(t)}{hG_o}, t \geq L \quad (2.52)$$

Integrando $y(t)$ de $t=0$ até $t= \tau$ com $t \geq L$ ($y(t) = 0$ para $t < L$), tem-se

$$\int_0^\tau y(t)dt = hG_o(\tau - T_1 e^{-\frac{t-L}{T_1}})_{t=L}^{t=\tau} + \int_0^\tau \omega(t)dt$$

Usando a equação (2.52), e o fato de que $y(L) = 0$, tem-se então

$$\int_0^\tau y(t)dt = hG_o(\tau - L - T_1 \frac{y(t)}{hG_o}) + [T_1 \omega(t)]_{t=L}^{t=\tau} + \int_0^\tau \omega(t)dt \quad (2.53)$$

Defina

$$A(\tau) = \int_0^\tau y(t) dt$$

e

$$\delta(\tau) = [T_1 \omega(t)]_{t=L}^{t=\tau} + \int_0^\tau \omega(t) dt$$

Então, a equação (2.53) pode ser reescrita como

$$A(\tau) = hG_o(\tau - L - T_1 \frac{y(t)}{hG_o}) + \delta(t) = \begin{bmatrix} h\tau & -h & y(\tau) \end{bmatrix} \cdot \begin{bmatrix} G_o \\ G_o L \\ T_1 \end{bmatrix} + \delta(t)$$

De modo que

$$A(\tau) = \phi^T(\tau) \cdot \theta + \delta(\tau) \quad (2.54)$$

Que é equivalente ao sistema linear usado na solução de mínimos quadrados da seção anterior com ruído dado pelo termo $\delta(\tau)$. Portanto,

$$\phi(\tau) = \begin{bmatrix} h\tau & -h & y(\tau) \end{bmatrix}^T \quad (2.55)$$

$$\theta = \begin{bmatrix} G_o & G_o L & T_1 \end{bmatrix}^T \quad (2.56)$$

As matrizes ϕ e Y são montadas usando as expressões (2.49) e (2.50) para todas as amostras no tempo τ , com $\tau \geq L$.

O teste só tem validade para processos lineares ou não-lineares linearizados em pontos de operação. Não permite a identificação de modelos de ordem superior, pois o degrau tem pobre composição em frequência.

Método do Relé

Uma forma de determinar o ganho e a frequência crítica de uma planta, K_u e ω_u , utilizando um relé, como o da figura (2.8), é a proposta de [8]. Considere o sistema em malha fechada da figura (2.9) na qual o relé tem amplitude d . Para várias classes de processos, o sistema exibe um ciclo de limite cujas condições podem ser determinadas aproximadamente usando o método de análise por funções descritivas. A oscilação ocorre no ponto em que

$$N(a) = G_p(j\omega) = -1$$

onde

$$N(a) = \frac{4d}{\pi a}$$

É uma função descritiva do relé da figura (2.8-a), que relaciona a amplitude do primeiro harmônico do sinal na entrada do relé, a , com a amplitude do primeiro harmônico do sinal na entrada do relé, $N(a)$. O método da função descritiva tem uma interpretação física interessante. Na figura (2.10-a) são sobrepostas a função descritiva do relé e a curva de Nyquist de um processo $G_p(s)$. Assim, se a é a amplitude da saída do processo, a intersecção das curvas mostradas ocorre no ponto crítico, de modo que,

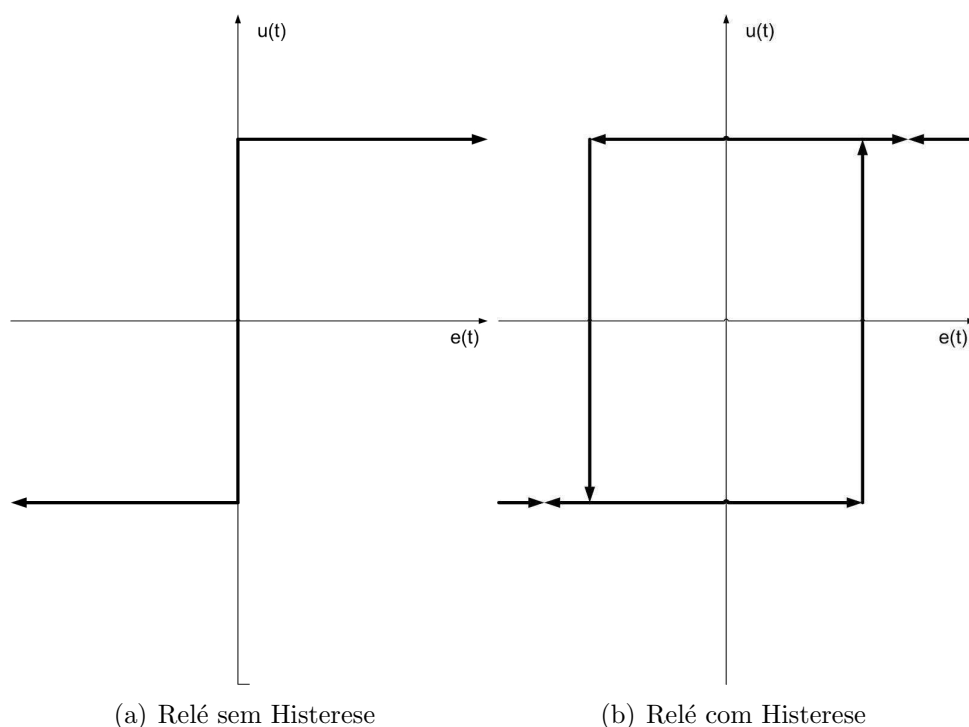


Figura 2.8: Saída $u(t)$ com Relé com entrada $e(t)$

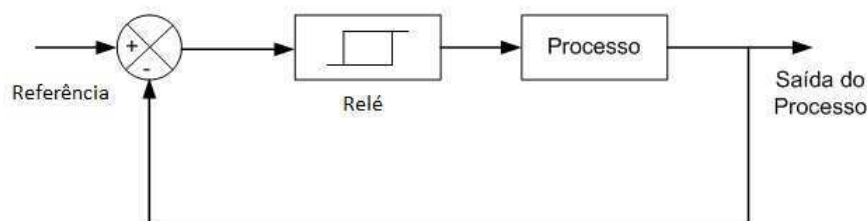


Figura 2.9: Processo com realimentação por relé.

$$G_p(j\omega_u) = \frac{\pi a}{4d}$$

Note que a amplitude na saída do processo pode ser controlada a partir da amplitude d do relé. Lembre-se que os resultados das funções descritivas é um resultado aproximado, uma vez que apenas a primeira harmônica do sinal na saída do relé é considerada.

Levando em conta a presença de ruído no experimento apresentado na figura (2.10), a utilização de um relé puro provoca chaveamento aleatório. Este problema pode ser contornado adicionando uma histerese ao relé puro, como na figura (2.8-b). Para um relé com histerese ϵ , o ponto da resposta em frequência do processo é determinado agora pela intersecção de uma curva de Nyquist com

$$\frac{-1}{N(a)} = -\frac{\pi}{4d} \cdot \sqrt{a^2 - \epsilon^2} - j \frac{\pi \epsilon}{4d} \quad (2.57)$$

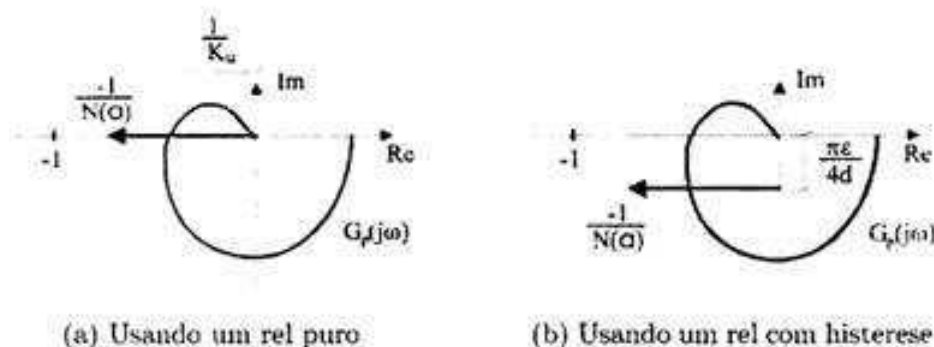


Figura 2.10: Curva de Nyquist de um processo G_p de função descritiva de um relé $N(a)$.

Que são retas paralelas ao semi-eixo real negativo. Da figura (2.9-a). Utilizando o relé com histerese no experimento da figura (2.10-b). Ainda com $a(s) = 1$, obtém-se valores aproximados para K_u e ω_u , desde que a histerese, ϵ , seja pequena.

2.4 Linearização e Escalonamento

As equações diferenciais de movimentação para quase todos os processos selecionados para controle são não lineares, a análise e o design de controle são bem mais fáceis para sistemas lineares, entretanto. Linearização é então o processo de encontrar um modelo linear que aproxima um modelo não-linear ([1]).

2.4.1 Linearização de Pequenos Sinais

Uma equação diferencial não-linear não pode ser escrita da forma:

$$\dot{x} = Fx + Gu \quad (2.58)$$

Ao invés disso, ela deve ser deixada na forma:

$$\dot{x} = f(x, u) \quad (2.59)$$

Para linearização de pequenos sinais devemos determinar os pontos de equilíbrio x_0, u_0 , ou seja, os pontos onde $\dot{x}_0 = 0 = f(x_0, u_0)$. Então expandimos a equação não-linear em termos de perturbações desses pontos de equilíbrio, ou seja, consideramos $x = x_0 + \delta x$ e $u = u_0 + \delta u$, temos então:

$$\dot{x}_0 + \delta \dot{x} \cong f(x_0, u_0) + F\delta x + G\delta u \quad (2.60)$$

onde F e G são os melhores termos lineares para a função não-linear $f(x, u)$ em x_0 e u_0 , ou seja:

$$F = \left[\frac{\delta f}{\delta x} \right]_{x_0, u_0} \quad (2.61)$$

$$G = \left[\frac{\delta f}{\delta u} \right]_{x_0, u_0} \quad (2.62)$$

Subtraindo a solução de equilíbrio, reduzimos a equação para:

$$\delta \dot{x} = F\delta x + G\delta u \quad (2.63)$$

que é a equação linear diferencial que aproxima a dinâmica do sistema próximo do ponto de equilíbrio. Normalmente, a notação δ é descartada e se subentende que x e u se referem a um desvio do ponto de equilíbrio.

2.5 Considerações

Finalizado o capítulo, é possível concluir e mostrar que pode-se implementar um sistema de controle completo em malha fechada através do método de espaços de estados quando se tem em mente o comportamento desejado para o sistema através de seu lugar de raízes. Além disso, também pode-se concluir que, quando não se tem informações suficiente sobre uma planta, pode-se utilizar métodos de identificação de sistemas para encontrar os parâmetros desejados e então implementar um sistema de controle de acordo com o comportamento desejado para a planta.

Em geral, foi possível fornecer um embasamento teórico relevante para os conceitos implementados nos guias de laboratórios para ambos os kits didáticos de Levitador Magnético e Módulo Servomotor de Precisão da *Feedback Instruments* (que serão abordados no capítulo 3), os conceitos de alocação de polos serão utilizados para a implementação de um projeto de controle completo no Módulo Servomotor de Precisão, o qual também será utilizado para experimentos de identificação usando o método dos mínimos quadrados, em contrapartida o experimento que aborda o Levitador Magnético se aproveita principalmente dos conceitos de linearização e discretização também apresentados nesse capítulo.

Capítulo 3

Plataformas Didáticas para Realização dos Experimentos

3.1 Introdução

Nesse capítulo são apresentadas as duas plataformas didáticas utilizadas na elaboração e implementação dos guias de experimentos (que serão apresentados no capítulo 4). O capítulo é dividido em três seções, a seção 3.2 mostra o material utilizado e necessário para utilização das plataformas, em seguida cada uma é apresentada individualmente em suas devidas seções.

Na seção 3.3 é apresentado o kit didático do Servomotor de Precisão da *Feedback Instruments* em sua visão generalista. As sub-seções 3.3.1 e 3.3.2 dissecam então os aspectos de hardware da plataforma, explorando cada componente individualmente, tanto da unidade mecânica quanto da unidade elétrica, respectivamente. Em seguida, 3.3.3 dá início a uma introdução do que acontece em termos de software na plataforma, a sub-seção 3.3.4 aborda o Motor DC, especificando seus parâmetros e seu comportamento. Em seguida alguns aspectos de controle são abordados em 3.3.5 e 3.3.6, aspectos esses que serão utilizados quando implementados em guias de experimentos no capítulo 4.

Por fim, na seção 3.4 é apresentado o kit didático do Levitador Magnético da *Feedback Instruments* em sua visão generalista. A sub-seção 3.4.1 revela então os aspectos de hardware da plataforma enquanto as sub-seções seguintes (3.4.2 e 3.4.3) os aspectos de software. Na sub-seção 3.4.4 é apresentado o modelo fenomenológico do Levitador, modelo este que é não-linear, sua linearização é então apresentada em 3.4.5. Por fim as sub-seções seguintes revelam aspectos gerais sobre identificação e controle do Levitador Magnético.

Todas as informações presentes nesse capítulo foram tiradas de [5], [4], [6].

3.2 Material Utilizado

Os seguintes equipamentos e manuais são requeridos para o uso das plataformas didáticas Servomotor de Precisão e Levitador Magnético:

1. Computador com Windows 2000 ou Windows XP.

2. Matlab V7.3 (R2006b) ou mais novo com Simulink
3. Os seguintes toolboxes do Matlab:
 - Real Time Workshop com Real Time Windows Target.
 - Toolbox de Identificação de sistemas (Necessário para realizar exercícios de identificação)
 - Toolbox de Controle (Necessário para realizar a síntese de controladores)
4. Cartão Advantech PCI1711
5. CDs de instalação do Software
6. Manual Installation and Commissioning (um diferente para cada plataforma)
7. Manual Control Experiments (um diferente para cada plataforma)
8. Manual Matlab Guide

3.3 Plataforma Didática - Servomotor de Precisão

A bancada do Módulo Servomotor, como mostrada na figura (3.1) serve como um kit didático de um Servomotor DC, um dispositivo de grande popularidade. A palavra servo vem do latim *servus* que significa escravo. O servomotor destina-se a reagir a um determinado comando, por exemplo, uma posição ou velocidade desejada. Para que o dispositivo possa ser chamado de servomotor é necessário que ele seja equipado com uma unidade de medição de velocidade e posição e um acionamento.

As principais características que distinguem os servomotores dos demais motores são:

- Inércia do rotor minimizada;
- Indutância do motor minimizada;
- Suporte de tensão e magnetização melhorados;
- Projetado para suportar acelerações repentinas e operar em altas frequências.

O layout da bancada de experimentos do Módulo Servomotor é mostrado na figura (3.2). Podemos dividi-la em duas unidades: mecânica e elétrica. Os dispositivos que compõem cada uma das unidades são descritos a seguir.

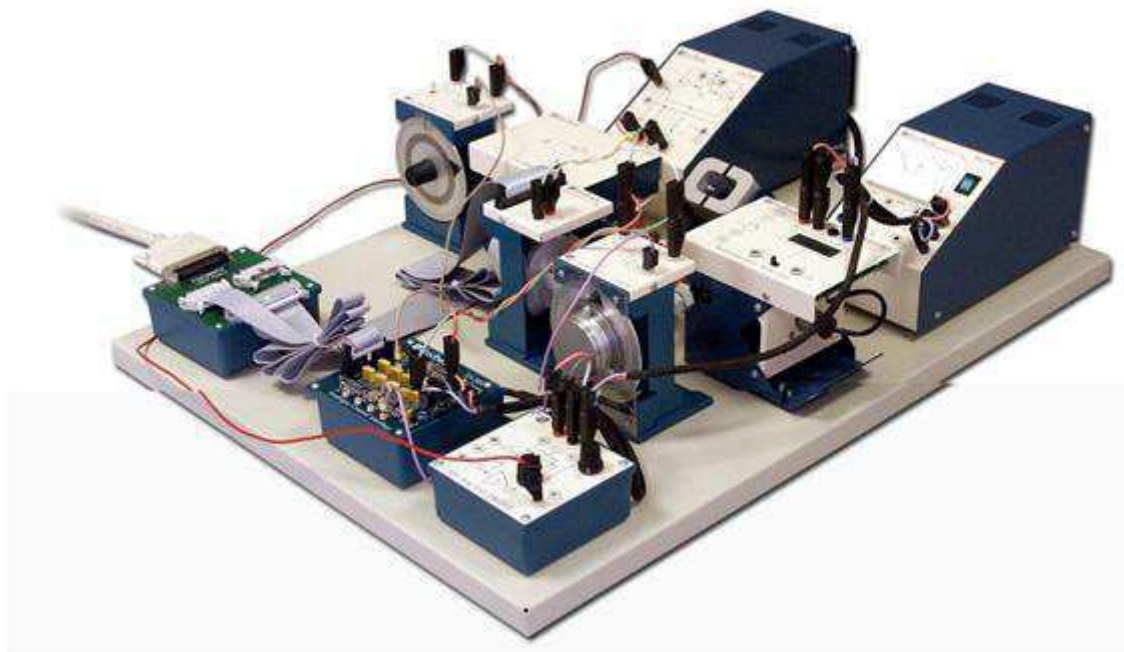


Figura 3.1: Visão geral de um módulo servomotor

3.3.1 Unidade Mecânica

A Unidade Mecânica do módulo servomotor, mostrada na figura (3.3), é constituída dos seguintes componentes: Motor DC, Freio Magnético, Gearbox/Tacômetro, Potenciômetro de Saída e o *Encoder* Digital.

O motor DC (DCM150F) é controlado através do Amplificador Servo e é a fonte primária de controle da tensão. No eixo do motor DC temos um disco que é usado junto com o freio magnético para simular uma carga usando o efeito de correntes parasitas. O Amplificador Servo compõe a Unidade Elétrica e será detalhado mais adiante.

O freio magnético (LU150L) é um ímã que pode ser aumentado ou diminuído de modo a simular uma carga menor ou maior no eixo do motor DC e logo sua posição em relação ao motor DC afeta a dinâmica do sistema.

A unidade GT150X tem duas funções: *gearbox* ou tacômetro. Como *gearbox* ele fornece uma diferente taxa de rotações para o Potenciômetro de Saída (OP150K). Como tacômetro ele fornece um sinal de tensão analógico que é proporcional a velocidade do eixo do motor DC. Este sinal de tensão analógica que será proporcional à velocidade do motor DC, o qual pode então ser usado para controle de velocidade.

O potenciômetro de saída (OP150K) fornece um sinal analógico de tensão proporcional a posição do eixo do *gearbox*. Este sinal pode ser usado para controle de posição do eixo do *gearbox* ou do eixo do motor DC e o mesmo pode ser usado como Display Visual da posição.

Um sinal digital de posição do motor DC é fornecido pelo *encoder* digital (33-300). Ele entrega um sinal digital pulsante para a placa de E/S do PC. Seus pulsos podem ser contados fornecendo uma medição precisa correspondente à posição do eixo do motor DC. A taxa de variação angular fornece o sinal de velocidade angular. Esta unidade é

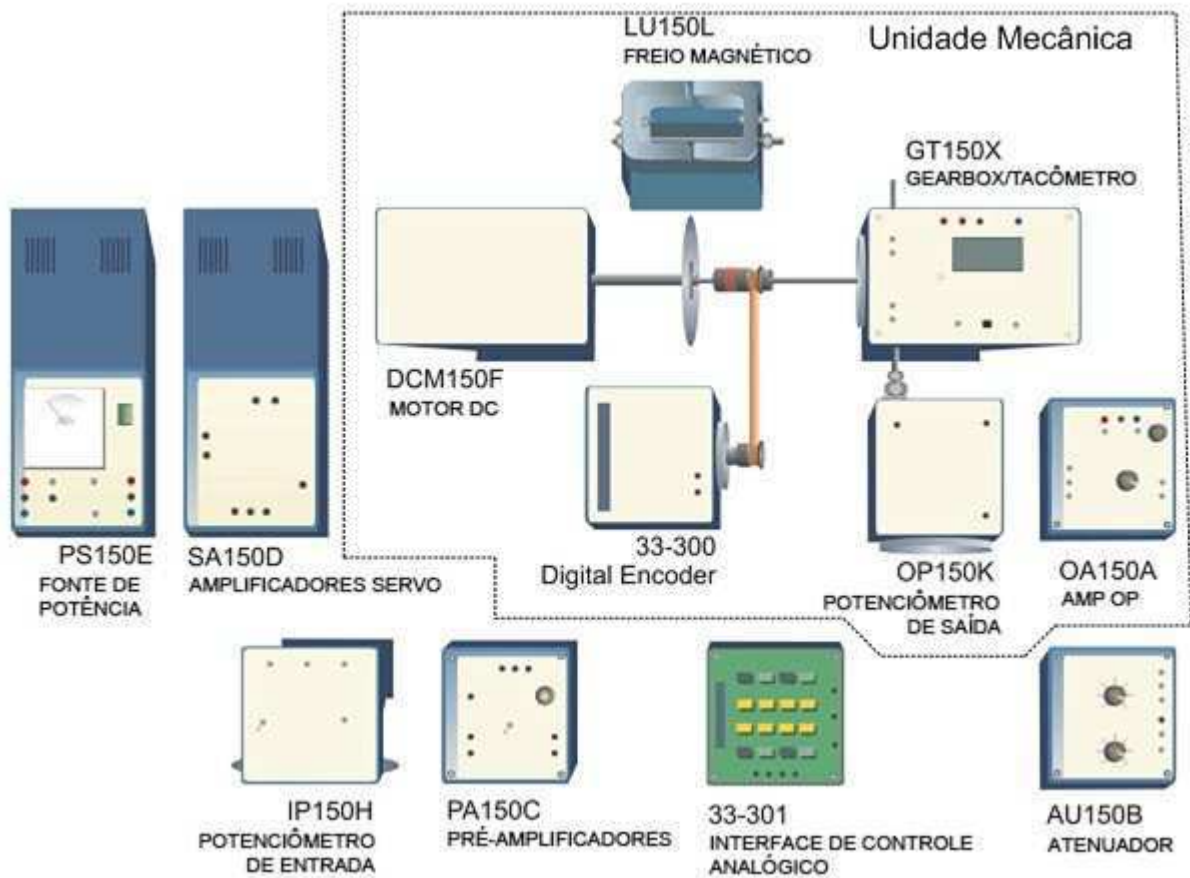


Figura 3.2: Layout da bancada do módulo servomotor

conectada ao motor por meio de uma correia de transmissão e duas rodas dentadas. E logo ele fornece uma medida precisa do ângulo correspondente à posição do motor.

3.3.2 Unidade Elétrica

A unidade elétrica, mostrada na figura (3.4), realiza o condicionamento dos sinais permitindo que estes sejam transferidos para o computador através do cartão E/S. Os amplificadores (Amplificadores Servo e Pré-Amplificadores) são usados para transferir os sinais de controle, que foram gerados pelos algoritmos de controle, do computador para o motor DC. Os componentes dessa unidade são: fonte de potência, amplificador servo, pré-amplificador, unidade de interface, atenuador e potenciômetro de entrada.

Os componentes que correspondem à unidade Elétrica são listados abaixo:

- A fonte de potência (PS150E) alimenta todo o módulo com tensões entre -15V e +15V.
- O amplificador servo (SA150D) fornece a potência controlada para o motor DC, controle este que é realizado por um cabo especial.

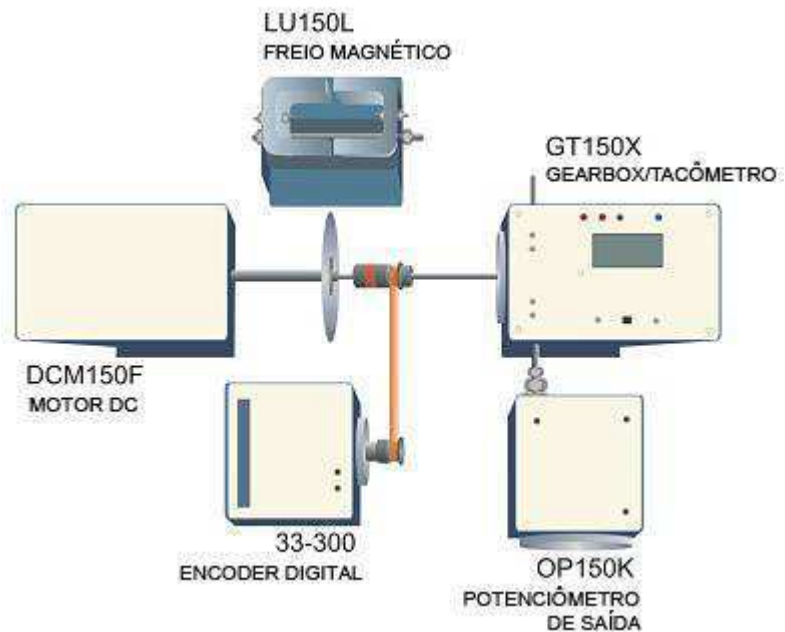


Figura 3.3: Unidade Mecânica do Módulo Servomotor

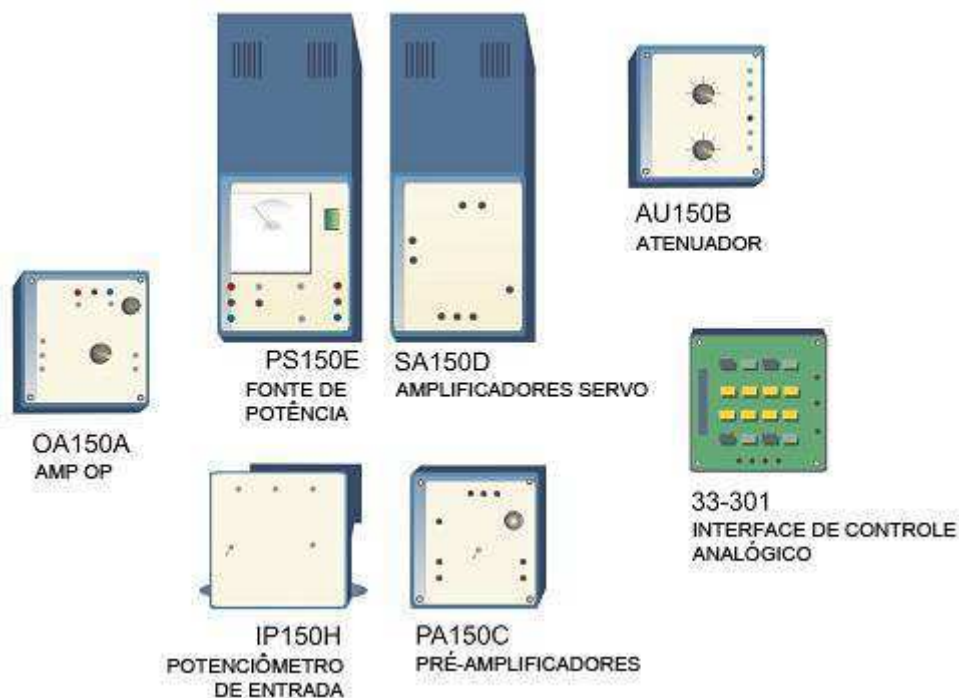


Figura 3.4: Unidade Elétrica do Módulo Servomotor

- Pré-amplificador (PA150C) e o atenuador (AU150B) realizam o condicionamento de sinais da bancada. O Pré-Amplificador fornece o sinal no intervalo correto para o Servo Amplificador enquanto o atenuador é usado para definir a entrada do circuito

de controle do motor no nível correto de operação.

- Com o potenciômetro de entrada (IP150H) podemos estabelecer uma referência quando optarmos por um controle de posição manual. Ele fornece ainda uma tensão analógica proporcional à sua posição angular.
- Na unidade de interface (33-301) é realizado o dimensionamento das entradas e saídas analógicas da placa de E/S no computador para o intervalo de $-2.5V$ a $2.5V$. Ela possui vários potenciômetros de OFFSET e de GANHO/ATENUAÇÃO responsáveis pela calibração do módulo.

3.3.3 Sistema de Controle

Na bancada de experimentos servomotor temos um sistema de controle que é organizado e caracterizado como mostrado na figura (3.5). Um computador com o cartão de E/S da Advantech e com os ambientes Matlab e Simulink servem como unidade principal de controle.

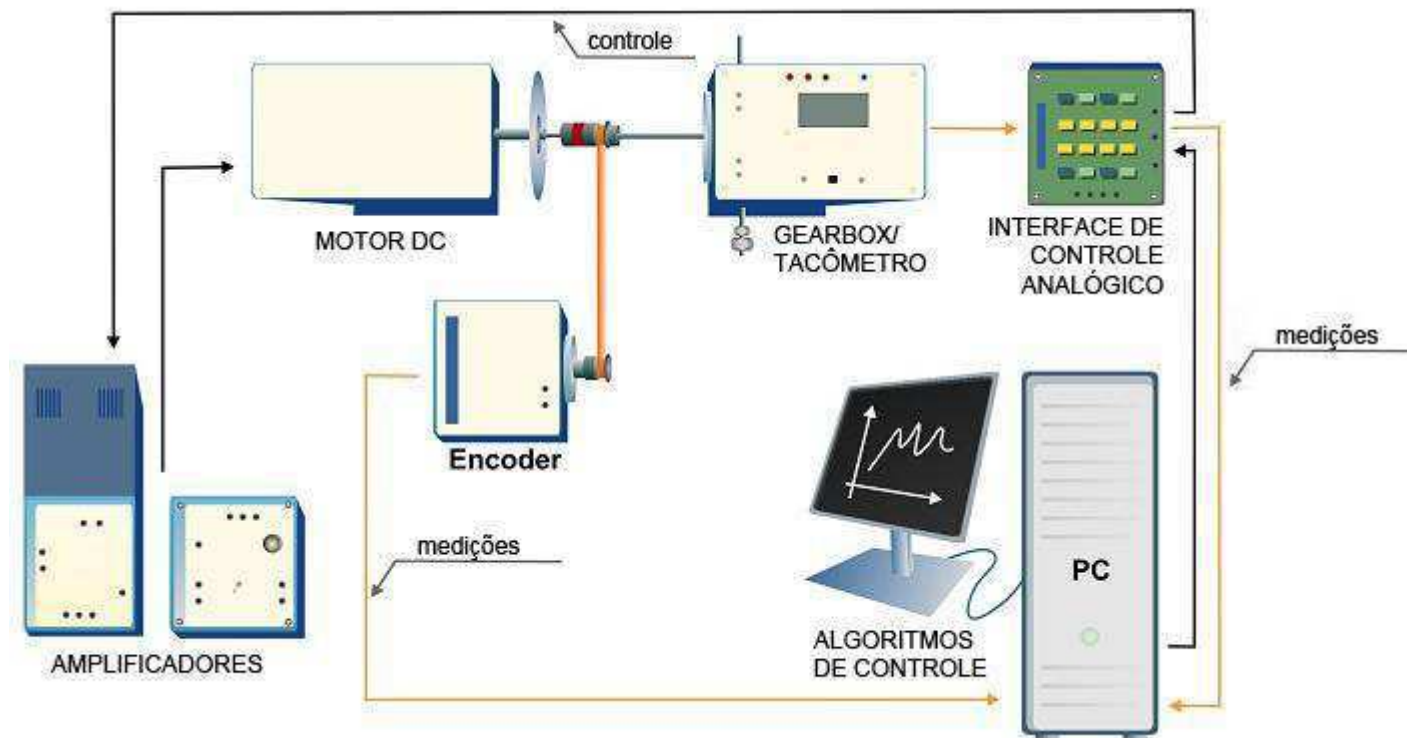


Figura 3.5: Sistema de controle do Módulo Servomotor

Os sinais de controle são sinais de tensão no intervalo de $-2.5V$ a $2.5V$ e são gerados pelos algoritmos de controle presentes no computador. Esses sinais são direcionados para uma série de amplificadores os quais são responsáveis pelo acionamento do motor DC.

A posição é medida por meio do *encoder* digital ou potenciômetro de saída e a velocidade pode ser medida por meio do tacômetro. Estas medições são transferidas

para o computador, onde todos os algoritmos de controle são localizados, através de uma unidade de interface analógica.

3.3.4 Modelo do Motor DC

Normalmente, modelos fenomenológicos são não-lineares, o que significa que ao menos uma das variáveis de estado (i - corrente, θ - posição) é argumento de uma função não linear. Para representarmos o modelo como uma função de transferência este modelo deve ser linearizado. No motor DC presente no Módulo Servomotor estas não-linearidades são pequenas, podendo ser desprezadas. Na figura (3.6) temos o modelo elétrico-mecânico do Motor DC.

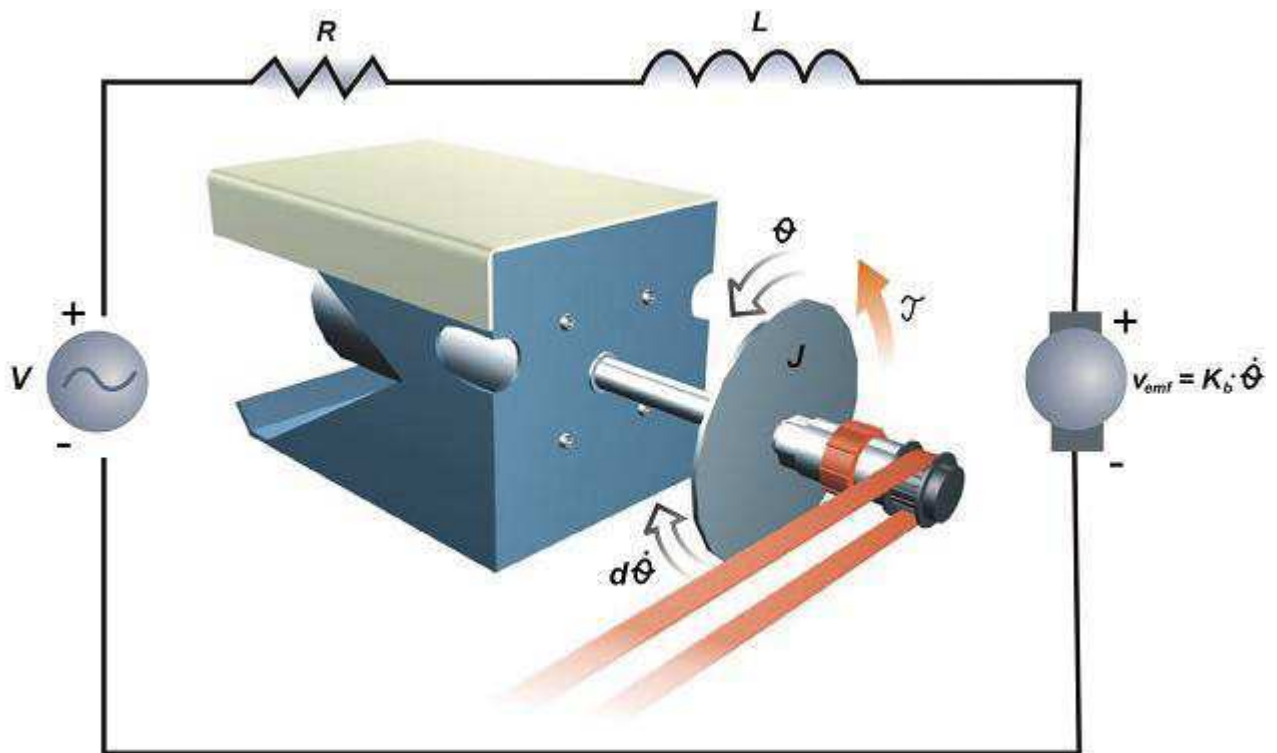


Figura 3.6: Modelo elétrico-mecânico do motor DC

O torque gerado pelo motor DC é proporcional a corrente do motor (i):

$$\tau = K_t \cdot i \quad (3.1)$$

onde K_t é a constante de torque.

A força eletromotriz induzida, v_{emf} , é uma tensão proporcional á velocidade angular do eixo:

$$v_{emf} = K_b \cdot \dot{\theta} = K_b \cdot \omega \quad (3.2)$$

onde K_b é uma constante que depende das propriedades físicas do motor e $\dot{\theta} = \omega$ é a velocidade angular. Usando a lei de Newton para o modelo mecânico do motor, obtemos:

$$J \cdot \ddot{\theta} = \sum \tau_i = d \cdot \dot{\theta} + K_t \cdot i \quad (3.3)$$

onde d é uma aproximação linear da fricção e J é o momento de inércia. A equação que governa o comportamento elétrico do motor é:

$$u = -L \frac{di}{dt} + R \cdot i + v_{emf} = -L \cdot \frac{di}{dt} + R \cdot i + K_b \cdot \dot{\theta} \quad (3.4)$$

A partir das equações (3.3) e (3.4) podemos construir o espaço de estados:

$$\frac{d}{dt} \begin{bmatrix} \dot{\theta} \\ i \end{bmatrix} = \begin{bmatrix} \frac{d}{J} & \frac{K_t}{J} \\ -\frac{K_b}{L} & -\frac{R}{L} \end{bmatrix} \cdot \begin{bmatrix} \dot{\theta} \\ i \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} \cdot u \quad (3.5)$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \dot{\theta} \\ i \end{bmatrix} \quad (3.6)$$

O motor usado é um motor DC de 24V com velocidade em vazio de 4050 rpm. A tabela (3.1) contém os valores de todos os parâmetros do motor.

Parâmetro	Valor
J - momento de inércia	$140 \cdot 10^{-7} \text{ kg} \cdot \text{m}^2$
K_t - constante de torque	$0,052 \frac{\text{N} \cdot \text{m}}{\text{A}}$
K_b - constante de força eletromotriz	$0,057 \frac{\text{V} \cdot \text{s}}{\text{rad}}$
d - aproximação linear da viscosidade de fricção	$10^{-8} \frac{\text{N} \cdot \text{m} \cdot \text{s}}{\text{rad}}$
R - resistência	$2,5\omega$
L - Indutância	$2,5m \cdot H$

Tabela 3.1: Parâmetros do Motor DC

Exercício Ilustrativo - Teste do modelo

O modelo descrito na equação 3.5 é fornecido pela *Feedback Instruments* e pode ser aberto no simulink pelo nome de *DCmotor_ManModel.mdl*. O modelo do motor DC básico responde diferente quando utilizado dentro do PMS (*Precision Modular Servo*), isso ocorre graças á dinâmica dos componentes mecânicos e elétricos adicionais que não são incluídos no modelo básico. Entretanto, tais diferenças podem ser facilmente detectadas no processo de identificação. O modelo fenomenológico pode ser alterado adicionando um ganho ou um certo parâmetro de sintonização, isso é mostrado no arquivo *DCmotor_TunModel.mdl*, onde uma amplificação é adicionada ao modelo e logo o modelo PMS é criado e controlado com um sinal de amplitude 2.5 [V], assim como seria em uma planta real.

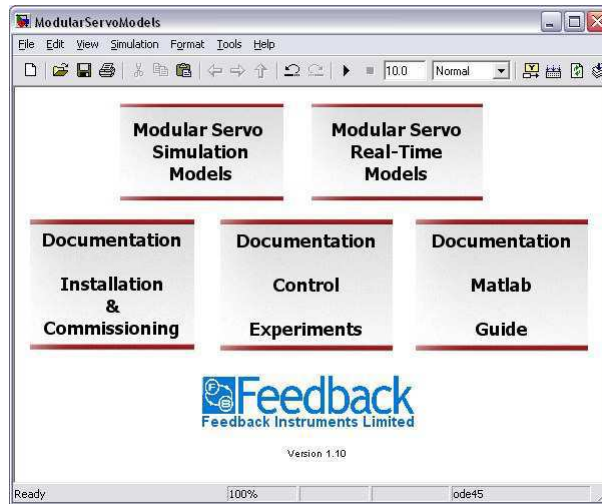
O primeiro passo é abrir o Matlab pelo menu do sistema operacional utilizando o atalho instalado junto ao *toolkit*, o "Simulink Models", como mostrado na figura (3.7).

O Matlab irá ser executado e o menu de modelos do *Simulink* irá abrir, como mostrado na figura (3.8(a)). Ele leva aos menus de modelos limitados a simulação (*Modular Servo Simulation Models*), ao menu dos modelos em tempo real (*Modular Servo*

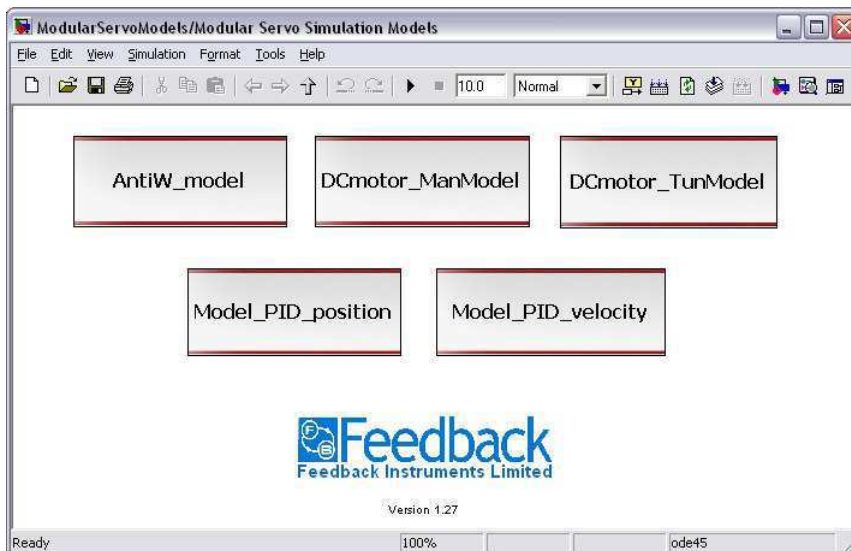


Figura 3.7: Menu para inicialização do Servo Motor

Real-Time Models) e aos documentos dos manuais. Escolhendo os modelos limitados a simulação, teremos o menu da figura (3.8(b)). Escolhe-se então com um duplo-clique o modelo *DCmotor_ManModel.mdl*. A janela do simulink com o modelo irá abrir. Alternativamente, o arquivo pode ser aberto de modo manual pelo matlab, seguindo o caminho: *C* → *Feedback* → *ModularServo* → *Examples* → *Simulation_models* → *DCmotor_ManModel.mdl*.



(a) Menu de modelos do *Simulink*



(b) Menu com modelos de simulação

Figura 3.8: Interface entre o Matlab e os arquivos do Servo Motor

De início se faz interessante testar as respostas do modelo do servo motor com as tensões zero, constante e senoidal aplicadas. Uma tensão de 24V irá fazer o motor girar em velocidade máxima (aproximadamente 4.000rpm). É possível mudar os valores dos parâmetros para verificar como a resposta varia, como por exemplo o momento de inércia J (clique duplo no bloco do motor para mudar seus parâmetros).

A figura 3.9 mostra a resposta da tensão a um degrau para o modelo sintonizado com carga mínima.

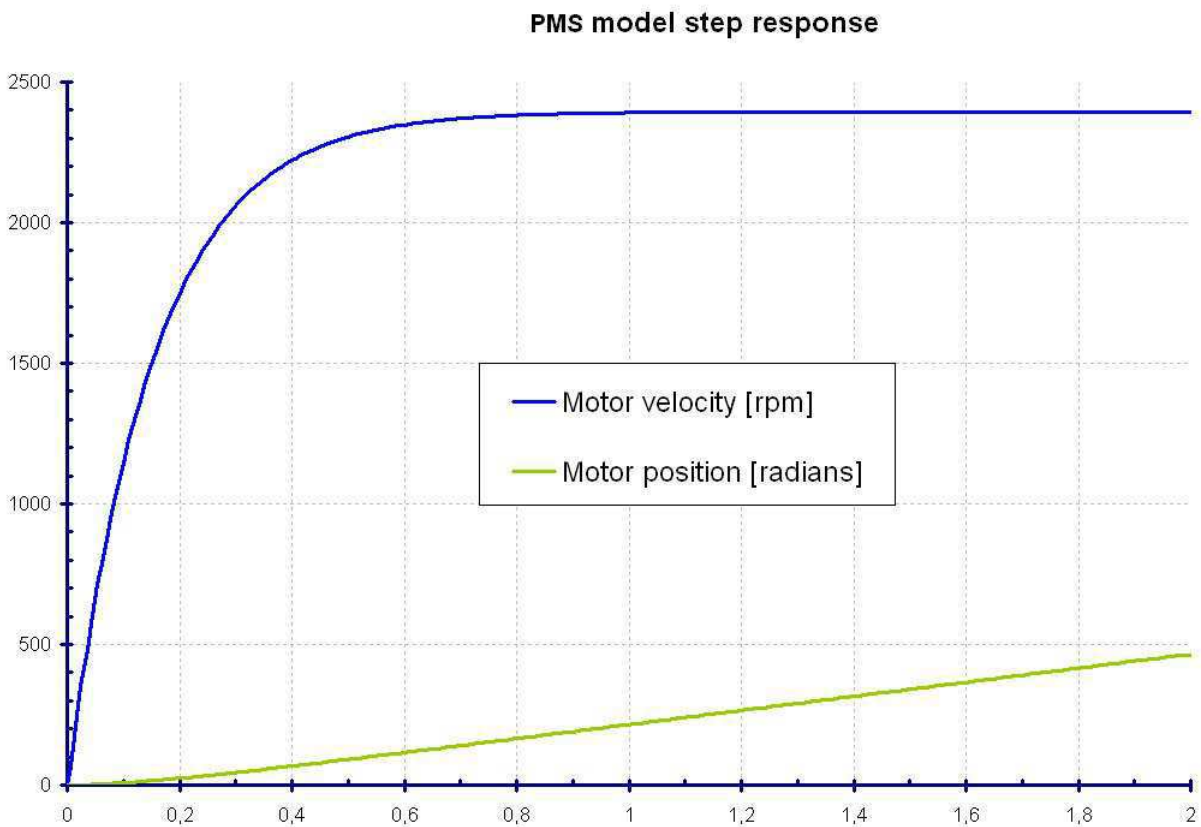


Figura 3.9: Exemplo de resposta ao modelo do servo motor sintonizado (carga mínima)

3.3.5 Controle por computador

Devido a cooperação da Advantech e da Marthworks, o projeto de um sistema de controle para o Módulo Servomotor foi simplificado de forma significativa, de modo que não torna-se necessário construir a aplicação em tempo real desde o início. Usando o kernel do Real-Time, é possível projetar diversos tipos de controladores e executá-los em um processo externo.

O sistema de controle por computador, como mostrado na figura (3.10) consiste em quatro elementos: um computador, conversores D/A e A/D e o processo em si em conjunto com os sensores. No computador encontra-se o algoritmo de controle síncrono que gera o sinal de controle digital. Este sinal é convertido para a forma analógica pelo conversor D/A e é aplicado ao processo. As medições realizadas pelos sensores são transformadas

em sinais digitais pelo conversor A/D e utilizadas pelo algoritmo de controle síncrono presente no computador.

Basicamente o sinal de controle é transferido para uma série de amplificadores, que gerencia o Motor DC. O *Encoder Digital* fornece o parâmetro da posição enquanto o Tacômetro fornece a velocidade e a unidade de interface transfere os dados para o computador.

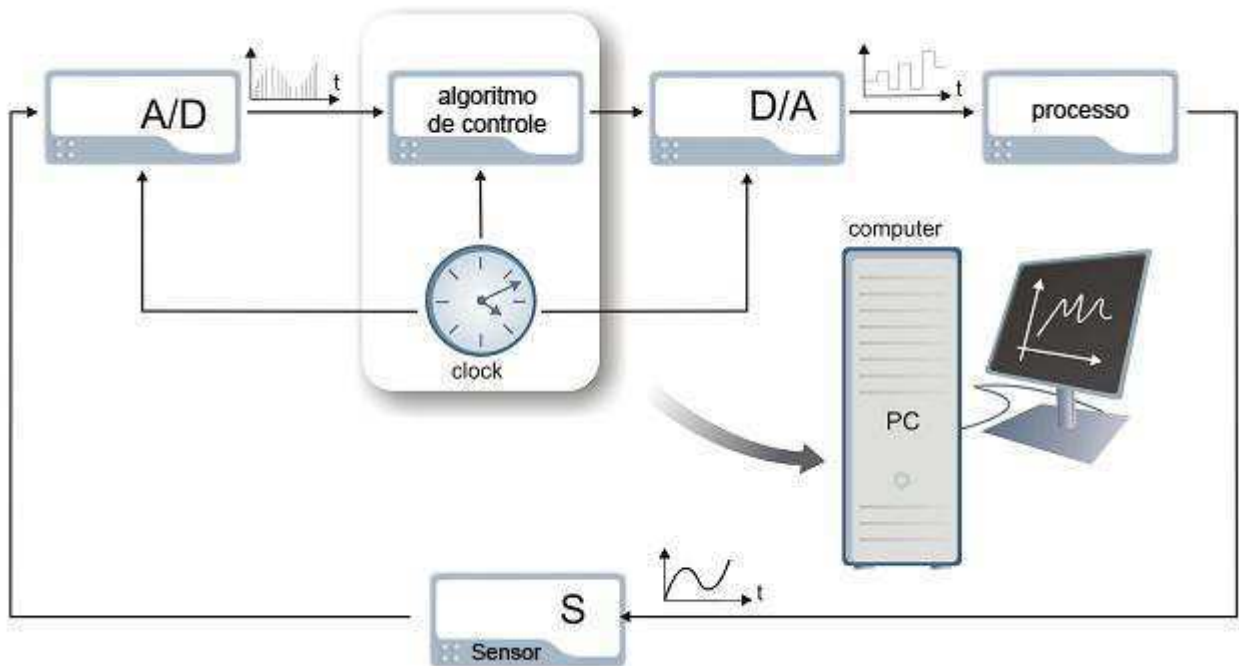


Figura 3.10: Diagrama do sistema de controle por computador

3.3.6 Projeto de Controladores

No projeto de controladores para o Módulo Servomotor faz-se necessária a utilização de três blocos introduzidos pela *Feedback Instruments* para a interação com o ambiente *Matlab*, são eles: *Feedback Encoder*, *Feedback DAC* e *Analog Input*, os quais são mostrados na figura (3.11). Estes blocos fornecem uma interface entre o computador e alguns dispositivos do módulo.

O *Feedback Encoder* tem apenas uma saída que corresponde à posição do eixo do motor em radianos. O bloco *Analog Input* tem uma saída e dependendo do canal escolhido na unidade de interface (33-301), estará lendo um sinal de tensão proporcional à posição ou à velocidade do dispositivo analógico apropriado (potenciômetro de saída ou tacômetro). Por fim o *Feedback DAC* é o responsável pela realimentação do sistema analógico, enviando o sinal de controle depois que as operações com o sinal recebido tenha sido trabalhado.

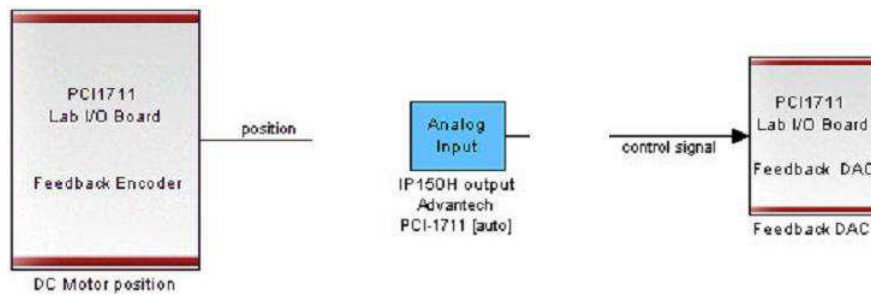


Figura 3.11: Blocos de equipamentos externos no projeto do Servomotor usando o Matlab

Se faz necessário relatar que a própria *Feedback Instruments* recomenda que os parâmetros dos blocos não sejam modificados a menos que se faça necessário, por essa razão os blocos não serão aprofundados além de suas funcionalidades básicas.

Exercício Ilustrativo - Identificação de um Modelo

O modelo a ser apresentado descreve a relação entre a tensão de controle u e a velocidade angular do motor DC, $\omega = \dot{\theta}$. Todas as simulações de controle em tempo real são efetivadas com um tempo de amostragem $T_S = 0.001$ [s] e portanto o mesmo tempo de amostragem é utilizado em experimentos de identificação para fornecer total compatibilidade. Para a identificação, o toolbox *System Identification* do Matlab é utilizado.

O experimento é realizado através do modelo contido no arquivo *VelocityModel_Ident.mdl*, modelo este que excita o sistema do motor e registra suas respostas (figura 3.12). O sinal de excitação consiste em várias senoides (figura 3.13). O experimento dura 20 segundos e dois sinais são coletados na forma de vetores e se tornam disponíveis na área de trabalho do Matlab.

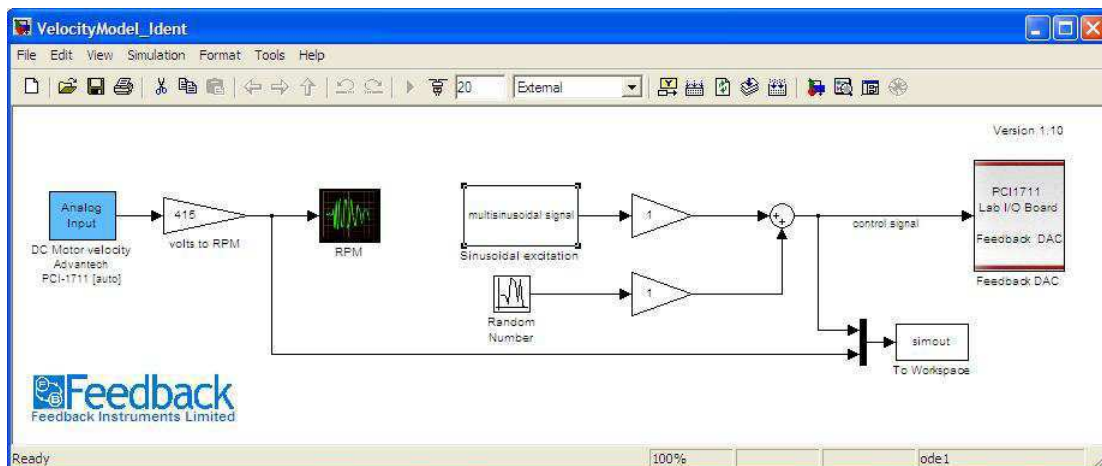


Figura 3.12: Modelo de identificação

Antes de executar o programa, ajuste o pré-amplificador PA150C de modo que o motor não esteja girando. Pode-se então prosseguir com o experimento de identificação e coletar os dados. Em seguida identificar o modelo utilizando a interface do matlab (escreva o comando "Ident").

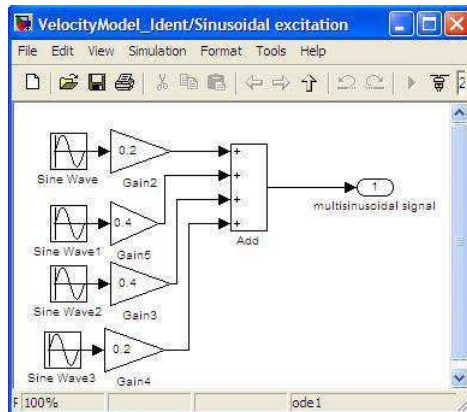


Figura 3.13: Sinal de Excitação utilizado na modelagem

Uma resposta ao degrau de um sistema identificado é similar á mostrada na figura 3.14, ela pode variar de acordo com as propriedades do sistema real, tais como os atrasos em gearbox, tacômetro, potenciômetro de saída, etc. O modelo pode ser comparado com o seu equivalente discreto utilizando o comando "c2d", especificado o correto tempo de amostragem.

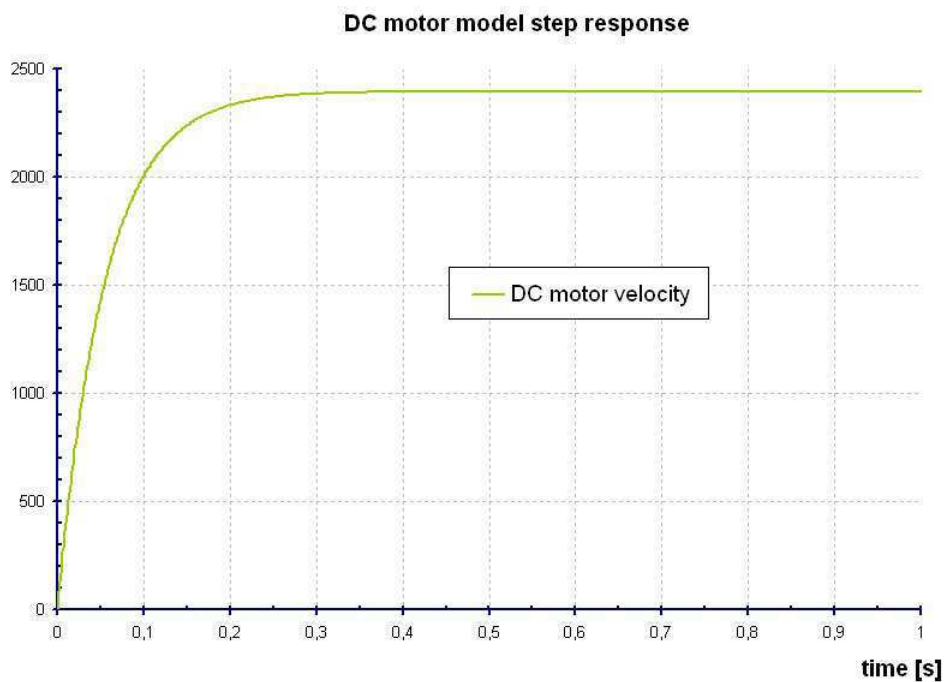


Figura 3.14: Resposta ao degrau de um sistema identificado

3.4 Plataforma Didática - Levitador Magnético

A palavra levitação tem origem no latim *levis*, que significa leveza, e é o processo com o qual se consegue suspender um objeto numa posição estável contrariando, assim as forças de gravidade, mediante o uso de forças exercidas sem contato com o objeto. Manter um corpo suspenso no ar sem qualquer apoio aparente é reconhecido como fenômeno de levitação.

A levitação estável de alguns materiais comuns se baseia em uma propriedade que todos os materiais possuem, chamada de diamagnetismo. Ao tentar aproximar os polos iguais de dois ímãs, estes se repelem, o polo positivo do campo externo repele os polos positivos de cada átomo magnetizado do material, quando os campos são contrários, essa força de repulsão gerada faz com que o material possa levitar quando a mesma for maior que o peso do material.

Existem três tipos de levitação magnética: na levitação eletrodinâmica ou por repulsão magnética se utiliza bobinas com baixa resistência elétrica para geração de dois campos magnéticos cuja interação gera uma repulsão capaz de suspender um objeto. Na levitação eletromagnética ou por atração magnética, o caso do kit didático em questão, um eletroímã é responsável por manter suspenso um corpo ferromagnético. Por fim há a levitação supercondutora, que utiliza materiais magnéticos e pastilhas supercondutoras que operam em altas temperaturas. Todas as informações presentes nessa seção foram retiradas de [2] e de [3].

3.4.1 Descrição da Unidade Magnética

A unidade mecânica (como mostrada na figura 3.15) do levitador magnético consiste em uma base, que constitui a interface de conexão. Nessa base a unidade mecânica é colocada com a bobina estando montada no topo da construção. Um sensor Infra-Vermelho é colocado nos dois lados da construção.

A interação com o sinal de controle é feita através da Unidade Analógica de Controle, que se resume a buffers e amplificadores como mostrado no Apêndice.

3.4.2 Sistema de Controle

O sistema Levitador Magnético é organizado de acordo com o esquemático apresentado na figura 3.16.

O computador com o cartão Advantech, o Matlab e o ambiente simulink servem como a principal unidade de controle. Os sinais de controle, que são voltagens entre -5 V e 5 V, são transferidos para a unidade, o que causa fluxo de corrente através da bobina e logo a geração de um campo magnético.

A posição da bola é medida pelo sensor infravermelho. A informação da posição é transferida para o computador pela unidade de interface, onde todos os algoritmos de controle são colocados no ambiente do Simulink. Do ponto de vista do Simulink, o Levitador Magnético é visto como o modelo da figura 3.17.

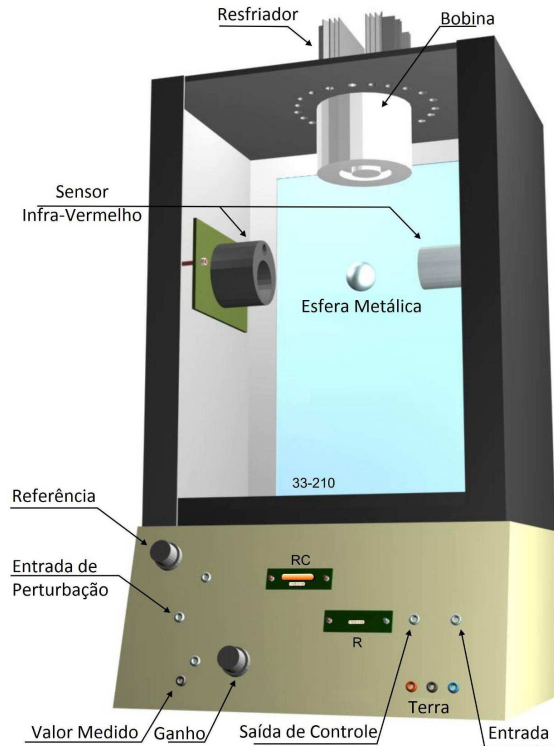


Figura 3.15: Unidade Mecânica do Levitador Magnético

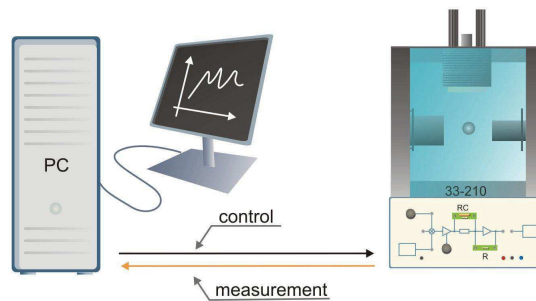


Figura 3.16: Esquemático do Sistema de Controle



Figura 3.17: Modelo Maglev

3.4.3 Controle do Computador

Usando o MathWorks e o software do Advantech juntos permite a elaboração de sistemas de controle para o sistema Levitador Magnético ser simplificado. Para construir um controlador não se faz necessário criar uma aplicação em Tempo Real desde o começo. Ela já está feita. Pode-se designar controladores arbitrários no ambiente do simulink e

através do Kernel em tempo real implementá-las no processo externo. Entretanto para aqueles não familiares com o sistema de controle computacional alguma informação básica é dada.

O diagrama do controle digital é apresentado na figura 3.18 e usado para tal propósito.

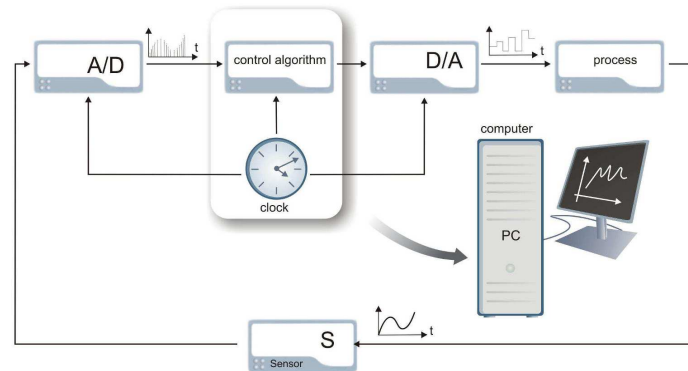


Figura 3.18: Diagrama do sistema de controle computacional

O sistema consiste em quatro elementos primários:

1. Computador com algoritmo de controle sincronizado pelo relógio(*clock*);
2. Conversores A/D e D/A, que servem como interface entre o computador e o ambiente externo.
3. O sistema controlado;
4. Sensores.

O algoritmo e os conversores operam de acordo com os pulsos de tempo distribuídos pelo Clock. A distância entre pulsos de tempo consecutivos é chamada Tempo de Amostragem. Usualmente o tempo de amostragem é constante embora algumas aplicações especiais usem amostragem intermitente. O clock é responsável pela interrupção que gera a Rotina de Serviço de Interrupção (ISR). Durante a rotina ISR, um conversor A/D entrega uma representação discreta da medida retornada pelo sensor. Baseado na medição o algoritmo calcula o valor do sinal de controle. No fim do ISR o valor do sinal de controle é atualizado e setado pelo conversor D/A para ser usado no próximo intervalo de amostragem.

3.4.4 Modelo de Levitação Magnética

Todo projeto de controle tem início pela modelagem da planta, logo a maior quantidade de informação possível é dada sobre o processo em si. O modelo mecânico-elétrico do Levitador Magnético é apresentado na Figura (3.19)

Normalmente, modelos fenomenológicos são não-lineares, isso significa que ao menos um dos estados (i - corrente, x - posição da bola) é um argumento de uma função

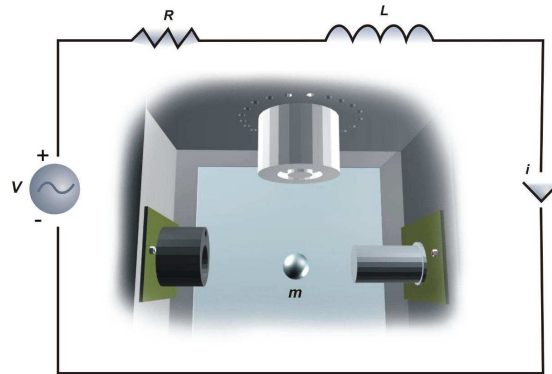


Figura 3.19: Modelo fenomenológico do Levitador Magnético

não-linear. Visando apresentar tal modelo como uma função de transferência (uma forma de representação dinâmica linear da planta usada em engenharia de controle), ela precisa ser linearizada.

De acordo com o diagrama elétrico-mecânico apresentado na figura (3.19) as equações do modelo não-linear podem ser derivadas.

O mais simples modelo não linear de um sistema de levitação magnética que relaciona a posição da bola e a corrente da bobina i é a seguinte:

$$m \cdot \ddot{x} = m \cdot g - k \frac{i^2}{x^2} \quad (3.7)$$

onde k é uma constante que depende dos parâmetros eletromagnéticos da bobina. Para apresentar o modelo fenomenológico total uma relação entre a tensão de controle u e a corrente da bobina deve ser introduzida analisando o circuito completo do Levitador Magnético. Entretanto o Levitador Magnético é equipado com um controle de loop interno que fornece a corrente proporcionado a tensão de controle que é gerada para propósitos de controle:

$$i = k_1 \cdot u \quad (3.8)$$

As equações (3.7) e (3.8) constituem um modelo não-linear, que foi montado no Simulink. Os limites do sinal de controle são estabelecidos em $[-5V \dots 5V]$. O Levitador Magnético é uma planta SISO - *Single Input Single Output* (Figura 3.20)



Figura 3.20: Modelo para controle de posição do Maglev

Seu modelo não-linear será então dado por

$$\ddot{x} = g - f(x, i), f(x, i) = k \frac{i^2}{m \cdot x^2} \quad (3.9)$$

Teste do modelo não-linear

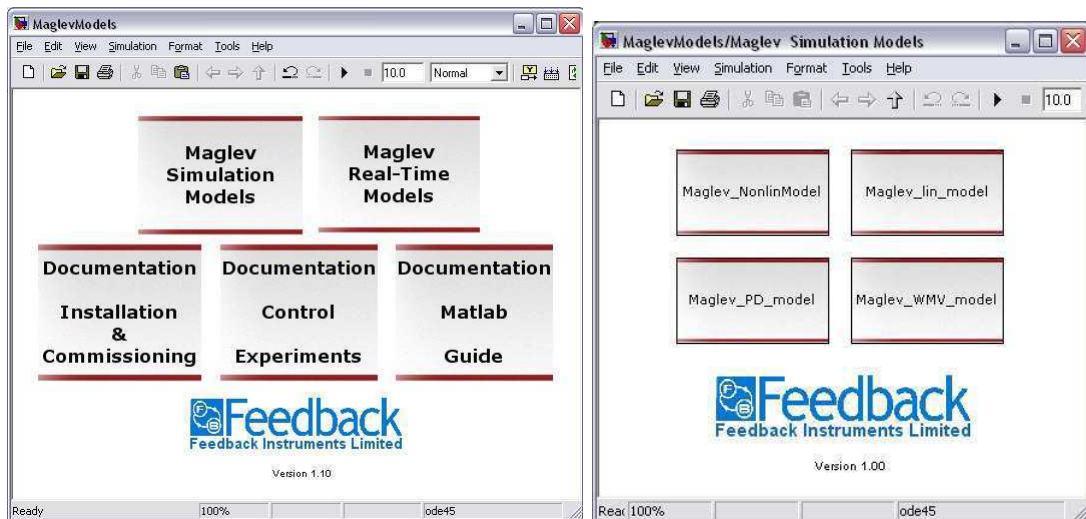
Visando ilustrar o modelo não-linear do levitador magnético descrito pelas equações (3.7) e (3.8). O modelo pode ser aberto no *simulink* - *Maglev_NonlinModel.mdl*. É possível que o modelo fenomenológico demonstrado apresente um comportamento diferente que a unidade magnética, isso pode ser gerado graças às diferenças na estrutura do próprio modelo, assim como graças ao valor dos parâmetros escolhidos, essas diferenças também podem ser facilmente detectadas durante o processo de identificação. O modelo pode ser modificado adicionando um ganho ou alterando algum parâmetro de sintonização.

O primeiro passo é abrir o Matlab pelo menu do sistema operacional utilizando o atalho instalado junto ao *toolkit*, o "Simulink Models", como mostrado na figura (3.21).



Figura 3.21: Menu para inicialização

O Matlab irá ser executado e o menu de modelos do textitSimulink irá abrir, como mostrado na figura (3.22(a)). Ele leva aos menus de modelos limitados a simulação (*Maglev Simulation Models*), ao menu dos modelos em tempo real (*Maglev Servo Real-Time Models*) e aos documentos dos manuais. Escolhendo os modelos limitados a simulação, teremos o menu da figura (3.22(b)). Escolhe-se então com um duplo-clique o modelo *Maglev_NonlinModel.mdl*. A janela do simulink com o modelo irá abrir. Alternativamente, o arquivo pode ser aberto de modo manual pelo matlab, seguindo o caminho: $C \rightarrow Feedback \rightarrow Maglev \rightarrow Examples \rightarrow Simulation_models \rightarrow Maglev_NonlinModel.mdl$.



(a) Menu de modelos do *Simulink*

(b) Menu com modelos de simulação

Figura 3.22: Interface entre o Matlab e os arquivos do Levitador Magnético

De início se faz interessante testar as respostas do modelo do levitador magnético. Sem nenhum controle aplicado, a bola irá continuar caindo, então é possível dizer de modo

intuitivo o que irá acontecer com a resposta do modelo se $u=0$. É interessante realizar o teste com outros valores no sinal de controle modificando o valor no seletor de entrada. Um exemplo é mostrado na figura (3.23), que mostra a resposta do modelo do levitador magnético quando nenhuma tensão de controle é aplicada.

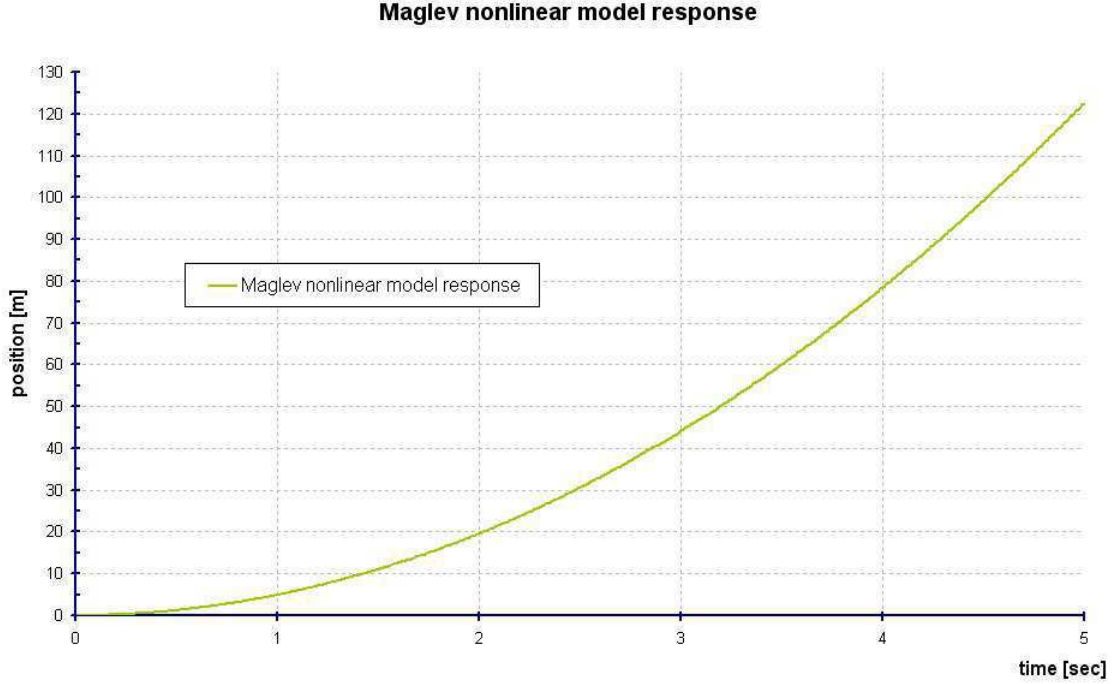


Figura 3.23: Exemplo de resposta ao modelo não-linear do Levitador Magnético (nenhuma tensão de controle aplicada)

3.4.5 Linearização do Modelo

Para dar procedimento as análises da dinâmica do modelo em malha aberta usando técnicas como diagramas de Bode, mapas de polos e zeros, diagramas de Nyquist, lugar das raízes (para sistemas de malha fechada apenas), o modelo deve ser linearizado. Tal linearização é feita no ponto de equilíbrio de $x_0 = -1.5$ [V] (A posição é expressa em volts), $I_0 = 0.8$ [A]:

$$\ddot{x} = g - f(x, i), f(x, i) = k \frac{i^2}{m \cdot x^2} \quad (3.10)$$

O ponto de equilíbrio pode ser calculado a partir de:

$$g = f(x, i) \Rightarrow i_0, x_0 \quad (3.11)$$

Os passos de linearização são os seguintes:

$$\ddot{x} = - \left(\frac{\partial f}{\partial i} \Big|_{i_0, x_0} \Delta i + \frac{\partial f}{\partial x} \Big|_{i_0, x_0} \Delta x \right)$$

$$s^2 \Delta x = -(K_i \Delta i + K_x \Delta x)$$

$$\begin{aligned} s^2 \Delta x + K_x \Delta x &= -K_i \Delta i \\ \Delta x (s^2 + K_x) &= -K_i \Delta i \end{aligned}$$

$$\frac{\Delta x}{\Delta i} = \frac{-K_i}{s^2 + K_x} \quad (3.12)$$

onde $K_i = \frac{2g}{i_0}$ e $K_x = \frac{-2g}{x_0}$

Exemplo Ilustrativo de um Modelo Linear

Um modelo linear que segue a equação (4.7) foi implementado no Simulink. Os passos para acessar o mesmo são similares aos demonstrados na figura (3.22), que também pode ser acessada diretamente pelo matlab em $C \rightarrow \text{Feedback} \rightarrow \text{Maglev} \rightarrow \text{Examples} \rightarrow \text{Simulation_models} \rightarrow \text{Maglev_lin_model.mdl}$. Também é possível realizar os mesmos testes do exemplo da sessão 3.4.4, executar o arquivo e comparar as respostas. Com o auxílio do matlab, também é possível ver diagramas de Bode e mapas de polos e zeros podem ser implementados para analisar a resposta dinâmica inicial do sistema do Levitador Magnético. A figura 3.24 mostra a resposta do modelo do Levitador Magnético quando nenhuma tensão de controle é aplicada.

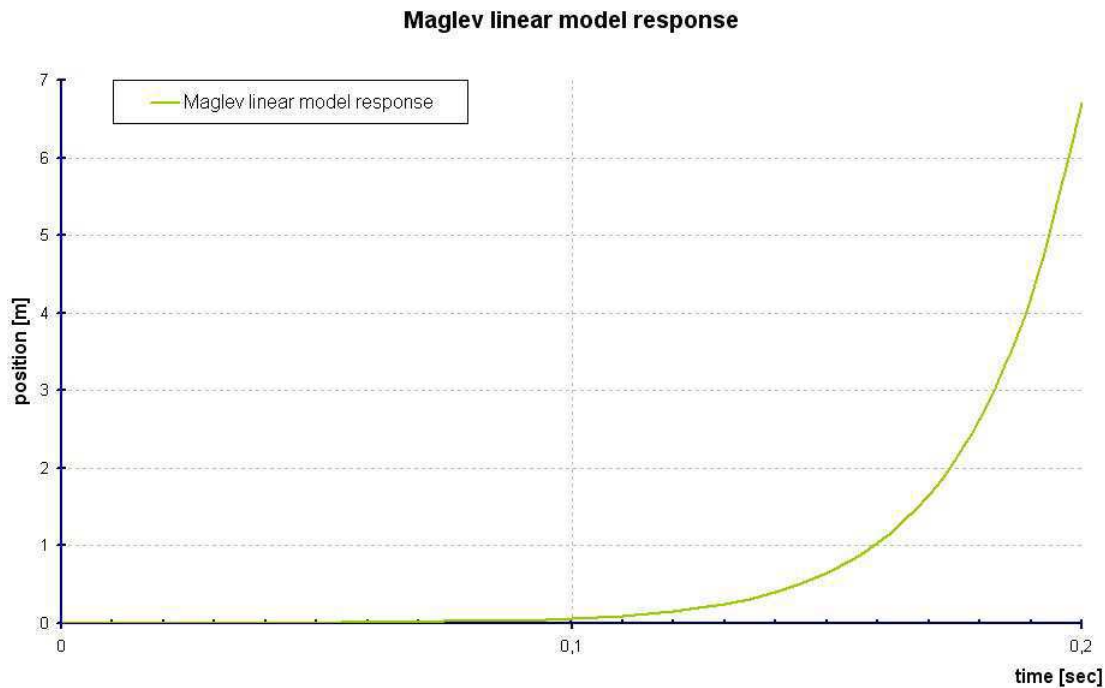


Figura 3.24: Exemplo de resposta ao modelo linear do Levitador Magnético (nenhuma tensão de controle aplicada)

3.4.6 Identificação de Modelos no Levitador Magnético

Na sessão anterior um modelo fenomenológico foi derivado e então linearizado. Um modelo pode ser identificado através de um experimento de identificação, no qual os

controladores serão designados e que é descrito na sessão 2.3. Essa sessão explica como a identificação do modelo discreto do levitador magnético é realizada.

Identificação do Modelo

O modelo a ser identificado descreve a relação entre a posição da bola expressa em volts e a corrente da turbina expressa em amperes. Quaisquer tarefas de identificação são difíceis por razão da instabilidade. Para realizar um experimento de identificação se faz necessária a inserção de uma malha de controle, caso contrário a bola irá cair e a dinâmica do sistema não poderá ser identificada. O melhor cenário se dá com o uso de um controlador discreto para propósitos de identificação de modelos discretos, essa é a razão de uma versão discreta do PD ser utilizada nesse experimento.

Como dito anteriormente, o levitador magnético é instável e deve ser identificado com um controlador responsável por estabilizá-lo, ou seja, se trata de uma identificação em malha fechada. Infelizmente, o controlador introduz uma saída de ruído e uma correlação com o sinal de controle, o que leva à corrupção do modelo, essa correlação pode ser evitada introduzindo um sinal extra de excitação r , que é adicionado ao sinal de controle u (Figura 3.25)

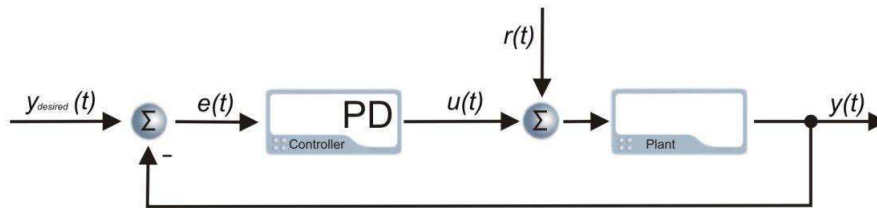


Figura 3.25: Identificação de sistemas instáveis

Se a potência do sinal r é substancial quando comparada com a potência do ruído n , o sistema deve ser identificável. Tal abordagem só irá permitir uma identificação linear do modelo da função de transferência entre a tensão do sinal de controle u e a posição x expressa em volts para pequenas variações da posição no ponto de equilíbrio de $X_0 = -1.5$ [V].

Sistemas de identificação mais inteligentes devem ser gerenciados para um modelo de identificação do levitador magnético completo (métodos de gradiente). Um sistema em malha fechada descrevendo a relação entre o sinal r e y é identificado como:

$$T(z^{-1}) = \frac{G(z^{-1})}{1 + C(z^{-1}) \cdot G(z^{-1})} \quad (3.13)$$

onde $T(z^{-1})$ é o sistema completo da função de transferência discreta e descreve a relação entre $C(z^{-1})$ (função de transferência do controlador discreto) e $G(z^{-1})$ (função de transferência discreta do modelo levitador magnético). A relação (3.13) pode ser transformada para conter o modelo levitador magnético:

$$G(z^{-1}) = \frac{T(z^{-1})}{1 - C(z^{-1}) \cdot T(z^{-1})} \quad (3.14)$$

Exemplo: Modelo de Identificação

A maior parte das simulações em tempo real são implementadas com um tempo de amostragem $T_S = 0.001[s]$. Nesse exemplo, entretanto, é utilizado $T_S = 0.01[s]$ de modo que o modelo possa ser utilizado para síntese de controladores avançados. O experimento de identificação é realizado utilizando o modelo *MaglevIdent.mdl*. Esse modelo controla a levitação da esfera e adiciona um sinal de excitação adicional para a saída de controle, a resposta é então gravada. O sinal de excitação é um sinal aleatório, o experimento dura 30 segundos e dois sinais são adquiridos na forma de vetores e logo disponíveis na área de trabalho (*Workspace*) do matlab.

O primeiro passo é executar o modelo no simulink. A esfera já deve estar preparada de modo manual em sua posição de equilíbrio, a identificação tem início após 10 segundos, o sensor infra-vermelho não deve receber cobertura da mão. Se a esfera cair durante a identificação, faz-se necessário um ajuste no valor "signal gain" e recomeçar o experimento.

Um dos possíveis sistemas para modelos de malha fechada é o *oe331*. Quando a identificação da função de transferência $T(z^{-1})$ for realizada, com a função *mextract* teremos uma equação similar á mostrada em (3.13). O modelo também é transformado em um modo contínuo. O modelo contínuo pode ser utilizado na implementação do controlador PID (que será mostrado mais á frente). Sua versão discreta é utilizada para síntese de controladores avançados.

O modelo do levitador magnético é instável e difícil em realizar uma comparação com sua resposta ao degrau. Graças a isso, o sistema em malha fechada é comparado com um modelo de dados experimentais (*ModelTest.mdl*). O experimento utiliza um controlador PD para identificação na unidade do levitador. Em paralelo a isso o mesmo controlador é utilizado para o controle do modelo. Durante o experimento a posição da esfera é mudada em degraus. É possível comparar o sistema em malha fechada do levitador magnético e a execução do modelo identificado, como mostrado na figura 3.26. Para testar qualquer outro modelo, basta substituí-lo pelo *ModelTest.mdl*.

3.4.7 Implementação de controle do Maglev

O aspecto de controle do levitador magnético cobre uma área, que é o controle da posição. O Matlab fornece vários métodos de análise para sistemas lineares onde a dinâmica é o foco (lugar de raízes, análise de frequência - diagramas de Bode, gráficos de Nyquist, mapas de zeros e polos, etc). Com a informação que o matlab fornece sobre a dinâmica do sistema, controladores podem ser desenvolvidos. As subseções a seguir explicam como o controlador PID funciona, e como ele pode ser aprimorado.

3.5 Considerações

Ao término do capítulo, foi possível introduzir todos os conceitos básicos e necessários para familiarização com os kits didáticos da *Feedback Instruments* Servomotor de Precisão e Levitador Magnético, fez-se também possível inter-relacionar até certo grau os kits com o conteúdo apresentado no capítulo 2.1 e expor alguns exemplos práticos utilizando as bancadas. Exemplos mais sofisticados serão apresentados no capítulo 4,

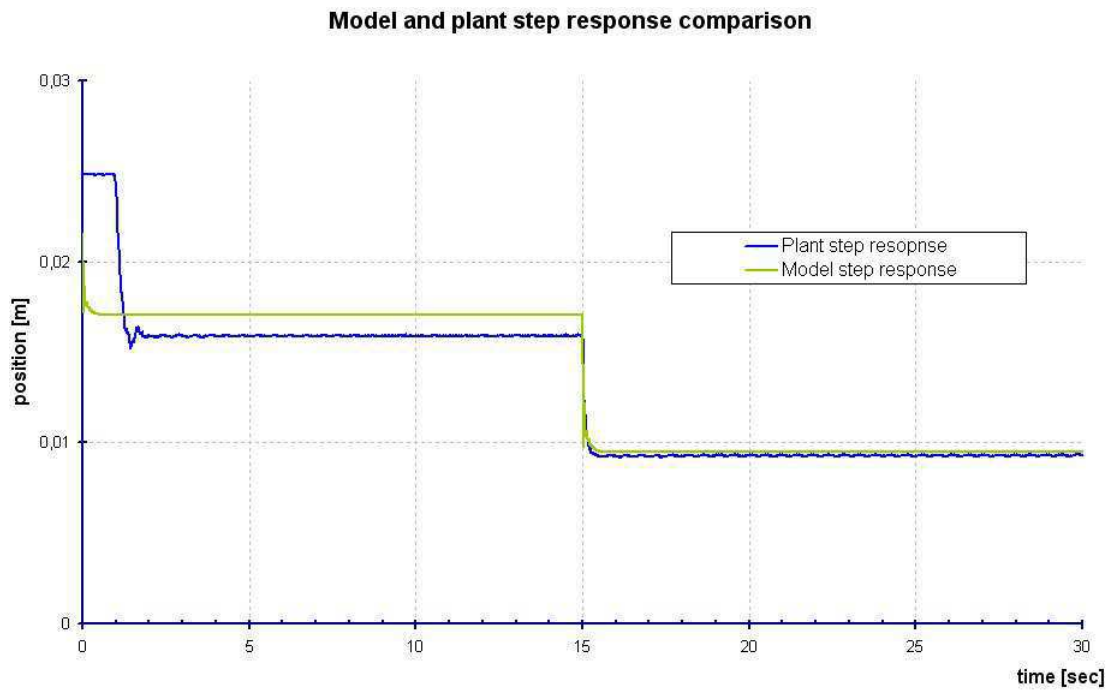


Figura 3.26: Exemplo de Identificação: respostas do modelo e da planta em malha fechada

através dos experimentos propostos e guias de laboratório para a disciplina Laboratório de Controle Digital.

Capítulo 4

Experimentos Propostos

4.1 Introdução

O trabalho proposto teve como resultado a elaboração de guias de experimentos que utilizam as bancadas Servomotor de Precisão e Levitador Magnético, ambos da *Feedback Instruments*. Durante a elaboração dos guias de experimentos, atividades práticas foram feitas para comprovar resultados satisfatórios dos métodos estudados.

Para o kit didático do Servo Motor de Precisão foram implementados dois experimentos, um de alocação de polos utilizando o espaço de estados e outro para identificação de sistemas. Para o Levitador Magnético foi abordado discretização e o método do relé.

4.2 Experimento 1 - Servo Motor de Precisão - Alocação de polos utilizando o Espaço de Estados

Nesse experimento é estudado um método de projeto chamado alocação de polos do ponto de vista de realimentação de estados. O aluno irá revisar os conceitos de realimentação linear de estados e de observadores e o método de projeto de sistema de controle completo, lecionados na disciplina Controle Digital.

Nas práticas preparatórias e experimentais o aluno utilizará os conceitos presentes no guia para o desenvolvimento do sistema de controle para a bancada do Servomotor DC.

4.2.1 Atividade de Preparação

1. Discretize o modelo do Módulo Servomotor utilizando o espaço de estado dado no guia deste.

$$\frac{d}{dt} \begin{bmatrix} \dot{\theta} \\ i \end{bmatrix} = \begin{bmatrix} \frac{d}{J} & \frac{K_t}{J} \\ -\frac{K_b}{L} & -\frac{R}{L} \end{bmatrix} \cdot \begin{bmatrix} \dot{\theta} \\ i \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} \cdot u$$
$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \dot{\theta} \\ i \end{bmatrix}$$

2. Verifique se o sistema discreto achado no item anterior é controlável e/ou observável.
3. Calcule as matrizes ganho de realimentação de estados, L, e ganho do observador, K, para que o polinômio característico do sistema em malha fechada tenha os seguintes parâmetros:
 - **Tempo de Subida**, $t_r = 100$ ms
 - **Tempo de pico**, $t_p = 1500$ ms
 - **Tempo de Acomodação**, $t_s = 0.2$ s

Dicas: Para sistemas de segunda ordem, temos que:

$$t_r = \frac{\pi - \beta}{\omega_d}$$

e

$$\beta = \operatorname{tg}^{-1} \frac{\omega_d}{\sigma}$$

onde $\sigma = \zeta \cdot \omega_n$ e $\omega_d = \omega_n \sqrt{1 - \zeta^2}$ além disso, temos que $t_p = \frac{\pi}{\omega_d}$ e que para $t \geq 0$, $t_s = \frac{4}{\sigma}$

Tendo a matriz de ganho de realimentação linear L e resolvido o problema de cálculo de estados a partir das saídas do sistema, podemos combinar essas duas soluções para obtermos a solução desejada para o problema de alocação de polos e, considerando que nem todos os estados possam ser medidos, calcule sua função de transferência num projeto de sistema de controle completo.

4.2.2 Atividade Experimental

Deseja-se projetar um sistema de controle utilizando alocação de pólos e a técnica de observadores de estados no processo construído na preparação com amplificadores operacionais.

- Especificar a função de transferência do processo. Construir no Simulink, o diagrama de simulação para o sistema apresentado anteriormente observando a resposta do sistema em malha-aberta.
- Determinar as equações de estado do item anterior, comparar por simulação com o resultado anterior.
- Escrever a forma discreta do sistema.
- Especificar a alocação dos pólos desejados para atender os requisitos de projeto.
- Projetar o controlador utilizando os passos fornecidos na seção projeto por alocação de pólos.
- Projetar um observador de estados para estimar os estados.
- Avaliar o observador projetado no item anterior funcionando em conjunto com a lei de controle. Verificar o comportamento da variável de saída, provocando variações de 10

- Implementar a lei de controle e o observador de estados utilizando equações discretas para ser implementada no servomotor.
- Avaliar os resultados experimentais com os obtidos na prática.

4.2.3 Resoluções

Resolução da Preparação

Dados: Tempo de Subida, $t_r = 100$ ms Tempo de pico, $t_p = 1500$ ms Tempo de Acomodação, $t_s = 0.2$ s

para sistemas de segunda ordem, temos que:

$$t_r = \frac{\pi - \beta}{\omega_d}$$

e

$$\beta = \operatorname{tg}^{-1} \frac{\omega_d}{\sigma}$$

onde $\sigma = \zeta \cdot \omega_n$ e $\omega_d = \omega_n \sqrt{1 - \zeta^2}$

além disso, sabemos que $t_p = \frac{\pi}{\omega_d}$ e que para $t \geq 0$, $t_s = \frac{4}{\sigma}$

logo, se $t_p = \frac{\pi}{\omega_d}$, $\omega_d = \frac{\pi}{t_p}$

além disso, como $t_r = \frac{\pi - \beta}{\omega_d}$, $t_r \cdot \omega_d = \pi - \beta$ e portanto $\beta = \pi - t_r \cdot \omega_d$

Seja

$$\begin{aligned} \beta &= \operatorname{tg}^{-1} \frac{\omega_d}{\sigma} \\ \operatorname{tg}(\beta) &= \frac{\omega_d}{\sigma} \\ \sigma &= \frac{\omega_d}{\operatorname{tg}(\beta)} \end{aligned}$$

Mas

$$\begin{aligned} \sigma &= \zeta \cdot \omega_n \\ \zeta &= \frac{\sigma}{\omega_n} \end{aligned}$$

Por fim, temos que $\omega_d = \omega_n \sqrt{1 - \zeta^2}$, o que desenvolvido nos permite encontrar que $\omega_n^2 = \omega_d^2 + \sigma^2$

Substituindo os dados fornecidos para um sistema de segunda ordem ($s^2 + 2 \cdot \zeta \cdot \omega_n \cdot s + \omega_n^2$), encontramos que $\zeta = 0,766$ e $\omega_n = 54,3$, o que nos dá os polos em $-12.09 + 20.94 \cdot j$ e $-12.09 - 20.94 \cdot j$

Discretizando o sistema, encontramos

$$P(Z) = \frac{0.02006 \cdot Z + 0.02225}{Z^2 - 2.323 \cdot Z + 1.366}$$

Polos discretos: $1.16 + 0.13 \cdot j$ e $1.16 - 0.13 \cdot j$

Para encontrar o L então basta fazer $u(k) = -Lx(K)$ e encontrar $\det(Z \cdot I - [\phi * -\Gamma \cdot L]) = 0$, o que nos dá $L_1 = 2,49$ e $L_2 = -0.057$

Analogamente encontramos K através da equação $\det(Z \cdot I - [\phi - K \cdot C]) = 0$, o que nos dá $K_1 = -1000$ e $K_2 = 292.05$

4.3 Experimento 2 - Servo Motor de Precisão - Identificação de Sistemas

Nesse experimento foram introduzidas as técnicas dos Mínimos Quadrados, Resposta ao Degrau e Método do Relé para identificação de sistemas. Ao final desse experimento o aluno deverá ser capaz de realizar a identificação de sistemas a partir de dados do processo obtidos de forma experimental.

Nas práticas experimentais o aluno utilizará o programa Matlab para a implementação das técnicas discutidas na identificação do sistema da bancada Servomotor DC. Esse sistema será representado pelos dados experimentais fornecidos aos alunos. Este guia é uma atualização do guia de experimentos já utilizado na disciplina.

4.3.1 Atividade de Preparação

Preparação 1

Escreva dois programas no Matlab para calcular os parâmetros de um sistema linear discreto usando o método dos Mínimos Quadrados (Sessão 1.1). Os programas devem calcular cada um dos parâmetros dos modelos discretos

$$G_1(s) = \frac{b_1q^{-1}}{1+a_1q^{-1}}$$

$$G_1(s) = \frac{b_1q^{-1}+b_2q^{-2}}{1+a_1q^{-1}+a_2q^{-2}}$$

A partir de N amostras dos sinais y e u. Para facilitar, escreva a saída y em função da entrada u usando os modelos indicados na forma de regressão linear:

- Modelo G_1

$$\begin{aligned} y(t) + a_1y(t-1) &= b_1u(t-1) \\ y(t) &= -a_1y(t-1) + b_1u(t-1) \end{aligned}$$

$$y(t) = \begin{vmatrix} -y(t-1) & u(t-1) \end{vmatrix} \cdot \begin{vmatrix} a_1 \\ b_1 \end{vmatrix} = \phi_1^T(t) \cdot \theta_1 \quad (4.1)$$

- Modelo G_2

$$\begin{aligned} y(t) + a_1y(t-1) + a_2y(t-2) &= b_1u(t-1) + b_2u(t-2) \\ y(t) &= -a_1y(t-1) - a_2y(t-2) + b_1u(t-1) + b_2u(t-2) \end{aligned}$$

$$y(t) = \begin{vmatrix} -y(t-1) & -y(t-2) & u(t-1) & u(t-2) \end{vmatrix} \cdot \begin{vmatrix} a_1 \\ a_2 \\ b_1 \\ b_2 \end{vmatrix} = \phi_2^T(t) \cdot \theta_2 \quad (4.2)$$

Para fazer os programas, siga os seguintes passos para cada um individualmente:

- Defina o tamanho do vetor de parâmetros θ , a partir das expressões (4.1) e (4.2).
- Crie matrizes R e f com valores nulos, nas dimensões: $R \rightarrow$ matriz quadrada de ordem igual ao número de colunas de θ ; $f \rightarrow$ Matriz coluna, com tamanho igual ao tamanho de θ .
- Neste passo, deve-se criar um laço usando o comando *for...end* para implementar os somatórios das expressões (2.45) e (2.46). Lembre-se de que no Matlab os vetores possuem índices a partir do numeral 1 (não existe $y(0)$, $y(-1)$, etc). Observe portanto o índice do vetor na hora de definir a variável de contagem do laço *for...end*. A cada interação do laço:
 - Monte uma variável Phi contendo o vetor $\phi(t)$ com t a interação atual, usando a equação (4.1) para o primeiro programa e a equação (4.2) para o segundo programa.
 - Calcule a matriz R como sendo: $R = R + \phi(t)\phi^T(t)$.
 - Calcule a matriz f como sendo: $f = f + \phi(t)y(t)$.
- Ao sair do laço, crie uma variável $teta$ que será o vetor de parâmetros obtidos a partir da equação (2.47), i.e., $\theta = R^{-1}.f$.
- Para cada programa, monte variáveis num e den contendo o numerador e denominador do modelo respectivamente, extraindo os parâmetros a partir da variável θ criada no passo anterior. Por exemplo, para o segundo programa

$$\begin{aligned} num &= | b_1 \quad b_2 | \\ den &= | 1 \quad a_1 \quad a_2 | \end{aligned}$$

onde os elementos destes vetores são obtidos de θ .

Preparação 2

Adapte o programa desenvolvido na preparação 1 para a estimação de parâmetros a partir da resposta ao degrau. Não se esqueça de incluir o cálculo de $A(\tau)$ dentro do laço principal do programa. Para o cálculo de $A(\theta)$ a cada interação, aproxime e integral por uma soma, i.e.,

$$A(\tau) = \int_0^\tau y(t)d(t) \cong \Delta t \sum_{n=0}^{\tau} y(n\Delta t)$$

Dica:

- O tamanho do vetor θ neste caso é 3 (equação 2.56);
- A variável de controle do laço deve ser modificada para suportar apenas valores $\tau > L$;
- A variável Phi montada no programa anterior a cada laço de iteração deve ser substituída pela expressão 2.55. Deste modo, o cálculo da matriz R continua sendo feito da mesma forma;

- Note que no cálculo da matriz f (equação 2.46) o sinal $y(t)$ deve ser substituído por $A(\theta)$ (compare as equações (2.42) e (2.54));
- Assuma $\Delta = 1$ e use a função *sum.m* do Matlab para calcular o somatório a cada iteração. Lembre-se de fornecer os limites no sinal na hora de utilizar a função (Ex: $A = \text{sum}(y(1:\tau));$).

4.3.2 Atividade Experimental

Coletando os Dados para Identificação

Para o motor DC do Módulo Servomotor, faremos a coleta de dados para a identificação tanto de velocidade quanto de posição através do seguinte passo-a-passo (mais informações sobre o Servomotor DC podem ser adquiridas na apostila referente ao mesmo embasada no material disponibilizado pela *Feedback Instruments* [5], [4], [6]):

1. Ainda não ligue o Módulo Servomotor. Com o matlab, abra o modelo para simulação de Velocidade, encontrado no diretório $C \rightarrow \text{Feedback} \rightarrow \text{Modular Servo} \rightarrow \text{Examples} \rightarrow \text{Real} - \text{timemodels} \rightarrow \text{VelocityModel_Ident}$
2. No Simulink será exibido o modelo como mostrado na figura (4.1), abrindo o "Multisteps excitation", visualiza-se o sinal de controle aplicado no sistema como mostrado na figura (4.2), que consiste em um somatório de degraus, inicialmente sem ruído.

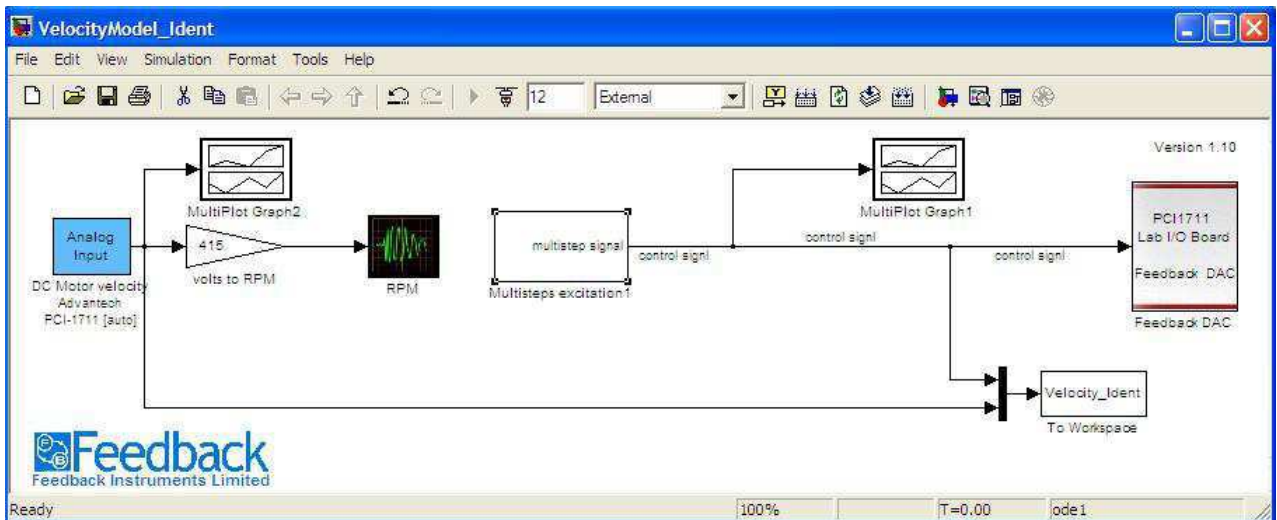


Figura 4.1: Modelo para identificação de Velocidade

3. Renomeie a saída do bloco "To Workspace" para "Vel_Ident" Agora use a função *Build* (aperte "Control+B") e o modelo será construído e em seguida aperte Connect to Target.
4. Ligue a alimentação do Módulo Servomotor através do *Power Supply* (PS150E), se o motor estiver efetuando algum trabalho, use o *Pre-Amplificator Unit* (PA150C) e o *Attenuator Unit* (AU150B) para atenuar até que este fique parado em regime .

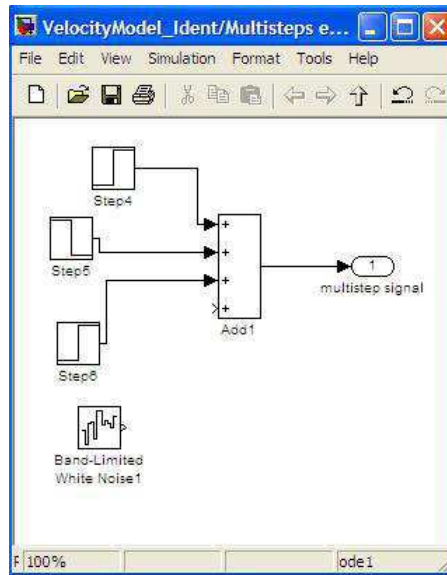


Figura 4.2: Sinal de controle implementado durante a identificação de Velocidade

5. De volta no simulink, selecione *"Real Time Code"* e o processo terá início.
6. Ao fim do processo, será gerado um arquivo na área de trabalho do Matlab chamado *Vel_Ident*, salve esses dados com o nome *Velocity_Ident* (Clique com o botão direito e aperte *"Save as.."*). Este arquivo será utilizado no experimento de identificação.
7. Agora iremos adicionar um ruído na entrada do sistema. Renomeie a saída do bloco *"To Workspace"* para *"Vel_Ident_Temp"* e então abra o *"Multisteps excitation"*, ligando o *"Band-Limited White Noise"* como mostrado na figura (4.3).

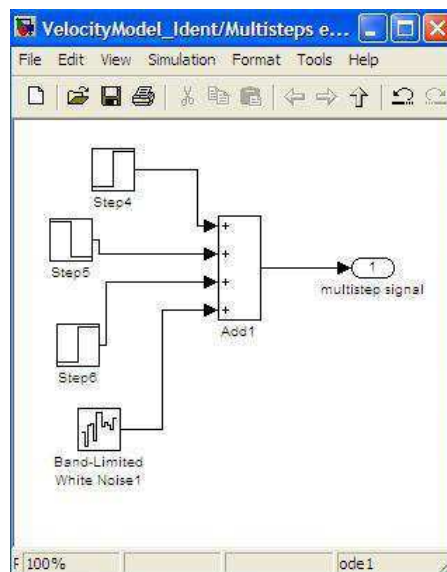


Figura 4.3: Sinal de controle implementado durante a identificação de Velocidade com Ruído Branco

8. Use a função *Build*, selecione *Connect to Target* e então *Real Time Code*, depois de concluído o experimento, mescle a entrada do *Velocity_Ident* e a saída do *Vel_Ident_Temp* para gerar o *Velocity_Ident_Ruido*, que será utilizado na identificação do sistema.
9. Use as funções *Clear all* e *Clc* no matlab e feche a simulação do *Velocity_Ident*, agora abra o modelo para simulação de Posição, encontrado no diretório *C* → *Feedback* → *ModularServo* → *Examples* → *Real-time models* → *PositionModel_Ident*
10. No Simulink será exibido o modelo como mostrado na figura (4.4) para o cálculo de posição angular do Motor DC, abrindo o *Multisteps excitation*, visualiza-se o mesmo sinal de controle aplicado no sistema anterior, como mostrado na figura (4.2).

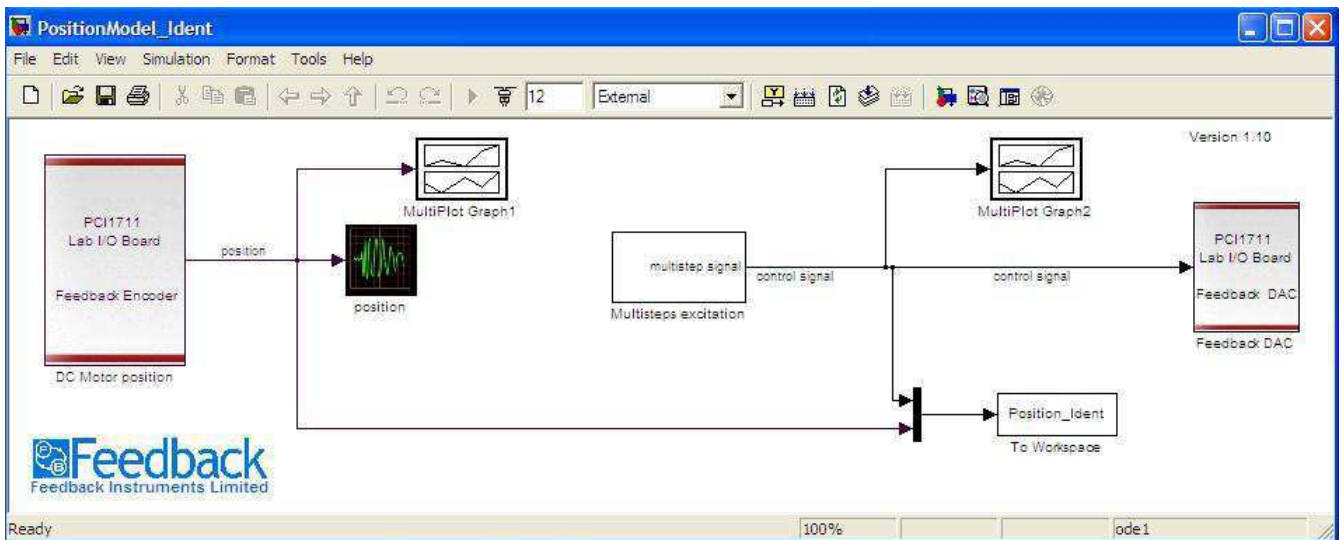


Figura 4.4: Modelo para identificação de Posição

11. Renomeie a saída do bloco *To Workspace* para *Pos_Ident*. Agora use a função *Build* (aperte *Control + B*) e o modelo será construído e em seguida aperte *Connect to Target*. Selecione então *Real Time Code* e o processo terá início.
12. Ao fim do processo, será gerado um arquivo na área de trabalho do Matlab chamado *Pos_Ident*, salve esses dados com o nome *Position_Ident*. Este arquivo será utilizado no experimento de identificação.

Atividade Experimental 1

Abra na linha de comando do Matlab os dados coletados na simulação do Motor DC de velocidade: *Velocity_Ident* (Utilize o comando *Load* para abrir o arquivo e então os comandos $u = \text{Velocity_Ident}(:,1)$ e $y = \text{Velocity_Ident}(:,2)$ para separar entrada e saída). Plote na tela os sinais discretos u e y . Verifique o primeiro programa da preparação 1 usando os dados recém adquiridos. Simule o modelo obtido usando o mesmo sinal u como

entrada (função *dlsim*, e plote em uma mesma figura ambos, o sinal *y* e o simulado. Em seguida faça o mesmo para o *Position_Ident*. Use *Npts* igual ao tamanho de *u*.

- Questão 1: Obtenha o equivalente contínuo do modelo discreto obtido e anote o resultado, indicando o(s) polo(s) do modelo obtido (use a função *d2c.m* e assuma que o período de amostragem é unitário).
- Questão 2: Há alguma discrepância entre o sinal de saída do sistema real e aquele obtido usando o modelo estimado? Explique a resposta.

Atividade Experimental 2

Verifique agora o segundo programa da preparação 1 usando os dados coletados na simulação do Motor DC *Velocity_Ident*. Simule o modelo obtido usando o mesmo sinal *u* como entrada (função *dlsim*), e plote em uma mesma figura ambos, o sinal *y* e o simulado. Em seguida faça o mesmo para *Position_Ident*. Use *Npts* igual ao tamanho de *u*.

- Questão 1: Obtenha o equivalente contínuo do modelo discreto obtido e anote o resultado, indicando o(s) polo(s) do modelo obtido.
- Questão 2: Há alguma discrepância entre o sinal de saída do sistema real e aquele obtido usando o modelo estimado? Explique a resposta.

Atividade Experimental 3

Verifique o segundo programa da preparação 2 usando os dados coletados na simulação do Motor DC *Velocity_Ident_Ruido*. Simule o modelo obtido usando o mesmo sinal *u* como entrada (função *dlsim*), e plote em uma mesma figura ambos, o sinal *y* e o simulado. Use *Npts* igual ao tamanho de *u*.

- Questão 1: Obtenha o equivalente contínuo do modelo discreto obtido e anote o resultado, indicando o(s) polo(s) do modelo obtido.
- Questão 2: Há alguma discrepância entre o sinal de saída do sistema real e aquele obtido usando o modelo estimado? Explique a resposta.

Atividade Experimental 4

Verifique o programa da preparação 2 usando os dados gerados pelo programa fornecido *degrau.m* (Utilização: `[y;u] = degrau(Npts;L);`). Simule o modelo obtido usando o sinal *u* como entrada e plote em uma mesma figura ambos, o sinal *y* e o simulado. Use *Npts* = 200 e *L* = 30.

- Questão 1: Anote o modelo obtido na forma:

$$G(s) = \frac{G_o}{T_1 s + 1} \cdot e^{-Ls}$$

Atividade Experimental 5

Monte no Simulink o método do relé da figura (2.8), usando a planta

$$G(s) = \frac{1}{s+1} \cdot e^{-Ls}$$

Simule o experimento de modo a ver um número razoável de oscilações na saída da planta. Blocos necessários no Simulink: *Transfer Fnc*, *Relay*, *Auto-Scale*, *Graph*, *Sum*, *Step* e *Transport Delay*.

- Questão 1: Obtenha estimativas para o ganho crítico e a frequência crítica observando a saída do sistema quando submetido a uma realimentação com relé.
- Questão 2: Existe alguma diferença entre os valores exatos (calculados na preparação) e os valores obtidos com o experimento com relé? Justifique sua resposta.

4.3.3 Resoluções

Resoluções das Atividades de Preparação

As figuras 4.5(a) e 4.5(b) mostram os códigos que devem ser desenvolvidos no ambiente de compilação do matlab como atividades de preparação.

Resoluções das Atividades Experimentais

Atividade Experimental 1

As figuras 4.6(a) e 4.6(b) mostram, respectivamente, o código do modelo para os dados de velocidade e então de posição. Em seguida, as figuras 4.7(a) e 4.7(b) mostram, respectivamente, a resposta do modelo para os dados de velocidade e posição.

Notamos que o equivalente contínuo para o *Velocity_Ident* dos dados obtidos experimentalmente foi de:

$$TF = \frac{0.02298}{s + 0.0131}$$

onde TF é a função de transferência.

E para o *Position_Ident* foi de:

$$TF = \frac{0.06685}{s - 2.366^{-5}}$$

Notamos que mesmo com o ruído, o modelo conseguiu acompanhar o sistema sem muitos contra-tempos.

Atividade Experimental 2

As figuras 4.8(a) e 4.8(b) mostram, respectivamente, o código do modelo para os dados de velocidade e então de posição. Em seguida, as figuras 4.9(a) e 4.9(b) mostram, respectivamente, a resposta do modelo para os dados de velocidade e posição.

```

1 - clc;
2 - clear;
3 - l=2;
4
5 - r=zeros(1);
6 - f=zeros(1,1);
7
8 - for t=2:Npts
9 -     phi= [-y(t-1) u(t-1)]';
10 -    r=r+phi*phi';
11 -    f=f+phi*y(t);
12 - end
13
14 - Teta=inv(r)*f;
15 - num1= [0 Teta(2)];
16 - den1=[1 Teta(1)];
17 - t = [0:0.001:(Npts-1)*0.001];
18 - figure(1)
19 - graf = dsim(num1,den1,u);
20 - plot(t,graf,'b');
21 - grid;
22 - hold
23 - plot(t,y,'r');
24 - grid;
25 -

```

(a) Programa da Preparação 1

```

1 - clear;
2 - clc;
3 - l=4;
4 - r=zeros(1);
5 - f=zeros(1,1);
6 - for t=3:Npts
7 -     phi= [-y(t-1) -y(t-2) u(t-1) u(t-2)]';
8 -     r=r+phi*phi';
9 -     f=f+phi*y(t);
10 - end
11 - Teta=inv(r)*f;
12 - num1= [0 Teta(3) Teta(4)];
13 - den1=[1 Teta(1) Teta(2)];
14 - t = [0:0.001:(Npts-1)*0.001];
15 - figure(2)
16 - graf = dsim(num1,den1,u);
17 - plot(t,graf,'b');
18 - grid;
19 - hold
20 - plot(t,y,'r');
21 - grid;

```

(b) Programa da Preparação 2

Figura 4.5: Programas das Preparações

Notamos que o equivalente contínuo para o $Velocity_Ident$ dos dados obtidos experimentalmente foi de:

$$TF = \frac{-0.01889 \cdot s + 0.02853}{s^2 + 1.201 \cdot s + 0.01628}$$

onde TF é a função de transferência.

E para o $Position_Ident$ foi de:

$$TF = \frac{-0.0004989 \cdot s + 0.002}{s^2 + 0.02942 \cdot s - 3.549 \cdot 10^{-7}}$$

Notamos também que só existe discrepância no caso de frequências muito altas acima de $10^0 \frac{rad}{s}$. Abaixo dessa frequência sendo praticamente análogo. Essa frequência é maior em relação á da atividade experimental anterior exatamente graças ao aumento da ordem do modelo.


```

1 -   clc;
2 -   clear;
3 -   l=2;
4 -   r=zeros(1);
5 -   f=zeros(1,1);
6 -   load Velocity_Ident;
7 -   u = Velocity_Ident(:,1)
8 -   y = Velocity_Ident(:,2)
9 -   y = y+0.21;
10 -  Npts=length(u);
11 -  for t=2:Npts
12 -      phi= [-y(t-1) u(t-1)]';
13 -      r=r+phi*phi';
14 -      f=f+phi*y(t);
15 -  end
16 -  Teta=inv(r)*f;
17 -  num1= [0 Teta(2)];
18 -  den1=[1 Teta(1)];
19 -  t = [0:0.001:(Npts-1)*0.001];
20 -  figure(1)
21 -  graf = dlsim(num1,den1,u);
22 -  plot(t,graf,'-');
23 -  grid;
24 -  hold
25 -  plot(t,y,'r');
26 -  grid;

```

```

1 -   clc;
2 -   clear;
3 -   l=2;
4 -   r=zeros(1);
5 -   f=zeros(1,1);
6 -
7 -   load Position_Ident;
8 -   u = Position_Ident(:,1)
9 -   y = Position_Ident(:,2)
10 -  Npts=length(u);
11 -  for t=2:Npts
12 -      phi= [-y(t-1) u(t-1)]';
13 -      r=r+phi*phi';
14 -      f=f+phi*y(t);
15 -  end
16 -  Teta=inv(r)*f;
17 -  num1= [0 Teta(2)];
18 -  den1=[1 Teta(1)];
19 -  t = [0:0.001:(Npts-1)*0.001];
20 -  figure(1)
21 -  graf = dlsim(num1,den1,u);
22 -  plot(t,graf,'b');
23 -  grid;
24 -  hold
25 -  plot(t,y,'r');
26 -  grid;
27 -
28 -

```

(a) Programa da Preparação 1 para *Velocity_Ident* (b) Programa da Preparação 1 para *Position_Ident*

Figura 4.6: Códigos da Atividade Experimental 1

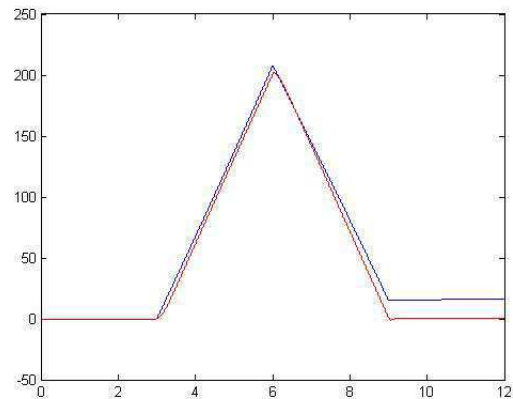
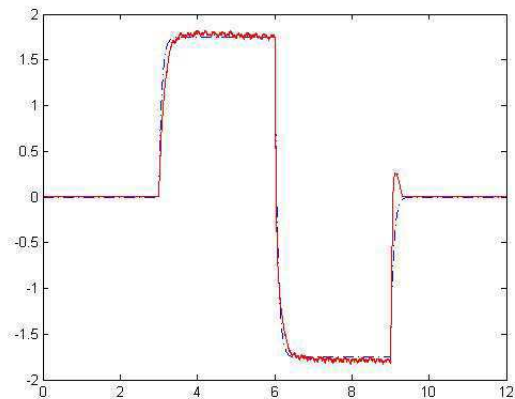
Atividade Experimental 3

As figuras 4.10(a) e 4.10(b) mostram, respectivamente, o código do modelo para os dados de velocidade e sua resposta.

A função de transferência (TF) do modelo contínuo é dada por:

$$TF = \frac{-0.004482 \cdot s + 007149}{s^2 + 1.2297 \cdot s + 0.03901}$$

Notamos que o ruído afetou diretamente a função de transferência e tentou acompanhar não apenas o sinal mas também o ruído, ainda assim há uma exatidão igual á observada no experimento anterior.



(a) Modelo da Preparação 1 para *Velocity_Ident* (b) Modelo da Preparação 1 para *Velocity_Ident*

Figura 4.7: Respostas dos modelos da Atividade Experimental 1

```

1 - clear;
2 - clc;
3 - l=4;
4 - r=zeros(1);
5 - f=zeros(1,1);
6 - load Position_Ident;
7 - u = Position_Ident(:,1)
8 - y = Position_Ident(:,2)
9 - Npts=length(u);
10 - for t=3:Npts
11 -     phi = [-y(t-1) -y(t-2) u(t-1) u(t-2)]';
12 -     r=r+phi*phi';
13 -     f=f+phi*y(t);
14 - end
15 - Teta=inv(r)*f;
16 - num1= [0 Teta(3) Teta(4)];
17 - den1=[1 Teta(1) Teta(2)];
18 - t = [0:0.001:(Npts-1)*0.001];
19 - figure(2)
20 - graf = dlsim(num1,den1,u);
21 - plot(t,graf,'b');
22 - grid;
23 - hold
24 - plot(t,y,'r');
25 - grid;
    
```

```

1 - clear;
2 - clc;
3 - l=4;
4 - r=zeros(1);
5 - f=zeros(1,1);
6 - load Velocity_Ident;
7 - u = Velocity_Ident(:,1)
8 - y = Velocity_Ident(:,2)
9 - y = y+.21;
10 - Npts=length(u);
11 - for t=3:Npts
12 -     phi = [-y(t-1) -y(t-2) u(t-1) u(t-2)]';
13 -     r=r+phi*phi';
14 -     f=f+phi*y(t);
15 - end
16 - Teta=inv(r)*f;
17 - num1= [0 Teta(3) Teta(4)];
18 - den1=[1 Teta(1) Teta(2)];
19 - t = [0:0.001:(Npts-1)*0.001];
20 - figure(2)
21 - graf = dlsim(num1,den1,u);
22 - plot(t,graf,'r');
23 - grid;
24 - hold
25 - plot(t,y,'b');
26 - grid;
    
```

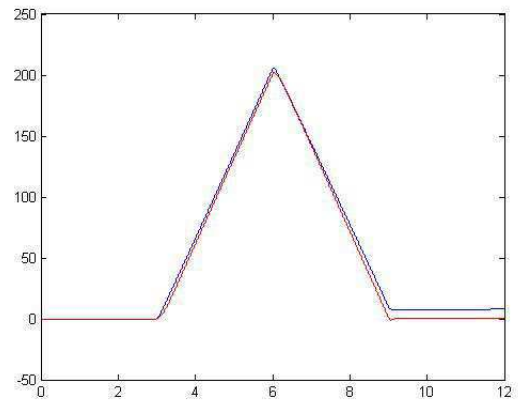
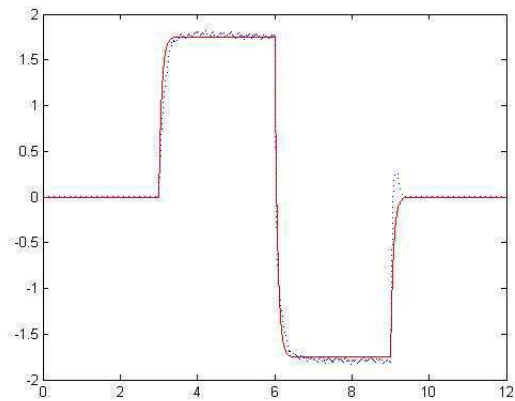
(a) Programa da Preparação 2 para *Velocity_Ident* (b) Programa da Preparação 2 para *Position_Ident*

Figura 4.8: Códigos da Atividade Experimental 2

Atividade Experimental 4

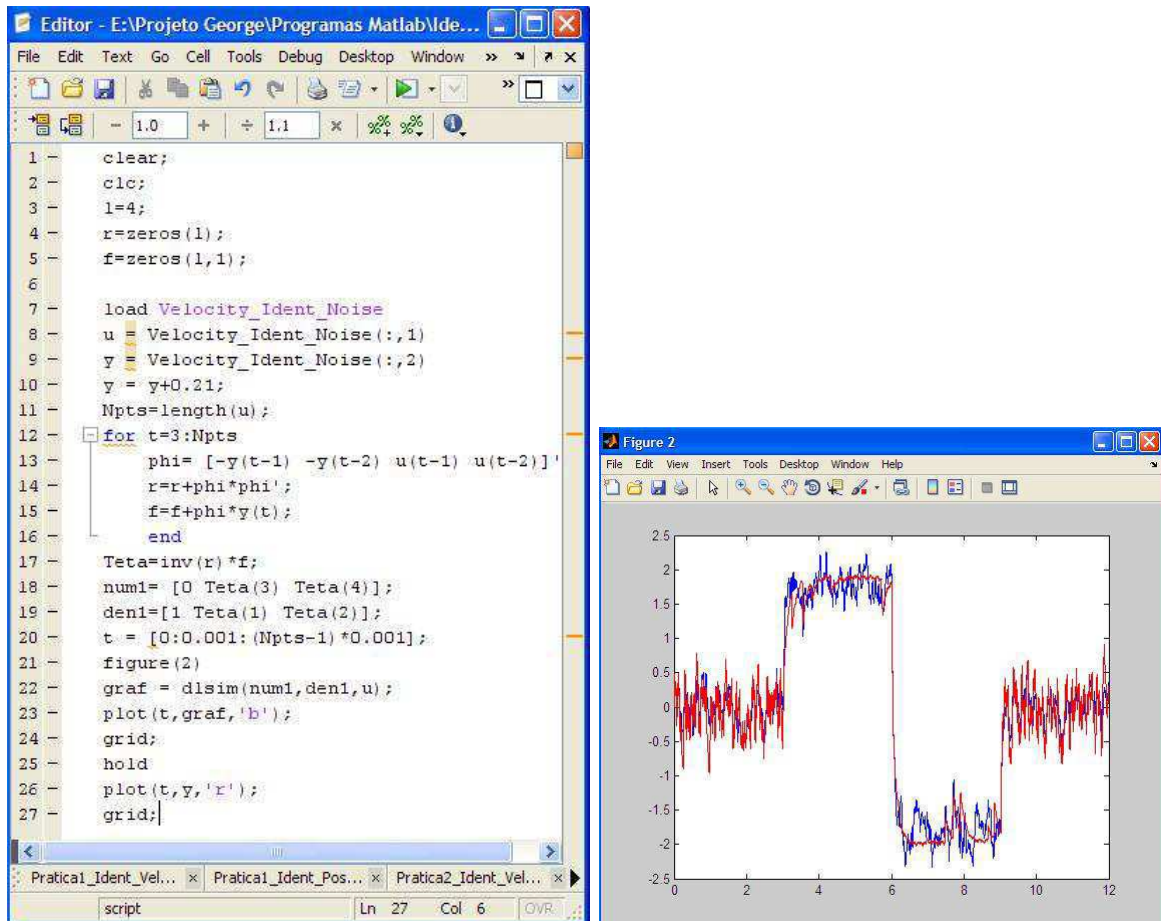
As figuras 4.11(a) e 4.11(b) mostram, respectivamente, o código do modelo para o degrau e sua resposta.

Experimentos Propostos



(a) Modelo da Preparação 2 para *Velocity_Ident* (b) Modelo da Preparação 2 para *Velocity_Ident*

Figura 4.9: Respostas dos modelos da Atividade Experimental 2

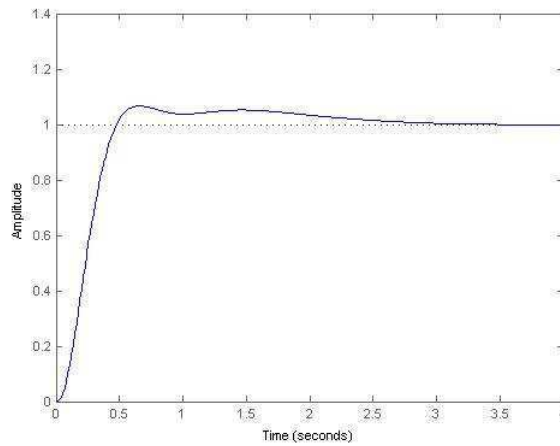


(a) Modelo da Preparação 2 para *Velocity_Ident_Noise* (b) Modelo da Preparação 2 para *Velocity_Ident_Noise*

Figura 4.10: Código e resposta do modelo da Atividade Experimental 3

```
Editor - C:\Documents and Settings\usuario\De...
File Edit Text Go Cell Tools Debug Desktop
- 1.0 + 1.1 x %>% %>%
1 - clc
2 - clear
3 - n=3;
4 - Npts=200;
5 - L=30;
6 - [y,u]=degrau(Npts,L);
7 - r=zeros(n);
8 - f=zeros(n,1);
9 - for tau=L+1:Npts,
10 -     A=sum(y(2:tau-1)+(y(1)+y(tau)))/2;
11 -     phi=[tau -1 -y(tau)]';
12 -     r=r+phi*phi';
13 -     f=f+phi*A;
14 - end
15 - teta=inv(r)*f
16 - figure(4)
17 - plot(u)
18 - hold on
19 - plot(y,'g')
```

(a) Modelo da Preparação 4 para degrau



(b) Resposta da Preparação 4 para para degrau

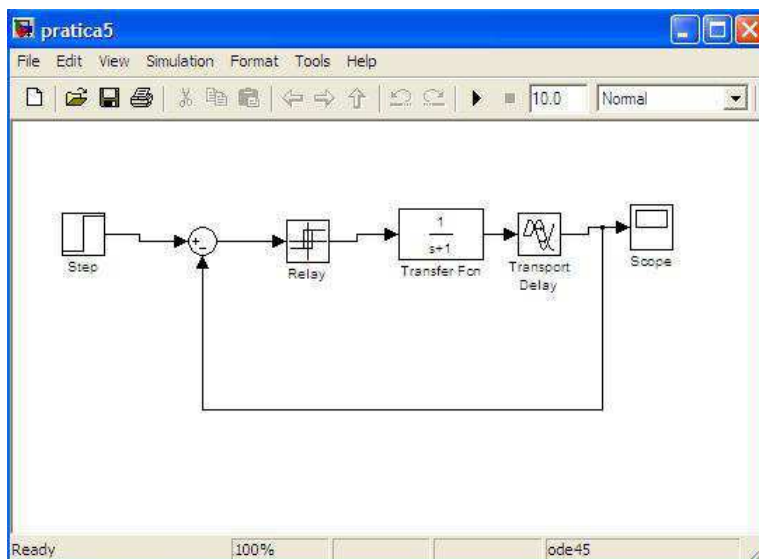
Figura 4.11: Código e resposta do modelo da Atividade Experimental 4

A função de transferência (TF) do modelo contínuo é dada por:

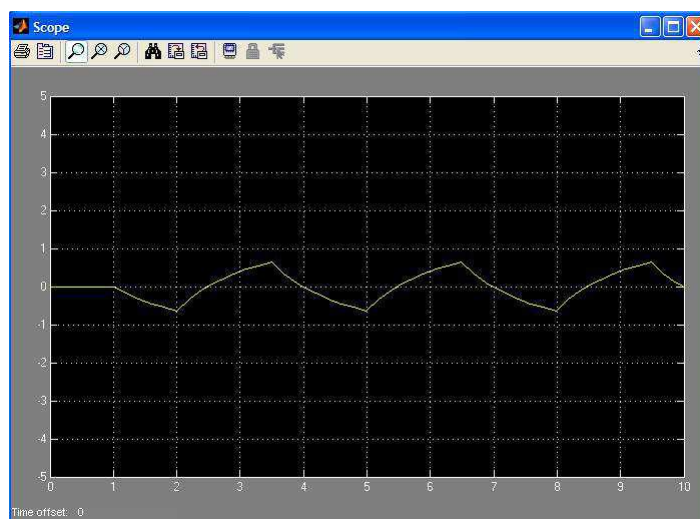
$$TF = \frac{5.1075}{17.3792 \cdot s + 1} \cdot e^{-30 \cdot s}$$

Atividade Experimental 5

As figuras 4.12(a) e 4.12(b) mostram, respectivamente, o sistema no simulink e sua resposta quando atinge o ganho crítico. Basicamente o ganho deve ser aumentado até o ponto em que a senóide começa e demonstrar-se distorcida e então observa-se seu período para que com sua inversa se tenha a frequência crítica. No geral a diferença entre os valores práticos e os da preparação não devem ser grandes, embora sempre existirão.



(a) Modelo da Preparação 4 para degrau



(b) Resposta da Preparação 4 para para degrau

Figura 4.12: Código e resposta do modelo da Atividade Experimental 5

4.4 Experimento 3 - Levitador Magnético - Linearização e Método do Relé

Nesse experimento foi introduzida a técnica de Linearização de Pequenos Sinais. Ao final desse experimento o aluno deverá ser capaz de realizar a linearização simples a partir de um modelo não-linear obtido através de conhecimento prévio.

Nas práticas experimentais o aluno utilizará o programa Matlab para a verificação do modelo obtido na bancada Servomotor do Levitador Magnético.

4.4.1 Atividade de Preparação

Sabendo que para o Levitador Magnético da *Feedback Instruments* o ponto de equilíbrio se encontra nos valores $x_0 = -1.5$ [V] (A posição é expressa em volts), $I_0 = 0.8$ [A]:

Onde sua equação mais simplificada é representada abaixo:

$$\ddot{x} = g - f(x, i), f(x, i) = k \frac{i^2}{m \cdot x^2} \quad (4.3)$$

Ou seja, o ponto de equilíbrio pode ser calculado a partir de:

$$g = f(x, i) \Rightarrow i_0, x_0 \quad (4.4)$$

Implemente um processo de linearização simples para este sistema.

4.4.2 Atividade Experimental

Utilizando o Matlab

Para o Levitador Magnético, iremos estudar seu comportamento quando implementado com um sistema de controle PD e um sistema de controle PID para as seguintes entradas: um pulso, um sinal senoidal e um sinal de onda quadrada.

1. Ainda não ligue o Levitador Magnético. Com o matlab, abra o modelo para simulação de controle PD, encontrado no diretório $C \rightarrow Feedback \rightarrow Maglev \rightarrow Examples \rightarrow Real - timemodels \rightarrow Maglev_PD$
2. No Simulink será exibido o modelo como mostrado na figura (4.13).
3. Escolha o tipo de entrada desejada. Agora use a função *Build* (aperte "Control+B") e o modelo será construído e em seguida aperte *Connect to Target*.
4. Ligue a alimentação do Levitador Magnético encontrada em sua traseira.
5. De volta no simulink, selecione "*Real Time Code*" e o processo terá início;
6. A esfera metálica deve ser guiada de modo manual até o ponto de equilíbrio;

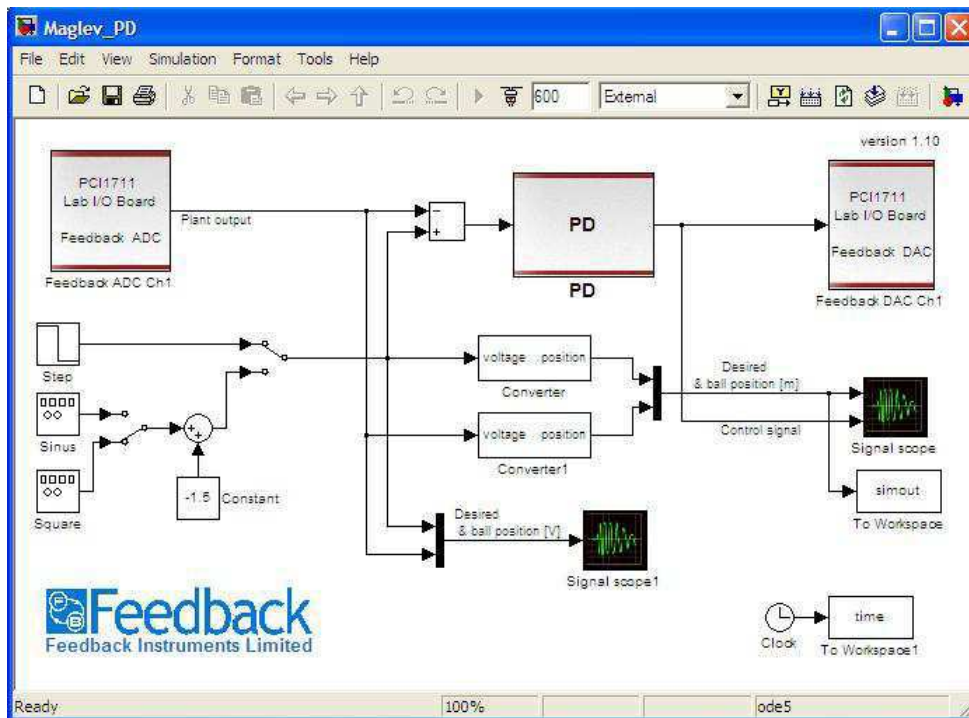


Figura 4.13: Programa *Maglev_PD*

7. Modifique os tipos de entrada, assim como suas amplitude e frequências e estude o comportamento do sistema.
8. Desligue o Levitador Magnético. Com o matlab, abra o modelo para simulação de controle PID, encontrado no diretório $C \rightarrow \text{Feedback} \rightarrow \text{Maglevo} \rightarrow \text{Examples} \rightarrow \text{Real} - \text{timemodels} \rightarrow \text{Maglev_PID}$
9. No Simulink será exibido o modelo como mostrado na figura (4.14).
10. Escolha o tipo de entrada desejada. Agora use a função *Build* (aperte "Control+B") e o modelo será construído e em seguida aperte *Connect to Target*.
11. Ligue a alimentação do Levitador Magnético encontrada em sua traseira.
12. De volta no simulink, selecione "*Real Time Code*" e o processo terá início;
13. A esfera metálica deve ser guiada de modo manual até o ponto de equilíbrio antes do sistema alcançar 10 segundos (é nesse instante que o processo integrativo é implementado);
14. Modifique os tipos de entrada, assim como suas amplitude e frequências e estude o comportamento do sistema.
15. Agora implemente uma entrada com realimentação por relé no sistema de Controle PD com amplitudes de -0.5 a 0.5 Volts.

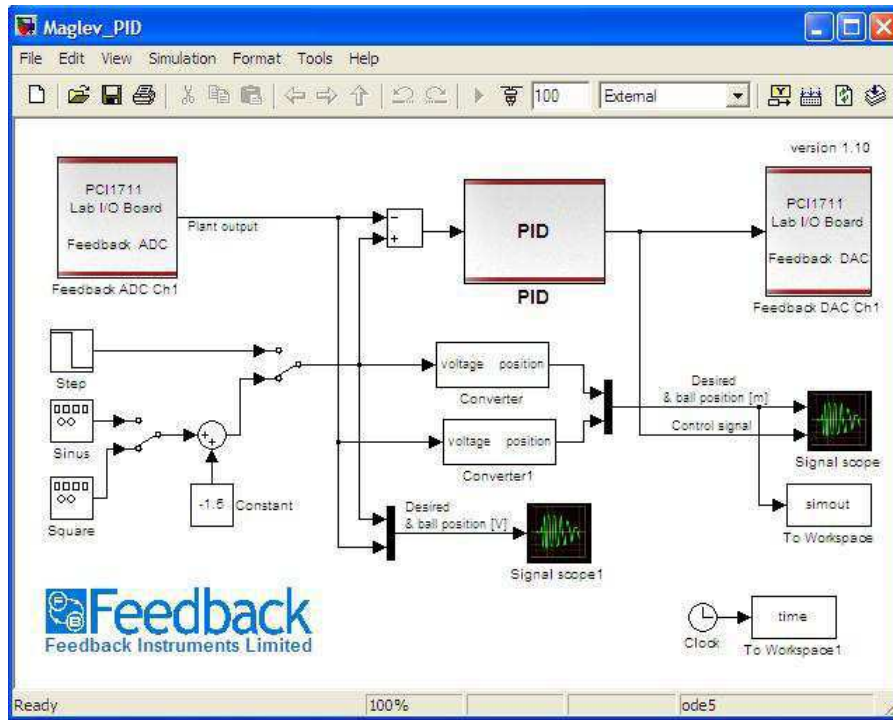


Figura 4.14: Programa *Maglev_PID*

4.4.3 Resoluções

Resoluções das Atividades de Preparação

No ponto de equilíbrio de $x_0 = -1.5$ [V]), $I_0 = 0.8$ [A]:

$$\ddot{x} = g - f(x, i), f(x, i) = k \frac{i^2}{m \cdot x^2} \quad (4.5)$$

O ponto de equilíbrio pode ser calculado a partir de:

$$g = f(x, i) \Rightarrow i_0, x_0 \quad (4.6)$$

Os passos de linearização são os seguintes:

$$\begin{aligned} \ddot{x} &= - \left(\frac{\partial f(x, i)}{\partial i} \Big|_{i_0, x_0} \Delta i + \frac{\partial f(x, i)}{\partial x} \Big|_{i_0, x_0} \Delta x \right) \\ s^2 \Delta x &= - (K_i \Delta i + K_x \Delta x) \\ s^2 \Delta x + K_x \Delta x &= -K_i \Delta i \\ \Delta x (s^2 + K_x) &= -K_i \Delta i \\ \frac{\Delta x}{\Delta i} &= \frac{-K_i}{s^2 + K_x} \end{aligned} \quad (4.7)$$

onde $K_i = \frac{2g}{i_0}$ e $K_x = \frac{-2g}{x_0}$

Resoluções das Atividades Experimentais

A figura 4.15 mostra o programa de PD em execução e a figura 4.16 mostra um comportamento da esfera para quando tanto a amplitude quanto a frequência da onda sinusoidal estavam estabelecidos em 0.25. Nota-se que a esfera acompanha o comportamento da onda, mas não com total precisão, havendo um pequeno erro. Alterando esses valores notamos também que ela ainda consegue acompanhar o comportamento da onda por uma considerável faixa de frequências e amplitudes.

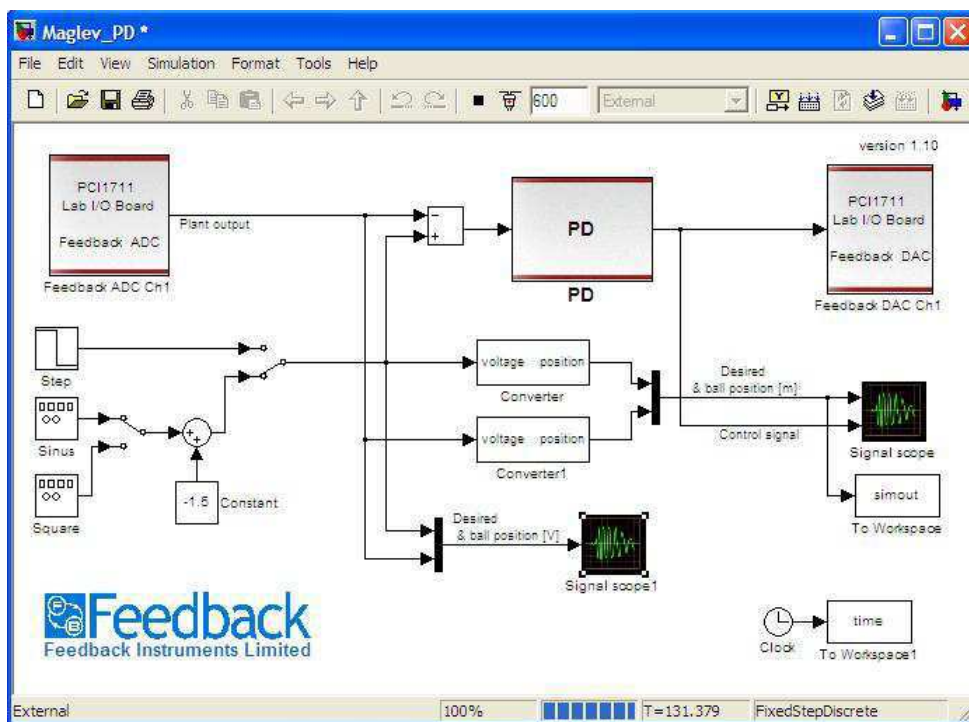


Figura 4.15: Programa *Maglev_PD* em execução

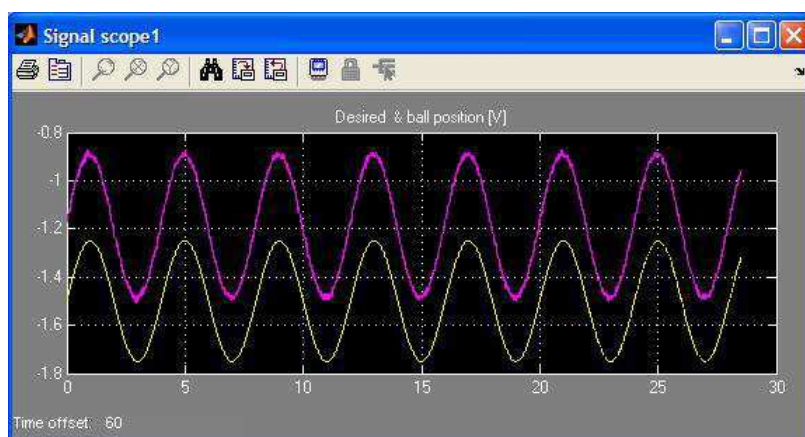


Figura 4.16: Sinal desejado e sinal adquirido durante a execução do *Maglev_PD*

A figura 4.17 mostra o programa PD em execução e a figura 4.18 mostra um

comportamento da esfera quando tanto a amplitude quanto a frequência da onda sinusoidal estavam estabelecidos em 0.25. Nota-se que a esfera acompanha o comportamento da onda com total precisão. Entretanto, alterando esses valores notamos que o integrador tem uma ação ruim e acaba integrando o erro e causando instabilidade no sistema, o que diminui sua faixa de frequências e amplitudes aceitáveis.

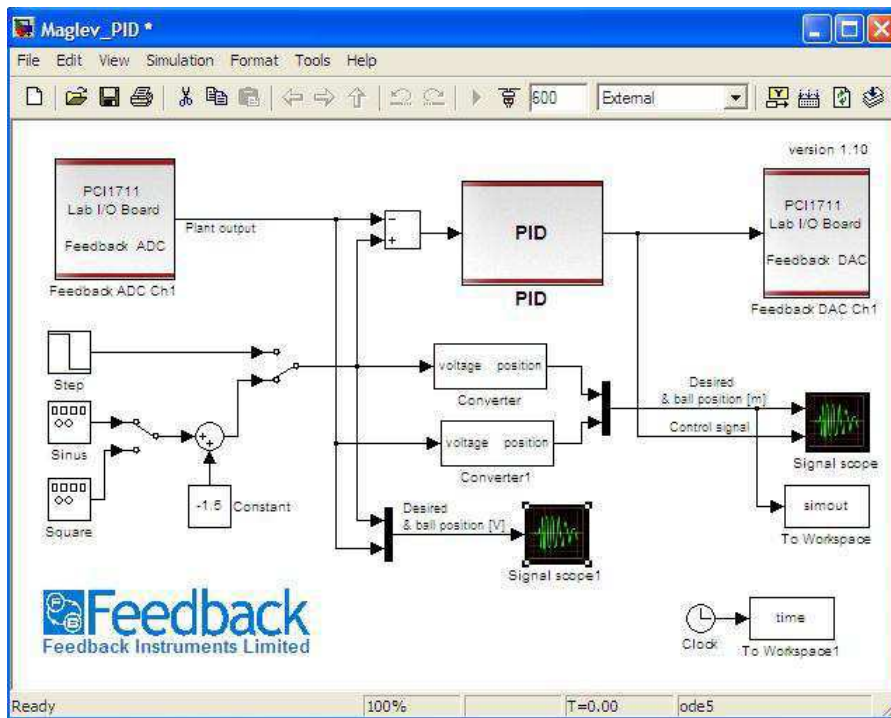


Figura 4.17: Programa *Maglev_PID* em execução

Por fim, a figura 4.19 mostra o programa PD modificado para operar com um Relé. As figuras 4.20 e 4.21 mostram o comportamento do sinal e como a esfera tentou acompanhá-lo.

4.5 Considerações

Ao término do capítulo, mostrou-se que é possível implementar a teoria vista no assunto da Disciplina de Controle Digital em seu laboratório para aplicações totalmente práticas utilizando Kits Didáticos. Foram desenvolvidos experimentos utilizando os conceitos de Alocação de Polos em Espaço de Estados, Discretização e Identificação de Sistemas. Suas resoluções são também demonstradas nesse trabalho.

Os resultados apresentados nesse trabalho podem ser explorados ainda mais para incontáveis outras aplicações, como rastreamento de posição no Servomotor e síntese de controle avançado no Levitador Magnético, logo, assim como esse trabalho foi a continuação de outro, pode servir como embasamento para várias outras atividades utilizando os kits didáticos.



Figura 4.18: Sinal desejado e sinal adquirido durante a execução do *Maglev_PID*

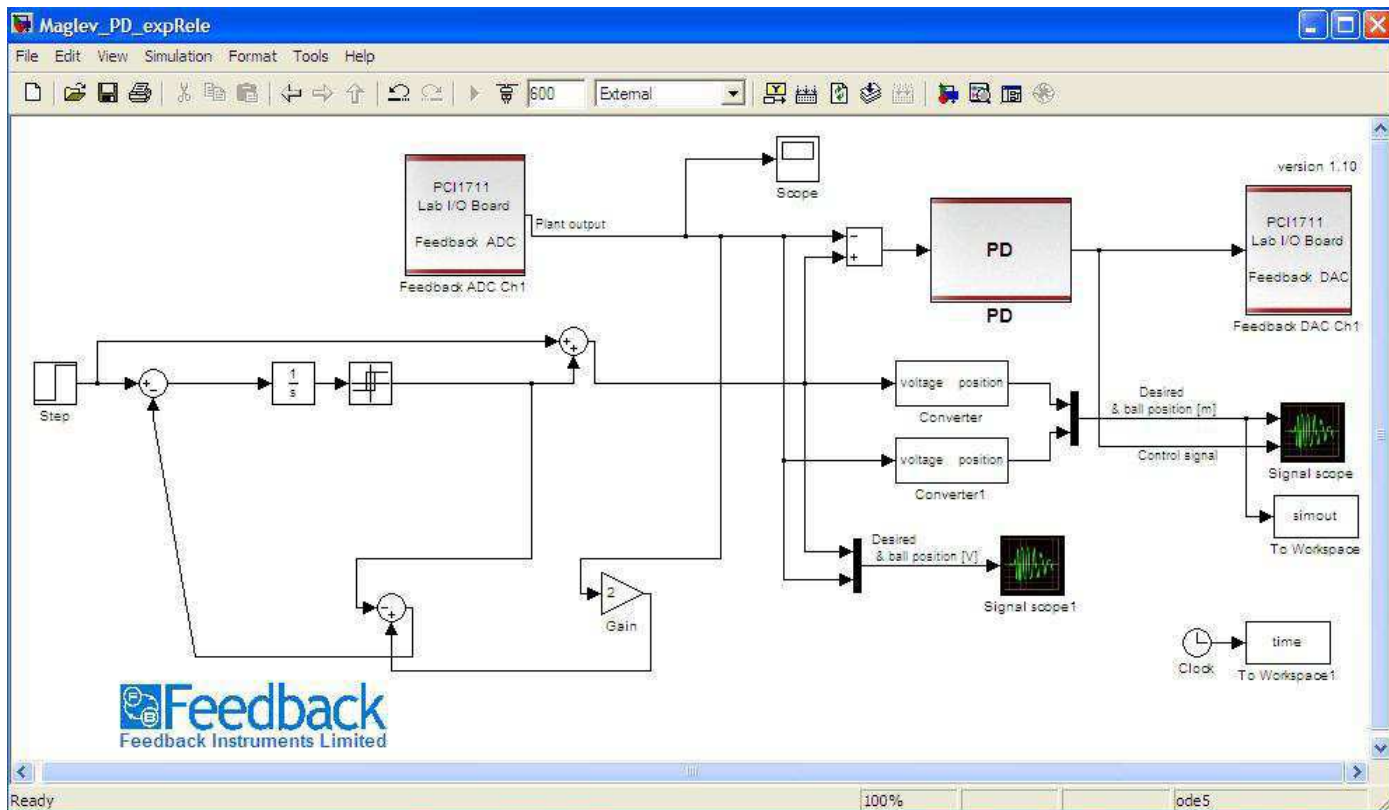


Figura 4.19: Programa *Maglev_PID_Rele* em execução

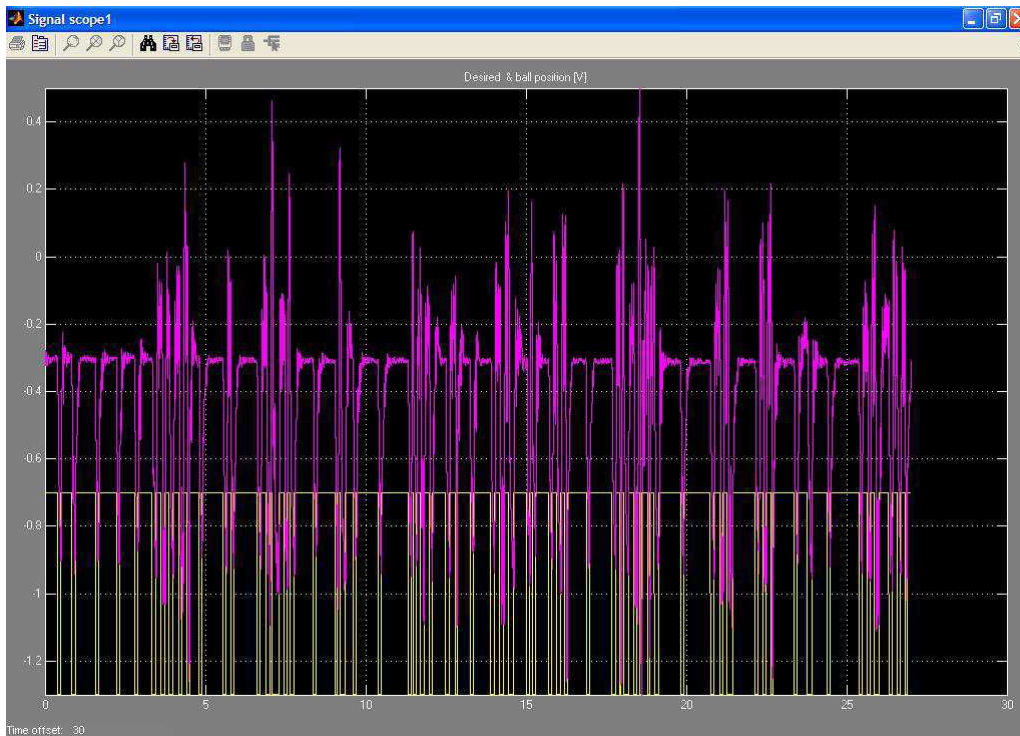


Figura 4.20: Sinal desejado e sinal adquirido durante a execução do *Maglev_PD_Rele*

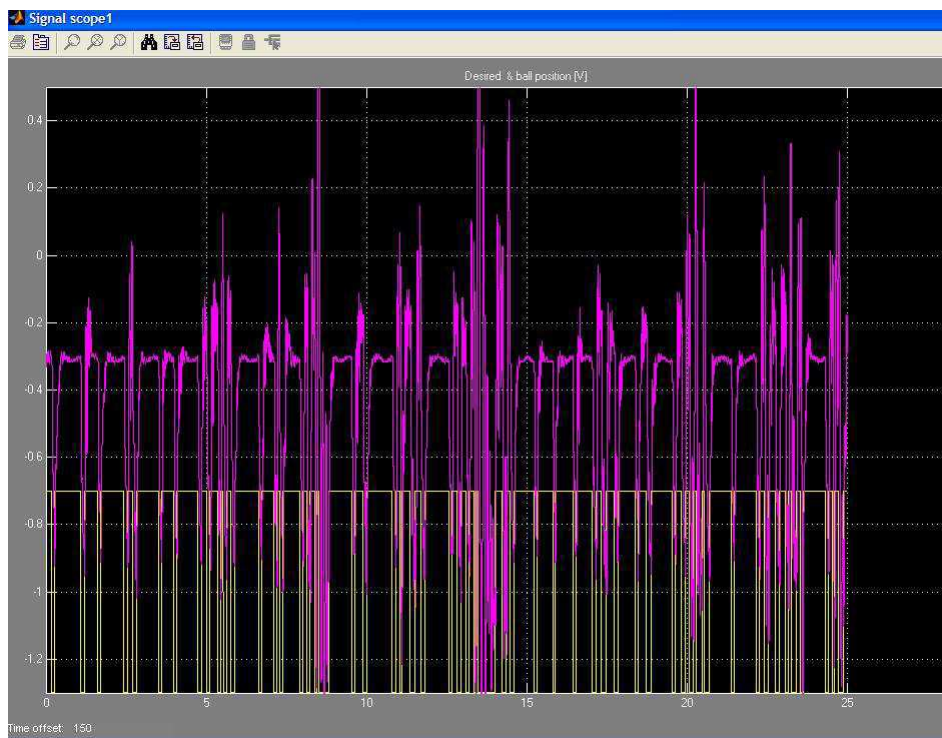


Figura 4.21: Sinal desejado e sinal adquirido durante a execução do *Maglev_PD_Rele*

Capítulo 5

Conclusão

Neste trabalho foram desenvolvidos guias de experimentos utilizando a bancada Servomotor DC do *Feedback Instruments*. Estes guias irão fornecer aos alunos da disciplina Laboratório de Controle Digital um material que os auxiliem na utilização da bancada, no projeto de sistemas de controle e na identificação de sistemas a partir de dados experimentais.

Durante o trabalho pôde-se revisar e fixar os conceitos de discretização de sistemas, alocação de polos, realimentação linear de estados, observadores e identificação de sistemas, importantes na Engenharia de Controle.

Pode-se concluir que o objetivo final deste trabalho foi atingido e que ele contribuiu para o ambiente acadêmico do curso de Engenharia Elétrica visto que com ele foi possível revisar, fixar e implementar os conceitos ministrados no curso e que os guias produzidos servirão de apoio aos alunos da disciplina Laboratório de Controle Digital.

Agradecimentos

Á Petrobrás através da ATECEL pelo financiamento do projeto que deu origem a esse trabalho; Aos colegas do Laboratório de Instrumentação Eletrônica e Controle (LIEC) da Unidade Acadêmica de Engenharia Elétrica, onde este projeto foi desenvolvido e realizado, por todo apoio e contribuição; Aos professores George Acioli Jr. e Péricles Resende Barros pela orientação e aos graduandos Chisthian Charles Dias, Rafael Soares do Egito e Rienzy de Azevedo pelo auxílio prestado. Finalmente ao engenheiro Sérgio Paulo Melo de Souza, que foi o precursor desse trabalho e cuja parte de seu material foi aproveitada.

Referências Bibliográficas

- [1] G.F. Franklin, M.L. Workman, and D. Powell. *Digital control of dynamic systems*. Addison-Wesley Longman Publishing Co., Inc., 1997.
- [2] Feedback Instruments. *Magnetic Levitation Control Experiments*.
- [3] Feedback Instruments. *Magnetic Levitation Installation and Commissioning*.
- [4] Feedback Instruments. *MATLAB GUIDE for Feedback Control Instrumentation*.
- [5] Feedback Instruments. *Precision Modular Servo Control Experiments*.
- [6] Feedback Instruments. *Precision Modular Servo Installation and Commissioning*.
- [7] Björn Wittenmark Karl Johan Åström. *Computer-Controlled Systems: Theory and Design*. Prentice Hall, New Jersey, 3 edition, 1997.
- [8] Tore Hagglund Karl Karl Johan Åström. *Advanced Pid Control*. 2006.

Apêndice A

Apêndice

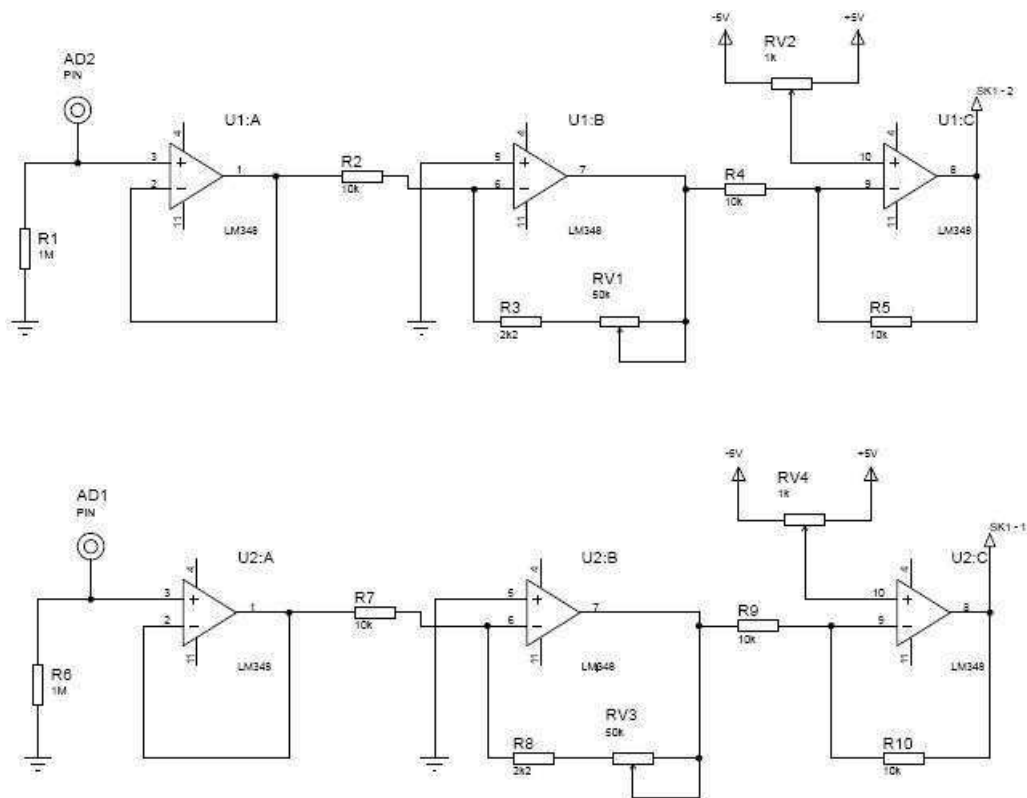


Figura A.1: Diagrama Elétrico do AD1 e AD2 na Interface de Controle Analógica

Agradecimentos

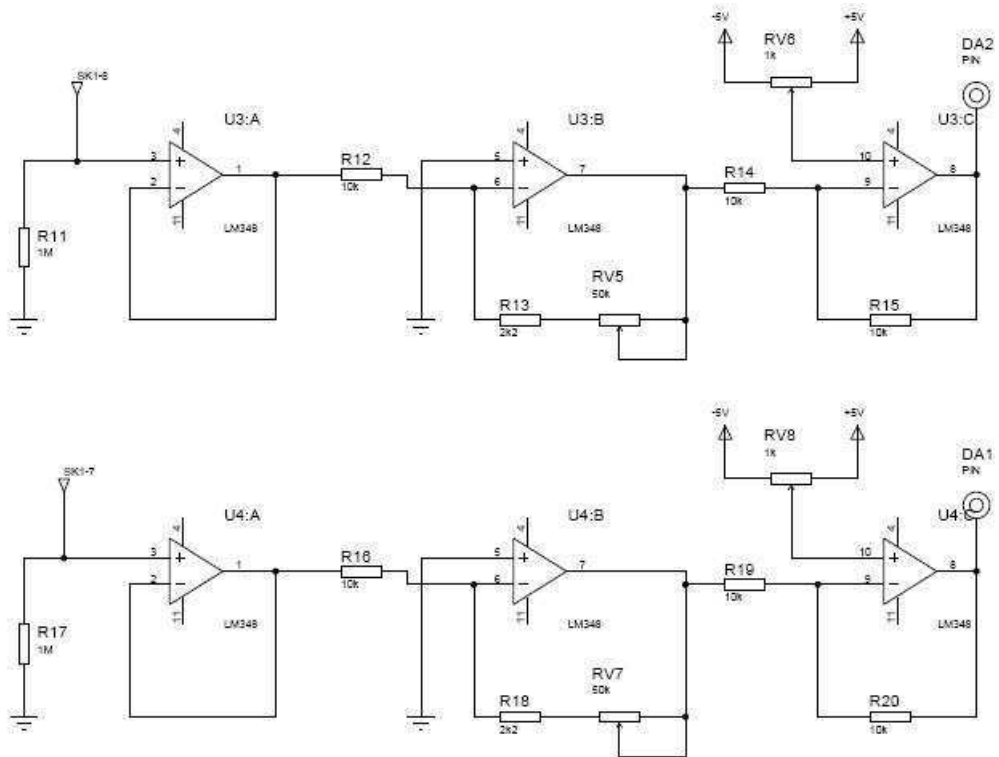


Figura A.2: Diagrama Elétrico do DA1 e DA2 na Interface de Controle Analógica

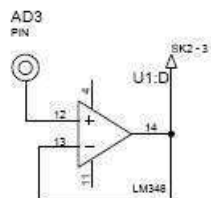


Figura A.3: Diagrama Elétrico do AD3 na Interface de Controle Analógica, AD4, AD5, AD6 e AD7 também são buffers com a mesma estrutura de AD3

Agradecimentos

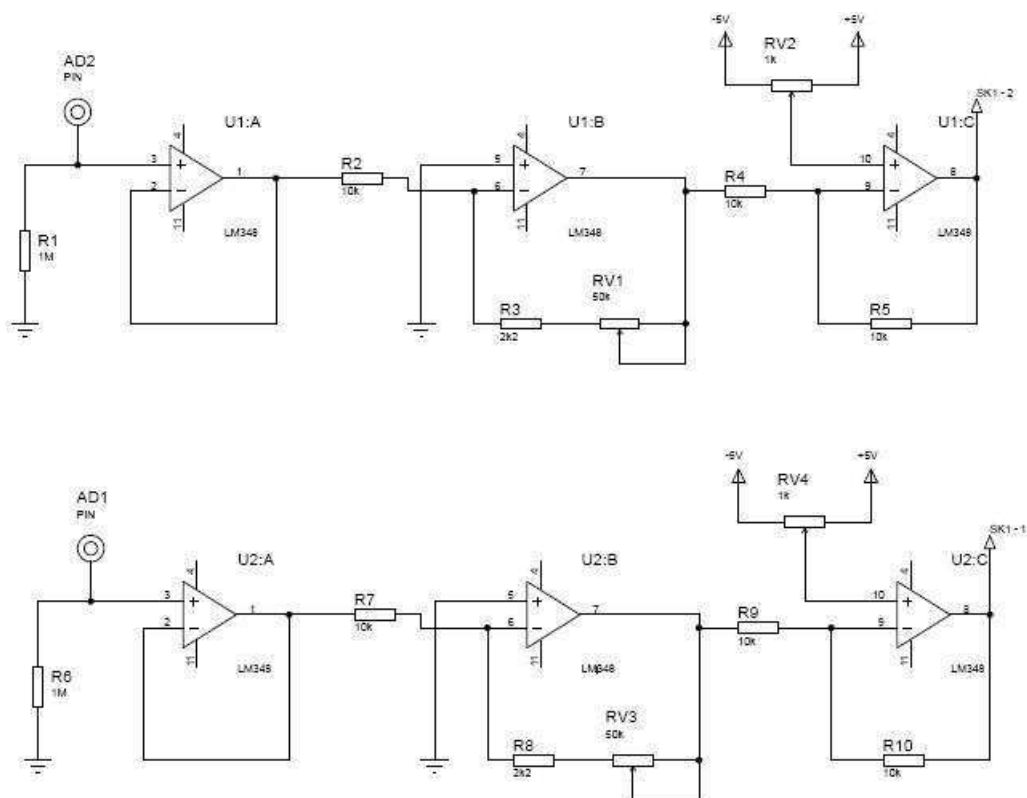


Figura A.4: Diagrama Elétrico do AD1 e AD2 na Interface de Controle Analógica

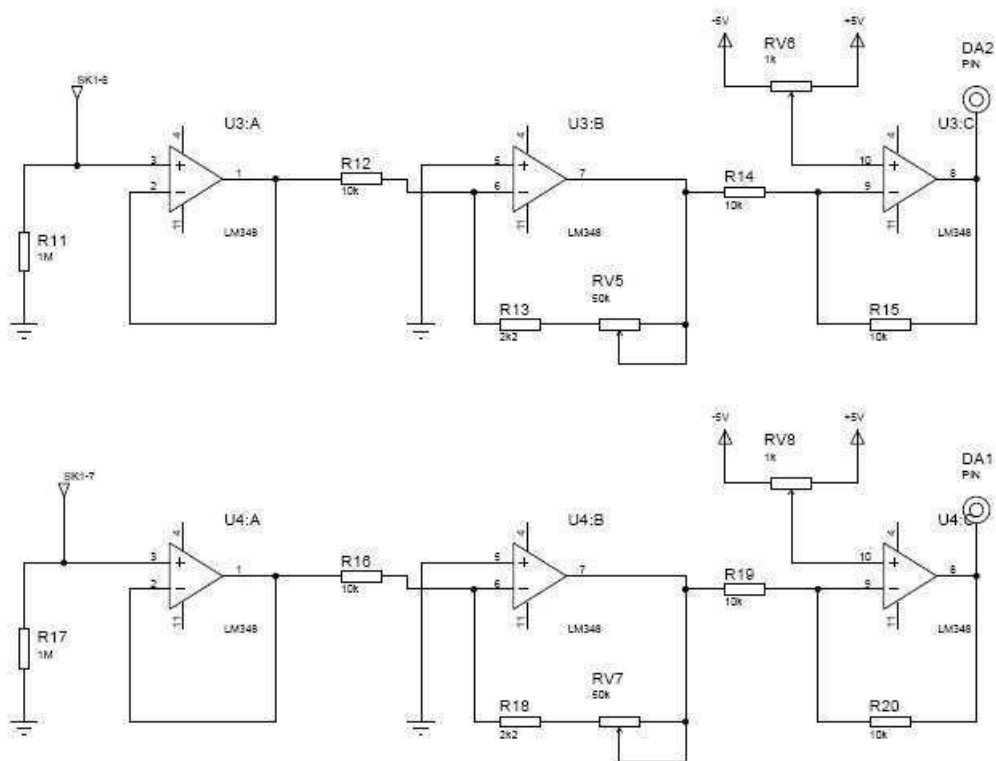


Figura A.5: Diagrama Elétrico do DA1 e DA2 na Interface de Controle Analógica

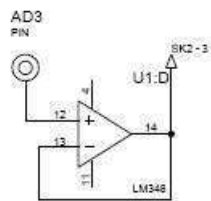


Figura A.6: Diagrama Elétrico do AD3 na Interface de Controle Analógica, AD4, AD5, AD6 e AD7 também são buffers com a mesma estrutura de AD3