

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTO ACADÊMICO DE ELÉTRICA  
CURSO DE ENGENHARIA ELÉTRICA**

**ARTHUR NASCIMENTO ROSA LIMA**

**CONTROLE ROBUSTO DE UM MOTOR CC USANDO  
REALIMENTAÇÃO DE ESTADOS VIA PROJETO LMIS COM  
ESCALARES OTIMIZADOS POR EVOLUÇÃO DIFERENCIAL**

**TRABALHO DE CONCLUSÃO DE CURSO**

**CORNÉLIO PROCÓPIO**

**2019**

**ARTHUR NASCIMENTO ROSA LIMA**

**CONTROLE ROBUSTO DE UM MOTOR CC USANDO  
REALIMENTAÇÃO DE ESTADOS VIA PROJETO LMIS COM  
ESCALARES OTIMIZADOS POR EVOLUÇÃO DIFERENCIAL**

Proposta para Trabalho de Conclusão de Curso de graduação, apresentado à disciplina Trabalho de Conclusão de Curso 2, do curso de Engenharia Elétrica da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para a obtenção do título de Bacharel.

Orientador: Prof. Dr. Emerson Ravazzi  
Pires da Silva

Coorientador: Prof. Dr. Marcelo Favoretto  
Castoldi

**CORNÉLIO PROCÓPIO**

**2019**



**Universidade Tecnológica Federal do Paraná**  
**Campus Cornélio Procópio**  
**Departamento Acadêmico de Elétrica**  
**Curso de Engenharia Elétrica**



## **FOLHA DE APROVAÇÃO**

**Arthur Nascimento Rosa Lima**

**Controle Robusto de um motor CC usando realimentação de estados via projeto LMIs com escalares otimizados por evolução diferencial**

Trabalho de conclusão de curso apresentado às 10:30hs do dia 27/06/2019 como requisito parcial para a obtenção do título de Engenheiro Eletricista no programa de Graduação em Engenharia Elétrica da Universidade Tecnológica Federal do Paraná. O candidato foi arguido pela Banca Avaliadora composta pelos professores abaixo assinados. Após deliberação, a Banca Avaliadora considerou o trabalho aprovado.

---

Prof(a). Dr(a). Emerson Ravazzi Pires da Silva - Presidente (Orientador)

---

Prof(a). Dr(a). Marcelo Favoretto Castoldi - (Coorientador)

---

Prof(a). Dr(a). Carolina Ribeiro Rodrigues - (Membro)

---

Prof(a). Dr(a). Wagner Endo - (Membro)

A folha de aprovação assinada encontra-se na coordenação do curso.

Dedico esse trabalho à minha mãe, Rita  
de Cassia Nascimento Rosa, sem ela, eu  
nada seria.

## RESUMO

LIMA, Arthur Nascimento Rosa. **Controle robusto de um motor CC usando realimentação de estados via projeto LMIs com escalares otimizados por evolução diferencial**. 2019. 50 f. Trabalho de Conclusão de Curso – Graduação em Engenharia Elétrica. Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2019.

Quando empregadas em sistemas industriais, as máquinas de corrente contínua (CC) podem fazer o trabalho que as mãos humanas não têm a precisão ou a força física necessária. Problemas surgem, por exemplo, com o desgaste da máquina, que muitas vezes trabalha todos os dias durante anos até ser substituída, ou, inclusive, durante sua operação rotineira, pois seus parâmetros internos podem variar por fatores como a temperatura e vibrações. Porém, controladores podem ser projetados afim de obter desempenho aceitável mesmo nas condições apresentadas, são os chamados controladores robustos. Um dos métodos para a síntese desses controladores são as LMIs (do inglês, *Linear Matrix Inequalities*). Esse trabalho propõe resolver o problema que surge em projetos baseados em LMIs dependentes de escalares para a síntese de controladores robustos. Escalares esses que, normalmente, são escolhidos empiricamente ou por tentativa e erro e, normalmente, resultam em controladores ineficientes. Para fazer a otimização desses escalares será utilizado a evolução diferencial (no inglês, *Differential Evolution* – DE), que procura através de diversas combinações aleatórias encontrar soluções ótimas dentro de seus limites de busca. O algoritmo implementado demonstrou capacidade de convergir para um mínimo ótimo dentro da área de busca selecionada, com algumas limitações. Os ganhos obtidos pelo algoritmo demonstraram bom desempenho em controlar um módulo didático sob diversos testes. E, em simulações, demonstra a capacidade de estabilizar sistemas incertos como proposto no trabalho.

**Palavras-chave:** Motor CC. Realimentação de Estados. LMIs. Controle robusto. Evolução Diferencial.

## ABSTRACT

LIMA, Arthur Nascimento Rosa. **Robust control of a DC motor using state feedback through LMI Project with scalars optimized via differential evolution**. 2019. 50 f. Trabalho de Conclusão de Curso – Graduação em Engenharia Elétrica. Federal Technology University - Parana. Cornélio Procópio, 2019.

When employed in industrial systems, Direct Current (DC) machines can do work that human hands do not have the necessary physical precision or strength. Problems arise, for example, from the wear and tear of the machine, which often works every day for years until it is replaced, or even during its routine operation, since its internal parameters can vary by factors such as temperature and vibrations. However, controllers can be designed to achieve acceptable performance even under the conditions presented, the so-called robust controllers. One of the methods for the synthesis of these controllers is the Linear Matrix Inequalities (LMIs). This work proposes to solve the problem that arises in projects based on LMIs dependent on scalars for the synthesis of robust controllers. These scalars are usually chosen empirically or by trial and error and usually result in inefficient drivers. In order to optimize these scalars, we will use the Differential Evolution (DE), which searches through various random combinations to find optimal solutions within their limits of search. The employed algorithm demonstrated the ability to converge to an optimal minimum within the selected search area, with some limitations. The gains obtained by the algorithm demonstrated good performance in controlling a didactic module under several tests. And, in simulations, demonstrated the ability to stabilize uncertain systems as proposed in the paper.

**Keywords:** DC motor. State Feedback. LMI. Robust Control. Differential Evolution.

## LISTA DE FIGURAS

Figura 1 – Diagrama do sistema de controle com rastreamento.....	14
Figura 2 – Diagrama do Crossover .....	21
Figura 3 – Gráfico dos picos da função <i>fitness</i> .....	23
Figura 4 – Geração 1: população inicial e distribuição de vetores. ....	23
Figura 5 – Geração 6: A população já converge em dois mínimos. ....	24
Figura 6 – Geração 16: A população se concentra em volta do mínimo local.....	24
Figura 7 – Geração 34: A população convergiu totalmente. ....	25
Figura 8 – Módulo didático usado no trabalho.....	26
Figura 9 – Diagrama de implementação. ....	27
Figura 10 – Placa NI USB-6009 .....	28
Figura 11 – Fluxograma do algoritmo aplicado no trabalho .....	33
Figura 12 – Desenvolvimento dos valores <i>fitness</i> . ....	37
Figura 13 – Resposta ao degrau de cada controlador. ....	37
Figura 14 – Resposta da velocidade à mudança de referência. ....	39
Figura 15 – Resposta da corrente à mudança de referência. ....	39
Figura 16 – Ação de controle. ....	40
Figura 17 – Resposta da velocidade à diferentes rampas de referência.....	41
Figura 18 – Resposta da corrente à diferentes rampas de referência.....	41
Figura 19 – Ação de controle. ....	42
Figura 20 – Resposta da velocidade ao degrau.....	43
Figura 21 – Resposta da velocidade ao degrau - Instante de falha. ....	44

## LISTA DE TABELAS

Tabela 1 – Valores nominais do módulo. ....	27
Tabela 2 – Portas utilizadas na implementação. ....	28
Tabela 3 – Parametros do módulo. ....	30
Tabela 4 – Área de busca 1 – Maior escala. ....	34
Tabela 5 – Ganhos obtidos – Área de busca 1. ....	35
Tabela 6 – Área de busca 2 – Menor escala. ....	35
Tabela 7 – Ganhos obtidos – Area de busca 2. ....	36



## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>7</b>
1.1 PROBLEMA .....	8
1.2 JUSTIFICATIVA .....	8
1.3 OBJETIVOS .....	9
1.3.1 Objetivos Gerais .....	9
1.3.2 Objetivos Específicos .....	9
<b>2 FUNDAMENTAÇÃO TEÓRICA</b> .....	<b>10</b>
2.1 ESPAÇO DE ESTADOS .....	10
2.2 MÁQUINA CC DE IMÃS PERMANENTES .....	11
2.3 SISTEMAS DE CONTROLE .....	13
2.4 RASTREAMENTO DE SISTEMAS VIA REALIMENTAÇÃO DE ESTADOS .....	14
<b>3 ESTABILIDADE DE LYAPUNOV</b> .....	<b>16</b>
3.1 ESTABILIDADE QUADRÁTICA DE SISTEMAS LINEARES .....	16
3.2 ESTABILIDADE QUADRÁTICA DE SISTEMAS LINEARES INCERTOS .....	17
3.3 CONTROLE LP- $\gamma$ VIA REALIMENTAÇÃO DE ESTADOS .....	18
<b>4 EVOLUÇÃO DIFERENCIAL</b> .....	<b>20</b>
4.1 MUTAÇÃO .....	20
4.2 CROSSOVER .....	21
4.3 SELEÇÃO .....	22
4.4 DEMONSTRAÇÃO .....	22
4.5 INTEGRAL TIME ABSOLUTE ERROR (ITAE) .....	25
<b>5 MATERIAIS E MÉTODOS</b> .....	<b>26</b>
5.1 MÓDULO 2208 SERVO MECANISMO .....	27
5.2 PLACA DE AQUISIÇÃO DE DADOS NI USB-6009 .....	28
<b>6 DESENVOLVIMENTO</b> .....	<b>30</b>
6.1 IDENTIFICAÇÃO DO MÓDULO .....	30
6.2 MODELAGEM DO SISTEMA AUMENTADO E INCERTO .....	30
6.3 IMPLEMENTAÇÃO DA EVOLUÇÃO DIFERENCIAL .....	32
<b>7 RESULTADOS</b> .....	<b>34</b>
7.1 PROJETO DE CONTROLADORES .....	34
7.2 RESULTADOS EXPERIMENTAIS E DE SIMULAÇÃO .....	38
7.2.1 Teste de mudança de referência de velocidade .....	38
7.2.2 Teste de rampa de velocidade .....	40
7.2.3 Teste de falha .....	42
<b>8 CONSIDERAÇÕES FINAIS</b> .....	<b>45</b>
<b>REFERÊNCIAS</b> .....	<b>46</b>

## 1 INTRODUÇÃO

Desde a implementação dos motores elétricos, na substituição da mão de obra humana, o controle dos mesmos tornou-se substancial com a evolução das tecnologias e das técnicas de produção. É de extrema importância que as máquinas sejam precisas e eficientes no desempenho de suas tarefas. As teorias de controle moderno são utilizadas e aprimoradas para suprir as necessidades das tecnologias emergentes (OGATA, 1998).

No projeto de um sistema de controle clássico é utilizado, em alguns casos, um processo baseado em tentativa e erro para iterativamente determinar os parâmetros de um sistema aceitável. Performance aceitável é, geralmente, definida em termos de critérios no domínio do tempo e da frequência, como o tempo de subida, o tempo de estabelecimento, o ganho, a margem de fase e a largura de banda. Porém, critérios de performance radicalmente diferentes devem ser satisfeitos pelos complexos sistemas de múltiplas entradas e saídas para atender as demandas de novas e modernas tecnologias (KIRK, 2014).

Os motores de corrente contínua (CC), como todos sistemas utilizados no ambiente industrial, estão submetidos a desgastes decorrente das mais diversas fontes que podem alterar seus parâmetros de construção e assim modificar sua dinâmica de funcionamento. Os controladores convencionais, normalmente, não toleram grandes mudanças nos parâmetros originais de projeto da planta controlada. A teoria de controle robusto possibilita projetar controladores para agir mesmo na ocorrência de alteração nas características dinâmicas do sistema, são incertezas que surgem no sistema (ZHOU; DOYLE, 1999).

Para problemas envolvendo controle robusto, a síntese de controladores para a obtenção dos objetivos de desempenho dinâmico é uma tarefa mais complexa que para problemas de controle convencional. A utilização de Desigualdades Matriciais Lineares (do inglês, *Linear Matrix Inequalities* – LMI) é um método utilizado para projetar os ganhos de controle de forma menos árdua, diminuindo o problema a apenas modelar o sistema incerto e a LMI com as restrições necessárias e, então, resolver o problema de otimização para obter os ganhos do controlador. O uso de LMIs apresenta certas vantagens como a simplicidade de se tratar incertezas presentes no modelo do sistema e a facilidade de incluir índices de desempenho na abordagem do problema (BOYD et al., 1994). Além disso, LMIs podem ser resolvidas

eficientemente por meio de algumas ferramentas disponíveis na literatura de programação matemática, como, por exemplo, o software MatLab® (GAHINET et al.,1995).

No entanto, vários projetos baseados em LMIs dependem de escalares que devem ser adotados para efetivar o projeto do controlador, esses escalares geralmente são escolhidos arbitrariamente, por exemplo, em Vieira, Oliveira e Peres (2014) e Chesi (2014) os autores escolhem valores arbitrários ou, em Romão, Oliveira e Peres (2015) e Ebiharara e Hagiwara (2002) onde os autores fazem uma varredura unidimensional dos escalares. As infinitas possibilidades de escolha geram projetos diferentes e, conseqüentemente, respostas transitórias diferentes e, na maioria das vezes, respostas não desejadas. Esse trabalho irá utilizar a evolução diferencial (no inglês, *Differential Evolution* – DE), como método de otimização da seleção desses escalares de forma a obter desempenho dinâmico satisfatório.

## 1.1 PROBLEMA

O problema a ser sanado é a otimização dos escalares utilizados durante a síntese de controladores robustos para o controle de motores CC a partir de LMIs que contêm escalares que devem ser pré-definidos antes dos projetos. Normalmente esses escalares, em trabalhos encontrados na literatura especializada, são escolhidos aleatoriamente ou por tentativa e erro, que resultam em controladores factíveis, porém ineficientes, ocasionando ao sistema controlado respostas transitórias insatisfatórias. Com isso, a premissa do trabalho é utilizar o algoritmo de evolução diferencial como otimizador dos escalares utilizados para a síntese dos controladores robustos via LMIs.

## 1.2 JUSTIFICATIVA

O ambiente industrial costuma apresentar diversos fatores que podem danificar os motores, por exemplo, as variações de temperatura constantes e agentes corrosivos, além disso, normalmente, um motor é utilizado por anos até que seja

recondicionado. Esses desgastes podem causar modificação nos parâmetros que são levados em consideração no projeto dos controladores, fazendo com que o sistema perca o desempenho inicial de projeto.

Ao projetar controladores para motores, a serem instalados numa planta industrial, deve-se levar em conta o fato de que haverá desgastes nos mesmos, e o projeto pode se tornar uma tarefa complicada de ser feita empiricamente. A existência de um algoritmo que encontre um controlador ótimo pode poupar algumas horas de trabalho, e prevenir problemas com o desempenho de um motor a longo prazo. Além disso, algoritmos podem ser facilmente customizados para funcionar com outros sistemas e índices de desempenho.

### 1.3 OBJETIVOS

#### 1.3.1 Objetivos Gerais

Implementar e verificar a eficácia da evolução diferencial como método de otimização para a resolução de LMIs que contêm escalares na sintonia de controladores robustos, projetando controladores que otimizem a resposta dinâmica da máquina CC de forma robusta.

#### 1.3.2 Objetivos Específicos

- Modelagem matemática do motor CC;
- Identificação do motor CC;
- Estudo sobre otimização via evolução diferencial;
- Estudo de controladores robustos via LMIs;
- União da evolução diferencial no projeto de controladores robustos via LMIs, considerando o índice ITAE;
- Simulação em software para verificação dos controladores projetados;
- Implementação prática dos ganhos projetados.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 ESPAÇO DE ESTADOS

Segundo Ogata (1998), os sistemas de engenharia têm a tendência de tornarem-se mais complexos, com cada vez mais entradas e saídas. O controle convencional é aplicável apenas à sistemas SISO (do inglês, *Single Input, Single Output*, entrada única e saída única) enquanto os sistemas modernos geralmente são do tipo que têm diversas entradas e saídas (ou MIMO, *Multiple Input, Multiple Output*).

Uma forma conveniente de modelar um sistema MIMO é através do modelo em espaço de estados. O espaço de estados consiste em utilizar de todas as equações diferenciais que governam um sistema conhecido afim de poder obter a resposta transitória e o estado permanente do mesmo.

Considerando um sistema de  $n$  estados,  $m$  entradas e  $q$  saídas:

$$\begin{aligned}\dot{x}_1 &= a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n + b_{11}u_1 + \cdots + b_{1m}u_m \\ \dot{x}_2 &= a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n + b_{21}u_1 + \cdots + b_{2m}u_m \\ &\vdots \\ \dot{x}_n &= a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n + b_{n1}u_1 + \cdots + b_{nm}u_m\end{aligned}\tag{1}$$

$$\begin{aligned}y_1 &= c_{11}x_1 + c_{12}x_2 + \cdots + c_{1n}x_n + d_{11}u_1 + \cdots + d_{1m}u_m \\ y_2 &= c_{21}x_1 + c_{22}x_2 + \cdots + c_{2n}x_n + d_{21}u_1 + \cdots + d_{2m}u_m \\ &\vdots \\ y_q &= c_{q1}x_1 + c_{q2}x_2 + \cdots + c_{qn}x_n + d_{q1}u_1 + \cdots + d_{qm}u_m\end{aligned}\tag{2}$$

Esse sistema pode ser representado em espaço de estados da seguinte forma:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} b_{11} & \cdots & b_{1m} \\ \vdots & \ddots & \vdots \\ b_{n1} & \cdots & b_{nm} \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix}\tag{3}$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_q \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{q1} & c_{q2} & \cdots & c_{qn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} d_{11} & \cdots & d_{1m} \\ \vdots & \ddots & \vdots \\ d_{q1} & \cdots & d_{qm} \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix} \quad (4)$$

Ou simplesmente:

$$\begin{aligned} \dot{x} &= Ax(t) + Bu(t) \\ y &= Cx(t) + Du(t) \end{aligned} \quad (5)$$

sendo  $A \in \mathbb{R}^{n \times n}$  e  $B \in \mathbb{R}^{n \times m}$  matrizes que descrevem o comportamento do sistema,  $x(t) \in \mathbb{R}^n$  o vetor de estados,  $u(t) \in \mathbb{R}^m$  o vetor de entrada,  $y(t) \in \mathbb{R}^q$  o vetor de saída,  $C \in \mathbb{R}^{q \times n}$  e  $D \in \mathbb{R}^{q \times m}$  as matrizes de saída e de transmissão direta, respectivamente.

## 2.2 MÁQUINA CC DE IMÃS PERMANENTES

O motor modelado e utilizado para o trabalho é o motor CC de imãs permanentes (CCIP). Esses motores apresentam a vantagem de não necessitar de um circuito de campo externo e não contam com as perdas relacionadas ao mesmo. Porém, os imãs que compõem os polos não conseguem prover mesma intensidade de fluxo campo magnético que os enrolamentos de campo externo, além disso os imãs perdem sua intensidade com o tempo. Normalmente usados para aplicações de menor porte e potencias menores que 10 HP (do inglês, *horse power*) (CHAPMAN, 2013). É importante salientar que o trabalho foi feito em cima desse tipo de motor, mas pode ser facilmente adaptado a outros modelos de motores.

Baseando-se no modelo encontrado em Krishnan (2001), é possível traçar a relação entre as grandezas mecânicas e elétricas, sendo:

$$V = R_a i_a + L_a \dot{i}_a + E \quad (6)$$

Segundo Jr, Umans e Fitzgerald (2006):

$$E = \frac{pC_a}{2\pi m} \theta \omega_m \quad (7)$$

e

$$K_e = \frac{pC_a}{2\pi m} \theta \quad (8)$$

sendo  $V$  a tensão do terminal do motor  $[V]$ ,  $R_a$  a resistência da armadura  $[\Omega]$ ,  $i_a$  a corrente de armadura  $[A]$ ,  $\dot{i}_a$  a taxa de variação da corrente  $[A/s]$ ,  $L_a$  a indutância de armadura  $[H]$ ,  $E$  a tensão induzida no rotor  $[V]$ ,  $p$  o número de polos do motor,  $C_a$  o número de condutores no enrolamento de armadura,  $m$  o número de caminhos em paralelo no enrolamento,  $\theta$  o fluxo magnético constante proveniente dos ímãs  $[Wb]$ ,  $\omega_m$  a velocidade angular do rotor  $[rad/s]$  e  $K_e$  a constante de torque elétrico do motor  $[Wb/rad]$ .

Pode-se simplificar (6) usando (7) e (8), como:

$$\dot{i}_a = \frac{V - R_a i_a - K_e \omega_m}{L_a} \quad (9)$$

A partir das propriedades fundamentais de movimento angular é possível relacionar os parâmetros do rotor:

$$T = \tau_m - \tau_a - \tau_l \quad (10)$$

sendo:

$$\tau_m = K_e i_a \quad (11)$$

$$\tau_a = B \omega_m \quad (12)$$

$$\tau_l = I \dot{\omega}_m \quad (13)$$

sendo  $\tau_m$  o torque mecânico do motor  $[Nm]$ ,  $\tau_a$  o torque de atrito com o ar  $[Nm]$ ,  $\tau_l$  o torque de momento de inércia  $[Nm]$ ,  $T$  o torque de carga  $[Nm]$ ,  $B$  o coeficiente de atrito com o ar  $[Nms]$ ,  $I$  o momento de inercia do rotor  $[kgm^2]$  e  $\dot{\omega}_m$  a aceleração angular do eixo  $[rad/s^2]$ .

Utilizando (11), (12) e (13) podemos simplificar para a Equação (10) como:

$$\dot{\omega}_m = \frac{K_e i_a - B\omega_m - T}{I} \quad (14)$$

Assim, podemos escrever o modelo matemático no espaço de estados como:

$$\begin{bmatrix} \dot{i}_a \\ \dot{\omega}_m \end{bmatrix} = \begin{bmatrix} \frac{-R_a}{L_a} & \frac{-K_e}{L_a} \\ \frac{K_e}{I} & \frac{-B}{I} \end{bmatrix} \begin{bmatrix} i_a \\ \omega_m \end{bmatrix} + \begin{bmatrix} \frac{1}{L_a} & 0 \\ 0 & -\frac{1}{I} \end{bmatrix} \begin{bmatrix} V \\ T \end{bmatrix} \quad (15)$$

$$\begin{bmatrix} i_a \\ \omega_m \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} i_a \\ \omega_m \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V \\ T \end{bmatrix}$$

## 2.3 SISTEMAS DE CONTROLE

Sistemas de controle são utilizados para obter comportamentos específicos em certos sistemas, por exemplo que os estados sigam uma referência, ou que o sistema obtenha alguns índices de desempenho transitório, como coeficiente de amortecimento, ou desempenho físico, como consumo de energia. No caso desse trabalho, o controlador robusto será utilizado para fazer com que o sistema (motor CC) obtenha os seguintes índices de desempenho: baixo tempo de acomodação, baixo *overshoot*, erro de regime permanente nulo e robustez quanto a variação de parâmetros.

Para obter esses índices, diversas técnicas de controle podem ser empregadas, como o PID (controlador proporcional, integrativo e derivativo), o LQR

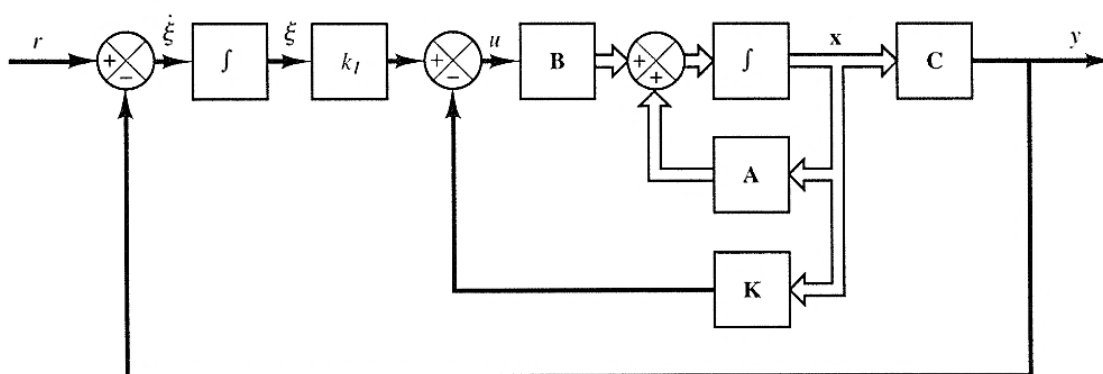


(do inglês, *Linear Quadratic Regulator*) (BETETO,2016), ou, como nesse trabalho, o projeto robusto baseado em LMIs.

## 2.4 RASTREAMENTO DE SISTEMAS VIA REALIMENTAÇÃO DE ESTADOS

Baseando-se em Ogata (1998), considera-se um sistema controlável, linear e invariante no tempo descrito pela Equação (5). Insere-se um controlador integrador com rastreamento, como na Figura 1:

**Figura 1 – Diagrama do sistema de controle com rastreamento.**



Fonte: Ogata (1998)

Assim, temos:

$$u = -Kx + k_I \xi \quad (16)$$

$$\dot{\xi} = r - y \quad (17)$$

sendo  $u$  o sinal de controle,  $K \in \mathbb{R}^{m \times n}$  a matriz de ganhos de realimentação de estados,  $k_I \in \mathbb{R}^{m \times q}$  a matriz de ganhos da saída do integrador,  $\xi$  o vetor de erros do sistema,  $\xi$  a integral do vetor de erros (estados adicionais do sistema), e  $r$  a referência desejada.

Podemos escrever a nova dinâmica do sistema através do espaço de estados aumentado de malha fechada:

$$\begin{bmatrix} \dot{x} \\ \dot{\xi} \end{bmatrix} = \begin{bmatrix} A - BK & Bk_I \\ -C & 0 \end{bmatrix} \begin{bmatrix} x \\ \xi \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r \quad (18)$$

$$y = [C \quad 0] \begin{bmatrix} x \\ \xi \end{bmatrix} \quad (19)$$

Podemos decompor (18) em:

$$\begin{bmatrix} \dot{x} \\ \dot{\xi} \end{bmatrix} = \left\{ \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} \begin{bmatrix} -K & k_1 \end{bmatrix} \right\} \begin{bmatrix} x \\ \xi \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r \quad (20)$$

Assim, determinamos as matrizes aumentadas do sistema,  $\hat{A}$  e  $\hat{B}$ .

$$\hat{A} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \quad (21)$$

$$\hat{B} = \begin{bmatrix} B \\ 0 \end{bmatrix} \quad (22)$$

O projeto das matrizes de realimentação,  $K$  e  $k_1$ , de forma apropriada faz com que a planta seja controlada satisfatoriamente. Neste trabalho, o objetivo é obter  $K$  e  $k_1$  através do projeto robusto via LMIs com escalares otimizados por meio da evolução diferencial.

### 3 ESTABILIDADE DE LYAPUNOV

#### 3.1 ESTABILIDADE QUADRÁTICA DE SISTEMAS LINEARES

Segundo Boyd et al. (1994 apud BETETO, 2016), para o sistema:

$$\dot{x}(t) = Ax(t) \quad (23)$$

Considera-se uma candidata a função de Lyapunov:

$$V(x(t)) = x(t)^T Px(t) > 0 \quad (24)$$

Com  $\dot{V}(x(t)) < 0$  para todo  $x(t) \neq 0$ , onde

$$\dot{V}(x(t)) = \dot{x}(t)^T Px(t) + x(t)^T P\dot{x}(t) \quad (25)$$

Substituindo o sistema (23) em (25), obtêm-se:

$$\dot{V}(x(t)) = Ax(t)^T Px(t) + x(t)^T PAx(t) < 0 \quad (26)$$

$$\dot{V}(x(t)) = x(t)^T (A^T P + PA)x(t) < 0 \quad (27)$$

Logo, uma condição necessária e suficiente para que (23) seja estável é existir uma matriz  $P$ , simétrica e positiva definida, tal que:

$$\begin{aligned} A^T P + PA &< 0 \\ P &> 0 \end{aligned} \quad (28)$$

### 3.2 ESTABILIDADE QUADRÁTICA DE SISTEMAS LINEARES INCERTOS

Segundo Boyd et al.(1994 apud BUZACHERO, 2010), considera-se que  $A$  não seja precisamente conhecida, mas pertencente a um politopo de incertezas  $\Lambda$ , o domínio de incertezas pode ser escrita como a combinação convexa dos vértices  $A_j$ ,  $j = 1, \dots, N$ , assim:

$$\Lambda = \{A(\alpha) \in \mathbb{R}^{n \times n} : A(\alpha) = \sum_{j=1}^N \alpha_j A_j, \sum_{j=1}^N \alpha_j = 1, \alpha_j \geq 0, j = 1, \dots, N\} \quad (29)$$

A condição para estabilidade do politopo  $\Lambda$  é dada pela da matriz  $P$ , simétrica e positiva definida, tal que para todo  $A(\alpha) \in \Lambda$  :

$$A(\alpha)^T P + PA(\alpha) < 0 \quad (30)$$

Segundo Boyd et al. (1994 apud BUZACHERO, 2010), esta condição é conhecida como estabilidade quadrática. Pode-se verificar pela convexidade da desigualdade de Lyapunov, substituindo  $A(\alpha)$  por  $\sum_{j=1}^N \alpha_j A_j$  e colocando o somatório em evidencia:

$$\sum_{j=1}^N \alpha_j (A_j^T P + PA_j) < 0 \quad (31)$$

Para a inequação ser verdadeira, logo, verifica-se a estabilidade quadrática

$$A_j^T P + PA_j < 0 \quad (32)$$

com  $j = 1, \dots, N$ .

A estabilidade quadrática garante a estabilidade do sistema, porém apresenta resultados conservadores, contudo, vem sendo muito utilizada para a síntese de controladores robustos (LEITE et al., 2004 apud BUZACHERO, 2010).

### 3.3 CONTROLE LP- $\gamma$ VIA REALIMENTAÇÃO DE ESTADOS

Neste trabalho, o projeto de controladores robusto via LMIs a ser otimizado foi derivado em Vieira, Oliveira e Peres (2014), e é exposto abaixo na forma de Lema.

**Lema 1 (LP- $\gamma$ ).** Existe um ganho  $K$  tal que  $A(\alpha) - B(\alpha)K$  é robustamente estável em  $\dot{x} = A(\alpha)x + B(\alpha)u$  se existirem uma matriz simétrica definida positiva  $P(\alpha) \in \mathbb{R}^{n \times n}$ , matrizes  $F(\alpha)$ ,  $G(\alpha) \in \mathbb{R}^{n \times n}$ ,  $L \in \mathbb{R}^{m \times n}$  e  $S \in \mathbb{R}^{n \times n}$ , e escalares  $\gamma_1, \gamma_2, \gamma_3$ ,  $\epsilon, \zeta$  e  $\rho$  tais que:

$$\begin{aligned} & \begin{bmatrix} A(\alpha)F(\alpha) + \zeta B(\alpha)L + F^T(\alpha)A^T(\alpha) + \zeta L^T B^T(\alpha) + \gamma_1(\zeta S - \rho F(\alpha)) + \gamma_1(\zeta S - \rho F(\alpha))^T \\ G^T(\alpha)A^T(\alpha) + \epsilon L^T B^T(\alpha) - F(\alpha) + \gamma_1(\epsilon S - \rho G(\alpha))^T + \gamma_2(\zeta S - \rho F(\alpha)) \\ L^T B^T(\alpha) + \gamma_1 S^T + \gamma_3(\zeta S - \rho F(\alpha)) \end{bmatrix} \\ & \quad * \\ & \quad -G(\alpha) - G^T(\alpha) + \gamma_2(\epsilon S - \rho G(\alpha)) + \gamma_2(\epsilon S - \rho G(\alpha))^T \\ & \quad \gamma_2 S^T + \gamma_3(\epsilon S - \rho G(\alpha)) \\ & \quad \left. \begin{array}{c} * \\ * \\ \gamma_3(S - S^T) \end{array} \right] + \Psi < 0 \end{aligned} \quad (33)$$

para todo  $\alpha \in \Lambda$ , com:

$$\Psi = \Psi_c = \text{diag} \left( \begin{bmatrix} 0 & P(\alpha) \\ P(\alpha) & 0 \end{bmatrix}, 0 \right) \quad (34)$$

para sistemas contínuos no tempo. No caso afirmativo, o ganho robusto é dado por:

$$K^* = -\rho L S^{-1} \quad (35)$$

**Demonstração:** Consulte Vieira, Oliveira e Peres (2014).

Neste caso, a função de Lyapunov,  $P(\alpha)$ , é dependente de  $\alpha \in \Lambda$ , permitindo resultados menos conservadores.

Normalmente, em trabalhos envolvendo LMIs com escalares para a sintonia, como  $\gamma_1, \gamma_2, \gamma_3, \epsilon, \zeta$  e  $\rho$  neste caso, os mesmos costumam ser obtidos por técnicas empíricas, isto é, de tentativa e erro. Assim, neste trabalho é proposto a otimização e a obtenção desses escalares via evolução diferencial, com o objetivo de projetar controladores mais eficientes.

## 4 EVOLUÇÃO DIFERENCIAL

Será utilizado como método de otimização dos escalares usados na síntese dos controladores robustos via LMIs. O algoritmo será aplicado com base no algoritmo apresentado em Bosco (2016).

Segundo Price, Storn e Lampinen (2006), o algoritmo funciona criando uma população  $P_{x,g}$  de  $Np$  número de indivíduos na população, vetores  $x_{i,g}$  de dimensão  $D$  número de parâmetros, ou cromossomos, que se deseja otimizar, assim:

$$P_{x,g} = \begin{pmatrix} x_{1,g} \\ \vdots \\ x_{i,g} \\ \vdots \\ x_{Np,g} \end{pmatrix} \quad (36)$$

$$x_{i,g} = (x_{i,1,g} \quad \dots \quad x_{i,j,g} \quad \dots \quad x_{i,D,g}) \quad (37)$$

sendo  $g$  o índice que demonstra a geração em que se encontra o algoritmo,  $i$  o índice dos indivíduos e  $j$  o índice dos cromossomos.

Quando o algoritmo é inicializado, os vetores da população inicial,  $P_{x,1}$ , recebem valores aleatórios, podendo ser estabelecida agora uma área de busca limitada desejada.

### 4.1 MUTAÇÃO

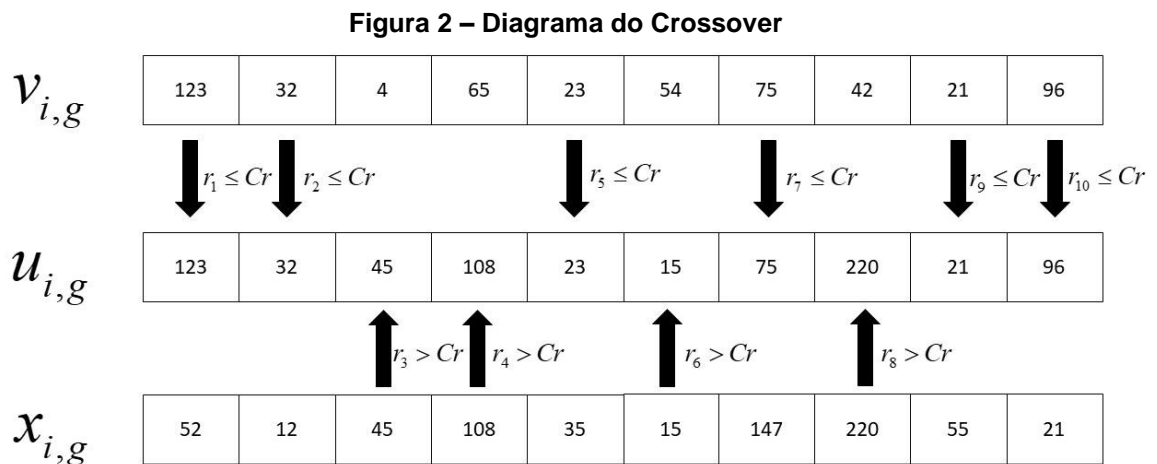
Durante a mutação, uma nova população  $P_{v,g}$ , chamada doadora, com indivíduos  $v_{i,g}$ , é gerada a partir dos indivíduos originais da seguinte forma:

$$v_{i,g} = x_{i_1,g} + F(x_{i_2,g} - x_{i_3,g}) \quad (38)$$

sendo  $r_1$ ,  $r_2$  e  $r_3$  números inteiros aleatórios de 1 a  $Np$  e  $F \in [0, \infty]$  um valor escalar que controla a taxa de evolução da população e, normalmente, tem melhor efeito quando menor que 1. Além disso, esse valor pode ser considerado aleatório para cada uso, aleatorizando esse fator é possível prevenir a estagnação do algoritmo e aumentar a diversificação dos vetores nessa etapa (PRICE; STORN; LAMPINEN, 2006).

## 4.2 CROSSOVER

Durante a etapa de *Crossover*, uma nova população,  $P_{u,g}$ , é criada com indivíduos  $u_{i,g}$ , que são compostos por cromossomos escolhidos aleatoriamente das duas populações criadas anteriormente, como demonstrado na Figura 2 e na Equação (39):



Fonte: Adaptado de Price et al. (2006).

$$u_{i,j,g} = \begin{cases} v_{i,j,g} & \text{se } r_j \leq Cr \\ x_{i,j,g} & \text{se } r_j > Cr \end{cases} \quad (39)$$

sendo  $r_j$  um número aleatório para cada cromossomo e  $Cr \in [0, 1]$  um parâmetro que controla a probabilidade do novo indivíduo receber o cromossomo do vetor doador ou do original.



### 4.3 SELEÇÃO

Durante a última etapa, a de seleção, utiliza-se a função *fitness* para escolher os indivíduos entre as populações  $P_{u,g}$  e  $P_{x,g}$ . Os que obterem o menor valor *fitness* devem ser mantidos e são levados à próxima geração  $P_{x,g+1}$ , o resto é descartado. Assim:

$$x_{i,g+1} = \begin{cases} u_{i,g} & \text{se } f(u_{i,g}) < f(x_{i,g}) \\ x_{i,g} & \text{se } f(u_{i,g}) > f(x_{i,g}) \end{cases} \quad (40)$$

sendo  $f$  a função *fitness* utilizada como critério de parada e será comentada na próxima seção. Dessa forma, se o indivíduo  $u_{i,g}$  obtiver um valor *fitness* menor ele será colocado em  $P_{x,g+1}$ , caso contrário,  $x_{i,g}$  será copiado em seu lugar. Dessa forma, o algoritmo se repete até que alcance um ótimo valor *fitness* ou o número de gerações pré-estabelecido.

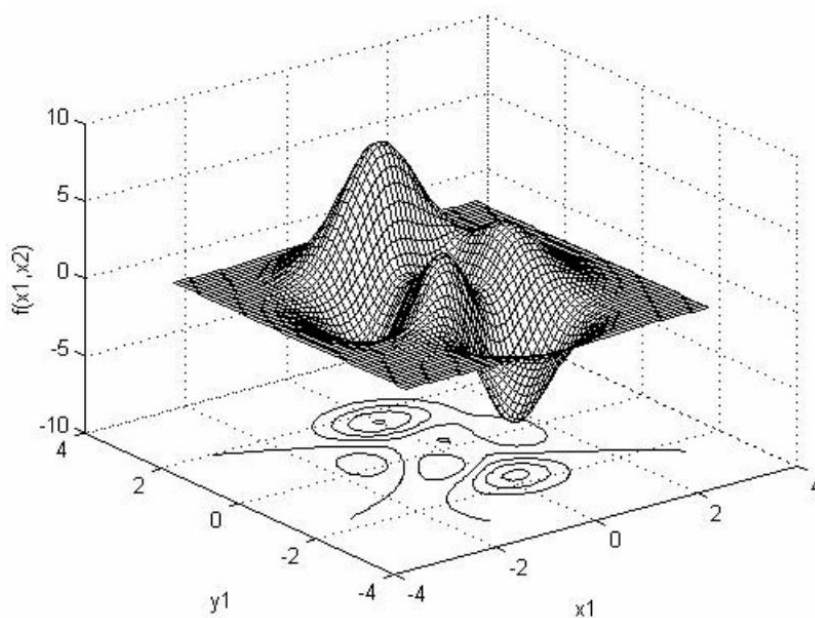
### 4.4 DEMONSTRAÇÃO

O autor Price, Storn e Lampinen (2006) faz um teste no Capítulo 2 para demonstrar a eficácia do algoritmo. Para o teste foi usado a função *fitness* abaixo:

$$f(x_1, x_2) = 3(1-x_1)^2 \cdot \exp(x_1^2 + (x_2+1)^2) - 10 \left( \frac{x_1}{5} - x_1^3 - x_1^5 \right) \cdot \exp(x_1^2 + x_2^2) - \frac{1}{3} \cdot \exp((x_1+1)^2 + x_2^2) \quad (41)$$

O gráfico dos picos dessa função pode ser visto na Figura 3:

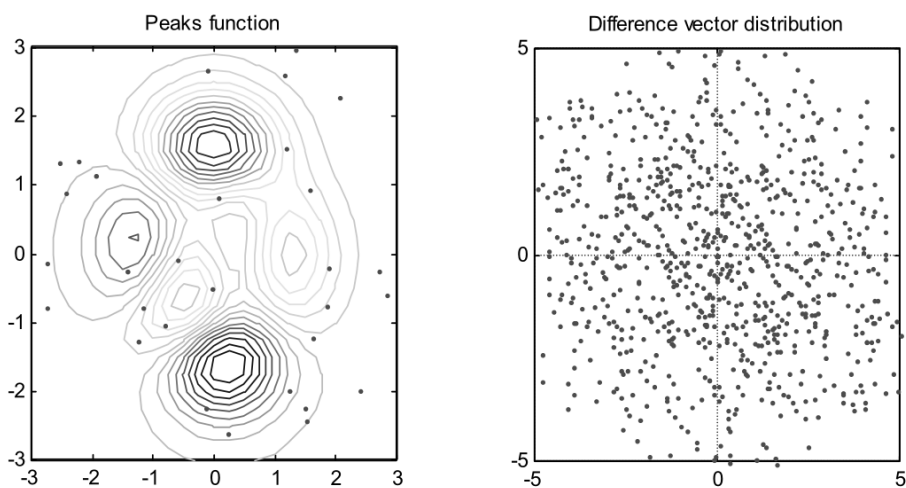
**Figura 3 – Gráfico dos picos da função *fitness*.**



Fonte: Price, Storn e Lampinen (2006)

O progresso do algoritmo pode ser visualizado nas Figuras 4, 5, 6 e 7:

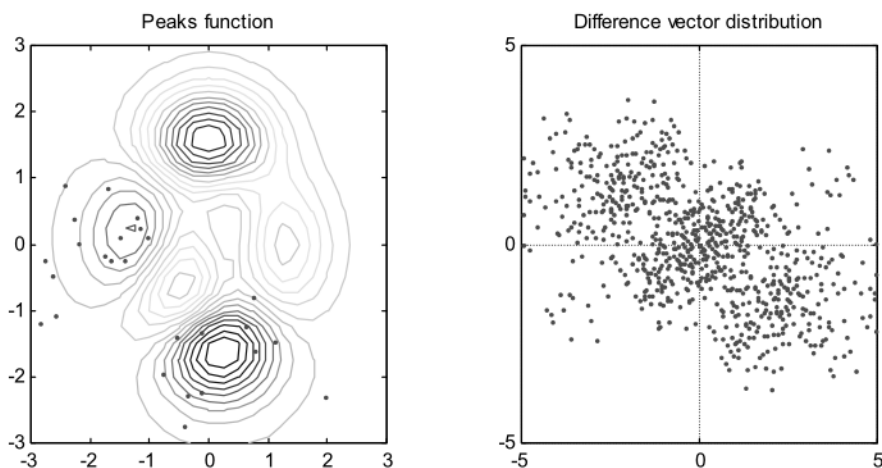
**Figura 4 – Geração 1: população inicial e distribuição de vetores.**



Fonte: Price, Storn e Lampinen, 2006.

Na Figura 4, é possível ver como na primeira geração o algoritmo inicia a população em locais aleatórios. Estão demonstrados a função e os indivíduos nos limites de busca, no primeiro gráfico, e a diferença entre cada um no segundo gráfico.

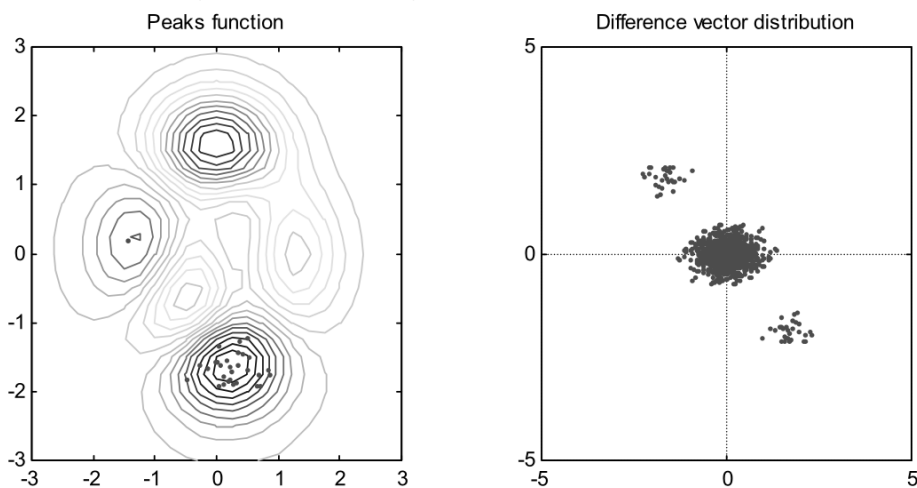
**Figura 5 – Geração 6: A população já converge em dois mínimos.**



Fonte: Price, Storn e Lampinen, 2006.

Na geração 6, já se percebe a população aglomerando-se em volta dos dois mínimos locais, vide Figura 5.

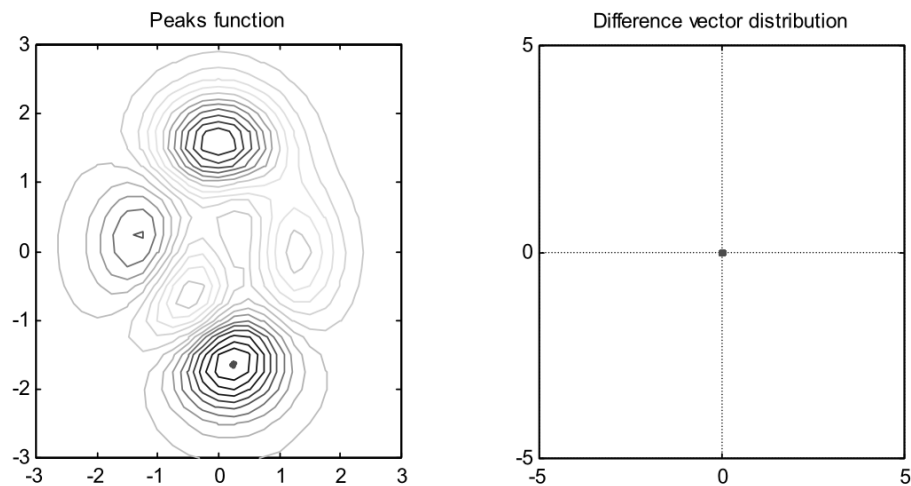
**Figura 6 – Geração 16: A população se concentra em volta do mínimo local.**



Fonte: Price, Storn e Lampinen, 2006.

Na 16ª geração, os indivíduos agora estão centralizados nos dois mínimos locais. A maioria já se encontra no mínimo mais expressivo, vide Figura 6.

**Figura 7 – Geração 34: A população convergiu totalmente.**



Fonte: Price, Storn e Lampinen, 2006.

Na geração 34, vide Figura 7, toda população convergiu ao mínimo da função, na área de busca, e o algoritmo termina.

#### 4.5 INTEGRAL TIME ABSOLUTE ERROR (ITAE)

A função *fitness* foi escolhida com base no algoritmo de Bosco (2016). A função pondera o erro absoluto em função do tempo em que o mesmo ocorreu e, por isso, erros de grande duração deverão expressar maiores valores *fitness*, ou seja, controladores com performance pior. Esse erro é dado por:

$$f = \int_0^{t_0} t |e(t)| dt \quad (42)$$

onde  $t_0$  é o tempo final de simulação.

## 5 MATERIAIS E MÉTODOS

Para a realização desse trabalho, foi utilizado como base para o algoritmo e a validação dos dados obtidos em simulação o Módulo 2208 – Servomecanismo da Datapool. Esse módulo pode ser visto na Figura 8:

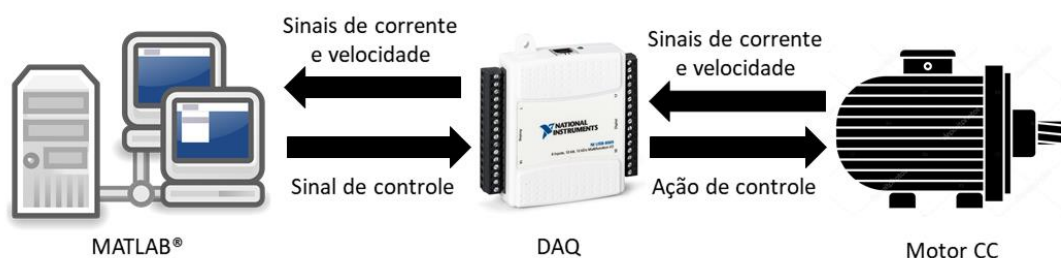
**Figura 8 – Módulo didático usado no trabalho**



**Fonte: Datapool, [201-].**

O módulo conta com um motor CCIP e sensores de corrente e de velocidade integrados, sendo necessário apenas fazer a captação dos sinais desses sensores, o processamento, e a atuação de controle. Para fazer a captação e atuação foi empregada uma placa DAQ (do inglês, *Data acquisition*, aquisição de dados). A placa DAQ faz a interface entre o módulo e o software MATLAB®, que faz o controle do mesmo. Um diagrama de implementação do controle pode ser visto na Figura 9:

**Figura 9 – Diagrama de implementação.**



Fonte: Autoria própria.

## 5.1 MÓDULO 2208 SERVO MECANISMO

O motor utilizado no trabalho, apresentado na Figura 8, tem seus valores nominais descritos na Tabela 1.

**Tabela 1 – Valores nominais do módulo.**

Parâmetros	Valores [Unidades]
Tensão nominal de armadura	12 [V]
Corrente nominal de armadura	1,5 [A]
Tensão nominal de comando	5 [V]
Corrente nominal de comando	20 [mA]
Velocidade nominal	7,2 [V]

Infelizmente, a documentação do módulo não descreve a velocidade nominal nem a taxa de  $\left[\frac{rad/s}{V}\right]$  do transdutor de velocidade. Por isso, não é possível obter o valor de velocidade nominal demonstrada na Tabela 1 em unidades de  $[rad/s]$  ou  $[rpm]$ . Além disso, o valor nominal de velocidade foi obtido de forma empírica. A tensão e corrente nominais de comando, indicados na Tabela 1, se referem a tensão e corrente que o driver integrado no módulo suportam, esses sinais são amplificados para a tensão e corrente de armadura que vão alimentar o motor.

As portas disponíveis no módulo a serem utilizadas durante a implementação são:

**Tabela 2 – Portas utilizadas na implementação.**

<b>Porta</b>	<b>Descrição</b>
+VU	Tensão de comando
-VU	Referência da tensão de comando
+VN	Sinal de tensão do transdutor de velocidade
JST	Sinal de tensão do transdutor de corrente

**Fonte: Autoria própria.**

## 5.2 PLACA DE AQUISIÇÃO DE DADOS NI USB-6009

A placa utilizada para fazer a aquisição de dados e atuação do motor está ilustrada na Figura 10.

**Figura 10 – Placa NI USB-6009**



**Fonte: National Instruments, [201-].**

Essa placa conta com oito entradas analógicas, cada entrada é capaz de fazer amostras de 14 bits em até 6 kS/s. Porém, essa placa não conta com um *clock* interno, isto é, a amostragem depende de um gatilho externo. Por isso, a frequência de

amostragem pode variar com a eficiência do algoritmo e do hardware utilizado para acionar a placa. Os sinais amostrados podem ter uma diferença de potencial de  $\pm 20$  [V], porém os mesmos não podem ter mais de  $\pm 10$  [V] de diferença de potencial com o pino "GND". Dessas entradas, duas serão utilizadas para ler os sinais das portas +VN e JST.

A placa conta também com duas saídas analógicas com resolução de 12 bits e taxa de atualização de até 150 Hz, novamente acionada externamente. As saídas tem o alcance de 0 a +5 V. Dessas saídas, uma será utilizada como sinal de comando.

A placa também tem portas de I/O digitais, que não serão utilizadas no trabalho.



## 6 DESENVOLVIMENTO

### 6.1 IDENTIFICAÇÃO DO MÓDULO

A identificação foi feita excitando o motor com um sinal do tipo PRBS (do inglês, pseudorandom binary signal) através da entrada de tensão de comando. A resposta de corrente e velocidade foram registradas. Com os dados, de entrada e de resposta, utilizou-se a *System Identification Toolbox*, uma biblioteca de identificação de sistemas disponível no software MATLAB™, para estimar os parâmetros do modelo do sistema, de forma que para a mesma entrada, se obtenha uma resposta aproximada em simulação. Os parâmetros obtidos após o processo de identificação estão inseridos na Tabela 3.

**Tabela 3 – Parametros do módulo.**

Parâmetros	Valores [Unidades]
Resistencia de armadura ( $R_a$ )	3,62 [ $\Omega$ ]
Indutância de armadura ( $L_a$ )	61,54 [ $mH$ ]
Constante de torque elétrico ( $K_e$ )	0,477 [ $Wb / rad$ ]
Coeficiente de atrito ( $B$ )	$2,46 \times 10^{-3}$ [ $Nms$ ]
Momento de inercia ( $I$ )	$18,575 \times 10^{-3}$ [ $kgm^2$ ]

**Fonte: Autoria própria.**

### 6.2 MODELAGEM DO SISTEMA AUMENTADO E INCERTO

Com os parâmetros do módulo, é possível fazer a modelagem do sistema para iniciar o processo de sintonia dos ganhos do controlador robusto. Substituindo os valores da Tabela 3, e considerando  $T = 0$ , na equação (15), chega-se em:

$$\begin{bmatrix} \dot{i}_a \\ \dot{\omega}_m \end{bmatrix} = \begin{bmatrix} -58,8222 & -7,7546 \\ 25,6933 & -0,1325 \end{bmatrix} \begin{bmatrix} i_a \\ \omega_m \end{bmatrix} + \begin{bmatrix} 16,2485 \\ 0 \end{bmatrix} V \quad (43)$$

$$y = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} i_a \\ \omega_m \end{bmatrix} \quad (44)$$

É possível, a partir de (16), verificar que os ganhos de controle terão a seguinte forma:

$$\begin{aligned} K^* &= [-K \quad k_I] \\ K &= [K_{i_a} \quad K_{\omega_m}] \end{aligned} \quad (45)$$

E, substituindo (43) em (21) e (22) obtém-se:

$$\hat{A}_1 = \begin{bmatrix} -58,8222 & -7,7546 & 0 \\ 25,6933 & -0,1325 & 0 \\ 0 & -1 & 0 \end{bmatrix} \quad (46)$$

$$\hat{B}_1 = \begin{bmatrix} 16,2485 \\ 0 \\ 0 \end{bmatrix} \quad (47)$$

Para reproduzir uma situação onde os parâmetros da máquina são incertos, foram assumidos, para fins de projeto, 4 motores. O primeiro, utilizado como base, é o representado pelos modelos (46) e (47). O segundo modelo apresenta uma queda de 10% na resistência de armadura e suas matrizes aumentadas estão exibidas nas equações (48) e (49). O terceiro modelo apresenta uma queda de 10% na indutância de armadura e suas matrizes aumentadas estão exibidas nas equações (50) e (51). O quarto modelo apresenta uma queda de -10% na resistência e na indutância de armadura e suas matrizes aumentadas estão (52) e (53)

$$\hat{A}_2 = \begin{bmatrix} -52,94 & -7,7546 & 0 \\ 25,6933 & -0,1325 & 0 \\ 0 & -1 & 0 \end{bmatrix} \quad (48)$$

$$\hat{B}_2 = \begin{bmatrix} 16,2485 \\ 0 \\ 0 \end{bmatrix} \quad (49)$$

$$\hat{A}_3 = \begin{bmatrix} -65,3581 & -7,7546 & 0 \\ 25,6933 & -0,1325 & 0 \\ 0 & -1 & 0 \end{bmatrix} \quad (50)$$

$$\hat{B}_3 = \begin{bmatrix} 18,0539 \\ 0 \\ 0 \end{bmatrix} \quad (51)$$

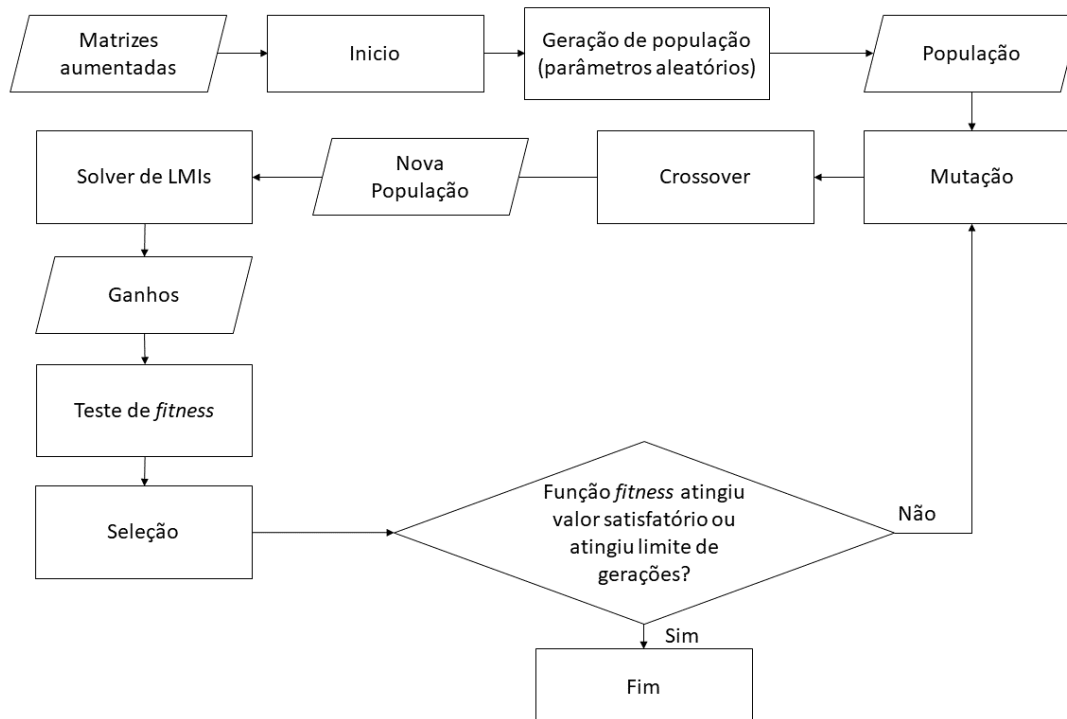
$$\hat{A}_4 = \begin{bmatrix} -58,8222 & -8,6163 & 0 \\ 25,6933 & -0,1325 & 0 \\ 0 & -1 & 0 \end{bmatrix} \quad (52)$$

$$\hat{B}_4 = \begin{bmatrix} 16,2485 \\ 0 \\ 0 \end{bmatrix} \quad (53)$$

### 6.3 IMPLEMENTAÇÃO DA EVOLUÇÃO DIFERENCIAL

A partir da fundamentação teórica, o algoritmo é alimentado com as matrizes aumentadas de cada vértice do polítopo e retorna como resposta valores de ganhos que podem ser utilizados para otimizar a reposta de malha fechada do motor CCIP, seja o motor qualquer combinação linear dos modelos considerados na seção anterior fazendo com que o mesmo siga uma referência de velocidade, desde que a referência seja limitada aos limites físicos do motor. A forma com que o *solver* se integra ao algoritmo DE para retornar os ganhos robustos otimizados está exibida no fluxograma na Figura 11.

Figura 11 – Fluxograma do algoritmo aplicado no trabalho



Fonte: Autoria própria.

## 7 RESULTADOS

### 7.1 PROJETO DE CONTROLADORES

Inicialmente, constatou-se que o algoritmo estava projetando ganhos com valores elevados, que em simulação funciona perfeitamente, mas em bancada apresenta oscilações elevadas. Para corrigir isso, os ganhos dos controladores foram limitados para valores menores que 1.

O algoritmo aplicado permite configurar como um parâmetro o limite da área de busca de cada escalar a ser otimizado. Para efeito de comparação, foram projetados dois controladores utilizando o algoritmo em duas áreas de buscas distintas. Além disso, dois outros controladores foram projetados com escalares aleatórios, cada um dentro de uma área de busca.

A primeira área de busca, de maior escala, está exposta na Tabela 4. Os escalares otimizados e os ganhos robustos projetados nessa configuração estão dispostos na Tabela 5.

**Tabela 4 – Área de busca 1 – Maior escala.**

Escalares	Limites
$\gamma_1, \gamma_2, \epsilon, \zeta$ e $\rho$	[1e-6; 5000]
$\gamma_3$	[-5000; -1e-6]

**Fonte: Autoria própria.**

**Tabela 5 – Ganhos obtidos – Área de busca 1.**

Controlador otimizado		Controlador aleatório	
Escalares/Ganhos	Valores	Escalares/Ganhos	Valores
$\gamma_1$	4743,5648	$\gamma_1$	2956,7637
$\gamma_2$	336,0156	$\gamma_2$	5000
$\gamma_3$	-56,4596	$\gamma_3$	-550,4019
$\epsilon$	1e-06	$\epsilon$	1445,4357
$\zeta$	1673,2501	$\zeta$	1473,9398
$\rho$	1e-06	$\rho$	1e-06
$K_1$	-0,0110	$K_1$	0,0846
$K_2$	-0,0033	$K_2$	0,0507
$k_1$	2.6425	$k_1$	4.002

Fonte: Autoria própria.

A segunda área de busca está exposta na Tabela 6. Os ganhos robustos e os escalares que foram otimizados para o projeto estão na Tabela 7.

**Tabela 6 – Área de busca 2 – Menor escala.**

Escalares	Limites
$\gamma_1, \gamma_2, \epsilon, \zeta$ e $\rho$	[1e-6; 10]
$\gamma_3$	[-10; -1e-6]

Fonte: Autoria própria.

**Tabela 7 – Ganhos obtidos – Area de busca 2.**

<b>Controlador otimizado</b>		<b>Controlador aleatório</b>	
Escalares/Ganhos	Valores	Escalares/Ganhos	Valores
$\gamma_1$	0,8301	$\gamma_1$	6,4005
$\gamma_2$	1e-06	$\gamma_2$	2,3040
$\gamma_3$	-1e-06	$\gamma_3$	-4,1360
$\epsilon$	2,7054	$\epsilon$	0,5288
$\zeta$	1e-06	$\zeta$	2,3117
$\rho$	3,9531	$\rho$	0,9852
$K_1$	-0,4508	$K_1$	-0,0211
$K_2$	-0,1706	$K_2$	0,1058
$k_1$	5.5936	$k_1$	0,5112

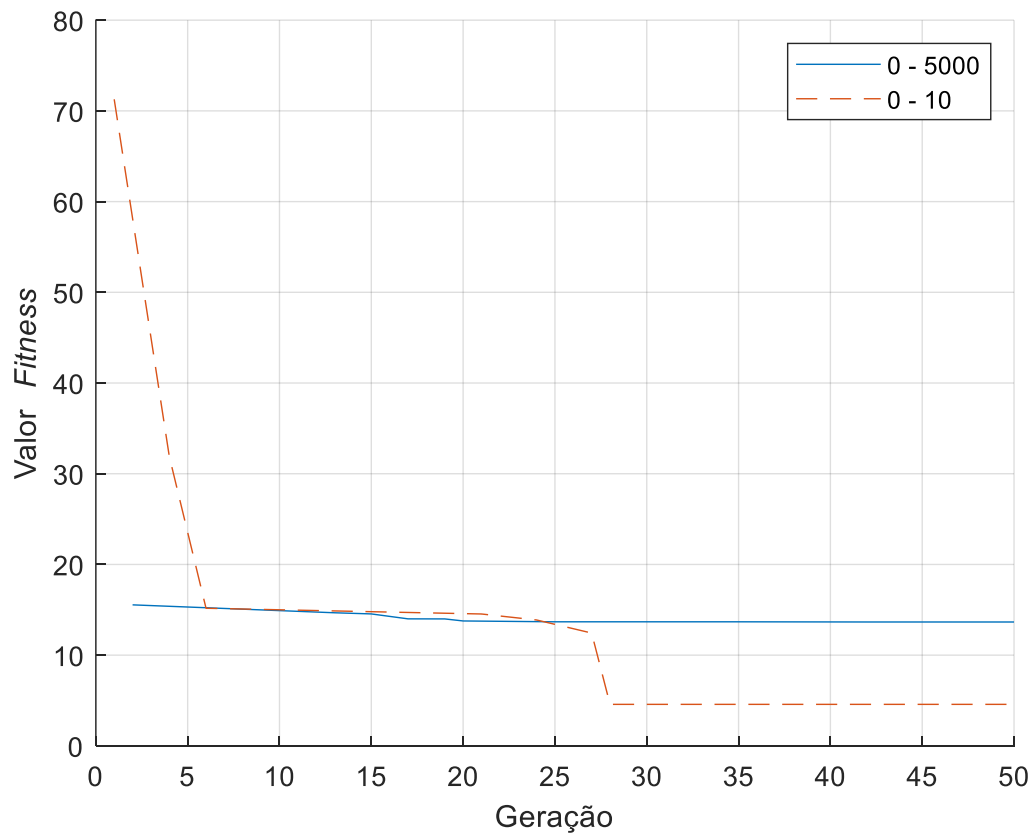
**Fonte: Autoria própria.**

A Figura 12 apresenta a redução da função *fitness* durante a execução do algoritmo para cada área de busca. Percebe-se que quando o algoritmo foi executado considerando a área 1, o mesmo começou com uma percepção muito boa e houve pouca melhora no decorrer da execução. Quando executado na área 2, aconteceu o oposto, o algoritmo iniciou com uma percepção ruim e conseguiu convergir satisfatoriamente.

A Figura 13 mostra a resposta ao degrau de cada controlador robusto projetado, os dados demonstrados corroboram com os dados exibidos na Figura 12. É possível ver porque o controlador ótimo da área 2 é melhor que o controlador ótimo da área 1, seu *overshoot*, sua oscilação e o seu tempo de acomodação são menores. O controlador aleatório da área 1 é regular, porém, a resposta demora muito mais para se acomodar e apresenta elevado *overshoot*. O controlador aleatório da área 2 é insatisfatório, não conseguindo acomodar no tempo de simulação estipulado.

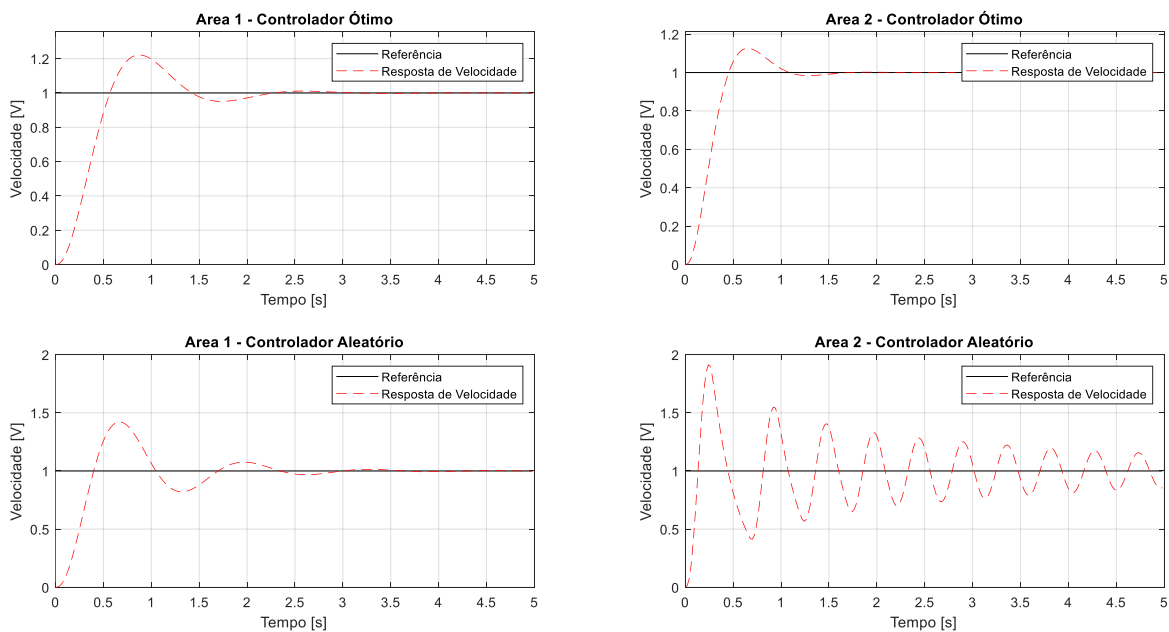
Observe que a LMI (33) do Lema 1 garante, se factível, apenas a estabilidade do sistema, no caso o motor CC. Contudo, a flexibilidade de projeto da LMI, por meio das escolhas adequadas dos escalares, permite obter controladores capazes de gerar respostas transitórias desejadas.

**Figura 12 – Desenvolvimento dos valores *fitness*.**



Fonte: Autoria própria.

**Figura 13 – Resposta ao degrau de cada controlador.**



Fonte: Autoria própria.



## 7.2 RESULTADOS EXPERIMENTAIS E DE SIMULAÇÃO

Foi escolhido para os testes simulados e práticos o controlador ótimo obtido utilizando a área de busca 2 por apresentar a melhor resposta dinâmica simulada. Com o controlador foram feitos os seguintes testes em bancada:

- Teste de mudança de referência de velocidade;
- Teste de rampa de velocidade;
- Teste de falha (simulação).

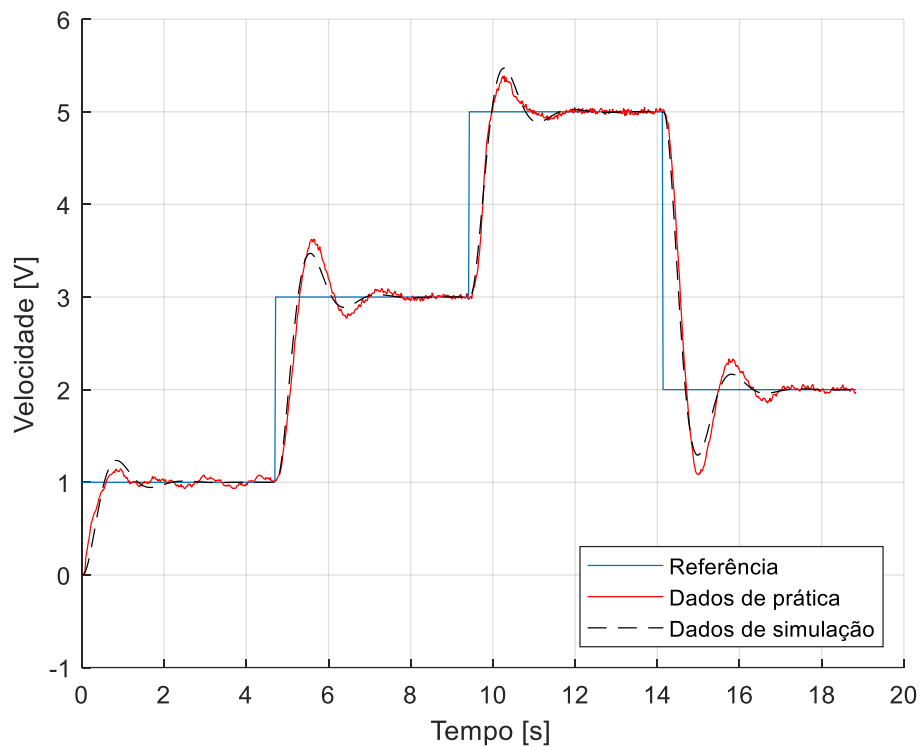
Os testes foram reproduzidos também em simulação para comparação e validação dos dados.

### 7.2.1 Teste de mudança de referência de velocidade

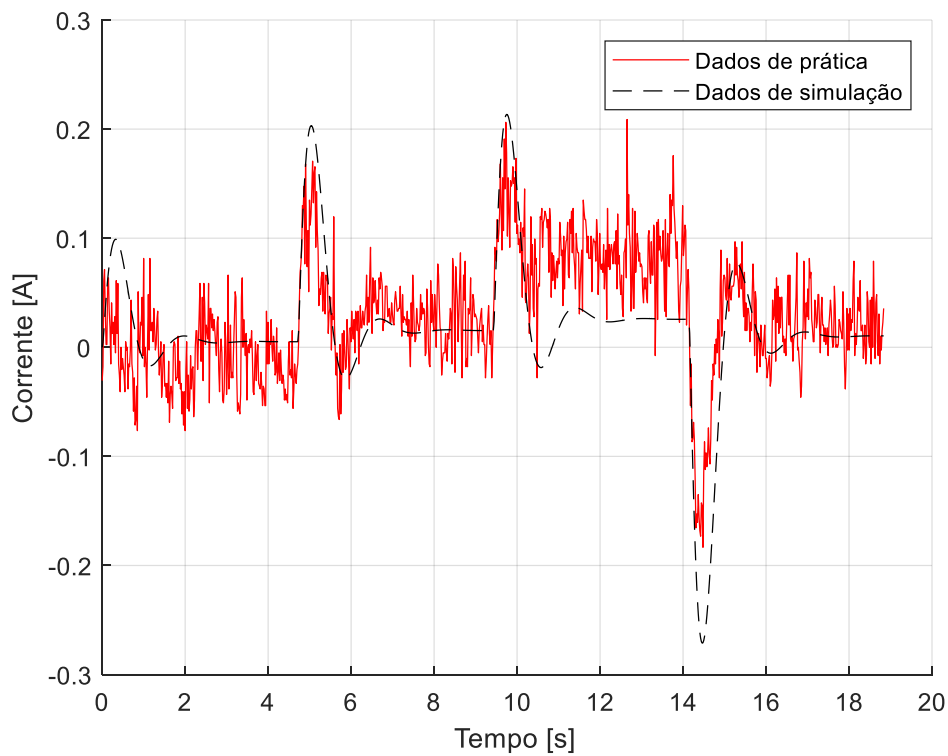
A resposta de velocidade obtida no teste está exibida na Figura 14. Pode-se observar que o sistema reage bem a mudança de referência, sempre seguindo-a com pouco erro, esse erro pode dar-se pela presença de ruído de amostragem. A velocidade leva aproximadamente 2 segundos para acomodar e apresenta um *overshoot* de 20%. É possível também perceber que a resposta prática é bem fiel ao resultado de simulação.

A resposta de corrente pode ser vista na Figura 15, podemos ver que a corrente do teste prático contém muito ruído, porém, é possível notar que a mesma tem o mesmo perfil que a corrente simulada.

A ação de controle também foi registrada durante o teste, e pode ser vista na Figura 16. Percebe-se uma pequena diferença entre as curvas a partir do quinto segundo do registro. Como a Figura 14 exhibe essa diferença no sinal de controle não faz com que a velocidade reaja de forma diferente. Pode-se creditar essa diferença ao fato de que a placa de aquisição não tem frequência de amostragem constante, diferente do software de simulação, fazendo com que a ação de controle tenha perfis diferentes. Essa diferença pode ser eliminada com uma placa de aquisição que contenha um *clock* interno com boa precisão.

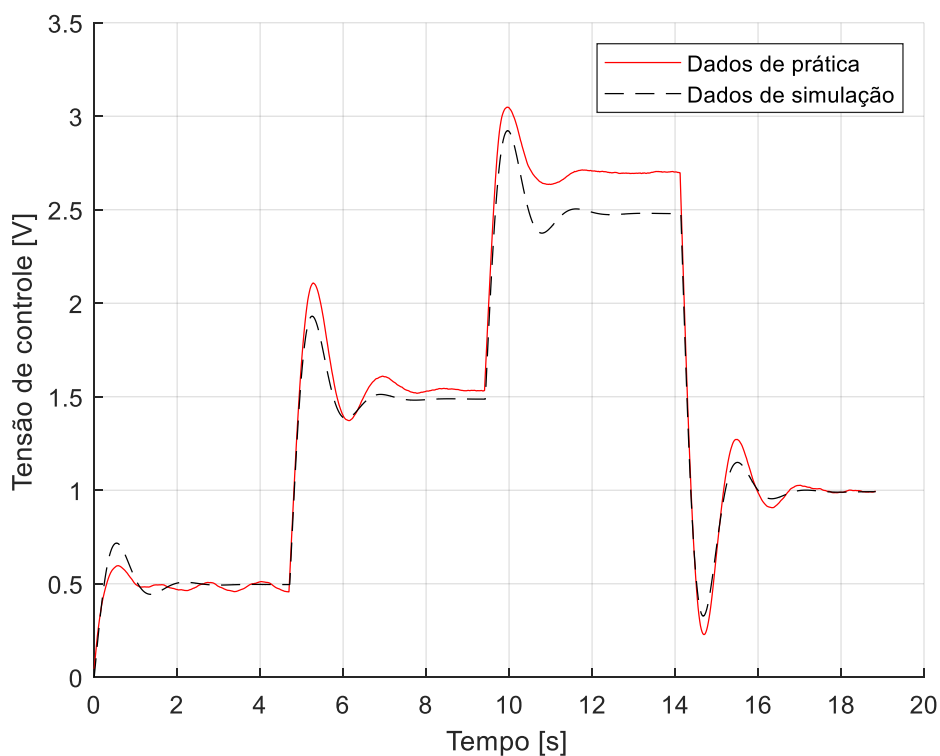
**Figura 14 – Resposta da velocidade à mudança de referência.**

Fonte: Autoria própria.

**Figura 15 – Resposta da corrente à mudança de referência.**

Fonte: Autoria própria.

**Figura 16 – Ação de controle.**



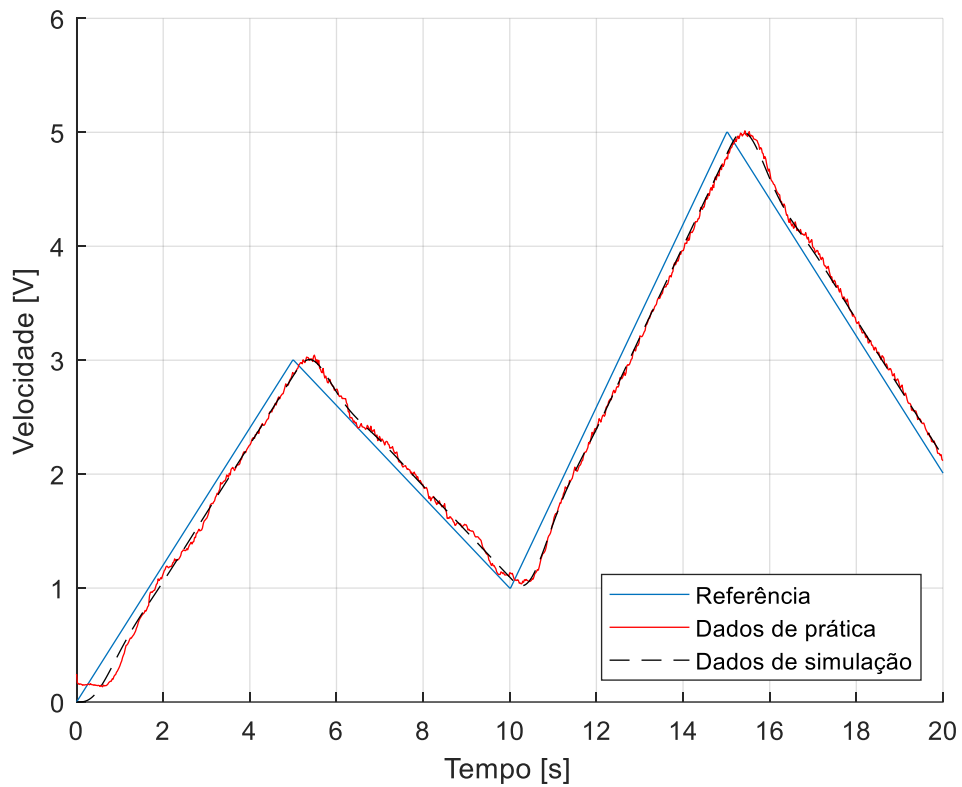
**Fonte: Autoria própria.**

### 7.2.2 Teste de rampa de velocidade

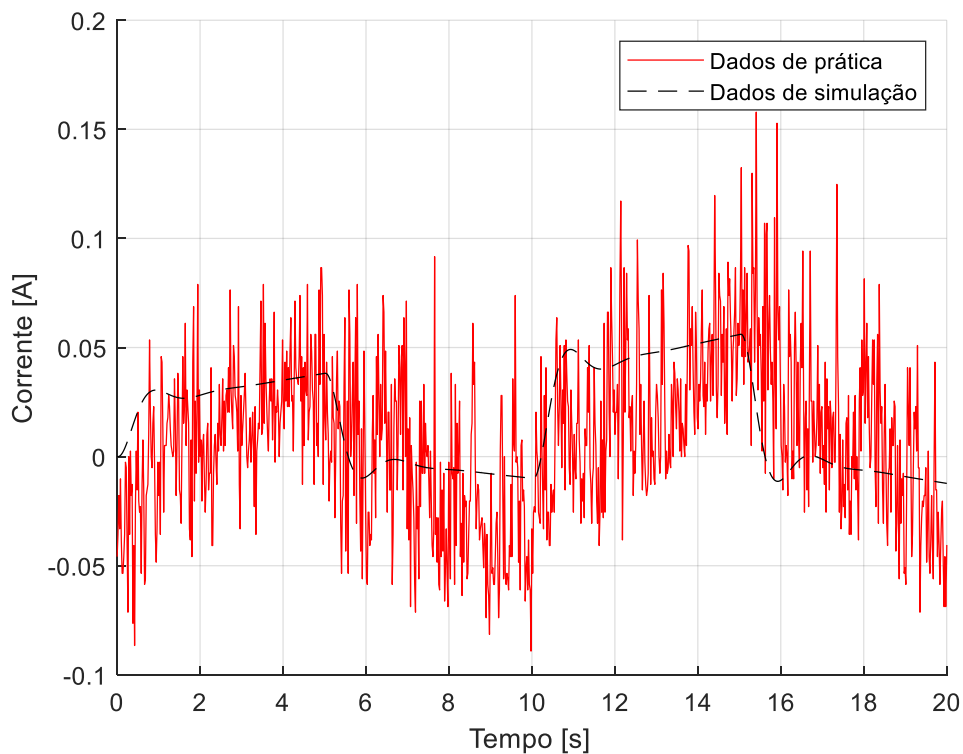
Pode-se observar na Figura 17 a resposta de velocidade a diferentes rampas de referência. Há um pequeno erro constante na resposta do sistema, porém, o controlador robusto não foi projetado para eliminar o erro para entradas de rampa. A simulação tem o mesmo perfil da prática, como no teste anterior.

A Figura 18 mostra a resposta de corrente. Nesse teste, a resposta prática apresenta ainda mais ruído, mesmo assim é possível ver que o perfil da resposta prática é similar ao da simulação.

Dessa vez, a ação de controle, mostrada na Figura 19, tem uma menor diferença entre a simulação e a prática. Como comentado na seção 7.2.1, a diferença entre as curvas pode ser atribuída ao fato de que a DAQ não tem um *clock* interno.

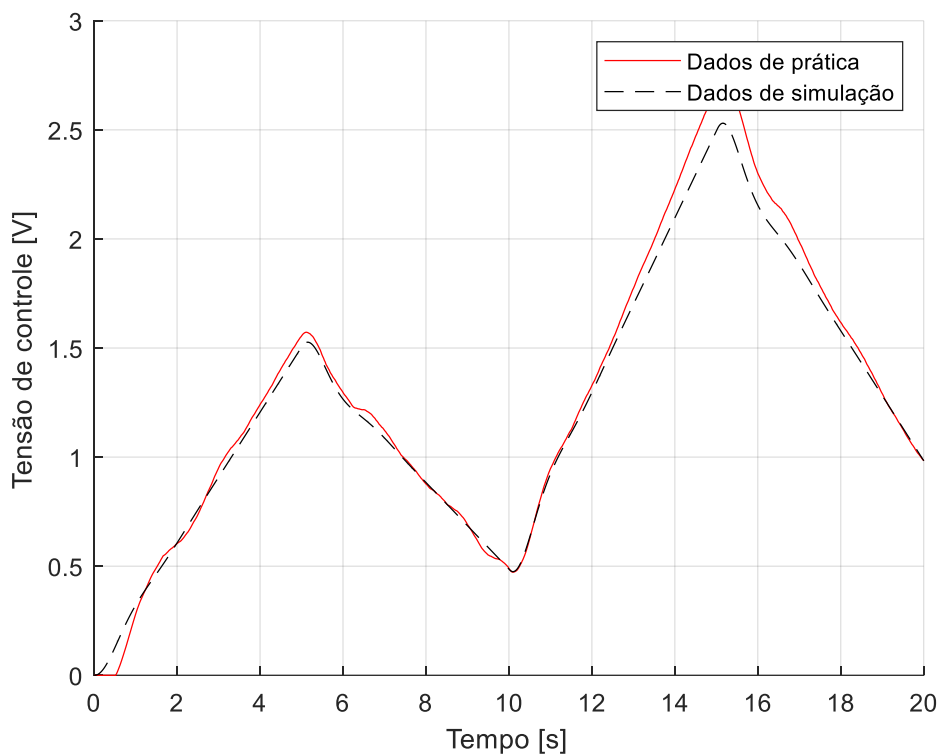
**Figura 17 – Resposta da velocidade à diferentes rampas de referência.**

Fonte: Autoria própria.

**Figura 18 – Resposta da corrente à diferentes rampas de referência.**

Fonte: Autoria própria.

**Figura 19 – Ação de controle.**



**Fonte: Autoria própria.**

### 7.2.3 Teste de falha

Infelizmente, o teste de falha não pode ser executado no módulo pois não há acesso ao circuito do motor (não há como modificar fisicamente o motor, considerando os componentes incertos,  $R_a$  e  $L_a$ ). Porém, é possível fazer os testes via simulação, e visto que, em todos os testes anteriores, os dados de velocidade obtidos em simulação e na prática foram muito próximos os resultados obtidos em simulação podem auxiliar na validação da eficácia do controlador robusto.

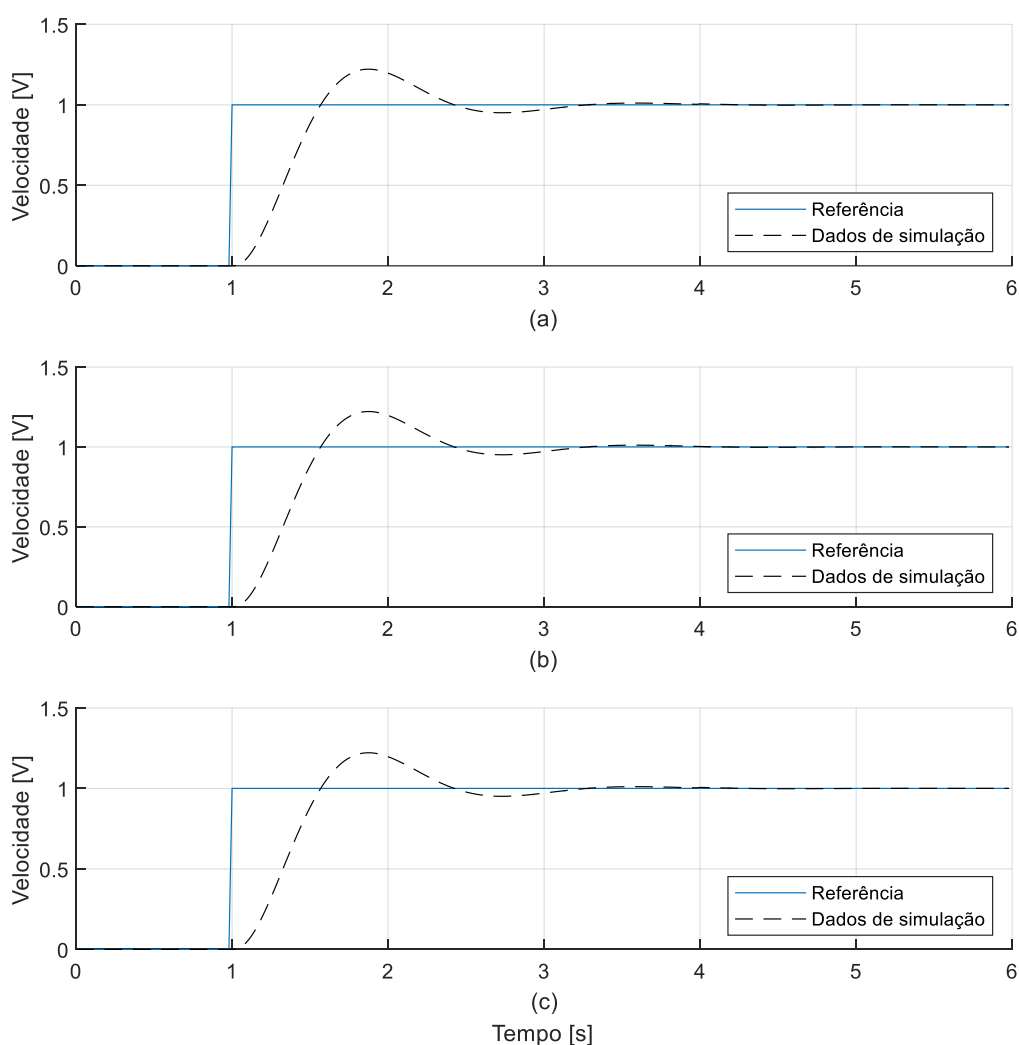
Nesse teste foram feitas três simulações. Em cada a simulação é iniciada com o modelo 1, aos 4 segundos de simulação ocorre a falha e o modelo é trocado pelo segundo, terceiro e quarto modelo respectivamente. Todos modelos foram controlados pelo mesmo controlador utilizado nos testes anteriores, e foram alimentados por um degrau unitário de tensão.

A Figura 20 mostra a resposta de velocidade ao degrau em cada simulação, (a), (b) e (c) representam a ocorrência de falha para os modelos 2, 3 e 4 respectivamente. A Figura 21 demonstra os mesmos dados com foco no instante de

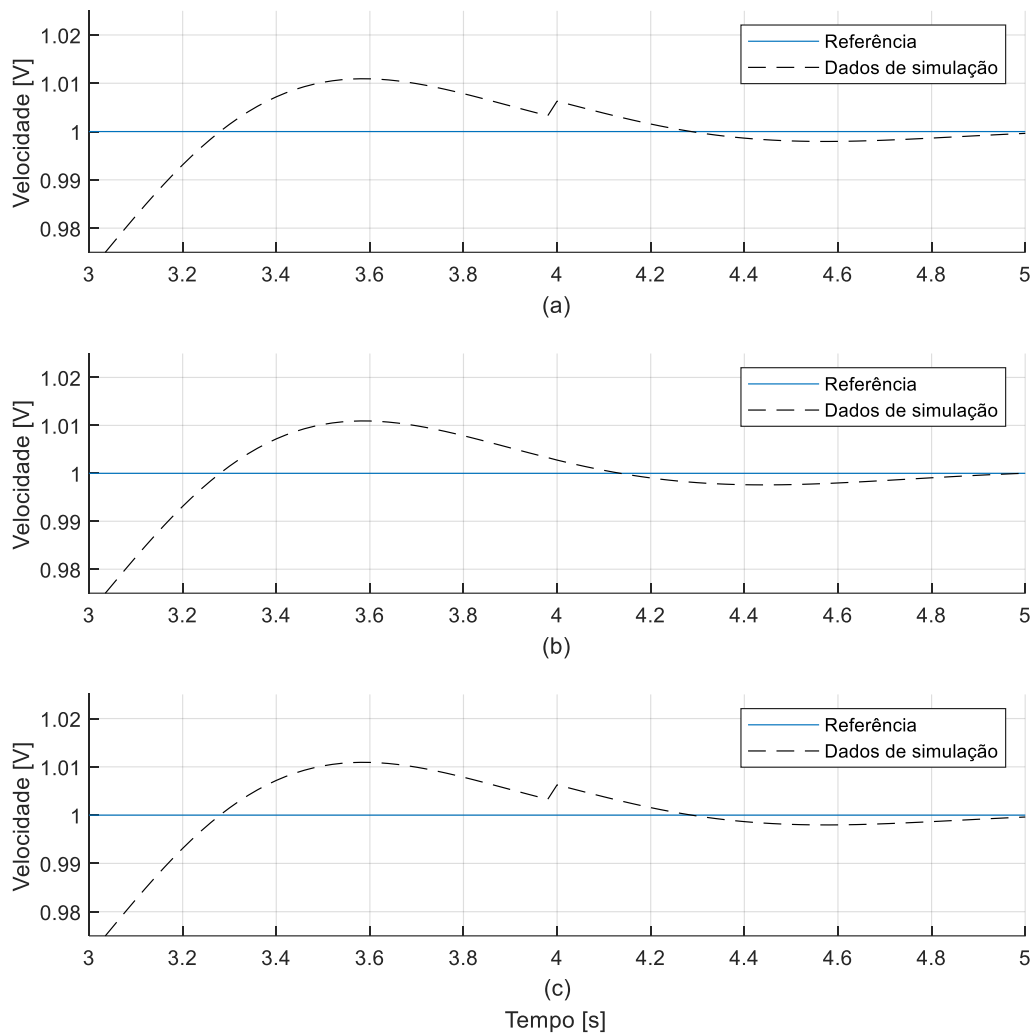
falha, (a), (b) e (c) representam novamente a ocorrência de falha para os modelos 2, 3 e 4 respectivamente.

É possível observar que as respostas de todos os modelos são parecidas, com exceção da simulação (b) onde a falha não apresentou variação visível na resposta de velocidade, e estáveis. Valida-se então a capacidade do controlador de estabilizar sistemas incertos.

**Figura 20 – Resposta da velocidade ao degrau.**



**Fonte: Autoria própria.**

**Figura 21 – Resposta da velocidade ao degrau - Instante de falha.**

Fonte: Autoria própria.

## 8 CONSIDERAÇÕES FINAIS

Utilizando a teoria apresentada foi implementado um algoritmo de evolução diferencial com o objetivo de otimizar escalares necessários para projetar robustamente ganhos de realimentação de estados e ganho de integração. Esses ganhos foram projetados através de um conjunto de LMIs com escalares, e avaliados pela função ITAE.

O algoritmo foi executado com diferentes configurações e os ganhos robustos obtidos em cada execução foram comparados. O melhor controlador foi selecionado para testes, e demonstrou boa performance no controle do motor CC. O motor foi submetido a uma entrada degrau e manteve-se o desempenho mesmo na presença de mudanças dos parâmetros do motor.

Os ganhos foram testados em um módulo didático e, mesmo com as limitações do módulo e da placa DAQ, foram obtidos resultados próximos aos de simulação. Não foi possível realizar testes para validar a robustez do controlador em bancada, pois o módulo utilizado não permite acesso ao circuito do motor, portanto não é possível fazer mudanças no mesmo. Além disso, também não foi possível testar a reação do sistema sob entrada e saída de carga do eixo, pois o módulo também não permite estas modificações.

Vale ressaltar que, a LMI utilizada para o projeto dos controladores robustos permite a apenas garantir a estabilidade do sistema. Todavia, o conjunto possui escalares que devem ser fornecidas para a execução dos projetos e, se otimizados de forma adequada como neste trabalho, podem proporcionar controladores com performance satisfatória e conseqüentemente respostas transitórias adequadas.

Dito tudo isto, é possível concluir que os objetivos do algoritmo foram atingidos e que é possível otimizar, com a LMI utilizada, a performance de motores maiores e de até outros sistemas, bastando fazer a modelagem e a obtenção correta dos parâmetros dos mesmos.



## REFERÊNCIAS

BETETO, Marco Antonio Leite. **Síntese de controladores robustos LQR-LMI via realimentação derivativa: aplicações em sistemas dinâmicos com falhas estruturais**. 2016. 61 f. Trabalho de Conclusão de Curso (Graduação), Engenharia de Controle e Automação. Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2016.

BOSCO, Maycon Chimini. **Estimação de parâmetros e controle de velocidade de um motor CC de imã permanente usando evolução diferencial**. 2016. 59f. Trabalho de Conclusão de Curso – Graduação em Engenharia Elétrica, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2016.

BOYD, S.; GHAOUI, L.E.; FERON, E.; et al. **Linear Matrix Inequalities in System and Control Theory**. Philadelphia: Society for Industrial and Applied Mathematics, 1994. (Studies in Applied Mathematics).

BUZACHERO, L. F. S. **Otimização de controladores robustos de sistemas dinâmicos sujeitos a falhas estruturais**. 2010. 72 f. p. Dissertação (Mestrado) — Universidade Estadual Paulista, Faculdade de Engenharia de Ilha Solteira, 2010. Disponível em: <<<https://repositorio.unesp.br/handle/11449/87054> >>.

CHAPMAN, S.J. **Fundamentos de Máquinas Elétricas**. 5. ed. [s.l.]: AMGH Editora, 2013.

CHESI, G. **LMI-Based Computation of the Instability Measure of Continuous-Time Linear Systems with a Scalar Parameter**. Hong Kong. 2014.

DATAPPOOL. **Módulo 2208 – Servomecanismo**. [201-]. Disponível em: <<http://eletronica.datapool.com.br/produtos/controle-e-automacao/modulo-2208-servomecanismo/>>. Acesso em 06 de maio de 2019.

EBIHARA, Y.; HAGIWARA, T. Robust Controller Synthesis with Parameter-Dependent Lyapunov Variables: A Dilated LMI Approach. Las Vegas. In: IEEE Conference on Decision and Control. 41., 2002, Las Vegas, Nevada. **Proceedings...** Kyoto: Kyoto University. p. 4179-4184.

GAHINET, P.; NEMIROVSKI, A.; LAUB, A.; CHILALI, M. **LMI Control Toolbox – For use with MATLAB**. The Math Works Inc., 1995.

JR, C. K.; UMANS, S. D.; FITZGERALD, A. E. **Máquinas Elétricas - 6.ed.:** Com Introdução à Eletrônica de Potência. 6. Ed. [s.l.]: Bookman, 2006.

KIRK, D. E. **Optimal Control Theory:** An Introduction. New York: Dover Publications, 2004. (Dover Books on Electrical Engineering Series).

KRISHNAN, R. **Electric Motor Drives:** Modeling, Analysis, and Control. New Jersey: Prentice Hall, 2001.

NATIONAL INSTRUMENTS. **USB-6009.** [201-]. <<http://www.ni.com/pt-br/support/model.usb-6009.html>>. Acesso em 06 de maio de 2019.

OGATA, K. **Engenharia de controle moderno.** 3. ed. [s.l.]: LTC, 1998.

PRICE, K.; STORN, R. M.; LAMPINEN, J. A. **Differential Evolution:** A Practical Approach to Global Optimization. [s.l.]: Springer Science & Business Media, 2006. (Natural Computing Series).

ROMÃO, L. B. R. R.; OLIVEIRA, R. C. L. F.; PERES, P. L. D. **Projeto de filtros robustos  $\mathcal{H}_2$  usando LMIs com escalares.** Campinas, SP, 2015.

VIEIRA, H. S. **Estabilização e controle  $H^\infty$  por realimentação de estados para sistemas lineares politópicos utilizando desigualdades matriciais com escalares.** 2015. 54 f. Dissertação (Mestrado) — Universidade Estadual De Campinas, Faculdade de Engenharia Elétrica e de Computação. Campinas, SP, 2015.

VIEIRA, H. S.; OLIVEIRA, R. C. L. F.; PERES, P. L. D. Relaxações LMIs com escalares para realimentação de estados robusta de sistemas politópicos. In: Congresso Brasileiro de Automática, 20., 2014, Belo Horizonte, MG, Brasil. **Anais...** Belo Horizonte: UFMG. p. 1112–1118.

ZHOU, K.; DOYLE, J. C. **Essentials of Robust Control.** [s.l.]: Prentice Hall, 1999. (Prentice Hall Modular Series for Eng).