



Universidade Federal
do Rio de Janeiro

Escola Politécnica

AVALIAÇÃO DE ESTRATÉGIAS DE CONTROLE POR BUSCA EXTREMAL:
TEORIA E EXPERIMENTOS

Eduardo Rizzo Soares Mendes de Albuquerque

Projeto de Graduação apresentado ao Curso de Engenharia de Controle e Automação da Escola Politécnica, Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Engenheiro.

Orientador: Alessandro Jacoud Peixoto

Rio de Janeiro
Setembro de 2013

Albuquerque, Eduardo Rizzo Soares Mendes de
Avaliação de Estratégias de Controle por Busca
Extremal: Teoria e Experimentos/Eduardo Rizzo Soares
Mendes de Albuquerque. – Rio de Janeiro: UFRJ/ Escola
Politécnica, 2013.

XII, 79 p.: il.; 29, 7cm.

Orientador: Alessandro Jacoud Peixoto

Projeto de Graduação – UFRJ/ Escola Politécnica/
Curso de Engenharia de Controle e Automação , 2013.

Referências Bibliográficas: p. 77 – 79.

1. Controle por Busca Extremal. 2. Otimização em
Tempo-Real. 3. Sistemas Não-Lineares Incertos. 4.
Detecção de Cores. I. Peixoto, Alessandro Jacoud. II.
Universidade Federal do Rio de Janeiro, Escola Politécnica,
Curso de Engenharia de Controle e Automação . III.
Título.

Agradecimentos

Aos meus pais Paulo Sérgio Mendes de Albuquerque e Cristina Menezes Rizzo Soares pelo amor incondicional e atenção que me dedicaram todos esses anos. Sou muito grato pela preocupação que tiveram tanto com a minha formação social quanto com a minha formação acadêmica.

A Gustavo Rizzo Albuquerque, meu irmão, grande companheiro e, obviamente, meu fã (hehehe...).

Aos professores Otto Carlos Muniz Bandeira Duarte e Luiz Eduardo Azambuja Sauerbronn que foram além dos papéis que lhes fora designado. Muito obrigado por todos os seus conselhos e conversas.

Ao meu orientador Alessandro Jacoud Peixoto pelas inúmeras horas que me dedicou (às vezes, até mesmo em fim de semana!) e pela genuína boa vontade e entusiasmo que sempre demonstrou ao ajudar a mim e todos os seus alunos em geral.

A todos os meus amigos, que tornaram a jornada até agora certamente mais divertida, e cuja admiração que tenho por cada um deles me incentiva a me superar e melhorar todos os dias.

Resumo do Projeto de Graduação apresentado à Escola Politécnica/ UFRJ como parte dos requisitos necessários para a obtenção do grau de Engenheiro de Controle e Automação.

Avaliação de Estratégias de Controle por Busca Extremal: Teoria e Experimentos

Eduardo Rizzo Soares Mendes de Albuquerque

Setembro/2013

Orientador: Alessandro Jacoud Peixoto

Curso: Engenharia de Controle e Automação

A busca por eficiência e o aumento da complexidade dos sistemas de engenharia têm aumentado a necessidade de soluções de otimização em tempo-real e de adaptação às incertezas desses sistemas. Nesse contexto, controle por busca extremal tem despertado interesse de pesquisadores por se tratar de um método de controle robusto capaz maximizar ou minimizar uma função de performance de uma planta, sem requerer conhecimento explícito de seu modelo.

No trabalho foram estudados e avaliados alguns dos diferentes métodos existentes de controle por busca extremal. Para tanto, considerou-se aspectos teóricos, simulações numéricas e um experimento prático que consistiu no controle da posição angular de um alvo fixado ao eixo de um motor, visando maximizar a área projetada desse alvo no plano da imagem capturada por uma câmera. Ao fim, foi proposta uma pequena modificação do esquema de controle de forma a melhorar seu desempenho em regime permanente.

Palavras-chave: Controle por Busca Extremal, Otimização em Tempo-Real, Sistemas Não-Lineares Incertos, Detecção de Cores

Abstract of Undergraduate Project presented to POLI/UFRJ as a partial fulfillment of the requirements for the degree of Engineer.

AVALIAÇÃO DE ESTRATÉGIAS DE CONTROLE POR BUSCA EXTREMAL:
TEORIA E EXPERIMENTOS

Eduardo Rizzo Soares Mendes de Albuquerque

September/2013

Advisor: Alessandro Jacoud Peixoto

Course: Control and Automation Engineering

The search for efficiency and the increasing complexity of engineering systems are demanding adaptable real-time optimization solutions. In this context, Extremum Seeking Control has attracted attention of researchers as a real-time optimization approach able to minimize and maximize a performance function of a plant without requiring explicit knowledge of its model.

In this work, we have addressed some of the different methods of extremum seeking control. We have considered theoretical aspects, numerical simulations and an experiment, which consists in controlling the angular position of a target attached to the shaft of a DC motor in order to maximize the target area observed by a camera. Finally, we suggested a small modification in the control scheme in order to improve its performance in steady state.

Keywords: Extremum Seeking Control, Real-Time Optimization, Non-Linear Systems with Uncertainties, Color Detection

***AVALIAÇÃO DE ESTRATÉGIAS DE
CONTROLE POR BUSCA EXTREMAL :
TEORIA E EXPERIMENTOS***

EDUARDO RIZZO SOARES MENDES DE ALBUQUERQUE

*PROJETO DE GRADUAÇÃO SUBMETIDO AO CORPO DOCENTE DO CURSO
DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO DA ESCOLA POLITÉCNICA/
COPPE/ESCOLA DE QUÍMICA DA UNIVERSIDADE FEDERAL DO RIO DE
JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO
DO GRAU DE ENGENHEIRO DE CONTROLE E AUTOMAÇÃO.*

Aprovada por:

Prof. Alessandro Jacoud Peixoto , D.Sc.
(Orientador)

Prof. Tiago Roux de Oliveira, D.Sc.

Eng. Rodrigo Fonseca Carneiro, B.Sc.

Rio de Janeiro, RJ - Brasil

Setembro de 2013

Sumário

Lista de Figuras	x
1 Introdução	1
1.1 Breve Histórico	2
1.2 Aplicações	3
1.3 Justificativa	6
1.4 Organização do Trabalho	7
2 Busca Extremal: Fundamentos	9
2.1 Estratégias de Controle Não Baseadas em Deslizamento	9
2.1.1 Método das Perturbações Senoidais	9
2.1.2 Método da Função Periódica	15
2.2 Estratégias de Controle Baseadas em Deslizamento	19
2.2.1 Controle Extremal por Modos Deslizantes Via Função de Cha- veamento Periódica	19
3 Implementação Prática: Descrição do Sistema	27
3.1 Visão Geral do Experimento	27
3.2 Funcionamento do Software	30
3.2.1 Algoritmo de Visão	33
3.2.2 Comunicação e Tratamento de Dados	35
3.2.3 Interface Gráfica	37
4 Identificação, Simulações e Experimentos	39

4.1	Identificação do Modelo e da Função Objetivo	39
4.2	Simulações - Método das Perturbações Senoidais	44
4.2.1	Projeto	45
4.2.2	Planta de Grau Relativo 0	47
4.2.3	Planta de Grau Relativo 2	49
4.3	Simulações - Controle Extremal por Modos Deslizantes Via Função de Chaveamento	52
4.3.1	Projeto	53
4.3.2	Planta de Grau Relativo 0	54
4.3.3	Planta de Grau Relativo 2	57
4.4	Resultados Experimentais	61
4.4.1	Método das Perturbações Senoidais	61
4.4.2	Controle Extremal por Modos Deslizantes Via Função de Cha- veamento	66
4.5	Proposta para Atenuar a Oscilação em Regime	70
5	Conclusões e Trabalhos Futuros	74
	Referências Bibliográficas	77

Lista de Figuras

1.1	Esquema da bancada do experimento. Desejamos que o plano do alvo fique perpendicular ao eixo óptico da câmera	2
1.2	Foto da bancada	2
1.3	Número de publicações de ESC por década. Adaptado de [2].	4
2.1	Modelo ESC perturbação senoidal. Retirado de [3]	10
2.2	Desempenho da saída no método da perturbação senoidal.	14
2.3	Diagrama de blocos do controle extremal por Modos Deslizantes Via Função de Chaveamento Periódica	20
2.4	Análise das variáveis de controle.	24
2.5	Efeito da variação dos parâmetros.	25
3.1	Outra foto da bancada	28
3.2	Esquema da bancada	29
3.3	(a)Eletrônica do microcontrolador (b)circuito de potência	30
3.4	Esquema da bancada	32
3.5	Esquema da bancada	32
3.6	Etapas do processamento de imagem. (a)imagem original (b)componente R (c)escala de cinza (d)resultado subtração (e)imagem binária	34
3.7	Interface com o usuário	38
4.1	Resultado do rastreamento	40
4.2	Simulink do Motor + Controle P	42
4.3	Comparação do Modelo Identificado com a Realidade	43

4.4	Os dois modelos e os pontos obtidos experimentalmente	43
4.5	Comparação do modelo cossenoidal e polinomial.	44
4.6	Simulink do Método das perturbações senoidais	45
4.7	Simulação com grau 0	47
4.8	Estimação do gradiente	48
4.9	Segunda simulação com grau 0	48
4.10	Simulação com grau 2.	49
4.11	Segunda simulação com grau 2.	50
4.12	Simulação diminuindo o transitório.	51
4.13	Simulação instável!	51
4.14	Modelo em Simulink de Controle Extremal por Modos Deslizantes Via Função de Chaveamento	52
4.15	Bloco ESC do modelo em Simulink	52
4.16	Simulação 1 com grau 0.	54
4.17	Gráfico theta X eta para simulação 1.	55
4.18	Simulação 2 com grau 0.	56
4.19	Gráfico theta X eta para simulação 2.	56
4.20	Simulação com grau 0 e extremo distante.	57
4.21	Simulação com grau 2.	58
4.22	Simulação com grau 2 e ponto ótimo mais distante.	59
4.23	Tentativa 1 de melhorar transitório.	60
4.24	Tentativa 1 de melhorar transitório. Falha do controle.	60
4.25	Comparação do ângulo do motor com a referência fornecida por ESC.	62
4.26	Sinal da área.	62
4.27	Sinal de controle do motor.	63
4.28	Comparação do ângulo do motor com a referência fornecida por ESC.	64
4.29	Sinal da área.	65
4.30	Sinal de controle do motor.	65
4.31	Comparação do ângulo do motor com a referência fornecida por ESC.	67

4.32 Sinal da área.	67
4.33 Sinal de controle do motor.	68
4.34 Comparação do ângulo do motor com a referência fornecida por ESC.	69
4.35 Sinal da área.	69
4.36 Sinal de controle do motor.	70
4.37 Modelo Simulink com detecção de chegada ao regime	72
4.38 Simulação com detecção de chegada ao regime	73

Capítulo 1

Introdução

Controle por busca extremal (*Extremum Seeking Control - ESC*) é um método de controle robusto capaz de otimizar uma função não-linear de desempenho de uma planta. Esse método não requer conhecimento explícito da planta nem da função que se deseja otimizar, desde que se saiba que ela possui um extremo. Como será visto mais adiante, esse é um método de controle que vem ganhando cada vez mais atenção no meio acadêmico.

O objetivo deste trabalho é estudar algumas das técnicas existentes de controle extremal, analisar seu funcionamento e desempenho, e realizar algumas observações ainda não existentes na literatura científica. Para isso, serão realizadas simulações numéricas, assim como análise matemática e também um experimento prático. Ao final do trabalho, o trabalho contribuirá com uma proposta de adaptação dos esquemas de controle estudados, visando melhorar o desempenho em regime permanente.

O experimento prático proposto pelo autor nesse trabalho consiste em maximizar a área de um alvo vermelho visto por uma webcam ligada a um computador rodando um software escrito em linguagem Matlab. O alvo está fixo no eixo de rotação de um motor DC. Os métodos de controle por busca extremal serão usados de forma que, medindo-se apenas o ângulo do motor e a área do alvo vista pela webcam, possamos calcular um sinal de controle capaz de levar o alvo a posição desejada. Um desenho esquemático da bancada pode ser visto na figura 1.1 e foto pode ser vista e na figura 1.2.

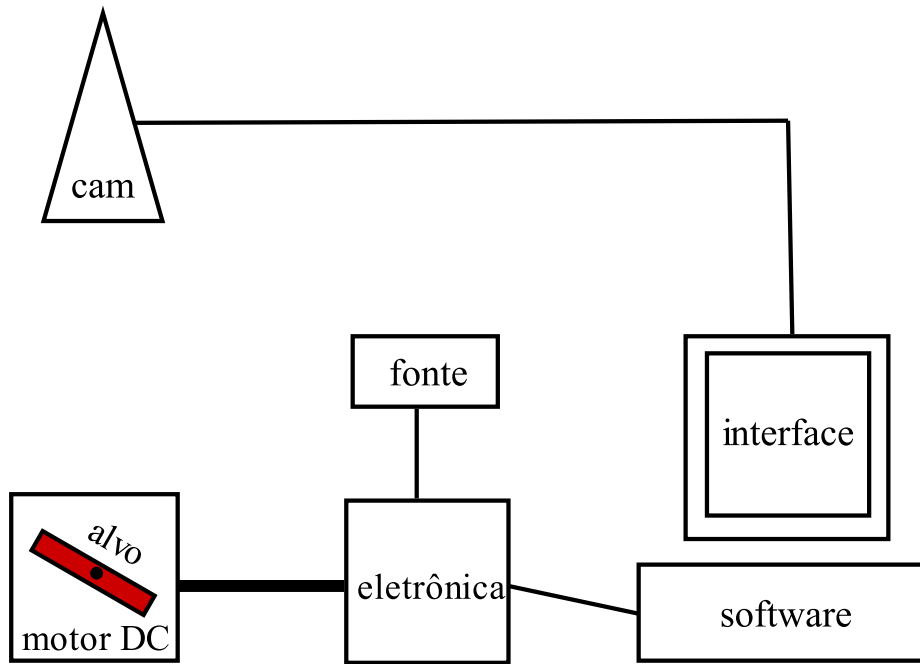


Figura 1.1: Esquema da bancada do experimento. Desejamos que o plano do alvo fique perpendicular ao eixo óptico da câmera

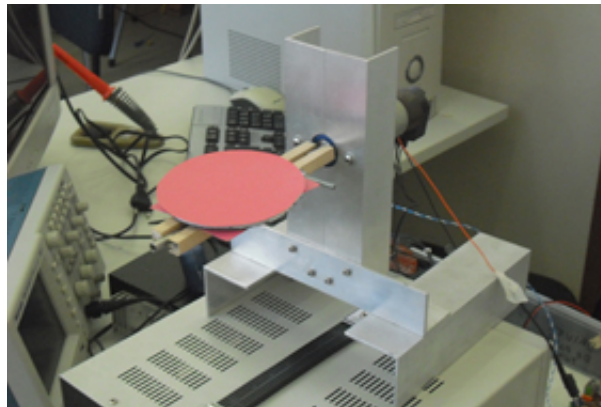


Figura 1.2: Foto da bancada

Neste capítulo, será feita uma breve revisão histórica de controle extremal, seguida da apresentação de algumas aplicações para motivar o leitor. Por fim, serão apresentadas as justificativas para este trabalho e sua organização.

1.1 Breve Histórico

A maioria dos métodos de controle adaptativo para sistemas lineares e não-lineares são aplicáveis em problemas de regulação onde são conhecidos o set-point ou a

trajetória de referência. Existe, no entanto, algumas aplicações onde existe uma função com um extremo (máximo ou mínimo) e é necessário descobrir qual é a referência que manterá a saída dessa função no seu extremo. Para isso, precisamos de algum tipo controle adaptativo capaz de gerar um sinal de referência que fará a saída de uma planta convergir para o extremo de uma função objetivo desconhecida. A revisão histórica a seguir é baseada em [1], [2], [3] e [4].

O primeiro artigo de controle por busca extremal foi escrito em 1922 por Leblanc [5]. No artigo, era proposto um esquema de controle que mantivesse máxima transferência de potência entre uma linha de transmissão e um bonde elétrico. Durante a segunda guerra mundial, controle extremal foi um importante tópico de pesquisa na Rússia. Em 1951 o livro de Draper e Li [6] foi a primeira publicação da literatura inglesa detalhando um algoritmo de controle extremal e estudando o seu desempenho, propondo um método para otimizar um motor de combustão interna. De 1970 a 2000, a pesquisa sobre o tema continuou, mas o foco dos estudos sobre controle adaptativo passaram para outras formas de controle. Foi só em 2000 com a publicação de [1] por Krstić e Wang que as estratégias de controle por busca extremal receberam uma nova atenção do meio acadêmico. No artigo, prova-se a estabilidade local de um método baseado em perturbação para toda uma classe geral de sistemas não-lineares. Em [7] estudou-se melhorias e limitações desse algoritmo de controle.

Em [2], podemos notar claramente esse reaparecimento do interesse acadêmico em ESC. No artigo, é dito que o número de publicações sobre controle extremal aumentou cerca de oito vezes em relação à década de 90, como pode ser visto no gráfico da figura 1.3.

1.2 Aplicações

O controle por busca extremal tem sido usado para otimizar sistemas onde existe a necessidade de se manter a saída de função de interesse no seu valor extremo. Por se tratar de um controle adaptativo e robusto à incertezas dos sistemas, não é necessário um conhecimento analítico da função objetivo nem da planta, desde

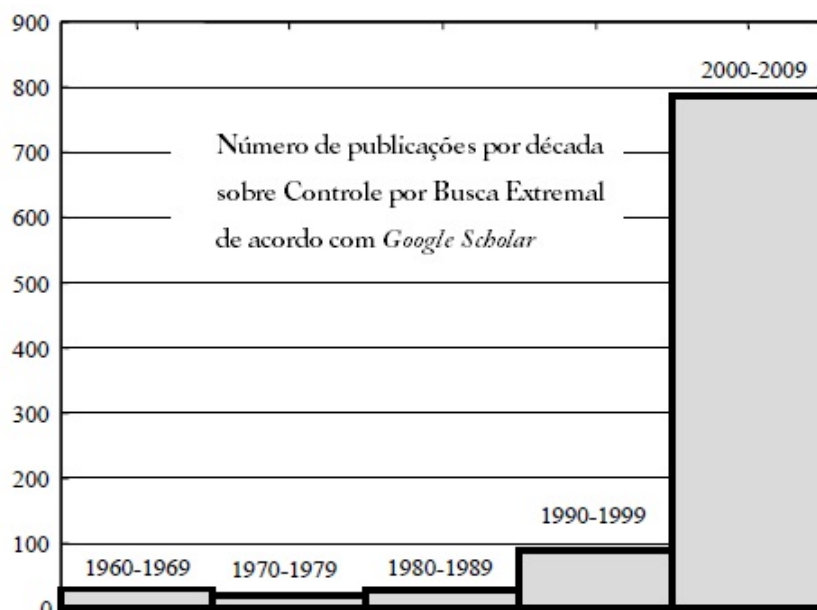


Figura 1.3: Número de publicações de ESC por década. Adaptado de [2].

que se possa medir a saída dessa função em tempo real. ESC se apresenta, então, como uma ferramenta poderosa para tratar problemas complexos onde os modelos possuem muitas incertezas.

Alguns algoritmos de otimização mais populares utilizam a informação da derivada ou gradiente da função objetivo, no entanto, em muitos problemas a obtenção dessas informações só pode ser obtida a um custo alto ou talvez nem seja possível obtê-las em tempo real. Além disso, a inserção de sensores ou derivadores, aumentam o custo de projeto e número de pontos de falha, podendo até mesmo amplificar o ruído em altas frequências, levando até mesmo os sistema à instabilidade [4]. Logo, controle por busca extremal também aparece como uma atraente alternativa para tornar sistemas mais eficientes e reduzir seus custos.

Em [8] foi estudada a possibilidade de usar esse método para otimizar a produção de petróleo em poços que operam por elevação artificial através da injeção de gás lift. O estudo seria uma alternativa aos métodos existentes hoje em dia, que em geral são lentos e não automatizados. O objetivo é encontrar e manter o sistema em malha fechada em torno do ponto ótimo da Curva de Produção de Petróleo, o que aumenta o Valor Presente Líquido do sistema e os lucros de operação. A modelagem deste problema envolve várias incertezas, e o controle por busca extremal aparece

como uma boa alternativa, já que se trata de um controle robusto que não necessita do conhecimento explícito da planta.

O projeto de freios ABS (*Antilock Braking System*) usando ESC, tem gerado grande número de publicações acadêmicas. No problema, o coeficiente de força de fricção alcança valor máximo para um valor pequeno não-nulo de deslizamento da roda. Essa função varia dependendo do asfalto onde o carro se encontra, exigindo que o controle se adapte a diferentes tipos de terreno. Sendo assim, esse é um problema onde existe um ponto-ótimo de entrada numa função que queremos maximizar e podemos usar controle por busca extremal para projetar controladores que alcançarão esse valor independentemente do conhecimento do terreno [3] [9] [10].

Uma outra aplicação que vem sendo bastante estudada é a navegação de veículos autônomos, que é um problema fundamental para a realização de outras tarefas em robótica móvel. Controle por busca extremal tem sido usado de forma a levar o veículo a uma certa posição desejada sem medir sua posição ou sua velocidade. Para isso, o veículo deve medir a potência do sinal de uma fonte. Sob a hipótese que o sinal da fonte será maior, quanto menor for a distância entre o veículo e ela, basta maximizar a função da potência medida. Essa fonte pode ser um sinal eletromagnético ou a concentração de um agente químico, por exemplo. Esse tipo de estudo pode reduzir o custo das aplicações ao descartar o uso de sensores para medir posição e velocidade. Além disso, quanto menos sensores, menor será o número de pontos de falha. O artigo [11] descreve o problema da navegação de um veículo com restrições no raio máximo de curva e confinado a uma certa região que deve buscar uma trajetória ótima até uma fonte móvel. O controle por busca extremal é usado para controlar o ângulo de uma antena que, medindo o sinal da fonte, deve apontar a sua direção. O ângulo da antena é usado então como valor de orientação do veículo. Como hipótese, a dinâmica da antena é muito mais rápida que a dinâmica do veículo. A proposta desse artigo é que esse estudo possa ser usado para melhorar a qualidade de comunicação de uma rede com agentes móveis.

Uma das aplicações mais importantes a nível industrial do controle extremal é

a otimização dos coeficientes do PID como em [12]. Nesse artigo é feita uma comparação de desempenho com outros métodos de sintonia de ganhos do controlador PID, como Ziegler-Nichols, mostrando que ESC tinha potencial para sintonizá-los melhor. Um dos seus possíveis usos está na regulação de motores. Em [13], deseja-se controlar o tempo de combustão do motor atuando nas válvulas de controle de temperatura. O controle por busca extremal por perturbação senoidal é usado para fornecer o *setpoint* ótimo, maximizando a eficiência do motor, e também ajustando os ganhos do controlador PID rapidamente.

Além disso, pode-se citar outras aplicações de ESC como a otimização de formação de vôo descrito no artigo [14], e também a otimização de bioreatores em [15]. Em [16], o autor mostrou ser possível usar esse algoritmo para diminuir o impacto em válvulas eletromecânicas em motores de combustão.

Em [4] uma nova proposta controle extremal via de função de monitoração foi aplicada em simulações de freio ABS e posicionamento de painel solar. Por meio de um experimento prático, o trabalho mostrou a viabilidade de usar essa estratégia no problema de *source seeking*.

1.3 Justificativa

Como foi visto anteriormente, controle por busca extremal vem ganhando cada vez mais atenção no meio acadêmico e várias aplicações estão sendo propostas. Além de ser capaz de levar um sistema para uma região em torno do ponto ótimo de uma função de interesse, como produção, lucro, eficiência ou outra medida desejada qualquer, o esquema muitas vezes possibilita utilizar menos sensores, reduzindo custos e, se supormos sistemas onde a falha de um único sensor é fatal, diminuindo o número de pontos de falha num sistema. Também trata-se um método de controle robusto que não necessita do conhecimento do modelo da planta, o que na prática é muito importante, visto que nem sempre esse modelo está disponível ou seu estudo pode ser muito custos, podendo mesmo assim estar sujeito a muitas incertezas. Sendo assim, é justificável e importante dedicar tempo à investigação do funcionamento

e das especificidades desse método de controle, assim como explorar maneiras de melhorá-lo.

Ademais, o experimento prático permitirá estudar o desempenho dessa estratégia de controle na presença de incertezas no modelo, ruídos nos sensores, dinâmicas não-modeladas durante o projeto do controlador e taxa de amostragem reduzida. A própria montagem da bancada ajudará a compreender as dificuldades da implementação prática desse controlador. Uma vez montado o experimento, é fácil perceber que a mesma ideia pode ser usada para resolver outros problemas técnicos que envolvam a maximização de um sinal de potência. Por exemplo, a área visível do alvo vermelho pode ser entendida como a potência do sinal de um satélite e desejamos posicionar uma antena de forma a receber potência máxima desse sinal. Outro exemplo é usar a soma da potência do sinal recebido por vários receptores móveis para posicionar uma antena de maneira a maximizar essa medida, garantindo uma recepção melhor para todos em média.

É desejado que ao fim do trabalho o experimento implementado possa se tornar uma bancada do Laboratório de Engenharia de Controle e Automação da UFRJ (LabECA). Essa é uma das razões que fez com que o *software* fosse implementado em Matlab, pois se trata de uma linguagem exaustivamente estudada pelos alunos do curso de Engenharia de Controle e Automação. O código foi escrito de forma que pudesse ser facilmente modificado depois pelos alunos e também foi implementada uma pequena interface gráfica para facilitar a realização de experimentos.

1.4 Organização do Trabalho

O trabalho está organizado da seguinte maneira: no capítulo 2 serão apresentados e estudados os métodos de controle por busca extremal que serão avaliados nesse trabalho. O capítulo 3 será dedicado a descrição da implementação prática, tal como suas especificidades e limitações.

O capítulo 4 começará com a identificação do modelo da planta e da função objetivo. Em seguida, serão projetados dois controladores usando dois métodos di-

ferentes de busca extremal. Os dois métodos serão avaliados por meio de simulações com e sem a presença da dinâmica da planta. Ainda no mesmo capítulo, será feita uma análise dos dados experimentais coletados. O capítulo se encerrará com a avaliação, por meio de simulações, de uma proposta de modificação dos controladores utilizados, visando melhorar a performance em regime permanente.

Por fim, o capítulo 5 descreverá as conclusões obtidas e comentará possíveis trabalhos futuros.

Capítulo 2

Busca Extremal: Fundamentos

Neste capítulo, serão apresentados os métodos de controle por busca extremal que serão estudados, avaliados e implementados neste trabalho. A análise mostrada aqui vale para plantas de grau relativo 0, ou seja, plantas cuja a dinâmica pode ser desprezada. Mais adiante, nos capítulos finais do trabalho, será avaliada a eficiência desses controladores ao adicionarmos plantas de grau relativo maior que 0.

Este capítulo está dividido em duas seções. A primeira estudará estratégias de controle de busca extremal não baseadas em deslizamento e a segunda estudará estratégia de controle de busca extremal com presença de deslizamento.

2.1 Estratégias de Controle Não Baseadas em Deslizamento

2.1.1 Método das Perturbações Senoidais

O método da perturbação senoidal é o mais popular dos métodos de controle por busca extremal e é caracterizado pela sua simplicidade e rápida adaptação. Ele é de fácil implementação e não necessita de conhecimento explícito da planta nem da função objetivo [3]. É necessário, no entanto, que planta seja estável. O diagrama de blocos pode ser visto na figura 2.1.

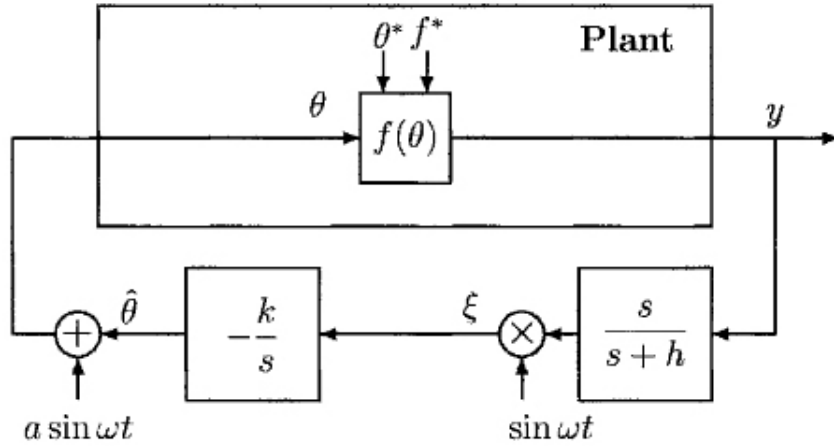


Figura 2.1: Modelo ESC perturbação senoidal. Retirado de [3]

Suponha que queiramos otimizar a saída de uma função $y = f(\theta)$ do tipo:

$$f(\theta) = f^* + \frac{f''}{2}(\theta - \theta^*)^2 \quad (2.1)$$

onde $f'' > 0$. Qualquer função do tipo C^2 pode ser aproximada localmente para uma função do tipo 2.1. A hipótese $f'' > 0$ também é feita sem perda de generalidade. Caso $f'' < 0$, basta invertermos o sinal de k . O objetivo do algoritmo será fazer $\theta - \theta^*$ o menor possível para levar a saída $f(\theta)$ para seu valor extremo.

A perturbação senoidal ajuda a estimar o gradiente da função $f(\theta)$. Somada a $\hat{\theta}$, ela gera a entrada da função objetivo. A saída y da função custo passa por um filtro passa-alta, que ajuda a retirar a componente DC do sinal, e em seguida é multiplicada por $\sin(\omega t)$ num processo de demodulação do sinal. O resultado é o sinal ξ que é uma estimativa do gradiente da função $f(\theta)$ que passa pelo integrador para gerar o sinal $\hat{\theta}$, a estimativa do mínimo local caso $k > 0$, ou a estimativa do máximo local se $k < 0$.

Os parâmetros do controlador são a , k , h e ω . Quanto maior a , maior será a amplitude do seno de busca, diminuindo as chances da busca ficar presa em um mínimo (ou máximo) local. No entanto, um aumento em a significa que o erro residual em regime será maior. Caso a diminua, as oscilações em torno de θ^* diminuirão também, em compensação, será mais fácil ficar preso em torno de um mínimo(ou

máximo) local. A variável k junto com a variável a influenciam na velocidade de convergência do algoritmo. A variável ω controla a separação entre a escala de tempo de estimação do processo, conduzida pelo integrador, e a estimação do gradiente, conduzida pelo seno multiplicativo e aditivo. Quanto maior for ω , melhor será a estimativa do gradiente e menor serão os efeitos dos harmônicos de ordem superior. Em contrapartida, mais rápidas serão as oscilações em torno do ponto ótimo, o que pode, na prática, reduzir a vida dos atuadores. Por fim, o parâmetro h é responsável pela filtragem da componente DC do sinal de saída y e deve ser menor que a frequência ω . O filtro passa-alta não é obrigatório, pois a demodulação do sinal também realiza essa tarefa, mas sua presença ajuda a filtrar a componente DC do sinal sem requerer que ω seja muito alto. [17]

A seguir é apresentada uma análise de estabilidade intuitiva para plantas de grau 0, baseado no modelo da figura 2.1. A prova rigorosa pode ser visto em [1] ou em [3].

Seja $\hat{\theta}$ o valor estimado da entrada ótima θ^* . Façamos

$$\tilde{\theta} = \theta^* - \hat{\theta}$$

que representa o erro de valor estimado. Logo,

$$\theta - \theta^* = a \sin(\omega t) - \tilde{\theta}$$

Substituindo na equação 2.1

$$y = f^* + \frac{f''}{2}(\tilde{\theta} - a \sin(\omega t))^2$$

Utilizando a identidade trigonométrica $2 \sin^2(\omega t) = 1 - \cos(2\omega t)$, temos:

$$\begin{aligned} y &= f^* + \frac{f''}{2} \tilde{\theta}^2 - a f'' \tilde{\theta} \sin(\omega t) + \frac{a^2 f''}{2} \sin^2(\omega t) \\ &= f^* + \frac{a^2 f''}{4} + \frac{f''}{2} \tilde{\theta}^2 - a f'' \tilde{\theta} \sin(\omega t) + \frac{a^2 f''}{4} \cos(\omega t) \end{aligned} \quad (2.2)$$

Aplicando o filtro passa-alta

$$\frac{s}{s+h}$$

na saída da planta, elimina-se a componente DC f^* , obtendo-se

$$\frac{s}{s+h}[y] \approx \frac{f''}{2} \tilde{\theta}^2 - a f'' \tilde{\theta} \sin(\omega t) + \frac{a^2 f''}{4} \cos(\omega t) \quad (2.3)$$

Multiplicando esse valor por $\sin(\omega t)$ o sinal é demodulado. Tem-se então

$$\xi \approx \frac{f''}{2} \tilde{\theta}^2 \sin(\omega t) - a f'' \tilde{\theta} \sin^2(\omega t) + \frac{a^2 f''}{4} \cos(\omega t) \sin(\omega t) \quad (2.4)$$

Novamente, aplicando a identidade $2 \sin^2(\omega t) = 1 - \cos(2\omega t)$ e também a identidade $2 \cos(2\omega t) \sin(\omega t) = \sin(3\omega t) - \sin(\omega t)$ obtemos

$$\xi \approx -\frac{a f''}{2} \tilde{\theta} + \frac{a f''}{2} \tilde{\theta} \cos(2\omega t) + \frac{a^2 f''}{8} (\sin(\omega t) - \sin(3\omega t)) + \frac{f''}{2} \tilde{\theta}^2 \sin(\omega t) \quad (2.5)$$

Como θ^* é constante,

$$\dot{\tilde{\theta}} = -\dot{\tilde{\theta}}$$

então

$$\tilde{\theta} \approx \frac{k}{s} \left[-\frac{a f''}{2} \tilde{\theta} + \frac{a f''}{2} \tilde{\theta} \cos(2\omega t) + \frac{a^2 f''}{8} (\sin(\omega t) - \sin(3\omega t)) + \frac{f''}{2} \tilde{\theta}^2 \sin(\omega t) \right] \quad (2.6)$$

Por se tratar de uma análise local, podemos negligenciar o último termo quadrático

da equação

$$\tilde{\theta} \approx \frac{k}{s} \left[-\frac{af''}{2} \tilde{\theta} + \frac{af''}{2} \tilde{\theta} \cos(2\omega t) + \frac{a^2 f''}{8} (\sin(\omega t) - \sin(3\omega t)) \right] \quad (2.7)$$

Sendo os dois últimos termos de alta frequência, ao passar pelo integrador, eles serão bastante atenuados, sobrando

$$\tilde{\theta} \approx \frac{k}{s} \left[-\frac{af''}{2} \tilde{\theta} \right] \quad (2.8)$$

ou

$$\dot{\tilde{\theta}} \approx -k \frac{af''}{2} \tilde{\theta} \quad (2.9)$$

Como $kf'' > 0$, o sistema é estável. Logo, $\tilde{\theta} \rightarrow 0$ e a entrada ótima estimada $\hat{\theta}$ convergirá a uma pequena região em torno de θ^* . Vale lembrar que essas aproximações são válidas desde que ω seja grande comparativamente aos parâmetros k, a, h e f'' .

O seguinte teorema resume as propriedades da malha básica de busca extremal.

Teorema 2.1 (Busca Extremal) *Para o sistema da figura 2.1 o erro da saída $y - f^*$ possui convergência exponencial local para uma vizinhança de ordem $O(a^2 + \frac{1}{\omega^2})$ da origem desde que a frequência ω seja suficientemente grande, e $\frac{1}{1+L(s)}$ seja assintoticamente estável, onde*

$$L(s) = \frac{kaf''}{2s} \quad (2.10)$$

Simulando esse esquema de controle num problema de minimização de uma função objetivo

$$y = f(\theta) = 10 + 0.5(\theta - 3)^2$$

onde $\theta^* = 3$ e $y^* = 10$, e usando os seguintes parâmetros de controlador $a = 1, w = 10, h = 1$ e $k = -1$, obtemos o seguinte sinal de y que pode ser visto no gráfico da figura 2.2

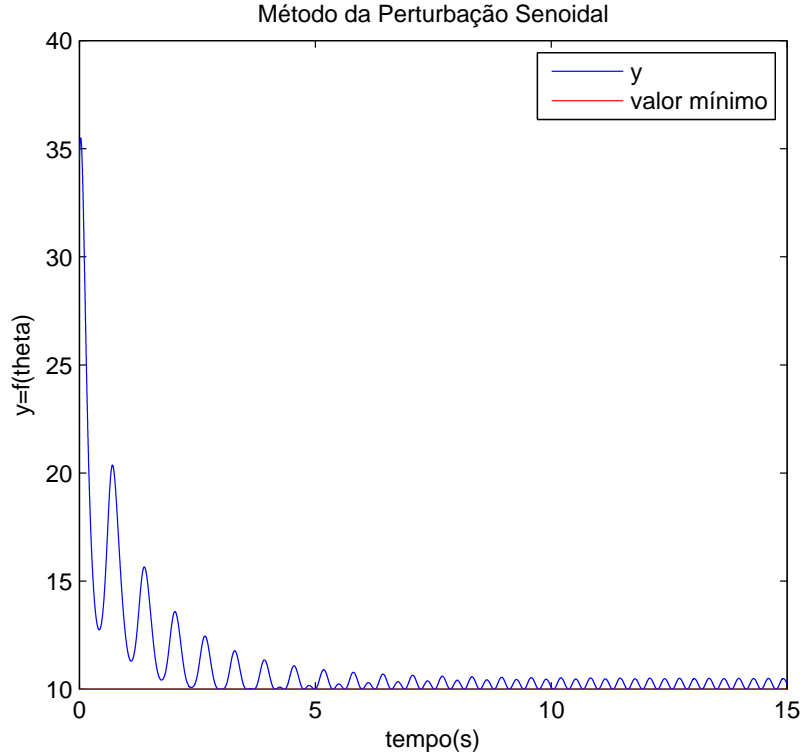


Figura 2.2: Desempenho da saída no método da perturbação senoidal.

A entrada da função custo foi levada com sucesso até seu valor ótimo e permaneceu oscilando em torno desse ponto.

Observação 1 (Esquema Simplificado do Método das Perturbações Senoidais)

Para análise futura, vale observar o resultado que obtemos ao simplificarmos essa estratégia de controle retirando o filtro passa-altas no caso do problema da maximização de $f(\theta)$. Seja u tal que $\theta = u$. Observe o diagrama de blocos da figura 2.1. Supondo que estamos tratando de um problema de maximização, devemos multiplicar a variável k por -1 . Além disso, passemos a perturbação senoidal, a $\sin(\omega t)$, para antes do bloco de integração. Então, obtemos

$$u = \int_0^t -\frac{a}{\omega} \cos(\omega t) + ky \sin(\omega t) dt \quad (2.11)$$

Essa equação será comparada mais adiante com o resultado do método da função periódica.

2.1.2 Método da Função Periódica

O método da função periódica servirá como estudo para melhor compreender o funcionamento o método de controle extremal por modos deslizantes via função de chaveamento periódica que será apresentado mais adiante.

Suponha que queiramos maximizar saída da função $y = f(\theta)$. O método da função periódica utiliza o seguinte esquema de controle:

$$\theta = u = \int_0^t \rho \sin(\omega t - y) dt \quad (2.12)$$

onde $\omega, \rho > 0$ são constantes. Esse esquema é válido para problemas de maximização. Caso queiramos minimizar a função $f(\theta)$, basta multiplicarmos y em 2.12 por -1.

Abaixo, será feita uma demonstração matemática intuitiva, mostrando que essa proposta de controlador deverá fazer com que a entrada de uma determinada função objetivo $f(\theta)$ convirja para uma região em torno do seu valor ótimo. Nessa seção será adotada a notação $x = \theta$.

A intenção dessa demonstração é mostrar que a média do sinal $x(t)$ somada a um pequeno sinal de dither d tende a x^* . Logo, x deverá oscilar em torno de x^* também. Como hipótese, assumiremos que o ganho do controlador ρ é muito menor que ω . Esse prova usa o conceito de *Averaging* que pode ser visto em [18] ou em [19].

Suponhamos a função objetivo $y = -x^2$. Então de 2.12 temos

$$\frac{dx}{dt} = \rho \sin(\omega t + x^2) \quad (2.13)$$

Criemos agora um novo sinal $\tilde{x}(t)$ tal que

$$\tilde{x} = x + d$$

Onde d é um dither de amplitude A do tipo $d = A \sin(\omega t)$. Por hipótese, A é muito

pequeno comparado aos outros parâmetros do problema. Logo, de 2.13

$$\dot{\tilde{x}} = \rho \sin(\omega t + (\tilde{x} - d)^2) + \dot{d}$$

$$\dot{\tilde{x}} = \rho \sin(\omega t + (\tilde{x} - d)^2) + A\omega \cos(\omega t)$$

$$\dot{\tilde{x}} = \rho[\sin(\omega t) \cos((\tilde{x} - d)^2) + \sin((\tilde{x} - d)^2) \cos(\omega t)] + A\omega \cos(\omega t) \quad (2.14)$$

Usando série de Taylor uma função $g(\cdot)$ pode ser aproximada por

$$g(u + v) = g(u) + \frac{g'(u)}{1!}v + \frac{g''(u)v^2}{2!} + \dots$$

Definindo

$$g_1(a) = \sin(a^2) \Rightarrow g'_1(a) = \cos(a^2)2a$$

$$g_2(a) = \cos(a^2) \Rightarrow g'_2(a) = -\sin(a^2)2a$$

Então, para pequenas variações em \tilde{x} temos:

$$\sin((\tilde{x} - d)^2) = g_1(\tilde{x} - d) = g_1(\tilde{x}) - g'_1(\tilde{x})d + r_1(\tilde{x}, d)$$

e

$$\cos((\tilde{x} - d)^2) = g_2(\tilde{x} - d) = g_2(\tilde{x}) - g'_2(\tilde{x})d + r_2(\tilde{x}, d)$$

onde r_1 e r_2 são termos de ordem $O(A^2)$ desprezíveis em comparação ao resto da expressão se assumirmos A suficientemente pequeno. De 2.14

$$\dot{\tilde{x}} = \rho[\sin(\omega t)(g_2 - g'_2d + r_2) + \cos(\omega t)(g_1 - g'_1d + r_1)] + A\omega \cos(\omega t)$$

$$\dot{\tilde{x}} = P_1 + P_2 + P_3 \quad (2.15)$$

onde

$$P_1 = \rho[\sin(\omega t)g_2 + \cos(\omega t)g_1] + A\omega \cos(\omega t)$$

$$P_2 = \rho[\sin(\omega t)r_2 + \cos(\omega t)r_1]$$

$$P_3 = \rho[-\sin(\omega t)g'_2d - \cos(\omega t)g'_1d]$$

Analisemos agora o comportamento médio desses termos durante um período $T = \frac{2\pi}{\omega}$. Primeiramente, observe que

$$\begin{aligned} \frac{1}{T} \int_0^T \sin(\omega t)d \, dt &= \frac{A}{T} \int_0^T A \sin(\omega t)^2 \, dt \\ &= \frac{A}{T} \int_0^T A \frac{(1 - \cos(2\omega t))}{2} \, dt = \frac{A}{2} \, dt \end{aligned}$$

e

$$\begin{aligned} \frac{1}{T} \int_0^T \cos(\omega t)d \, dt &= \frac{1}{T} \int_0^T A \sin(\omega t) \cos(\omega t)d \, dt \\ \frac{1}{T} \int_0^T \frac{1}{2} \sin(2\omega t)d \, dt &= 0 \end{aligned}$$

Calculemos, então, a média de cada termo P_i , onde

$$\overline{P}_i = \frac{1}{T} \int_0^T P_i dt$$

No caso de \overline{P}_1 , g_1 e g_2 dependem de \tilde{x} . Como $\rho \ll \omega$, \tilde{x} possui uma dinâmica muito mais lenta quando comparada a ωt , e seu valor pode ser considerado constante durante o período T . Dessa forma, g_1 e g_2 são constantes dentro integral. Resolvendo a integral, temos: $\overline{P}_1 = 0$

Em \overline{P}_2 , r_1 e r_2 são termos de ordem $O(A^2)$. Como $d = A \cos(\omega t)$ e A é muito pequeno, r_1 e r_2 são desprezíveis e $\overline{P}_2 = 0$.

Por fim, em \overline{P}_3 , as funções g'_1 e g'_2 também podem ser considerados constantes durante o período T , pela mesma razão vista anteriormente. Logo, resolvendo a integral

$$\overline{P}_3 = -\rho \frac{A}{2} g'_2$$

Isso significa que o comportamento da média de \tilde{x} tem comportamento próximo de \overline{P}_3 quando as hipóteses anteriores são verdadeiras.

Para estudar esse comportamento, criemos um sinal $\tilde{x}_{av}(t)$.

Lembrando que $g_2(\tilde{x}) = -2\tilde{x} \sin(\tilde{x}^2)$, concluímos que

$$\dot{\tilde{x}}_{av} = -\rho \frac{A}{2} (-2\tilde{x}_{av} \sin(\tilde{x}_{av}^2)) \quad (2.16)$$

Vamos analisar a estabilidade desse sistema. É fácil ver que equação tem um ponto de equilíbrio em 0. Seja $V(\tilde{x}_{av})$ uma função semi-definida positiva de Liapunov tal que

$$V = \frac{\tilde{x}_{av}^2}{2} \geq 0$$

de 2.16, sua derivada será

$$\dot{V} = \tilde{x}_{av} \dot{\tilde{x}}_{av}$$

$$\dot{V} = \rho A \tilde{x}_{av}^2 \sin(\tilde{x}_{av}^2)$$

Essa função é definida negativa se $\tilde{x}_{av} = (-\sqrt{\pi}, \sqrt{\pi})$ e $A < 0$, provando que a função $\tilde{x}_{av}(t)$ é localmente estável dentro da região $\tilde{x}_{av} < |\sqrt{\pi}|$ e tende ao ponto de equilíbrio $\tilde{x}_{av}(t) = 0$.

A equação 2.16 representa o comportamento médio da variável \tilde{x} sob as hipóteses citadas. É intuitivo dizer que \tilde{x} deverá tender a oscilar em torno de zero quando sua média tende a esse mesmo valor. Como $x = \tilde{x} - d$, onde d é um sinal de dither muito pequeno, $x(t)$ também deverá oscilar em torno de zero.

Observação 2 (Comparação com o método das perturbações senoidais) *Caso*

realizemos uma transformação trigonométrica $\sin(a+b) = \sin(a) \cos(b) + \sin(b) \cos(a)$ na equação 2.12 obtemos o seguinte equação $\dot{x}(t) = u$

$$u = \int_0^t -\rho \sin(y) \cos(\omega t) + \rho \cos(y) \sin(\omega t) dt$$

Para $\rho \ll \omega$ a dinâmica de $y(t)$ é muito lenta se comparada a ωt e $\sin(y)$ e $\cos(y)$ podem ser considerados constante, que chamaremos de C_1 e C_2 respectivamente.

Logo

$$u = \int_0^t -\rho C_1 \cos(\omega t) + \rho C_2 \sin(\omega t) dt \quad (2.17)$$

Note a semelhança com o resultado da equação 2.11. Isso mostra que esse método de controle apresenta um funcionamento muito semelhante ao método das perturbações senoidais quando certas condições são obedecidas.

O método da busca extremal por função periódica será usado na próxima seção para ajudar a compreender o funcionamento do controle extremal por modos deslizantes via função de chaveamento periódica.

2.2 Estratégias de Controle Baseadas em Deslizamento

2.2.1 Controle Extremal por Modos Deslizantes Via Função de Chaveamento Periódica

A origem desse algoritmo remonta ao método proposto por Drakunov e Ozguner em [20] e [21] para resolver o problema de plantas cuja a direção do sinal de realimentação fosse desconhecido. Para isso, ele propôs uma função de chaveamento periódica que define múltiplas superfícies de deslizamento onde pelo menos uma é estável independentemente do sinal de realimentação. No entanto, esse método exigia a necessidade do conhecimento do vetor de estados completo da planta. Em [22] foi apresentada uma solução para o problema que envolvia realimentação da saída. Esse solução apresenta convergência global, podendo alcançar o extremo de uma função objetivo independentemente da condição inicial.

Considere um função objetivo $y = f(\theta)$ que queremos maximizar. O esquema de controle por modos deslizantes e realimentação de saída é dado por:

$$\theta = u$$

tal que

$$u = \int_0^t \rho \operatorname{sgn}\left(\sin\left(\frac{\pi}{\epsilon}(y_m - y)\right)\right) dt \quad (2.18)$$

onde $\rho, \epsilon > 0$ são constantes e y_m é uma função de modelo de referência. O diagrama de blocos desse esquema pode ser visto na figura 2.3. Caso queiramos resolver um problema de minimização, basta multiplicar y por -1 .

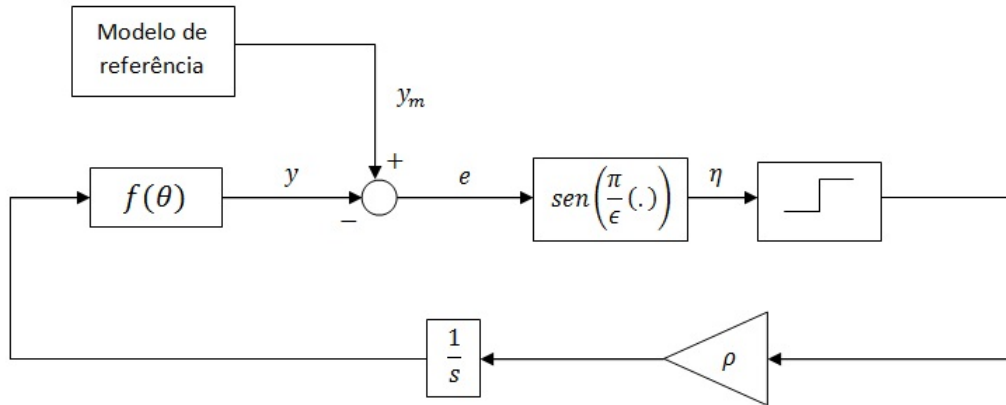


Figura 2.3: Diagrama de blocos do controle extremal por Modos Deslizantes Via Função de Chaveamento Periódica

O sinal de erro $e(t)$ é dado por

$$e(t) = y_m(t) - y(t)$$

onde o modelo de referência $y_m(t)$ adotado para esse trabalho será tal que

$$\dot{y}_m = \omega, \quad y_m(0) = y_{m0}$$

onde $\omega > 0$ e y_{m0} são constantes que serão ajustadas durante o projeto do controlador. Para evitar o problema de um sinal ilimitado dentro do controlador, pode-se simplesmente saturar esse sinal sem que isso modifique o seu desempenho. A função de controle fica

$$u = \int_0^t \rho \operatorname{sgn}\left(\sin\left(\frac{\pi}{\epsilon}(\omega t - y)\right)\right) dt \quad (2.19)$$

Essa estratégia deverá fazer que $y(t)$ rastreie $y_m(t)$. No processo $y(t)$ será forçado

a alcançar a região do se máximo y^* . Esse rastreamento ocorrerá somente enquanto houver deslizamento. Os próximos cálculos estudarão em que condições ele ocorrerá.

Seja $K_p = \frac{\partial y}{\partial \theta}$, então:

$$\dot{y} = K_p \dot{\theta}$$

Derivando o sinal de erro $e(t)$, obtemos

$$\dot{e} = \omega - \dot{y}$$

$$\dot{e} = \omega - K_p \dot{\theta}$$

então

$$v = \dot{x} = \rho \cdot \text{sgn}(\sin(\frac{\pi}{\epsilon}e))$$

Definamos $\sigma = \frac{\pi}{\epsilon}e$ e $\eta = \sin(\sigma)$. Suas derivadas são

$$\dot{\sigma} = \frac{\pi}{\epsilon}\omega - \frac{\pi}{\epsilon}K_p v$$

e

$$\begin{aligned} \dot{\eta} &= \cos(\sigma)\dot{\sigma} \\ &= \cos(\sigma)\left(\frac{\pi}{\epsilon}\omega - \frac{\pi}{\epsilon}K_p v\right) \\ &= \cos(\sigma)\frac{\pi}{\epsilon}\omega - \frac{\pi}{\epsilon}K_p \cos(\sigma)\rho \cdot \text{sgn}(\eta) \end{aligned} \quad (2.20)$$

Para que ocorra deslizamento, é necessário que $\eta\dot{\eta} < 0$. De 2.20 temos

$$\eta\dot{\eta} = \eta \cos(\sigma)\frac{\pi}{\epsilon}\omega - \frac{\pi}{\epsilon}K_p \cos(\sigma)\rho \cdot |\eta| < 0$$

$$\eta \cos(\sigma)\frac{\pi}{\epsilon}\omega < \frac{\pi}{\epsilon}K_p \cos(\sigma)\rho \cdot |\eta|$$

$$|K_p| > \frac{\omega}{\rho} \quad (2.21)$$

Dessa forma, enquanto a condição da equação 2.21 for obedecida, haverá deslizamento e $y(t)$ se aproximará do máximo y^* . Ao chegar perto suficiente do máximo,

$|K_p|$ irá tender a zero, e a condição 2.21 deixará de ser verdadeira. Neste momento, o controle deixará de funcionar por deslizamento, mas a região do máximo já terá sido atingida como o desejado. Fora da região de deslizamento o controle exibirá um funcionamento semelhante ao método da função periódica, tendendo a oscilar em torno da região do máximo.

Observação 3 (Comportamento do controlador sem deslizamento) *Note que quando $y \rightarrow y^*$, a variável $K_p \rightarrow 0$. Quando o controle perde a condição de deslizamento numa região suficientemente pequena em torno de y^* , K_p possui um valor muito pequeno, e conseqüentemente $y(t)$ varia muito pouco, podendo ser considerado praticamente constante pois sua dinâmica torna-se muito lenta em relação a ωt .*

Aplicando a relação trigonométrica $\sin(a + b) = \sin(a) \cos(b) + \sin(b) \cos(a)$ na equação 2.19 obtemos

$$u = \int_0^t \rho \operatorname{sgn}\left(\sin\left(\frac{\pi}{\epsilon}(\omega t - y)\right)\right) dt$$

$$u = \int_0^t \rho \operatorname{sgn}\left[-\sin\left(\frac{\pi}{\epsilon}y\right) \cos\left(\frac{\pi}{\epsilon}\omega t\right) + \cos\left(\frac{\pi}{\epsilon}y\right) \sin\left(\frac{\pi}{\epsilon}\omega t\right)\right] dt$$

Como y pode ser considerado constante próximo de y^ , podemos reescrever a última equação como*

$$u = \int_0^t \rho \operatorname{sgn}\left[-C_1 \cos\left(\frac{\pi}{\epsilon}\omega t\right) + C_2 \sin\left(\frac{\pi}{\epsilon}\omega t\right)\right] dt \quad (2.22)$$

onde C_1 e C_2 são constantes. É possível expandir esse resultado em série de Fourier. Se considerarmos apenas o primeiro harmônico e compararmos com a com as equações 2.11 e 2.17 podemos perceber que o método de controle extremal por modos deslizantes via função de chaveamento periódica exibe um comportamento muito semelhante aos outros dois controladores apresentados anteriormente quando y se encontra próximo ao seu valor extremo.

O exemplo adiante detalhará melhor o comportamento do controlador. Teoremas e provas relativas a esse método de controle podem ser encontradas em [22].

Suponhamos uma função objetivo $f(\theta)$ como descrita a seguir

$$f(\theta) = \left| \cos\left(\frac{\pi}{180}(\theta - \theta_{max})\right) \right|$$

De posse da condição 2.21, podemos escolher os parâmetros ρ e ω de forma a garantir que ocorrerá deslizamento enquanto θ se mantiver fora de uma região Δ em torno do ponto ótimo θ^* da função objetivo $f(\theta)$. Dentro de Δ o controle não funcionará por deslizamento. Se consideramos ω constante, então podemos obter o valor do parâmetro ρ em função do tamanho da região Δ que queremos em torno de θ^* . Para $f(\theta)$ suposta acima, temos

$$|K_p| = \frac{\partial f(\cdot)}{\partial \theta} = \frac{-\pi}{180} \sin\left(\frac{\pi}{180} \Delta\theta\right) \quad (2.23)$$

onde $\Delta\theta = \frac{\Delta}{2}$. Da equação 2.21

$$\rho > \frac{\omega}{|K_p|}$$

Se desejarmos uma região Δ de 30 graus, fazemos $\Delta\theta = 15$ e calculamos $K_p = 0.0045$ na fronteira dessa região. Dentro da região Δ o valor de K_p será necessariamente menor que do lado de dentro. Sendo assim, fixando $\omega = \frac{2\pi}{8}$ e fazendo

$$\rho = \frac{\omega}{|K_p|} = \frac{\frac{2\pi}{8}}{|0.0045|} = 173.87$$

garantimos a condição de deslizamento para a região fora de Δ . Para ilustrar melhor o comportamento do controle foi realizada uma simulação com os parâmetros calculados. O gráfico de θ X η pode ser visto na figura 2.4

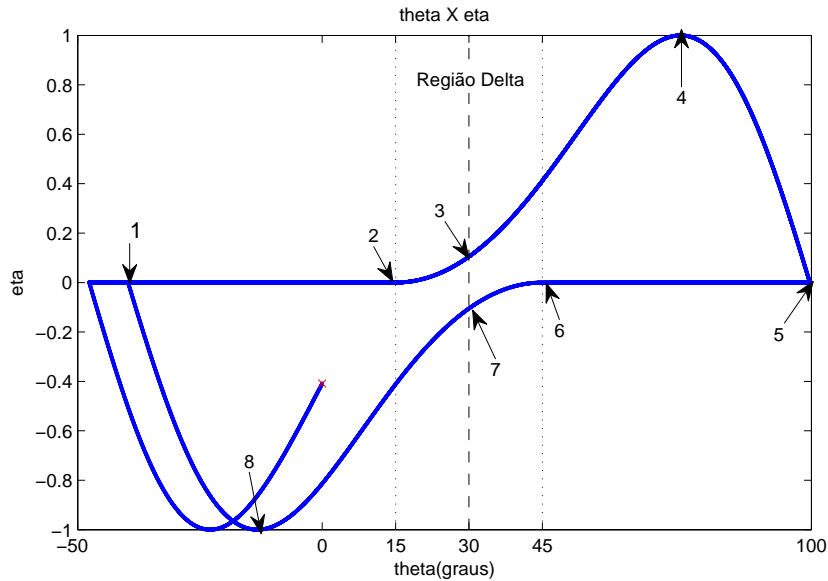


Figura 2.4: Análise das variáveis de controle.

A região Δ vai de 15 graus até 45 graus conforme foi calculado. De 1 a 2 ocorre deslizamento até que θ entra na região Δ . Durante o deslizamento a variável σ estará fixa na superfície de deslizamento $\sigma = K\pi$ onde $K = 0, 1, 2, 3, \dots$. A partir de 2 a condição da equação 2.21 para de ser verdadeira e o controle perde o deslizamento. Nesse momento, o controle exibe um comportamento semelhante ao do método da função periódica. Ao chegar em 3, θ alcança θ^* e y seu valor máximo. Entre 3 e 4, θ se afasta do máximo da função e controle apresenta realimentação positiva. Ao passar por 4 o ganho do controlador inverte de sinal e a realimentação volta a ser negativa. De 5 a 6 volta ocorrer deslizamento e a variável σ se fixa no valor $(K + 1)\pi$. Entre 6 e 7 não existe mais deslizamento, mas a realimentação ainda é negativa. Em 7, novamente alcança-se o máximo valor de y e a partir daí começa-se a se afastar do máximo com realimentação positiva. No ponto 8, a realimentação volta a ser negativa e no ponto 1 o ciclo se repete outra vez, fazendo com que θ permaneça oscilando em torno de θ^* .

Para estudar os efeitos da variação dos parâmetros do controlador foram feitas diversas simulações variando os valores de ϵ , que está ligado a frequência do seno da variável η , e de ρ .

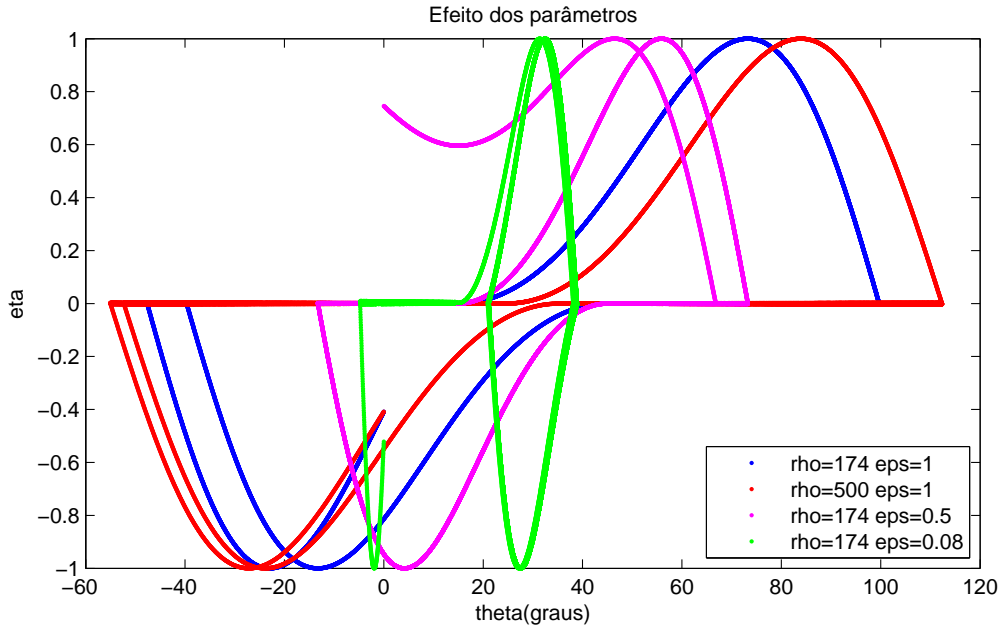


Figura 2.5: Efeito da variação dos parâmetros.

Observe o gráfico da figura 2.5. Comparando os resultados com a curva azul, podemos observar que um aumento em ρ diminui o tamanho da região Δ (curva vermelha). Isso é esperado de acordo com os cálculos feitos anteriormente. Além disso, o aumento de ρ também aumentou a amplitude das oscilações do ângulo θ . Intuitivamente, um aumento em ρ pode ser interpretado como um aumento na velocidade com que nos deslocamos no eixo θ , enquanto que a frequência do seno pode ser entendida como o tempo de resposta que o controlador percebe mudanças no sinal de K_p . Quanto maior a frequência, menor o tempo de resposta. Como a frequência não aumentou junto com a variável ρ , é normal que a saída do controlador percorra distâncias maiores em θ antes que seja percebida a mudança de sinal de K_p .

Nas curvas rosa e verde, o valor de ϵ foi reduzido mantendo os outros parâmetros constantes. Percebe-se que o parâmetro ϵ altera o tamanho das oscilações de θ sem alterar o tamanho da região Δ . Quanto menor ϵ mais próximo de θ^* ocorrerá o momento em que o controlador percebe a mudança de sinal de K_p . Uma redução de ϵ significa um aumento na frequência do seno, melhorando o tempo de reação do controle. Quanto menor o parâmetro, menor será o trecho de deslizamento,

sendo possível até mesmo que θ nunca saia da região Δ , fazendo com que o controle funcione sem o deslizamento.

Apesar de diminuir as oscilações em torno do ponto ótimo, uma redução de ϵ acarreta aumento da frequência do seno da variável η . Sendo assim, o valor de ϵ mínimo estará restrito pelas limitações físicas do experimento prático.

Capítulo 3

Implementação Prática: Descrição do Sistema

3.1 Visão Geral do Experimento

O objetivo da montagem é criar uma bancada física onde se possa aplicar a teoria de controle por busca extremal fora do ambiente de simulação numérica. Dessa forma, poderemos coletar dados de desempenho do controlador na presença de ruído, dinâmicas desconsideradas durante seu projeto, taxa de amostragem reduzida, não-linearidades desprezadas e uma série de outros comportamentos não compensados no projeto. Só então, será possível confrontar os dados reais obtidos com os resultados previstos pela teoria, estimar a robustez dos diferentes métodos e observar suas limitações.

Na figura 3.1 é possível ver a foto da versão final da bancada que foi montada no LabECA.

No experimento, desejamos maximizar a área visível de um alvo vermelho observada por uma câmera. Esse alvo está preso ao eixo de um motor DC de forma que a direção da normal do plano do alvo seja perpendicular ao eixo girante. Assumindo que a câmera é posicionada de forma de forma seu eixo óptico também seja perpendicular ao eixo do motor, o controle, ao maximizar a área visível do alvo, deverá

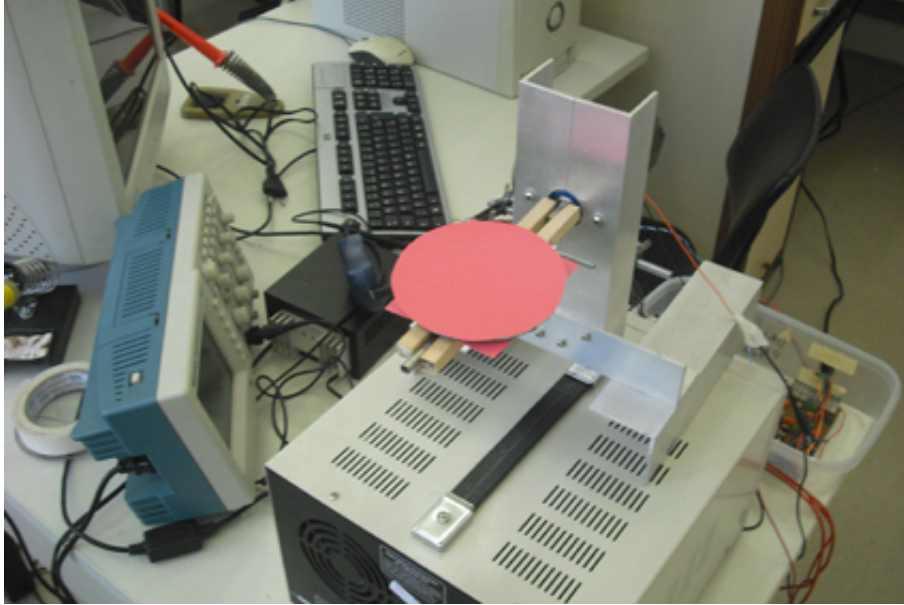


Figura 3.1: Outra foto da bancada

fazer o motor girar de forma que o vetor normal do alvo e eixo óptico da câmera se alinhem. Devido às características do controle ESC, é esperado que a posição angular do motor oscile em torno desse ponto ótimo.

A função objetivo será a área do alvo projetada no plano da imagem da câmera

$$area = y = f(\theta)$$

onde θ é o ângulo do eixo do motor. Espera-se que o controle leve esse ângulo a uma região em torno de θ^* que maximiza a saída da função objetivo.

Visando agilizar a implementação da parte prática e aproveitar o trabalho e conhecimento agregado por antigos alunos de graduação, grande parte da estrutura física foi aproveitada de projetos finais de alunos antigos ou trabalhos de iniciação científica. Muitas das decisões de projeto acabaram sendo influenciadas pelo material que já estava disponível em laboratório. Dessa forma, foi possível forçar mais tempo no desenvolvimento do software e controle.

O software é responsável por várias tarefas, entre elas o algoritmo de visão e cálculo de controle. Devido à sua complexidade e importância ele será apresentado detalhadamente em outra seção. Já o microcontrolador é responsável por converter

o sinal de controle vindo do computador em sinal PWM e também por enviar os valores lidos no encoder do motor para o computador. Seu protocolo de comunicação será discutido quando estivermos falando da comunicação do software.

O sistema funciona da seguinte maneira. Uma webcam captura uma imagem e a envia para um computador via USB(*Universal Serial Bus*). O computador executa um software escrito em Matlab e se comunica com uma placa eletrônica contendo um microcontrolador Atmel AT90CAN por meio de comunicação serial padrão RS-485. Conectado ao computador está um cabo conversor USB/RS-232 que, por sua vez, se conecta a um conversor RS-232/RS-485. O microcontrolador aciona um circuito de potência por meio de um sinal PWM(*Pulse Width Modulation*) que então amplifica o sinal de nível TTL(0-5V) para um sinal PWM de potência necessário para atuar no motor(0-12V). Por fim, um encoder acoplado ao motor envia o sinal de posição para a placa eletrônica que contém um decodificador apropriado (HCTL2020) como interface com o microcontrolador. Um desenho esquematizado pode ser visto na figura 3.2

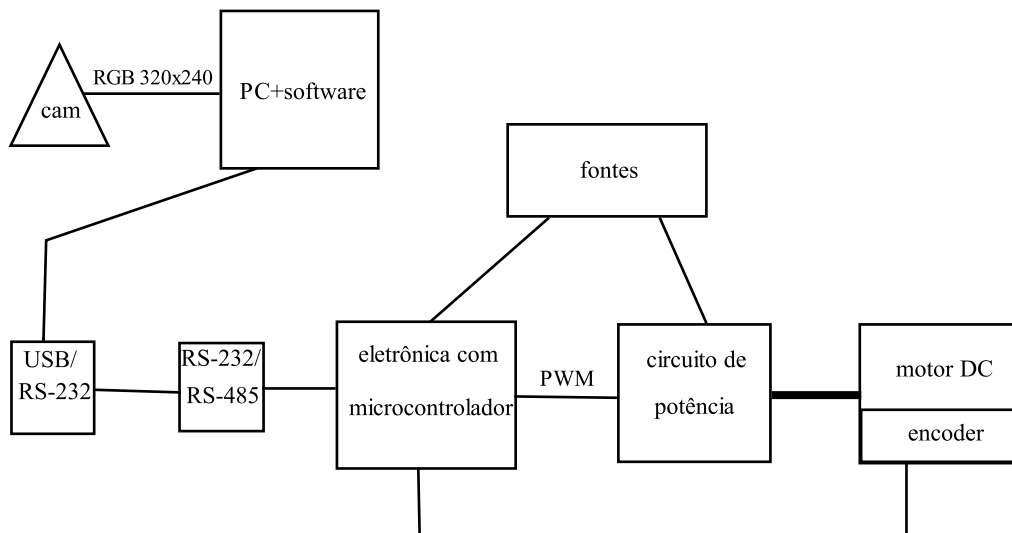


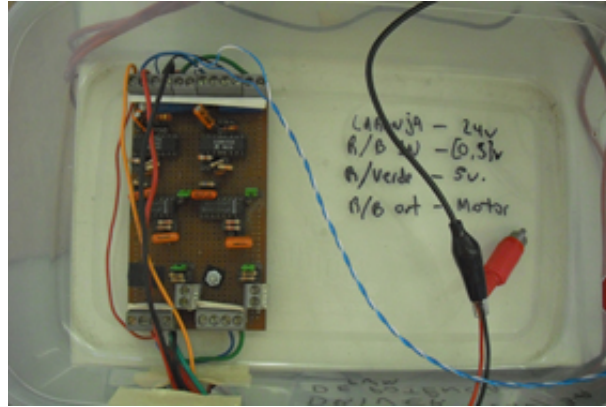
Figura 3.2: Esquema da bancada

A figura 3.3 mostra uma foto da eletrônica do microcontrolador e uma foto do circuito de potência.

Além disso, bancada utiliza duas fontes de potência. A placa eletrônica com o microcontrolador é alimentada por uma fonte estabilizada a 12V. A segunda fonte



(a)



(b)

Figura 3.3: (a)Eletrônica do microcontrolador (b)circuito de potência

alimenta o circuito de potência que aciona o motor. O microcontrolador satura o sinal do PWM impedindo que o motor funcione em potência máxima, ajudando a proteger a parte eletrônica.

3.2 Funcionamento do Software

O software foi todo escrito em Matlab. Tal linguagem ofereceu grande facilidade e rapidez de programação. Sua desvantagem foi velocidade de execução mais lenta quando comparada a linguagem C, por exemplo. Isso acabou reduzindo a taxa de amostragem do experimento, problema que será discutido detalhadamente mais adiante. O software possui três partes: inicialização, loop principal, finalização. A primeira parte, inicialização, tem a função definir constantes, inicializar variáveis que precisem de um valor inicial e criar os vetores onde serão salvos os dados experimentais. Além disso, essa parte inicializa e configura a câmera, abre a comunicação com a porta serial e chama a interface gráfica.

Em seguida, começa a execução de do loop principal. Essa é a parte mais importante do programa. Primeiramente, a câmera recebe uma imagem e executa

algoritmo de visão para obter o valor da área visível do alvo. Então, o software lê o valor do encoder vindo do microcontrolador e obtém o ângulo em graus do motor. De posse da área e do ângulo, o programa executa o algoritmo de controle e envia o sinal de controle para o microcontrolador. Ao final, atualiza-se as variáveis, armazena-se os dados desejados em vetores que serão salvos mais tarde e imprime-se na tela os valores relevantes. Esses passos são, então, repetidos até que o programa receba um comando *STOP* da interface gráfica.

Por fim, na fase de finalização, o programa manda o sinal para o motor parar, fecha o canal de comunicação, desliga a câmera e exibe a taxa de amostragem média naquela execução. O programa também pergunta ao usuário se deseja salvar os dados experimentais.

Os fluxogramas das figuras 3.4 e 3.5 resumem essas etapas.

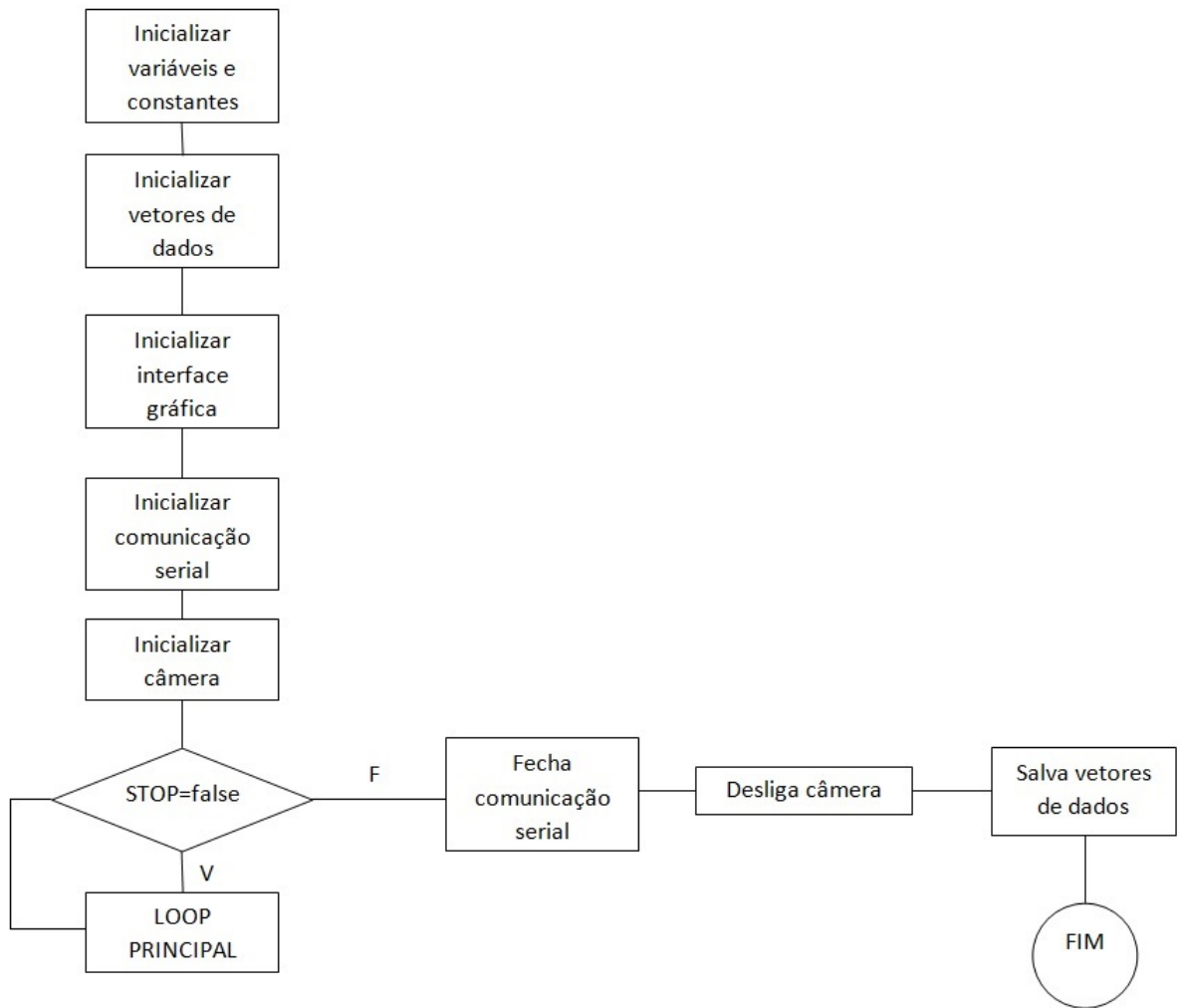


Figura 3.4: Esquema da bancada

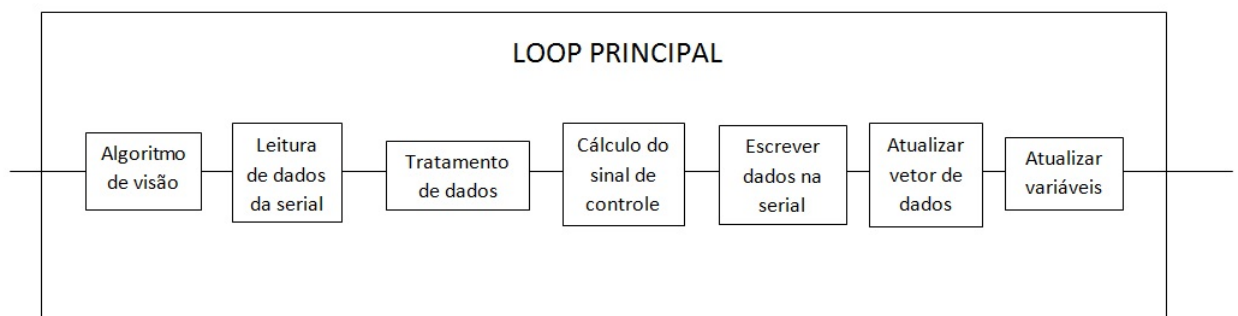


Figura 3.5: Esquema da bancada

As partes mais importantes do software serão discutidas de forma mais aprofundada nas subseções a seguir.

3.2.1 Algoritmo de Visão

O objetivo do algoritmo de visão é obter um valor numérico que represente área do alvo que pode ser vista pela imagem gerada pela câmera posicionada acima do motor. O valor da área atingirá valor máximo quando o alvo estiver perpendicular a direção apontada pela câmera, e atingirá valor mínimo quando o plano do alvo for paralelo a direção apontada pela câmera. Caso a direção apontada pela câmera passe perfeitamente pelo eixo de rotação do alvo, o valor mínimo da área será 0. A área medida pelo algoritmo será o valor que buscaremos otimizar usando o controle ESC. A entrada da função objetivo será o ângulo θ medido pelo o encoder no motor e a saída será o resultado do algoritmo de imagem.

Para permitir que diferenciemos o alvo de todo o resto da imagem, ele possui cor vermelha, enquanto evitamos que qualquer outro objeto visível pela câmera tenha essa cor. Será necessário fazer um esquema de reconhecimento de cores durante o algoritmo para que possamos separar o alvo do resto da imagem. Uma descrição detalhada de espaço de cores e de esquemas de detecção de cores foge ao escopo desse trabalho. Maiores informações sobre o assunto podem ser encontradas em [23].

O algoritmo de visão é executado logo no início do loop principal do programa. Inicialmente, o software recebe um quadro da câmera contendo 3 matrizes no formato red-green-blue($R G B$). A cada matriz contém a intensidade de sua respectiva cor para cada pixel variando de 0(sem presença da determinada cor) a 255(intensidade máxima daquela cor). Essas matrizes tem um tamanho 320x240 que foi a resolução escolhida como formato de saída câmera. Uma resolução maior significaria matrizes maiores, o que aumentaria o tempo de processamento do algoritmo, ou seja, menor taxa de amostragem, e ocuparia mais memória. Por outro lado, uma resolução menor significaria menor precisão na medida da área do alvo. A resolução de 320x240 foi escolhida após diversos testes de forma a equilibrar tempo de processamento e precisão.

Antes de medir a área do alvo, deseja-se retirar da imagem tudo aquilo que não seja predominantemente vermelho. Assumindo a hipótese que não outros objetos

vermelhos na área de visão da câmera, tudo que for vermelho, deverá fazer parte do objeto, cuja área desejamos medir. Para isso, gera-se duas matrizes. A primeira contém a apenas a componente R, vermelha, da imagem da câmera. A segunda matriz é obtida transformando a imagem original numa escala de cinza. A seguir, realiza-se uma operação de subtração da segunda com primeira, obtendo como resultado uma matriz onde cada pixel apresenta um valor de 0 a 255. Quanto maior o valor, mais vermelho é aquele pixel. A figura 3.6 ilustra as etapas do processamento.

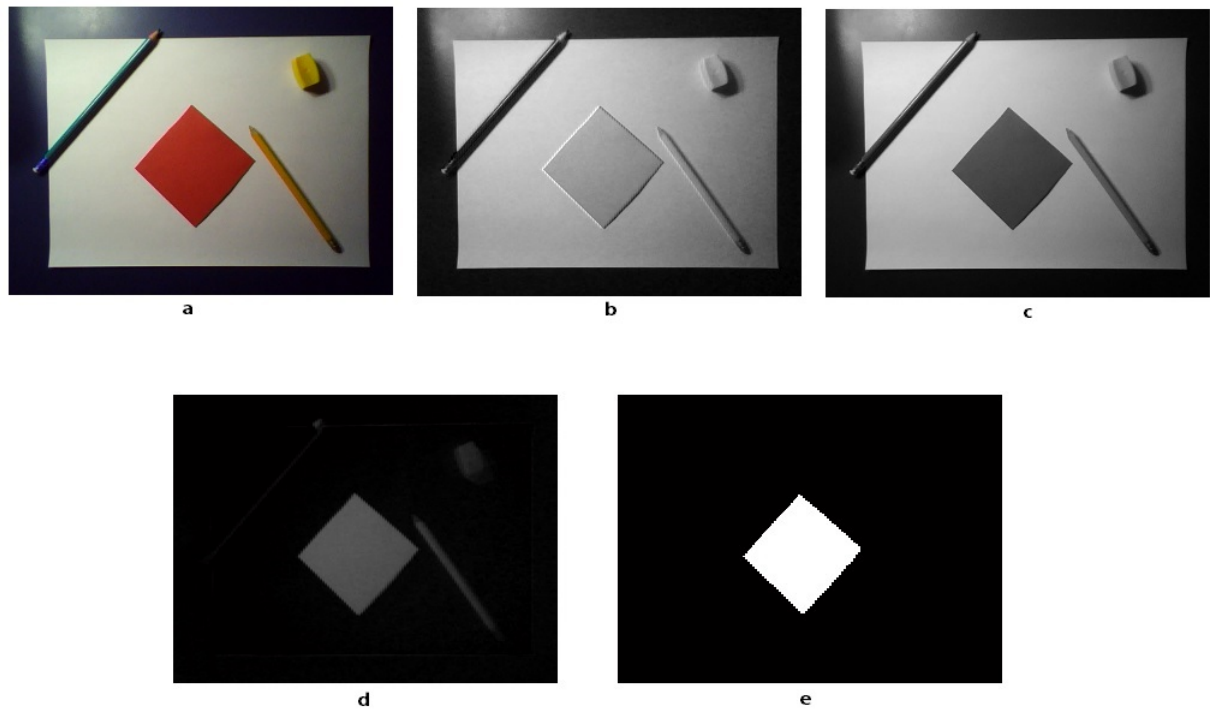


Figura 3.6: Etapas do processamento de imagem. (a)imagem original (b)componente R (c)escala de cinza (d)resultado subtração (e)imagem binária

Após essa etapa, experimentou-se fazer a filtragem da imagem obtida para melhorar o resultado final. Para isso, foi usado o comando *medfilt2* do Matlab. No entanto, não foi notada vantagem significativa que justificasse o aumento no tempo de processamento, e a ideia do filtro foi abandonada.

A etapa seguinte é responsável pela detecção dos pixels vermelhos. Converte-se a imagem já processada em uma imagem binária, ou seja, composta apenas de 1 e de 0. Para isso, usou-se o comando *im2bw* do MATLAB. Basicamente, define-se um valor de *threshold*. Na matriz de entrada, qualquer pixel com valor acima desse valor de *threshold* será atribuído o valor 1(branco) na matriz de saída, enquanto

que cada pixel abaixo do valor de *threshold* receberá o valor 0(preto). Dessa forma, o resultado será uma imagem preto-e-branco, onde os pontos brancos representam vermelho, ou seja, aquilo que faz parte do alvo, e os pontos pretos serão outras cores que não interessam para análise. Esse valor de *threshold* varia dependendo do ambiente e do tom de vermelho do alvo. Ele é obtido experimentalmente e pode ser mudado no script se necessário. Para calcular a área, basta contar o número de pixels brancos, de valor 1 na matriz imagem. Isso pode ser feito usando a função *nnz* do MATLAB, ou simplesmente multiplicando a matriz binária à esquerda por um vetor linha composto de 1 e, depois, multiplicando à direita por um vetor coluna de composto de 1. Esse valor é armazenado na variável *objArea*. O valor da área contado pelo algoritmo varia de acordo com a distância da câmera com o alvo. Para facilitar a escolha dos parâmetros do controlador, normalizamos o valor da área para valores entre 0 e 1. Seja *areaMax* o valor máximo de pixels vermelhos contados para uma determinada montagem da bancada. O valor da área normalizada *y* calculada será:

$$y = \frac{objArea}{areaMax}$$

O valor de *y* será a saída do algoritmo de visão do programa.

Durante os testes, percebeu-se que o algoritmo de imagem é um dos grandes limitadores da taxa de amostragem do experimento. A execução do algoritmo de imagem é responsável por cerca de 40% (aproximadamente 0,04s) do tempo de execução do loop principal. Junto com a comunicação serial (mais especificamente, a leitura serial, como será visto mais a seguir), eles serão responsáveis pela limitação da taxa de amostragem do experimento.

3.2.2 Comunicação e Tratamento de Dados

A comunicação entre o MATLAB e o microcontrolador do experimento é feita por fio usando comunicação serial. A comunicação serial se mostrou uma opção mais barata, de rápida e fácil implementação. O MATLAB já possui inúmeras funções para realizar e tratar a comunicação serial também. Ademais, a presença de fio não

oferece problemas na implementação física da bancada.

A taxa de transmissão (*BaudRate*) usada foi 9600 bits por segundo. Para realizar o pedido de leitura do encoder do motor, é necessário enviar 1 byte contendo o caractere ASCII "d". Após isso, o microcontrolador envia dois bytes contendo o valor do encoder que deverá ser lido pelo MATLAB. Os valores recebidos são ,então, tratados de forma converter a leitura em graus. Além disso, compara-se o valor do ângulo atual com o valor de ângulo do passo anterior para se detectar overflow. Caso seja detectado, soma-se ou subtrai-se uma volta completa conforme o necessário.

Os valores de saída do controlador, também devem ser tratados antes de serem enviados para o microcontrolador. O valor de controle é saturado entre 1 e -1 e, a seguir, convertido em um número inteiro entre 0 e 255, para serem enviados ao motor. Para enviar os dados, é necessário primeiramente enviar um caractere "e" para o microcontrolador, sinalizando que o próximo valor enviado será o valor de entrada do motor. Depois disso, envia-se um byte contendo esse valor. Resumindo, temos o seguinte protocolo de comunicação:

- Envio de um caractere "d" para pedido de leitura (1 byte enviado)
- Leitura de dois bytes do encoder (2 bytes recebidos)
- Envio de um caractere "e" seguido de um byte contendo o sinal do motor (2 bytes enviados)

Já o programa do microcontrolador apresenta um loop infinito de recepção (pooling). O primeiro caractere recebido posiciona o programa numa estrutura switch/case na qual um dos tipos é o do protocolo implementado. Caso ele receba "d", o microcontrolador envia os dados do encoder. Caso ele receba "e", ele escreverá nos motores, sendo o valor 0 correspondente ao giro com velocidade máxima num sentido e 255 velocidade máxima no sentido contrário.

Ao realizarmos alguns testes com o software escrito em MATLAB, percebemos que o a os pedidos de leitura somados com os pedidos de escrita levavam cerca de 0.06 segundos por execução do loop principal do programa, correspondendo a cerca

de 50% do tempo total. Quase todo esse tempo de execução estava concentrado na leitura de dados vindos do microcontrolador. Isso acontece porque logo depois de se fazer o pedido de leitura ao microcontrolador, existia um certo tempo de espera até que os dados chegassem ao computador e pudessem ser lidos. Numa tentativa de tentar diminuir o tempo de comunicação, alterou-se um pouco a ordem do programa. O pedido de leitura passou a ser feito antes do algoritmo de visão e a leitura de dados continuou sendo feita depois do algoritmo de visão. Dessa forma, após o pedido de leitura, o computador continua trabalhando, dando tempo para que a resposta do microcontrolador chegue ao buffer da máquina, eliminando o tempo ocioso da máquina esperando pela chegada dos dados.

Com as alterações feitas o tempo total de comunicação caiu para 0.04 segundos. Com isso, o tempo total de execução do loop principal reduziu também. Fazendo alguns testes, conseguimos uma média de taxa de amostragem de:

$$\textit{Taxa de Amostragem} = 10.6865\textit{hz}$$

Como havia sido dito anteriormente, a comunicação e o algoritmo de visão foram os grandes limitadores dessa taxa de amostragem. Listando a contribuição de cada parte do programa, temos:

- Algoritmo de visão: 40% do tempo total
- Comunicação serial: 40% do tempo total
- Algoritmo de controle, tratamento de dados, armazenamento de dados e outros: 20% do tempo total

3.2.3 Interface Gráfica

A interface gráfica é simples e seu objetivo é permitir que o usuário altere o parâmetros do controle no decorrer da execução do programa ou encerrar a execução do loop principal para terminar o programa. A primeira versão da gui apenas possuía um

botão de *STOP*, para encerrar o loop principal, e uma caixa de texto para alterar o sinal da referência. Na segunda versão foi adicionada algumas outras caixas de texto, permitindo alterar outras variáveis como a frequência angular ω do controle ESC, o ganho K_p da realimentação do motor e o valor de uma variável *dist* que simula um distúrbio na entrada da planta. A figura 3.7 mostra o layout da interface gráfica. Para alterar os valores do programa em execução basta inserir um valor numérico da caixa de texto da variável que se quer alterar e clicar no botão *change*.

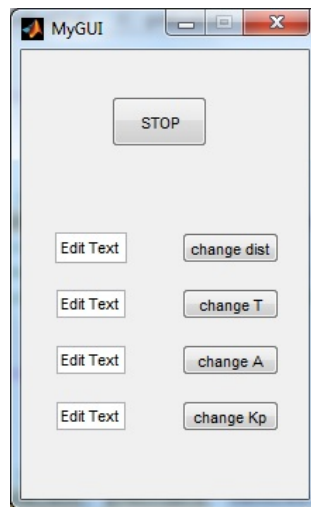


Figura 3.7: Interface com o usuário

Capítulo 4

Identificação, Simulações e Experimentos

4.1 Identificação do Modelo e da Função Objetivo

Com a bancada montada, o primeiro passo tomado foi fechar a malha de controle com um ganho proporcional K_p , de forma que, ao aplicarmos o controle ESC mais tarde, possamos simplesmente fornecer uma referência de ângulo desejado. Idealmente, essa malha de controle dever ser muito mais rápida quando comparada à malha de ESC. Inicialmente, desejou-se encontrar o ganho proporcional que ofereceria um bom tempo de resposta, sem saturar o sinal de controle do motor. Para isso foram aplicados diversos sinais degrau na referência do sistema em malha fechada. Na tentativa inicial, o ganho escolhido foi $K_p = -1$. O ganho é negativo por causa do sentido de rotação do motor. Para esse valor, o motor atuava basicamente em saturação. O valor, então, foi diminuído e constatamos que ganhos de -0.002 ou -0.004 seriam satisfatórios e manteriam o sinal de entrada do motor v em uma região segura longe dos valores de saturação. Em seguida, desejou-se saber qual era o desempenho do sistema ao rastrear referências do tipo $A \sin(\omega t)$ para diversos valores de ω . Devido à reduzida taxa de amostragem do sistema, era importante saber qual seria a frequência máxima ω_{max} no sinal da referência antes que a perda

de amostras se tornasse grande demais e causasse comportamentos indesejados no sistema, como o aparecimento de não-linearidades. Após alguns testes, chegou-se a conclusão que esse valor seria algo em torno de

$$\omega_{max} = \frac{2\pi}{3}$$

Além disso, foi constatado que para valores baixos de A , a presença de atrito seco, gerava alguns comportamentos não-lineares significativos capazes de prejudicar o rastreamento de referências senoidais. O controlador deverá oscilar com amplitudes relativamente altas para impedir o aparecimento desses comportamentos.

A figura 4.1 demonstra os resultados experimentais obtidos para o rastreamento de um sinal senoidal e de um sinal degrau.

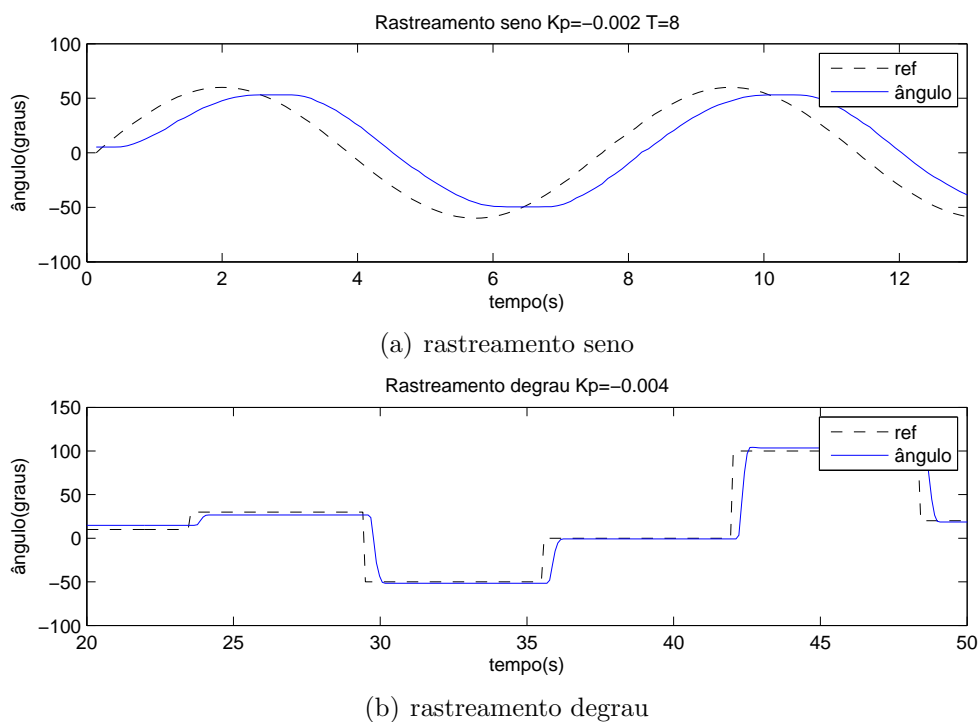


Figura 4.1: Resultado do rastreamento

Para realizar as simulações no ambiente Simulink foi necessário estimar um modelo linear da planta em malha fechada. Desprezando atrito seco, e inércia da carga do motor, admitiu-se que a função de transferência entre o sinal de entrada do motor

e a velocidade angular de saída seria da forma

$$P(s) = \frac{a}{s(s+b)}$$

É razoável desprezar a inércia do alvo se levarmos em conta a redução motor e a massa desprezível do alvo. Ao multiplicarmos a planta pelo ganho proporcional do controle temos uma função de transferência em malha aberta

$$MA = \frac{K_p a}{s(s+b)}$$

Fechando a malha obtemos

$$\begin{aligned} MF &= \frac{\frac{K_p a}{s(s+b)}}{1 + \frac{K_p a}{s(s+b)}} \\ &= \frac{K_p a}{s^2 + bs + K_p a} \end{aligned} \tag{4.1}$$

De posse do modelo linear em malha fechada, foram realizados alguns testes experimentais usando ganho proporcional $K_p = -0.002$ para obtenção de dados. Usando a toolbox de identificação do Matlab para estimar os valores dos parâmetros da equação 4.1, os resultados obtidos foram:

$$a = -10005$$

$$b = 11.42$$

Para complementar o estudo, possibilitando entender o efeito da introdução de dinâmicas não modeladas no sistema, também foi gerado um modelo de grau relativo 1, admitindo o modelo de motor

$$MA = \frac{K_p a}{s}$$

Em malha fechada

$$\begin{aligned}
 MF &= \frac{\frac{K_p a}{s}}{1 + \frac{K_p a}{s}} \\
 &= \frac{K_p a}{s + K_p a}
 \end{aligned}
 \tag{4.2}$$

Conhecendo o valor de $K_p = -0.002$, calculou-se

$$a = -779$$

Esses resultados foram modelados simulink e transformados em único bloco *subsystem* conforme pode ser visto na figura 4.2.

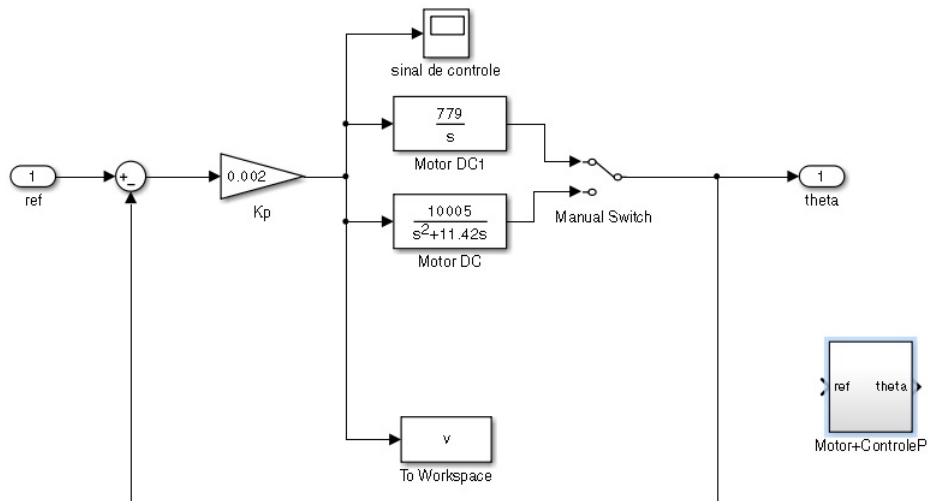


Figura 4.2: Simulink do Motor + Controle P

O gráfico da figura 4.3 compara os resultados do modelo de 2a ordem com o experimento físico. O erro entre os dois sinais não chega a passar de 3 graus.

Além disso, para realizar as simulações numéricas, foi necessário modelar a função custo $f(\theta)$ com base nos dados experimentais obtidos. Foram criados dois modelos de $f(\theta)$, um modelo polinomial de segunda ordem, $a_2x^2 + a_1x + a_0$, e outro modelo baseado em geometria do tipo $|A \cos(\frac{\pi}{180}x)|$. Usando MATLAB para obter os

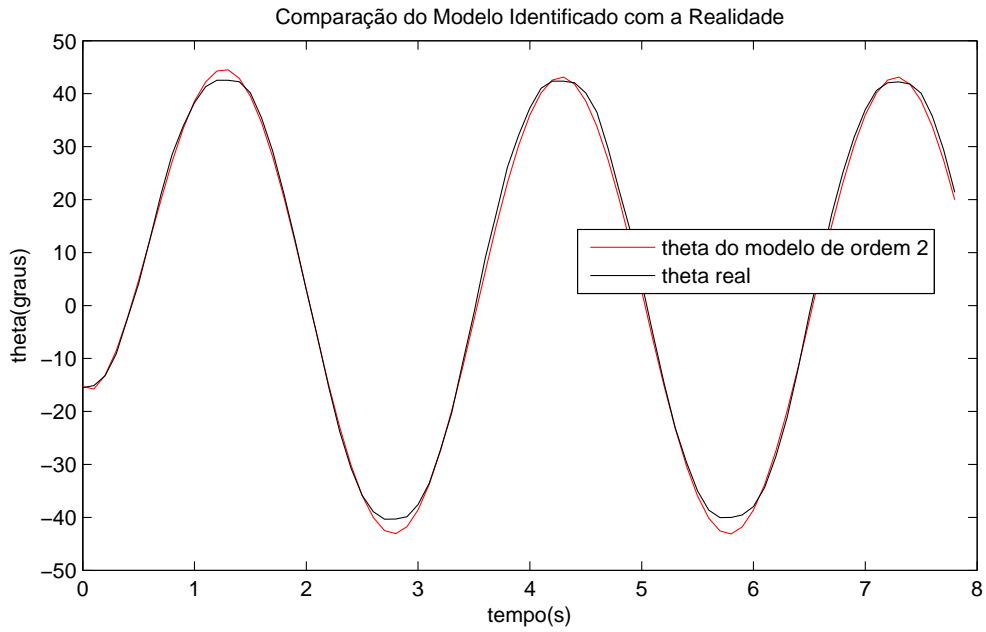


Figura 4.3: Comparação do Modelo Identificado com a Realidade

parâmetros das curvas, obtemos os seguintes resultados abaixo.

$$f(\theta) = 0.0001 \theta^2 + 0.0015 \theta + 0.7805$$

ou

$$f(\theta) = |0.7963 \cos(\frac{\pi}{180}(\theta - 8.611))|$$

A representação visual pode ser vista na figura 4.4.

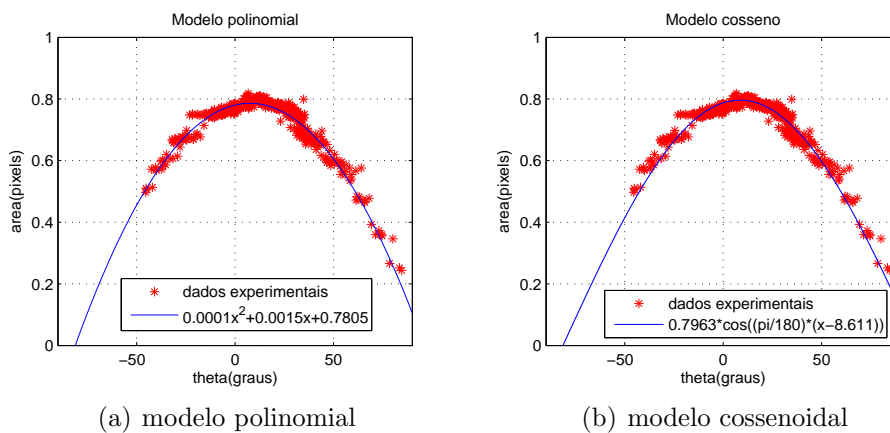


Figura 4.4: Os dois modelos e os pontos obtidos experimentalmente

O modelo polinomial facilita a análise matemática do problema, enquanto que o

modelo cossenoidal se mantém mais fiel a realidade fora da região do máximo. Os dois modelos serão usados nas simulações conforme o necessário. Para facilitar a visualização dos resultados durante as simulações, os dois modelos foram centralizados em torno de θ_{max} e o valor máximo foi deslocado para 1, obtendo as seguintes funções

$$f(\theta) = -0.0001(\theta - \theta_{max})^2 + 1 \quad (4.3)$$

ou

$$f(\theta) = \left| \cos\left(\frac{\pi}{180}(\theta - \theta_{max})\right) \right| \quad (4.4)$$

A comparação dos dois modelos pode ser visto na figura 4.5. No caso $\theta_{max} = 0$.

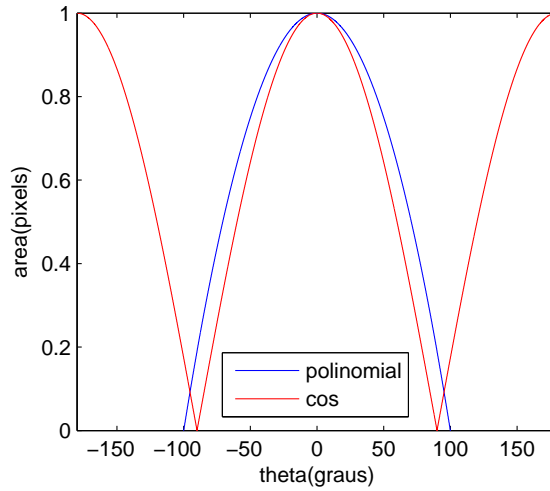


Figura 4.5: Comparação do modelo cossenoidal e polinomial.

4.2 Simulações - Método das Perturbações Senoidais

Nessa seção serão realizadas simulações numéricas relativas ao desempenho do método das perturbações senoidais. Durante as simulações será usado o modelo cossenoidal da função objetivo. O modelo em Simulink pode ser visto na figura 4.6. Os blocos em amarelo são os blocos do ESC.

Variando os *switchs* do modelo em Simulink podemos inserir as dinâmicas da

planta no sistema. Durante as simulações estudaremos os efeitos da inserção de dinâmicas que não foram levadas em conta no projeto do controlador, que será projetado sob a hipótese que a dinâmica do motor realimentado é desprezível.

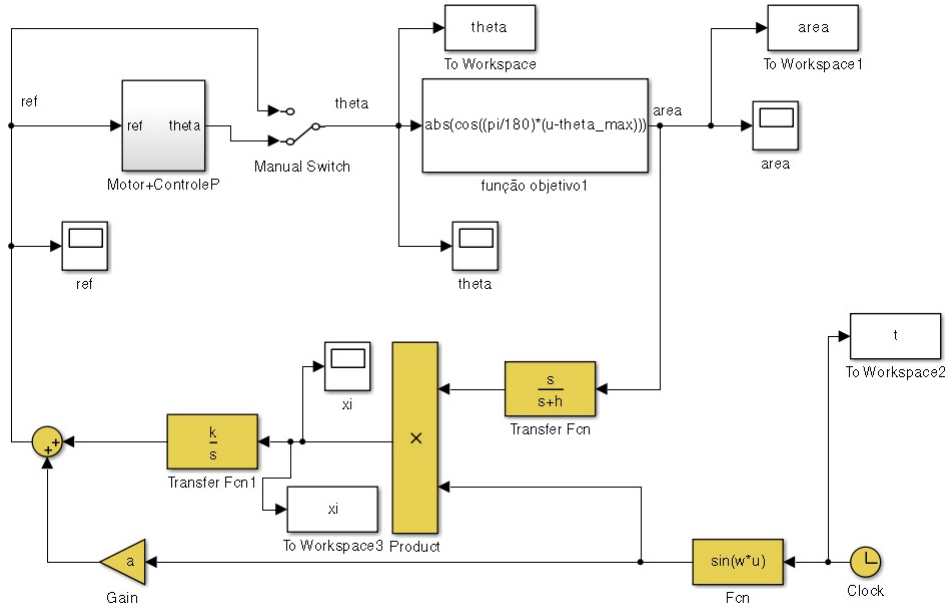


Figura 4.6: Simulink do Método das perturbações senoidais

4.2.1 Projeto

A partir da equação 2.9, podemos escolher os parâmetros do controlador de forma a conseguir o desempenho desejado no domínio do tempo. Resolvendo a equação diferencial obtemos

$$\tilde{\theta}(t) = e^{-k \frac{af''}{2} t} \quad (4.5)$$

Logo a constante de tempo é

$$\tau = \frac{2}{ka f''} \quad (4.6)$$

Sabendo que o fim do transitório acontece em aproximadamente 5 constantes de tempo τ , pode-se escolher o valor dos parâmetros k e a em função de quanto tempo desejamos que leve o transitório e do valor de f'' . Usando o modelo polinomial da equação 4.3, calcula-se que $f'' = -0.0002$. Esse valor também é válido para a equação do modelo cossenoidal 4.4 para valores de θ próximos de θ^* . Fazendo o

tempo de transitório igual a $T_{transitorio} = 60$, calculamos

$$ka = 5 \frac{2}{T_{transitorio} f''} = 5 \frac{2}{60 \cdot 0.0002}$$

$$ka = 833.3333$$

Como foi visto anteriormente, o sistema com motor apresenta um comportamento não-linear devido à presença de atrito seco. Fazendo a amplitude das oscilações relativamente grande, diminuimos o efeito dessa não-linearidades, apesar de aumentar o tamanho das oscilações no regime permanente. Dessa forma, faremos $a=15$. Então

$$k = 55.5556$$

A escolha do valor de ω apresenta alguns problemas práticos. Idealmente, deveria ser escolhido um valor alto de ω , o que melhoraria o valor da estimativa do gradiente e eliminaria os efeitos dos harmônicos de ordem superior. No entanto, devido a pequena taxa de amostragem do sistema implementado, cerca de 10 hz, foi constatado experimentalmente que para ondas senoidais de frequência angular maior que $\frac{2\pi}{3}$ hz começam aparecer efeitos não-lineares significativos que devem ser tratados com teoria de controle discreto, o que foge do escopo desse trabalho. Para facilitar a implementação prática foi escolhido

$$\omega = \frac{2\pi}{4} = 1.5708$$

Por último, a constante h do filtro passa-alta deve ser menor que ω de forma a filtrar a componente DC do sinal de saída. Experimentalmente, chegou-se ao valor

$$h = 0.1$$

O modelo de controle implementado vale para plantas de grau relativo 0. Inicialmente, simularemos o problema sem a presença de dinâmicas, assumindo que

o tempo de transitório entre o sinal da referência da planta e o ângulo θ é desprezível. Em seguida, adicionaremos a planta de grau relativo 2 e observaremos os efeitos da inserções de dinâmicas extras não levadas em consideração no projeto do controlador.

4.2.2 Planta de Grau Relativo 0

Para $\theta^* = 30$ foi obtido o resultado visto na figura 4.7. A variável θ atinge θ^* no tempo esperado e a área oscila próximo ao valor máximo da função. As oscilações são grandes para evitar o aparecimento de não linearidades. Já na figura 4.8, podemos ver o sinal da variável ξ , que representa a estimação do gradiente. Quando a área atinge a região do máximo, observe que o gradiente oscila em torno de 0 como esperado.

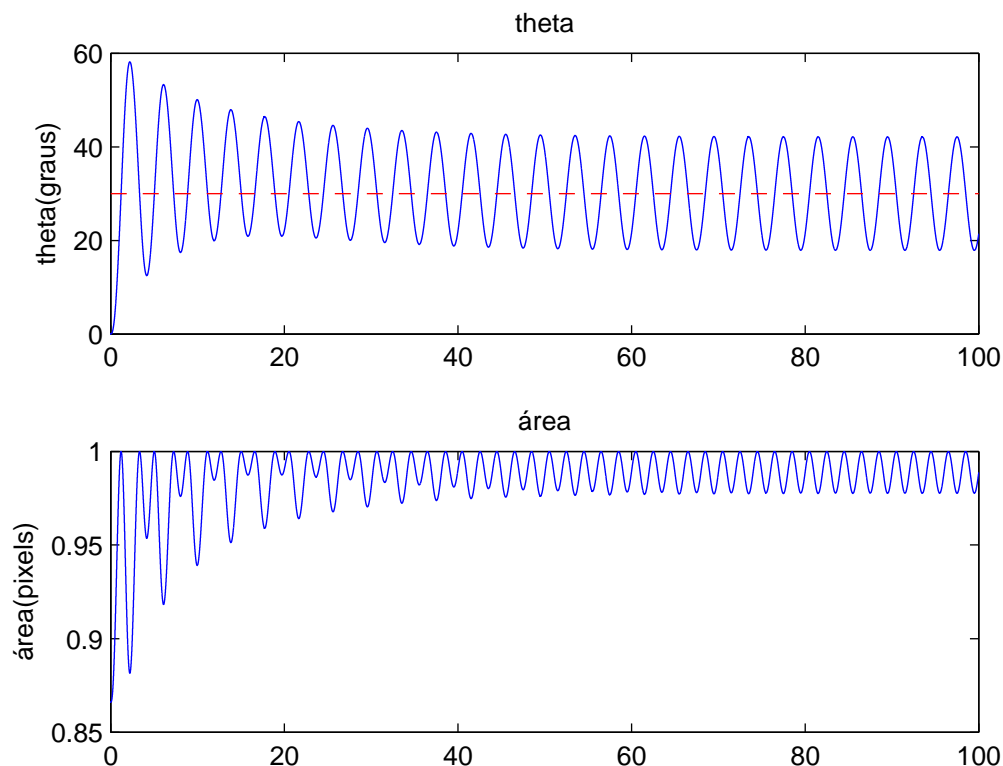


Figura 4.7: Simulação com grau 0

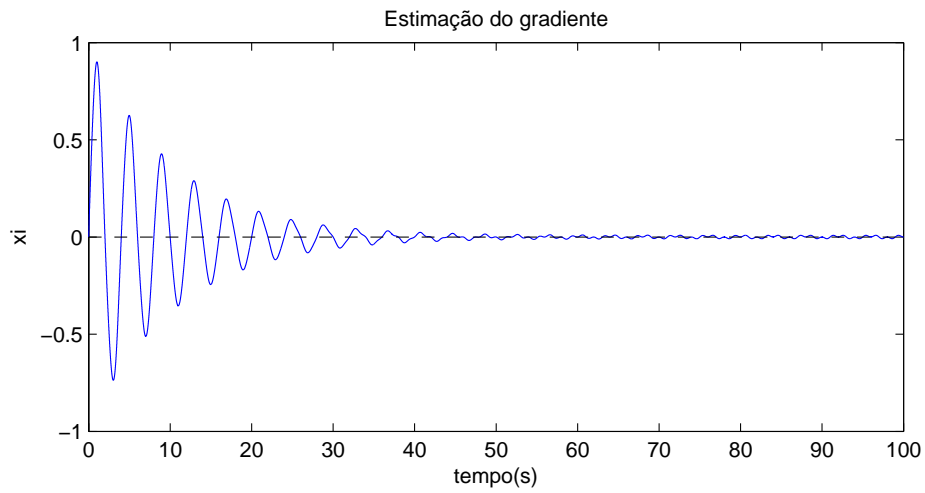


Figura 4.8: Estimação do gradiente

Na figura 4.9 fizemos $\theta^* = 90$. Novamente o sistema se comporta como o esperado.

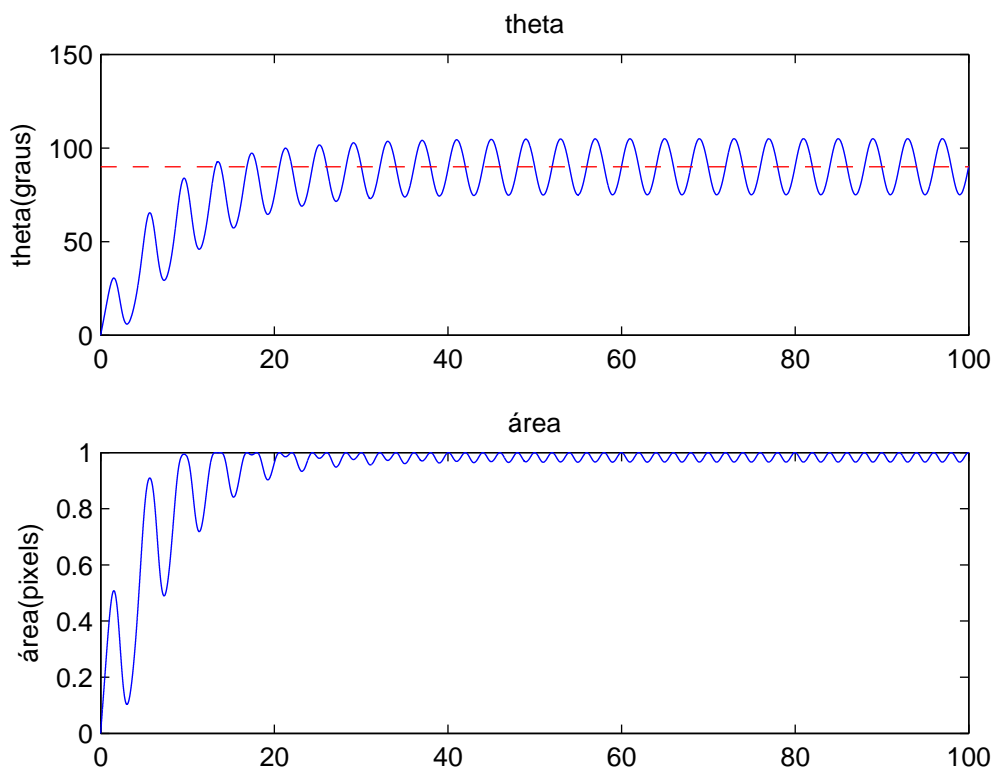


Figura 4.9: Segunda simulação com grau 0

4.2.3 Planta de Grau Relativo 2

A adição da dinâmica do motor realimento com controle P, insere uma planta de grau relativo 2 no sistema. A figura 4.10 mostra os resultados para $\theta^* = 30$. Perceba que a inserção dessas dinâmicas não alterou significativamente o comportamento do sistema. Se aproximarmos bastante o gráfico de θ podemos perceber que as oscilações estão com uma amplitude 2 graus menor do que no caso onde não existiam dinâmicas, o que é esperado de um modelo linear. O sinal de controle atinge valor máximo de 0.05V, muito abaixo de 1V, seu valor de saturação. Isso se deve à presença de reduções no motor e à baixa frequência das oscilações. O sinal do controlador e de θ oscilam com período de 4 segundos como o esperado.

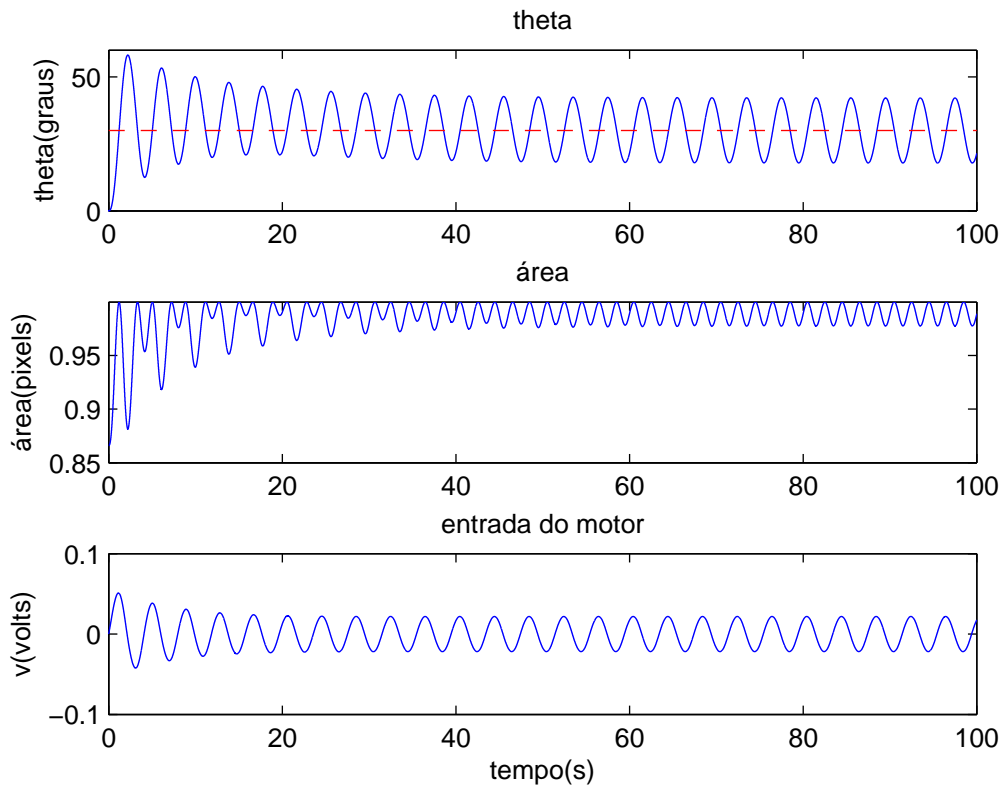


Figura 4.10: Simulação com grau 2.

Como segundo teste, fez-se $\theta^* = 90$. A figura 4.11 apresenta o resultado. Podemos observar que dessa vez aparece uma mudança significativa no comportamento do sistema em relação ao comportamento visto na ausência de dinâmicas, o tran-

sitório até o extremo é muito mais longo. As oscilações também estão menores como na simulação anterior. No entanto, o período permanece 4 segundos, como esperado em um sistema linear.

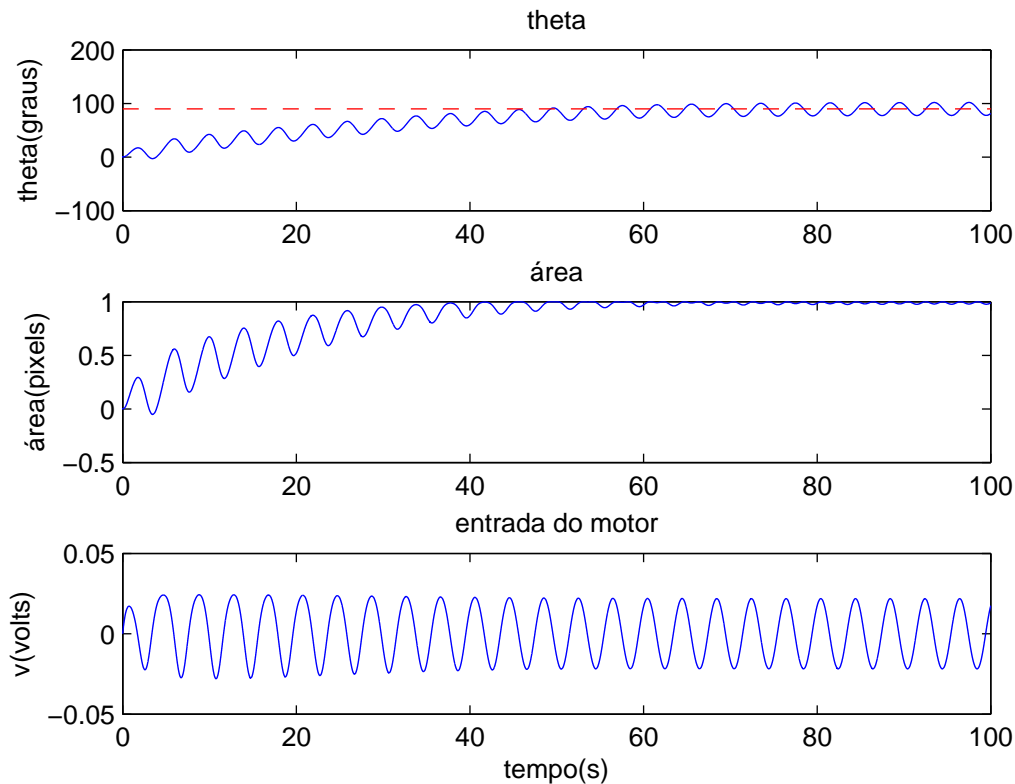


Figura 4.11: Segunda simulação com grau 2.

Podemos tentar melhorar o transitório diminuindo o valor de $T_{transitorio}$. Fazamos no igual a 20, mantendo $a = 15$ e $h = 0.1$, e calculando os outros parâmetros como feito anteriormente. Obtemos o resultado da figura 4.12, que não melhorou significativamente o tempo de transitório. Se escolhermos um valor menor ainda, digamos $T_{transitorio} = 10$, o valor de k se tornará grande demais levando o sistema à instabilidade como pode ser visto na figura 4.13. Vale lembrar que com uma planta de grau relativo 0, isso não acontece. Isso mostra que o controlador projetado para planta de grau relativo 0 pode acabar levando um sistema com uma planta de grau relativo 2 à instabilidade, caso k seja grande demais.

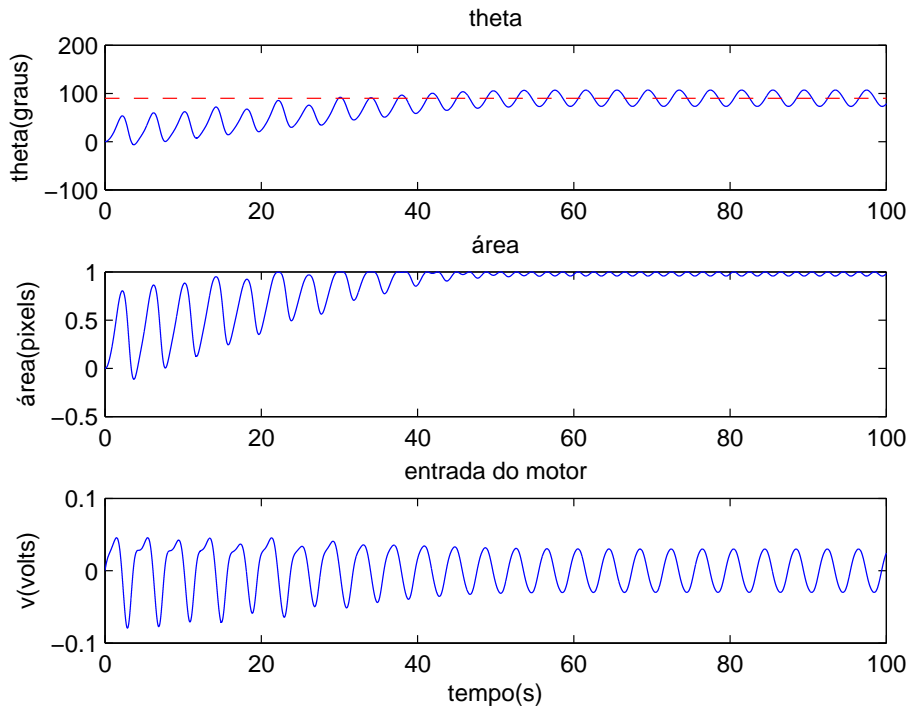


Figura 4.12: Simulação diminuindo o transitório.

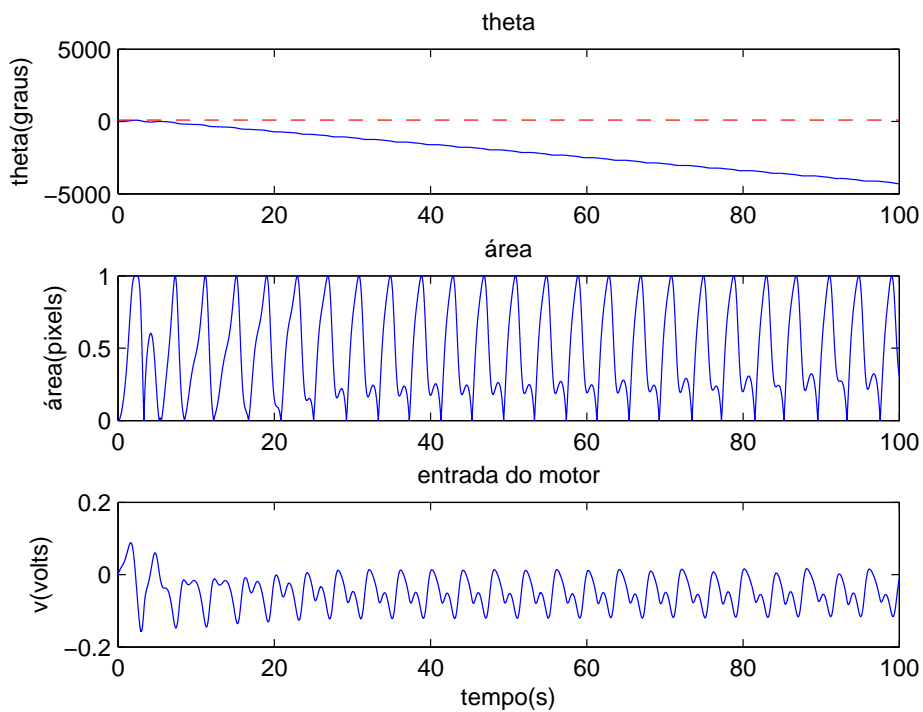


Figura 4.13: Simulação instável!

4.3 Simulações - Controle Extremal por Modos Deslizantes Via Função de Chaveamento

Nesta Seção realizaremos simulações usando o método de Controle Extremal por Modos Deslizantes Via Função de Chaveamento Periódica como visto no capítulo 2. O modelo implementado em Simulink pode ser visto nas figuras 4.14 e 4.15.

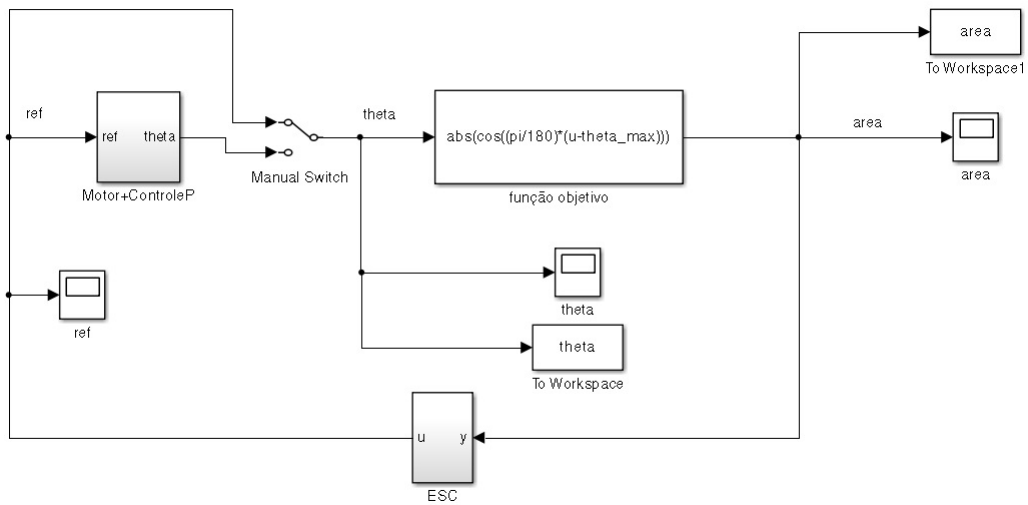


Figura 4.14: Modelo em Simulink de Controle Extremal por Modos Deslizantes Via Função de Chaveamento

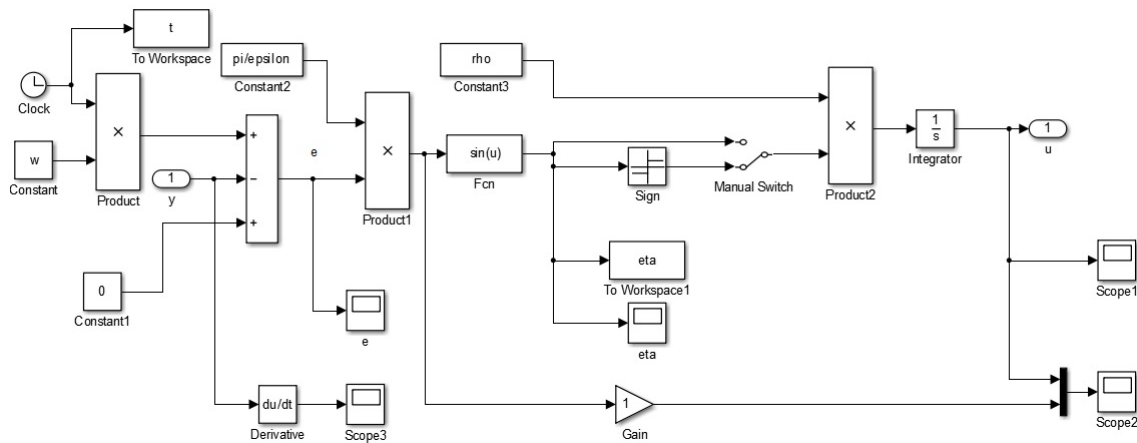


Figura 4.15: Bloco ESC do modelo em Simulink

4.3.1 Projeto

Para o projeto desse controlador, temos dois problemas: Dimensionar a região Δ de forma a garantir que haja deslizamento durante o transiente e que ao chegar em regime não haja mais, e que a frequência angular de η seja menor que o valor máximo obtido experimentalmente $\frac{2\pi}{3}$.

A primeiro problema pode ser resolvido usando a abordagem do capítulo 2. Se fixarmos o tamanho de Δ e um valor para ω , pode-se calcular o valor do parâmetro ρ . Dessa maneira, enquanto θ estiver fora da região Δ , haverá deslizamento. Dentro dela, o controlador exibirá um comportamento semelhante ao método da função periódica sem deslizamento.

Para resolver o segundo problema, basta observar que a frequência angular de η , que chamaremos de Ω , é igual a

$$\Omega = \frac{\pi\omega}{\epsilon}$$

Logo, fixando ω e escolhendo $\Omega < \frac{2\pi}{3}$, basta calcular ϵ tal que

$$\epsilon = \frac{\pi\omega}{\Omega} \tag{4.7}$$

Com essa abordagem, basta que fixemos os valores de Ω e ω , e o tamanho da região Δ , onde Ω está restrito ao problema da baixa taxa de amostragem. Vamos fixar seu valor em

$$\Omega = \frac{2\pi}{4}$$

Os valores de ω e Δ serão alterados de acordo com o necessário em cada simulação. Experimentalmente, percebeu-se que ω altera a amplitude da oscilação. Junto com o tamanho de Δ , esses parâmetros influenciarão a existência ou não de deslizamento.

4.3.2 Planta de Grau Relativo 0

Primeiramente foi escolhido $\theta^* = 30$, uma região Δ de tamanho 30 graus e $\omega = 0.004$. O resultado da simulação pode ser visto na figura 4.16. O transitório dura 5.8 segundos e o controle leva com sucesso a entrada da função objetivo a θ^* . A amplitude das oscilações é algo em torno de 8.5 graus. O período tem 4 segundos como esperado. Em 4.17, podemos observar os valores da variável η . Os limites da região Δ estão pontilhados em preto na figura. Observe que durante o transitório o controle funciona por deslizamento e que após θ oscila dentro da região Δ , de forma que não ocorre mais deslizamento.

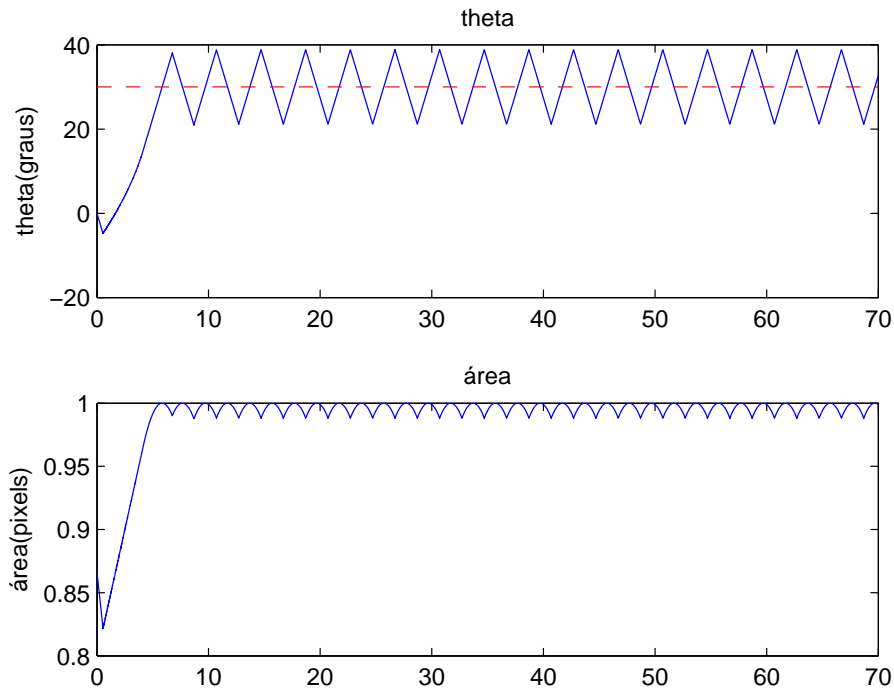


Figura 4.16: Simulação 1 com grau 0.

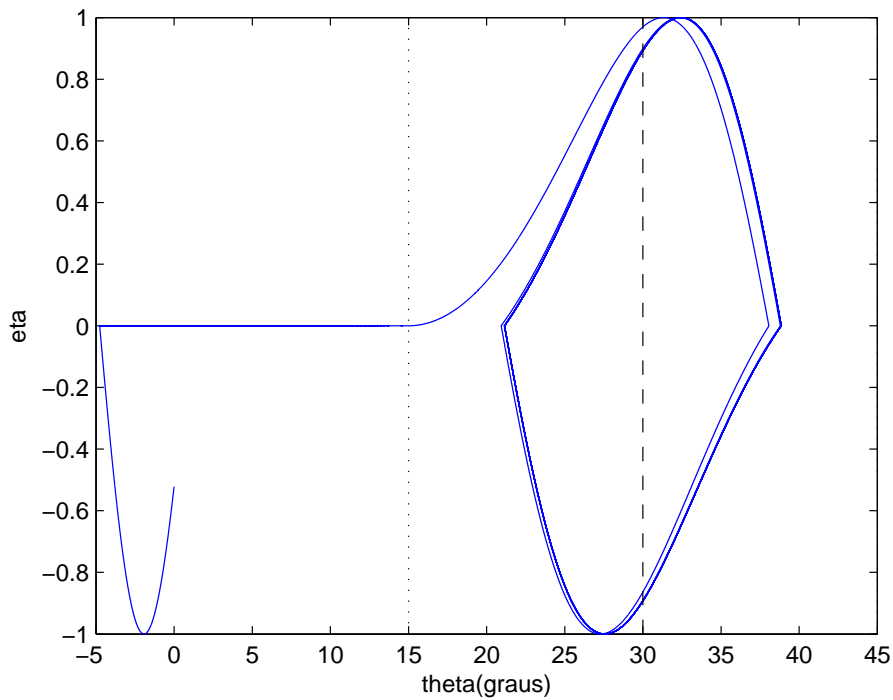


Figura 4.17: Gráfico theta X eta para simulação 1.

Diminuindo a região Δ para 10 graus e fazendo $\omega = 0.02$, obtemos os resultados da figura 4.18. A chegada ao máximo leva 6.1 segundos. Agora, as oscilações tem cerca de 11.2 graus, o que mostra que uma diminuição de ω aumenta as amplitudes de oscilação em torno do extremo. Podemos notar que a curva das oscilações em torno do extremo é ligeiramente diferente agora quando θ volta a se aproximar de seu valor ótimo. Isso se deve à presença de deslizamento durante as oscilações, o que pode ser visto na figura 4.19.

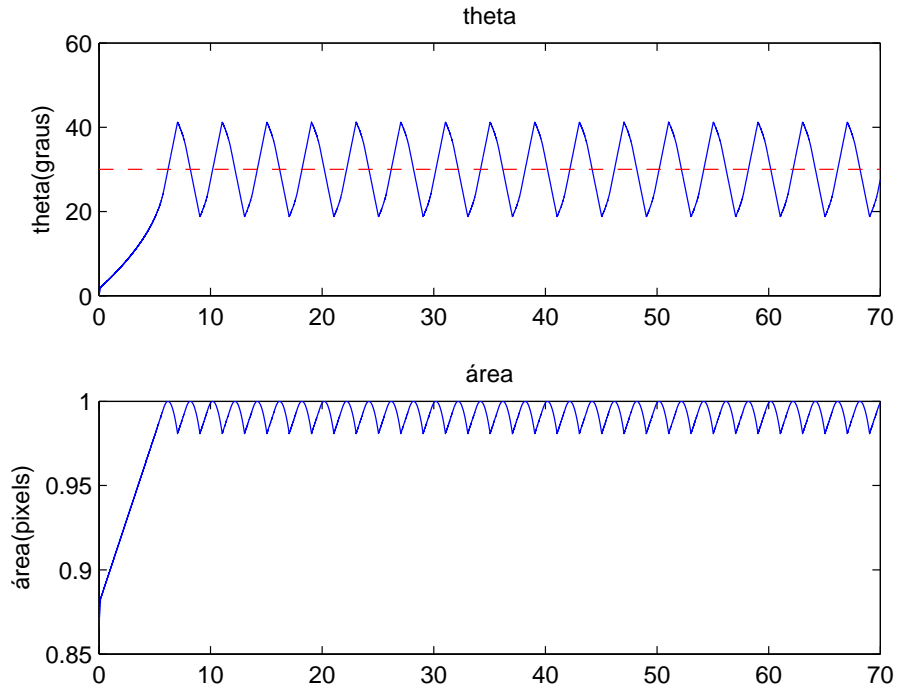


Figura 4.18: Simulação 2 com grau 0.

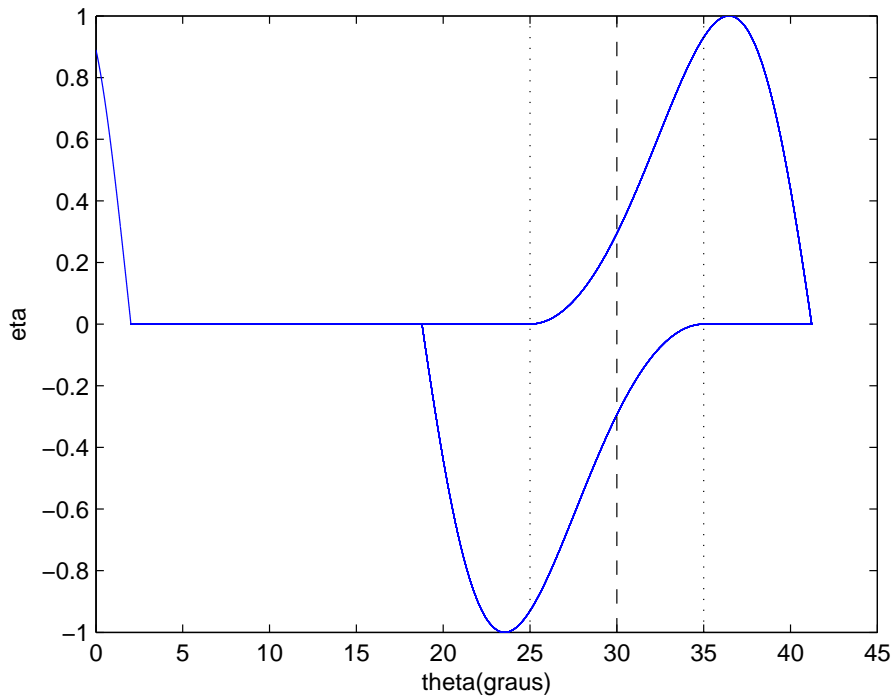


Figura 4.19: Gráfico theta X eta para simulação 2.

Por meio de simulações pode ser mostrado que quanto menor a região Δ , ou seja,

maior o trecho onde ocorre deslizamento, mais rápido o transitório. Por outro lado, quanto menor o valor de ω , mais lento é o transitório. Ou seja, o custo de oscilações menores é o aumento do tempo de transitório. Note que a presença de deslizamento aumenta a velocidade do transitório, porém aumenta as oscilações quando acontece no regime. Então, desejamos deslizamento somente no transitório.

Por fim, fazendo novamente a região Δ igual a 30 graus e fazendo $\omega = 0.04$, e fixando $\theta^* = 90$, obtemos os resultados apresentados na figura 4.20. Mesmo mais distante do máximo, o controlador alcançou seu objetivo. As oscilações se mantiveram com amplitude de 8.5 graus como anteriormente. Caso seja necessário melhorar o transitório basta alterar Δ ou ω como dito no parágrafo anterior.

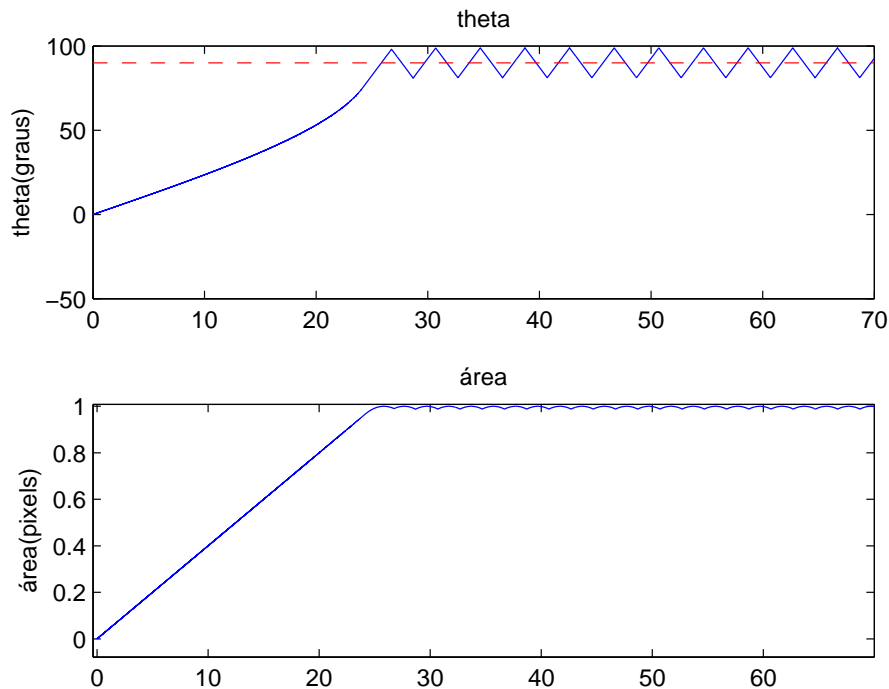


Figura 4.20: Simulação com grau 0 e extremo distante.

4.3.3 Planta de Grau Relativo 2

Adicionando a dinâmica do motor realimentado e fazendo $\Delta = 30$, $\omega = 0.04$ e $\theta^* = 30$, obtemos o resultado da figura 4.21. Em relação à simulação de grau 0 com esses mesmos parâmetros de controlador podemos notar que o tempo de transitório

ficou um pouco mais lento, agora 6.2 segundos. Como esperado de um sistema linear, as oscilações continuam com o mesmo período e amplitude ficou um pouco menor, 6.5 graus. As oscilações também estão mais próximas do desenho de uma função senoidal. O sinal de controle do motor possui valor bem pequeno em torno de 0.01V.

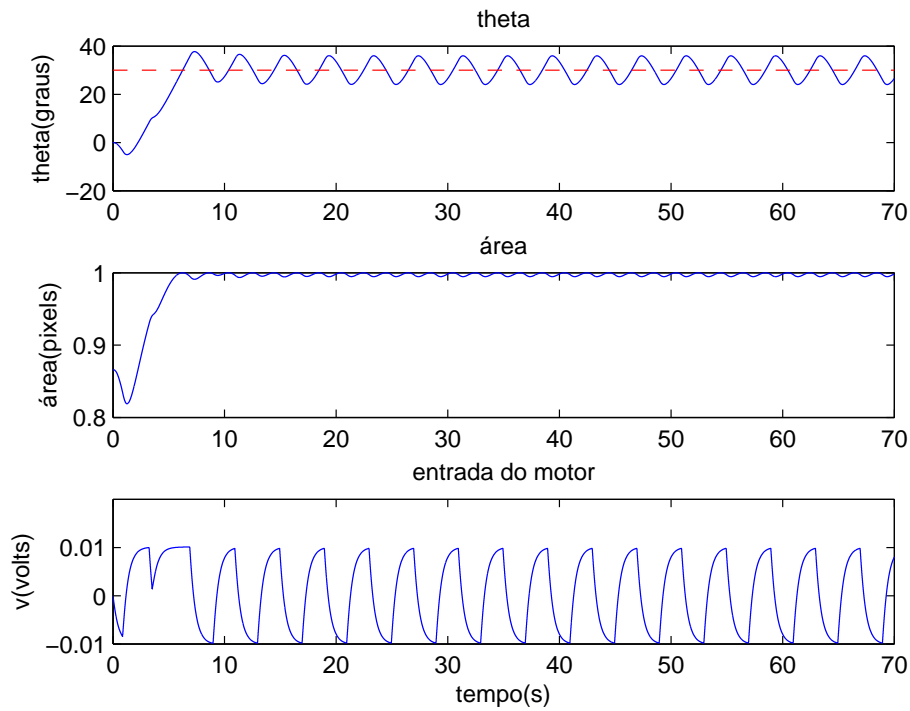


Figura 4.21: Simulação com grau 2.

A seguir, com $\theta^* = 90$ coletou-se os seguintes dados apresentados nos gráficos da figura 4.22. O transitório levou 28.5 segundos, valor próximo do transitório com planta de grau relativo 0. A amplitude das oscilações foi 10 graus. Note que o sinal de controle do motor possui módulo maior durante o regime que durante o transitório.

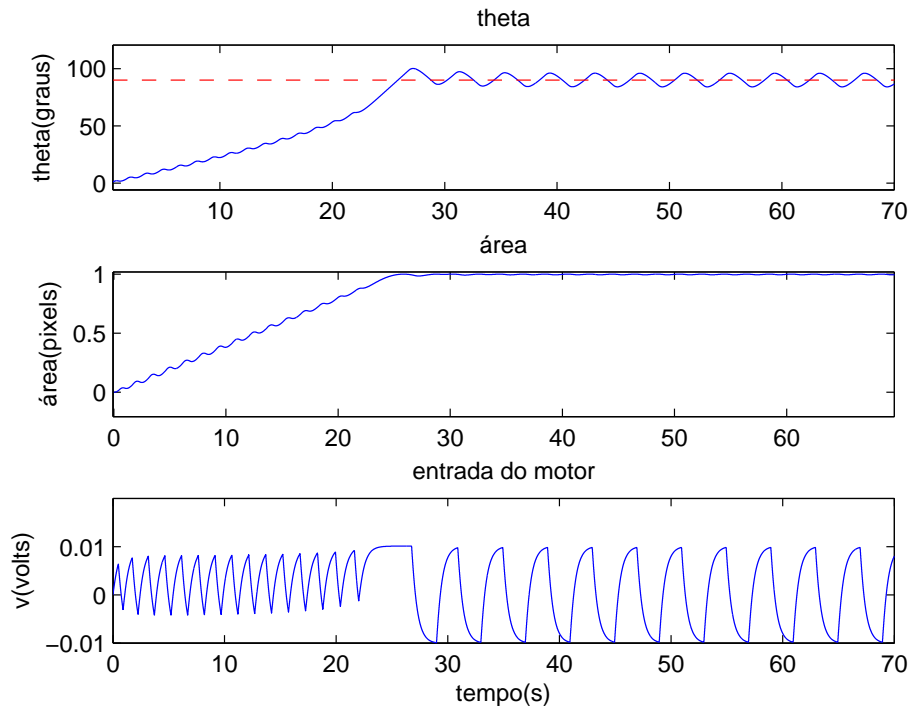


Figura 4.22: Simulação com grau 2 e ponto ótimo mais distante.

Na tentativa de melhorar o tempo de transitório diminui-se Δ para 10 graus. Isso resultou no comportamento da figura 4.23. O transitório ficou cerca de 3 segundos mais rápido a custo de maiores amplitude de oscilação em regime. No entanto ao fazer $\Delta = 6$ o controle parou de funcionar como pode ser visto em 4.24. O mesmo não acontece com uma planta de grau relativo 0.

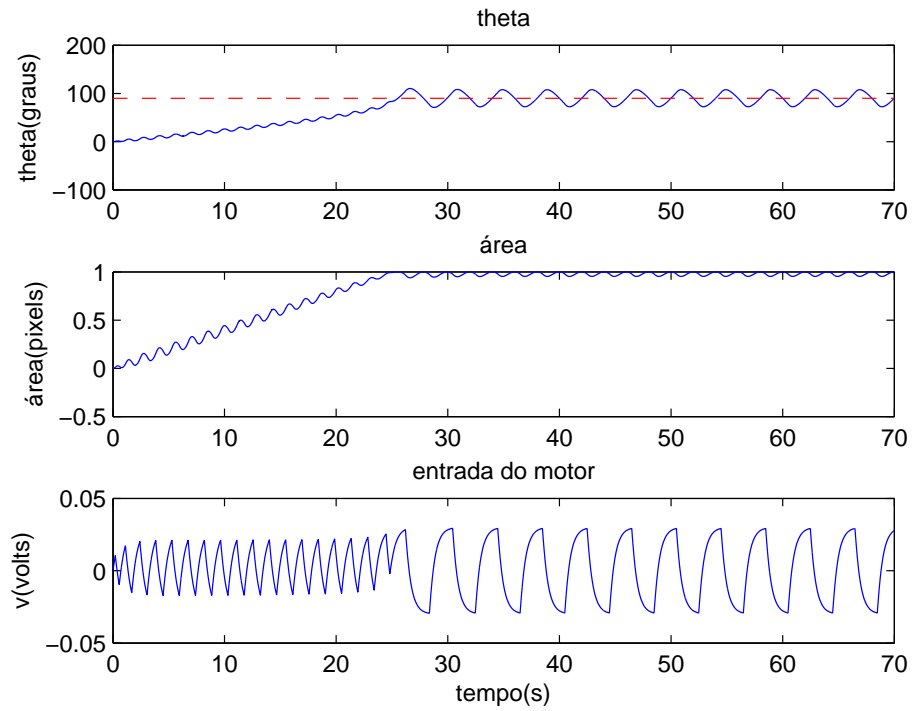


Figura 4.23: Tentativa 1 de melhorar transitório.

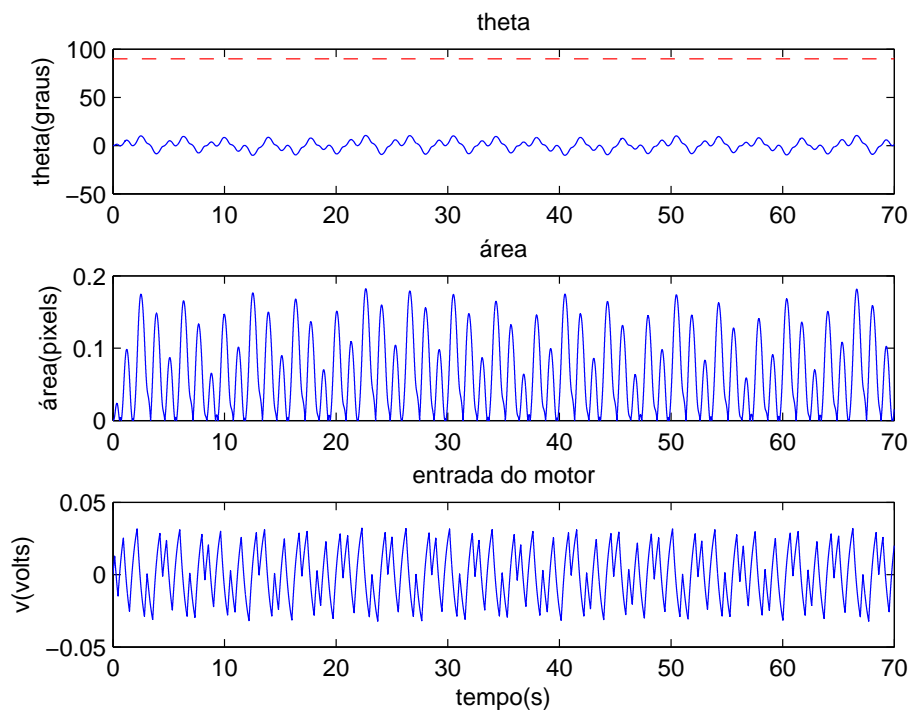


Figura 4.24: Tentativa 1 de melhorar transitório. Falha do controle.

4.4 Resultados Experimentais

Esta seção tratará da análise dos dados obtidos experimentalmente. Pretende-se estudar como os do controladores projetados neste capítulo se comportam na presença de dinâmicas não-modeladas, incertezas, não-linearidades como atrito seco, sinal de ruído e taxa de amostragem relativamente baixa. Ao comparar esses resultados com os obtidos por meio simulações espera-se adquirir uma noção da robustez do controlador, seu desempenho na prática e suas limitações. Os controladores usados aqui serão projetados da mesma forma que foram nas simulações.

4.4.1 Método das Perturbações Senoidais

Para seguir a ordem que vem sendo mantida nesse trabalho, começaremos os experimentos pelo método das perturbações senoidais. Para facilitar comparações, os parâmetros do controlador serão próximos àqueles usados durante as simulações numéricas.

Na primeiro experimento, ajustamos os parâmetros da seguinte maneira: $a = 10$, $w = \frac{2\pi}{4}$, $h = 0.5$, $f'' = -0.0002$ e, para conseguir um transitório mais rápido, $T_{transitorio} = 20$. O parâmetro k foi calculado usando o método já apresentado. Os resultados do experimento podem ser vistos nas figuras 4.25, 4.26 e 4.27. Podemos notar que o sinal de y converge para uma região bem estreita em torno máximo da função. Note que esse valor não é necessariamente 1 porque o valor máximo depende da distância da câmera até alvo. O controlador permitiu que θ rastreasse o sinal de referência com sucesso. Se prestarmos atenção podemos observar um leve achatamento no pico das oscilações do sinal de θ , efeitos não-lineares devido à presença de atrito, mas que não chegam a prejudicar o rastreamento do sinal de referência. O período das oscilações tem 4 segundos como o esperado. O sinal de controle não chegou nem a 1V e está muito longe da região de saturação.

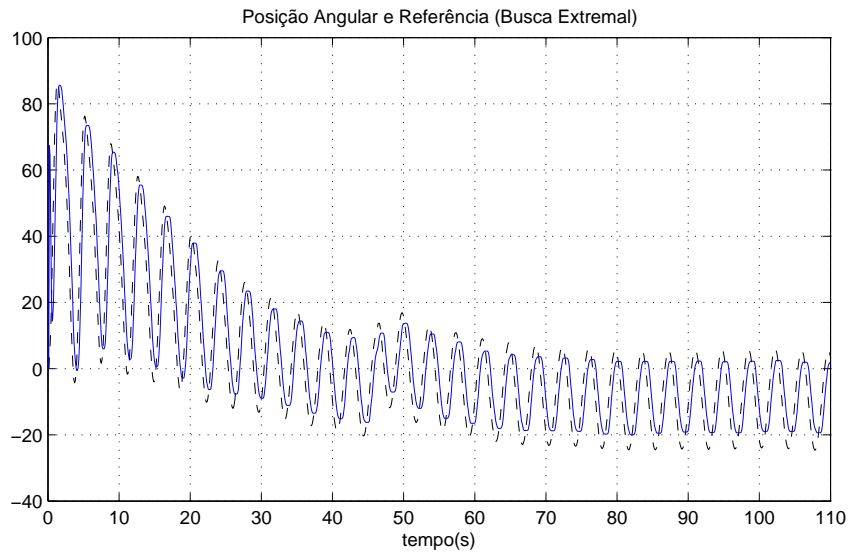


Figura 4.25: Comparação do ângulo do motor com a referência fornecida por ESC.

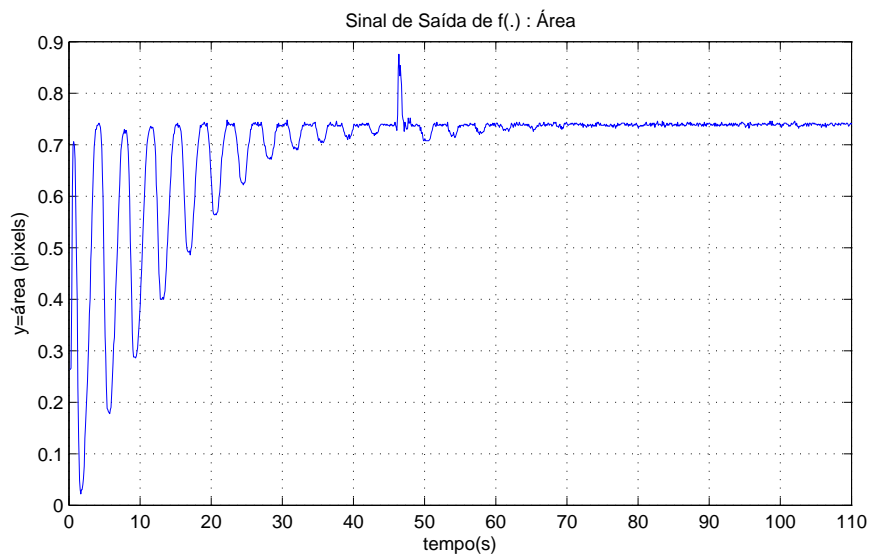


Figura 4.26: Sinal da área.

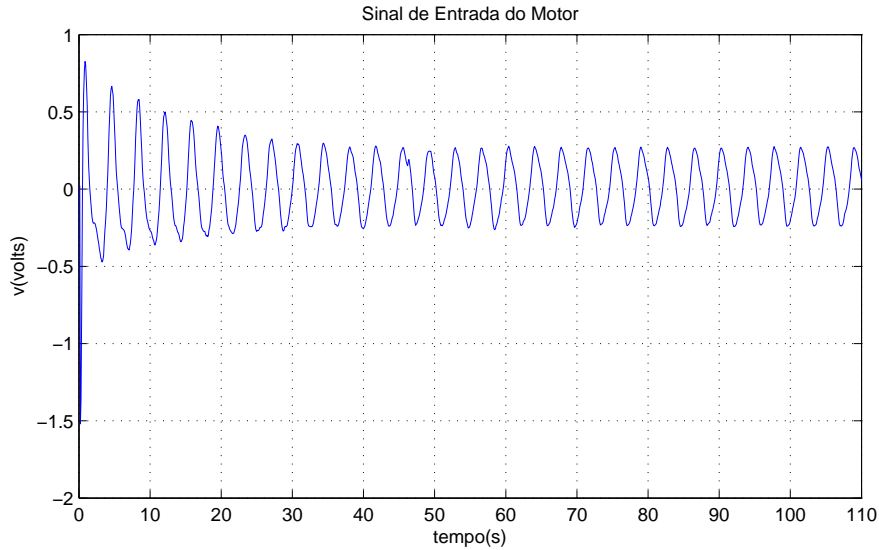


Figura 4.27: Sinal de controle do motor.

Observe o sinal de y , a área do alvo projetada no plano da imagem da câmera, e perceba o aparecimento súbito de um pico no instante 48s. Isso aconteceu porque a câmera perdeu o foco distorcendo a imagem por algum tempo. Enquanto a imagem esteve desfocada a área do alvo foi calculada maior do que realmente era. Note que o ângulo θ chegou a se deslocar, a procura de um novo ótimo, mas logo que o sinal y retornou ao normal, θ também voltou. Isso mostra que esse método de controle é robusto quanto à adição de impulsos no sinal de saída da função objetivo.

No segundo teste, queremos saber se o algoritmo de controle é robusto à mudanças do ponto-ótimo θ^* , assim como mudanças do valor máximo da função y^* . Para testar isso, moveu-se a posição da câmera algumas vezes durante a realização do experimento. Tanto o seu ângulo em torno do eixo do motor, quanto à sua distância até seu centro, foram mudados durante o teste. Para tentar melhorar o tempo de transitório em relação ao teste anterior, foi escolhido $T_{transitorio} = 15$. Para melhorar a busca pelo máximo global da função $f(\theta)$ também escolhemos $a = 15$, melhorando a detecção de mudança do ponto-ótimo e diminuído a chance de se ficar preso a um máximo local, caso exista um. O resultado pode ser visto nas figuras 4.28, 4.29 e 4.30. Inicialmente, a câmera está na posição angular 10 graus. Ao movermos a câmera quase que instantaneamente para a posição 90 graus, o contro-

lador atuou de forma a mandar θ para a nova região do máximo. Isso mostra que o controlador percebeu a mudança do valor de θ^* . Observado o sinal de y vemos que a nova região de máximo foi encontrada. Em seguida, levamos a câmera rapidamente para o ângulo 0 e, logo depois, para o ângulo -90 graus. O controlador reagiu fazendo o alvo seguir a câmera e podemos ver em 4.29 que o sinal da área volta a crescer. No entanto, antes que o transitório tivesse acabado movemos a câmera de volta ao ângulo zero e a afastamos bastante do alvo. O controlador novamente seguiu a câmera alterando o ângulo θ , que é levado até em torno de 0 graus. Como a câmera havia sido afastada, podemos perceber que o novo valor y^* é bastante menor que os valores anteriores.

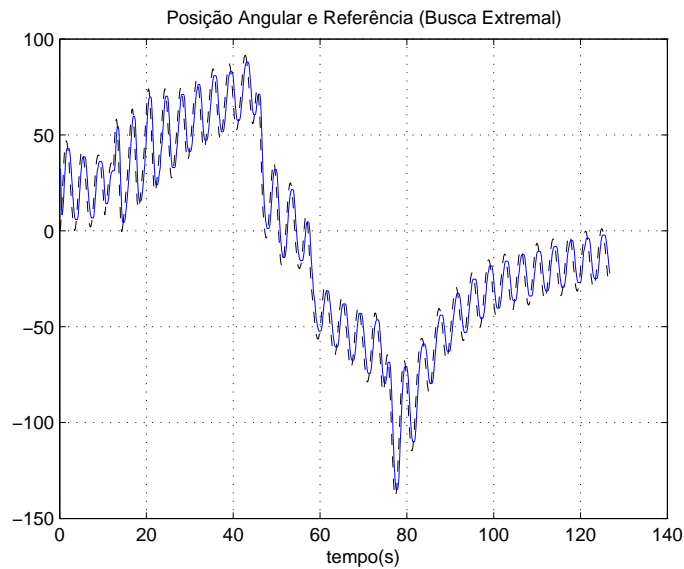


Figura 4.28: Comparação do ângulo do motor com a referência fornecida por ESC.

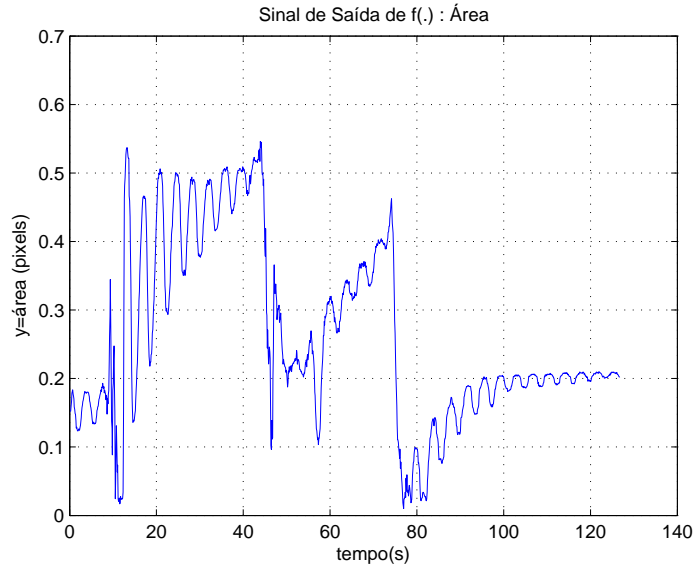


Figura 4.29: Sinal da área.

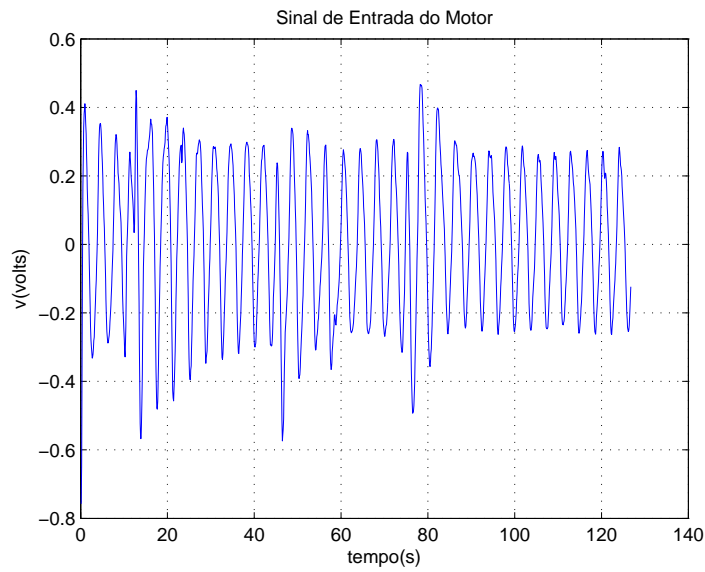


Figura 4.30: Sinal de controle do motor.

O método das perturbações senoidais foi capaz de realizar seu objetivo com sucesso, mesmo quando a função objetivo $f(\theta)$ passou por diversas mudanças. O controlador projetado para uma planta-linear de dinâmica desprezível se mostrou robusto tanto diante de mudanças modelo da planta quanto na função objetivo. O ponto fraco do controlador projetado foi o longo tempo de transitório. Devido ao valor muito pequeno de f'' , o controlador demora a estimar a posição de θ^*

corretamente. O sistema projetado não foi, por exemplo, capaz de seguir a câmera em movimento de maneira satisfatória. Apesar disso, para mudanças lentas dos parâmetros do sistema, o controlador funciona eficientemente.

4.4.2 Controle Extremal por Modos Deslizantes Via Função de Chaveamento

Como foi dito anteriormente devido a taxa de amostragem baixa do experimento, para evitar o aparecimento de não-linearidades indesejadas, Ω deve ser pequeno. Para realizar o primeiro experimento o controlador foi projetado usando os mesmos parâmetros das simulações numéricas, ou seja

$$\Delta = 30, \omega = 0.04 \text{ e } \Omega = \frac{2\pi}{4}$$

O primeiro experimento consistia tão somente em θ alcançar θ^* , enquanto a webcam era mantida fixa na mesma posição. O resultado obtido pode ser visto em 4.31, 4.32 e 4.33. Observe que menos de 10 segundos são o bastante para que θ alcance seu valor ótimo, que no caso é aproximadamente -4 graus. Podemos ver que o sinal de θ apresenta um leve achatamento no pico das oscilações o que é provavelmente devido a presença de atrito seco. O sinal y alcança seu valor máximo e oscila muito pouco em torno dele. É difícil saber se a variação de sinal é devido à oscilação do ângulo ou devido ao ruído da câmera. Em geral, o controlador cumpriu seu objetivo com sucesso e portou-se de maneira muito parecida com as simulações, apesar do desenho do sinal de θ estar um pouco distorcido quando comparado ao sinal das simulações numéricas.

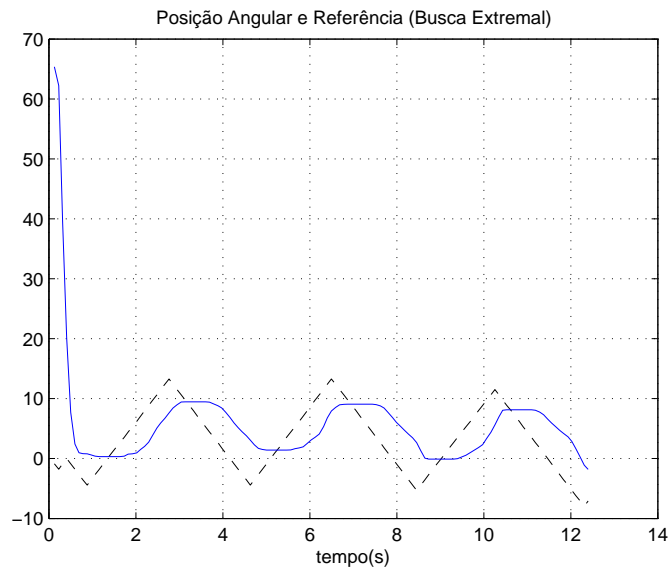


Figura 4.31: Comparação do ângulo do motor com a referência fornecida por ESC.

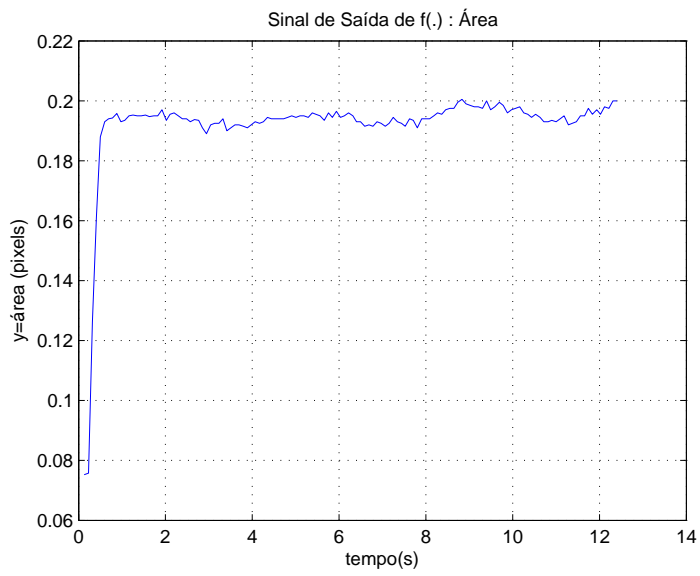


Figura 4.32: Sinal da área.

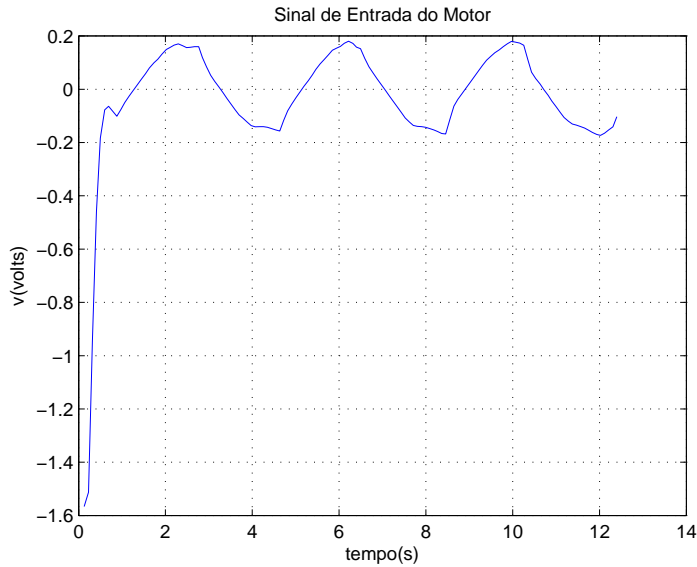


Figura 4.33: Sinal de controle do motor.

A seguir foi realizado um novo teste fazendo $\omega = 0.08$ e mantendo Ω e Δ iguais. Dessa vez a câmera foi transladada em torno da eixo do motor. Como a posição da câmera não é medida, é difícil saber qual é o seu ângulo exato, mesmo assim, ainda é possível saber se o controlador resiste a mudanças na posição de θ^* . O resultado é apresentado nas figuras 4.34, 4.35 e 4.36. Inicialmente a câmera está em -60 graus, em seguida é levada para próximo de zero graus, depois para 60 graus aproximadamente e, então, de volta para próximo a 0 graus. Em todos os movimentos, o controlador logo achou o máximo de y e permaneceu oscilando em torno dessa região. Perceba também que, como previsto nas simulações, o aumento em ω aumentou o tamanho das oscilações como pode ser visto claramente tanto em θ quanto em y . Além disso, como havia sido dito anteriormente oscilações de amplitude maiores diminuiram o efeito das não-linearidades, fazendo com que θ siga melhor o sinal da referência e se assemelhe bastante aos resultados numéricos obtidos anteriormente. Por fim, perceba que o sinal de entrada do motor se mantém em uma faixa segura muito distante de sua saturação.

O controlador foi capaz de alcançar o máximo da função mesmo quando o ponto ótimo mudou diversas vezes. Se o movimento da câmera for suficientemente lento, o controlador é capaz de seguir a trajetória da câmera. Numa aplicação mais indus-

trial, como o rastreamento de um satélite, por exemplo, temos fortes indícios que a metodologia desse controlador poderia realizar com sucesso tal tarefa.

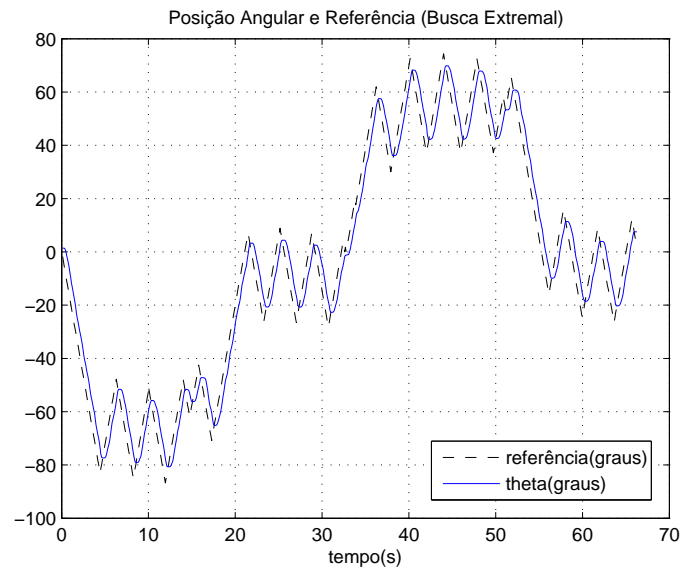


Figura 4.34: Comparação do ângulo do motor com a referência fornecida por ESC.

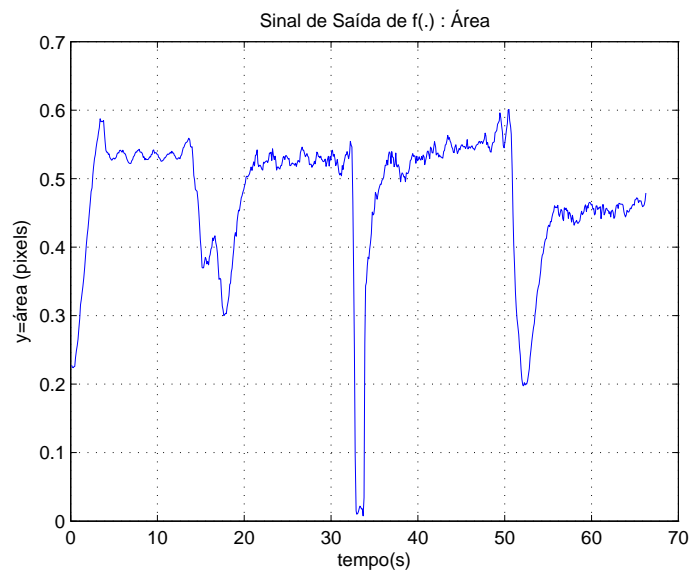


Figura 4.35: Sinal da área.

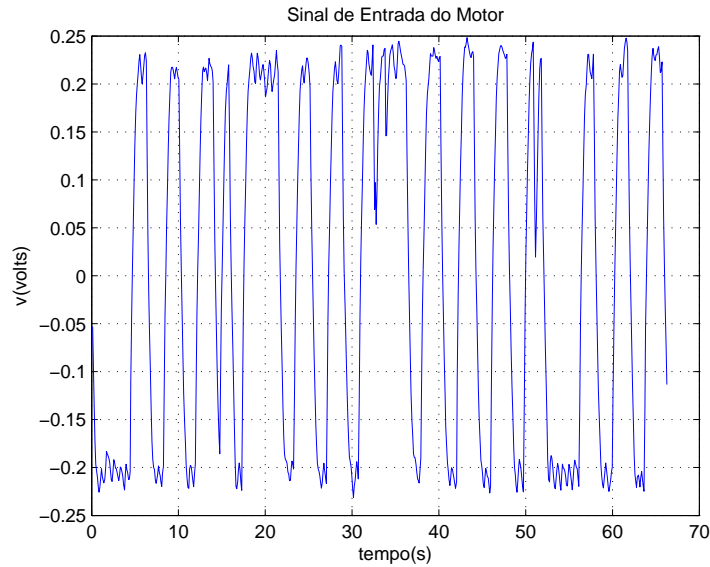


Figura 4.36: Sinal de controle do motor.

Percebemos que o controlador projetado foi capaz de realizar sua função mesmo quando exposto a uma série de condições adversas que não foram levadas em conta durante o seu projeto, como taxa de amostragem baixa, ruídos e incertezas tanto no modelo da planta quanto no modelo da função objetivo. Comparando-o com o método das perturbações senoidais, podemos dizer que o controle baseado em deslizamento apresenta um transitório muito mais rápido.

4.5 Proposta para Atenuar a Oscilação em Regime

Um dos problemas apresentado pelos métodos de controle estudados nesse trabalho é o fato que ao chegar em regime a variável de entrada θ da função de interesse continua oscilando rapidamente e com grande amplitude em torno do ponto ótimo quando isso não seria mais necessário, pois o extremo já foi alcançado. Isso provoca um desgaste desnecessário no atuador e prejudica o desempenho da variável que queremos otimizar, que idealmente gostaríamos que permanecesse fixa no valor máximo (ou mínimo, caso se trate de um problema de minimização). De forma a

mitigar esses problemas, precisamos mudar a estratégia de controle quando o ponto ótimo tiver sido alcançado. Para isso, precisamos de

- uma maneira de detectar a mudança de transiente para regime
- uma nova estratégia de controle capaz de melhorar o desempenho em regime e desgastar menos o atuador
- caso haja uma mudança de posição do ponto ótimo θ^* , a nova proposta deve ser capaz de detectá-la e retornar a estratégia de controle original até que seja percebida a chegada a um novo regime

Para esse fim, foi proposto o seguinte modificação do esquema controle das perturbações senoidais. Sabendo que a variável ξ do modelo da figura representa a estimação do gradiente da função $f(\theta)$, é esperado que, ao alcançar o regime, esse valor oscile em torno de zero. Isso pode ser visto na figura ???. Calculando a média de ξ , que chamaremos de ξ_{med} , e estabelecendo um valor de *threshold*, usou-se a seguinte regra de controle:

- Se $|\xi_{med}| < threshold$, então diminua os valores dos parâmetros ω e a do controlador original
- Caso contrário, utilize os valores originais

A média ξ_{med} leva em conta somente os último 32 segundos de simulação, ou seja, 8 períodos do seno de perturbação.

Para testar a estratégia, o modelo das perturbações senoidais usado durante as simulações numéricas foi ligeiramente adaptado como pode ser visto na figura 4.38. Os resultados da nova simulação podem ser vistos na figura 4.37. O problema simulado foi o seguinte: Inicialmente, θ^* é 30 graus e no instante 200 segundos θ^* assume o valor 0 graus. Isso é equivalente a mudar a posição da câmera no experimento prático. Durante o transiente, os valores dos parâmetros do controlador são os mesmos usados durante as simulações numéricas para planta de grau relativo

0. Ao chegarmos na região de máximo, ξ_{med} se aproxima de zero, e detectamos a chegada ao regime permanente. Adaptamos o controlador fazendo $\omega = 2 * \pi / 16$ e $a = 5$ de forma a desgastar menos o atuador e diminuir as oscilações. Quando θ^* muda de posição, o gradiente estimado ξ_{med} lentamente se afasta de zero e então é detectada o início do novo transiente, fazendo com que o controlador volte a sua configuração inicial, até atingir novo regime. Nota-se também que a mudança de controle permite que o sinal de y mantenha-se muito mais próximo do valor máximo. Perceba que a simulação foi realizada com a presença da dinâmica do motor.

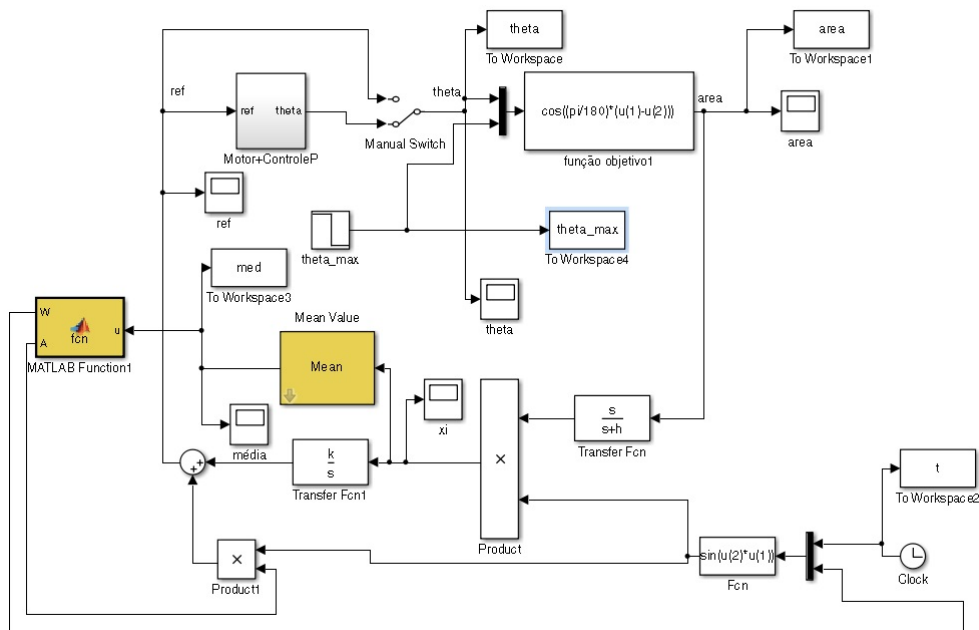


Figura 4.37: Modelo Simulink com detecção de chegada ao regime

Observe que no instante 107 é detectado o fim do transiente e acontece a mudança na estratégia de controle. Observando o sinal de controle do motor, percebemos que as oscilações e a amplitude são muito menores o que indica que o atuador está sofrendo menos desgaste. Note também que y se mantém mais próximo de seu valor máximo. No instante 200, ao mudarmos a posição de θ^* , a média de ξ começa a se deslocar e no instante 213 segundos, o controlador volta a atuar de forma normal. O chegar no extremo, o controlador troca novamente com sucesso.

Mostramos que o simples cálculo da média de um sinal e

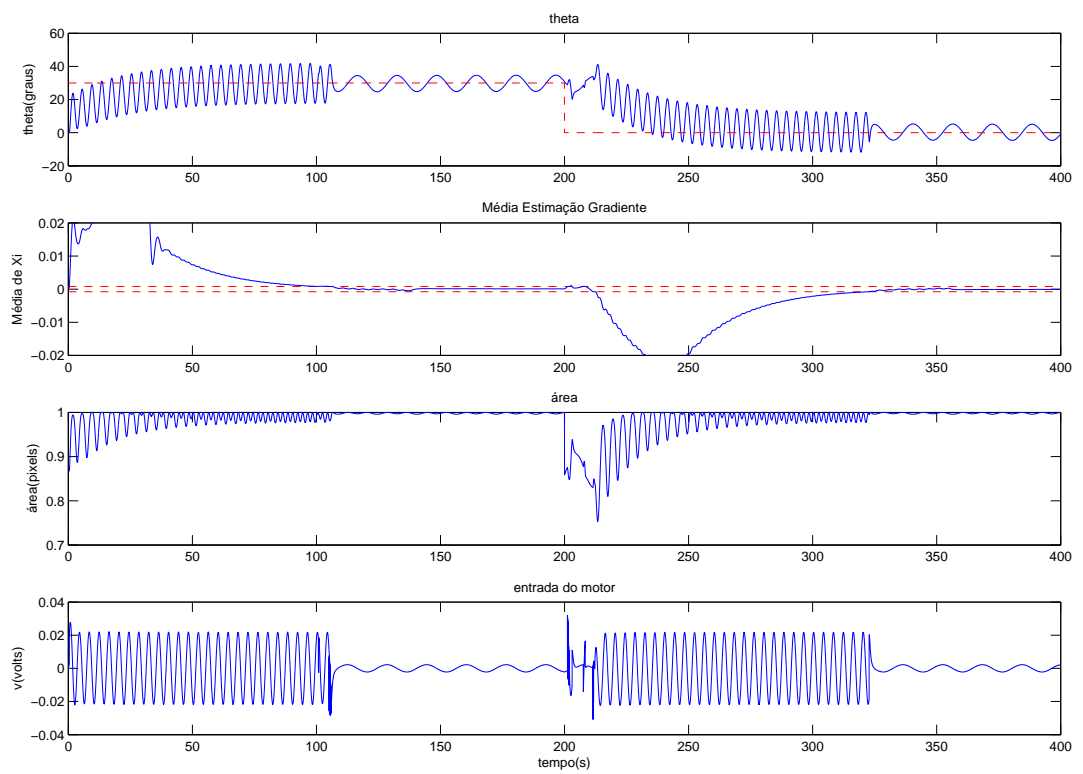


Figura 4.38: Simulação com detecção de chegada ao regime

Capítulo 5

Conclusões e Trabalhos Futuros

Durante o trabalho foi possível estudar o funcionamento e o desempenho de alguns métodos de controle por busca extremal. Inicialmente, fomos introduzidos ao assunto, a sua história e a algumas das principais aplicações que vem sendo pesquisadas. Logo em seguida, nos aprofundamos nos métodos de busca extremal que seriam o foco desse, analisando aspectos teóricos, seus comportamentos, especificidades e semelhanças.

No capítulo 3, nos concentramos nos detalhes da implementação do experimento prático, no funcionamento do algoritmo do software e também no desempenho e limitações do sistema físico construído.

Durante as simulações, vimos que o controladores projetados alcançaram seu objetivo com sucesso e de maneira satisfatória. Mesmo quando inserimos uma planta de grau relativo 2, sem alterar o controlador, o máximo da função objetivo foi alcançado com sucesso sem perdas significativas de desempenho. Comparando os métodos estudados, percebemos nas simulações que o método do controle extremal por modos deslizantes via função de chaveamento periódica apresentou um transitório mais rápido e menos oscilatório do que o método das perturbações senoidais. Comparando as simulações de grau relativo 2 com θ^* era 90 graus, o desempenho máximo alcançado pelo primeiro método foi cerca de 2 vezes mais rápido que o desempenho máximo alcançado pelo método das perturbações senoidais.

No experimento prático, constatamos que o controlador projetado na seção das

simulações numéricas mostrou-se robusto e alcançou a região do extremo da função objetivo com um bom desempenho mesmo na presença de ruído, incertezas nos modelos, dinâmicas não-modeladas, não-linearidades e baixa taxa de amostragem. Mesmo ao mudarmos a posição do ponto ótimo da função objetivo, o controlador do experimento foi capaz de encontrá-lo outra vez.

Por fim, por meio da proposta de adaptação do controlador, vimos que é possível identificar a chegada ao extremo da função objetivo e então alterar o funcionamento do sinal de controle, de forma a exigir menos dos atuadores e obter melhor desempenho em regime. Além disso, a adaptação proposta é capaz de identificar mudanças no ponto-ótimo da função objetivo e, então, voltar ao esquema de funcionamento original até alcançar o novo regime.

Esse trabalho contribui com novas análises do funcionamento do controle extremal por modos deslizantes via função de chaveamento periódica, uma nova interpretação dos efeitos dos parâmetros desse controlador e estudo de seu funcionamento fora e dentro da região de deslizamento, comparando-o com o método das perturbações senoidais. Além disso, durante o desenvolvimento desse trabalho foi implementado um experimento original para testar os resultados teóricos obtidos. Ademais, foi proposto um novo esquema de controle para melhorar os desempenhos obtidos.

Todos os bons resultados obtidos são mais um indicador da viabilidade da aplicação dessa estratégia de controle na indústria e em outros problemas de engenharia.

Os trabalhos futuros são inúmeros. Pode-se estudar outros métodos de controle por busca extremal e compará-los com o desempenho dos métodos utilizados nesse trabalho. Além disso, podemos melhorar os controladores projetados aqui tentando compensar o efeito das dinâmicas das plantas.

Quanto a proposta de adaptação do controlador, é possível estudar outras formas de se detectar o transiente e como alterar o esquema do controlador em regime. Além disso, é necessário fazer uma análise matemática mais rigorosa, o que não foi feito nesse trabalho, e testar a adaptação em situações mais variadas. Seria

interessante também implementar a proposta no controlador do experimento prático para observar seu desempenho.

O sistema montado para o experimento prático teve como grande limitador sua baixa taxa de amostragem. Para melhorá-la são propostas duas soluções para trabalhos futuros. A primeira seria calcular o sinal de controle da busca extremal no microcontrolador em vez de no programa em Matlab. Isso eliminaria a necessidade do software realizar a leitura dos dados do encoder via serial, o que representa cerca de 40% do tempo de cada loop. O software se concentraria apenas na medida da área do alvo e enviaria os resultados por serial, tarefa bastante rápida.

A segunda solução seria implementar todo software em linguagem C, que realiza a maioria das tarefas, especialmente comunicação serial, de maneira significativamente mais veloz que a linguagem Matlab. Para realizar o processamento de imagem poder-se-ia usar a biblioteca *open-source* OpenCV, que é muito rápida, intuitiva e possui muita documentação para consulta.

Referências Bibliográficas

- [1] KRSTIC, M., W. H.-H. “Stability of extremum seeking feedback for general nonlinear dynamic systems”, *Automatica*, pp. 595–601, 2000.
- [2] TAN, Y., MOASE, W., MANZIE, C., et al. “Extremum seeking from 1922 to 2010”. In: *Control Conference (CCC), 2010 29th Chinese*, pp. 14–26. IEEE, 2010.
- [3] ARIYUR, K., KRSTIC, M. *Real-Time Optimization by Extremum-Seeking Control*. Wiley-interscience publication. Wiley, 2003. ISBN: 9780471468592. Disponível em: <<http://books.google.com.br/books?id=7529G1SFaPIC>>.
- [4] AMINDE, N. O. *CONTROLE EXTREMAL VIA FUNÇÕES DE MONITORAÇÃO E REALIMENTAÇÃO DE SAÍDA*. Tese de mestrado, coppe/ufrj, 2013.
- [5] LEBLANC, M. “Sur l’électrification des chemins de fer au moyen de courants alternatifs de fréquence élevée”, *Revue Générale de l’Electricité*, 1922.
- [6] DRAPER, C., L. Y. *Principles of Optimizing Control Systems and an Application to the Internal Combustion Engine*. R. Oldenburger, 1951.
- [7] KRSTIĆ, M. “Performance improvement and limitations in extremum seeking control”, *Systems & Control Letters*, v. 39, n. 5, pp. 313–326, 2000.
- [8] NOGUEIRA, E. M. “Controle via Busca Extremal da Produção de Petróleo em Poços Operando com Elevação Artificial por Injeção de Gás”, 2012.
- [9] YU, H., OZGUNER, U. “Extremum-seeking control strategy for ABS system with time delay”. In: *American Control Conference, 2002. Proceedings of the 2002*, v. 5, pp. 3753–3758. IEEE, 2002.
- [10] ZHANG, C., ORDÓNEZ, R. “Non-gradient extremum seeking control of feedback linearizable systems with application to ABS design”. In: *Decision and Control, 2006 45th IEEE Conference on*, pp. 6666–6671. IEEE, 2006.

- [11] FU, L., OZGUNER, U. “Extremum-Seeking Control in Constrained Source Tracing With Nonholonomic Vehicles”, *Industrial Electronics, IEEE Transactions on*, v. 56, n. 9, pp. 3602–3608, 2009. ISSN: 0278-0046. doi: 10.1109/TIE.2009.2023637.
- [12] KILLINGSWORTH, N., KRSTIC, M. “Auto-tuning of PID controllers via extremum seeking”. In: *American Control Conference, 2005. Proceedings of the 2005*, pp. 2251–2256. IEEE, 2005.
- [13] KILLINGSWORTH, N. J., ACEVES, S. M., FLOWERS, D. L., et al. “HCCI Engine Combustion-Timing Control: Optimizing Gains and Fuel Consumption Via Extremum Seeking”, *Control Systems Technology, IEEE Transactions on*, v. 17, n. 6, pp. 1350–1361, 2009. ISSN: 1063-6536. doi: 10.1109/TCST.2008.2008097.
- [14] BINETTI, P., ARIYUR, K. B., KRSTIC, M., et al. “Formation flight optimization using extremum seeking feedback”, *Journal of guidance, control, and dynamics*, v. 26, n. 1, pp. 132–142, 2003.
- [15] KRSTIC, M., W. H.-H. B. G. “Optimizing Bioreactors by Extremum Seeking”, *Adaptive Control and Signal Processing*, 1998.
- [16] PETERSON, K. S., STEFANOPOULOU, A. G. “Extremum seeking control for soft landing of an electromechanical valve actuator”, *Automatica*, v. 40, n. 6, pp. 1063–1069, 2004.
- [17] LIU, SHU-JUN, K.-M. *Stochastic Averaging and Stochastic Extremum Seeking*. Springer-Verlag London, 2012. ISBN: 9781447140870.
- [18] KHALIL, H. K. *Nonlinear systems*, v. 3. Prentice hall Upper Saddle River, 2002.
- [19] TEEL, A., NEŠIĆ, D. “Averaging with disturbances and closeness of solutions”, *Systems & control letters*, v. 40, n. 5, pp. 317–323, 2000.
- [20] DRAKUNOV, S. V., Ö.-U. “Optimization of nonlinear system output via sliding mode approach”, pp. 61–62, 1992.
- [21] DRAKUNOV, S. “Sliding mode control of the systems with uncertain direction of control vector”. In: *Decision and Control, 1993., Proceedings of the 32nd IEEE Conference on*, pp. 2477–2478. IEEE, 1993.
- [22] OLIVEIRA, T. R., PEIXOTO, A. J., HSU, L. “Global real-time optimization by output-feedback extremum-seeking control with sliding modes”, *Journal of the Franklin Institute*,

v. 349, n. 4, pp. 1397 – 1415, 2012. ISSN: 0016-0032. doi:
<http://dx.doi.org/10.1016/j.jfranklin.2011.07.013>. Disponível em:
<<http://www.sciencedirect.com/science/article/pii/S0016003211002031>>.
Special Issue on Optimal Sliding Mode Algorithms for Dynamic Systems.

- [23] VEZHNEVETS, V., SAZONOV, V., ANDREEVA, A. “A survey on pixel-based skin color detection techniques”. In: *Proc. Graphicon*, v. 3, pp. 85–92. Moscow, Russia, 2003.