

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
Campus DIVINÓPOLIS
GRADUAÇÃO EM ENGENHARIA MECATRÔNICA

Michelle Ferreira de Faria Castro

SÍNTESE DE CONTROLADORES ROBUSTOS POR REALIMENTAÇÃO DE SAÍDA PARA SISTEMAS
INCERTOS DISCRETOS NO TEMPO



Divinópolis
2013

Michelle Ferreira de Faria Castro

SÍNTESE DE CONTROLADORES ROBUSTOS POR REALIMENTAÇÃO DE SAÍDA PARA SISTEMAS
INCERTOS DISCRETOS NO TEMPO

Relatório Final de Trabalho de Conclusão de Curso
em Engenharia Mecatrônica como parte dos requi-
sitos exigidos para a obtenção do título de Enge-
nheiro Mecatrônico.

Eixos de Formação: Modelagem e Controle de Pro-
cessos e Circuitos Elétricos e Eletrônicos.

Orientador: Valter Júnior de Souza Leite

Coorientador: João Carlos de Oliveira



Divinópolis
2013

Michelle Ferreira de Faria Castro

CEFET/MG *Câmpus* Divinópolis

SÍNTESE DE CONTROLADORES ROBUSTOS POR REALIMENTAÇÃO DE SAÍDA PARA SISTEMAS
INCERTOS DISCRETOS NO TEMPO

Monografia de Trabalho de Conclusão de Curso apresentada ao Colegiado de Graduação em Engenharia Mecatrônica como parte dos requisitos exigidos para a obtenção do título de Engenheiro Mecatrônico. Áreas de concentração: Modelagem e Controle de Processos e Circuitos Elétricos e Eletrônicos.

Banca Examinadora:

Prof. Dr. João Carlos de Oliveira
CEFET/MG *Câmpus* V

Prof. Dr. Wagner Custódio de Oliveira
CEFET/MG *Câmpus* V

Prof. M. Sc. Emerson de Sousa Costa
CEFET/MG *Câmpus* V

Divinópolis
2013

Agradecimentos

Agradeço primeiramente a Deus pela oportunidade.

Agradeço à minha mãe, pelo amor incondicional, apoio e incentivo.

Agradeço aos meus demais familiares pela torcida, força e oração, em especial ao meu avô Geraldo, que tanto se orgulhou e quis que eu concluísse com sucesso a minha graduação.

Agradeço ao meu namorado Leonardo pelo carinho, companheirismo, cumplicidade e ajuda para que esse trabalho pudesse ser realizado.

Agradeço às minhas amigas que estiveram comigo em tantos momentos dessa caminhada.

Agradeço aos meus colegas de sala pelo tempo que passamos juntos, principalmente àqueles que estiveram sempre presente.

Agradeço especialmente, ao professor Valter por toda confiança durante minha graduação, e por não ter poupado esforços para a orientação deste trabalho mesmo estando distante.

Agradeço à todos meus professores pelos ensinamentos passados, principalmente ao João Carlos pela co-orientação.

Enfim, agradeço à todos que de alguma forma contribuíram para minha formação.

Resumo

Nesse trabalho será abordado um estudo para projeto de um controlador por realimentação de saída para sistemas incertos discretos no tempo usando alocação de pólos. Também será mostrado o novo projeto eletrônico de uma placa móvel acionada por fluxo de ar já existente no CEFET - MG câmpus Divinópolis. O objetivo é controlar essa planta em um determinado ponto de operação, ou seja, projetar um controlador que faça com que a mesma fique na posição desejada. Aqui também será apresentada a modelagem de tal planta em um sistema incerto discreto no tempo, bem como o projeto de um controlador para esse caso. O projeto do controlador utiliza métodos baseados na otimização convexa, sendo os problemas de otimização expressos em termos de desigualdades matriciais lineares (LMIs, do inglês *Linear Matrix Inequalities*). O problema central na síntese do controlador consiste em determinar um ganho de realimentação estática de saída que assegure ao sistema em malha fechada uma alocação de autovalores em uma região circular interna ao disco de raio unitário do plano complexo. Para isso foi desenvolvido um algoritmo para relaxação do problema proposto, resolvendo-se em cada etapa um conjunto de LMIs.

Palavras-chave: Controlador. Realimentação de Saída. Sistema incerto discreto no tempo.

Abstract

This paper addresses the study and development of a feedback controller for uncertain and discrete time systems by the use of pole placement. In addition to this discussion, an electronic project of a control plant, consisting of a moving flap triggered by airflow, will also be shown. The objective of this project is to control the plant at a determined operation point, in other words, the objective is to design a controller that keeps the flap at a determined inclined position. Furthermore, the model of the plant in question in an uncertain discrete time system, as well as its controller will be presented. The controller design uses methods based on convex optimization, where optimization problems are expressed in Linear Matrix Inequalities(LMI). The main problem in the synthesis of this controller is determining the static feedback gain of the closed-loop system output so to ensure the allocation of eigenvalues in an internal circular region inside a unitary disk on the complex plane. Due to this problem an algorithm was developed to solve a number of LMI at each stage.

Key-words: Controller. Output feedback. Uncertain discrete-time system

Sumário

Lista de Figuras	xvi
Lista de Tabelas	xvii
Lista de Acrônimos e Notação	xix
1 Introdução Geral	1
1.1 Relevância do tema	1
1.2 Objetivos do Trabalho	2
1.3 Metodologia	3
2 Controle - Preliminares e Definições	5
2.1 Realimentação	5
2.1.1 Realimentação de Estados	5
2.1.2 Realimentação de Saída	6
2.1.3 Realimentação Dinâmica de Saída	7
2.1.4 Realimentação Estática de Saída	7
2.2 Controle robusto	7
2.3 Teoria de Lyapunov	8
2.3.1 Teorema da Estabilidade de Lyapunov	8
2.4 Descrição do problema	9
3 Controle - Estudos iniciais	13
3.1 Projeto de Compensadores	13
3.2 Projeto de Compensadores com Alocação Regional de Pólos	16
3.3 Projeto do K_o	20
4 Eletrônica - Desenvolvimento dos circuitos eletrônicos	27
4.1 Placa Acelerômetro	28
4.2 Fonte	28
4.3 Placa de Controle	32
4.4 Placa de Potência	35

4.5	Planta Final	37
5	Controle - Modelagem da Planta Real	39
5.1	Método da Resposta Complementar	39
5.2	Obtenção de um Modelo Usando o Método da Resposta Complementar	40
5.2.1	Modelo Médio	50
6	Controle - Obtenção do Sistema Discreto	53
6.1	Discretização do Sistema	53
6.2	Obtenção do Sistema Incerto Discreto no Tempo	54
7	Controle - Projeto e Simulação do Controlador	57
7.1	Projeto do controlador para o sistema precisamente conhecido	57
7.2	Projeto do controlador para o sistema incerto	58
7.3	Implementação	60
8	Conclusão	61
8.1	Perspectivas Futuras	61
A	Códigos	1
	Bibliografia	14

Lista de Figuras

2.1	Realimentação de estados	6
2.2	Realimentação de saída	7
3.1	Resposta ao degrau unitário	14
3.2	Resposta ao degrau unitário	15
3.3	Auto valores do sistema em malha fechada	15
3.4	Resposta ao degrau unitário	19
3.5	Auto valores do sistema em malha fechada	20
3.6	Resposta ao degrau unitário	21
3.7	Auto valores do sistema em malha fechada	22
3.8	Resposta ao degrau unitário	23
3.9	Auto valores do sistema em malha fechada	23
3.10	Raio mínimo (realimentação de saída) para $\sigma=0$	24
3.11	Raio máximo (realimentação de saída) para $\sigma=0$	25
3.12	Raio mínimo (realimentação de saída) para $\sigma=0.4$	25
3.13	Raio máximo (realimentação de saída) para $\sigma=0.4$	26
4.1	Placa Móvel Acionada por um Fluxo de Ar antes das modificações feitas	27
4.2	Circuito do Acelerômetro	28
4.3	Circuito da Fonte de Alimentação 1	29
4.4	Layout da Fonte de Alimentação 1	30
4.5	Foto da Fonte de Alimentação 1	30
4.6	Circuito da Fonte de Alimentação 2	31
4.7	Layout da Fonte de Alimentação 2	31
4.8	Foto da Fonte de Alimentação 2	32
4.9	Circuito da Placa de Controle	33
4.10	Layout da Placa de Controle	34
4.11	Foto da Placa de Controle	34
4.12	Circuito da Placa de Potência	35
4.13	Layout da Placa de Potência	36
4.14	Foto da Placa de Potência	36
4.15	Placa Móvel Acionada por um Fluxo de Ar após modificações	37

4.16	Placa Móvel Acionada por um Fluxo de Ar finalizada	37
5.1	Interface do programa desenvolvido em LabVIEW	41
5.2	Dados coletados	41
5.3	Regiões 1, 2, 3, 4, 5 e 6	43
5.4	Regiões 1, 2, 3, 4, 5 e 6 deslocadas	44
5.5	Regiões 1, 2, 3, 4, 5 e 6 deslocadas e normalizadas	46
5.6	Regiões 1, 2, 3, 4, 5 e 6 linearizadas	48
5.7	Comparação entre o modelo obtido e o sistema real - Regiões 1, 2, 3, 4, 5 e 6 . .	50
5.8	Comparação entre o modelo obtido e o sistema real	51
6.1	Diagrama do sistema com o hold e o controlador	53
7.1	Resposta do sistema modelado	57
7.2	Auto valores dos sistema em malha aberta e malha fechada	58
7.3	Resposta do sistema modelado	59
7.4	Auto valores dos sistema em malha aberta e malha fechada	59
7.5	Programa para implementação do controlador na planta real	60

Lista de Tabelas

4.1	Reguladores de tensão utilizados nas fontes	32
5.1	Ganhos das regiões de interesse	45
5.2	Atrasos das regiões de interesse	47
5.3	Constantes de tempo das regiões de interesse	49
5.4	Parâmetros médios	51
6.1	Combinações dos parâmetros τ , θ e k	55

Lista de Acrônimos e Notação

LMI	Linear Matrix Inequality (desigualdade matricial linear)
PWM	Pulse Width Modulation (modulação por largura de pulso)
TBJ	Transistor Bipolar de Junção
MOSFET	Metal Oxide Semiconductor Field Effect Transistor (transistor de efeito de campo metal - óxido - semicondutor)
\star	indica bloco simétrico nas LMIs
$P > 0$	indica que a matriz L é simétrica definida positiva
A	notação para matrizes (letras maiúsculas do alfabeto latino)
A'	('), pós-posto a um vetor ou matriz, indica a operação de transposição
\mathbf{I}	matriz identidade de dimensão apropriada
N	especialmente utilizada para denotar o número de vértices de um politopo
n	especialmente utilizada para representar a ordem uma matriz quadrada
α	especialmente utilizada para representar as incertezas de um sistema

Introdução Geral

1.1 Relevância do tema

Síntese de controladores robustos por realimentação estática ou dinâmica de saída é um dos mais importantes problemas em aberto na teoria de controle segundo Syrmos et al. [1997]. Tais controladores têm chamado a atenção de muitos pesquisadores nas últimas décadas. Isso se deve ao fato de que em muitos processos industriais, o vetor de estados pode não ser ou não estar completamente acessível, o que dificulta a utilização da realimentação por estados como mostrado em Mehdi et al. [2004]. Além disso, há a impossibilidade da instalação ou construção de alguns sensores e o seu custo elevado. A realimentação de estados até poderia ser feita, entretanto a planta teria que ser observável e teria que ser projetado um observador que estimasse os estados da planta.

Uma possível solução é o projeto de controladores por realimentação estática ou dinâmica de saída, que requer apenas a leitura das variáveis de saída da planta em tempo real como apresentado por Moreira [2009]. O projeto de controladores dinâmicos pode ser formulado como um problema de realimentação estática de saída, envolvendo o sistema aumentado. Uma abordagem utilizada para a realimentação de saída é baseada na construção de uma função de Lyapunov, resultando em um problema de otimização convexa com restrições matriciais, formulado como desigualdades matriciais lineares (LMIs, do inglês *Linear matrix inequality*).

A otimização convexa é uma classe especial de otimização matemática, que inclui mínimos quadrados e problemas de programação linear como citado em Boyd and Vandenberghe [2004]. É conhecido que mínimos quadrados e problemas de programação linear podem ser resolvidos numericamente de forma muito eficiente. Enquanto a matemática de otimização convexa tem sido estudada há cerca de um século, vários problemas relacionados com os desenvolvimentos recentes têm estimulado o interesse no novo tópico. O primeiro é que o método desenvolvido para resolver os problemas de programação linear, pode também ser usado para resolver os problemas de otimização convexa. O segundo desenvolvimento é a descoberta de que os problemas de otimização convexa são mais predominantes na prática do que se pensava anteriormente. Estes novos métodos permitem resolver novas classes de problemas de otimização convexos, como os programas semidefinidos e programas de otimização-cone de segunda ordem. Muitas aplicações foram encontradas em áreas como sistemas de controle automáticos, estimativa e processamento

de sinais, comunicações e redes, desenho de circuito eletrônico, análise de dados e modelagem, estatísticas, e finanças. Uma grande vantagem para o reconhecimento ou a formulação de um problema como um problema de otimização convexa é o fato de tal método ser muito confiável e eficiente numericamente. A otimização convexa é uma continuidade da evolução das técnicas de otimização numéricas, que utilizam fundamentalmente de ferramentas da Álgebra Linear, como por exemplo as técnicas de mínimos quadrados ou a programação linear.

É possível trabalhar com o sistema na forma de função de transferência ou na forma de espaço de estados. Entretanto, nesses dois casos é necessário obter uma descrição matemática do sistema a ser controlado (um modelo). É, em geral, impossível obter um modelo, sobretudo linear, exato do sistema a ser controlado. Essa impossibilidade advém de incertezas de medição de parâmetros ou mesmo do negligenciamento de dinâmicas presentes no sistema real, para simplificar a representação matemática desse sistema. Por isso, sempre haverá um grau de incerteza associado ao modelo usado para a síntese do controlador. Portanto, projeta-se um controlador para um modelo matemático e aplica-se esse controlador a um sistema real que, provavelmente, difere do modelo usado.

Os métodos de síntese que são estudados neste trabalho levam em conta possíveis incertezas que existem nos modelos em relação ao sistema real, assegurando que o controlador projetado estabiliza o sistema real, eventualmente garantindo também algum índice de desempenho, mediante um determinado conjunto de incertezas associado ao modelo/sistema.

Em geral os métodos de realimentação de saída focam apenas na questão da estabilização para a regulação, e não se aprofundam nem na inclusão de índices de desempenho nem na síntese para o seguimento de referência, ou seja, quando deseja-se que o sistema siga um determinado comportamento.

No CEFET-MG, campus V - Divinópolis, há uma Placa Móvel Acionada por um Fluxo de Ar. Tal planta foi projetada e construída pelos alunos Leonardo Farias Campos, Lucas Silva de Oliveira e Thiago Silveira Costa, sob a orientação do professor Valter Júnior Souza Leite, na disciplina optativa Sinais e Sistemas, ofertada no 7º período de Engenharia Mecatrônica dessa Instituição. A parte mecânica da planta está correta, entretanto a parte eletrônica apresenta algumas falhas, as quais estão fazendo com que o circuito queime frequentemente.

Tais falhas foram corrigidas nesse trabalho. O circuito em questão foi modificado, fazendo com que o sistema eletromecânico funcione adequadamente. Depois de feita a modelagem, será projetado um controlador utilizando a técnica estudada para a planta especificada.

1.2 Objetivos do Trabalho

1. Projetar compensadores: regulação, servomecanismo, alocação regional de pólos. Para isso serão programados algoritmos de otimização.
2. Integrar as metodologias de projeto citadas anteriormente para obter o controlador desejado.
3. Projetar e implementar circuitos controle, atuador e de alimentação do sistema eletromecânico escolhido.

4. Obter um modelo incerto para a Placa Móvel Acionada por um Fluxo de Ar, na forma de um politopo de matrizes.
5. Aplicar o controlador projetado, na planta especificada.

1.3 Metodologia

Primeiramente foi feita uma revisão bibliográfica, em periódicos de primeira linha da área de controle/ engenharia elétrica/ mecatrônica, pesquisando trabalhos relacionados à realimentação de saída. Foi utilizado o portal Scopus (www.scopus.com) para o levantamento dos trabalhos na área.

Em seguida, foram utilizadas ferramentas de Álgebra Linear para manipulação de matrizes: complemento de Schur, Lema de Finsler, Lema da Projeção, método de Lyapunov, dentre outros para desenvolver as condições na forma de desigualdades matriciais para a síntese dos controladores. Para isso foi feita uma programação usando o software MATLAB[®] e os toolboxes YALMIP (parser) e SeDuMi (solver).

Posteriormente foram testados a realimentação estática e dinâmica de saída com e sem alocação de pólos para diferentes topologias. Foram realizadas simulações computacionais com o auxílio do software MATLAB[®]. Após, foi desenvolvida a parte eletrônica da planta com as modificações e correções necessárias para que a mesma funcionasse adequadamente. Para isso foram colocadas duas fontes de alimentação independentes, uma para o circuito de acionamento do motor e outra para o controle do mesmo. O objetivo dessa separação das fontes foi o isolamento dos circuitos como forma de proteção.

Em sequência, com a planta funcionando devidamente, foi obtido um modelo incerto.

Controle - Preliminares e Definições

Ao ser modelado, um sistema pode ser escrito na forma de equações diferenciais (sistemas contínuos) ou a diferenças (sistemas discretos), função de transferência (aplicando Transformada de Laplace para o caso contínuo no tempo ou Transformada Z para sistemas discreto no tempo) ou no espaço de estados. Um sistema linear discreto no tempo é o sistema cujo o tempo é dividido em amostras discretas e é dado por:

$$\begin{aligned}x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k\end{aligned}\tag{2.1}$$

2.1 Realimentação

Realimentação é o nome dado ao procedimento através do qual um conjunto de sinais de um sistema (ou circuito) é transferida para a entrada deste mesmo sistema, com o objetivo de diminuir, amplificar ou controlar a saída de um sistema. Esse conjunto de sinais é em geral formado pela saída do sistema ou por seus estados. Assim, a realimentação pode ser de estados ou de saída.

2.1.1 Realimentação de Estados

A realimentação de estados de um sistema discreto no tempo, consiste em realimentar os estados x_k , de forma a gerar um sinal de entrada u_k , que produza o sinal de saída y_k desejado como mostrado na Figura 2.1.

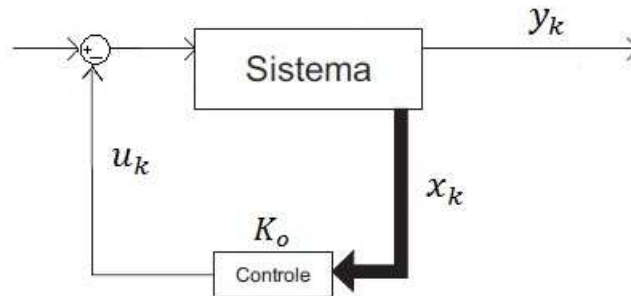


Figura 2.1: Realimentação de estados

O sinal de controle u_k por realimentação de estados é dado por:

$$u_k = K_o x_k \quad (2.2)$$

Para um sistema incerto no tempo, K_o pode ser projetado usando uma LMI adequada. Esse processo será descrito melhor no capítulo seguinte.

Fechando a malha, isto é, levando (2.2) em (2.1) encontra-se:

$$x_{k+1} = (A + BK_o)x_k \quad (2.3)$$

Entretanto, muitas vezes o vetor de estados não está disponível, pois em alguns casos os estados não podem ser mensurados, o que faz com que a realimentação de saída seja na prática uma alternativa mais viável.

2.1.2 Realimentação de Saída

A realimentação de saída consiste em encontrar um controlador K tal que a lei de controle

$$u_k = K y_k \quad (2.4)$$

estabilize o sistema (2.1), e adicionalmente satisfaça algumas especificações de desempenho. Como $y_k = C x_k$, a lei de controle (2.4) levada em (2.1) resulta:

$$x_{k+1} = (A + BKC)x_k \quad (2.5)$$

Ela é representada na Figura 2.2 e pode ser dinâmica ou estática.

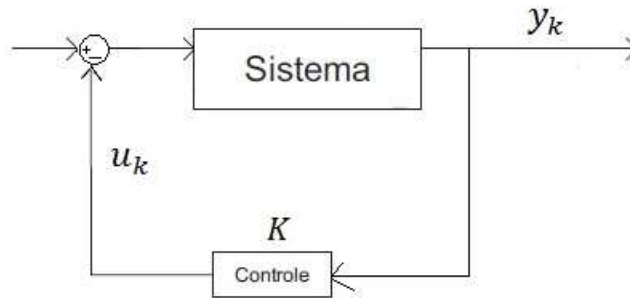


Figura 2.2: Realimentação de saída

2.1.3 Realimentação Dinâmica de Saída

Nesse caso, o controlador a ser encontrado, consiste em uma matriz de ganho K dada por:

$$K = \begin{bmatrix} A_c & B_c \\ C_c & D_c \end{bmatrix} \quad (2.6)$$

Para isso, o sistema deve ser aumentado para que as dimensões das matrizes sejam compatíveis. O controlador (2.6) também pode ser escrito na forma de função de transferência.

2.1.4 Realimentação Estática de Saída

Nesse caso, o controlador a ser encontrado, consiste em uma matriz simples de ganho K . Resolver a realimentação estática de saída implica em resolver a realimentação dinâmica de saída.

Nesse trabalho será usada a realimentação estática de saída. Para projetar o controlador K , será utilizado a Teoria de Lyapunov citada adiante.

2.2 Controle robusto

Sistemas incertos no tempo, são sistemas cujos parâmetros (matrizes A , B , C e D) não são precisamente conhecidos. Há várias formas de representar essa imprecisão. Duas formas bastante usadas são a representação por limitações em normas e a representação por politopos. Essa última é utilizada nesse trabalho. Na representação politópica, apenas os vértices da região são conhecidos. Portanto somente as matrizes que formam os vértices do politopo são conhecidas. O sistema incerto no tempo é representado por:

$$A = \sum_{i=1}^N \alpha_i A_i, \quad B = \sum_{i=1}^N \alpha_i B_i, \quad C = \sum_{i=1}^N \alpha_i C_i \quad (2.7)$$

com restrição que

$$\sum_{i=1}^N \alpha_i = 1$$

e as matrizes A_i , B_i e C_i , para $i = 1, \dots, N$ são conhecidas e denotados pelos vértices do politopo. Pode-se definir o vetor α :

$$\alpha = [\alpha_1 \quad \dots \quad \alpha_N]^T \quad (2.8)$$

e a equação (2.8) deve ser maior ou igual a zero.

Então as matrizes A , B e C podem ser reescritas como $A(\alpha)$, $B(\alpha)$ e $C(\alpha)$ quando necessário.

Várias dificuldades surgem ao se trabalhar com sistemas incertos. Uma das principais é como modelar e descrever as incertezas no problema como apresentada em Trofino [2000]. Incertezas descritas de forma genérica podem acarretar restrições na busca de soluções. A este processo de busca de solução de um problema de controle envolvendo o sistema nominal e uma família de incertezas em torno dele, chama-se de Controle Robusto. No Controle Robusto ao encontrar o ganho do controlador busca-se também minimizar o efeito sobre certas variáveis do sistema devido a perturbações externas a este.

2.3 Teoria de Lyapunov

Uma forma de garantir a estabilidade de um sistema é utilizando a técnica de Lyapunov.

2.3.1 Teorema da Estabilidade de Lyapunov

Seja o sistema (2.9).

$$x_{k+1} = f(x(k)) \quad (2.9)$$

$$x(0) = x_o$$

Se existe para o sistema (2.9) uma função definida positiva

$$V(x_1, x_2, \dots, x_n) \quad (V > 0)$$

e

$$V(0) = 0$$

sendo 0 o ponto de equilíbrio e cuja a variação (caso discreto no tempo) da função

$$\Delta V(x_k) = V(x_{k+1}) - V(x_k) < 0$$

deve ser negativa

$$V' \leq 0$$

então a solução é assintoticamente estável.

Então, definindo

$$V_{x_k} = x_k^T P x_k$$

Tem-se

$$\Delta V_{x_k} = x_{k+1}^T P x_{k+1} - x_k^T P x_k < 0$$

usando

$$x_{k+1} = A x_k$$

obtem-se

$$\Delta V_{x_k} = x_k^T A^T P A x_k - x_k^T P x_k = x_k^T (A^T P A - P) x_k < 0$$

que é verificado se e somente se

$$A^T P A - P < 0 \quad (2.10)$$

$$P > 0 \quad (2.11)$$

em que A é substituído pela malha fechada e P é uma matriz simétrica definida positiva com dimensões apropriadas a ser determinada.

Intuitivamente, a estabilidade de um sistema dinâmico está relacionada com a função “energia” associada a esse sistema. Se a função energia do sistema é sempre não negativa e estritamente decrescente com relação ao tempo, as trajetórias do sistema tendem ao ponto de equilíbrio, que para sistemas lineares pode ser considerado a origem, sem perda de generalidade. A técnica de Lyapunov proporciona encontrar uma matriz definida positiva usada para testar a estabilidade, bem como para calcular o ganho do controlador. A construção de uma função de Lyapunov pode ser obtida por um problema de otimização convexa formulado como uma LMI do Inglês “Linear Matrix Inequalities”, ou em Português Desigualdades Matriciais Lineares.

2.4 Descrição do problema

São utilizados os seguintes lemas para se descrever o problema de realimentação de saída.

1. • Existe uma matriz P simétrica definida positiva e um ganho K_o de realimentação de estados tais que:

$$(A + B K_o)^T P (A + B K_o) - P < 0 \quad (2.12)$$

- Existe uma matriz P simétrica definida positiva e um ganho K de realimentação de saída tal que:

$$(A + B K C)^T P (A + B K C) - P < 0 \quad (2.13)$$

- Existe uma matriz P simétrica definida positiva, uma matriz não singular G , um ganho K_o de realimentação de estados, e um ganho K de realimentação de saída tal que:

$$\begin{bmatrix} -P & A_o^T P & S^T G^T \\ PA_o & -P & PB \\ GS & B^T & -(G + G^T) \end{bmatrix} < 0 \quad (2.14)$$

em que A_o é dado por (2.16) e S é dado por :

$$S = KC - K_o \quad (2.15)$$

Fazendo-se a realimentação de estados cujo a lei de controle é dada por (2.2), e a malha fechada é representada por (2.3). Logo, define-se:

$$A_o = (A + BK_o) \quad (2.16)$$

Substituindo a malha fechada (2.16) em (2.10) tem-se (2.12).

Fazendo a realimentação de saída cuja a lei de controle é dada por (2.4), e a malha fechada é representada por (2.5). Define-se:

$$A_{cl} = (A + BKC) \quad (2.17)$$

Substituindo a malha fechada (2.17) em (2.10) tem-se (2.13).

As equações (2.12) e (2.13) podem ser reescritas usando Complemento de Schur, o qual é definido pelo seguinte lema.

- Seja uma matriz $M \in \mathbb{R}^{n \times n}$ simétrica particionada:

$$M = \begin{bmatrix} A & B \\ B^T & D \end{bmatrix} > 0 \quad (2.18)$$

com $A = A^T$, e $C = C^T$, se $\det(A) \neq 0$, a matriz $C - B^T A^{-1} B$ é o complemento de Schur de M em relação a A , e M é equivalente a: $A > 0$, $C - B^T A^{-1} B > 0$. Analogamente, se $\det(C) \neq 0$, a matriz $A - B^T C^{-1} B$ é o complemento de Schur de M em relação a C , e M é equivalente a: $C > 0$, $A - B^T C^{-1} B > 0$.

Dessa forma as equações (2.12) e (2.13) podem ser reescritas respectivamente nas formas (2.19) e (2.20).

$$\begin{bmatrix} P & (A + BK_o)^T P \\ \star & P \end{bmatrix} > 0 \quad (2.19)$$

$$\begin{bmatrix} P & (A + BKC)^T P \\ \star & P \end{bmatrix} > 0 \quad (2.20)$$

em que \star representa o termo simétrico em relação à diagonal principal.

2. • Existe uma matriz P simétrica definida positiva, uma matriz F_2 não nula, duas matrizes G e F_4 não singulares e duas matrizes arbitrárias F_1 e F_3 , tais que:

$$\begin{bmatrix} -P & 0 & S^T G^T & 0 \\ 0 & -P & 0 & P \\ GS & 0 & -(G + G^T) & 0 \\ 0 & P & 0 & 0 \end{bmatrix} + \text{sym} \left\{ \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} [A_o \ 0 \ B \ -I] \right\} < 0 \quad (2.21)$$

- Existe uma matriz P simétrica definida positiva e uma matriz G não singular tal que:

$$\begin{bmatrix} -P & A_o^T & S^T G^T \\ PA_o & -P & PB \\ GS & B^T P & -(G + G^T) \end{bmatrix} \quad (2.22)$$

para dadas matrizes A_o , B e S .

O seguinte teorema é estabelecido:

- Um ganho K de realimentação estática de saída torna o sistema estável se e somente se existir uma matriz definida positiva $P = P^T > 0$ em $\mathbb{R}^{n \times n}$, matrizes não singulares $G \in \mathbb{R}^{m \times m}$ e $F_4 \in \mathbb{R}^{n \times n}$, matrizes não nulas $F_1 \in \mathbb{R}^{n \times n}$ e $L \in \mathbb{R}^{m \times p}$, e matrizes arbitrárias $F_2 \in \mathbb{R}^{n \times n}$, $F_3 \in \mathbb{R}^{m \times n}$, tal que a LMI seguinte:

$$\begin{bmatrix} F_1 A_o + A_o^T F_1^T - P & \star & \star & \star \\ F_2 A_o & -P & \star & \star \\ F_3 A_o + B^T F_1^T + (LC - GK_o) & B^T F_2^T & F_3 B + B^T F_3^T - (G + G^T) & \star \\ F_4 A_o - F_1^T & P - F_2^T & F_4 B - F_3^T & -F_4 - F_4^T \end{bmatrix} < 0 \quad (2.23)$$

O ganho K de realimentação de saída, para um dado ganho K_o de realimentação de estados que estabiliza o par (A, B) , é dado por:

$$K = G^{-1} L \quad (2.24)$$

Controle - Estudos iniciais

3.1 Projeto de Compensadores

Baseado na LMI (2.23) foi feito um algoritmo usando o software MATLAB[®] e os toolboxes YALMIP Löfberg [2004] (parser) e SeDuMi Sturm [1999] (solver). O algoritmo trata-se de um programa principal que passava os parâmetros do sistema como argumento para uma função que resolve a LMI a fim de encontrar um controlador K tal que a lei de controle (2.4) estabilizasse um dado sistema. É necessário fazer um algoritmo para esse caso, por se tratar de um problema de otimização. Dessa forma, deve-se sempre encontrar a melhor solução, ou seja, a solução ótima, para cada caso. Foi feito também uma função que plota o círculo com o raio r e centro σ .

Para os testes iniciais foi utilizado o sistema modelo (2.1) com matrizes:

$$A = \begin{bmatrix} 0.50 & 0.00 & 0.20 & 1.00 \\ 0.00 & -0.30 & 0.00 & 0.10 \\ 0.01 & 0.10 & -0.50 & 0.00 \\ 0.10 & 0.00 & -0.10 & -1.00 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ -1 & 0 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix} \quad (3.1)$$

O sistema (3.1), usado em Mehdi et al. [2004], que por sua vez foi baseado em Garcia et al. [2001], que é instável em malha aberta, quando aplicado um degrau unitário, tem o comportamento mostrado na Figura 3.1.

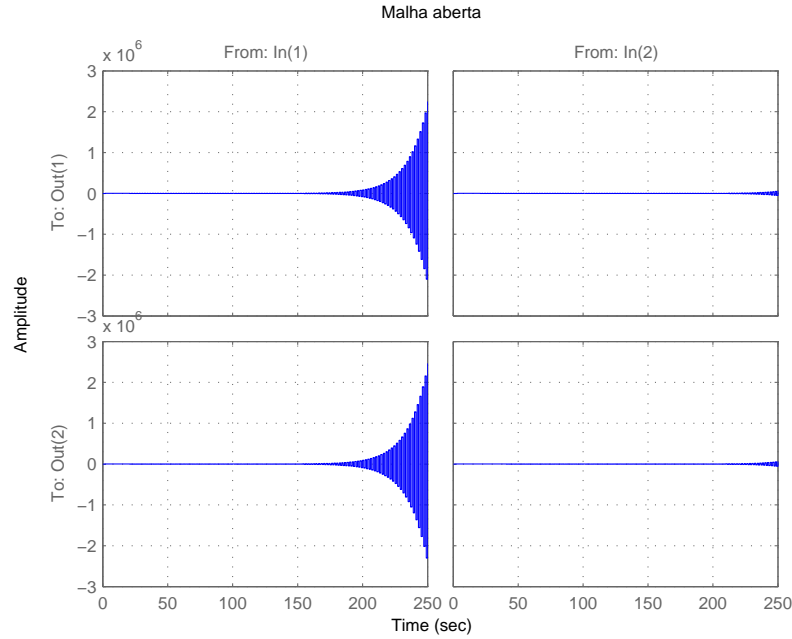


Figura 3.1: Resposta ao degrau unitário

Essa instabilidade é justificada pelos auto valores da matriz A dados por:

$$\begin{bmatrix} 0.5653 & -1.0677 & -0.3079 & -0.4897 \end{bmatrix} \quad (3.2)$$

Para um sistema discreto no tempo ser estável, o módulo de seus auto valores devem estar dentro de um círculo de raio unitário. Percebe-se que o módulo do segundo auto valor é maior do que um, portanto está fora desse círculo. Logo, esse auto valor é responsável pela instabilidade do sistema em malha aberta.

Em seguida foi feita a realimentação de de saída. Para isso, também foi utilizado o K_o de realimentação de estados (3.3), dado em Mehdi et al. [2004].

$$K_o = \begin{bmatrix} 0.0318 & -0.0190 & -0.0024 & -1.1667 \\ 0.0608 & 1.5014 & -6.6183 & -0.0834 \end{bmatrix} \quad (3.3)$$

Com (3.1) e (3.3) foi resolvida a LMI (2.23), obtendo um K de realimentação de saída dado por (3.4).

$$K = \begin{bmatrix} -0.4688 & -0.1948 \\ -0.5373 & 0.3918 \end{bmatrix} \quad (3.4)$$

A resposta ao degrau da malha fechada utilizando o controlador (3.4) projetado por realimentação de saída é mostrada na Figura 3.2.

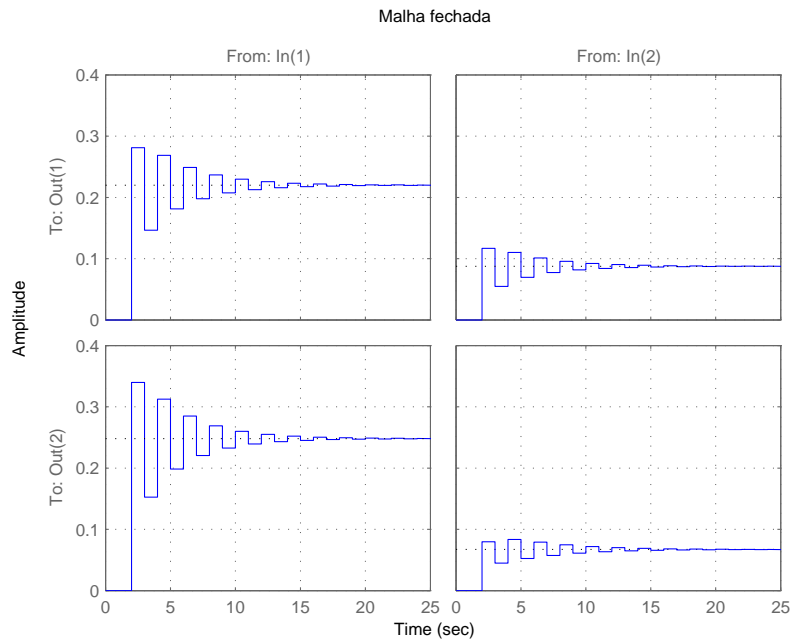


Figura 3.2: Resposta ao degrau unitário

Pode ser observado pela resposta ao degrau unitário que o sistema em malha fechada é estável. Esse fato é comprovado analisando os auto valores do sistema em malha fechada, já que esses estão contidos dentro do círculo unitário com centro em zero como mostra a Figura 3.3.

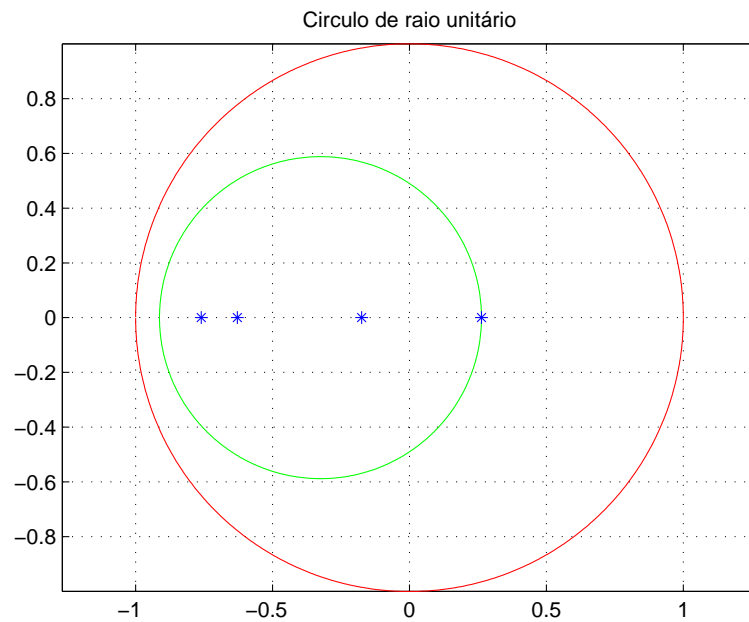


Figura 3.3: Auto valores do sistema em malha fechada

O círculo maior, possui raio unitário, dentro do qual os auto valores são estáveis. Já o círculo menor é o que contém todos os auto valores do sistema. Quanto menor esse último, ou seja,

quanto mais próximo os auto valores tiverem uns dos outros, menor será o raio do círculos que os contém, o que significa que mais estável é o sistema. Portanto deseja-se projetar controladores que deixam o sistema em malha fechada o mais estável possível. Para isso utiliza-se a técnica de alocação de pólos, pois dessa maneira pode-se alocar os pólos no lugar desejado, fazendo com que o sistema tenha uma maior estabilidade.

3.2 Projeto de Compensadores com Alocação Regional de Pólos

Em Mehdi et al. [2004], o autor faz apenas a estabilização, sem se preocupar com índices de desempenho. Há vários tipos de índices de desempenho, como por exemplo overshoot, erro em regime permanente, alocação de pólos, dentre outros. Nesse trabalho será usada a alocação de pólos, ou seja, será especificado onde deseja-se alocar os pólos do sistema em malha fechada. Para tal, foi usada a idéia da construção da LMI (2.23) do artigo Mehdi et al. [2004], e então desenvolvida uma nova LMI fazendo alocação de pólos representada por:

$$A_o = \frac{A + BK_o + \sigma I}{r} \quad (3.5)$$

$$A_{cl} = \frac{A + BKC + \sigma I}{r} \quad (3.6)$$

Em que A_o e A_{cl} representam respectivamente a malha fechada para a realimentação de estados e para a realimentação de saída. Substituindo (3.7) e (3.8) em (2.10), tem-se um teste para verificar se os pólos do sistema ficam alocados dentro de um círculo de raio r e centro em $(\sigma, 0)$.

As condições obtidas são:

Para a realimentação de estados:

$$\left(\frac{A + BK_o + \sigma I}{r} \right)^T P \left(\frac{A + BK_o + \sigma I}{r} \right) - P < 0 \quad (3.7)$$

$$P > 0$$

Para a realimentação de saída:

$$\left(\frac{A + BKC + \sigma I}{r} \right)^T P \left(\frac{A + BKC + \sigma I}{r} \right) - P < 0 \quad (3.8)$$

$$P > 0$$

Com a alocação de pólos a equação (2.14) torna-se:

$$\begin{bmatrix} -P & \left(\frac{A+BK_o+\sigma I}{r} \right)^T P & S^T G^T \\ P \left(\frac{A+BK_o+\sigma I}{r} \right) & -P & PB \\ GS & B^T & -(G + G^T) \end{bmatrix} < 0 \quad (3.9)$$

Usando Complemento de Schur, as equações (3.7) e (3.8) podem respectivamente ser reescritas como:

$$\begin{bmatrix} -P & \left(\frac{A+BK_o+\sigma I}{r}\right)^T P \\ \star & -P \end{bmatrix} < 0 \quad (3.10)$$

$$\begin{bmatrix} -P & \left(\frac{A+BKC+\sigma I}{r}\right)^T P \\ \star & -P \end{bmatrix} < 0 \quad (3.11)$$

Podemos reescrever (3.11) como:

$$\begin{bmatrix} I & 0 & S^T \\ 0 & I & 0 \end{bmatrix} \begin{bmatrix} -P & A_o^T P & 0 \\ PA_o & -P & PB \\ 0 & B^T P & 0 \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & I \\ S & 0 \end{bmatrix} = \begin{bmatrix} -P & \left(\frac{A+BKC+\sigma I}{r}\right)^T P \\ \star & -P \end{bmatrix} \quad (3.12)$$

Resolvendo (3.12), tem-se:

$$\begin{bmatrix} -P & A_o^T P + S^T B^T P & 0 \\ PA_o & -P & PB \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & I \\ S & 0 \end{bmatrix} = \begin{bmatrix} -P & \left(\frac{A+BKC+\sigma I}{r}\right)^T P \\ \star & -P \end{bmatrix}$$

$$\begin{bmatrix} -P & A_o^T + S^T B^T P \\ \star & -P \end{bmatrix} = \begin{bmatrix} -P & \left(\frac{A+BKC+\sigma I}{r}\right)^T P \\ \star & -P \end{bmatrix} \quad (3.13)$$

Igualando o bloco (2,1) (segunda linha e primeira coluna) das duas matrizes de (3.13), pode-se encontrar a matriz S para a alocação de pólos.

$$P^T A_o + P^T B S = P^T \left(\frac{A + BKC + \sigma I}{r} \right) \quad (3.14)$$

Substituindo A_o por (3.5), tem-se:

$$P^T \left(\frac{A + BK_o + \sigma I}{r} \right) + P^T B S = P^T \left(\frac{A + BKC + \sigma I}{r} \right) \quad (3.15)$$

Colocou-se a matriz P em evidência e resolveu-se a equação.

$$P^T \left(\frac{A + BK_o + \sigma I}{r} + B S \right) = P^T \left(\frac{A + BKC + \sigma I}{r} \right) \quad (3.16)$$

Igualou-se o termo dentro dos parênteses que multiplicam a matriz P :

$$\frac{A + BK_o + \sigma I}{r} + B S = \frac{A + BKC + \sigma I}{r} \quad (3.17)$$

Multiplicou-se ambos os lados por r , para eliminar o denominador e a equação foi desenvolvida:

$$A + BK_o + \sigma I + B S r = A + BKC + \sigma I$$

$$BK_o + B S r = BKC$$

$$BSr = BKC - BK_o$$

$$BSr = B(KC - K_o) \quad (3.18)$$

Pré multiplicou-se todos os termos por (B^\dagger) , como a matriz B não é quadrada, logo não tem inversa, foi utilizada a pseudo inversa da matriz B .

$$B^\dagger BSr = B^\dagger B(KC - K_o)$$

Pode ser feito sem a inversa de forma mais geral: comparando os dois lados de (3.18), tem-se

$$Sr = KC - K_o$$

Portanto a matriz S para o projeto de compensadores com alocação de pólos é:

$$S = \frac{KC - K_o}{r} \quad (3.19)$$

Substituindo (3.5) e (3.19) em (2.21), encontra-se:

$$\begin{bmatrix} F_1A_o + A_o^T F_1^T - P & \star & \star & \star \\ F_2A_o & -P & \star & \star \\ F_3A_o + B^T F_1^T + \frac{LC - GK_o}{r} & B^T F_2^T & F_3B + B^T F_3^T - (G + G^T) & \star \\ F_4A_o - F_1^T & P - F_2^T & F_4B - F_3^T & -F_4 - F_4^T \end{bmatrix} < 0 \quad (3.20)$$

que provê uma formulação que assegura a alocação regional de pólos.

O ganho K de realimentação de saída para um dado ganho K_o de realimentação de estados que estabiliza o par (A, B) é dado pela equação 2.24 do caso sem alocação de pólos.

Foi então programado usando como base o programa do caso anterior, alterando apenas a *LMI* (3.20). Dessa forma foi alterado apenas a função que resolvia a *LMI*.

Para $\sigma = 0.01$ foram testados vários valores de r , de modo a encontrar o r mínimo que resultasse em um controlador k factível. O r mínimo encontrado foi $r_{min} = 0.78$ e o controlador k correspondente é dado por (3.21).

$$K = \begin{bmatrix} -1.4594 & -0.7510 \\ 13.0750 & 10.2332 \end{bmatrix} \quad (3.21)$$

Para esse controlador, o raio que contém os autovalores da malha fechada é $r = 0.4708$.

A resposta ao degrau da malha fechada utilizando o controlador (3.21) projetado por realimentação de saída é mostrada na Figura (3.4).

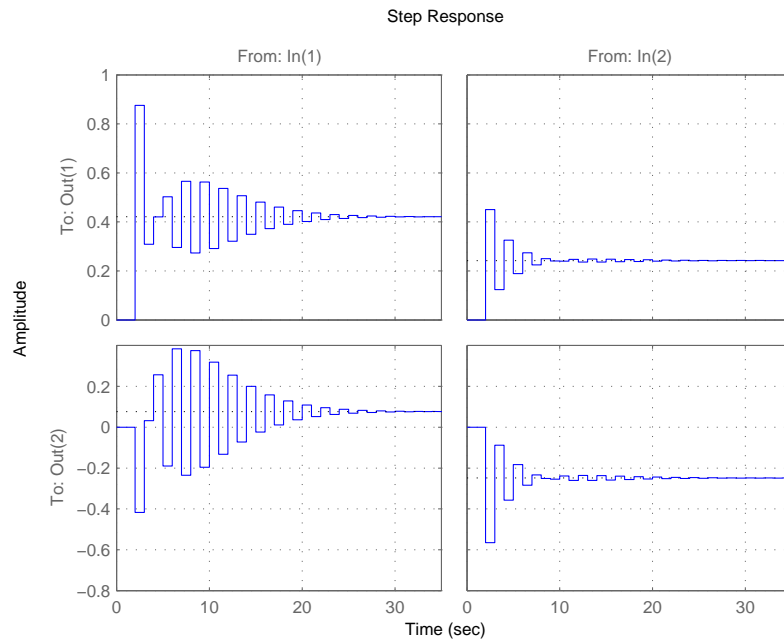


Figura 3.4: Resposta ao degrau unitário

Pode ser observado pela resposta ao degrau unitário que o sistema em malha fechada é estável. Um fator que determina a velocidade com que o sistema estabiliza é onde seus pólos foram alocados. Quanto mais próximo um do outro os pólos estiverem, mais rápido será o sistema. A estabilidade é comprovada analisando os auto valores do sistema em malha fechada, já que esses estão contidos dentro do círculo unitário com centro em zero como mostra a Figura 3.5.

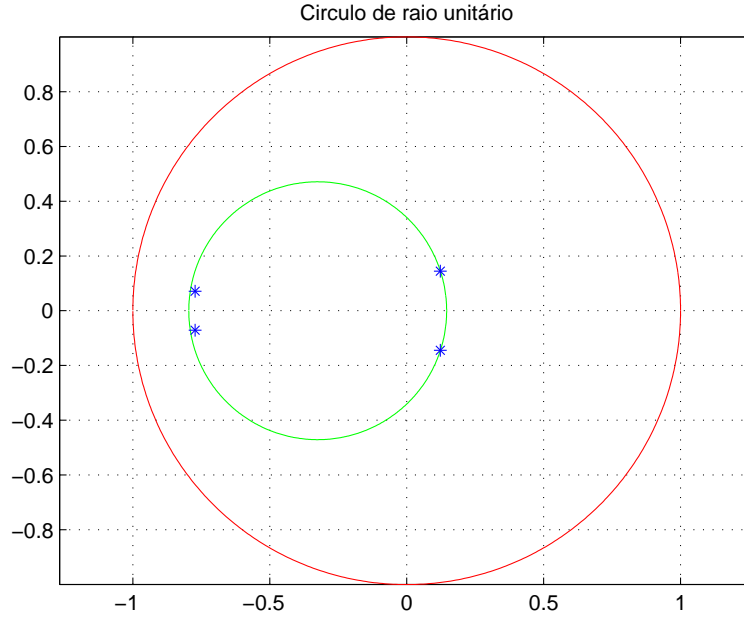


Figura 3.5: Auto valores do sistema em malha fechada

3.3 Projeto do K_o

Para os primeiros testes foi utilizado o K_o (3.3) dado no artigo Mehdi et al. [2004]. Entretanto para se obter melhores resultados, é viável que o K_o por realimentação de estados também seja projetado. Para o projeto de K_o foi utilizada a equação de Lyapunov (2.10), em que a matriz A é substituída pela matriz A_o de malha fechada por realimentação de estados representada pela equação (3.5). Usando o Lema de Schur, a equação (2.10) pode ser reescrita na forma (2.12), como já mencionado. Dessa forma a matriz (2.12) é pré e pós multiplicada por:

$$\begin{bmatrix} P^{-1} & 0 \\ 0 & P^{-1} \end{bmatrix} \quad (3.22)$$

Dessa forma, tem-se:

$$\begin{bmatrix} P^{-1} & 0 \\ 0 & P^{-1} \end{bmatrix} \begin{bmatrix} -P & A_o^T P \\ A_o P & -P \end{bmatrix} \begin{bmatrix} P^{-1} & 0 \\ 0 & P^{-1} \end{bmatrix} < 0 \quad (3.23)$$

$$\begin{bmatrix} P^{-1}(-P) & P^{-1}A_o^T P \\ P^{-1}A_o P & P^{-1}(-P) \end{bmatrix} \begin{bmatrix} P^{-1} & 0 \\ 0 & P^{-1} \end{bmatrix} < 0$$

$$\begin{bmatrix} -P^{-1} & P^{-1}A_o^T \\ A_o P^{-1} & -P^{-1} \end{bmatrix} < 0 \quad (3.24)$$

Definindo $W = P^{-1}$, e substituindo A_o por (3.5), obtém-se

$$\begin{bmatrix} -W & W(\frac{A+BK_o+\sigma I}{r})^T \\ (\frac{A+BK_o+\sigma I}{r})W & -W \end{bmatrix} < 0 \quad (3.25)$$

$$\begin{bmatrix} -W & \frac{WA^T}{r} + \frac{WK_o^T B^T}{r} + \frac{W\sigma}{r} \\ \star & -W \end{bmatrix} < 0$$

Definindo $Z_o^T = WK_o^T$ encontra-se a LMI final para o cálculo de K_o .

$$\begin{bmatrix} -W & \frac{WA^T}{r} + \frac{WZ_o^T}{r} + \frac{W\sigma}{r} \\ \star & -W \end{bmatrix} < 0 \quad (3.26)$$

Em que

$$K_o = Z_o W^{-1} \quad (3.27)$$

Foi então programado a *LMI* de modo a encontrar o controlador de realimentação de estados, dado agora por (3.27).

O controlador K por realimentação de saída encontrado programando (3.20) e (3.26) é dado por (3.28).

$$K = \begin{bmatrix} -0.4918 & -0.1342 \\ -1.6751 & 1.8321 \end{bmatrix} \quad (3.28)$$

Foi então feita a simulação para o controlador encontrado (3.28) e a *LMI* de alocação de pólos (2.23). Para isso foram utilizados para calcular k_o de realimentação de estados $\sigma_o = -0.3250$ e r_o . Nele foi utilizado o método da Bisecção para testar os valores de diferentes raios de modo a economizar tempo. A função que calcular a *LMI* é a mesma do caso anterior.

A resposta ao degrau da malha fechada utilizando o controlador (3.28) projetado por realimentação de saída é mostrada na Figura (3.6).

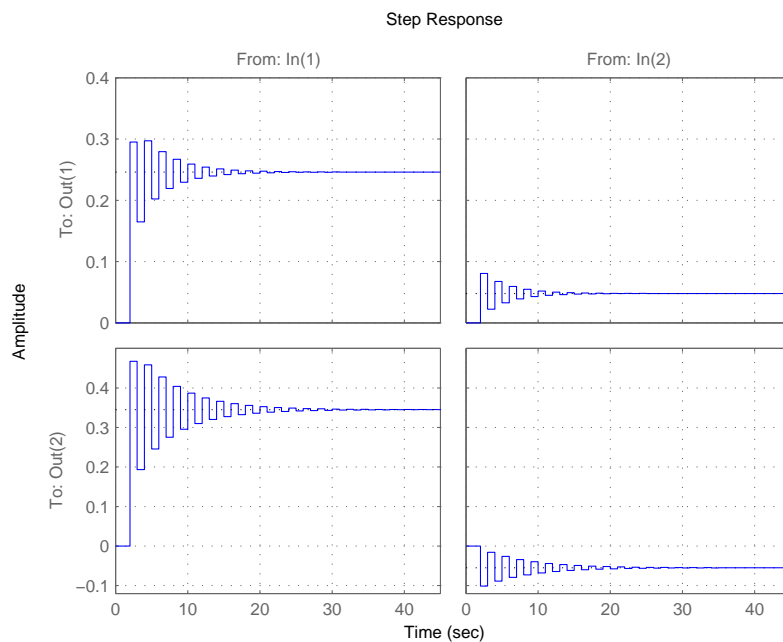


Figura 3.6: Resposta ao degrau unitário

Pode ser observado pela resposta ao degrau unitário que o sistema em malha fechada é estável. Esse fato é comprovado analisando os auto valores do sistema em malha fechada, já que esses estão contidos dentro do círculo unitário com centro em zero como mostra a Figura 3.7.

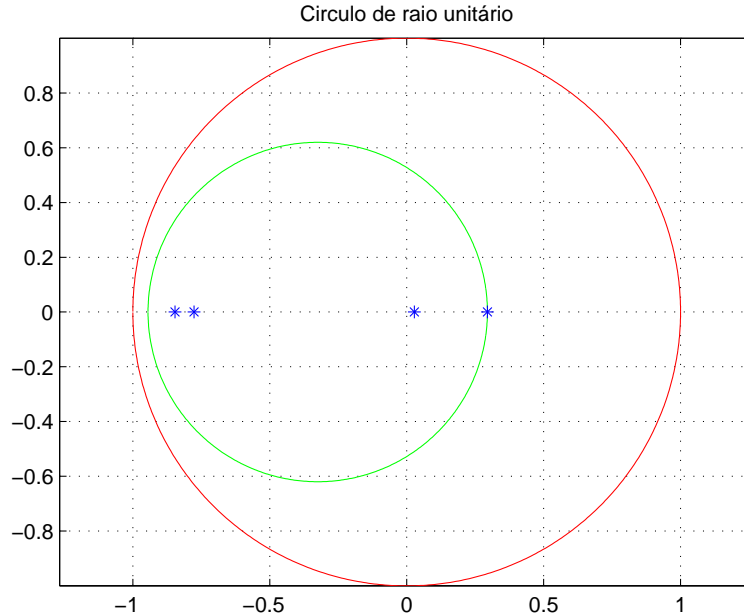


Figura 3.7: Auto valores do sistema em malha fechada

Portanto, para o seguimento desse trabalho foram utilizadas as LMIs (3.26) e (3.20) para encontrar K_o e K respectivamente.

Há uma grande dificuldade em encontrar-se K_o e K que faça com que o sistema tenha um bom desempenho. Portanto, para um melhor resultado, foi feita uma análise de qual seria o melhor raio (r_1) e o melhor centro ($\sigma_1, 0$) para resolver a realimentação de estados, de forma a obter um melhor raio (r_2) e o melhor centro ($\sigma_2, 0$) para a realimentação de saída, ou seja, de modo que o raio do círculo que contivesse os auto valores do sistema em malha fechada, usando realimentação de saída, fosse o menor possível. Para isso primeiramente foi feito um algoritmo que calculava vários k_o e armazenava-os. Em seguida foi feito outro algoritmo que variava r_1 , σ_1 , r_2 e σ_2 para cada um dos k_o encontrado anteriormente. Dessa forma foram feitas todas as combinações e foi armazenada a combinação que obteve o menor círculo que contivesse os autovalores de malha fechada. O melhor resultado obtido foi para:

$$K_o = \begin{bmatrix} -1.8909 & -0.0137 & -0.2795 & -1.3967 \\ -0.0415 & -0.3039 & -0.0073 & -0.1210 \end{bmatrix}$$

O raio mínimo que contém os auto valores para o melhor caso é $r_{min} = 0.0753$, que é um valor pequeno como o desejado, e o controlador K obtido foi:

$$K = \begin{bmatrix} -0.0024 & -1.1790 \\ -0.4182 & 0.2956 \end{bmatrix}$$

A resposta ao degrau unitário e os autovalores são mostrados respectivamente nas figuras 3.8 e 3.9.

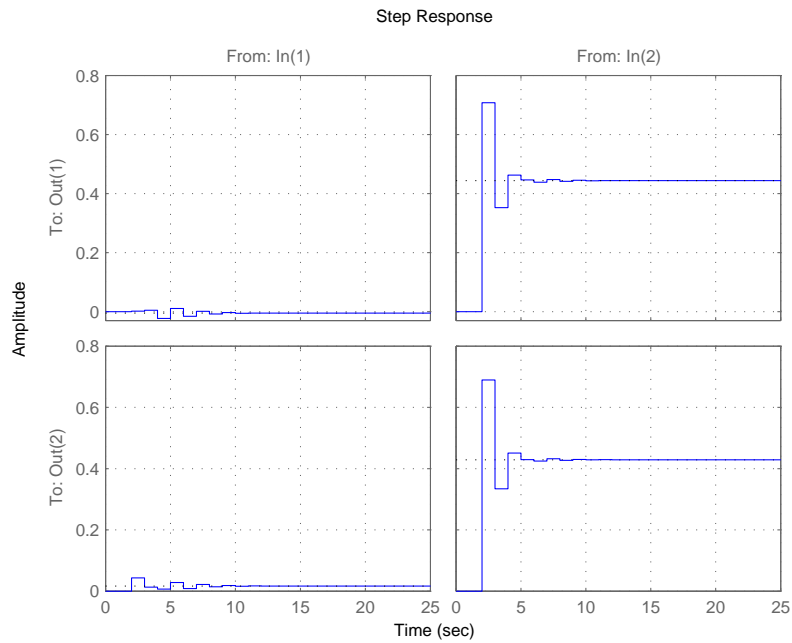


Figura 3.8: Resposta ao degrau unitário

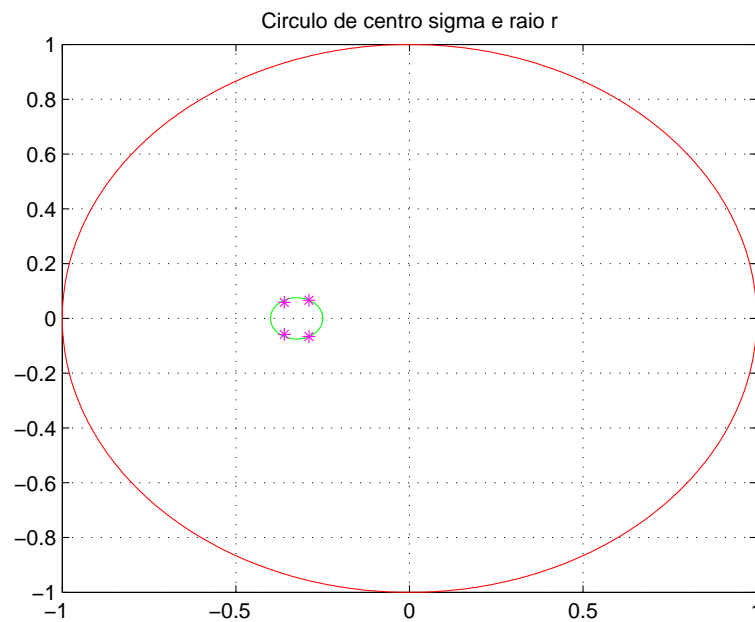


Figura 3.9: Auto valores do sistema em malha fechada

Foi feito um estudo em $3D$ que verificasse o comportamento do raio mínimo e raio máximo que contivesse os auto valores do sistema em malha fechada (realimentação de saída) em função

do raio r_1 (realimentação de estado) e do sigma σ_2 (realimentação de saída) passados como argumentos para resolver as realimentações de estado e saída. Para realizar os testes foram fixados cinco valores de $\sigma_1 = -0.8; -0.4; 0; 0.4; 0.8$ (realimentação de estados).

Com a análise desse gráficos, verificou-se que os valores mais próximos da origem tem maior influência no raio que contém os auto valores da malha fechada por realimentação de saída do que os que estão mais distantes, como pode ser comprovado por exemplo nos casos em que o sigma σ_1 é igual a 0 e a 0.4 mostrados respectivamente nas figuras a seguir.

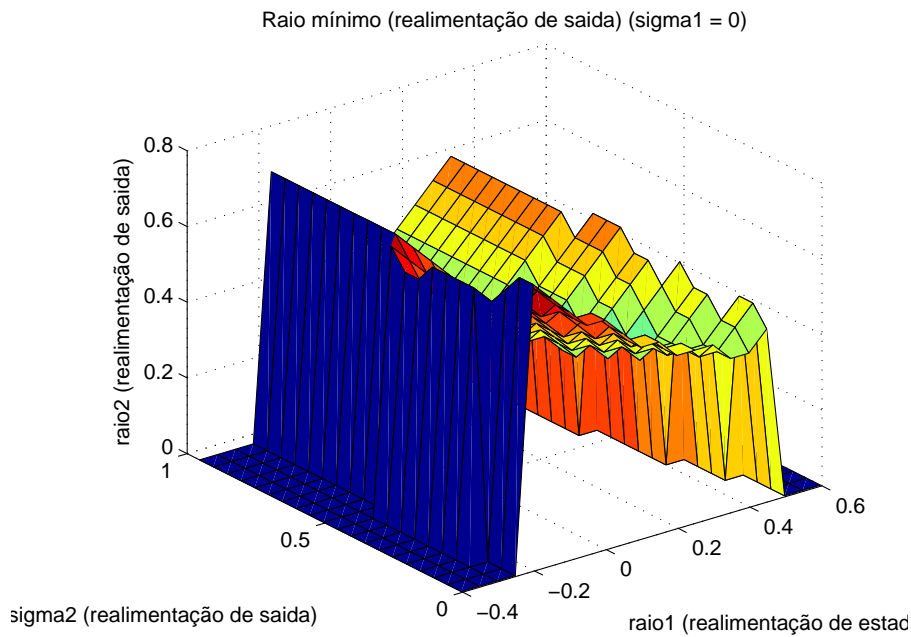
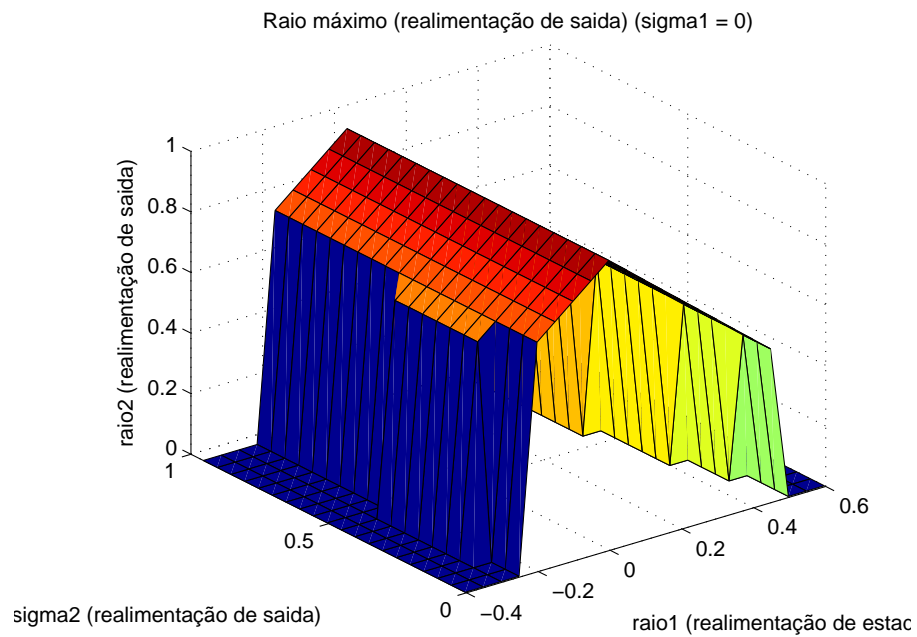
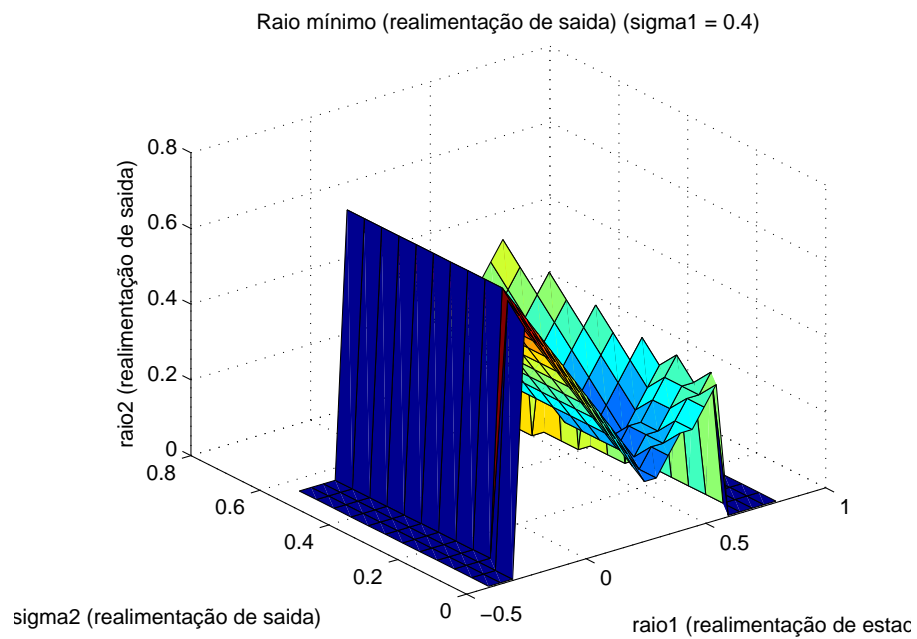


Figura 3.10: Raio mínimo (realimentação de saída) para sigma=0

Figura 3.11: Raio máximo (realimentação de saída) para $\sigma_1=0$ Figura 3.12: Raio mínimo (realimentação de saída) para $\sigma_1=0.4$

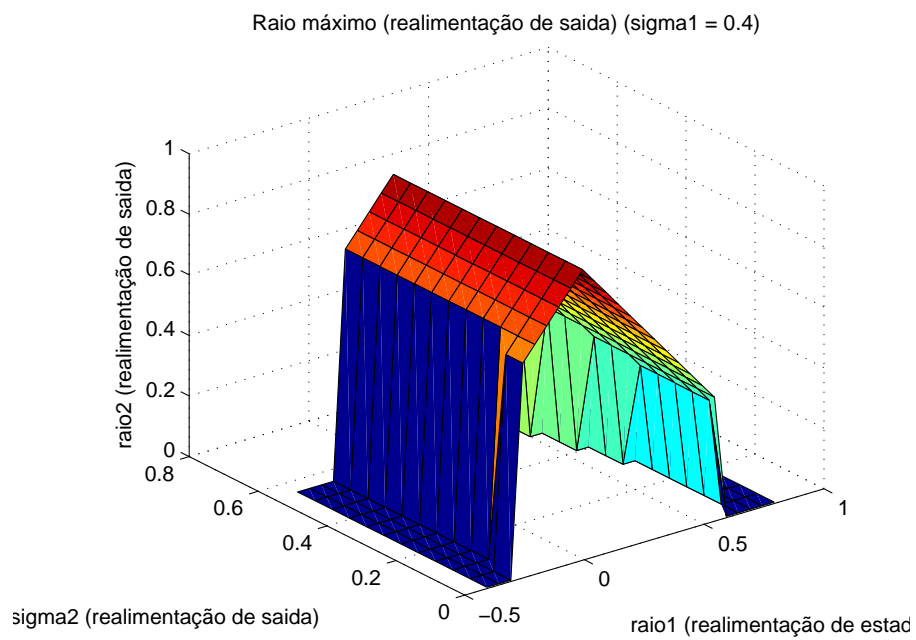


Figura 3.13: Raio máximo (realimentação de saída) para $\sigma_1=0.4$

Capítulo 4

Eletrônica - Desenvolvimento dos circuitos eletrônicos

A planta Placa Móvel Acionada por um Fluxo de Ar, mostrada na Figura 4.1, foi projetada e construída pelos alunos Leonardo Farias Campos, Lucas Silva de Oliveira e Thiago Silveira Costa, juntamente com o professor Valter Júnior Souza Leite, na disciplina optativa *Sinais e Sistemas*, ofertada no 7º período de Engenharia Mecatrônica do CEFET-MG, campus V - Divinópolis, no decorrer do primeiro semestre do ano de 2011.

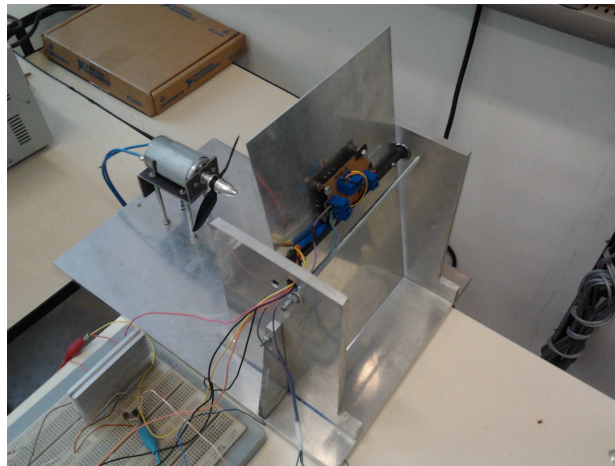


Figura 4.1: Placa Móvel Acionada por um Fluxo de Ar antes das modificações feitas

O objetivo da disciplina foi projetar, construir e validar uma planta para o estudo de controladores, além de obter uma planta com resposta rápida, pois a saída, após cada degrau, já é estável com aproximadamente 10 segundos. Seu projeto foi desenvolvido utilizando o software *AutoCad* para a parte mecânica, o software *KiCad* para a parte eletrônica e o desenvolvimento de sua interface gráfica foi feito através do software *LabView*.

A idéia inicial era medir a posição angular da placa móvel, porém essa parte ainda não foi implementada. Atualmente mede-se a tensão de saída do ventilador que controla a posição da placa. Para isso, o sensor utilizado é um acelerômetro por célula capacitiva, o que justifica uma saída bastante ruidosa.

No primeiro projeto, a parte eletrônica consistia em duas plaquinhas: uma com o sensor (acelerômetro) para fazer a medição, e outra para fazer o controle e acionamento do motor. Ambas as placas eram alimentadas com fontes industriais presentes no Cefet. Entretanto, devido a alimentação incorreta e ao fato de ser a mesma placa que fazia o controle e o acionamento do motor, os circuitos estavam com problemas de sobrecorrente e por isso alguns componentes estavam queimando com frequência. Para solucionar tal problema primeiramente foram construídas fontes com valores de tensão apropriados para cada alimentação do circuito, evitando assim a sobrecorrente. Além disso a placa que fazia o controle e tratamento do sinal foi separada (isolada), através de um optoacoplador, da placa que regulava o acionamento do motor, dessa forma, algum eventual problema que ocorra em uma placa, não interfere na outra. Foi então mantida a placa com o sensor, e foram produzidas as demais: duas fontes, uma placa de controle, e uma placa de potência. As placas foram refeitas usando o software *Proteus*.

4.1 Placa Acelerômetro

Essa placa não sofreu alteração. Ela consiste em um sensor acelerômetro *LIS344ALH* que é alimentado pela fonte com uma tensão de $+3.3V$, e um filtro *RC* passa-baixa cujo a frequência de corte é $10krad/s$ para eliminar os ruídos de alta frequência. Assim, o sinal medido é enviado pelo borne de saída. O circuito do acelerômetro foi projetado usando o software *KiCad* e pode ser visto na Figura 4.2.

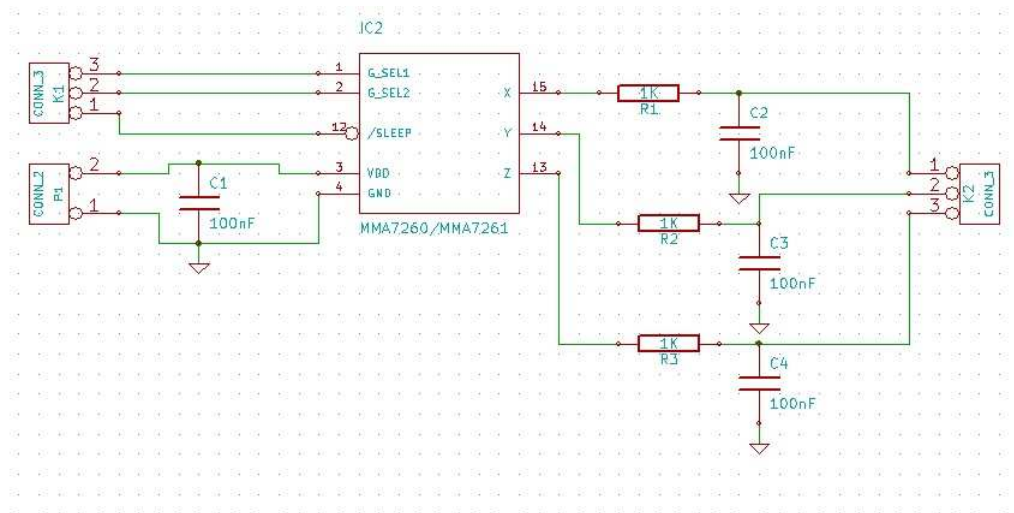


Figura 4.2: Circuito do Acelerômetro

4.2 Fonte

Foram feitas duas fontes de alimentação. A primeira com saídas de $+15V$, $-15V$, $+5V$, $-5V$ e $3.3V$. A saída de $+15V$ foi utilizada para alimentar os amplificadores, a saída de $+5V$ foi

utilizada para alimentar o PIC, e a saída de $+3.3V$ para alimentar o sensor. As saídas de $-15V$ e $-5V$ estão livres para um possível uso futuro. A segunda fonte tem uma saída de $+24V$ usada para alimentar o mosfet IRF840, e $+15V$ para alimentar o amplificador da placa de potência. Como necessitava de mais $+5V$ para alimentar o optoacoplador foi feita uma mini placa cuja entrada são os $+15V$ dessa fonte, e é composta de um regulador de tensão para fornecer a tensão de $+5V$. Nessa fonte havia também uma saída de $-15V$, mas essa foi eliminada. As fontes seguem o mesmo padrão. A tensão alternada da rede $127V$ passa pelo transformador cuja saída é $+24V$ $2A$ e entra no borne da fonte. Em seguida passa pelos diodos $1N4007$ que retificarão a onda senoidal, posteriormente os capacitores de $4700nF$ vão diminuir a tensão de ripple para tentar fornecer uma tensão contínua mais estável possível nos reguladores de tensão. As entradas dos reguladores possuem um capacitor de $330nF$ para contribuir na regulação do sinal de entrada. Na saída existe um capacitor de $100nF$ para manter a estabilidade do sinal e diminuir o período de transitório. Nessa etapa cada regulador de tensão vai oferecer a tensão correspondente. Nas figuras 4.3, 4.4, 4.5, 4.6, 4.7 e 4.8 são mostrados respectivamente os circuitos projetados, os layouts e as fotos das fontes 1 e 2.

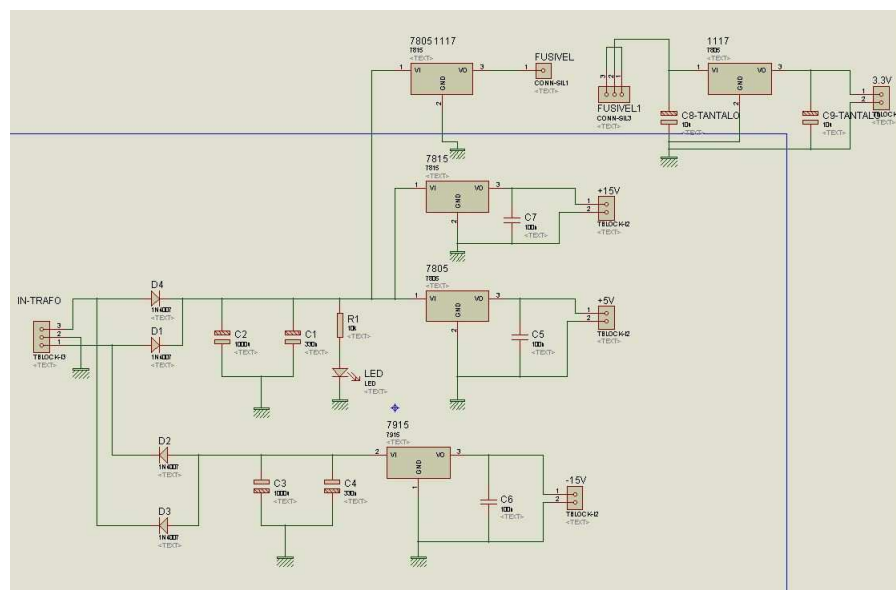


Figura 4.3: Circuito da Fonte de Alimentação 1

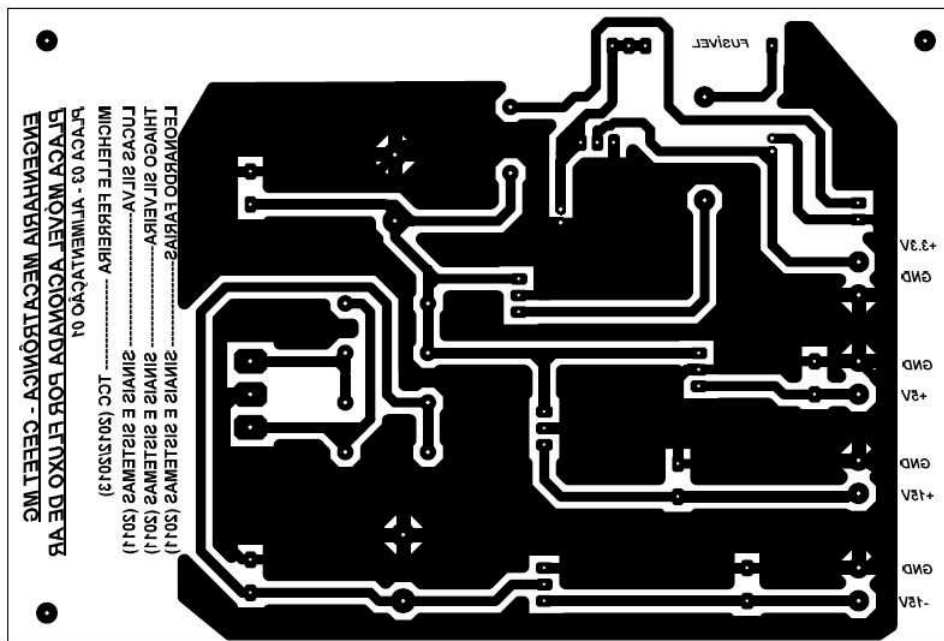


Figura 4.4: Layout da Fonte de Alimentação 1



Figura 4.5: Foto da Fonte de Alimentação 1

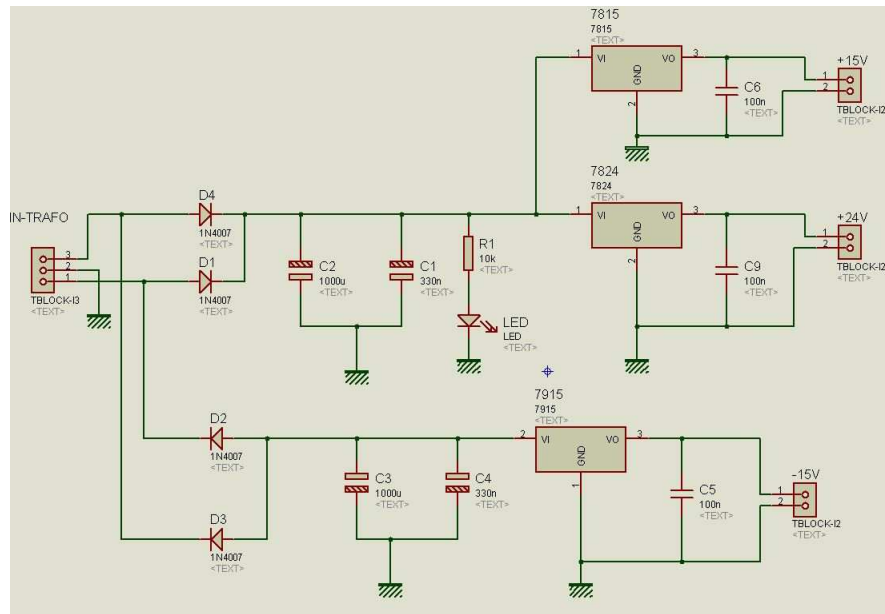


Figura 4.6: Circuito da Fonte de Alimentação 2



Figura 4.7: Layout da Fonte de Alimentação 2

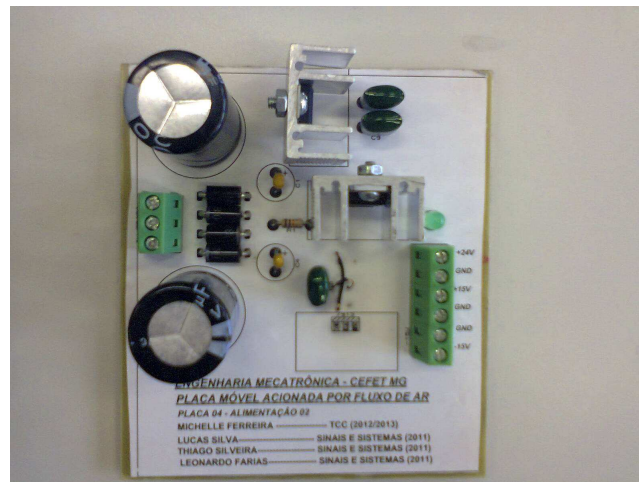


Figura 4.8: Foto da Fonte de Alimentação 2

A tabela 4.1 mostra os reguladores de tensão utilizados. Eles são componentes que fornecem uma tensão de saída fixa. A família $78xx$ fornece uma tensão positiva e a família $79xx$ uma tensão negativa.

Regulador	Tensão de saída
$LM7805$	$+5V$
$LM7905$	$-5V$
$LM7815$	$+15V$
$LM7915$	$-15V$
$LM7824$	$+24V$
$LM1117$	$+3.3V$

Tabela 4.1: Reguladores de tensão utilizados nas fontes

4.3 Placa de Controle

A placa de controle foi toda modificada a fim de melhorar o seu funcionamento. Primeiramente o filtro RC passa-baixa cuja frequência de corte é $100krad/s$ foi colocado depois do amplificador $LM358$ cuja entrada positiva é o sinal do sensor, dessa forma ele filtra sinais maiores. Em seguida foi acrescentado um diodo como forma de proteção para deixar a corrente passar somente em um sentido. Foi colocado um amplificador na configuração de buffer como mais uma forma de proteção para o circuito de amplificação, composto por mais um $LM358$. Este buffer tem como função isolar a entrada da saída do $LM358$, protegendo o circuito de um eventual retorno de corrente. Foi retirado o display, uma vez que ele não estava sendo usado. O sinal amplificado é enviado para uma placa de aquisição de dados da National Instruments.

O PIC 16F877A recebe o sinal do controlador que é retornado pelo computador e gera um sinal PWM que controla a velocidade do motor que é direcionado para a placa de potência. Foi projetado um jumper para ligar/desligar a função “sleep” do PIC, que permite parar o funcionamento do programa para verificar algum periférico ou instalar um novo. Mais dois jumpers permitem fazer a conexão de algum outro periférico caso seja necessário. Uma opção de fazer uma gravação no microcontrolador é utilizando os bornes “configura PIC” que permitem gravação de programas. Acrescentou-se um potenciômetro para, caso seja desejado, fazer o controle da rotação do motor manualmente. Por último a placa de controle foi isolada da placa de potência mais uma vez como forma de proteção contra sobrecorrentes. Pode-se perceber que as principais modificações na parte eletrônica da planta, foram com o intuito de evitar que o circuito queime, já que isso estava acontecendo com frequência no uso da planta em trabalho de disciplinas anteriores (Controle Digital e Controle Moderno). O circuito, o layout, e a foto da placa de controle são mostrados respectivamente nas figuras 4.9, 4.10 e 4.11.

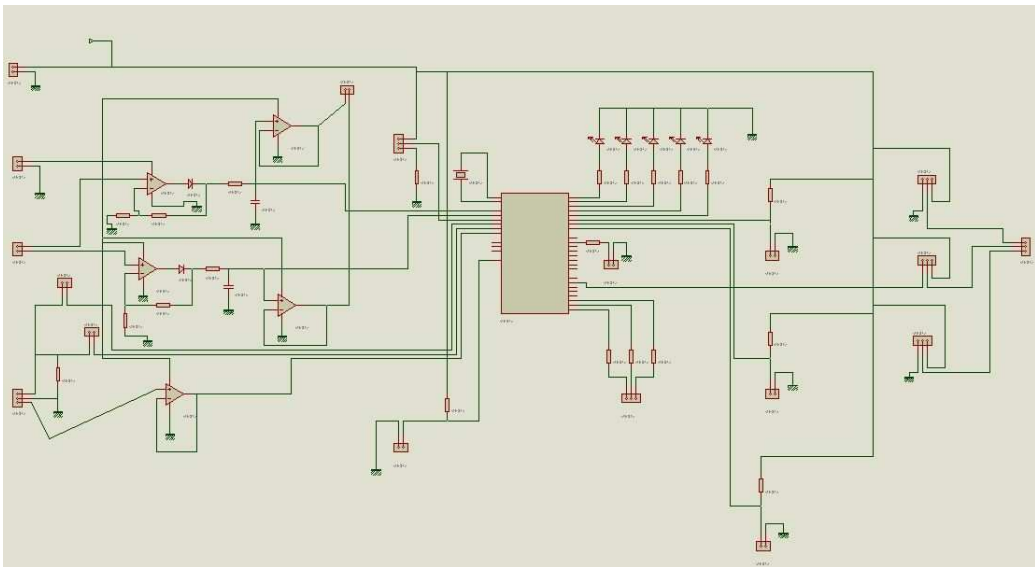


Figura 4.9: Circuito da Placa de Controle

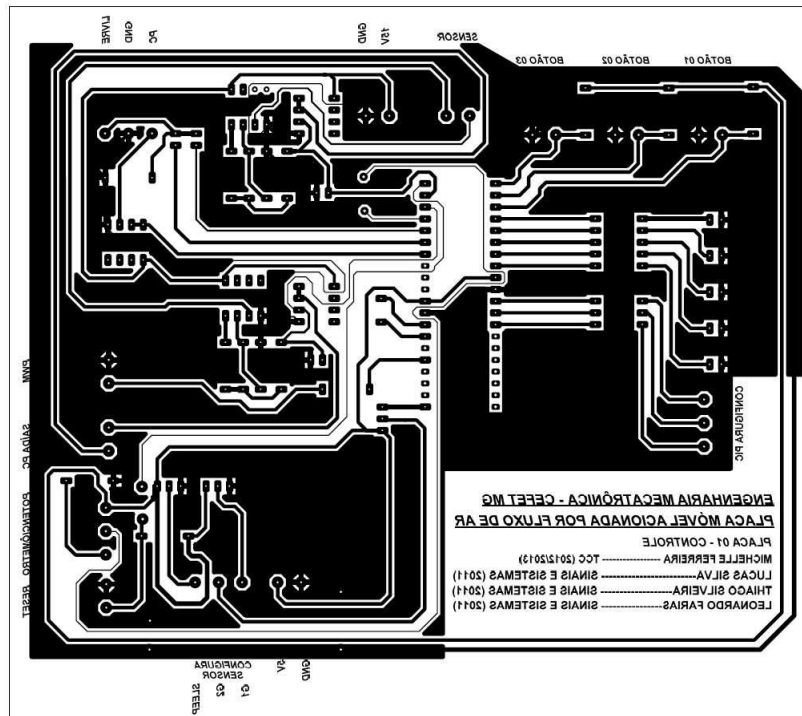


Figura 4.10: Layout da Placa de Controle

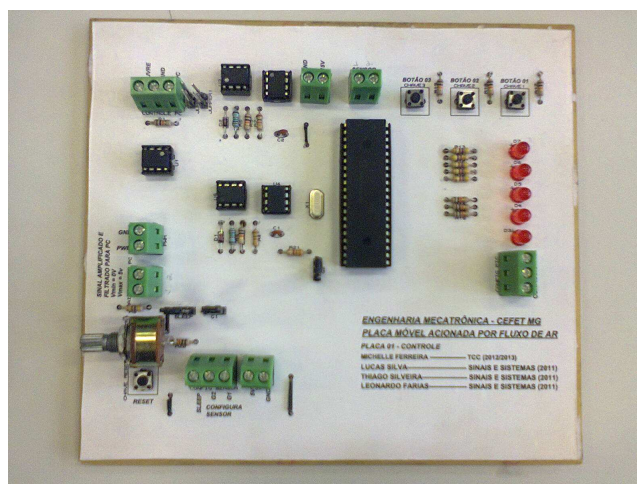


Figura 4.11: Foto da Placa de Controle

4.4 Placa de Potência

A placa de potência é usada para mandar o sinal controlado para o motor, para que o mesmo gire com a velocidade de rotação adequada para que a placa móvel estabilize em uma determinada posição. No projeto anterior essa placa foi construída juntamente com a placa controle. No novo projeto, essas duas placas foram separadas usando um optoacoplador 4N25 que isola o sistema de potência do sistema de sinal da placa de controle, pois ele é acionado através de uma emissão de luz a base de um transistor que faz o chaveamento do circuito de potência. A fonte de +5V isolada pelo opto vem da mesma fonte que alimenta o amplificador LM358 presente para amplificar o sinal PWM. Em seguida o sinal entra em um circuito denominado "Push Pull", que permite a passagem de 0V ou uma tensão contínua estabilizada, no caso desta aplicação, permite a passagem da parte baixa e alta do sinal PWM, desconsiderando um eventual transitório. Por último o sinal chega à chave do circuito de potência, no IRF840 que é um Mosfet de potência que chaveia a passagem de corrente pelo motor CC. Existe um diodo de potência MUR860 para impedir o retorno de corrente entre os terminais do motor, isso impede que o enrolamento do motor produza um retorno de corrente e provoque danos na fonte. Antes a placa de potência era feita utilizando TBJs. No novo projeto foi utilizado MOSFET, pois ele usa sinal de tensão (corrente baixa) que é mais fácil de processar.

O circuito, o layout e a foto da placa de potência são mostrados respectivamente nas figuras 4.12, 4.13 e 4.14.

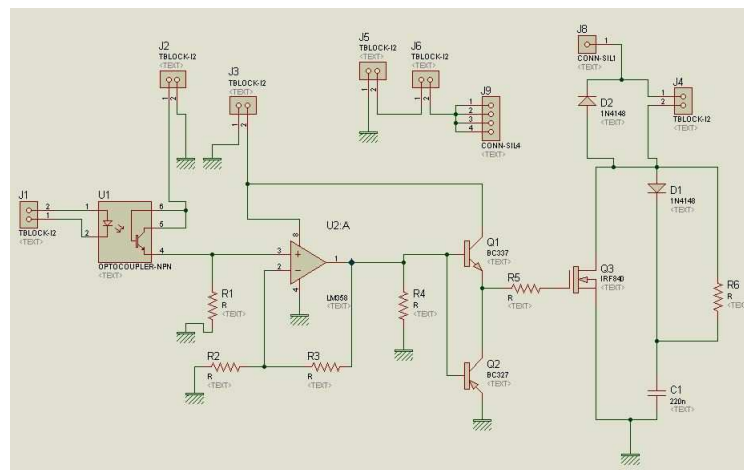


Figura 4.12: Circuito da Placa de Potência

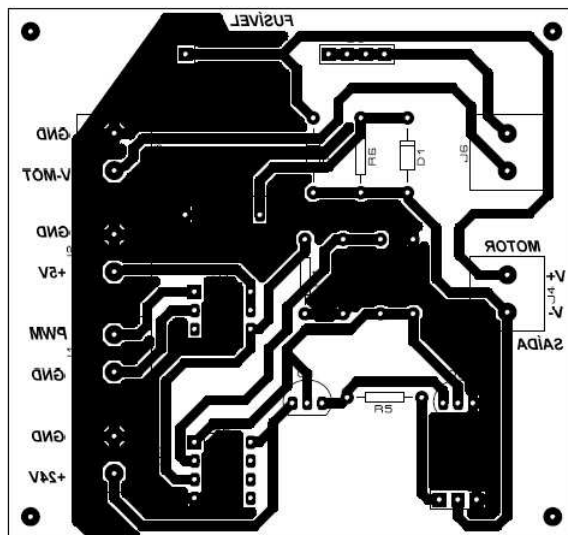


Figura 4.13: Layout da Placa de Potência

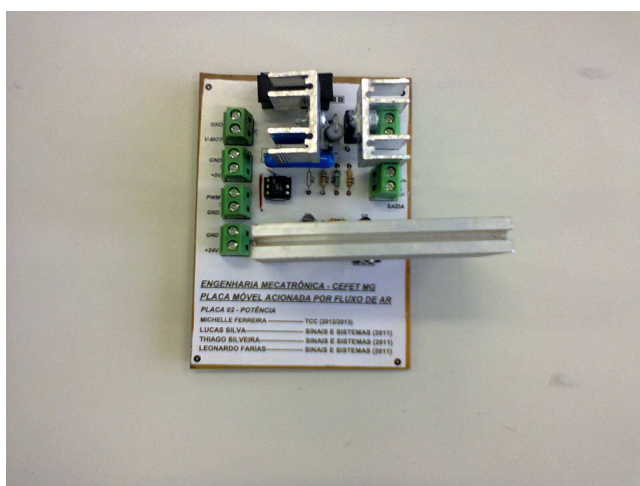


Figura 4.14: Foto da Placa de Potência

4.5 Planta Final

A planta, após todas as modificações feitas nos circuitos eletrônicos, pode ser vista na Figura 4.15.

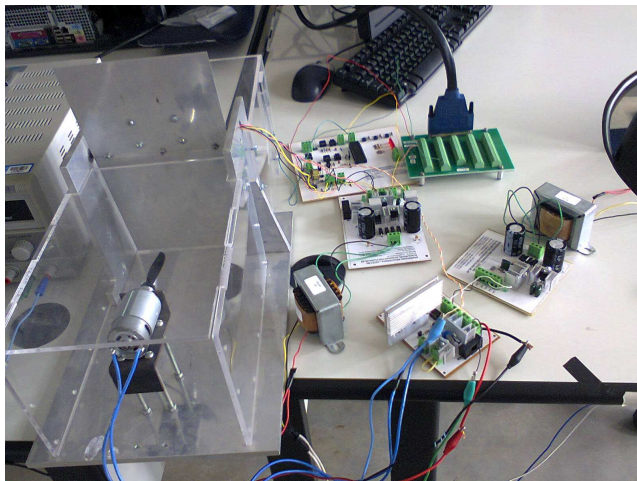


Figura 4.15: Placa Móvel Acionada por um Fluxo de Ar após modificações

Por último foi feita uma montagem a fim de facilitar o manuseio dos circuitos eletrônicos e da planta mostrada na Figura 4.16.

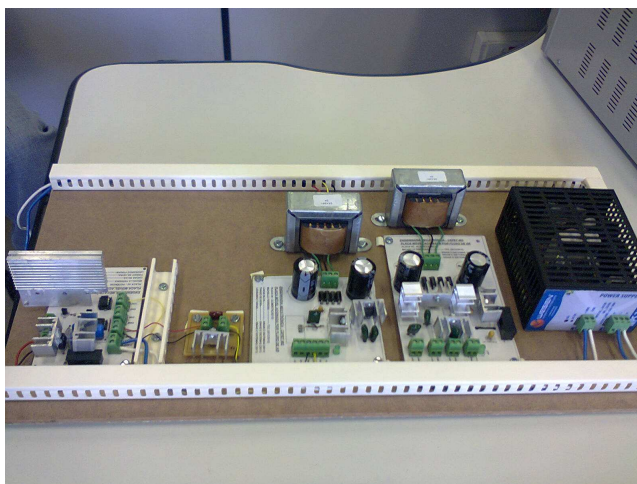


Figura 4.16: Placa Móvel Acionada por um Fluxo de Ar finalizada

Controle - Modelagem da Planta Real

5.1 Método da Resposta Complementar

O método da Resposta Complementar é um método que aproxima um sistema dinâmico por um sistema de primeira ou segunda ordem com atraso puro de tempo.

Esse método é aplicável a sistemas de segunda ordem sobreamortecidos com atraso puro de tempo ou de primeira ordem com atraso puro de tempo, os quais são representados pelas equações a seguir. A equação que representa o sistema de primeira ordem é:

$$H(s) = \frac{Y(s)}{U(s)} = \frac{Ke^{-s\theta}}{\tau s + 1} \quad (5.1)$$

Já a equação que representa o sistema de segunda ordem é:

$$H(s) = \frac{Y(s)}{U(s)} = \frac{Ke^{-s\theta}}{(\tau_1 s + 1)(\tau_2 s + 1)} \quad (5.2)$$

Tal método é utilizado quando uma entrada $u(t)$ em degrau é aplicada e sua respectiva saída $y(t)$ está disponível. Os dados de entrada $u(t)$ e saída $y(t)$ são então normalizados e a curva $\ln\left(1 - \frac{y(t)}{u(t)}\right)$ lineariza a resposta do degrau. Em seguida, a constante de tempo dominante do sistema τ_1 pode ser calculada a partir do inverso da inclinação da assíntota à curva. Assim:

$$\begin{aligned} \ln\left(1 - \frac{y(t)}{u(t)}\right) &= \frac{-1}{\tau_1}t + \ln\left(\frac{\tau_1}{\tau_1 - \tau_2}\right) \\ &= at + b. \end{aligned} \quad (5.3)$$

Sendo $at + b$ a equação da reta, a representa a inclinação da assíntota e b representa o deslocamento na ordenada. Logo, a constante de tempo dominante é calculada por:

$$\tau_1 = \frac{-1}{a} \quad (5.4)$$

Caso a curva $\ln\left(1 - \frac{y(t)}{u(t)}\right)$ seja uma reta, o sistema pode ser modelado como um sistema de primeira ordem, onde $\tau = \tau_1$. Caso contrário é necessário calcular a segunda constante de tempo τ_2 através da curva:

$$\begin{aligned}
 \ln \left[\frac{\tau_1}{\tau_1 - \tau_2} e^{t/\tau_1} \left(1 - \frac{y(t)}{u(t)} \right) \right] &= \ln \left[e^b e^{t/\tau_1} \left(1 - \frac{y(t)}{u(t)} \right) \right] \\
 &= \frac{-1}{\tau_2} t + \ln \left(\frac{\tau_2}{\tau_1 - \tau_2} \right). \\
 &= ct + d.
 \end{aligned} \tag{5.5}$$

Assim, na equação da reta $ct + d$, c é a inclinação da nova assíntota e d é o novo deslocamento na ordenada.

Portanto, a constante de tempo τ_2 será:

$$\tau_2 = \frac{-1}{c}. \tag{5.6}$$

Logo, o método da Resposta Complementar revela se a dinâmica do processo é de primeira ou de segunda ordem.

Neste trabalho, usou-se o método da Resposta Complementar de primeira ordem para obter um modelo da dinâmica da Placa Móvel Acionada por um Fluxo de Ar.

5.2 Obtenção de um Modelo Usando o Método da Resposta Complementar

Para controlar a tensão de saída da planta, foram medidos dados que possibilitaram levantar a curva de tensão de saída do ventilador por tempo. Assim, foi necessário medir a tensão aplicada no motor do ventilador (aplica-se uma proporção da tensão do ventilador - a tensão máxima que pode ser aplicada é 6V) e a tensão de saída da planta, em cada instante de tempo. Para realizar estas medições utilizou-se uma placa de aquisição de dados da National Instruments em conjunto como software computacional LabVIEW. No LabVIEW, o grupo que construiu a planta desenvolveu um programa para medir a tensão aplicada no motor do ventilador, bem como a tensão de saída do mesmo para cada instante de tempo. A interface do programa desenvolvido é mostrado na Figura 5.1.

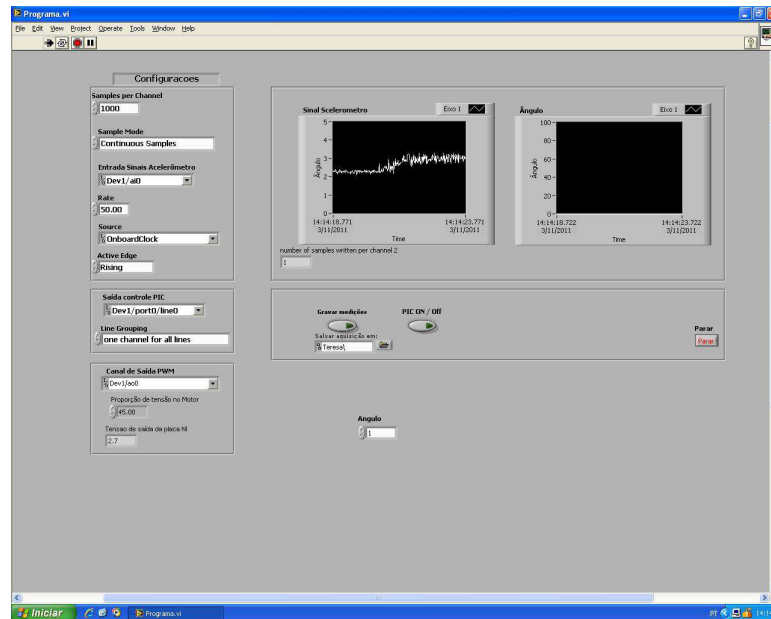


Figura 5.1: Interface do programa desenvolvido em LabVIEW

Assim, estabeleceu-se o ponto de operação de 60% da tensão máxima, valor que corresponde à 3.6V. A proporção a ser aplicada varia de aproximadamente 30% da tensão a 90% da mesma (1.8V a 5.4V), pois tensões que não estejam nessa faixa permitem uma variação angular da placa móvel acima da variação normal. Os dados foram coletados aplicando-se degraus de variação de 20% da tensão (degraus de subida para 80% e degraus de descida para 40% da tensão), com duração de 10 segundos cada.

Com os dados obtidos obteve-se o gráfico mostrado na Figura 5.2.

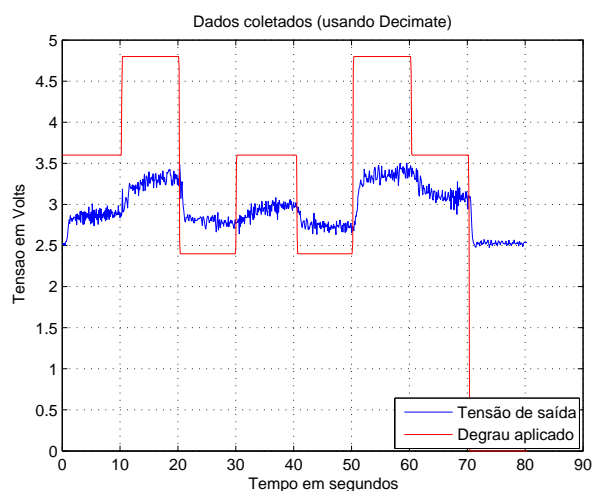


Figura 5.2: Dados coletados

Na Figura 5.2, a curva vermelha representa o degrau aplicado (em *Volts*), enquanto a curva azul representa a tensão de saída correspondente (em *Volts*), para cada degrau aplicado. Como forma de diminuir os ruídos foi utilizada a função *Decimate* para filtrar os dados, diminuindo a

amostragem.

O comando *Decimate* reduz a taxa de amostragem original para uma seqüência com uma taxa mais baixa. Em seu processo, filtra os dados de entrada com um filtro passa-baixa obtendo uma amostra do sinal suavizada. Usou-se o comando da forma:

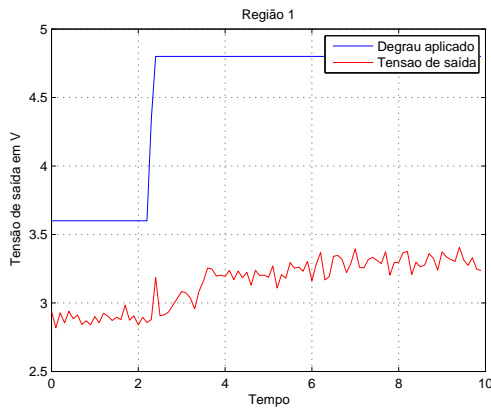
$$y = \text{decimate}(x, r, n, 'fir')$$

A taxa de amostragem original é de 50 pontos por segundo, ou seja, a cada 0.02 segundo obtém-se uma medida do sinal de saída. Assim, para modelar a planta, reduziu a taxa de amostragem tanto do degrau, quanto do sinal de saída por um fator de 5, de ordem 10, usando o filtro *fir*. O resultado do uso desse filtro foi a redução da taxa de amostragem original de 50 pontos por segundo para uma taxa de amostragem de 10 pontos por segundo, o que representa uma medida do sinal de saída a cada 0.1 segundo.

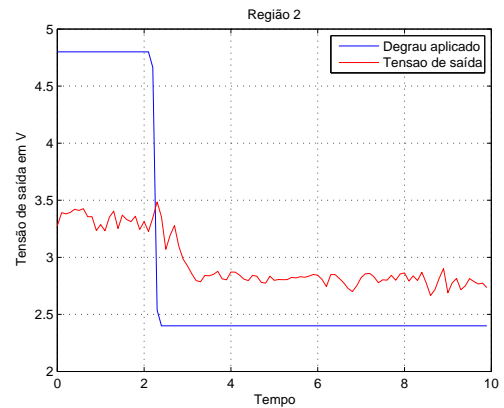
Para modelar a planta foi utilizado o método da resposta complementar de primeira ordem. Para usar esse método, primeiramente foram determinadas 6 regiões de interesse: 3 de subida e 3 de descida.

A **Região 1** corresponde à região do degrau de 60% (3.6V) a 80% (4.8V) da tensão máxima do motor (6V). A **Região 2** corresponde ao degrau de 80% da tensão para o degrau de 40% (2.4V). A **Região 3** corresponde ao degrau de 40% para o de 60%. A **Região 4** corresponde ao degrau de 60% para o de 40%. A **Região 5** corresponde ao degrau de 40% para o de 80%. A **Região 6** corresponde ao degrau de 80% para o de 60%.

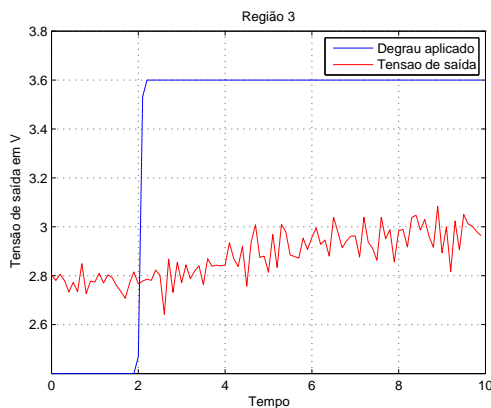
As regiões de interesse (em *V*), podem ser visualizadas na Figura 5.3.



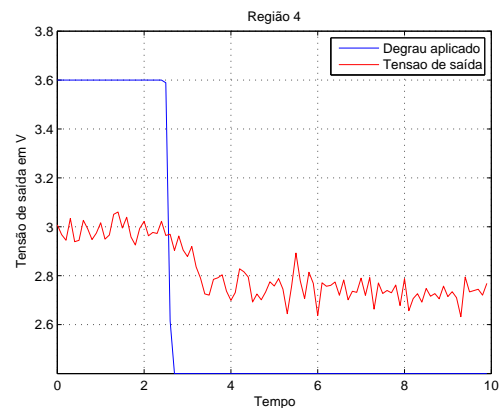
(a) Região 1



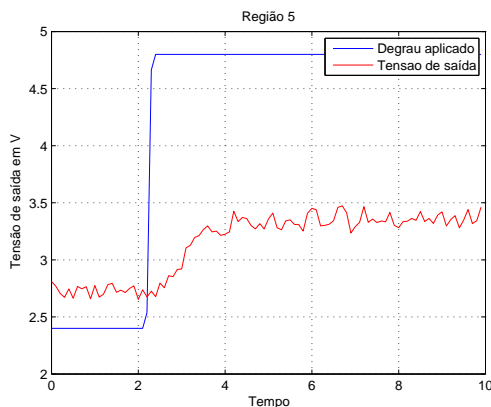
(b) Região 2



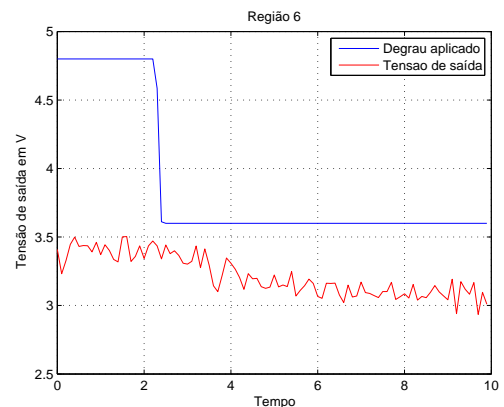
(c) Região 3



(d) Região 4



(e) Região 5



(f) Região 6

Figura 5.3: Regiões 1, 2, 3, 4, 5 e 6

Posteriormente, deslocou-se as curvas obtidas anteriormente para a origem dos sistema de coordenadas. Para isso, utilizou-se:

$$\text{região deslocada} = \text{região de interesse} - \text{média dos primeiros valores da região de interesse}$$

A Figura 5.4 mostra, respectivamente, as curvas deslocadas obtidas para cada uma das

regiões de interesse.

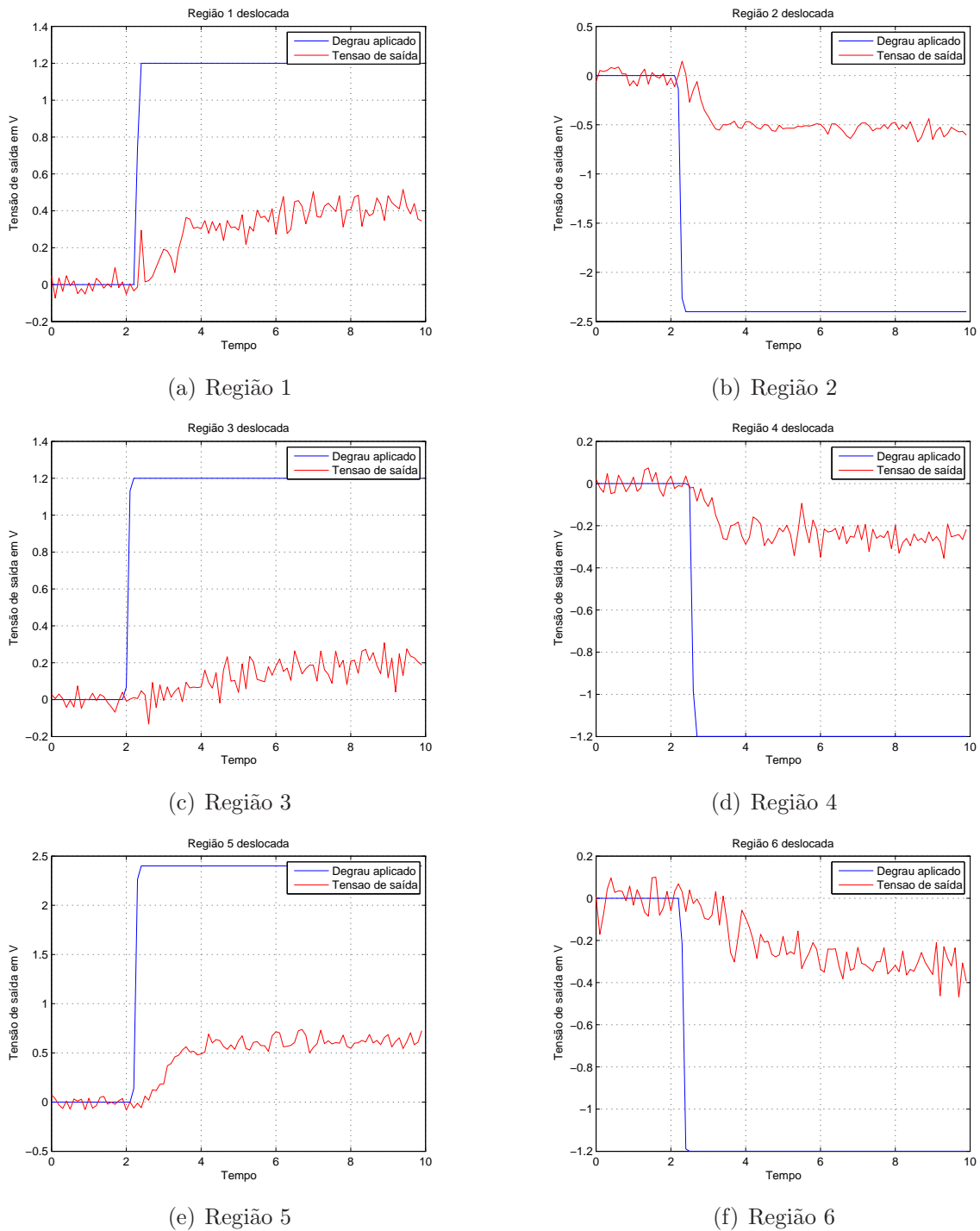


Figura 5.4: Regiões 1, 2, 3, 4, 5 e 6 deslocadas

A partir das curvas deslocadas foram calculados os ganhos de cada uma das regiões. O ganho é dado por:

$$K = \text{média dos últimos valores da tensão de saída deslocada} \div \text{média dos últimos valores da tensão de entrada deslocada}$$

Os ganhos obtidos para cada uma das regiões podem ser visualizados na Tabela 5.1:

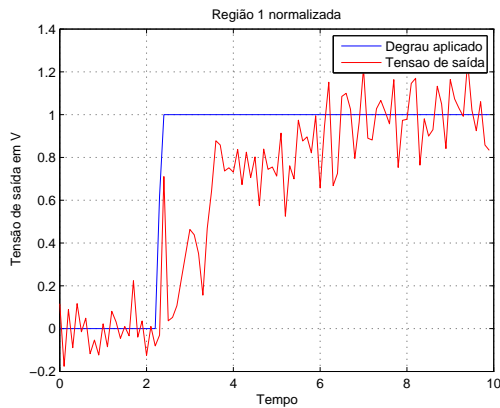
Região	Ganho (K)
1	0.3453
2	0.2270
3	0.1602
4	0.2141
5	0.2563
6	0.2645

Tabela 5.1: Ganhos das regiões de interesse

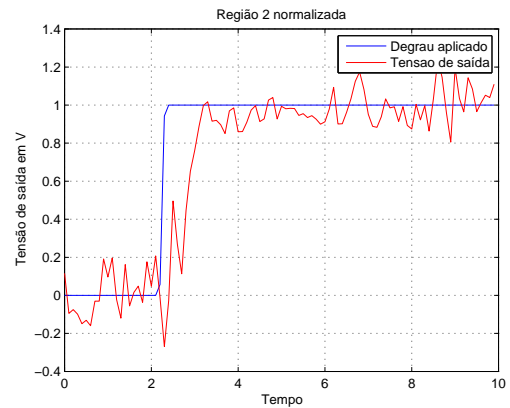
Em sequência, as curvas obtidas anteriormente foram normalizadas. Para isso, utilizou-se:

$$\textit{região normalizada} = \textit{região deslocada} \div \textit{média dos últimos valores da região de interesse deslocada}$$

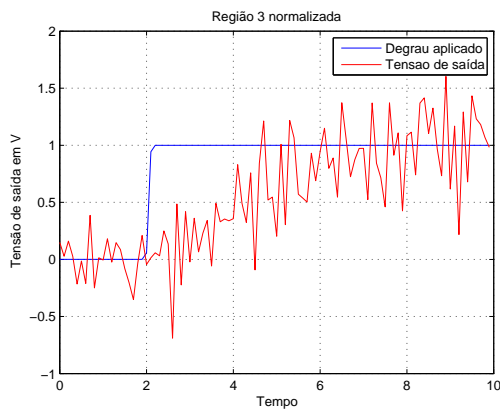
A Figura 5.5 mostra, respectivamente, as curvas deslocadas e normalizadas obtidas para cada uma das regiões de interesse.



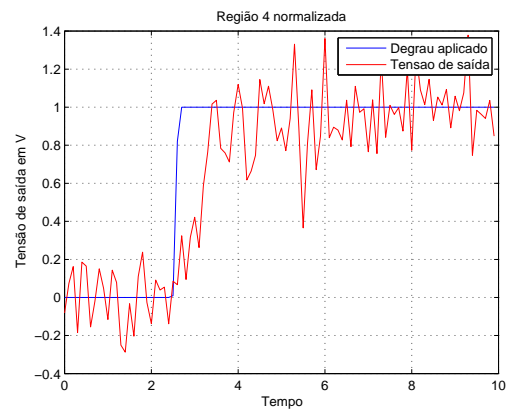
(a) Região 1



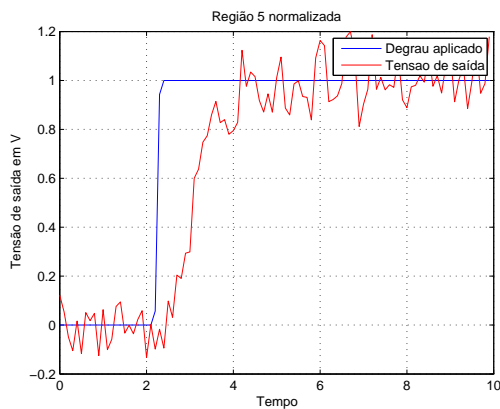
(b) Região 2



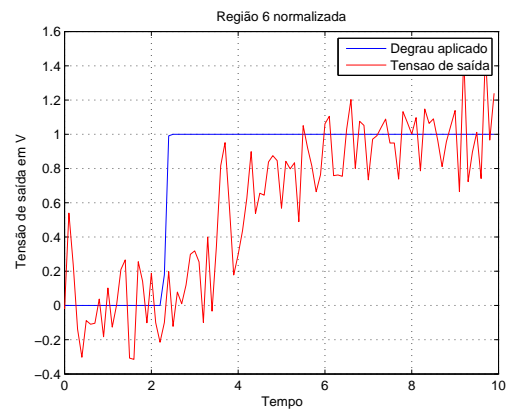
(c) Região 3



(d) Região 4



(e) Região 5



(f) Região 6

Figura 5.5: Regiões 1, 2, 3, 4, 5 e 6 deslocadas e normalizadas

Através da Figura 5.5 foram encontrados os atrasos em cada uma das curvas. Os atrasos obtidos podem ser visualizados na Tabela 5.2.

Região	Atraso(θ)
1	0.1
2	0.2
3	0.1
4	0.1
5	0.3
6	0.3

Tabela 5.2: Atrasos das regiões de interesse

Posteriormente, as curvas obtidas foram linearizadas pela equação 5.3. As curvas linearizadas, obtidas para cada região, podem ser vistas na Figura 5.6.

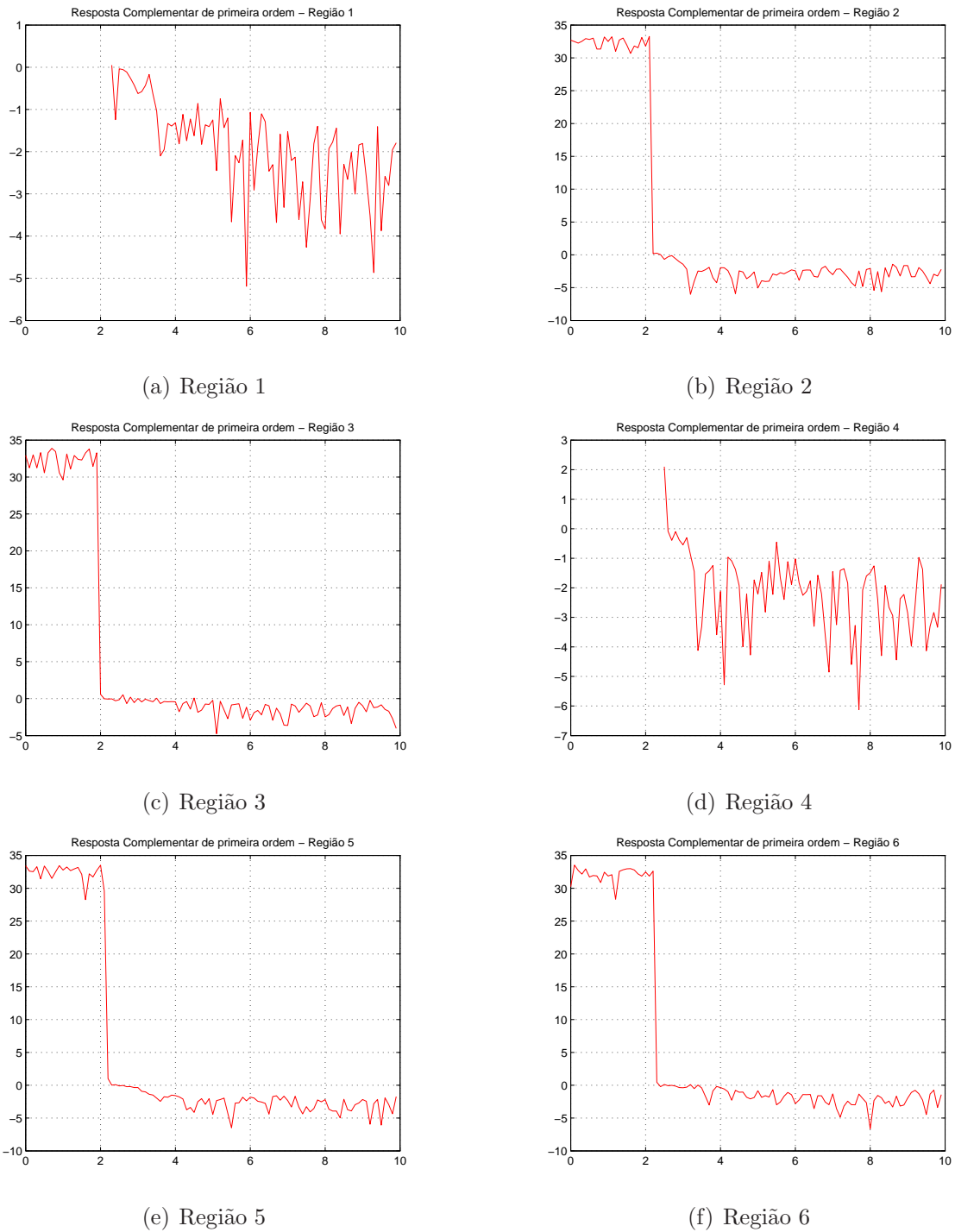


Figura 5.6: Regiões 1, 2, 3, 4, 5 e 6 linearizadas

A partir das curvas linearizadas, escolheu-se os pontos que melhor formassem uma reta, e a partir deles, utilizando o comando *polyfit*, do *MatLab*, obteve-se a equação de cada reta.

A constante de tempo do sistema é dada pelo coeficiente angular da reta encontrada. Ou seja, a constante de tempo denominada τ_1 é dada pela equação 5.4. Assim, as constantes de tempo obtidas para cada região são mostradas na Tabela 5.3.

Região	Constante de tempo(τ_1)
1	0.4425
2	0.3703
3	3.7021
4	0.5936
5	0.3373
6	1.0036

Tabela 5.3: Constantes de tempo das regiões de interesse

A partir das retas obtidas, utilizando o ganho K , o atraso e a constante de tempo τ_1 foi possível estabelecer a função de transferência de primeira ordem para cada região de interesse. As equações obtidas foram:

- Região 1:

$$G_1(s) = \frac{0.3453}{0.4425s + 1} e^{-0.1s} \quad (5.7)$$

- Região 2:

$$G_2(s) = \frac{0.2270}{0.3703s + 1} e^{-0.2s} \quad (5.8)$$

- Região 3:

$$G_3(s) = \frac{0.1602}{3.7021s + 1} e^{-0.1s} \quad (5.9)$$

- Região 4:

$$G_1(s) = \frac{0.2141}{0.5936s + 1} e^{-0.1s} \quad (5.10)$$

- Região 5:

$$G_2(s) = \frac{0.2563}{0.3373s + 1} e^{-0.3s} \quad (5.11)$$

- Região 6:

$$G_3(s) = \frac{0.2645}{1.0036s + 1} e^{-0.3s} \quad (5.12)$$

Como em cada linearização, obteve-se uma reta, o método da Resposta Complementar permitiu que a dinâmica do sistema fosse modelada como um sistema de primeira ordem. A comparação entre o modelo obtido e o sistema real é mostrada na Figura 5.7.

Em seguida, utilizando os parâmetros obtidos para cada uma das regiões modeladas, obteve-se um modelo médio.

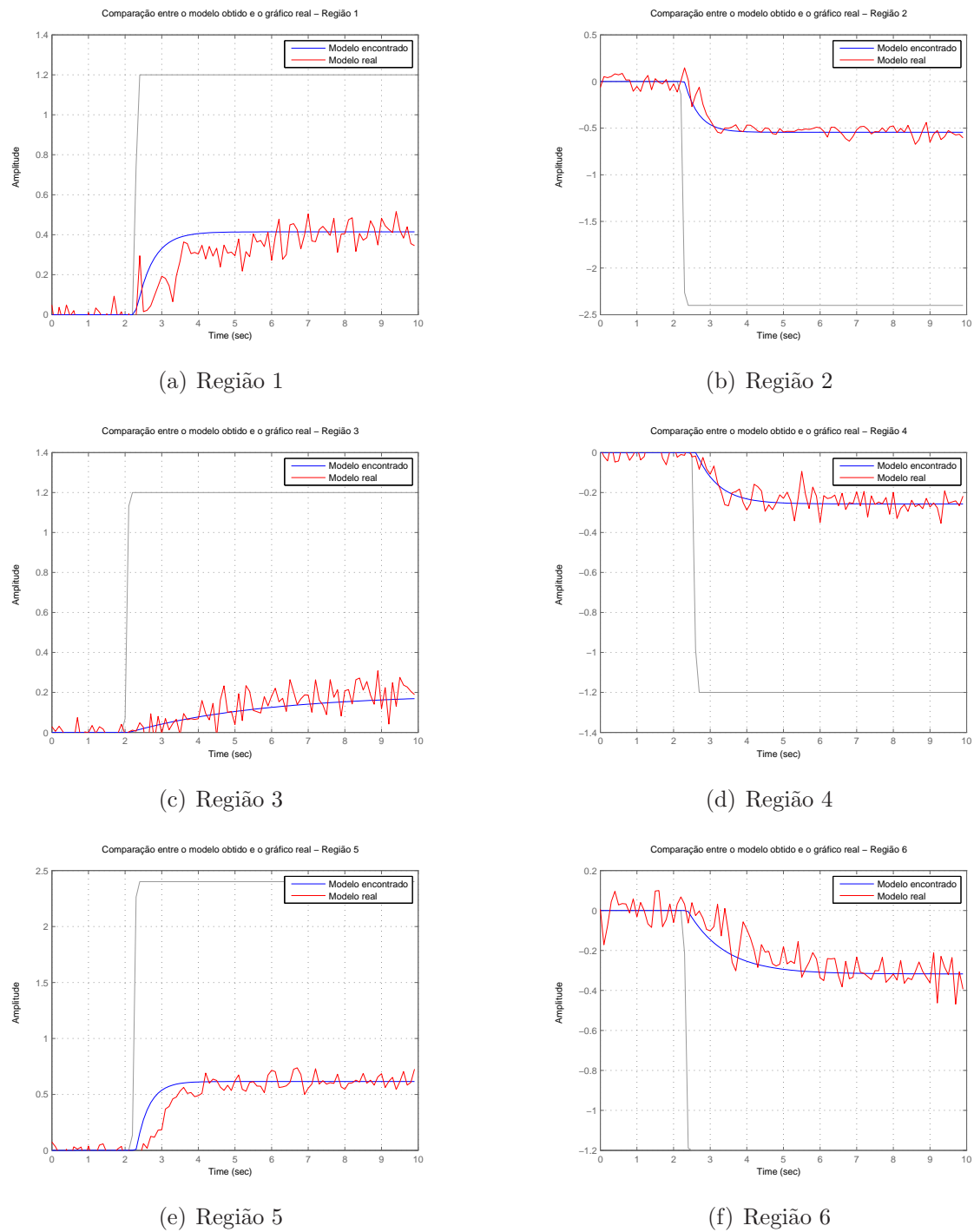


Figura 5.7: Comparação entre o modelo obtido e o sistema real - Regiões 1, 2, 3, 4, 5 e 6

5.2.1 Modelo Médio

No intuito de obter uma função que modelasse melhor os dados coletados, fez-se uma média dos parâmetros obtidos para cada região modelada. Assim, os parâmetros encontrados são mostrados na Tabela 5.4.

Parâmetro	Média
Ganho (K)	0.2446
Constante de tempo (τ_1)	1.0749
Atraso (θ)	0.1833

Tabela 5.4: Parâmetros médios

Em seguida, utilizando-se a função *pade*, que aproxima as funções de transferência de primeira ordem obtida por uma função de transferência racional, totalizando um sistema de segunda ordem, obteve-se o seguinte modelo médio.

$$G_{pade}(s) = \frac{0.2446}{1.0749s + 1} e^{-0.1833s} \approx \frac{-0.22751(s - 10.91)}{(s + 10.91)(s + 0.9303)} \quad (5.13)$$

Assim, com a função *lsim*, obteve-se a comparação do modelo médio final obtido com o sistema real, mostrada na Figura 5.8.

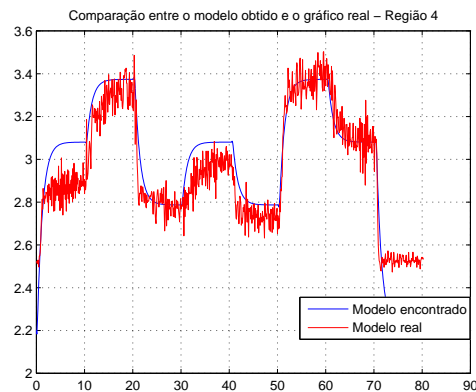


Figura 5.8: Comparação entre o modelo obtido e o sistema real

Como pode ser visto na Figura 5.8, tanto os modelos de cada região e o modelo médio aproximaram-se do comportamento real da planta. Entretanto esses modelos são contínuos no tempo e o foco desse trabalho não é esse tipo de sistema e sim sistemas incertos discretos no tempo, tema que será tratado no próximo capítulo.

Controle - Obtenção do Sistema Discreto

6.1 Discretização do Sistema

Para se obter um modelo incerto discreto no tempo, deve-se primeiramente obter o modelo contínuo, para então discretizá-lo. Para isso deve-se incluir o hold no sistema, cujo a função é segurar a próxima amostragem até o instante seguinte. Essa configuração pode ser vista na Figura 6.1.

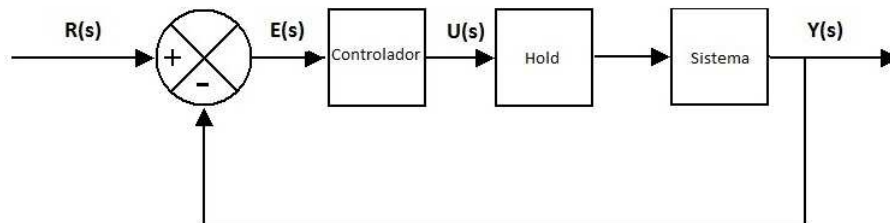


Figura 6.1: Diagrama do sistema com o hold e o controlador

O hold é dado pela equação:

$$\frac{1 - e^{Ts}}{s}$$

e o sistema é do tipo:

$$\frac{k(s + a)}{(s + b)(s + a)}$$

Define-se:

$$G(s) = \frac{1 - e^{Ts}}{s} \left(\frac{k(s + a)}{(s + b)(s + a)} \right)$$

Dessa forma tem-se que:

$$Y(s) = G(s)U(s)$$

Após algumas manipulações algébricas e aplicando a Transformada \mathcal{Z} para obter o sistema discreto no tempo, tem-se:

$$Y(z) = G(z)U(z)$$

$$\frac{Y(z)}{U(z)} = \left(\frac{z-1}{z}\right)k\mathcal{Z}\left\{\left(\frac{b-a}{b-a}\right)\left(\frac{1}{(s+b)(s+a)}\right) - \frac{b}{b}\left(\frac{a}{s(s+a)(s+b)}\right)\right\}$$

$$\frac{Y(z)}{U(z)} = \left(\frac{k}{b-a}\right)\left(\frac{(e^{-aT_s} - e^{-bT_s})z}{(z - e^{-aT_s})(z - e^{-bT_s})}\right) - \dots$$

$$\dots - \frac{k}{b}\left(\left(\frac{z}{z-1}\right) + \left(\frac{b}{a-b}\right)\left(\frac{z}{z - e^{-aT_s}}\right) - \left(\frac{a}{a-b}\right)\left(\frac{z}{z - e^{-bT_s}}\right)\right) \quad (6.1)$$

Reorganizando os termos obtém-se a função de transferência mostrada na equação (6.2):

$$\frac{Y(z)}{U(z)} = k \frac{(a(1 + 2e^{-aT_s} + e^{-bT_s}) - b(1 + 3be^{-bT_s}))z - (a + p)e^{-aT_s} + 2be^{bT_s} + (a - p)e^{-(a+b)T_s}}{(a - b)(z - e^{-aT_s})(z - e^{-bT_s})} \quad (6.2)$$

Substituindo os parâmetros na equação (6.2), obtém-se a seguinte função de transferência para o sistema precisamente conhecido discreto no tempo.

$$\frac{Y(z)}{U(z)} = \frac{-0.75551(s + 0.009111)}{(s^2 - 0.6708s + 0.6708)} \quad (6.3)$$

Usando a função ss do MATLAB[®] obtém-se a equação escrita na forma de espaço de estados. A partir daí foi obtido o modelo incerto discreto no tempo mostrado na seção a seguir.

6.2 Obtenção do Sistema Incerto Discreto no Tempo

Os valores de a e b da equação do sistema podem ser escritos em função de τ , k e θ .

$$a = \frac{2}{\theta} \quad (6.4)$$

$$b = \frac{1}{\tau} \quad (6.5)$$

De posse das equações (6.2), (6.4) e (6.5) pode-se obter o sistema incerto discreto no tempo, substituindo os parâmetros τ , θ e k na mesma. Foram modeladas 6 regiões de operação. Para cada uma delas foram obtidos τ , θ e k , portanto tem-se os valores mínimo e máximo de cada um desses parâmetros. Como não se sabe ao certo seus valores, modela-se um sistema incerto discreto no tempo composto das 8 combinações de cada um desses valores mínimo e máximo. Portanto forma-se um polítopo com 8 vértices, cujo cada vértice é dado por (6.2) substituindo τ , θ e k de acordo com o vértice correspondente na Tabela 6.1.

Vertice	τ	θ	k
1	minimo	minimo	minimo
2	minimo	minimo	maximo
3	minimo	maximo	minimo
4	minimo	maximo	maximo
5	maximo	minimo	minimo
6	maximo	minimo	maximo
7	maximo	maximo	minimo
8	maximo	maximo	maximo

Tabela 6.1: Combinações dos parâmetros τ , θ e k

Foi projetado um sistema em malha fechada duas vezes mais rápido do que em malha aberta, com isso o valor do período de amostragem é $T_s = 0.0337$. Para essa especificação, e substituindo os valores dos parâmetros mínimo e máximo na equação de espaço de estados obtida no MATLAB[®] obteve-se o seguinte sistema discreto incerto no tempo.

$$A_1 = \begin{bmatrix} 0.4609 & -0.4609 \\ 1.0000 & 0 \end{bmatrix} \quad A_2 = \begin{bmatrix} 0.7226 & -0.7226 \\ 1.0000 & 0 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} 0.4609 & -0.4609 \\ 1.0000 & 0 \end{bmatrix} \quad A_4 = \begin{bmatrix} 0.7226 & -0.7226 \\ 1.0000 & 0 \end{bmatrix}$$

$$A_5 = \begin{bmatrix} 0.5048 & -0.5048 \\ 1.0000 & 0 \end{bmatrix} \quad A_6 = \begin{bmatrix} 0.7914 & -0.7914 \\ 1.0000 & 0 \end{bmatrix}$$

$$A_7 = \begin{bmatrix} 0.5048 & -0.5048 \\ 1.0000 & 0 \end{bmatrix} \quad A_8 = \begin{bmatrix} 0.7914 & -0.7914 \\ 1.0000 & 0 \end{bmatrix}$$

$$B_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad B_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad B_3 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad B_4 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$B_5 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad B_6 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad B_7 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad B_8 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$C_1 = [1.8575 \quad 0.0892] \quad C_2 = [-0.9411 \quad 0.0941]$$

$$C_3 = [1.8575 \quad 0.0892] \quad C_4 = [-0.9411 \quad 0.0941]$$

$$C_5 = [1.5131 \quad 0.0043] \quad C_6 = [1.8318 \quad 0.0045]$$

$$C_7 = [1.51310.0043] \quad C_8 = [1.83180.0045]$$

$$D_1 = [0] \quad D_2 = [0] \quad D_3 = [0] \quad D_4 = [0]$$

$$D_5 = [0] \quad D_6 = [0] \quad D_7 = [0] \quad D_8 = [0]$$

Para achar esses vértices foi feito um algoritmo que fazia a combinação desses valores e substituiu em (6.2). O projeto do controlador para sistema incerto discreto no tempo é mostrado no capítulo a seguir.

Controle - Projeto e Simulação do Controlador

7.1 Projeto do controlador para o sistema precisamente conhecido

Primeiramente foi projetado um controlador para o sistema discreto precisamente conhecido obtido através dos parâmetros médios. Ele foi projetado para que a malha fechada fosse duas vezes mais rápida que a malha aberta. O algoritmo feito encontra um controlador k que deixa os pólos do sistema o mais próximo possível uns dos outros. Os controladores k_o e k projetados para o sistema (6.3) foram respectivamente:

$$k_o = [-0.0343 \quad -4.1165 \quad -1.6968] \quad (7.1)$$

$$k = -15.9364 \quad (7.2)$$

A resposta do sistema (5.13) em malha fechada, usando os controladores (7.1) e (7.2), e seus pólos são mostrados respectivamente nas figuras 7.1 e 7.2

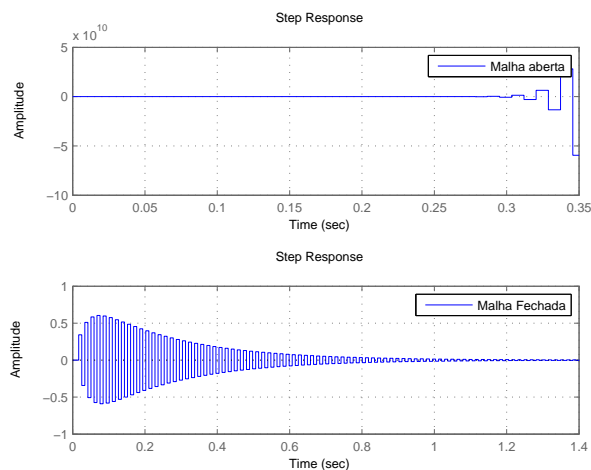


Figura 7.1: Resposta do sistema modelado

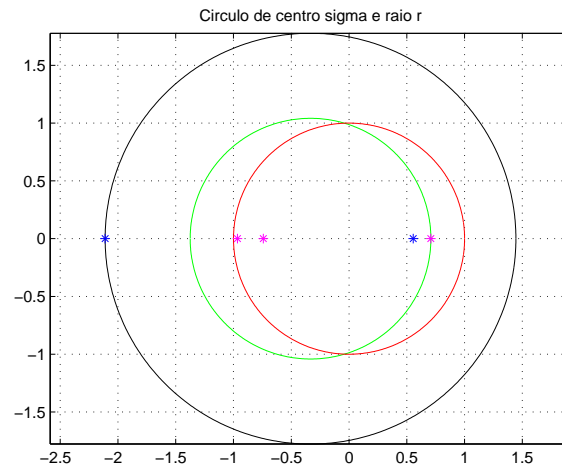


Figura 7.2: Auto valores dos sistema em malha aberta e malha fechada

Pode ser verificado que o sistema era instável em malha aberta, e que ao fechar a malha e ele se tornou estável. Os pólos em azul (malha aberta) estão fora do círculo de raio unitário (círculo vermelho), o que comprova sua instabilidade. Já os pólos em rosa (malha fechada) estão dentro do círculo unitário, ou seja, são estáveis.

7.2 Projeto do controlador para o sistema incerto

Posteriormente foi projetado o controlador para o sistema incerto discreto no tempo obtido para a placa acionada por um fluxo de ar. Os controladores k_o e k projetados foram respectivamente:

$$k_o = [-0.0030 \quad 0.3186] \quad (7.3)$$

$$k = 0.4414 \quad (7.4)$$

A resposta do sistema (5.13) em malha fechada, usando os controladores (7.3) e (7.4), e seus pólos são mostrados respectivamente nas figuras 7.3 e 7.4

Como especificação de projeto desejava-se alocar os pólos do sistema em malha fechada dentro de um círculo de raio 0.1071 e centro 0.6404. Essa especificação foi atendida como pode ser verificado na Figura 7.4.

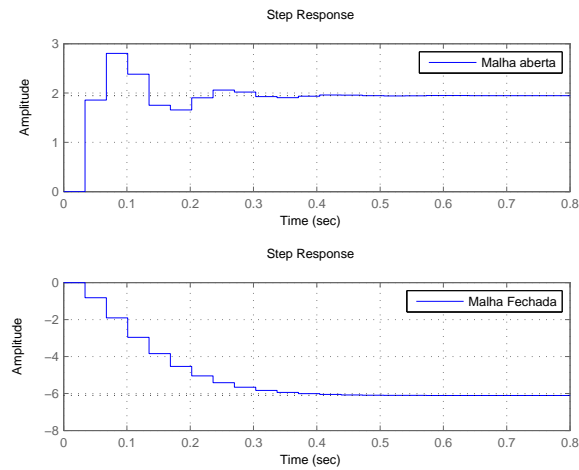


Figura 7.3: Resposta do sistema modelado

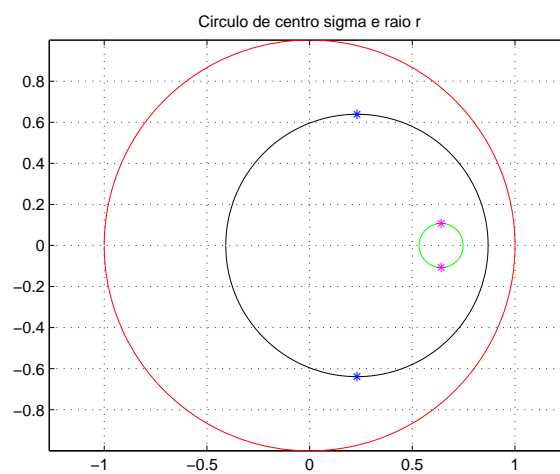


Figura 7.4: Auto valores dos sistema em malha aberta e malha fechada

7.3 Implementação

Para a implementação na planta em tempo real, foi feito um programa usando o *Simulink* mostrado na Figura 7.5.

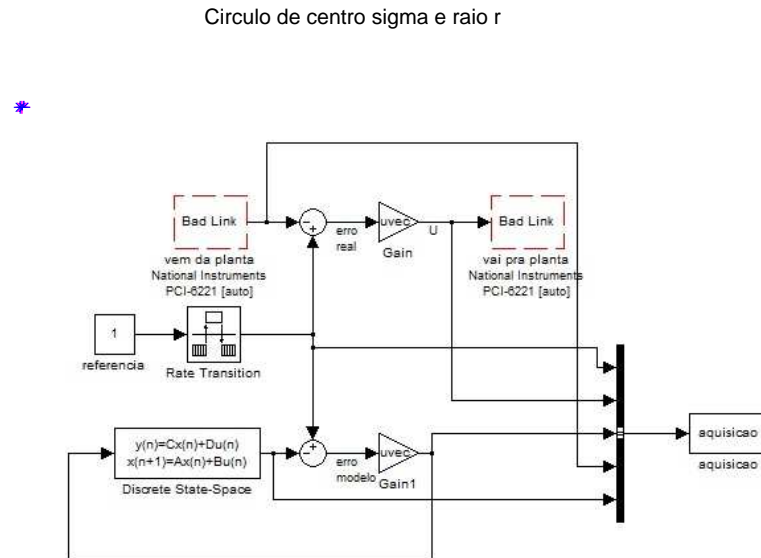


Figura 7.5: Programa para implementação do controlador na planta real

Tal programa faz a leitura do sensor da planta envia para o computador que projeta o controlador e aplica-o na planta real. A comunicação entre o computador e a planta foi feita através de uma placa de aquisição da *National Instruments*.

Embora o programa de implementação tenha sido feito, não foi possível implementar o controle em tempo real devido ao atraso na execução das etapas anteriores. A construção dos circuitos eletrônicos por exemplo, demandou um tempo maior do que o previsto no cronograma de trabalho, já que houve dificuldade na compra dos componentes, bem como problemas em alguns circuitos que tiveram que ser corrigidos.

Conclusão

A técnica de controle tratada nesse trabalho, requer bastante estudo por se tratar de um método novo e de difícil entendimento. Foram integradas algumas metodologias e projetado um controlador por realimentação de saída. Foram feitos vários testes para um sistema qualquer precisamente conhecido, e foram obtidos bons resultados, conseguindo alocar os pólos do sistema em malha fechada onde desejava-se. A parte eletrônica da placa móvel acionada por um fluxo de ar foi refeita, e está funcionando adequadamente conforme o desejado. Após o bom funcionamento da planta, foi obtido um modelo discreto precisamente conhecido através de modelagens de diferentes regiões usando o valor médio encontrado para os parâmetros. O modelo obtido possui um comportamento semelhante ao real, o que pode ser verificado comparando esse com a curva real da planta. Foi encontrado também um modelo incerto discreto no tempo, e projetado um controlador por realimentação de saída. O resultado foi conforme o esperado, ou seja, conseguiu-se alocar os pólos do sistema em malha fechada dentro do círculo de raio e centro especificados melhorando a estabilidade do sistema.

8.1 Perspectivas Futuras

Como projeto futuro pode-se acrescentar a ação integral como mais uma metodologia para o projeto de um controlador por realimentação de saída e aplicá-lo na planta. Além disso pode-se usar critérios de desempenho para avaliar os controladores e compará-los com outras técnicas de controle.

Códigos

Programa principal

```
% PROGRAMA PRINCIPAL DA MODELAGEM DO VENTILADOR – TCC

clear all
close all
clc

load 'aquisicao2.mat'

%dados coletados a uma taxa de 50 pontos por segundo
tensaoin = aquisicao2(:,3); %tansao em V
degrauin = aquisicao2(:,6); % degrau em V

tensao = decimate(tensaoin,5,10,'fir ');
degrau = decimate(degrauin,5,10,'fir ');

f=10; % pontos por segundo
t=1/f;
tempo=(0:t:length(degrau)/f-t)';

% função q plota os dados coletados
% plotaDadosColetados(tensao ,degrau ,tempo)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Funções MODELAGEM

[tau1 ,k1 ,sys1 ,theta1 ,syspade1]=modelagem1 ( tensao , degrau , tempo );

[tau2 ,k2 ,sys2 ,theta2 ,syspade2]=modelagem2 ( tensao , degrau , tempo );

[tau3 ,k3 ,sys3 ,theta3 ,syspade3]=modelagem3 ( tensao , degrau , tempo );
```

```

[tau4 , k4 , sys4 , theta4 , syspade4]=modelagem4 ( tensao , degrau , tempo );

[tau5 , k5 , sys5 , theta5 , syspade5]=modelagem5 ( tensao , degrau , tempo );

[tau6 , k6 , sys6 , theta6 , syspade6]=modelagem6 ( tensao , degrau , tempo );

tau=[tau1 tau2 tau3 tau4 tau5 tau6 ];
theta=[theta1 theta2 theta3 theta4 theta5 theta6 ];
k=[k1 k2 k3 k4 k5 k6 ];

% Funca q plota tau , theta e k de subida e descida
% plotaTauThetaK(tau , theta , k)

taumin = min(tau);
taumax = max(tau);
taumedio = mean(tau);

kmin = min(k);
kmax = max(k);
kmedio = mean(k);

thetamin = min(theta);
thetamax = max(theta);
thetamedio = mean(theta);

% calcula o modelo medio continuo com os parametros medios
%de tau , theta e k
[syspademedio]=modelomedio ( taumedio , kmedio , thetamedio
, degrau , tempo , tensao );

% Calcula modelo incerto
[Ai , Bi , Ci , Di , Ts]=novomodeloincerto ( taumin , taumax , kmin , kmax ,
thetamin , thetamax , thetamedio , taumedio );
%[Ai , Bi , Ci , Di]=modeloincerto ( Ts , taumin , taumax , kmin , kmax ,
thetamin , thetamax );

[ salvako ]=variako ( Ai , Bi , Ci , Di );
%
[rm , sm , km , kom , posrs , rsfactivel , rofactivel , Acl , avm , rfactivel ,
sfactivel]=encontraRmin ( Ai , Bi , Ci , Di , Ts );

```

Função que faz a modelagem de uma região

```

% MODELAGEM DA REGIÃO 1 – DEGRAU 1 – 60%(3.6V) A 80%(4.8V)
function [tau1 , k1 , sys1 , atraso1 , syspade1]=modelagem1 ( tensao , degrau , tempo )

```

```
degrau1=degrau(81:180);
tensao1=tensao(81:180);
tempo1=tempo(1:length(degrau1));

%Deslocamento
degrau1_des=degrau1-mean(degrau1(1:20));
tensao1_des=tensao1-mean(tensao1(1:20));
tempo1_des=tempo(1:length(degrau1_des));

% Normalização
degrau1_norm=degrau1_des/mean(degrau1_des(70:end));
tensao1_norm=tensao1_des/mean(tensao1_des(70:end));
tempo1_norm=tempo1_des;

% % GRAFICOS
figure % REAL
plot(tempo1,degrau1)
hold on
plot(tempo1,tensao1,'r')
grid
xlabel('Tempo')
ylabel('Tensão de saída em V')
legend('Degrau aplicado','Tensao de saída')
title('Região 1')

figure % DESLOCADO
plot(tempo1_des,degrau1_des)
hold on
plot(tempo1_des,tensao1_des,'r')
grid
xlabel('Tempo')
ylabel('Tensão de saída em V')
legend('Degrau aplicado','Tensao de saída')
title('Região 1 deslocada')

figure % NORMALIZADO
plot(tempo1_norm,degrau1_norm)
hold on
plot(tempo1_norm,tensao1_norm,'r')
grid
xlabel('Tempo')
ylabel('Tensão de saída em V')
title('Região 1 normalizada')
legend('Degrau aplicado','Tensao de saída')

% Método da resposta complementar
```

```

z1=log(1-tensao1_norm./degrau1_norm);

% Grafico p escolher o trecho de melhor reta
figure
plot(tempo1_norm,z1,'r')
grid
title('Resposta Complementar de primeira ordem - Região 1')

atraso1 = 0.1; % em segundos
p1=polyfit(tempo1_norm(33:36),z1(33:36),1);

tau1=-1/p1(1);
%b1=p1(2);
k1 = mean(tensao1_des(70:end)/mean(degrau1_des(70:end)));

% Construção da função de transferencia
num1=k1;
den1=[tau1 1];
sys1=tf(num1,den1);
%%%%%%%%%%set(sys1,'InputDelay',atraso1)

Comparando modelo obtido com grafico real
figure
lsim(sys1,degrau1_des,tempo1_norm')
grid
hold on
plot(tempo1_norm,tensao1_des,'r')
legend('Modelo encontrado','Modelo real')
title('Comparação entre o modelo obtido e o gráfico real
- Região 1')

[numpade1,denpade1] = pade(atraso1,1) ;
atraso_pade1=tf(numpade1,denpade1);
syspade1=atraso_pade1*sys1;

```

Função que calcula o modelo médio

```

function [syspademedio]=modelomedio(taumedio,kmedio,thetamedio)

num=kmedio;
den=[taumedio 1];
sysmedio=tf(num,den);

[numpade,denpade] = pade(thetamedio,1) ;
atraso_pade=tf(numpade,denpade);
syspademedio=atraso_pade*sysmedio;

```

Função que calcula o modelo incerto

```

% Matrizes A, B, C em função dos parametros tau, theta, k e
Ts (minimo e maximo)
function [A,B,C,D,T]=novomodeloincerto(taumin,taumax,kmin,kmax,
thetamin,thetamax, thetamedio,taumedio)

taucl = taumin/2; % deseja-se uma constante de tempo 2 x
mais rapida
T = taucl/5; % deve-se dividir o tau de malha fechada por
um numero
de 5 a 10
tauaux=[taumin taumax];
kaux=[kmin kmax];
thetaaux=[thetamin thetamax];
a=2/thetamedio;
p=1/taumedio;
i=1;

% fazendo todas as combinações possíveis dos parametros
minimos e maximos
for x=1:2
    for y=1:2
        for z=1:2
            vertice{i}=[tauaux(x) kaux(y) thetaaux(z)];
            i=i+1;
        end
    end
end

% definindo as matrizes incertas
for v=1:8
    tau=vertice{v}(1);
    k=vertice{v}(2);
    theta=vertice{v}(3);

    p=1/tau;
    a=2/theta;
    %Discretização do valter
    b0=-(a+p)*exp(-a*T)+2*p*exp(-p*T)+(a-p)*exp(-(a+p)*T);
    b1=a*(1+2*exp(-a*T)+exp(-p*T))-p*(1+3*p*exp(-p*T));
    b2=0;
    a0=(a-p)*exp(-a*T-p*T);
    a1=(a-p)*(-exp(-a*T-p*T));
    a2=a-p;

```

```

    sysinc{v}=tf([k*b2 k*b1 k*b0],[a2 a1 a0]);
    aux{v}=ss(sysinc{v});
    A{v}=aux{v}.a;
    B{v}=aux{v}.b;
    C{v}=aux{v}.c;
    D{v}=aux{v}.d;

```

```
end
```

Função que varia parâmetros para cálculo de K_o

```

function [salvako]=variako(Adisc,Bdisc,Cdisc,Ddisc)

A{1}=Adisc;
B{1}=Bdisc;
C{1}=Cdisc;
D{1}=Ddisc;

n=size(A{1},1);
i=1;

% Encontra ko
nomedoarquivo = 'Deus8x.mat';
if ~exist(nomedoarquivo,'file')

for sigmako=0:0.05:0.9%0:0.05:0.9
    rmax=1-abs(sigmako);
    for r=0.05:0.05:rmax%0.005:0.05:rmax;%
        [ko]=encontraKo(A,B,C,sigmako,r,'sedumi');
        %LMI q encontra ko
        salvako{i}=ko; %salva ko
        salvas(i)=sigmako;% salva o sigma
        salvar(i)=r;% salva o raio
        i=i+1;
    end
end

save(nomedoarquivo,'salvako','salvas','salvar');

else
    load(nomedoarquivo);
end

```

Função que calcula k_o

```

function [ko]=encontraKo(A,B,C,sigma,r,metodo)

if nargin<6
    metodo='sedumi';
end

% Dimensao
n=size(A{1},1);
m=size(B{1},2);
p=size(C{1},1);

%Variaveis
W=sdpvar(n,n,'symmetric');
Zo=sdpvar(m,n,'full');

%%

%linha 1
m11=-W;
m12=(W*A{1}'+Zo'*B{1}'+sigma*W)/r;

%linha 2
m21=m12';
m22=-W;

M{1}=[m11 m12;m21 m22];

%Declara LMI
LMIs=set(W>0,'W>0');
LMIs = LMIs+set(M{1}<0,'LMI principal <0');

options=sdpsettings('verbose',0,'warning',0,'solver',metodo,
'showprogress',1);

%chama solver
solucao=solvesdp(LMIs,[],options);

factibilidade=min(checkset(LMIs));

if factibilidade>0
    flag=1;
    Ws=double(W);
    Zos=double(Zo);
    kot=inv(Ws)*Zos';
    ko=kot';
else
    flag=0;

```

```

    k = [];
end

```

Função de varia parâmetros para cálculo de k

```

function [rm,sm,km,kom, posrs , rsfactivel , rofactivel , Acl ,avm,
rfactivel , sfactivel]=encontraRmin( Adisc , Bdisc , Cdisc , Ddisc , Ts)

% close all
% clear all
% clc
% Adisc=[0 1 0;0 0 1;-0.6922 1.6922 -1.0000];
% Bdisc=[0 0 1]';
% Cdisc=[-0.0753 0.0753 0];
% Ddisc=0;
% Ts=0.0337;

A{1}=Adisc;
B{1}=Bdisc;
C{1}=Cdisc;
D{1}=Ddisc;

load 'Deus8x.mat'

ko=salvako;% ko originado a partir do prog variako
ro=salvar;
f=length(ro);
n=size(A{1},1);
rm=2;
i=1;
cont=1;

nomedoarquivo = 'teste8x.mat';
if ~exist(nomedoarquivo, 'file')
    % Encontra k de realimentacao de saida
    for j=1:f
        for sigma=0.9:-0.1:0
            raio=abs(1-sigma);
            [flag , ks]=sinteseAo(A,B,C,ko{j},sigma , raio , 'sedumi');
            %LMI q encontra k (realimentação de saída)

            %Guardando os valores de cada iteração
            guardaR(i)=raio; % da realim. de saída
            guardaflag(i)=flag; % da realim. de saída
            guardak{i}=ks; % k da realim. de saída
            guardako{i}=ko{j}; % k da realim. de estados
        end
    end
end

```

```

if guardaflag(i)==1 % se for factivel
    rofactivel(cont)=ro(j); %[ro rin(j)];
    % armazena raio p realimentação de estados
    rsfactivel(cont)=guardaR(i); %[rr guardaR(i)];
    % armazena raio p realimentação de saída
    Aclfactivel{cont}=A{1}+B{1}*ks*C{1}; % mf
    avfactivel{cont}=eig(Aclfactivel{cont}); % av da mf
    sfactivel(cont)=(sum(real(avfactivel{cont}))/n);
    rfactivel(cont)=max(sqrt((sfactivel(cont)-
    real(avfactivel{cont}).^2
    +imag(avfactivel{cont}).^2)));
    % raio dos autovalores
    kfactivel{cont}=guardak{i};
    kofactivel{cont}=guardako{i};

    % %Armazenando o r minimo e seus
    paramentos correspondentes
    if rfactivel(cont)<rm
        rm=rfactivel(cont);
        sm=sfactivel(cont);
        avm=avfactivel{cont};
        Acl=Aclfactivel{cont};
        km=kfactivel{cont};
        kom=kofactivel{cont};
        posrs=cont;
    end
    cont=cont+1;
end
i=i+1;
end
end
save(nomedoarquivo, 'rm', 'sm', 'km', 'kom', 'posrs', 'rsfactivel',
'rofactivel', 'Acl', 'avm', 'rfactivel', 'sfactivel');
else
    load(nomedoarquivo);
end

disp('rmin')
disp(rm)
disp('sigma min')
disp(sm)
disp('k min')
disp(km)
disp('ko min')
disp(kom)

```

```

% Realimentação de saída usando o k factível encontrado
Bcl=-B{1}*km;
Ccl=C{1};
Dcl=D{1};

sysMF=ss(Acl,Bcl,Ccl,Dcl,Ts); % sistema em malha fechada
% G=tf(sysMF);
% ganho=dcgain(G);
% modulom=abs(avm);

% Malha aberta
sysMA=ss(A{1},B{1},C{1},D{1},Ts); %sistema em malha aberta
avMA=eig(A{1});
sigmaMA=(sum(real(avMA)))/n; % sigma
raioMA=max(sqrt((sigmaMA-real(avMA)).^2+imag(avMA).^2));

figure
subplot(211)
step(sysMA)
grid
legend('Malha aberta')

subplot(212)
step(sysMF)
grid
legend('Malha Fechada')

%Utilizando a funcao circulo p verificar se os polos estao
dentro do circulo de raio r
y=zeros(1,n);
folga=0.0001;
figure
[cir1] = circulo(-sigmaMA,raioMA+folga,'k',1); % malha aberta
hold on
[cir2] = circulo(-sm,rm+folga,'g',1); % malha fechada
hold on
[cir3] = circulo(0,1,'r',1); % circulo unitario
hold on
if imag(avm)==0
    plot(avm,y,'*m') % av malha fechada
    hold on
    plot(avMA,y,'*b') % av malha aberta
else
    plot(avm,'*m') % av malha fechada
    hold on
    plot(avMA,'*b') % av malha aberta
end
end

```

```

grid
axis('equal')
%legend('Malha aberta','Malha fechada','Circulo de raio unitario')
title('Circulo de centro sigma e raio r')

```

Função que calcula k

```

function [flag,k]=sinteseAo(A,B,C,ko,sigma,r,metodo)

if nargin<7
    metodo='sedumi';
end

% Dimensao
n=size(A{1},1);
m=size(B{1},2);
p=size(C{1},1);

%Variaveis
P=sdpvar(n,n,'symmetric');
G=sdpvar(m,m,'full');
F1=sdpvar(n,n,'full');
F2=sdpvar(n,n,'full');
F3=sdpvar(m,n,'full');
F4=sdpvar(n,n,'full');
L=sdpvar(m,p,'full');

%% Malha fechada usando o ko dado no artigo, com alocao de polos.
Ao{1}=(A{1}+B{1}*ko+sigma*eye(n))/r;

% No exemplo, F1 e F3 sao zero
% F1=zeros(n);
% F3=zeros(m,n);

%linha 1
m11=F1*Ao{1}+Ao{1}'*F1'-P;

%linha 2
m21=F2*Ao{1};
m22=-P;

%linha 3
m31=F3*Ao{1}+B{1}'*F1'+(L*C{1}-G*ko)/r; % r do denominador de
corrente da alocaçã;  $\frac{1}{2}$  ao de polos
m32=B{1}'*F2';
m33=F3*B{1}+B{1}'*F3'-(G+G');

```

```

%linha 4
m41=F4* $A_{1}$ -F1';
m42=P-F2';
m43=F4* $B_{1}$ -F3';
m44=-F4-F4';

%linha 1
m12=m21';
m13=m31';
m14=m41';

%linha 2
m23=m32';
m24=m42';

%linha 3
m34=m43';

M{1}=[m11 m12 m13 m14;m21 m22 m23 m24;m31 m32 m33 m34;
m41 m42 m43 m44];

%Declara LMI
LMIs=set(P>0,'P>0');
LMIs = LMIs+set(M{1}<0,'LMI principal <0');
%options = sdpsettings('savesolveroutput',1,'verbose',0,
'warning',1,'solver',metodo,'showprogress',0);
options=sdpsettings('verbose',0,'warning',0,'solver',metodo,
'showprogress',1);

%chama solver
solucao=solvesdp(LMIs,[],options);

factibilidade=min(checkset(LMIs));

if factibilidade>0
    flag=1;
    F1s=double(F1);
    F2s=double(F2);
    F3s=double(F3);
    F4s=double(F4);
    Ps=double(P);
    Gs=double(G);
    Ls=double(L);
    k=inv(Gs)*Ls;
else

```

```
    flag=0;  
    k=[];  
end
```


Bibliografia

- S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- G. Garcia, P. Pradin, and F. Zeng. Stabilization of discret time linear systems by static output feedback. *IEEE Trans. Auto. Control*, 2001.
- J. Löfberg. Yalmip : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004. URL <http://control.ee.ethz.ch/~joloef/yalmip.php>.
- D. Mehdi, E. K. Boukas, and O. Bachelier. Static output feedback design for uncertain linear discrete time systems. 2004.
- H. R. Moreira. Realimentação de saída robusta a partir de controladores dependentes de parâmetros para sistemas lineares incertos discretos no tempo. Master's thesis, Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e Computação, 2009.
- J. F. Sturm. Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. *Optimization Methods and Softwares*, 11-12 : 625-653, 1999.
- V. L. Syrmos, C. T. Abdallah, P. Dorato, and K. Grigoriadis. Static output feedback — a survey. *Automatica*, 33(2):125–137, February 1997.
- A. Trofino. Controle robusto. *Programa de Pós Graduação em Engenharia Elétrica*, 2000.