

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

DIEGO ROCHA GUEDES  
HUGO LEONARDO OLIVEIRA DA CUNHA

**SÍNTESE DE CONTROLADORES ROBUSTOS APLICADOS EM SISTEMAS COM  
ATRASOS TEMPORAIS**

TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO  
2018

DIEGO ROCHA GUEDES  
HUGO LEONARDO OLIVEIRA DA CUNHA

**SÍNTESE DE CONTROLADORES ROBUSTOS APLICADOS EM SISTEMAS COM  
ATRASOS TEMPORAIS**

Trabalho de Conclusão de Curso apresentado ao Curso Superior de Engenharia de Controle e Automação da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do título de Engenheiro de Controle e Automação.

Orientador: Prof. Dr. Cristiano Marcos Agulhari

Coorientador: Prof. Dr. André Sanches Fonseca Sobrinho

CORNÉLIO PROCÓPIO  
2018



## **FOLHA DE APROVAÇÃO**

**Diego Rocha Guedes**

### **Síntese de controladores robustos aplicados em sistemas com atrasos temporais**

Trabalho de conclusão de curso apresentado às 13:30hs do dia 07/06/2018 como requisito parcial para a obtenção do título de Engenheiro de Controle e Automação no programa de Graduação em Engenharia de Controle e Automação 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). Cristiano Marcos Agulhari - Presidente (Orientador)

---

Prof(a). Dr(a). André Sanches Fonseca Sobrinho - (Coorientador)

---

Prof(a). Dr(a). Luiz Francisco Sanches Buzachero - (Membro)

---

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



## **FOLHA DE APROVAÇÃO**

**Hugo Leonardo Oliveira da Cunha**

### **Síntese de controladores robustos aplicados em sistemas com atrasos temporais**

Trabalho de conclusão de curso apresentado às 13:30hs do dia 07/06/2018 como requisito parcial para a obtenção do título de Engenheiro de Controle e Automação no programa de Graduação em Engenharia de Controle e Automação 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). Cristiano Marcos Agulhari - Presidente (Orientador)

---

Prof(a). Dr(a). André Sanches Fonseca Sobrinho - (Coorientador)

---

Prof(a). Dr(a). Luiz Francisco Sanches Buzachero - (Membro)

---

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

*“Vivemos numa sociedade intensamente dependente da ciência e da tecnologia, em que quase ninguém sabe algo sobre ciência e tecnologia.”*

*Carl Sagan*

## RESUMO

GUEDES, Diego Rocha; CUNHA, Hugo Leonardo Oliveira. **Síntese de controladores robustos aplicados em sistemas com atrasos temporais**. 2018. 105 f. Trabalho de Conclusão de Curso (Graduação) – Engenharia de Controle e Automação. Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2018.

Este trabalho visa o estudo, desenvolvimento e aplicação da síntese de uma técnica de controle robusto para sistemas afetados por atrasos temporais. A formulação envolve uma abordagem via Desigualdades Matriciais Lineares (do inglês *Linear Matrix Inequalities - LMI*) em um problema de otimização convexa, incorporando Reguladores Quadráticos Lineares (do inglês *Linear-Quadratic Regulators - LQR*), para que se minimize um custo garantido associado ao controle do sistema. O desenvolvimento baseia-se em uma técnica de síntese que já foi apresentada na literatura, que considera atrasos nos estados de um sistema incerto, tendo agora a contribuição deste trabalho com a extensão da síntese para o caso em que há também atrasos nas entradas de controle. A resolução do problema de otimização é feita na plataforma MATLAB, e o resultado fornece ganhos estabilizantes, por realimentação de estados, e com custo garantido otimizado. É apresentada também a implementação do controlador em um sistema embarcado, em que um microcontrolador da família PIC18 é responsável por controlar a velocidade de um servomotor, enquanto que atrasos propositais são inseridos na comunicação entre os dispositivos, a fim de testar e validar a técnica de síntese desenvolvida.

**Palavras-chave:** Sistemas com atrasos temporais, Controle de custo garantido, Controle Robusto, Desigualdades Matriciais Lineares (*LMIs*), Reguladores Quadráticos Lineares (*LQR*).

## ABSTRACT

GUEDES, Diego Rocha; CUNHA, Hugo Leonardo Oliveira. **Synthesis of robust controllers applied in time-delayed systems**. 2018. 105 p. Course Conclusion Work (Graduation) – Control and Automation Engineering. Federal University of Technology - Parana. Cornélio Procópio, 2018.

This work aims the study, development and application of the synthesis of a robust control technique for time-delayed systems. The synthesis is modeled by a LMI approach in the formulation of a convex optimization problem, incorporating LQR, so that the guaranteed cost associated with the control system is minimized. The development is based on a synthesis technique already presented in the literature that considers delays in the states of an uncertain system, being the greatest contribution of this work the extension of the synthesis to the case of delayed inputs. The resolution of the optimization problem is performed on the MATLAB platform, and provides stabilizing state feedback gains with an optimized guaranteed cost. A physical implementation of the controller on an embedded system is also presented, composed of a microcontroller of the PIC18 series, responsible for controlling the speed of a servomotor, while intentional delays are inserted in the communication between the devices in order to test and validate the developed synthesis technique.

**Keywords:** Time Delayed Systems, Guaranteed Cost Control, Robust Control, Linear Matrix Inequalities (LMIs), Linear Quadratic Regulators (LQR).

## LISTA DE FIGURAS

Figura 1 – Conjuntos convexos e não convexos.....	19
Figura 2 – Kit didático 2208 DATAPOOL.....	28
Figura 3 – Kit didático XM118 EXSTO.....	29
Figura 4 – Pinagem do PIC 18F4550.....	30
Figura 5 – Dispositivo DAQ NI-6008.....	32
Figura 6 – Circuito para identificação do sistema do servomecanismo.....	33
Figura 7 – Esquemático de funcionamento.....	38
Figura 8 – Esquemático do circuito.....	38
Figura 9 – Esquemático do vetor pilha.....	40
Figura 10 – Valores de RB1 medidos pelo Osciloscópio.....	41
Figura 11 – Esquemático de Sistema com atraso nos estados e na entrada de controle (Simulink).....	57
Figura 12 – Simulação do sistema exemplo com atraso na dinâmica com ganhos calculados pela síntese de Yu e Chu.....	58
Figura 13 – Simulação do sistema exemplo com atraso na dinâmica e na entrada de controle com ganhos calculados pela síntese de Yu e Chu.....	59
Figura 14 – Simulação do sistema exemplo com atraso na dinâmica e na entrada de controle com ganhos calculados pela síntese proposta.....	60
Figura 15 – Comparação de resposta do sistema estimado com a resposta do sistema real.....	62
Figura 16 – Diagrama de blocos com integrador (Simulink).....	64
Figura 17 – Simulação do modelo matemático do servomotor para 0,408s de atraso.....	65
Figura 18 – Simulação do modelo matemático do servomotor para 0,816s de atraso.....	65
Figura 19 – Dados recebidos pela comunicação serial.....	66
Figura 20 – Resposta do sistema embarcado para 0,408s de atraso.....	67
Figura 21 – Comparação entre tensão $V_n$ e simulada para atraso de 0,408s.....	68
Figura 22 – Comparação entre tensão no conector JST real e simulada para atraso de 0,408s.....	68



Figura 23 – Comparação entre sinal de controle real e simulado para atraso de 0,408s.....	69
Figura 24 – Resposta do sistema embarcado para 0,816s de atraso.....	70
Figura 25 – Comparação entre tensão Vn real e simulada para atraso de 0,816s.....	71
Figura 26 – Comparação entre tensão no conector JST real e simulada para atraso de 0,816s.....	71
Figura 27 – Comparação entre sinal de controle real e simulado para atraso de 0,816s.....	72

## LISTA DE QUADROS

Quadro 1 – Descrição dos pinos do PIC18F4550.....	30
---	----

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO.....</b>	<b>12</b>
1.1	Justificativa.....	13
1.2	Problematização.....	13
1.3	Objetivo geral.....	14
1.4	Objetivos específicos.....	15
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>16</b>
2.1	Estabilidade de Lyapunov.....	16
2.2	Problemas de Otimização Via Análise Convexa.....	18
2.3	Desigualdades Matriciais Lineares.....	20
2.4	Complemento de Schur.....	21
2.5	Incertezas Paramétricas Limitadas em Norma.....	21
2.6	Reguladores Quadráticos Lineares.....	23
2.7	Ação Integral.....	23
2.8	Implementação Física – Microcontrolador PIC.....	24
2.8.1	Sinal PWM.....	24
2.8.2	Conversor A/D.....	25
2.8.3	Comunicação Serial Assíncrona.....	26
<b>3</b>	<b>MATERIAIS E MÉTODOS.....</b>	<b>27</b>
3.1	Servomecanismo.....	27
3.2	Microcontrolador PIC 18F4550.....	28
3.3	Placa DAQ NI-6008.....	32
3.4	Plataforma MATLAB.....	33
3.4.1	Identificação do Modelo Matemático do Kit Servomecanismo.....	33
3.4.2	Construção e Resolução das LMIs.....	35
3.4.3	Simulação dos controladores via Simulink.....	37
3.5	Funcionamento do Projeto Embarcado.....	37
3.5.1	Estruturação do Atraso Temporal.....	39
3.5.2	Aquisição de dados.....	41
3.5.3	Geração da saída de controle.....	42
3.5.4	Transmissão de dados.....	44
<b>4</b>	<b>RESULTADOS E DISCUSSÕES.....</b>	<b>46</b>

<b>4.1</b>	<b>Síntese para Sistemas com Atraso nos Estados.....</b>	<b>46</b>
<b>4.2</b>	<b>Desenvolvimento de Síntese para Sistemas com Atraso nos Estados e na Entrada de Controle.....</b>	<b>50</b>
<b>4.3</b>	<b>Simulação das Sínteses Abordadas.....</b>	<b>56</b>
<b>4.4</b>	<b>Modelo Matemático em Espaço de Estados do Servomecanismo....</b>	<b>61</b>
<b>4.5</b>	<b>Geração dos Ganhos de Controle e Simulação para Modelo Matemático do Servomecanismo.....</b>	<b>62</b>
<b>4.6</b>	<b>Resultados obtidos para a implementação física.....</b>	<b>66</b>
<b>5</b>	<b>CONSIDERAÇÕES FINAIS.....</b>	<b>74</b>
	<b>REFERÊNCIAS.....</b>	<b>75</b>
	<b>ANEXO I – Código para identificação do modelo em Espaço de Estados do Servomotor.....</b>	<b>78</b>
	<b>ANEXO II – Código para síntese de ganho dos controladores.....</b>	<b>81</b>
	<b>ANEXO III – Código do firmware.....</b>	<b>85</b>
	<b>ANEXO IV – Código para separação de dados da comunicação serial.....</b>	<b>91</b>
	<b>ANEXO V – Registradores conversor A/D.....</b>	<b>93</b>
	<b>ANEXO VI – Registradores sinal PWM.....</b>	<b>98</b>
	<b>ANEXO VII – Registradores Comunicação Serial Assíncrona.....</b>	<b>101</b>

## 1 INTRODUÇÃO

Sistemas de controle são estruturas que manipulam, a partir de um controlador, os atuadores de um sistema físico para que este sistema apresente um comportamento final pré-determinado. Entretanto, sistemas físicos nem sempre possuem dinâmicas triviais devido a diferentes possíveis interferências, como as causadas pela própria integração do controlador no sistema. No projeto de controladores, mesmo quando os sistemas propostos são contínuos no tempo, a sua implementação é feita em geral através de dispositivos digitais que inevitavelmente geram atrasos de comunicação (SIMEÃO, 2009)

Outro fator a ser considerado ao projetar controladores para plantas reais é de que, além de ser necessário assegurar a estabilidade do sistema proposto, também é apropriado considerar questões relativas ao desempenho adequado para o controlador. Assim, existe a necessidade de projetar uma técnica que resulte em controladores cujos ganhos apresentem grandezas coerentes à realidade e que sejam capazes de otimizar parâmetros pré determinados do sistema. Neste trabalho essa necessidade é suprida através dos Reguladores Quadráticos Lineares (do inglês *Linear Quadratic Regulators - LQR*). Tais questões foram anteriormente estudadas por Chang e Peng (1972), quando esse conceito fora estabelecido pela primeira vez como controle com custo garantido, e permanecendo em uso como é notado em Yu e Chu (1999), Leite et al. (2007), Miranda et. al (2008), Guedes et. al (2017) bem como nas referências usadas por estes.

Esse tratamento leva em conta o fator de desempenho e as possíveis alterações devido à presença de incertezas na dinâmica do sistema, delimitadas por um limite superior pré-determinado. As incertezas citadas podem ser oriundas de diversas fontes, como por variações de parâmetros, dinâmicas que não foram previamente modeladas ou por presença de atrasos de comunicação entre as partes de um sistema. Para o projeto proposto por este trabalho, são consideradas as incertezas que podem ser estruturadas entre limites físicos, conhecidas como incertezas limitadas em norma.

## 1.1 Justificativa

É perceptível que atrasos temporais são encontrados nos mais diversos âmbitos de aplicação. Em processos biológicos, por exemplo, atrasos nas dinâmicas dos sistemas modelados podem surgir pelo tempo de condução do sistema nervoso, tempo de maturação em populações celulares e animais e tempo circulatório no sistema cardiovascular (MACDONALD, 1989). Em sistemas de engenharia, Kolmanovskii e Myshkis (1999) citam aplicações em reatores químicos e nucleares, controle de curso de navio e laminação de metais. Além desses, há também os sistemas com atrasos temporais envolvidos em redes de comunicação de dados, como sistemas de tráfegos de veículos e sistemas de teleoperação (NICULESCU, 2001).

O presente trabalho é uma proposta de síntese de controladores que tratam dessa classe de sistemas. Dessa forma, se baseando na técnica desenvolvida por Yu e Chu (1999), que apenas considera atrasos nos estados do sistema, os autores deste trabalho desenvolvem - como pode ser visto no artigo de mesmo nome publicado no Simpósio Brasileiro de Automação Inteligente de 2017 - uma adaptação da técnica capaz de lidar com incertezas na entrada de controle. Além da construção da nova abordagem, também são apresentados os resultados e análises de uma implementação física, utilizando dos kits didáticos Módulo 2208 – Servomecanismo, da empresa DATAPOOL, e XM118 - Banco de Ensaio em Microcontroladores PIC18F. O servomecanismo tem seu sistema devidamente identificado e, através do PIC18F, o controlador é estabelecido.

## 1.2 Problematização

Sistemas com incertezas paramétricas são frequentemente abordadas por técnicas de controle robusto, que são baseadas no uso de funções de Lyapunov, como a caracterização da estabilidade quadrática, por exemplo. Boyd et al. (1994) foi um dos precursores na modelagem destas técnicas a partir de resoluções por

desigualdade matriciais lineares (do inglês *Linear Matrix Inequalities - LMIs*), que é uma importante ferramenta para formulação de problemas em sistemas de controle capaz de ser resolvida computacionalmente por um algoritmo de otimização convexa sempre que existir uma solução possível para o problema proposto. Dessa maneira, quando isso ocorre, a estabilidade do sistema é assegurada e o custo garantido é otimizado quando uma solução factível é encontrada para o conjunto de *LMIs* que representam as condições propostas pela síntese de controle.

Estes sistemas recebem uma atenção considerável na literatura pois, como já dito, atrasos em sistemas reais sempre foram características frequentes, seja devido a propriedades intrínsecas do sistema ou por consequência da própria ação de controle. Além de serem frequentes, estes atrasos temporais também impactam negativamente a estabilidade dos sistemas. Assim, a abordagem através das funcionais de Lyapunov-Krasovskii (Kolmanovskii et al, 1999), desenvolvidas por meio de *LMIs*, se apresentam como uma proposta de análise que atesta a estabilidade robusta dos sistemas.

### **1.3 Objetivo Geral**

Para este trabalho, tem-se como objetivo desenvolver um controlador robusto capaz de tratar das incertezas limitadas em norma e atrasos temporais nos estados do sistema e na realimentação do controlador, garantindo ainda um custo de controle otimizado. Técnicas que consideram atrasos na entrada são pouco exploradas na bibliografia atual, cabendo citar Mukaidani (2003) pelo qual fora o único trabalho encontrado desenvolvendo, de forma distinta, um técnica de síntese que considere tal atraso.

Como dito, tal síntese já havia sido implementada por Yu e Chu (1999) sem a consideração de atrasos temporais na entrada de controle, portanto esse trabalho se trata de uma contribuição à síntese desenvolvida por estes, além de possibilitar uma aplicação prática da técnica, que permite a validação de sua eficiência

## 1.4 Objetivos Específicos

- Estudar e projetar técnicas de síntese de controladores robustos aplicadas a sistemas com incertezas limitadas em norma e atrasos temporais;
- Desenvolver condições *LMIs* para otimizar o custo de ganho de controle considerando atrasos temporais nos estados e na entrada de controle;
- Aplicar a síntese de controle desenvolvida em um sistema físico composto de um kit didático Módulo 2208 – Servomecanismo, da empresa DATAPOOL;
- Utilizar o *kit* didático XM118 - Banco de Ensaio em Microcontroladores PIC18F para desenvolvimento da malha de controle do sistema;
- Validar o desempenho da técnica desenvolvida a partir de testes no sistema embarcado projetado.



## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são abordados os conceitos utilizados para desenvolvimento da técnica de síntese proposta. Dessa forma serão esclarecidos alguns conceitos como Desigualdades Matriciais Lineares (*LMIs*), que é a abordagem matemática utilizada para resolução do problema que visa minimizar o custo garantido de um sistema com atraso temporal nos estados e na entrada de controle. Por fim são apresentados aspectos da implementação física pretendida, com uma breve introdução sobre microcontroladores PIC.

### 2.1 Estabilidade de Lyapunov

A análise de estabilidade representa um papel central em teoria de sistemas e engenharia. Existem diferentes tipos de problemas de estabilidade no estudo de sistemas dinâmicos (SCHERER; WEILAND, 2005), sendo que, para o presente trabalho, refere-se à estabilidade no sentido Lyapunov, descrita a seguir.

Lyapunov foi um matemático e engenheiro russo que desenvolveu a teoria que hoje carrega seu nome. Sua principal contribuição foi redefinir o conceito de estabilidade, estabilidade assintótica e instabilidade de sistemas dinâmicos, além de estabelecer um método de verificação dessas propriedades em termos de existência das chamadas funções de Lyapunov (SCHERER; WEILAND, 2005). Os métodos de Lyapunov também podem ser utilizados para encontrar limitantes para critérios de desempenho definidos para um sistema (BOYD et al., 1994).

Inicialmente, considere o sistema

$$\dot{x}(t) = f(x(t)), \quad x(t) \in \mathbb{R}^n, \quad f(0) = 0. \quad (1)$$

Seja  $x_e$  um ponto de equilíbrio do sistema (1), e  $D$  um domínio de  $x(t)$  contendo  $x_e$ . O sistema (1) é estável em uma vizinhança  $\beta(x_e, \rho)$  em torno de  $x_e$  com raio  $\rho$ , se existir uma função contínua  $V: \mathbb{R}^n \rightarrow \mathbb{R}$  com derivadas contínuas em relação a  $t$ , tal que:

$$\begin{aligned} (i) \quad & V(x(t)) \geq 0; \\ (ii) \quad & V(x(t)) > 0 \text{ em } D - \{x_e\}; \\ (iii) \quad & \dot{V}(x(t)) \leq 0 \text{ em } D. \end{aligned} \tag{2}$$

A função  $V(x(t))$  que satisfaz o conjunto de condições (2) é chamada função de Lyapunov. Ainda cabe dizer que se  $\dot{V}(x(t)) < 0$  em  $D - \{x_e\}$ , então o sistema é assintoticamente estável para a vizinhança  $\beta(x_e, \rho)$  (KHALIL, 2002). Dessa forma, qualquer trajetória de  $x(t)$ ,  $t \geq 0$ , tende a se aproximar arbitrariamente de  $x_e$  quando  $t \rightarrow \infty$ . E, por último, a função  $V(x(t))$  é dita radialmente ilimitada se  $\|x(t)\| \rightarrow \infty$  implica em  $V(x(t)) \rightarrow \infty$ .

Estendendo a análise anterior, se para todo  $x(t) \in \mathbb{R}^n$ ,  $V(x(t))$  satisfaz o conjunto de condições (2), se  $V(x(t))$  for radialmente ilimitada, e  $\dot{V}(x(t)) < 0$ , o sistema é globalmente assintoticamente estável (KHALIL, 2002).

Considere agora um sistema na forma

$$\dot{x}(t) = Ax(t), \tag{3}$$

em que  $A \in \mathbb{R}^{n \times n}$ . Uma condição necessária e suficiente para estabilidade do sistema (3) é a existência de uma função quadrática  $V(x(t)) = x'(t)Px(t)$ ,  $P > 0$  que atende o conjunto de condições (2). Se a existência de  $P$  é garantida, diz-se que o sistema (3) é quadraticamente estável e  $V$  é chamada Função de Lyapunov Quadrática.

## 2.2 Problemas de Otimização via Análise Convexa

Um problema de otimização envolve a especificação das soluções candidatas e da formalização do conceito de decisão ótima para um problema proposto (SCHERER; WEILAND, 2005). Seja  $T$  o conjunto de todas as decisões possíveis em um problema de otimização, e  $H$  um subconjunto de  $T$  que contém todas as decisões factíveis (ou candidatas) de tal problema. A abordagem que visa quantificar o desempenho das soluções factíveis  $y \in H$  expressa os valores na forma  $J(y)$ , em que  $J: H \rightarrow \mathbb{R}$  é uma função de valores reais chamada função objetivo ou função de custo. O valor das decisões  $y \in H$  é dado por  $J(y)$ .

A partir da interpretação da função de custo, deseja-se minimizar ou maximizar os valores de  $J$  dentre todas as candidatas. Dessa forma, a decisão ótima se caracteriza como um elemento de  $H$  que minimize ou maximize  $J$  dentre todas as alternativas factíveis.

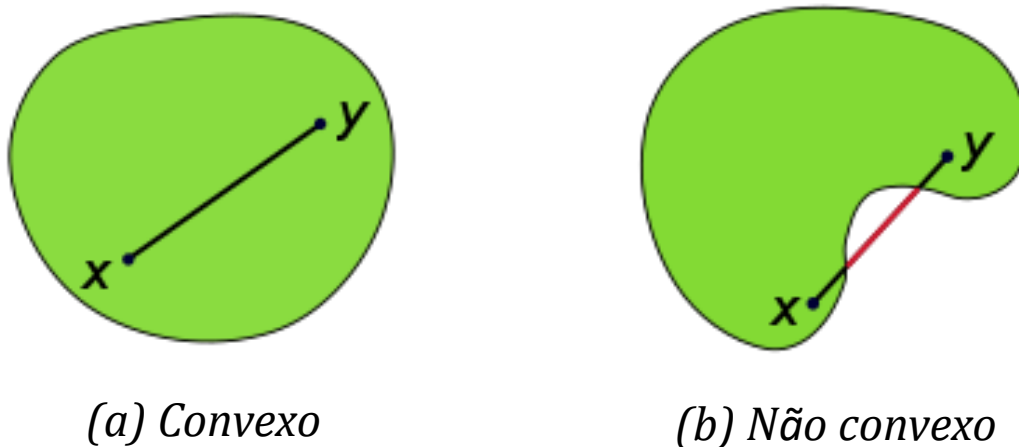
Em um problema de minimização, deve-se atenção a alguns fatores, como não assumir como solução ótima global candidatas que representam na verdade mínimos locais de  $J(y)$  (um erro comum em métodos numéricos), ou, se existe uma solução ótima, verificar se ela é única, e, caso contrário, escolher a solução que melhor se enquadra à realidade do problema.

A vantagem de uma análise em conjuntos convexos, ou funções convexas em um problema de otimização é a ausência de mínimos locais (SCHERER; WEILAND, 2005). Um conjunto  $H$  em um espaço vetorial linear é dito convexo se

$$\{y_1, y_2 \in H\} \Rightarrow \{y := \sigma y_1 + (1 - \sigma)y_2 \in H, \forall \sigma \in (0,1)\}. \quad (4)$$

Em termos geométricos, para quaisquer dois pontos em um conjunto convexo, o segmento de reta que os conecta deve pertencer a esse mesmo conjunto, como ilustrado pela Figura 1.

Figura 1 – Conjuntos convexos e não convexos



Fonte: [https://pt.wikipedia.org/wiki/Conjunto\\_convexo](https://pt.wikipedia.org/wiki/Conjunto_convexo)

De maneira semelhante, uma função  $J: H \rightarrow \mathbb{R}$ , é dita convexa se  $H$  é um conjunto convexo não vazio e se para todo  $y_1, y_2 \in H$  e  $\sigma \in (0,1)$ , têm-se

$$J(\sigma y_1 + (1 - \sigma)y_2) \leq \sigma J(y_1) + (1 - \sigma)J(y_2). \quad (5)$$

Ainda,  $J$  é dita estritamente convexa se a desigualdade (5) é válida para todo  $y_1, y_2 \in H$ , sendo  $y_1 \neq y_2$ , e todo  $\sigma \in (0,1)$ .

Por fim, considerando  $J$  convexa, e supondo  $y_0 \in H$  como uma solução ótima local de  $J$ , então para todo  $y \in H$ , e  $\sigma \in (0,1)$  suficientemente pequeno,

$$J(y_0) \leq J(y_0 + \sigma(y - y_0)) = J((1 - \sigma)y_0 + \sigma y) \leq (1 - \sigma)J(y_0) + \sigma J(y), \quad (6)$$

que implica em

$$0 \leq \sigma(J(y) - J(y_0)), \quad (7)$$

ou  $J(y_0) \leq J(y)$ . Assim, toda solução ótima local de  $J$  é também uma solução ótima global se a função  $J(y)$ , bem como o conjunto  $H$ , forem convexos. E então, se  $J$  for estritamente convexa,  $y_0$  é solução única (SCHERER; WEILAND, 2005).

### 2.3 Desigualdades Matriciais Lineares

Desigualdades Matriciais Lineares (*LMIs*) foram primeiro introduzidas por volta de 1890 por Lyapunov, quando este as usou para provar a estabilidade quadrática de um sistema dinâmico. Entretanto, só foram largamente exploradas como ferramentas de resolução de problemas da engenharia a partir da década de 1960. Desde a década de 1970, tornou-se ferramenta poderosa em teoria de sistemas e controle, uma vez que as *LMIs* originadas da área podem ser formuladas como problemas de otimização convexa, acessíveis para resolução computacional (BOYD et al., 1994).

Em termos matemáticos, uma *LMI* é uma expressão na forma

$$F(z) := F_0 + \sum_{i=1}^m z_i F_i < 0, \quad (8)$$

em que  $z \in \mathbb{R}^m$  é um vetor de números reais chamados de variáveis de decisão,  $F_i = F_i' \in \mathbb{R}^{m \times m}$ ,  $i = 0, \dots, m$  são matrizes simétricas reais e o símbolo de desigualdade em (8) significa “definida-negativa”. Há também *LMIs* na forma não estritamente negativas como

$$F(z) \leq 0. \quad (9)$$

É comum encontrar problemas em que ambos os tipos estão presentes.

## 2.4 Complemento de Schur

Como descrito na subseção 2.3, uma *LMI* pode estar nas formas (8) e (9). Quando uma *LMI* não se encontra nessas formas as desigualdades são não-lineares. Para estes é comum utilizar transformações ou outras desigualdades conhecidas na literatura para que as condições se tornem *LMIs*.

Desigualdades não-lineares, desde que sejam convexas, podem ser convertidas para a forma de *LMIs* utilizando complementos de Schur (BOYD et al., 1994). Sendo assim, considere as desigualdades não-lineares convexas

$$C > 0, \quad A - BC^{-1}B' > 0; \quad (10.1)$$

ou

$$A > 0, \quad C - B'A^{-1}B > 0. \quad (10.2)$$

A *LMI*

$$M = \begin{bmatrix} A & B \\ B' & C \end{bmatrix} > 0 \quad (11)$$

é o equivalente pelo complemento Schur das desigualdades (10.1) e (10.2).

## 2.5 Incertezas Paramétricas Limitadas em Norma

Modelos matemáticos de sistemas físicos são frequentemente representados na forma de espaço de estados, em que suas componentes

representam quantidades físicas bem definidas. Entretanto, variações, perturbações ou incertezas em tais parâmetros físicos levam à incerteza do modelo.

Como pequenas variações podem ter um impacto considerável na dinâmica de um sistema, faz-se necessário considerar essas perturbações no projeto de controladores. Visto que a natureza e extensão dessas incertezas atuando em um sistema pode ser parcialmente ou totalmente conhecida, durante a construção do modelo matemático utiliza-se uma inclusão de termos diferenciais lineares (do inglês *Linear Differential Inclusions*, ou *LDI*) (BOYD et al., 1994). Essas inclusões estão divididas em pelo menos dois casos distintos: incertezas paramétricas invariantes no tempo e incertezas paramétricas variantes no tempo (SCHERER; WEILAND, 2005). Neste trabalho, os atrasos temporais atuantes em um sistema dinâmico foram incluídos no modelo matemático na forma de incertezas paramétricas variantes no tempo e limitadas por norma.

Seja  $\delta = [\delta_1 \dots \delta_n]'$  o vetor que contém as incertezas do sistema. Suponha que seus elementos são desconhecidos e limitados por norma. Considere o sistema

$$\dot{x}(t) = A(\delta)x(t) + B(\delta)u(t), \quad (12)$$

afetado pelas incertezas  $\delta$ . A representação do sistema (12) com as incertezas limitadas por norma é dada por

$$\dot{x}(t) = (A + \Delta A)x(t) + (B + \Delta B)u(t), \quad \Delta(\cdot) = DF(t)E(\cdot) \quad (13)$$

em que  $F(t)$  é uma função matricial desconhecida, na forma de um bloco diagonal  $diag(\delta_i I_{n_i})$  e  $|\delta_i| \leq 1$ , satisfazendo  $F'(t)F(t) \leq I$ . As matrizes  $D$  e  $E$  são matrizes conhecidas que indicam as direções de entrada de  $F(t)$ .

## 2.6 Reguladores Quadráticos Lineares

A fim de otimizar o custo de controle de um sistema dinâmico através de critérios de desempenho, é possível utilizar os Reguladores Quadráticos Lineares (do inglês *Linear Quadratic Regulators*, ou *LQR*).

O *LQR* é uma técnica para cálculo de um ganho por realimentação de estados que minimiza a função de custo associado ao sistema. Um sistema dinâmico contínuo no tempo na forma (12), possui uma função de custo

$$J = \int_0^{\infty} [x'(t)Qx(t) + u'(t)Ru(t)]dt, \quad (14)$$

em que  $Q$  e  $R$  são matrizes simétricas positivas-definidas com a finalidade de ponderar, respectivamente, a velocidade com que os estados atingem seu valor de regime permanente e a energia de controle atuante (KWAKERNAAK; SIVAN, 1972).

## 2.7 Ação Integral

A ação integral tem por objetivo eliminar o erro em regime estacionário entre o valor de resposta de um sistema e sua referência. Isso acontece porque a componente integral soma o termo de erro ao longo do tempo, em intervalos regulares, de forma que mesmo um erro pequeno faz com que a componente integral aumente lentamente. Assim sendo, a resposta integral irá aumentando ao longo do tempo a menos que o erro seja nulo, conduzindo o erro em regime permanente para zero.

O sinal do integrador é dado pela equação (15)



$$I = K_i \int_0^t e(\tau) d\tau, \quad (15)$$

em que  $K_i$  é o ganho integral e  $e(\tau)$  é o erro entre a referência e a resposta do sistema.

## 2.8 Implementação Física – Microcontroladores PIC

Os microcontroladores PIC (*Peripheral Integrated Controller*) são fabricados pela *Microchip Technology* e baseiam-se na arquitetura Harvard, com conjunto de instruções no modelo RISC. Possuem barramento de dados de 8 *bits*, e barramento de instruções divididos em famílias de 12, 14 e 16 *bits*.

Entre os principais periféricos (variando de acordo com o modelo) estão: Conversores analógico-digitais, contadores (*timers*) de 8 e 16 *bits*, comparadores analógicos, USARTs, controladores de comunicação (I<sup>2</sup>C, SSP e USB), controladores PWM, controladores LCD, controladores Ethernet, *watchdog timer*, portas digitais com capacidade de 25 mA, entre outros. Os periféricos utilizados neste projeto terão mais atenção nas seções subsequentes.

O microcontrolador abordado na proposta de trabalho é o PIC18F4550 da família PIC 18 de microcontroladores 16 *bits*. Para tanto, utiliza-se de um *kit* didático fabricado pela *EXSTO*, no modelo MX118. Os detalhes do *kit* podem ser conferidos no Capítulo 3.

### 2.8.1 Sinal PWM

Sinais PWM, do inglês *Pulse Width Modulation*, nada mais são do que sinais de onda quadrada com período fixo e largura de pulso variável. A razão entre a largura do pulso em estado lógico alto e o período do sinal é definido como ciclo de

trabalho, mais conhecido pela sua tradução livre para o inglês: *duty cycle*. E a frequência do sinal é equivalente ao inverso do período. Todavia, esses dois parâmetros são os responsáveis por estabelecer o sinal PWM.

Os módulos CCPx são responsáveis pela geração deste sinal e necessitam de uma base de tempo para ditar a sua frequência. Para tal, o CCPx utiliza o TIMER2 - outro periférico do PIC em questão. Os registradores associados aos módulos requisitados são CCPRxL, CCPRxH, CCPxCON, PR2 e T2CON - disponíveis da seção ANEXO deste trabalho, e são através destes que são feitas as configurações necessárias.

Para a definição do período, isto é, o inverso da frequência do sinal, utiliza-se da seguinte equação:

$$T_{PWM} = (PR2 + 1) \times 4 \times T_{OSC} \times PS \quad (16)$$

Onde:

$T_{PWM}$  = Período do PWM;

PR2 = Registro de comparação do Timer2;

$T_{OSC}$  = Período de um clock do oscilador externo;

PS = Pré-escala do TMR2

Já o *duty cycle*, como mencionado, é definido pela razão do tempo do pulso em nível alto e o período do sinal:

$$DC = \frac{tp}{T} = \frac{[CCPRxL \text{ CCPxCON} \langle 5:4 \rangle]}{(PR2+1) \times 4} \quad (17)$$

### 2.8.2 Conversor A/D

Converter sinais analógicos em sinais digitais com amplitudes correspondentes dependem diretamente de conceitos como quantização e taxa de

amostragem. O primeiro estabelece a escala de valores, em bits, que a amplitude do sinal poderá ser representada. Já o segundo determina a frequência em que as amostras são coletadas. Essa frequência, segundo o teorema de amostragem, deve ser pelo menos duas vezes maior que a frequência do sinal que está sendo convertido, para que ele possa ser reconstruído com fidelidade (PEREIRA, 2010).

O PIC18F4550 possui um conversor de 10bits, com 13 canais, intitulado como ADC (do inglês *Analog-to-Digital Convert*). Ele é o responsável por converter a tensão lida em um dos canais em uma palavra binária. Os parâmetros necessários para a sua configuração são as tensões de referência,  $V_{ref+}$  e  $V_{ref-}$  - que determinam os valores de tensão máximo e mínimo, e o tempo de digitalização das amostras - definidos pelo clock de conversão ( $T_{AD}$ ) e tempo de aquisição ( $T_{ACQ}$ ). Os registradores associados à configuração necessária são ADCON0, ADCON1 e ADCON2, também anexados a esse trabalho.

### 2.8.3 Comunicação Serial Assíncrona

Em uma comunicação serial assíncrona os dados são transmitidos em forma de caracteres padronizados pelo código ASCII (do inglês *American Standard Code for Information Interchange*), de um transmissor para um receptor. Sendo assim, uma palavra binária correspondente ao caractere em questão é enviado sequencialmente em um mesmo meio físico para o seu destino. No PIC18F4550 a interface que define o padrão físico e elétrico do dispositivo de comunicação é a norma RS-232C.

O módulo EUSART (do inglês *Enhanced Universal Synchronous/Asynchronous Receiver/Transmitter*) do PIC em uso é um dos responsáveis pela configuração dessa comunicação. Os registradores envolvidos na operação são TXSTA, RCSTA e BAUDCON. E, por fim, para a delimitação da velocidade de comunicação é necessária a escolha do BRG (do inglês *Baud Rate Generator*) que deverá ser armazenado nos registradores SPBRGH e SPBRG.

### 3 MATERIAIS E MÉTODOS

Neste capítulo são apresentados os materiais necessários para a implementação do sistema físico a ser controlado pela técnica de controle desenvolvida. Além disso, também são relatados os métodos utilizados para a construção e execução do trabalho. Para tanto, faz-se necessário uma introdução sobre sistemas dinâmicos afetados por atrasos temporais. Um sistema do tipo  $\dot{x}(t) = Ax(t)$ , quando afetado por atrasos é dado na forma

$$\dot{x}(t) = Ax(t) + A_1x(t - d) \quad (18)$$

em que  $d$  representa o tamanho do atraso afetando o sistema,  $A_1$  representa uma matriz de dinâmica atrasada,  $x(t)$  são as variáveis de estado atuais e  $x(t - d)$  representam as variáveis de estado atrasadas.

Dessa forma o controle por realimentação de estados retorna os sinais de controle  $u(t) = Kx(t)$  e  $u(t - d) = K_u x(t - d)$ , sendo esse último resultante da técnica de síntese proposta neste trabalho. Os sinais de controle apresentados representam a entrada de controle atual e a entrada de controle atrasada do sistema.

#### 3.1 Servomecanismo

O sistema físico proposto para implementação do controlador desenvolvido é o *kit* didático Módulo 2208 – Servomecanismo, da empresa DATAPOOL, que possibilita a análise da posição e/ou da velocidade angular do servo motor. Tal equipamento possui transdutores de rotação (tacômetro) e sensores de deslocamento (servo-potenciômetros) para medirem as dinâmicas citadas com

seus respectivos sinais já devidamente condicionados internamente ao *kit* para funcionarem entre tensões de 0 a 5V e correntes de 0 a 20mA (DATAPOOL, 2006).

O motor empregado nesse *kit* didático é um servo motor de corrente contínua que sofre intervenção na sua tensão de armadura, através de um *driver* com sinais de comando responsáveis pela ação de controle. Os sinais de controle do motor também funcionam entre tensões de 0 a 5V e correntes de 0 a 20mA que, por sua vez, são convertidas pelo *driver* para sinais de atuação de tensões e correntes de +/- 12V e 1,5 A, respectivamente.

Esses sistemas ainda possuem variáveis K1 e K2 que geram distúrbios causados por variações do torque, que podem ser tratadas como incertezas paramétricas pela técnica de controle proposta. A Figura 2 ilustra o *kit* didático.

**Figura 2 – Kit didático 2208 DATAPOOL**

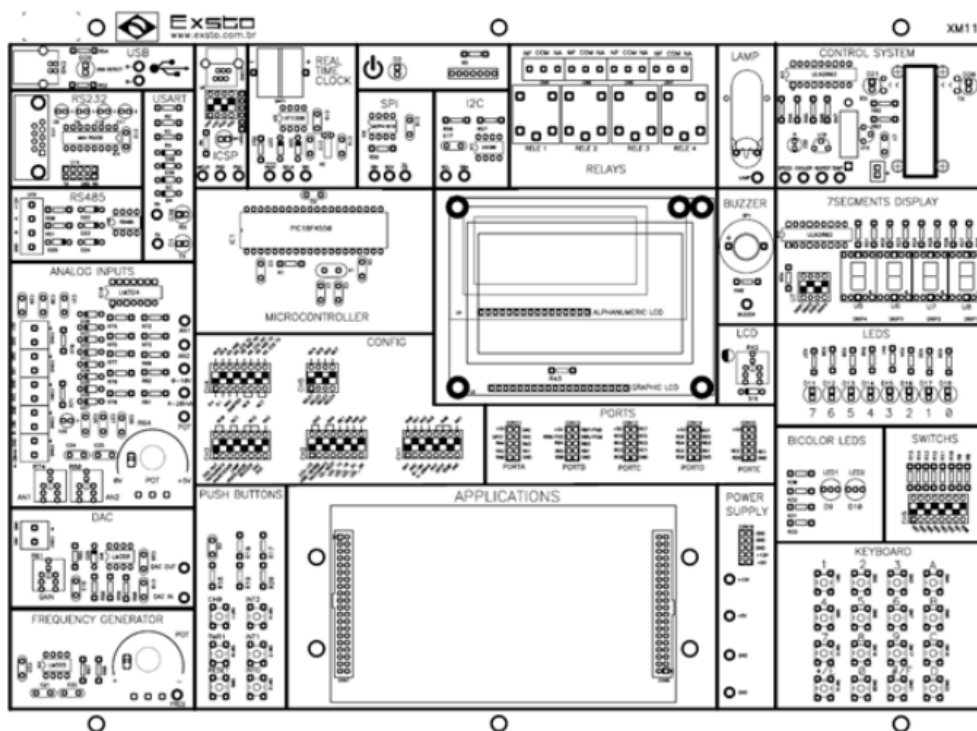


Fonte: Datapool Eletrônica Ltda

### **3.2 Microcontrolador PIC 18F4550**

Para a comunicação entre os componentes do sistema físico, bem como para a construção lógica da técnica de controle para o desenvolvimento do projeto, será utilizado o *kit* didático XM118 - Banco de Ensaio em Microcontroladores PIC18F. A Figura 3 apresenta o esquemático do *kit*.

Figura 3 – Kit didático XM118 EXSTO



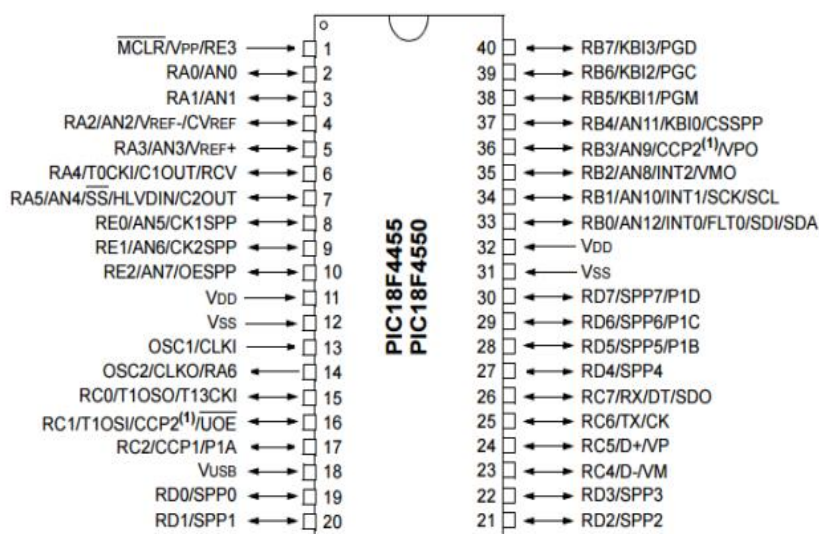
Fonte: Manual do Kit (EXSTO)

Este *kit* é baseado no PIC18F4550, mencionado anteriormente. Sua tensão de alimentação é da ordem de 4 a 5,5 V e, dada uma frequência de operação de 48MHz, ele chega a ser capaz de executar até 12 milhões de instruções por segundo (MIYADAIRA, 2009).

O condicionamento do sinal não se faz necessário, uma vez que cada pino de entrada/saída do microcontrolador possui tensão de 5V em nível lógico alto, e corrente máxima de 20 mA (valores condizentes para aplicação do sinal de controle no servomotor). Esses valores de tensão e corrente serão condicionados internamente ao *driver* como mencionado anteriormente. A seguir, a Figura 4 ilustra a pinagem do PIC18F4550.

Figura 4 – Pinagem do PIC 18F4550

## 40-Pin PDIP



Fonte: Microchip Technology (2009)

A descrição detalhada de cada pino se encontra no Quadro 1. A maioria dos pinos possuem mais de uma função, cabendo ao projetista configurar qual utilizar.

Quadro 1 – Descrição dos pinos do PIC18F4550

Pino	Nome	Tipo	Função
1	MCLR/VPP	In-In	Reset externo e programação CSP
2	RA0/AN0	I/O e Input A/D	I/O digital e entrada AD0
3	RA1/AN1	I/O e Input A/D	I/O digital e entrada AD1
4	RA2/AN2/Vref-	I/O e Input A/D	I/O digital e entrada AD2
5	RA3/AN3/Vref+	I/O e Input A/D	I/O digital e entrada AD3 e referência alta do A/D
6	RA4/T0CKI	I/O e Input A/D	I/O digital e entrada TMR0
7	RA5/AN4/SS/LVDIN	I/O e Inputs	I/O digital, entrada do A/D4, entrada do SPI e Detector de LV
8	RE0/RD/AN5	Fonte	I/O digital, Leitura da Porta Paralela e entrada do AD5
9	RE1/WR/AN6	Fonte	I/O digital, Escrita da Porta Paralela e entrada do AD6
10	RE2/CS/AN7	Fonte	I/O digital, Seleção da Porta Paralela e entrada AD7
11	VCC	Fonte	Positivo da Fonte de Alimentação

12	GND	Fonte	Negativo da Fonte de Alimentação
13	OSC1/CLK1	Input	Entrada do Cristal e entrada do Clock Externo
14	OSC2/CLK2/RA6	I/O e Inputs	I/O digital, Saída do Cristal e saída do Clock Externo
15	RC0/T10S0/T1CK1	I/O Out e In	I/O digital, saída do 2 oscilador e entrada do contador externo Timer1/Timer3
16	RC1/T10S1/CCP2	I/O In e Out	I/O digital, saída do 2 oscilador e saída do Módulo CCP2
17	RC2/CCP1	I/O e Out	I/O digital e saída do Módulo CCP1
18	RC3/SCK/SCL	I/O, I/O e I/O	I/O digital, in/out do Clock serial para modo SPI e in/out do Clock serial para modo I2C
19	RD0/PSP0	I/O e I/O	I/O digital e Porta de Comunicação Paralela
20	RD1/PSP1	I/O e I/O	I/O digital e Porta de Comunicação Paralela
21	RD2/PSP2	I/O e I/O	I/O digital e Porta de Comunicação Paralela
22	RD3/PSP3	I/O e I/O	I/O digital e Porta de Comunicação Paralela
23	RC4/SDI/SDA	I/O e I/O	I/O digital e Porta de Comunicação Paralela
24	RC5/SD0	I/O e I/O	I/O digital e Saída de Dados SP1
25	RC6/TX/CK	I/O e I/O	I/O digital, Transmissão UART e Clock de sincronismo UART
26	RC7/RX/DT	I/O e I/O	I/O digital, Recepção UART e Dados do UART
27	RD4/PS4	I/O e I/O	I/O digital e Porta de Comunicação Paralela
28	RD5/PS5	I/O e I/O	I/O digital e Porta de Comunicação Paralela
29	RD6/PS6	I/O e I/O	I/O digital e Porta de Comunicação Paralela
30	RD7/PS7	I/O e I/O	I/O digital e Porta de Comunicação Paralela
31	GND	Fonte	Negativo da Fonte de Alimentação
32	VCC	Fonte	Positivo da Fonte de Alimentação
33	RD0/INT0	I/O e In	I/O digital e entrada de Interrupção Externa 0
34	RD0/INT1	I/O e In	I/O digital e entrada de Interrupção Externa 1
35	RD0/INT2	I/O e In	I/O digital e entrada de Interrupção Externa 2
36	RB3/CCP2	I/O e I/O	I/O digital Módulo CCP2
37	RB4	I/O e In	I/O digital e entrada de Interrupção por Mudança de Estado
38	RB5/PGM	I/O e In	I/O digital, Interrupção por Mudança de Estado e Habilita ICSP baixa tensão
39	RB6/PGC	I/O e In	I/O digital, Interrupção por Mudança de Estado e ICSP in-circuit Debugger
40	RB7/PGD	I/O e In	I/O digital, Interrupção por Mudança de Estado e ICSP in-circuit Debugger

Fonte: Microchip Technology (2009)



### 3.3 Placa DAQ NI-6008

Para o processo de identificação do modelo em espaço de estados do *kit* servomecanismo utiliza-se a placa NI-6008 desenvolvida pela empresa *National Instruments*, ilustrada na Figura 5. Trata-se de um dispositivo multifuncional de baixo custo que oferece funções básicas de aquisição de dados (*Data Acquisition - DAQ*) para aplicações simples de medições portáteis e experimentos de laboratórios acadêmicos.

Figura 5 – Dispositivo DAQ NI-6008



Fonte: National Instruments (2017)

A placa é alimentada por cabo USB, possui 12 entradas/saídas digitais, 2 saídas analógicas (150 Hz), 8 entradas analógicas e um contador de 32 bits. Pode-se conectar facilmente sensores e sinais através de terminais de parafuso. Possui software que simplifica a configuração e medições, além de possuir fácil integração com MATLAB. Como os demais materiais utilizados neste trabalho, encontra-se disponível na universidade.

### 3.4 Plataforma MATLAB

MATLAB é uma ferramenta computacional utilizada no auxílio de resolução de problemas científicos e da engenharia. Possui suporte para processamento de sinais, processamento de imagens, robótica, aprendizado de máquina, construção de sistemas de controle, entre outros (MATHWORKS).

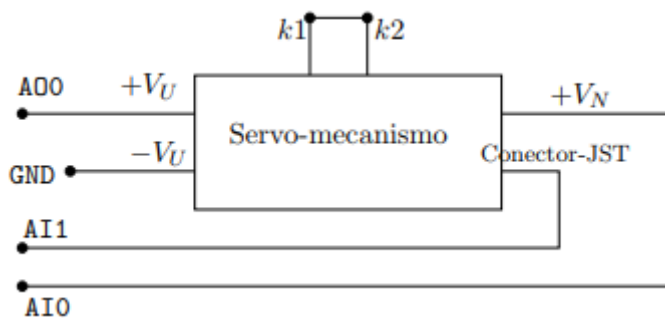
No presente trabalho, utiliza-se a plataforma na geração do modelo em espaço de estados do sistema do *kit* servomecanismo, através de um procedimento numérico de identificação. É ainda utilizada na resolução e simulação de resultados dos problemas de otimização convexa gerados a partir das abordagens de *LMI*s descritas no Capítulo 4.

#### 3.4.1 Identificação do Modelo Matemático do Kit Servomecanismo

O modelo em espaço de estados do sistema foi construído a partir da resposta a um sinal pseudo aleatório gerado pelo próprio MATLAB e enviado ao motor através da placa DAQ de aquisição de dados. O sinal em questão trabalha diferentes níveis de amplitude e frequência às variáveis de estado de forma a garantir que o modelo gerado represente o sistema com certa precisão para diferentes pontos de operação. O procedimento é descrito a seguir.

Inicialmente monta-se o circuito descrito na Figura 6.

**Figura 6 – Circuito para identificação do sistema do servomecanismo**



Fonte: Roteiro de experimento “Identificação em espaço de estados e projeto de controle de realimentação completa de estados com rastreamento” (UTFPR, 2017)

No esquemático apresentado, A00, AI0, AI1 e GND são os terminais da placa DAQ. A00 é responsável por enviar a tensão gerada pelo sinal pseudo aleatório ao terminal  $V_u$  de entrada do motor. AI0 e AI1 coletam os valores de tensão da velocidade angular e corrente JST (Conectores  $V_n$  e JST respectivamente). A tensão em  $V_n$  é diretamente proporcional à velocidade de rotação do motor, ou seja, uma tensão  $V_n$  de 1V equivale a uma velocidade de rotação de 1 radiano por segundo, de acordo com o manual fornecido pela fabricante. Dessa forma o cálculo dos ganhos serão realizados com base nos valores medidos, sem necessidade de condicionamento de sinal.

Uma vez aplicado o sinal pseudo aleatório, armazenam-se os dados de tensão  $V_n$  e de tensão no conector JST nos vetores “vecY” e “vecI” respectivamente, como expresso nos trechos de código a seguir.

```
t1 = tic;
...
for k = 1:numeroAmostras
    tensaoMedida = getsample(ai);
    vecY = [vecY tensaoMedida(1)];
    vecI = [vecI tensaoMedida(2)];
    ...
end
samprate = (numeroAmostras)/toc % average sampling rate
...
Ts = 1/samprate; %tempo de amostragem
```

Os vetores, juntamente com o tempo de amostragem e a tensão de referência fornecida como estímulo (sinal pseudo aleatório) são os parâmetros utilizados pelo comando `ssest` do MATLAB, que é responsável pela estimativa do modelo em espaço de estados do sistema, como segue:

```
x1real = vecY;
x2real = vecI;
t = [0:Ts:Ts*numeroAmostras];
data = iddata([x1real ; x2real]',vecRef',Ts);
```

```

data.InputName = {'entrada'};
data.OutputName = {'veloc','corrente'};
[modeloIni, x0] = ssest(data,2,'Ts',Ts);
A = modeloIni.A;
B = modeloIni.B;
...
Ac1 = Ts^-1 * (modelo.A - eye(size(modelo.A)))
Bc1 = Ts^-1 * modelo.B
sysc = ss(Ac1,Bc1,modelo.C,0);
figure(1)
compare(data, sysc)

```

O retorno da função *sses*t são as matrizes *A* e *B* estimadas. Como essas matrizes são calculadas considerando um sistema discretizado, realiza-se a multiplicação de ambas por um fator de correção em função do tempo de amostragem, para obter as matrizes de um sistema em tempo contínuo. O modelo é registrado na variável “*sysc*” e através do comando *compare*, gera-se um gráfico comparativo da resposta do *kit* do servomotor gravada anteriormente e a resposta simulada do sistema estimado para o mesmo sinal pseudo aleatório (vide Resultados e Discussões). O código utilizado na identificação pode ser conferido na íntegra no Anexo I.

Após a modelagem matemática do sistema, constrói-se o problema de otimização via *LMIs* para geração dos ganhos da técnica de controle desenvolvida.

### 3.4.2 Construção e Resolução das *LMIs*

Disponíveis online de forma gratuita, as bibliotecas YALMIP (LÖFBERG, 2012) e ROLMIP (AGULHARI; OLIVEIRA; PERES, 2012) são utilizadas na construção das *LMIs*, bem como na montagem do problema de otimização. A resolução do problema é feita através do *solver* SeDuMi (PÓLIK; ROMANKO; STURM, 2003), também disponível gratuitamente na internet.

As bibliotecas, em conjunto com o solver, tornam a elaboração do problema simples, intuitiva, e permitem o teste de factibilidade. Os resultados encontrados se caracterizam como os ganhos dos controladores para realimentação de estados, e do ganho integrativo.

A seguir um exemplo meramente ilustrativo da construção e resolução de um problema utilizando as ferramentas mencionadas:

```
M = sdpvar(numero_linhas, numero_colunas, 'full');
N = sdpvar(numero_linhas, numero_colunas, 'sym');
a = sdpvar(1,1);
```

No trecho de código, a função *sdpvar* cria variáveis simbólicas para o problema. O argumento da função recebe o número de linhas e colunas da matriz a ser criada, bem como o seu tipo. No exemplo, a matriz *M* é uma matriz não simétrica de dimensões  $n \times n$  ou  $n \times m$ , enquanto que a matriz *N* é uma matriz simétrica de dimensões  $n \times n$ . Para a criação de um valor escalar, basta fornecer o valor 1 para o número de linhas e colunas e omitir o argumento referente ao tipo, como é o caso da variável *a*.

A estruturação das *LMIs* pode ser feita da seguinte maneira:

```
LMIs = [-M N'; N -V] <= 0;
LMIs = [LMIs, [-a f'; f -X] <= 0];
```

Na primeira linha, a variável *LMIs* recebe a primeira desigualdade. Novas desigualdades podem ser concatenadas com as já existentes, como é o caso da segunda linha. Esse procedimento pode ser realizado até que todas as desigualdes do problema estejam gravadas na variável trabalhada.

A resolução do problema ocorre por meio da função *solvesdp*:

```
solvesdp(LMIs, funcaoObjetivo,...
         sdpsettings('solver','sedumi'));
```

Essa função recebe como argumento o conjunto de *LMIs*, a função a ser minimizada, e as configurações da função (como expresso no exemplo, deve ser feita a seleção do solver).

O teste de factibilidade é realizado através do comando *checkset*:

```
[primal,dual] = checkset(LMIs)
```

Essa função retorna os resíduos de restrição das *LMIs*. Idealmente se a variável “primal” retornada for positiva, ou um número negativo infinitesimal (da ordem de  $10^{-6}$ ), as *LMIs* compõe um problema factível.

O código para construção do problema de otimização via *LMIs* encontra-se no Anexo II.

### 3.4.3 Simulação dos controladores via Simulink

Para simulação de desempenho dos controladores gerados a partir da síntese desenvolvida utiliza-se a ferramenta *Simulink*, que é integrada ao MATLAB. Tal ferramenta permite a modelagem, simulação e análise de sistemas dinâmicos, através de um modelo de diagramação gráfica por blocos. Dentre os blocos disponíveis existem modelos prontos para espaço de estados, bem como modelos que geram atraso na transmissão de variáveis, tornando a ferramenta ideal para testes das técnicas de controle tratadas.

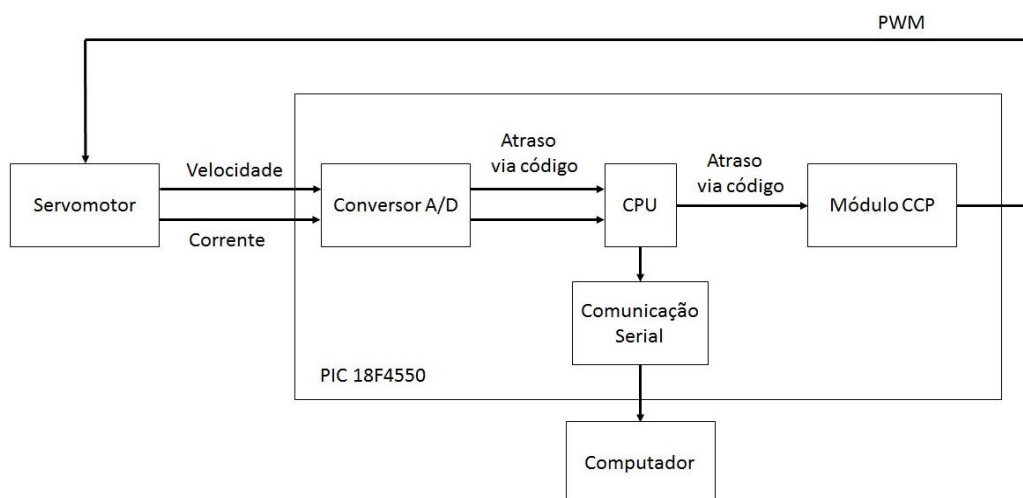
## 3.5 Funcionamento do Projeto Embarcado

Já tendo identificado o sistema servomecanismo e calculado os ganhos correspondentes através da síntese de controladores proposta (vide subseção 4.2) e das bibliotecas mencionadas (vide subseção 3.4.2), basta agora o projeto embarcado efetivar a implementação da malha fechada. Ou seja, fazer com que

uma referência seja sempre seguida pelo servomotor, independentemente da existência dos atrasos temporais nas dinâmicas dos estados e na saída de controle.

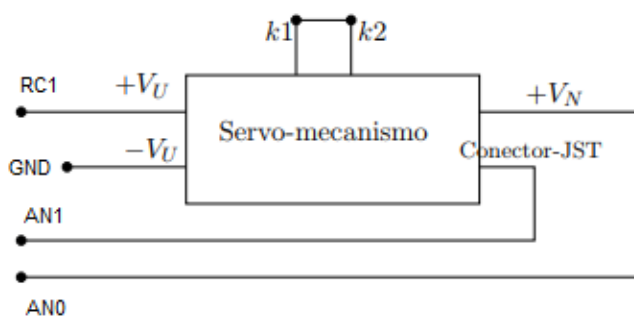
O PIC18F4550, portanto, controla a velocidade do servomotor através da lei de controle gerada pela relação entre os ganhos previamente calculados e os valores de tensão proporcionais à velocidade angular e corrente JST (conectores  $V_n$  e JST do servomecanismo, respectivamente), ambos lidos constantemente pelos canais AN0 e AN1 do PIC. O microcontrolador, sendo assim, faz as conversões A/D necessárias, desenvolve a saída de controle e a emite em um sinal PWM equivalente no canal RC1 associado ao periférico CCP2. Além dos periféricos já supracitados, a saída serial RS-232C também é utilizada para transmissão dos dados, para análises e validações posteriores. O esquemático geral de funcionamento e da montagem do circuito é dado pelas Figuras 7 e 8.

**Figura 7 – Esquemático de funcionamento**



Fonte: Autoria Própria

**Figura 8 – Esquemático do circuito**



Fonte: Autoria Própria

### 3.5.1 Estruturação do Atraso Temporal

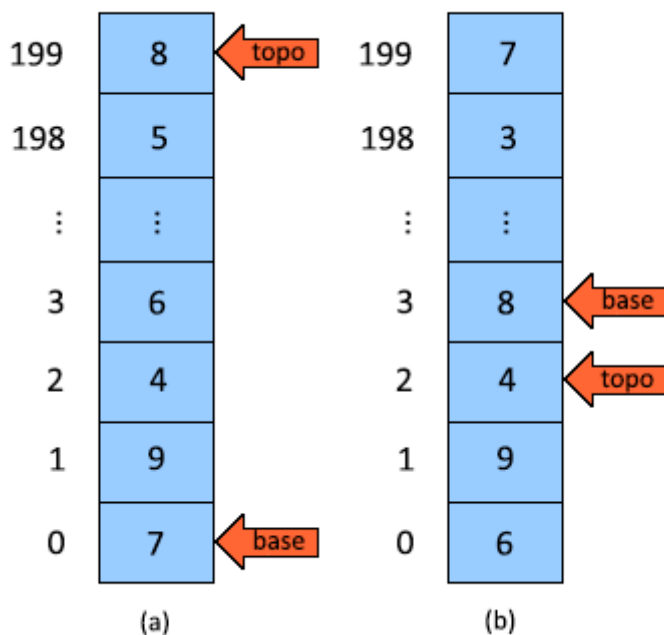
A implementação de atrasos temporais conhecidos no sistema é construída a partir de uma estrutura capaz de armazenar uma quantidade finita de dados das variáveis de estado, por um intervalo de tempo que será correspondente ao atraso. Isso é compreendido observando que a síntese desenvolvida para o controlador proposto utiliza também dos valores anteriores ao valor atual para gerar a sua saída de controle. Mais especificamente, o controlador usará constantemente  $x(t)$  e  $x(t - d)$  para a sua atuação.

Por conseguinte, implementou-se dois vetores, ambos delimitados em duzentas posições de memória, que armazenam os valores atuais e os cento e noventa e nove valores anteriores de tensão dos dois estados do sistema. A delimitação em até duzentas posições foi estabelecida devido a indisponibilidade de espaço na memória do PIC18F4550.

Tais vetores são tratados a partir do conceito de pilhas. Ou seja, os valores lidos pelo microcontrolador são ciclicamente empilhados nas posições de memória dos vetores. Portanto, ao preencher as últimas posições de memória desses vetores tem-se que esses são os valores mais atuais das variáveis de estado do sistema, isto é,  $x(t)$ . E, a primeira posição que fora preenchida, é o valor anterior mais antigo que se possui registro. Ou seja,  $x(t - d)$ . Cabe novamente ressaltar que esse processo de alocação é cíclico, logo, quando a última posição de memória é preenchida, o firmware desenvolvido direciona o próximo valor a ser alocado para a primeira posição de memória. Com isso, a segunda posição de memória passa ser o valor anterior mais antigo registrado. Denominando o valor atual como topo da pilha e o valor anterior mais antigo como base, o esquemático a seguir ilustra o conceito apresentado:



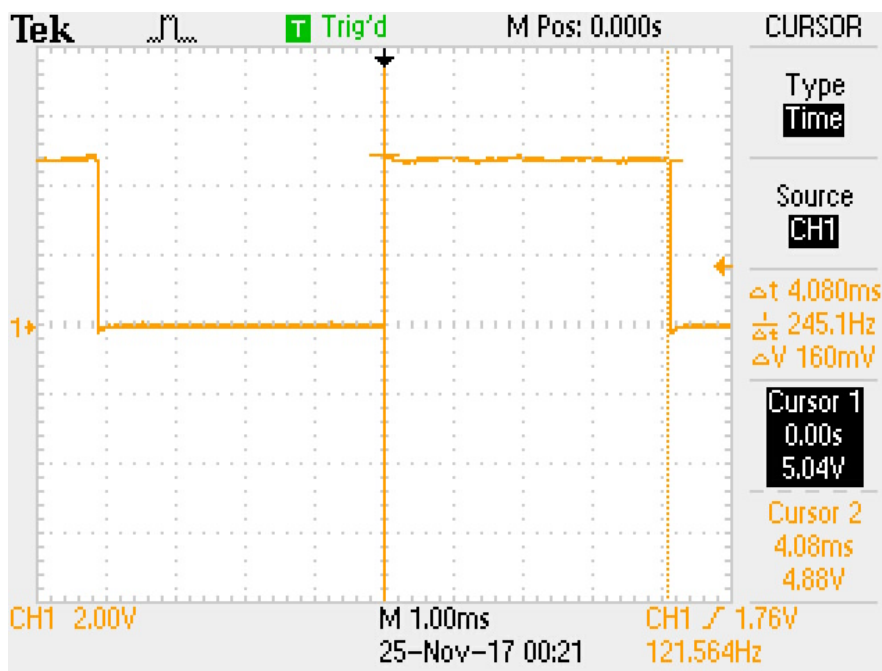
Figura 9 – Esquemático do vetor pilha



Fonte: Autoria Própria

Suprida tal necessidade de projeto, conseqüentemente, os atrasos temporais surgem pela próprio requerimento de processamento da estrutura criada, pois a criação e a atualização ininterrupta de tais vetores-pilha, somados a todas as outras linhas de código pertencentes ao ciclo 'while' do firmware, possuem um tempo de processamento específico e mensurável. Logo, entendeu-se que o atraso temporal a ser considerado é esse tempo multiplicado pelo número de posições definidas para os vetores, já que pretende-se considerar o maior intervalo de tempo possível que pode ser delimitado pela abordagem apresentada. Nesse sentido, configurou-se um canal auxiliar, o canal RB1, para que ele inverta o seu valor de saída toda vez que o ciclo 'while' reinicie. Tal canal, previamente definido como uma saída digital, foi acoplado ao osciloscópio para conferir o tempo de um ciclo 'while' com precisão. O resultado é ilustrado a seguir:

Figura 10 – Valores de RB1 medidos pelo Osciloscópio



Fonte: Autoria própria

Como foi possível constatar, um ciclo 'while' tem 4,08 milissegundos de duração. Portanto, o atraso temporal estabelecido é de 0,816 segundos. Em suma, é possível verificar o escopo das linhas de código desenvolvidas no firmware do projeto embarcado para implementação das estruturas apresentadas no Anexo III.

### 3.5.2 Aquisição de dados

Apesar do PIC em uso possuir diversos canais analógicos, só existe um único sistema de conversão. Por isso, é necessário respeitar o tempo de digitalização de cada amostra lida pelo canal analógico em uso, antes que se proceda para uma próxima leitura (seja novamente nesse mesmo canal ou em um outro). Posta a necessidade, é sabido que o tempo de digitalização das amostras é definido a partir de valores para o clock de conversão (TAD) e tempo de aquisição (TACQ), indicados no registrador ADCON2. O trecho de código a seguir representa o processo de configuração e consequente conversão A/D utilizado para o projeto embarcado.

```

...
ADCON1 = 0b00001101; // Habilita AN0 e AN1, Vref+=Vcc e
                    //Vref-=Vss
ADCON2 = 0b10010101; // Resultado alinhado à direita,
                    // TACQ = 4TAD e TAD = FOSC/16
while(1) // Loop infinito
{
    ADCON0 = 0b00000001; // Seleciona AN0
    ADCON2 = 0b10010101; // Resultado alinhado à
                        // direita,
    ADCON0bits.GO_DONE = 1; // Inicia a conversão
                        // A/D

    while (ADCON0bits.GO_DONE);
    Valor_convertor = 256 * ADRESH + ADRESL;
    // V_1bit = ( Vref+ - Vref-)/(2^10 -1) =
    //4,887585 mV
    // V_sinal_analógico = V_1bit *
Valor_convertor
    ...
}

```

### 3.5.3 Geração da Saída de Controle

A lei de controle é calculada utilizando as expressões apresentadas no início deste Capítulo e do termo de ação integral apresentado no Capítulo 2. O desenvolvimento da lei de controle pode ser conferido em mais detalhes no Capítulo 4. Os termos são relembrados a seguir:

$$u(t) = kx(t), u(t - d) = K_u x(t - d) \text{ e } I = K_i \int_0^t e(\tau) d\tau$$

Após o cálculo da lei de controle pelo firmware, o módulo CCP gera, como saída digital no canal RC1, uma onda quadrada definida pela sua frequência e seu

ciclo de trabalho (do inglês *duty cycle*): o sinal PWM. É possível verificar, a seguir, o trecho de código responsável pela formulação da solução proposta:

```
// Configuração TIMER2
// T = 1/FPWM = 1/10KHz = 100us
// PR2 = {T / [4 * TOSC * (Prescaler do TMR2)]} - 1
// Prescaler do TMR2 = 16
// TOSC = 1/FOSC = 1/20MHz = 50ns
// PR2 = 100us / (4 * 50ns * 16) - 1 = 30,25 = ~31
// BIT 7 = 0 -> Não implementado
// Bit 6-3 = 0000 -> Postscale 1:1
// Bit 2 = 1 -> Timer2 ligado
// Bit 1,0 = 11 -> Prescaler 16.
T2CON = 0b00000111;
CCPR2L = 0b00000000;
CCP2CON = 0b00011100; //inicia com velocidade zerada
...
while(1) // Loop infinito
{
    //Calculo de Duty Cycle
    dc = ((int)u_total)/50; //Normaliza de 0 a 100
    if(dc < 7) dc = 7;
    if(dc > 100) dc = 100;

    TDC = 125*dc/100; //Taxa 10kHz

    // Configurar CCP2 modo PWM com Duty Cycle
    // calculado
    // CCPRxL:CCPxCON<5:4> = 375 us/(50 ns *16)=
    // = 468,75 = ~469
    // CCPRxL:CCPxCON<5:4> = 0b0111010101
    // Os 8 bits mais significativos de ficam em CCPR2L
    // Os 2 bits menos significativos ficam em
```

```

//CCP2CON<5:4>

b8 = TDC&0b1111111100;
b1 = TDC&0b0000000010;
b0 = TDC&0b0000000001;

//Gera PWM
CCPR2L = b8/4;
CCP2CONbits.DC2B1 = b1/2;
CCP2CONbits.DC2B0 = b0;
CCP2CONbits.CCP2M3 = 1;
CCP2CONbits.CCP2M2 = 1;
CCP2CONbits.CCP2M1 = 0;
CCP2CONbits.CCP2M0 = 0;

```

### 3.5.4 Transmissão de dados

Para posteriores análises do desempenho da implementação do controlador, será necessário transmitir apenas dados numéricos simples. Mais especificamente, o microcontrolador deverá transmitir a um computador os valores de tensão da lei de controle, velocidade angular e corrente JST. Sendo assim, utilizou-se, para tal, uma comunicação serial assíncrona a partir do módulo EUSART do PIC 18F4550. Sua interface é padronizada pela RS-232C.

A configuração necessária é simples, e gira em torno da escolha do *Baud Rate Generator (BRG)*. Para o projeto em questão, optou-se pelo maior valor possível: 115200. A seguir, um trecho do código desenvolvido para esse aplicação:

```

//Configurações do canal serial
TXSTAbits.BRGH=1;
BAUDCONbits.BRG16=0;
SPBRG=10; //Configura o baud-rate para 115200

```

```

TXSTAbits.SYNC=0; //Habilita comunicação assíncrona
RCSTAbits.SPEN=1; //Habilita os pinos de I/O para a
                    //serial

//Configurações transmissor serial
TRISCbits.TRISC6=1; //Habilita o pino TX como entrada
TXSTAbits.TXEN=1
...
while(1)//loop infinito
{
    ...
    while (TXSTAbits.TRMT==0) {}
    TXREG=(0x30+(Valor_conversor/1000));
    while (TXSTAbits.TRMT==0) {}
    TXREG=(0x30+((Valor_conversor/100)%10));
    while (TXSTAbits.TRMT==0) {}
    TXREG=(0x30+((Valor_conversor/10)%10));
    while (TXSTAbits.TRMT==0) {}
    TXREG=(0x30+(Valor_conversor%10));
    while (TXSTAbits.TRMT==0) {}

    TXREG=0x0d;
    while (TXSTAbits.TRMT==0) {}
    TXREG=0x0a;
    ...
}

```

Por fim, para aquisição dos dados no computador, utilizou-se o software Serial Monitor. Ele basicamente interpreta os bits enviados e disponibiliza os valores numéricos correspondentes em um arquivo de texto padrão.

## 4 RESULTADOS E DISCUSSÕES

Esse capítulo é dividido em duas partes. Primeiramente, é apresentado o processo de desenvolvimento da técnica de síntese proposta. O modelo baseia-se em uma técnica já apresentada na literatura - vide Yu e Chu (1999) - em que são abordados o controle de um sistema incerto com atrasos em sua dinâmica, otimizando um custo garantido. Como uma extensão dessa síntese, elabora-se um problema de otimização convexa para o caso em que um sistema incerto é afetado por atrasos tanto em sua dinâmica quanto em suas entradas de controle, visando novamente uma solução ótima para o problema do custo garantido (GUEDES; CUNHA; AGULHARI, 2017). O exemplo numérico proposto por Yu e Chu no artigo em que apresentam sua síntese é resolvido, ilustrando o comparativo entre as abordagens propostas.

Em segunda instância, executam-se os passos apresentados acerca da implementação física proposta, como forma de análise de desempenho da síntese desenvolvida, no controle de um sistema real. Os resultados, bem como sua análise, estão aqui documentados.

### 4.1 Síntese para Sistemas com Atraso nos Estados

Como afirmado anteriormente, essa síntese foi apresentada por Yu e Chu em 1999. Os teoremas são descritos a seguir.

Considere inicialmente o sistema representado pela equação de estados

$$\dot{x}(t) = [A + \Delta A]x(t) + [A_1 + \Delta A_1]x(t - d) + [B + \Delta B]u(t), \quad (19)$$

$$x(t) = \varphi(t), \quad t \in (-d, 0).$$

Na equação (19), as matrizes  $A$  e  $B$  representam a dinâmica do sistema e a matriz de entrada, respectivamente. O elemento  $A_1$ , por sua vez, representa a dinâmica atrasada do sistema, enquanto que os termos  $\Delta A$ ,  $\Delta A_1$  e  $\Delta B$  são incertezas limitadas em norma presentes no modelo do sistema. Os vetores  $x(t)$ ,  $u(t)$  e  $\varphi(t)$  correspondem, respectivamente, ao vetor de estados, ao vetor de entradas de controle e à função vetorial que caracteriza as condições iniciais do sistema. Por fim, a constante  $d > 0$  representa o atraso temporal constante que afeta o sistema.

As incertezas paramétricas consideradas são definidas como

$$[\Delta A \ \Delta B \ \Delta A_1] = DF(t)[E_1 \ E_2 \ E_d], \quad (20)$$

sendo  $D, E_1, E_2$  e  $E_d$  matrizes constantes reais que representam as estruturas das incertezas. A função  $F(t)$  uma função matricial desconhecida com elementos mensuráveis, que satisfaz  $F'(t)F(t) \leq I$ , em que  $I$  é a matriz identidade de dimensões apropriadas.

Associado ao sistema (19), tem-se a função de custo (14), apresentada no Capítulo 2 e reproduzida a seguir,

$$J = \int_0^{\infty} [x'(t)Qx(t) + u'(t)Ru(t)]dt.$$

O principal objetivo do método apresentado por Yu e Chu (1999) é computar uma lei de controle  $u(t)$ , que garanta a estabilidade do sistema afetado pelas incertezas e pelo atraso, minimizando também o custo  $J$ . Tal método é brevemente apresentado nos Teoremas 1, 2 e 3.

**Teorema 1:** O controlador  $u(t) = Kx(t)$  estabiliza o sistema (19) e satisfaz o custo garantido  $J$  se existirem matrizes simétricas positivas-definidas  $P, S \in \mathbb{R}^{n \times n}$  tais que, para toda matriz de incertezas  $F(t)$  satisfazendo a condição  $F'(t)F(t) \leq I$ ,



$$\begin{bmatrix} \Sigma & P(A_1 + DF(t)E_d) \\ (A_1 + DF(t)E_d)'P & -S \end{bmatrix} < 0, \quad (21)$$

sendo

$$\begin{aligned} \Sigma = Q + K'RK + S + P[A + BK + DF(t)(E_1 + E_2K)] + \\ + [A + BK + DF(t)(E_1 + E_2K)]'P. \end{aligned} \quad (21.1)$$

**Teorema 2:** Para o sistema (19), existem matrizes simétricas positivas-definidas  $P, S$  tais que a desigualdade matricial (21) é verdadeira se, e somente se, existir um escalar  $\varepsilon > 0$ , uma matriz  $W \in R^{m \times n}$  e matrizes simétricas positivas-definidas  $X, V \in R^{n \times n}$  tais que a condição a seguir é satisfeita

$$\begin{bmatrix} \tilde{A} & A_1V & (E_1X + E_2W)' & X & W & X \\ VA_1' & -V & VE_d' & 0 & 0 & 0 \\ E_1X + E_2W & E_dV & -\varepsilon I & 0 & 0 & 0 \\ X & 0 & 0 & -Q^{-1} & 0 & 0 \\ W & 0 & 0 & 0 & -R^{-1} & 0 \\ X & 0 & 0 & 0 & 0 & -V \end{bmatrix} < 0, \quad (22)$$

sendo  $\tilde{A} = (AX + BW) + (AX + BW)' + \varepsilon DD'$ ,  $X = P^{-1}$ ,  $W = KP^{-1}$  e  $V = S^{-1}$ . A mudança de variáveis de  $K$ ,  $P$  e  $S$  é feita para que se mantenha a convexidade da condição. Se a equação (22) possuir soluções factíveis  $\varepsilon, W, X > 0, V > 0$ , então a lei de controle por realimentação de estados

$$u^*(t) = WX^{-1}x(t), \quad (23)$$

em que  $K = WX^{-1}$ , é uma lei de controle que garante um custo

$$J \leq J^* = \varphi'(0)X^{-1}\varphi(0) \int_{-d}^0 \varphi'(\tau)V^{-1}\varphi(\tau)d\tau \quad (24)$$

para o sistema incerto (19).

**Teorema 3:** O sistema (19) com função de custo (14) possui lei de controle (23) que garante a estabilidade e custo reduzido se existirem soluções factíveis  $\varepsilon$ ,  $\alpha$ ,  $W$ ,  $X$ ,  $V$  e  $M$  para o problema de otimização:

$$\min_{\varepsilon, \alpha, W, V, M} \alpha + \text{tr}(M), \quad (25)$$

sujeito a

$$(i)(22);$$

$$(ii) \begin{bmatrix} -\alpha & \varphi'(0) \\ \varphi(0) & -X \end{bmatrix} < 0;$$

$$(iii) \begin{bmatrix} -M & N' \\ N & -V \end{bmatrix} < 0;$$

em que  $\int_{-d}^0 \varphi'(\tau)\varphi(\tau)d\tau = NN'$ .

As demonstrações dos Teoremas apresentados podem ser conferidas com detalhes em Yu e Chu (1999). É importante detalhar que as demonstrações são baseadas na utilização da função de Lyapunov

$$V(x) = x'(t)Px(t) + \int_{t-d}^t x'(\tau)Sx(\tau)d\tau, \quad (26)$$

composta de um termo quadrático e da integral do atraso que afeta o sistema.

Embora o desenvolvimento apresentado nessa subseção não considere atrasos relativos à entrada de controle, essa é uma situação frequentemente

encontrada em sistemas físicos. Na próxima subsecção realiza-se a extensão desse caso, a fim de considerar tal situação.

## 4.2 Desenvolvimento de Síntese para Sistemas com Atraso nos Estados e na Entrada de Controle

A seguir, é apresentada a técnica de síntese desenvolvida e previamente publicada pelos autores do presente trabalho no Simpósio Brasileiro de Automação Inteligente (GUEDES; CUNHA; AGULHARI, 2017).

Considere agora um sistema que possui, além do atraso na dinâmica, um atraso na entrada de controle representado pela parcela  $[B_1 + \Delta B_1]u(t - d)$ , como descrito em (27)

$$\begin{aligned} \dot{x}(t) &= [A + \Delta A]x(t) + [A_1 + \Delta A_1]x(t - d) + [B + \Delta B]u(t) + [B_1 + \Delta B_1]u(t - d) \\ x(t) &= \varphi(t), \quad t \in (-d, 0), \end{aligned} \quad (27)$$

Associado também ao sistema (27), tem-se a função de custo (14). As incertezas paramétricas consideradas são dadas por

$$[\Delta A \quad \Delta B \quad \Delta A_1 \quad \Delta B_1] = DF(t)[E_1 \quad E_2 \quad E_d \quad E_u]. \quad (28)$$

O equivalente ao Teorema 1 para o sistema (27) é dado pelo Teorema 4.

**Teorema 4:** Os controladores  $u(t) = Kx(t)$  e  $u(t - d) = K_u x(t - d)$  satisfazem o custo garantido se existirem matrizes simétricas positivas-definidas  $P, S \in R^{n \times n}$  tais que, para toda matriz de incertezas  $F(t)$  satisfazendo a condição  $F'(t)F(t) \leq I$ ,

$$\begin{bmatrix} \Sigma & P[A_1 + B_1K_u + DF(t)(E_d + K_uE_u)] \\ [A_1 + B_1K_u + DF(t)(E_d + K_uE_u)]'P & -S \end{bmatrix} < 0, \quad (29)$$

sendo

$$\begin{aligned} \Sigma = Q + K'RK + S + P[A + BK + DF(t)(E_1 + E_2K)] + & (29.1) \\ + [A + BK + DF(t)(E_1 + E_2K)]'P. & \end{aligned}$$

**Prova.** Considere  $u(t) = Kx(t)$  e  $u(t - d) = K_u x(t - d)$ . A representação em malha fechada do sistema (27) é dada por

$$\begin{aligned} \dot{x}(t) = [A + BK + DF(t)(E_1 + E_2K)]x(t) + \\ + [A_1 + B_1K_u + DF(t)(E_d + E_uK_u)]x(t - d). \end{aligned} \quad (30)$$

O resultado da derivada em relação ao tempo da Função de Lyapunov (26) ao longo de qualquer trajetória do sistema de malha fechada (30) é

$$\dot{V}(x) = \dot{x}'(t)Px(t) + x'(t)P\dot{x}(t) + x'(t)Sx(t) - x'(t - d)Sx(t - d) \quad (31.1)$$

$$\begin{aligned} \dot{V}(x) = x'(t)[A + BK + DF(t)(E_1 + E_2K)]'Px(t) + \\ + x'(t)P[A + BK + DF(t)(E_1 + E_2K)]x(t) + \\ + x'(t - d)[A_1 + B_1K_u + DF(t)(E_d + E_uK_u)]'Px(t) + \\ + x'(t)P[A_1 + B_1K_u + DF(t)(E_d + E_uK_u)]x(t - d) + \\ + x'(t)Sx(t) - x'(t - d)Sx(t - d) \end{aligned} \quad (31.2)$$

$$\begin{aligned}
\dot{V}(x) &= \begin{bmatrix} x(t) \\ x(t-d) \end{bmatrix}' \begin{bmatrix} \phi \\ [A_1 + B_1K_u + DF(t)(E_d + E_uK_u)]'P \\ P[A_1 + B_1K_u + DF(t)(E_d + E_uK_u)] \\ -S \end{bmatrix} \begin{bmatrix} x(t) \\ x(t-d) \end{bmatrix} \\
&< \begin{bmatrix} x(t) \\ x(t-d) \end{bmatrix}' \begin{bmatrix} -Q - K'RK & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ x(t-d) \end{bmatrix}, \quad (31.3)
\end{aligned}$$

Sendo  $\phi = S + P[A + BK + DF(t)(E_1 + E_2K)] + [A + BK + DF(t)(E_1 + E_2K)]'P$ .

Logo

$$\begin{aligned}
\dot{V}(x) &= \begin{bmatrix} x(t) \\ x(t-d) \end{bmatrix}' \begin{bmatrix} \Sigma \\ [A_1 + B_1K_u + DF(t)(E_d + E_uK_u)]'P \\ P[A_1 + B_1K_u + DF(t)(E_d + E_uK_u)] \\ -S \end{bmatrix} \begin{bmatrix} x(t) \\ x(t-d) \end{bmatrix} < 0. \quad (31.4)
\end{aligned}$$

Assumindo a estabilidade assintótica (KHALIL, 2002) do sistema de malha fechada (27), tem-se que  $\dot{V}(x) \leq 0$ . A condição (29) implica

$$\dot{V}(x) < x'(t)(-Q - K'RK)x(t) < 0. \quad (32)$$

Segue da demonstração do Teorema 1 em Yu e Chu (1999) que a integral da equação (32) resulta em

$$\int_0^{\infty} x'(t)(Q + K'RK)x(t)dt \leq \varphi'(0)P\varphi(0) + \int_{-d}^0 \varphi'(\tau)S\varphi(\tau)d\tau. \quad (33)$$

Assim sendo,  $u(t) = Kx(t)$  e  $u(t - d) = K_u x(t - d)$  são controladores com custo garantido se existirem matrizes simétricas positivas-definidas  $P, S \in \mathbb{R}^{n \times n}$  tal que para toda matriz de incertezas  $F(t)$  satisfazendo a condição  $F'(t)F(t) \leq I$ , a equação (29) é satisfeita.

O Teorema 5 apresenta uma desigualdade matricial linear cuja solução garante a existência de  $P, S \in \mathbb{R}^{n \times n}$  que satisfaçam a condição (29), responsável também por gerar os ganhos de realimentação de estados  $K$  e  $K_u$  que estabilizam o sistema e minimizam um limitante para o custo  $J$ .

**Teorema 5:** Para o sistema (27), existem matrizes simétricas positivas-definidas  $P, S$  tais que a desigualdade matricial (29) é verdadeira se, e somente se, existirem um escalar  $\varepsilon > 0$ , matrizes  $W, Z \in \mathbb{R}^{m \times n}$  e matrizes simétricas positivas-definidas  $X, V \in \mathbb{R}^{n \times n}$  tais que a seguinte desigualdade matricial linear é satisfeita.

$$\begin{bmatrix} \tilde{A} & A_1V + B_1Z & (E_1X + E_2W)' & X & W & X \\ (A_1V + B_1Z)' & -V & (E_dV + E_uZ)' & 0 & 0 & 0 \\ E_1X + E_2W & E_dV + E_uZ & -\varepsilon I & 0 & 0 & 0 \\ X & 0 & 0 & -Q^{-1} & 0 & 0 \\ W & 0 & 0 & 0 & -R^{-1} & 0 \\ X & 0 & 0 & 0 & 0 & -V \end{bmatrix} < 0, \quad (34)$$

sendo  $\tilde{A} = (AX + BW) + (AX + BW)' + \varepsilon DD'$ ,  $X = P^{-1}$ ,  $W = KP^{-1}$ ,  $V = S^{-1}$ , e  $Z = K_u S^{-1}$ . A mudança de variáveis de  $K$ ,  $K_u$ ,  $P$  e  $S$  é feita para que se mantenha a convexidade da condição. Se a equação (34) possuir soluções factíveis  $\varepsilon, W, Z, X > 0, V > 0$ , então a lei de controle por realimentação de estados

$$u^*(t) = WX^{-1}x(t) \quad (35)$$

$$u^*(t - d) = ZV^{-1}x(t - d)$$

com  $K = WX^{-1}$  e  $K_u = ZV^{-1}$ , é uma lei de controle que garante um custo (14) para o sistema incerto (27).

**Prova.** Define-se

$$Y = \begin{bmatrix} Q + K'RK + S + P(A + BK) + (A + BK)'P & P(A_1 + B_1K_u) \\ (A_1 + B_1K_u)'P & -S \end{bmatrix} \quad (36)$$

Logo, a equação (29) equivale a

$$Y + \begin{bmatrix} PD \\ 0 \end{bmatrix} F(t) [E_1 + E_2K \quad E_d + E_uK_u] + [E_1 + E_2K \quad E_d + E_uK_u]' F'(t) \begin{bmatrix} PD \\ 0 \end{bmatrix}' < 0. \quad (36.1)$$

Pelo Lema 2.4 em Xie (1996), a desigualdade acima é satisfeita para todo  $F(t)$ , tal que  $F'(t)F(t) \leq I$  se, e somente se, existir um escalar  $\varepsilon > 0$  de modo que

$$Y + \varepsilon \begin{bmatrix} PD \\ 0 \end{bmatrix} \begin{bmatrix} PD \\ 0 \end{bmatrix}' + \varepsilon^{-1} [E_1 + E_2K \quad E_d + E_uK_u]' [E_1 + E_2K \quad E_d + E_uK_u] < 0, \quad (36.2)$$

o que equivale a

$$\begin{bmatrix} \Omega \\ (A_1 + B_1K_u)'P + \varepsilon^{-1}(E_d + E_uK_u)'(E_1 + E_2K) \\ P(A_1 + B_1K_u) + \varepsilon^{-1}(E_1 + E_2K)'(E_d + E_uK_u) \\ -S + \varepsilon^{-1}(E_d + E_uK_u)'(E_d + E_uK_u) \end{bmatrix} < 0, \quad (37)$$

Sendo

$$\begin{aligned} \Omega = Q + K'RK + S + P(A + BK) + (A + BK)'P + \\ + \varepsilon PDD'P + \varepsilon^{-1}(E_1 + E_2K)'(E_1 + E_2K). \end{aligned} \quad (37.1)$$

Pelo complemento Schur, a equação (37) pode ser reescrita como

$$\begin{bmatrix} \hat{A} & P(A_1 + B_1K_u) & (E_1 + E_2K)' & I & K & I \\ (A_1 + B_1K_u)'P & -S & (E_d + E_uK_u)' & 0 & 0 & 0 \\ (E_1 + E_2K) & E_dV + E_uZ & -\varepsilon I & 0 & 0 & 0 \\ I & 0 & 0 & -Q^{-1} & 0 & 0 \\ K & 0 & 0 & 0 & -R^{-1} & 0 \\ I & 0 & 0 & 0 & 0 & -S^{-1} \end{bmatrix} < 0, \quad (38)$$

sendo  $\hat{A} = P(A + BK) + (A + BK)'P + \varepsilon PDD'P$ . Multiplicando ambos os lados da equação (38) por

$$\begin{bmatrix} P^{-1} & 0 & 0 & 0 & 0 & 0 \\ 0 & S^{-1} & 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & 0 & I \end{bmatrix}, \quad (39)$$

e realizando a mudança de variáveis,  $X = P^{-1}$ ,  $W = KP^{-1}$ ,  $V = S^{-1}$ , e  $Z = K_uS^{-1}$ , dadas condições  $X = X' > 0$ ,  $V = V' > 0$ , obtém-se a equação (34).

Por fim, equivalente à abordagem de Yu e Chu (1999), computar uma lei de controle de custo garantido para o sistema de desigualdades matriciais lineares proposto equivale à resolução de um problema de otimização. Para o sistema (27) tal problema de otimização é dado a seguir.



**Teorema 6:** O sistema (27) com função de custo (14) possui lei de controle (35) que garante a estabilidade e custo reduzido se existirem soluções factíveis  $\varepsilon$ ,  $\alpha$ ,  $W, Z, X, V$  e  $M$  para o problema de otimização:

$$\min_{\varepsilon, \alpha, W, V, M} \alpha + tr(M), \quad (40)$$

sujeito a

$$(i) (34);$$

$$(ii) \begin{bmatrix} -\alpha & \varphi'(0) \\ \varphi(0) & -X \end{bmatrix} < 0;$$

$$(iii) \begin{bmatrix} -M & N' \\ N & -V \end{bmatrix} < 0;$$

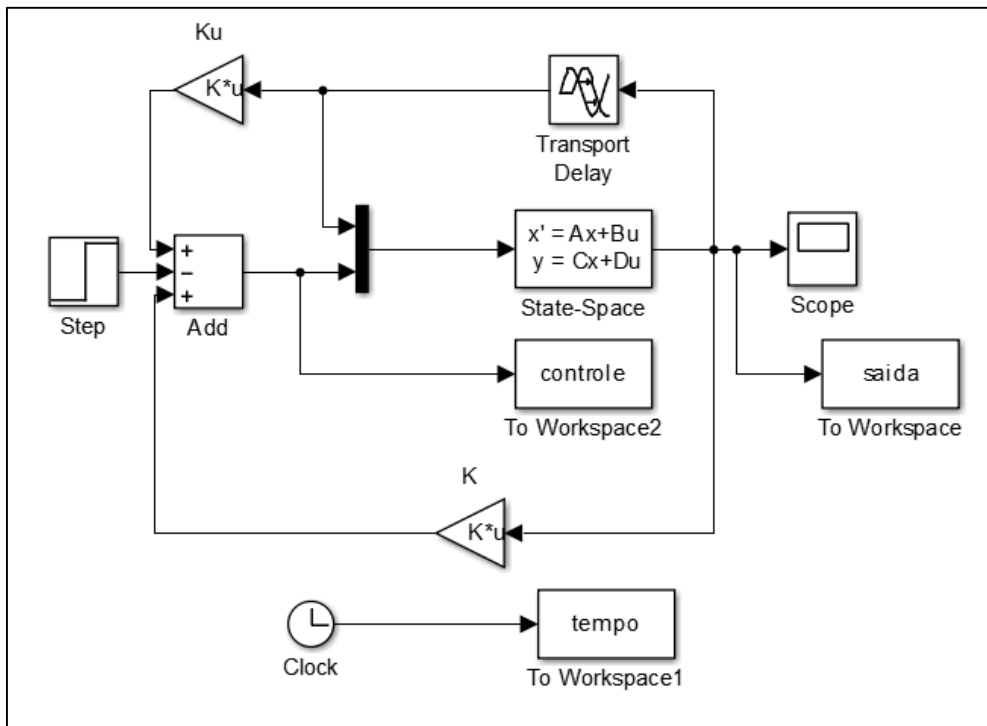
em que  $\int_{-d}^0 \varphi'(\tau)\varphi(\tau)d\tau = NN'$ .

A minimização de  $\alpha + tr(M)$  implica na minimização de um custo garantido para a lei de controle (35) para o sistema incerto (27).

### 4.3 Simulação das Sínteses Abordadas

A fim de testar a validade da síntese desenvolvida, o exemplo numérico proposto no artigo de Yu e Chu (1999) foi resolvido considerando as técnicas apresentadas nas Subseções 4.1 e 4.2. O esquemático do sistema simulado é dado pela Figura 11.

Figura 11 – Esquemático de Sistema com atraso nos estados e na entrada de controle (Simulink)



Fonte: Autoria Própria

Para a síntese projetada pelo Teorema 3, os ganhos foram calculados sem considerar os atrasos na entrada de controle. Isso permite observar como a técnica que não considera essa situação em seu desenvolvimento, se comporta quando estes se fazem presentes. O sistema proposto no artigo mencionado é descrito pela equação (19). As matrizes de dinâmica, dinâmica atrasada e a matriz de entrada, são dados por  $A$ ,  $A_1$  e  $B$  respectivamente. As matrizes  $D$ ,  $F$ ,  $E_1$ ,  $E_2$  e  $E_d$  estruturam as incertezas,  $Q$  e  $R$  são os parâmetros do LQR,  $x_1(t)$  e  $x_2(t)$  são os estados. O bloco de *step* aplica um degrau unitário na entrada do sistema.

$$A = \begin{bmatrix} 0 & 1 \\ -1 & 1 \end{bmatrix}, \quad A_1 = \begin{bmatrix} 0 & 0 \\ 0,1 & 0,1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

$$x_1(t) = e^{t+1}, \quad x_2(t) = 0, \quad t \in [-1,0] \quad \text{e} \quad Q = I, \quad R = 1,$$

$$D = \begin{bmatrix} 0,1 & 0 & 0,1 \\ 0 & 0,1 & 0 \end{bmatrix}, \quad F = 10 * \begin{bmatrix} r & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & q \end{bmatrix},$$

$$E_1 = \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad E_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad E_d = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix},$$

com  $|r| \leq 0,1$ ,  $|s| \leq 0,1$ ,  $|q| \leq 0,1$  e  $d = 1$ .

Para tal síntese, os autores obtiveram os ganhos de

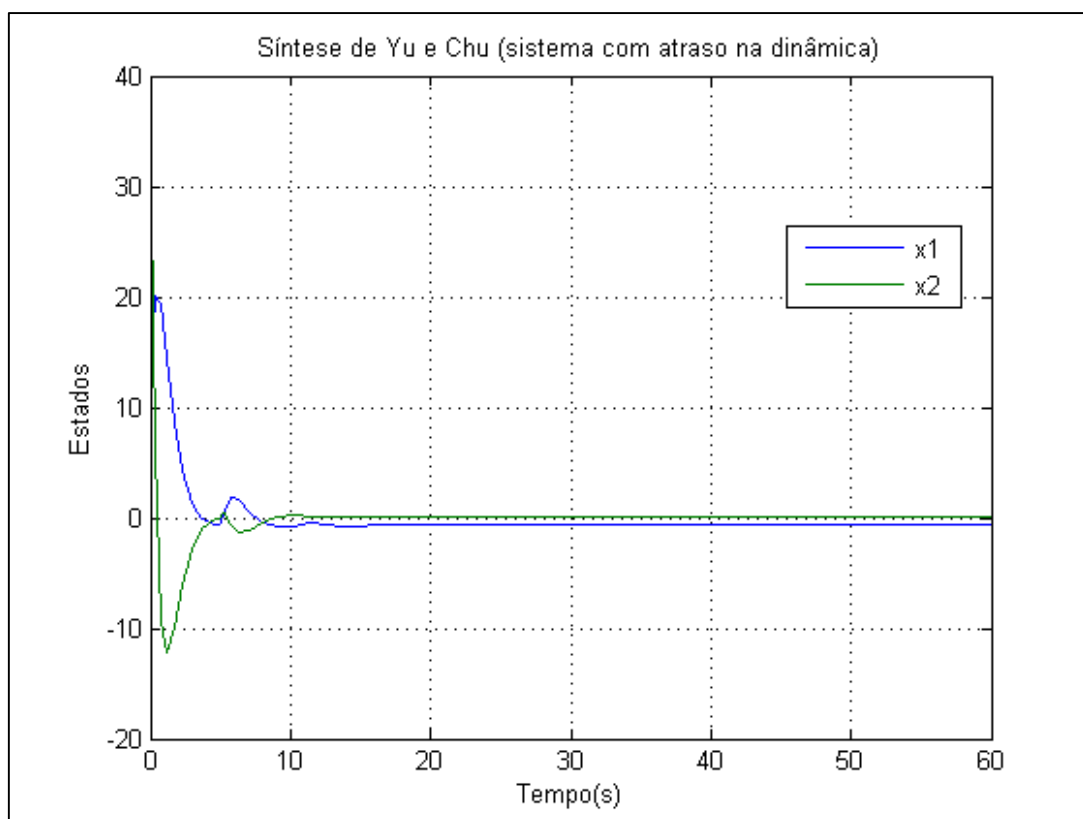
$$K = [-1,0047 \quad -3,3409],$$

com um custo garantido de

$$J = 45,4437.$$

A primeira simulação, apresenta o desempenho da técnica de síntese de Yu e Chu para a situação em que ela foi concebida: um sistema com atrasos somente em sua dinâmica. Note que para essa simulação basta igualar  $K_u = [0 \ 0]$  no esquemático da Figura 11. O resultado é ilustrado na Figura 12

**Figura 12 – Simulação do sistema exemplo com atraso na dinâmica com ganhos calculados pela síntese de Yu e Chu**

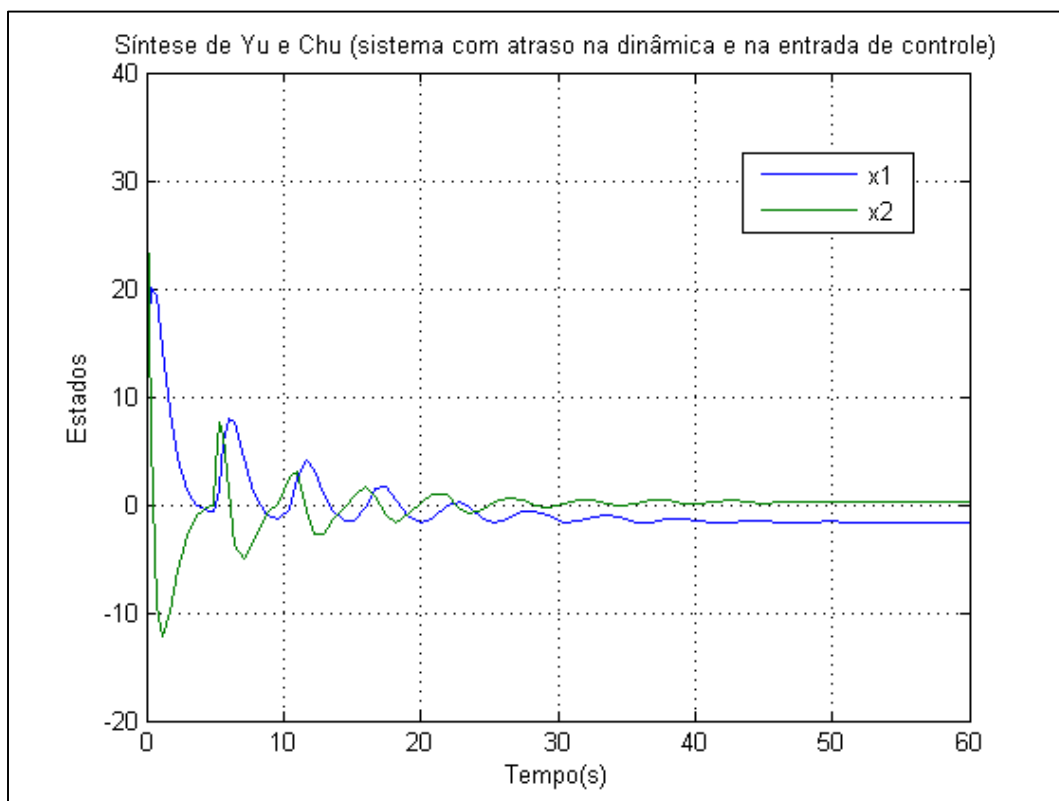


Fonte: Autoria Própria

Como esperado, a técnica realiza o controle do sistema de forma satisfatória. No entanto, a situação não se repete ao ser introduzido atraso na entrada de controle, caso apresentado a seguir.

Como a metodologia de Yu e Chu não considera tal atraso durante o cálculo dos ganhos, assume-se  $K_u = [1 \ 1]$  para que a informação atrasada aplicada à entrada de controle passe sem modificação. Simulando com os ganhos apresentados, obteve-se o resultado ilustrado pela Figura 13.

**Figura 13 – Simulação do sistema exemplo com atraso na dinâmica e na entrada de controle com ganhos calculados pela síntese de Yu e Chu**



Fonte: Autoria Própria

Nota-se que a técnica da subseção 4.1 realiza o controle do sistema, porém com desempenho insatisfatório, dado o tempo necessário para que se chegue ao regime permanente e a resposta transitória muito oscilatória.

Considere agora o controlador resultante da aplicação do Teorema 6. Como a técnica possui tratamento da parcela atrasada na entrada de controle, o sistema

considerado é da forma (27), com parâmetros idênticos ao sistema anterior. Definiu-se também a matriz de entrada atrasada  $B_1$  e a matriz de incerteza  $E_u$  como segue:

$$B_1 = \begin{bmatrix} 0 \\ 0,1 \end{bmatrix} \quad \text{e} \quad E_u = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}.$$

Obteve-se os ganhos

$$K = [-0,8950 \quad -3,1184] \text{ e}$$

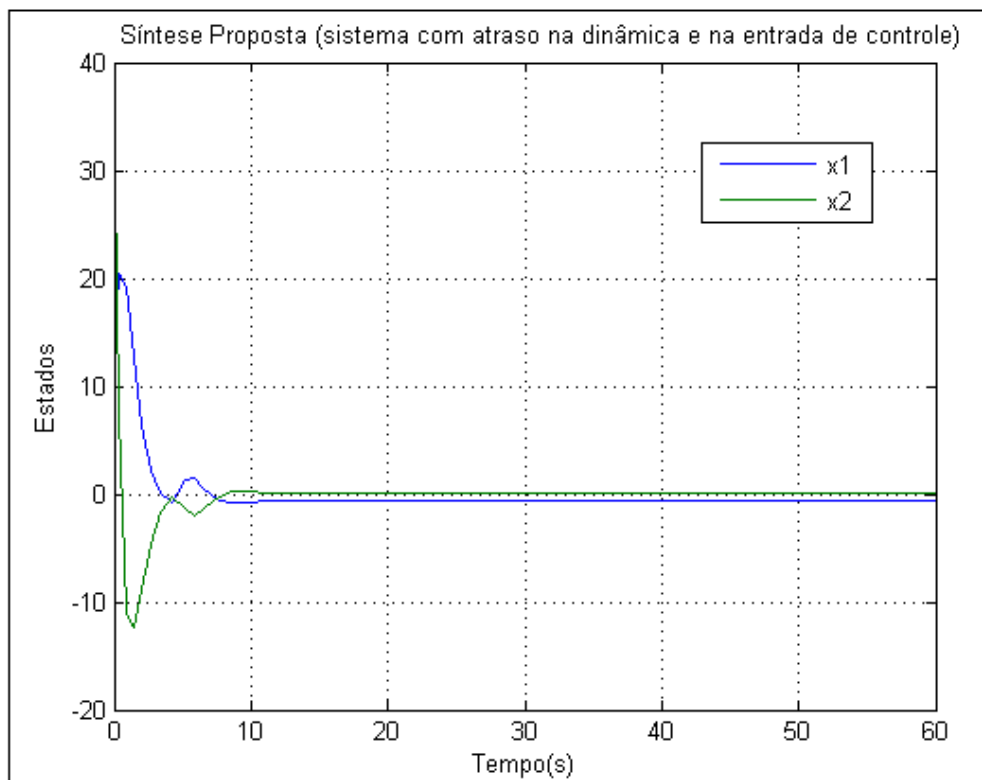
$$K_u = [-0,1350 \quad -0,1350],$$

com custo otimizado de

$$J = 42,0754$$

O resultado da simulação do esquemático da Figura 11 para os ganhos resultantes do Teorema 6 é ilustrado na Figura 14.

**Figura 14 – Simulação do sistema exemplo com atraso na dinâmica e na entrada de controle com ganhos calculados pela síntese proposta**



Fonte: Autoria Própria

A síntese da subseção 4.2 demonstra resultados melhores quando incluso o atraso na entrada de controle. A inclusão dos ganhos  $K_u$  fez com que o sistema se estabilizasse mais rápido e sem a característica oscilatória apresentada anteriormente. Observa-se também que o resultado obtido é muito semelhante à síntese de Yu e Chu para o primeiro caso simulado (atraso somente na dinâmica). Por fim, nota-se que apesar de utilizar dois controladores a técnica proposta possui um custo garantido menor que a síntese de Yu e Chu.

Através das simulações constata-se que um sistema afetado por atrasos temporais apresenta uma resposta transitória lenta para ambas as técnicas de síntese. Entretanto, é notável que a inclusão do atraso na realimentação do sistema durante a modelagem da técnica possibilita a construção de um modelo mais robusto e ideal. Os próximos tópicos são referentes à implementação física do sistema embarcado proposto.

#### 4.4 Modelo Matemático em Espaço de Estados do Servomecanismo

Para construção do modelo em espaço de estados do *kit* didático DATAPOOL 2208, seguiram-se os passos contemplados na subseção 3.4.1. Constrói-se o circuito do esquemático da Figura 6. Através do código para coleta de dados e identificação (Anexo I) implementado em MATLAB, em conjunto com a placa NI-6008, o sinal pseudo aleatório é enviado para excitação do motor e são coletados os níveis de tensão  $V_n$  e tensão no conector JST.

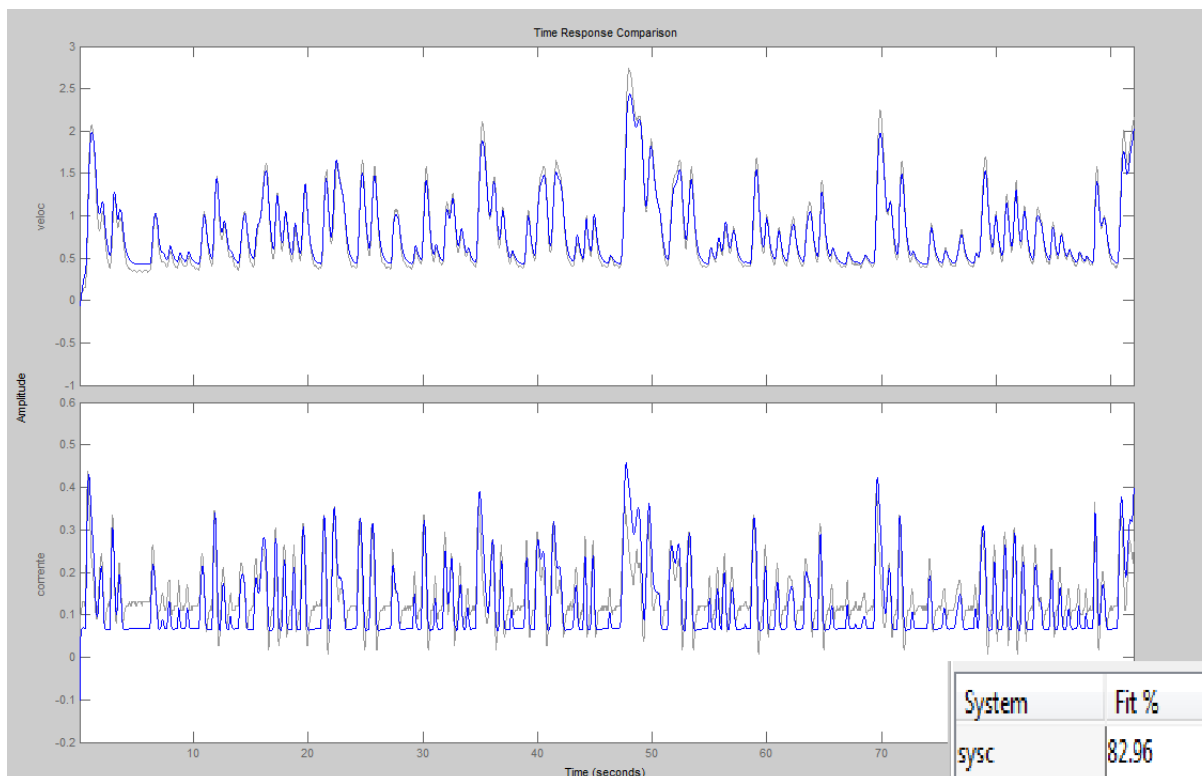
Com os dados coletados, o código estima o modelo:

$$A = \begin{bmatrix} -3,444069006348389 & 21,920454253808792 \\ -0,172299039151197 & -17,997643367284901 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 3,213380892558625 \end{bmatrix}$$

O procedimento foi repetido 5 vezes, variando os níveis de tensão do sinal pseudo aleatório entre 1V e 3V. O gráfico de comparação para a melhor resposta obtida é dado na Figura 15.

**Figura 15 – Comparação de resposta do sistema estimado (azul) com a resposta do sistema real (cinza)**



Fonte: Autoria própria

A comparação é feita com base na tensão  $V_n$ . Ao lado direito do gráfico comparativo pode-se conferir a porcentagem de semelhança entre a resposta do sistema real e o sistema estimado. A porcentagem de semelhança mais alta obtida foi de 82,96%, para uma amplitude de 2,5 V do sinal pseudo aleatório, no entanto, o estimado para variável de corrente apresenta diferenças significativas em relação ao modelo real.

#### **4.5 Geração dos Ganhos de Controle e Simulação para Modelo Matemático do Servomecanismo**

A partir do modelo estimado, eleva-se a ordem das matrizes  $A$  e  $B$  de modo que a resolução da síntese de controle proposta não só calcule os ganhos  $K$  e  $K_u$ , como também calcule o ganho do controlador integral  $K_i$  (NISE, 2004). Isso é feito

com o intuito de anular o erro em regime permanente do sistema inserindo a parcela de ação integral.

As matrizes de ordem aumentada são dadas a seguir:

$$\hat{A} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} = \begin{bmatrix} -3.444069006348389 & 21.920454253808792 & 0 \\ -0.172299039151197 & -17.997643367284901 & 0 \\ & -1 & 0 \end{bmatrix}$$

$$\hat{B} = \begin{bmatrix} B \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 3.213380892558625 \\ 0 \end{bmatrix}$$

em que  $C = [1 \ 0]$ , para que se controle apenas o primeiro estado (tensão  $V_n$ ) do sistema.

A resolução do problema de otimização com as matrizes aumentadas implica em um resultado na forma

$$K = [K_1 \ K_2 \ K_i] \text{ e}$$

$$K_u = [K_{u1} \ K_{u2} \ 0],$$

em que a última parcela do vetor de ganhos  $K$  é o ganho integral  $K_i$  e a última parcela do vetor de ganhos  $K_u$  não é implementada.

Como descrito no Capítulo 3, o máximo atraso simulado para o microcontrolador devido a limitações de memória é de 0,816s. Resolvendo o problema de otimização para este valor de atraso retorna os ganhos:

$$K = [-0,9590 \ -1,2274],$$

$$K_u = [-1,6263 \ -1,0000],$$

$$K_i = 1,5247,$$

com custo de controle

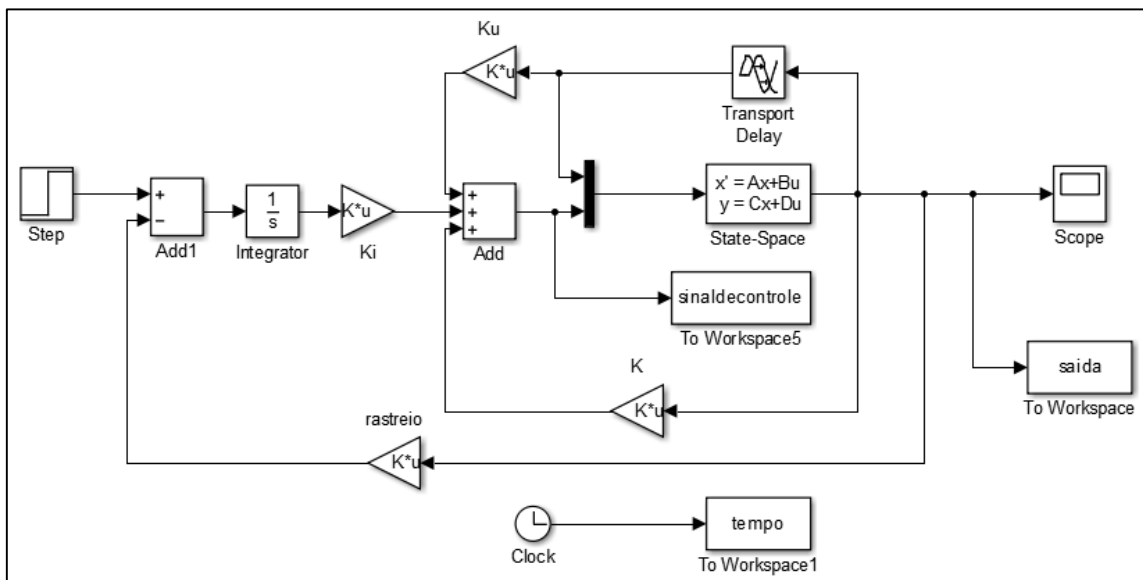
$$J = 6,2440.$$

Uma vez gerados os ganhos, utiliza-se o Simulink para simulação da resposta do sistema sob a ação de controle. Dessa forma, registra-se o



comportamento previsto para o sistema para fins de comparação de desempenho. O modelo de diagrama para simulação com integrador é apresentado na Figura 16.

Figura 16 – Diagrama de blocos com integrador (Simulink)

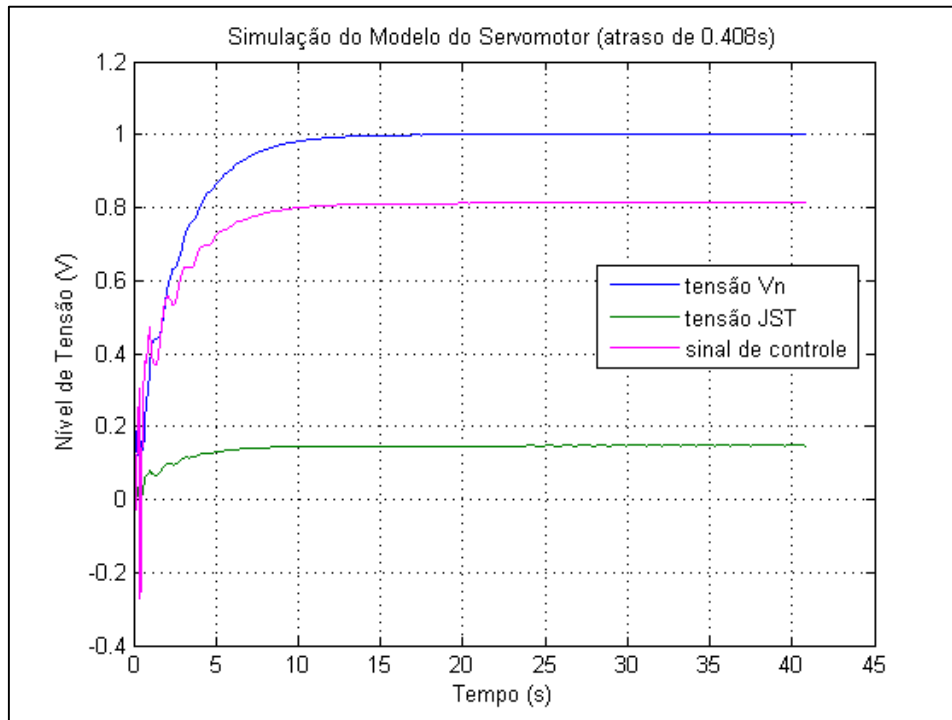


Fonte: Autoria própria

O ganho de rastreio é dado por  $[1 \ 0]$  para que a primeira variável de estado do sistema seja submetida à ação integral.

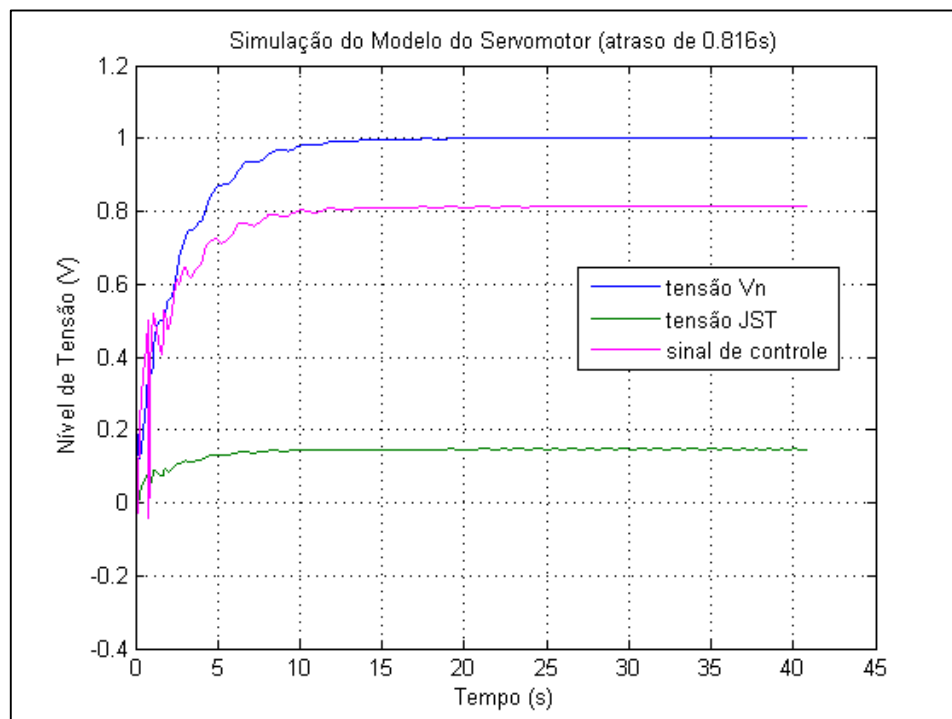
São simulados dois cenários, um em que o atraso afetando o sistema é de 0,408s e outro em que o atraso é de 0,816s, permitindo a análise da atuação dos controladores para um valor intermediário e o valor máximo de atraso definido ao se resolver o problema de otimização. A tensão de *setpoint* utilizada foi de 1V para que o *kit* apresente tensão  $V_n$  de 1V na saída, o que equivale a uma rotação de 1 rad/s. Os resultados são apresentados nas Figuras 17 e 18. O tempo total de simulação foi de 40,8s. Em azul tem-se a tensão proporcional à velocidade angular do *kit*, enquanto que em verde, tem-se a tensão gerada no conector JST, proporcional à corrente atuante no motor. Por fim, o sinal de controle é dado em roxo.

**Figura 17 – Simulação do modelo matemático do servomotor para 0,408s de atraso**



Fonte: Autoria própria

**Figura 18 – Simulação do modelo matemático do servomotor para 0,816s de atraso**



Fonte: Autoria própria

Observa-se que, em ambas as situações, o tempo da resposta transitória é longe do ideal. O tempo de subida e acomodação são ambos muito altos. Essa

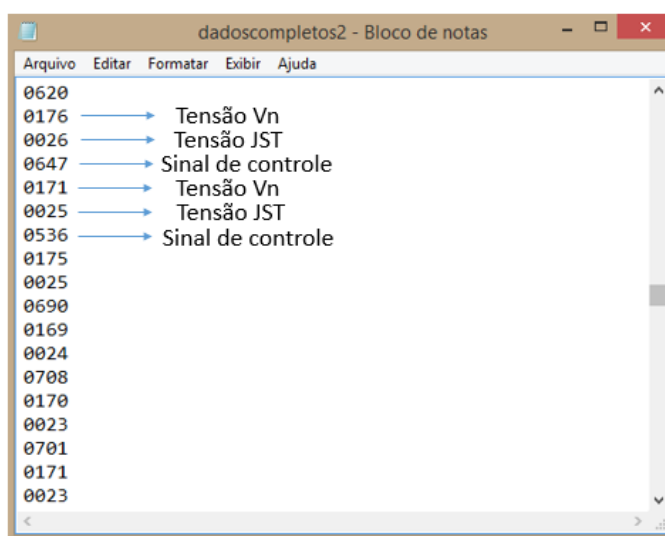
característica é inerente do resultado da aplicação dos atrasos no sistema como constatado na Subseção 4.3. Levando isso em conta, considera-se que o controlador obteve êxito.

#### 4.6 Resultados obtidos para a implementação física

Tendo em mãos os ganhos para controle do servomotor, constrói-se o sistema embarcado seguindo o procedimento descrito na subseção 3.5. Sendo assim, o microcontrolador gera o sinal de controle PWM enviado ao motor, enquanto simula um atraso temporal afetando o sistema. A coleta de dados da tensão  $V_n$  e da tensão no conector JST é feita pelo conversor A/D, e os seus valores, bem como o sinal de controle gerado, são enviados ao computador por meio da comunicação serial.

Os dados são recebidos sequencialmente na ordem: tensão  $V_n$ , tensão no conector JST, e sinal de controle e são salvos em um arquivo texto. A Figura 19 exemplifica alguns dos dados coletados.

**Figura 19 – Dados recebidos pela comunicação serial**



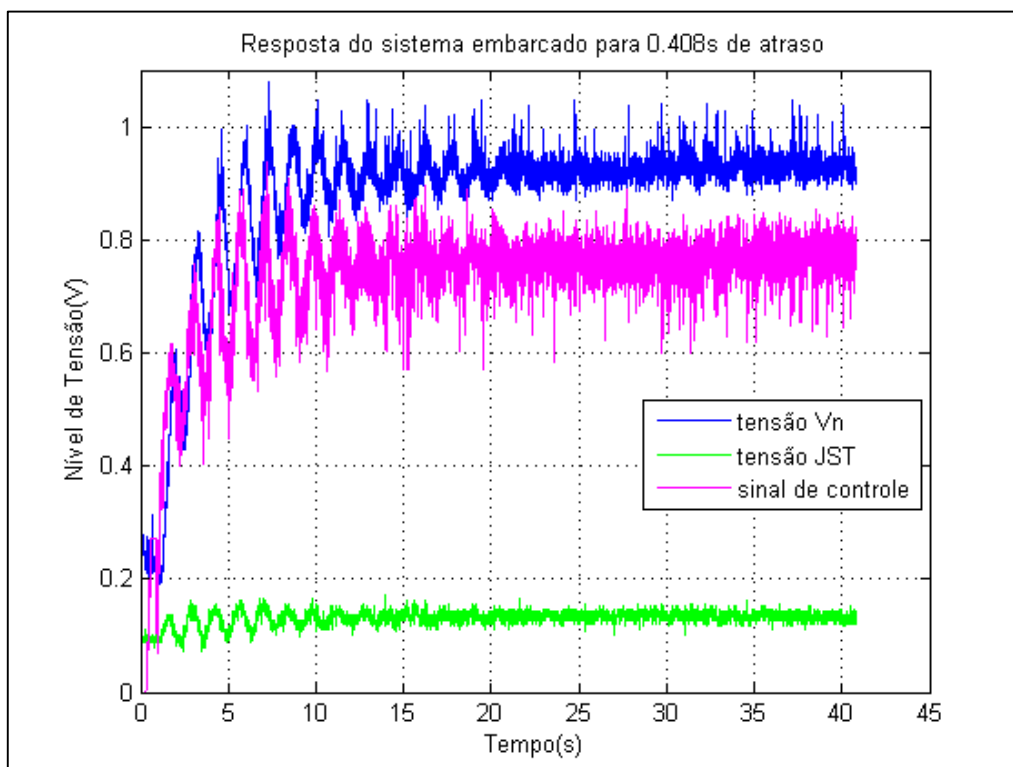
Fonte: Autoria própria

Uma vez encerrada a coleta, que envia 30000 valores em um tempo aproximado de 40,8s, efetua-se um tratamento para separação de dados (vide código do Anexo IV). Os dados de Tensões  $V_n$  e JST estão normalizados de 0 a 1024, sendo multiplicados pela resolução do conversor A/D para obter os valores entre 0 e 5V. Os dados relativos ao sinal de controle estão padronizados de 0 a 1000, sendo então divididos por 1000 para se obter seus valores de 0 a 1V.

O procedimento é realizado duas vezes, alterando o valor da pilha de coleta de dados para que o atraso inserido seja ajustado nos valores simulados anteriormente (0,408s e 0,816s).

O resultado para atraso de 0,408s pode ser conferido na Figura 20.

**Figura 20 – Resposta do sistema embarcado para 0,408s de atraso**

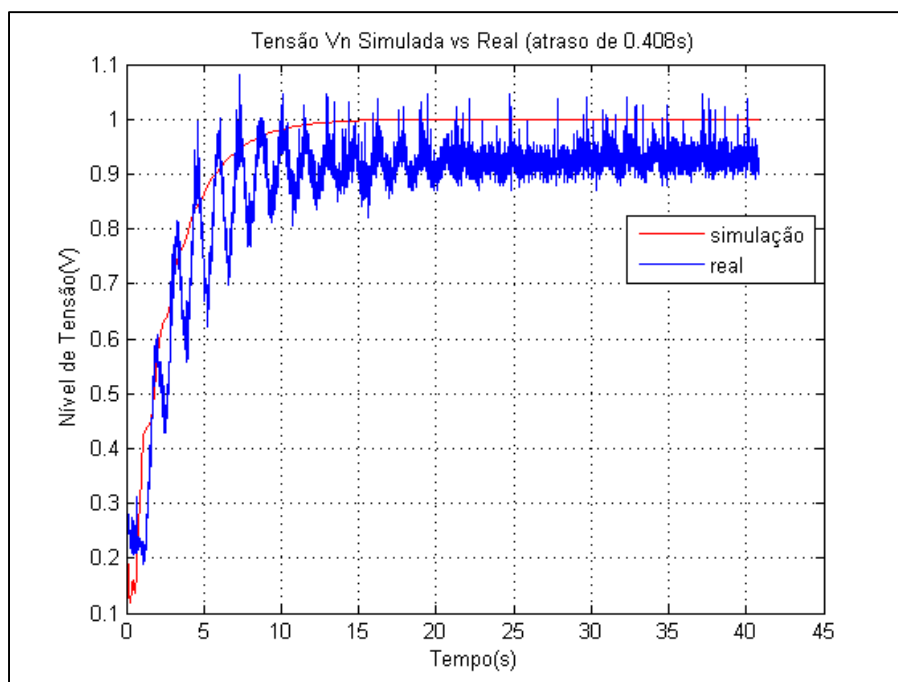


Fonte: Autoria própria

Nota-se que o servomotor não responde bem aos atrasos inseridos, e a resposta tem uma forma ruidosa e oscilada, além do já mencionado elevado tempo para atingir o valor de regime permanente. Vale notar também que o valor em regime permanente para a velocidade não segue a referência de 1V gerada. O

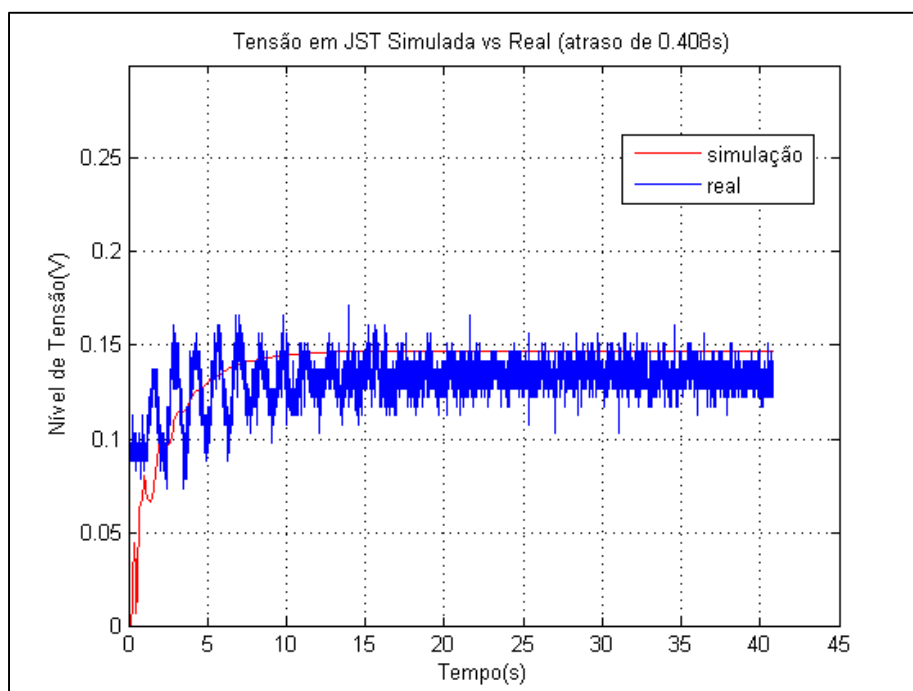
comparativo entre as tensões  $V_n$  e JST obtidas do sistema real e as tensões simuladas em ambiente Simulink para o atraso de 0,408s é dado pelas Figuras 21 e 22.

**Figura 21 – Comparação entre tensão  $V_n$  real e simulada para atraso de 0,408s**



Fonte: Autoria própria

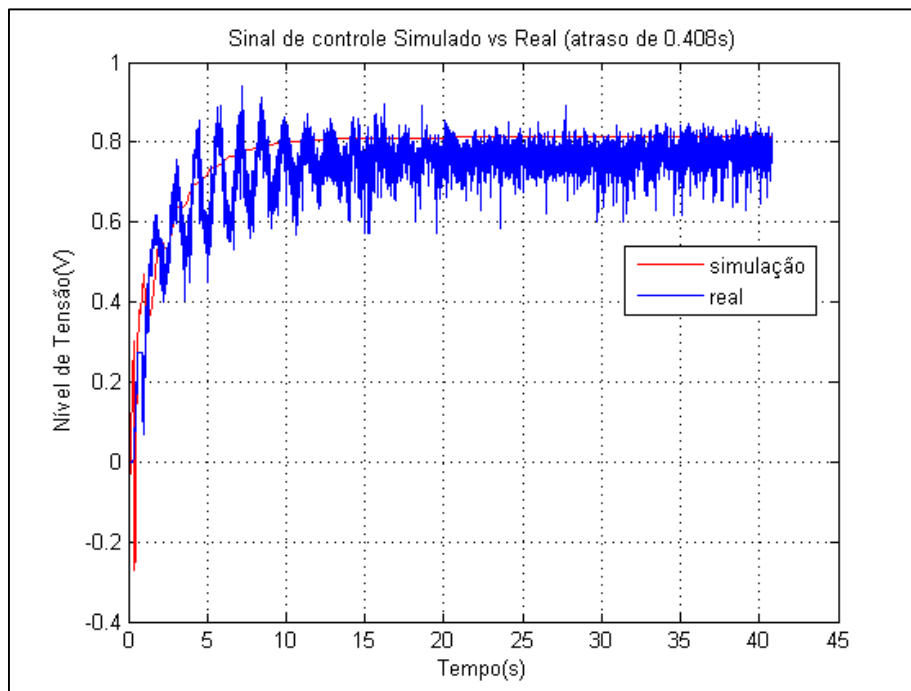
**Figura 22 – Comparação entre tensão no conector JST real e simulada para atraso de 0,408s**



Fonte: Autoria própria

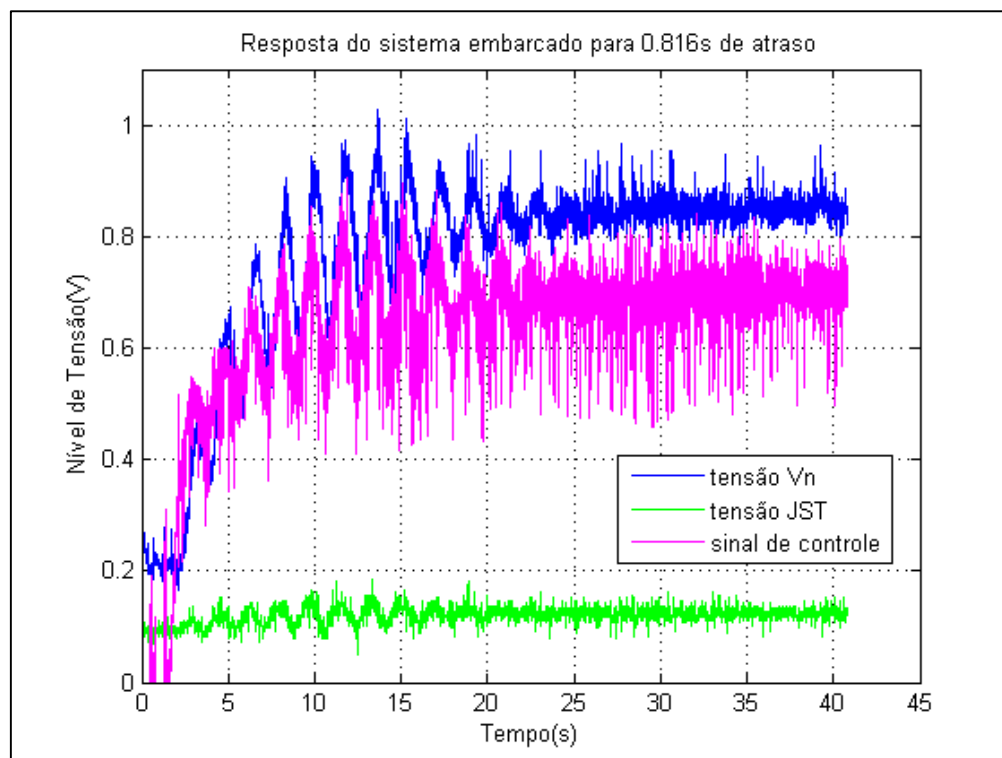
Por fim, ainda para o atraso de 0,408s, compara-se o sinal de controle emitido pelo PIC e o simulado na Figura 23.

**Figura 23 – Comparação entre sinal de controle real e simulado para atraso de 0,408s**



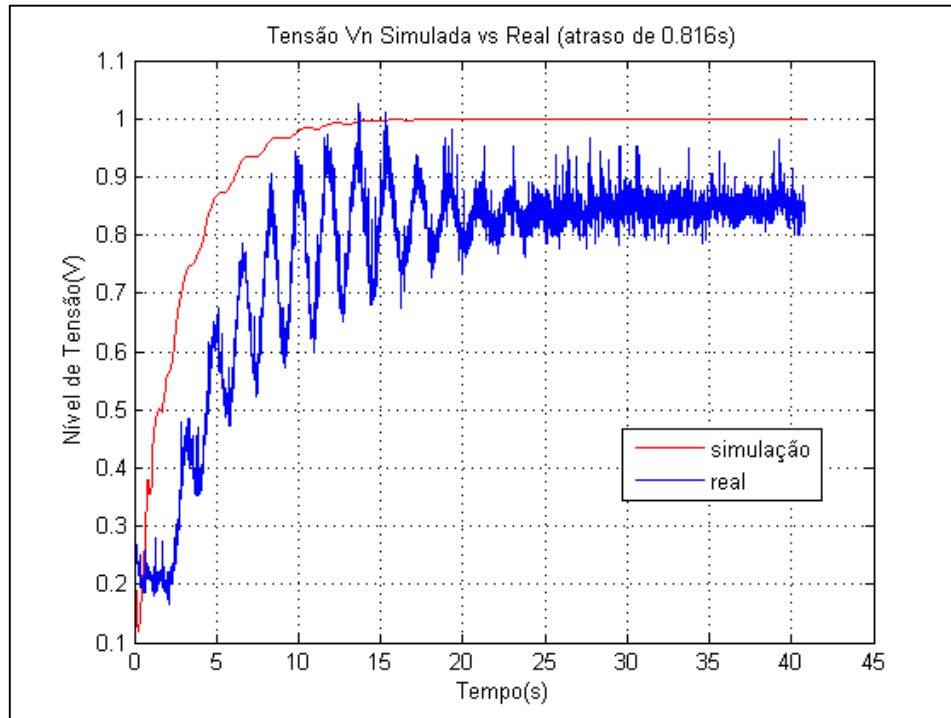
Fonte: Autoria própria

A resposta do sistema para o atraso limite de 0,816s é ilustrada na Figura 24. Novamente o comportamento da resposta é ruidoso e oscilatório, com a diferença de que dessa vez, pelo atraso inserido ser o dobro do anterior, o erro em regime permanente também é aproximadamente o dobro do erro registrado para o primeiro caso.

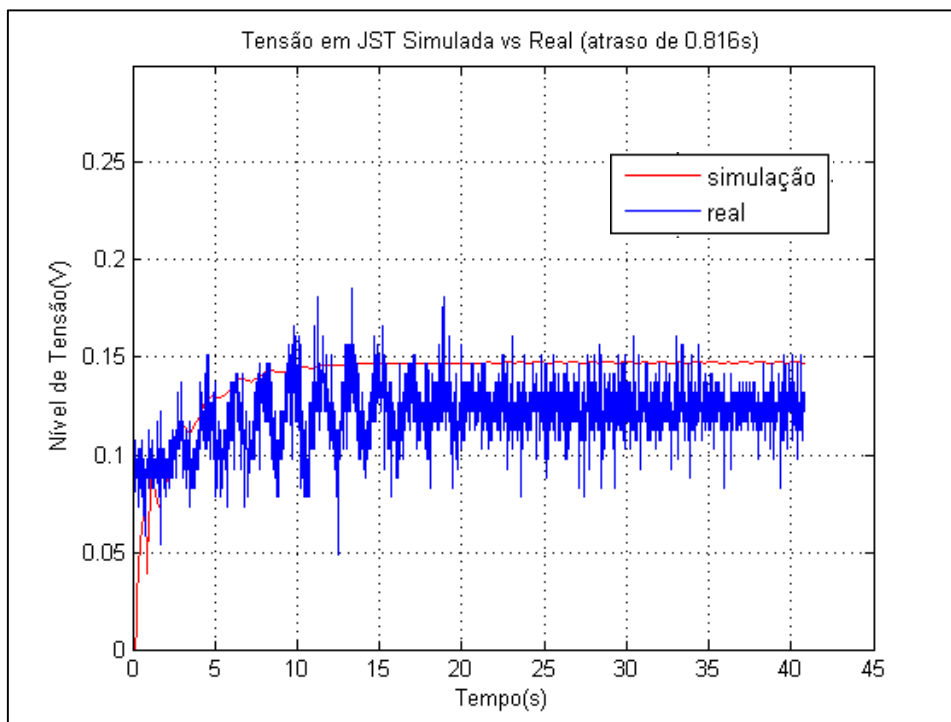
**Figura 24 – Resposta do sistema embarcado para 0,816s de atraso**

Fonte: Autoria própria

As Figuras 25, 26 e 27 apresentam os comparativos entre valores simulados e reais, para tensão  $V_n$ , tensão no conector JST e sinal de controle, respectivamente.

**Figura 25 – Comparação entre tensão  $V_n$  real e simulada para atraso de 0,816s**

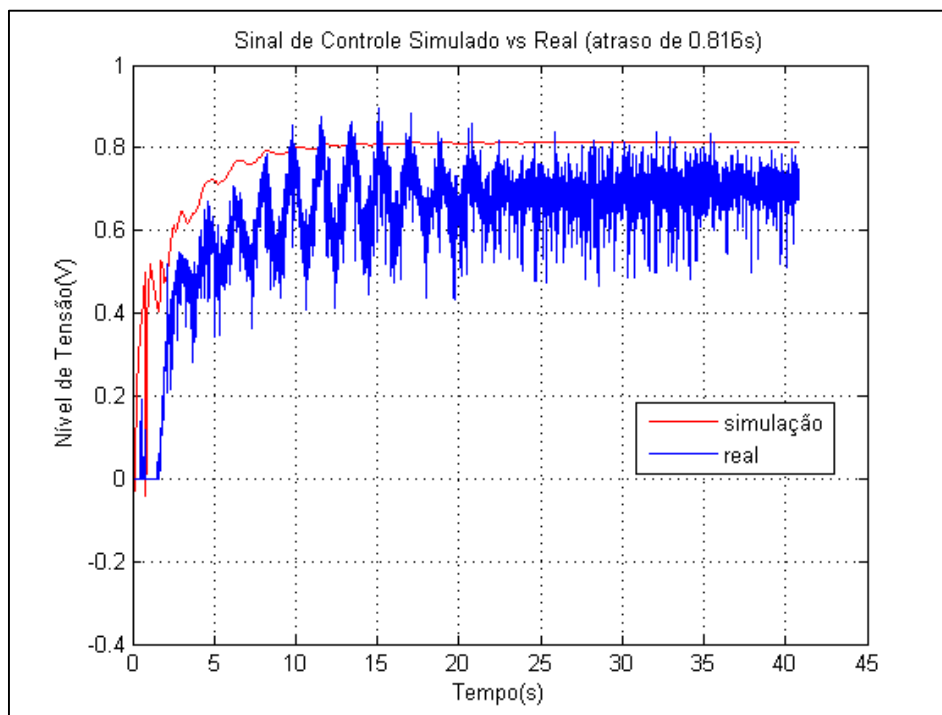
Fonte: Autoria própria

**Figura 26 – Comparação entre tensão no conector JST real e simulada para atraso de 0,816s**

Fonte: Autoria própria



**Figura 27 – Comparação entre sinal de controle real e simulado para atraso de 0,816s**



Fonte: Autoria própria

Apesar da síntese proposta ter tido um desempenho satisfatório no ambiente simulado, o mesmo não se pode dizer quando aplicada ao ambiente físico. A razão para tal discrepância pode se dar por diversos fatores. Algumas considerações acerca das possíveis causas:

- O sistema físico implementado apresentou problemas de funcionamento durante os ensaios, impactando negativamente a resposta obtida pois tais problemas não foram considerados para a síntese do controlador, tal problema pode estar presente no sensor de corrente, uma vez que o modelo estimado para a variável de corrente apresenta diferenças significativas em relação ao modelo real;
- O modelo dinâmico estimado para o sistema não é fiel ao sistema real, uma vez que a porcentagem de semelhança encontrada foi de 82,96%;
- O desempenho da técnica de controle proposta é limitado pelo desempenho do sistema físico ao qual é aplicada. Além dos problemas de funcionamento já mencionados, o servomecanismo

parece não se adequar bem à variação constante e descompassada do sinal PWM gerado;

- O sinal PWM gerado não reproduz com fidelidade o sinal de controle simulado pois seus níveis de tensão saturam entre 0 e 5V, enquanto que na simulação para 0,408s de atraso o sinal de controle tem um pico inferior a -0,2V no começo da simulação;
- A influência do atraso na dinâmica e na entrada de controle piora a resposta transitória do sistema. Esse fator, combinado com os pontos mencionados nos tópicos anteriores, pode ter contribuído para um efeito de *wind-up* da ação integral, impossibilitando o sistema de seguir a referência e gerando a forma ruidosa e oscilada das respostas medidas.

Pelo resultado das simulações e do modelo físico implementado, conclui-se que a síntese desenvolvida tem potencial para tratar sistemas afetados por atrasos tanto em sua dinâmica quanto em sua entrada de controle, porém está limitada ao desempenho que o sistema físico apresenta em tais condições.

## 5 CONSIDERAÇÕES FINAIS

No presente trabalho é elaborada uma técnica de síntese de controladores robustos para sistemas afetados por atrasos temporais, tanto em sua dinâmica, quanto em sua entrada de controle. Tal abordagem se apresenta como uma solução mais robusta a tais atrasos se comparada à síntese de Yu e Chu, a qual é o ponto de partida para o desenvolvimento deste trabalho. Isso fica constatado quando se comparam os resultados das simulações das duas técnicas aplicadas ao sistema proposto por Yu e Chu em seu trabalho, pois observa-se uma melhora no tempo de subida e acomodação do sistema, diminuição da característica oscilatória em regime permanente e, ainda assim, um menor custo associado. Todavia, denota-se que em todos os modelos tratados ao longo deste trabalho, em que há a presença de atrasos temporais na entrada de controle, o tempo de subida e acomodação em regime permanente do sistema é inerentemente elevado.

Por conseguinte, para validar o desempenho do controlador desenvolvido em sistemas físicos, aplicou-se a técnica no *kit* didático Módulo 2208 – Servomecanismo (DATAPOOL) utilizando do *kit* didático XM118 – Banco de Ensaio em Microcontroladores PIC18F (EXSTO) para elaboração da malha de controle do sistema e a implementação simulada de atrasos temporais na dinâmica e entrada de controle do sistema. Tal aplicação, apesar de possuir um comportamento minimamente coerente ao esperado, gerou um resultado ruidoso e com dificuldades para seguir o valor definido como referência, mesmo com a implementação de uma ação integral na malha de controle.

Portanto, mesmo que encontradas algumas falhas de desempenho, o presente trabalho entrega um resultado satisfatório para o que fora proposto. Algumas ressalvas importantes para posterioridades ficam divididas entre a necessidade de melhora para o elevado tempo de subida e acomodação do sistema e a dificuldade de se implementar o controlador em um sistema físico. Para as questões da implementação, acredita-se que uma potencial melhora poderá vir de uma abordagem de controle discreta, ao invés da proposta contínua apresentada.

## REFERÊNCIAS

- ADRIANO, J. D.; MARÇANO, R. de O. **XM118 - Microcontroladores PIC18 (Manual)**. Santa Rita do Sapucaí: Exsto Tecnologia, 2009.
- BOYD, S., GHAOUI, L. E., FERON, E., & BALAKRISHNAN, V. (1994). **Linear matrix inequalities in system and control theory**. Philadelphia, SIAM studies in applied mathematics, 1997. Vol. 15.
- BURTON, T. A. (1985). **Stability and periodic solutions of ordinary and functional differential equations**. New York: Academic Press.
- CHANG, S. S. L., & PENG, T. K. C. (1972). **Adaptive guaranteed cost control of systems with uncertain parameters**. IEEE Trans. Automat. Control, AC-17, p. 474-483.
- DATAPOOL ELETRÔNICA. **Sistema de Controle Servomecanismo 2208: Manual Teórico**. Itajubá: Datapool, 2006.
- GUEDES, D. R., CUNHA, H. L. O. & AGULHARI, C. M. **Síntese de controladores robusto aplicados em sistemas com atrasos temporais**. Simpósio Brasileiro de Automação Inteligente, Porto Alegre, v. 13, n. 594, s. 15C.
- KHALIL, H. K. (1992). **Nonlinear systems**. 3rd ed. Upper Saddle River, N.J.: Prentice-Hall, 2002.
- KOLMANOVSKII, V. B., NICULESCU, S. I. & RICHARD, J. P. (1999). **On the Liapunov-Krasovskii functionals for stability analysis of linear delay systems**. International Journal of Control, v. 72, n. 4, p. 374–384.
- KWAKERNAAK, H. & SIVAN, R. (1972). **Linear Optimal Control Systems**. First Edition. Wiley-Interscience, 1972.

LEITE, V. J. de S., PERES, P. L. D, CASTELAN, E. B. & TARBOURIECH, S.(2007). **Estabilidade robusta de sistemas neutrais com atrasos variantes no tempo.** Sba Controle & Automação, Natal, v. 18, n. 4, p. 434-446.

LIU, G. (2016). **New results on stability analysis of singular time-delay systems,** International Journal of Systems Science, DOI: 10.1080/00207721.2016.1258598.

MIRANDA, M. F. & LEITE, V. J. S (2008). **Síntese convexa para sistemas incertos discretos no tempo com atrasos variantes.** Sba Controle & Automação, Natal, v.19, n. 3, p. 242-255.

MUKAIDANI, H. (2003). **An LMI approach to guaranteed cost control for uncertain delay systems.** IEEE Trans. Circuits and Systems – I: Fundamental Theory and Application, v. 50, n. 6, p. 795-800.

NISE, N. S. (2004). **Control Systems Engineering,** 4. ed., John Wilhey and Sons, 2004.

OLIVEIRA NETTO, A. A. de. **Metodologia da pesquisa científica:** guia prático para a apresentação de trabalhos acadêmicos. 3. ed. rev. e atual. Florianópolis: Visual Books, 2008.

PEREIRA, F. (2010). **Microcontrolador PIC18 detalhado: Hardware e Software.** 1.Ed, São Paulo, Editora Érica, 2010.

RICHARD, J. P. (2003). **Time-delay systems: an overview of some recent advances and open problems.** Automatica, v. 39, p. 1667-1694.

SCHERER, C. & SIEP, W. (2005). **Linear Matrix Inequalities in Control.** Netherlands, Eindhoven University of Technology, 2005.

SIMEÃO, J. D. **Controle de Sistemas com Atrasos nos Estados: uma abordagem convexa**. 2009. 105f. Dissertação de Mestrado - Centro Federal de Educação Tecnológica de Minas Gerais, Belo Horizonte, 2009.

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ. Sistema de Bibliotecas. **Normas para elaboração de trabalhos acadêmicos**. Curitiba: UTFPR, 2009.

XIE, L. (1996). **Output feedback  $H_{\infty}$  control of systems with parameter uncertainty**. Int. J. Control, v. 63, p. 741-750.

XU, S. & LAM, J. (2008). **A survey of linear matrix inequality techniques in stability analysis of delay systems**. International Journal of Systems Science, v. 39, n. 12, p.1095-1113.

YU, L. & CHU, J (1999). **An LMI approach to guaranteed cost control of linear uncertain time-delay systems**. Automatica, v. 35, p. 1155-1159.

**ANEXO I – Código para identificação do modelo em Espaço de Estados do  
Servomotor**

## Código para identificação do modelo em Espaço de Estados do Servomotor

```

clear all, close all, clc,

ai = analoginput('nidaq','Dev1');
ao = analogoutput('nidaq','Dev1');

addchannel(ai,[0 1]);
set(ai,'InputType','SingleEnded');
addchannel(ao,0);
get(ai) % mostra todas as propriedades do objeto 'ai'
Nit=2000; %numero de amostras
uPRBS=idinput(Nit,'PRBS');
f=58.1;% frequência de amostra
f_cutoff = 04; % frequência de corte
fnorm =f_cutoff/(f/2); % frequência de corte normalizada
[b1,a1] = butter(5,fnorm,'low'); % Low pass Butterworth
                                     %filter of order 10
uPRBS_Filtered = filtfilt(b1,a1,uPRBS); % filtragem
uPRBS = uPRBS_Filtered;
figure(1);clf; h=plot(nan,nan,'b-');
grid,xlim([0 (1.25)*Nit]),ylim([0 2]),

tic; % inicio do timer para calcular a taxa de amostragem
Rep=1;
t1 = tic;
vecY=[];vecU=[];vecRef=[];vecI=[];
Amp=2.5; dc=0.4;
for k =1:Nit
    tensaoEntr= getsample(ai);
    vecY=[vecY tensaoEntr(1)];
    vecI=[vecI tensaoEntr(2)];
    Vout=Amp*max([uPRBS(k) 0]) + dc;
    vecRef=[vecRef Vout];
    putsample(ao , Vout);
    pause (0.000001);
    set(h,'xdata',[1:k],'ydata',[vecY([1:k])]);
end
samprate = (Nit)/toc % calculo da taxa de amostragem média

putsample(ao,0); pause(0.2); wait(ai,0.02); delete(ai);
clear ai; putsample(ao,0); delete(ao); clear ao; shg;
Ts=1/samprate;
save -v7 identkit.mat %salva os dados experimentais

xlreal = vecY;
x2real = vecI;
t = [0:Ts:Ts*Nit];
data = iddata([xlreal ; x2real'],'vecRef',Ts);
data.InputName = {'entrada'};

```



```

data.OutputName = {'veloc','corrente'};
[modeloIni, x0] = ssest(data,2,'Ts',Ts);
A = modeloIni.A;
B = modeloIni.B;
B(1) = 0;
C = [1 0;
     0 1];
D = [0];
K = 0 * modeloIni.K;
x0 = [0 0]';
init_sys = idss(A,B,C,D,K,x0,Ts);
init_sys.Structure.b.Free(1,1) = false;
init_sys.Structure.c.Free = false;

[modelo, x0] = ssest(data,init_sys,'Ts',Ts);
[yout, fit, x0] = compare(data,modelo);

modelo.K = modelo.K * 0;
disp(' ===== Matrices A e B encontradas ====='),
Ac1 = Ts^-1 * (modelo.A-eye(size(modelo.A)))
Bc1 = Ts^-1 * modelo.B
sysc = ss(Ac1,Bc1,modelo.C,0) ;
figure(1)
compare(data, sysc)

```

**ANEXO II – Código para síntese de ganho dos controladores**

### Código para síntese de ganho dos controladores

```

A = [-3.444069006348389  21.920454253808792;
     -0.172299039151197 -17.997643367284901];

%se não usar integradores considerar a matriz A normal
%e arrumar as dimensões das matrizes de incertezas
A = [A zeros(2,1);
     -1 0 0];

B = [0;
     3.213380892558625];

%se não usar integradores considerar a matriz B normal
%e arrumar as dimensões das matrizes de incertezas
B = [B;
     0];

A1 = [0.1 0.1 0; 0.1 0.1 0; 0 0 0];
B1 = [0.1; 0.1; 0];

t = -0.816:0.0001:0;

x1 = 0;
x2 = exp(-3.7115*t) + exp(-0.082*t);

f = [x1; x2(4001); 0];

%fi X fi'
H = [0 0 0;
     0 trapz(t,x2.*x2) 0;
     0 0 0
     ];

r = 0.1;
s = 0.1;
q = 0.1;

% r = -0.1;
% s = -0.1;
% q = -0.1;

F = 10*[r 0 0; 0 s 0; 0 0 q];

%Q, R e matrizes de incertezas aqui ilustradas foram
%as utilizadas para ensaio do sistema embarcado
Q = eye(3);
R = 0.5;

D = [0.1 0 0; 0.1 0 0; 0 0 0];

```

```

E1 = [1 0 0; 1 0 0; 0 0 0];
E2 = [0; 0; 0];
Ed = [1 0 0; 1 0 0; 0 0 0];
Eu = [0; 0; 0];

%exibe matrizes com as incertezas para usar no simulink
Av = A + D*F*E1
Bv = B + D*F*E2

colunaA1 = size(A1,2);
colunaE1 = size(E1,2);
colunaE2 = size(E2,2);

xi = sdpvar(1,1);
a = sdpvar(1,1);
W = sdpvar(colunaE2,colunaE1,'full');
M = sdpvar(colunaA1,colunaA1,'sym');
V = sdpvar(colunaA1,colunaA1,'sym');
X = sdpvar(colunaE1,colunaE1,'sym');
Z = [sdpvar(1, (colunaA1-1), 'full') 0];

T11 = A*X + B*W + (A*X + B*W)' + xi*(D*D');
T12 = A1*V+B1*Z;
T13 = (E1*X + E2*W)';
T14 = X;
T15 = W';
T16 = X;
T21 = T12';
T22 = -V;
T23 = V*Ed'+Z'*Eu';
T24 = zeros(3);
T25 = zeros(3,1);
T26 = zeros(3);
T31 = T13';
T32 = T23';
T33 = -xi*eye(3);
T34 = zeros(3);
T35 = zeros(3,1);
T36 = zeros(3);
T41 = X;
T42 = zeros(3);
T43 = zeros(3);
T44 = -inv(Q);
T45 = zeros(3,1);
T46 = zeros(3);
T51 = W;
T52 = zeros(1,3);
T53 = zeros(1,3);
T54 = zeros(1,3);
T55 = -inv(R);
T56 = zeros(1,3);

```

```

T61 = X;
T62 = zeros(3);
T63 = zeros(3);
T64 = zeros(3);
T65 = zeros(3,1);
T66 = -V;

%LMI 1
LMIs = [T11 T12 T13 T14 T15 T16;
        T21 T22 T23 T24 T25 T26;
        T31 T32 T33 T34 T35 T36;
        T41 T42 T43 T44 T45 T46;
        T51 T52 T53 T54 T55 T56;
        T61 T62 T63 T64 T65 T66] <=0;

%LMI 2
LMIs = [LMIs, [-a f'; f -X] <= 0];

%N
N = sqrtm(H);

%LMI 3
LMIs = [LMIs, [-M N'; N -V] <= 0];

solvesdp(LMIs, a+trace(M), sdpsettings('solver','sedumi'));

[prim,dual] = checkset(LMIs)
if(min(prim) >-1e-6)
    %Factivel
    K = double(W)*inv(double(X))
    Ku = double(Z)*inv(double(V))
    J = double(a)+double(trace(M))
else
    %Infactivel
    infactivel = 1
end

```

**ANEXO III – Código do firmware**

## Código do firmware

```

#include <P18F4550.h>
#include <delays.h>

#pragma config FOSC = HS // Fosc = 20MHz; Tcy = 200ns
#pragma config CPUDIV = OSC1_PLL2 // OSC/1 com PLL off
#pragma config WDT = OFF // Watchdog desativado
#pragma config LVP = OFF // Desabilita gravação em baixa
#pragma config DEBUG = ON // Habilita debug
#pragma config MCLRE = ON // Habilita MCLR
#pragma config CCP2MX = ON // Pino RC1 utilizado em CCP2

void HighPriorityISR(void);

// Interrupção de alta prioridade
#pragma code high_vector=0x08
void interrupt_at_high_vector(void)
{
    _asm GOTO HighPriorityISR _endasm
}
#pragma code

short long u_atual = 0, u_atrasado = 0, u_total = 0, aux;
int dc = 0;
int b8, b1, b0, i = 0, cont = 0;
int TDC;
int vel[200], jst[200];
int erro;
int intg = 0;

void HighPriorityISR(void)
{
}

void LowPriorityISR(void)
{
}

void main(void)
{
    // Variáveis para armazenar o resultado da conversão A/D
    unsigned int Valor_convertor = 0;
    unsigned int Valor_convertor2 = 0;

    // Configurações dos pinos de I/O
    TRISAbits.TRISA0 = 1; // AN0 é entrada
    TRISAbits.TRISA1 = 1; // AN1 é entrada
    TRISBbits.TRISB1 = 0; // RB1 é saída
    TRISCbits.TRISC1 = 0; // RC1 é saída

```

```

//Configurações do canal serial
TXSTAbits.BRGH=1;
BAUDCONbits.BRG16=0;
SPBRG=10;           //Configura o baud-rate para 115200
TXSTAbits.SYNC=0;   //Habilita comunicação assíncrona
RCSTAbits.SPEN=1;   //Habilita os pinos de I/O para a
                    //serial

//Configurações transmissor serial
TRISCbits.TRISC6=1; //Habilita o pino TX como entrada
TXSTAbits.TXEN=1;
ADCON1 = 0b00001101; // Habilita AN0 e AN1, Vref+=Vcc e
                    //Vref-=Vss
ADCON0 = 0b00000001; // Seleciona AN0 e liga módulo A/D
ADCON2 = 0b10010101; // Resultado alinhado à direita,
Delay10TCYx(5);      // Delay de 50 ciclos de máquina
                    // TACQ = 4TAD e TAD = FOSC/16

//Habilita interrupção
RCONbits.IPEN = 1; // Habilita prioridades de
                    //interrupção
INTCONbits.GIEL = 1; // Habilita interrupção de baixa
                    //prior.
INTCONbits.GIEH = 1; // Habilita interrupção de alta
                    //prior.

// Configuração TIMER2
// T = 1/FPWM = 1/10KHz = 100us
// PR2 = {T / [4 * TOSC * (Prescaler do TMR2)]} - 1
// Prescaler do TMR2 = 16
// TOSC = 1/FOSC = 1/20MHz = 50ns
// PR2 = 100us / (4 * 50ns * 16) - 1 = 30,25 = ~31
PR2 = 31; // taxa de 10kHz
// BIT 7 = 0 -> Não implementado
// Bit 6-3 = 0000 -> Postscale 1:1
// Bit 2 = 1 -> Timer2 ligado
// Bit 1,0 = 11 -> Prescaler 16.
T2CON = 0b00000111;
CCPR2L = 0b00000000;
CCP2CON = 0b00011100; //inicia com velocidade zerada

//Zera vetores de coleta de dados
for(i=0;i<200;i++)
{
    vel[i] = 0;
    jst[i] = 0;
}

Delay10KTCYx(1000);

```



```

i = 0;

while(1) // Loop infinito
{
    PORTBbits.RB1 = ~PORTBbits.RB1;

    ADCON0 = 0b00000001; // Seleciona AN0
    ADCON2 = 0b10010101; // Resultado alinhado à
                        //direita,
    ADCON0bits.GO_DONE = 1; // Inicia a conversão A/D
    while (ADCON0bits.GO_DONE);
    Valor_convertor = 256 * ADRESH + ADRESL;
    // V_1bit = ( Vref+ - Vref -)/(2^10 -1) = 4,887585
    //mV
    // V_sinal_analógico = V_1bit * Valor_convertor

    if(cont < 10000)
    {
        while (TXSTAbits.TRMT==0) {}
        TXREG=(0x30+(Valor_convertor/1000));
        while (TXSTAbits.TRMT==0) {}
        TXREG=(0x30+((Valor_convertor/100)%10));
        while (TXSTAbits.TRMT==0) {}
        TXREG=(0x30+((Valor_convertor/10)%10));
        while (TXSTAbits.TRMT==0) {}
        TXREG=(0x30+(Valor_convertor%10));
        while (TXSTAbits.TRMT==0) {}

        TXREG=0x0d;
        while (TXSTAbits.TRMT==0) {}
        TXREG=0x0a;
    }

    ADCON0 = 0b00000101; // Seleciona AN1
    ADCON2 = 0b10010101; // Resultado alinhado à
                        //direita,
    ADCON0bits.GO_DONE = 1; // Inicia a conversão A/D
    while (ADCON0bits.GO_DONE);
    Valor_convertor2 = 256 * ADRESH + ADRESL;
    // V_1bit = ( Vref+ - Vref -)/(2^10 -1) = 4,887585
    //mV
    // V_sinal_analógico = V_1bit * Valor_convertor

    if(cont < 10000)
    {
        while (TXSTAbits.TRMT==0) {}
        TXREG=(0x30+(Valor_convertor2/1000));
        while (TXSTAbits.TRMT==0) {}
        TXREG=(0x30+((Valor_convertor2/100)%10));
        while (TXSTAbits.TRMT==0) {}
        TXREG=(0x30+((Valor_convertor2/10)%10));
    }
}

```

```

while (TXSTAbits.TRMT==0) {}
TXREG=(0x30+(Valor_convertor2%10));
while (TXSTAbits.TRMT==0) {}

TXREG=0x0d;
while (TXSTAbits.TRMT==0) {}
TXREG=0x0a;
}

//valor normalizado de 0 a 5000
vel[i] = (Valor_convertor*5)/1.023;
jst[i] = (Valor_convertor2*5)/1.023;

//Calcula componente do Integrador
erro = 1000 - vel[i];
intg = intg + (1.5247*(erro)*0.00408);

//Monta sinal de controle
if(i==199)
{
    //realimenta os estados atuais
    u_atual = ((-0.9590*((short long)vel[i])) +
              (-1.2274*(short long)jst[i]));
    //realimenta os estados atrasados
    u_atrasado = (-1.6263*(short long)vel[0]) +
                (-1*(short long)jst[0]);
}
else
{
    //realimenta os estados atuais
    u_atual = ((-0.9590*((short long)vel[i])) +
              (-1.2274*(short long)jst[i]));
    //realimenta os estados atrasados
    u_atrasado = (-1.6263*(short long)vel[i+1]) +
                (-1*(short long)jst[i+1]);
}

u_total = (u_atual + u_atrasado +
           (short long)intg);

if(cont < 10000)
{
    while (TXSTAbits.TRMT==0) {}
    TXREG=(0x30+(u_total/1000));
    while (TXSTAbits.TRMT==0) {}
    TXREG=(0x30+((u_total/100)%10));
    while (TXSTAbits.TRMT==0) {}
    TXREG=(0x30+((u_total/10)%10));
    while (TXSTAbits.TRMT==0) {}
    TXREG=(0x30+(u_total%10));
    while (TXSTAbits.TRMT==0) {}
}

```

```

        TXREG=0x0d;
        while (TXSTAbits.TRMT==0) {}
        TXREG=0x0a;
    }

    //Calculo de Duty Cycle
    dc = ((int)u_total)/50; //Normaliza de 0 a 100
    if(dc < 7) dc = 7; //menor valor fisicamente
        //possível
    if(dc > 100) dc = 100;//maior valor fisicamente
        //possível
    TDC= 125*dc/100; //Taxa 10kHz

    // Configurar CCP2 modo PWM com Duty Cycle
    //calculado
    // CCPRxL:CCPxCON<5:4> = 375 us/(50 ns *16)=
    //468,75= ~469
    // CCPRxL:CCPxCON<5:4> = 0b0111010101
    // Os 8 bits mais significativos de ficam em CCPR2L
    // Os 2 bits menos significativos ficam em
    //CCP2CON<5:4>
    b8 = TDC&0b1111111100;
    b1 = TDC&0b0000000010;
    b0 = TDC&0b0000000001;

    //Gera PWM
    CCP2L = b8/4;
    CCP2CONbits.DC2B1 = b1/2;
    CCP2CONbits.DC2B0 = b0;
    CCP2CONbits.CCP2M3 = 1;
    CCP2CONbits.CCP2M2 = 1;
    CCP2CONbits.CCP2M1 = 0;
    CCP2CONbits.CCP2M0 = 0;

    cont++;
    i++;
    if(i==200) i=0;
}
}

```

**ANEXO IV – Código para separação de dados da comunicação serial**

**Código para separação de dados da comunicação serial**

```
function [velocidade, corrente, u] = trataDados(dados)
    ivel = 1;
    icor = 1;
    iu = 1;
    for i=1:length(vectest)
        resto = rem(i,3);
        switch resto
            case 1
                velocidade(ivel) = dados(i);
                ivel = ivel + 1;
            case 2
                corrente(icor) = dados(i);
                icor = icor + 1;
            otherwise
                u(iu) = dados(i);
                iu = iu + 1;
        end
    end
end
```

## **ANEXO V – Registradores conversor A/D**

## Registadores do conversor A/D

**TABLE 21-2: REGISTERS ASSOCIATED WITH A/D OPERATION**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	51
PIR1	SPPIF <sup>(4)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	54
PIE1	SPPIE <sup>(4)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	54
IPR1	SPPIP <sup>(4)</sup>	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	54
PIR2	OSCFIF	CMIF	USBIF	EEIF	BCLIF	HLVDIF	TMR3IF	CCP2IF	54
PIE2	OSCFIE	CMIE	USBIE	EEIE	BCLIE	HLVDIE	TMR3IE	CCP2IE	54
IPR2	OSCFIP	CMIP	USBIP	EEIP	BCLIP	HLVDIP	TMR3IP	CCP2IP	54
ADRESH	A/D Result Register High Byte								52
ADRESL	A/D Result Register Low Byte								52
ADCON0	—	—	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON	52
ADCON1	—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	52
ADCON2	ADFM	—	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0	52
PORTA	—	RA6 <sup>(2)</sup>	RA5	RA4	RA3	RA2	RA1	RA0	54
TRISA	—	TRISA6 <sup>(2)</sup>	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	54
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	54
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	54
LATB	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	54
PORTE	RDP <sup>(4)</sup>	—	—	—	RE3 <sup>(1,3)</sup>	RE2 <sup>(4)</sup>	RE1 <sup>(4)</sup>	RE0 <sup>(4)</sup>	54
TRISE <sup>(4)</sup>	—	—	—	—	—	TRISE2	TRISE1	TRISE0	54
LATE <sup>(4)</sup>	—	—	—	—	—	LATE2	LATE1	LATE0	54

**ADCON0: A/D CONTROL REGISTER 0**

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
—	—	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON	
bit 7								bit 0

**Legend:**

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 7-6 **Unimplemented:** Read as '0'bit 5-2 **CHS3:CHS0:** Analog Channel Select bits

0000 = Channel 0 (AN0)

0001 = Channel 1 (AN1)

0010 = Channel 2 (AN2)

0011 = Channel 3 (AN3)

0100 = Channel 4 (AN4)

0101 = Channel 5 (AN5)<sup>(1,2)</sup>0110 = Channel 6 (AN6)<sup>(1,2)</sup>0111 = Channel 7 (AN7)<sup>(1,2)</sup>

1000 = Channel 8 (AN8)

1001 = Channel 9 (AN9)

1010 = Channel 10 (AN10)

1011 = Channel 11 (AN11)

1100 = Channel 12 (AN12)

1101 = Unimplemented<sup>(2)</sup>1110 = Unimplemented<sup>(2)</sup>1111 = Unimplemented<sup>(2)</sup>bit 1 **GO/DONE:** A/D Conversion Status bitWhen ADON = 1:

1 = A/D conversion in progress

0 = A/D Idle

bit 0 **ADON:** A/D On bit

1 = A/D converter module is enabled

0 = A/D converter module is disabled



### ADCON1: A/D CONTROL REGISTER 1

U-0	U-0	R/W-0	R/W-0	R/W-0 <sup>(1)</sup>	R/W <sup>(1)</sup>	R/W <sup>(1)</sup>	R/W <sup>(1)</sup>
—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

#### Legend:

R = Readable bit                      W = Writable bit                      U = Unimplemented bit, read as '0'  
 -n = Value at POR                      '1' = Bit is set                      '0' = Bit is cleared                      x = Bit is unknown

bit 7-6                      **Unimplemented:** Read as '0'

bit 5                      **VCFG1:** Voltage Reference Configuration bit (VREF- source)

1 = VREF- (AN2)  
 0 = VSS

bit 4                      **VCFG0:** Voltage Reference Configuration bit (VREF+ source)

1 = VREF+ (AN3)  
 0 = VDD

bit 3-0                      **PCFG3:PCFG0:** A/D Port Configuration Control bits:

PCFG3: PCFG0	AN12	AN11	AN10	AN9	AN8	AN7 <sup>(2)</sup>	AN6 <sup>(2)</sup>	AN5 <sup>(2)</sup>	AN4	AN3	AN2	AN1	AN0
0000 <sup>(1)</sup>	A	A	A	A	A	A	A	A	A	A	A	A	A
0001	A	A	A	A	A	A	A	A	A	A	A	A	A
0010	A	A	A	A	A	A	A	A	A	A	A	A	A
0011	D	A	A	A	A	A	A	A	A	A	A	A	A
0100	D	D	A	A	A	A	A	A	A	A	A	A	A
0101	D	D	D	A	A	A	A	A	A	A	A	A	A
0110	D	D	D	D	A	A	A	A	A	A	A	A	A
0111 <sup>(1)</sup>	D	D	D	D	D	A	A	A	A	A	A	A	A
1000	D	D	D	D	D	D	A	A	A	A	A	A	A
1001	D	D	D	D	D	D	D	A	A	A	A	A	A
1010	D	D	D	D	D	D	D	D	A	A	A	A	A
1011	D	D	D	D	D	D	D	D	D	A	A	A	A
1100	D	D	D	D	D	D	D	D	D	D	A	A	A
1101	D	D	D	D	D	D	D	D	D	D	D	A	A
1110	D	D	D	D	D	D	D	D	D	D	D	D	A
1111	D	D	D	D	D	D	D	D	D	D	D	D	D

A = Analog input

D = Digital I/O

**ADCON2: A/D CONTROL REGISTER 2**

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0
bit 7							bit 0

**Legend:**

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

- bit 7      **ADFM:** A/D Result Format Select bit  
1 = Right justified  
0 = Left justified
- bit 6      **Unimplemented:** Read as '0'
- bit 5-3    **ACQT2:ACQT0:** A/D Acquisition Time Select bits  
111 = 20 TAD  
110 = 16 TAD  
101 = 12 TAD  
100 = 8 TAD  
011 = 6 TAD  
010 = 4 TAD  
001 = 2 TAD  
000 = 0 TAD<sup>(1)</sup>
- bit 2-0    **ADCS2:ADCS0:** A/D Conversion Clock Select bits  
111 = FRC (clock derived from A/D RC oscillator)<sup>(1)</sup>  
110 = FOSC/64  
101 = FOSC/16  
100 = FOSC/4  
011 = FRC (clock derived from A/D RC oscillator)<sup>(1)</sup>  
010 = FOSC/32  
001 = FOSC/8  
000 = FOSC/2

## **ANEXO VI – Registradores para sinal PWM**

## Registadores para sinal PWM

**TABLE 15-5: REGISTERS ASSOCIATED WITH PWM AND TIMER2**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	51
RCON	IPEN	SBOREN <sup>(1)</sup>	—	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$	52
PIR1	SPPIF <sup>(2)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	54
PIE1	SPPIE <sup>(2)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	54
IPR1	SPIIP <sup>(2)</sup>	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	54
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	54
TRISC	TRISC7	TRISC6	—	—	—	TRISC2	TRISC1	TRISC0	54
TMR2	Timer2 Register								52
PR2	Timer2 Period Register								52
T2CON	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	52
CCPR1L	Capture/Compare/PWM Register 1 Low Byte								53
CCPR1H	Capture/Compare/PWM Register 1 High Byte								53
CCP1CON	P1M1 <sup>(2)</sup>	P1M0 <sup>(2)</sup>	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	53
CCPR2L	Capture/Compare/PWM Register 2 Low Byte								53
CCPR2H	Capture/Compare/PWM Register 2 High Byte								53
CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	53
ECCP1AS	ECCPASE	ECCPAS2	ECCPAS1	ECCPAS0	PSSAC1	PSSAC0	PSSBD1 <sup>(2)</sup>	PSSBD0 <sup>(2)</sup>	53
ECCP1DEL	PRSEN	PDC6 <sup>(2)</sup>	PDC5 <sup>(2)</sup>	PDC4 <sup>(2)</sup>	PDC3 <sup>(2)</sup>	PDC2 <sup>(2)</sup>	PDC1 <sup>(2)</sup>	PDC0 <sup>(2)</sup>	53

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used by PWM or Timer2.

**Note 1:** The SBOREN bit is only available when BOREN<1:0> = 01; otherwise, the bit reads as '0'.

**2:** These bits are unimplemented on 28-pin devices; always maintain these bits clear.

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
— <sup>(1)</sup>	— <sup>(1)</sup>	DCxB1	DCxB0	CCPxM3	CCPxM2	CCPxM1	CCPxM0
bit 7							bit 0
bit 7-6	<b>Unimplemented:</b> Read as '0' <sup>(1)</sup>						
bit 5-4	<b>DCxB1:DCxB0:</b> PWM Duty Cycle Bit 1 and Bit 0 for CCPx Module <u>Capture mode:</u> Unused. <u>Compare mode:</u> Unused. <u>PWM mode:</u> These bits are the two LSBs (bit 1 and bit 0) of the 10-bit PWM duty cycle. The eight MSBs of the duty cycle are found in CCPR1L.						
bit 3-0	<b>CCPxM3:CCPxM0:</b> CCPx Module Mode Select bits 0000 = Capture/Compare/PWM disabled (resets CCPx module) 0001 = Reserved 0010 = Compare mode: toggle output on match (CCPxIF bit is set) 0011 = Reserved 0100 = Capture mode: every falling edge 0101 = Capture mode: every rising edge 0110 = Capture mode: every 4th rising edge 0111 = Capture mode: every 16th rising edge 1000 = Compare mode: initialize CCPx pin low; on compare match, force CCPx pin high (CCPxIF bit is set) 1001 = Compare mode: initialize CCPx pin high; on compare match, force CCPx pin low (CCPxIF bit is set) 1010 = Compare mode: generate software interrupt on compare match (CCPxIF bit is set, CCPx pin reflects I/O state) 1011 = Compare mode: trigger special event, reset timer, start A/D conversion on CCPx match (CCPxIF bit is set) 11xx = PWM mode						

**REGISTER 13-1: T2CON: TIMER2 CONTROL REGISTER**

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit 7							bit 0
bit 7	<b>Unimplemented:</b> Read as '0'						
bit 6-3	<b>T2OUTPS3:T2OUTPS0:</b> Timer2 Output Postscale Select bits 0000 = 1:1 Postscale 0001 = 1:2 Postscale • • • 1111 = 1:16 Postscale						
bit 2	<b>TMR2ON:</b> Timer2 On bit 1 = Timer2 is on 0 = Timer2 is off						
bit 1-0	<b>T2CKPS1:T2CKPS0:</b> Timer2 Clock Prescale Select bits 00 = Prescaler is 1 01 = Prescaler is 4 1x = Prescaler is 16						

## **ANEXO VII – Registradores Comunicação Serial Assíncrona**

## Registadores Comunicação Serial Assíncrona

REGISTER 20-3: BAUDCON: BAUD RATE CONTROL REGISTER							
R/W-0	R-1	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN
bit 7							bit 0
bit 7	<b>ABDOVF:</b> Auto-Baud Acquisition Rollover Status bit 1 = A BRG rollover has occurred during Auto-Baud Rate Detect mode (must be cleared in software) 0 = No BRG rollover has occurred						
bit 6	<b>RCIDL:</b> Receive Operation Idle Status bit 1 = Receive operation is Idle 0 = Receive operation is active						
bit 5	<b>RXDTP:</b> Received Data Polarity Select bit <u>Asynchronous mode:</u> 1 = RX data is inverted 0 = RX data received is not inverted <u>Synchronous modes:</u> 1 = Received Data (DT) is inverted. Idle state is a low level. 0 = No inversion of Data (DT). Idle state is a high level.						
bit 4	<b>TXCKP:</b> Clock and Data Polarity Select bit <u>Asynchronous mode:</u> 1 = TX data is inverted 0 = TX data is not inverted <u>Synchronous modes:</u> 1 = Clock (CK) is inverted. Idle state is a high level. 0 = No inversion of Clock (CK). Idle state is a low level.						
bit 3	<b>BRG16:</b> 16-Bit Baud Rate Register Enable bit 1 = 16-bit Baud Rate Generator – SPBRGH and SPBRG 0 = 8-bit Baud Rate Generator – SPBRG only (Compatible mode), SPBRGH value ignored						
bit 2	<b>Unimplemented:</b> Read as '0'						
bit 1	<b>WUE:</b> Wake-up Enable bit <u>Asynchronous mode:</u> 1 = EUSART will continue to sample the RX pin – interrupt generated on falling edge; bit cleared in hardware on following rising edge 0 = RX pin not monitored or rising edge detected <u>Synchronous mode:</u> Unused in this mode.						
bit 0	<b>ABDEN:</b> Auto-Baud Detect Enable bit <u>Asynchronous mode:</u> 1 = Enable baud rate measurement on the next character. Requires reception of a Sync field (55h); cleared in hardware upon completion. 0 = Baud rate measurement disabled or completed <u>Synchronous mode:</u> Unused in this mode.						



**REGISTER 20-2: RCSTA: RECEIVE STATUS AND CONTROL REGISTER**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
bit 7							bit 0
bit 7	<b>SPEN: Serial Port Enable bit</b> 1 = Serial port enabled (configures RX/DT and TX/CK pins as serial port pins) 0 = Serial port disabled (held in Reset)						
bit 6	<b>RX9: 9-Bit Receive Enable bit</b> 1 = Selects 9-bit reception 0 = Selects 8-bit reception						
bit 5	<b>SREN: Single Receive Enable bit</b> <u>Asynchronous mode:</u> Don't care. <u>Synchronous mode – Master:</u> 1 = Enables single receive 0 = Disables single receive This bit is cleared after reception is complete. <u>Synchronous mode – Slave:</u> Don't care.						
bit 4	<b>CREN: Continuous Receive Enable bit</b> <u>Asynchronous mode:</u> 1 = Enables receiver 0 = Disables receiver <u>Synchronous mode:</u> 1 = Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN) 0 = Disables continuous receive						
bit 3	<b>ADDEN: Address Detect Enable bit</b> <u>Asynchronous mode 9-bit (RX9 = 1):</u> 1 = Enables address detection, enables interrupt and loads the receive buffer when RSR<8> is set 0 = Disables address detection, all bytes are received and ninth bit can be used as parity bit <u>Asynchronous mode 8-bit (RX9 = 0):</u> Don't care.						
bit 2	<b>FERR: Framing Error bit</b> 1 = Framing error (can be updated by reading RCREG register and receiving next valid byte) 0 = No framing error						
bit 1	<b>OERR: Overrun Error bit</b> 1 = Overrun error (can be cleared by clearing bit CREN) 0 = No overrun error						
bit 0	<b>RX9D: 9th bit of Received Data</b> This can be address/data bit or a parity bit and must be calculated by user firmware.						



**REGISTER 20-1: TXSTA: TRANSMIT STATUS AND CONTROL REGISTER**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN <sup>(1)</sup>	SYNC	SENDB	BRGH	TRMT	TX9D
bit 7							bit 0
bit 7	<b>CSRC:</b> Clock Source Select bit <u>Asynchronous mode:</u> Don't care. <u>Synchronous mode:</u> 1 = Master mode (clock generated internally from BRG) 0 = Slave mode (clock from external source)						
bit 6	<b>TX9:</b> 9-Bit Transmit Enable bit 1 = Selects 9-bit transmission 0 = Selects 8-bit transmission						
bit 5	<b>TXEN:</b> Transmit Enable bit <sup>(1)</sup> 1 = Transmit enabled 0 = Transmit disabled						
bit 4	<b>SYNC:</b> EUSART Mode Select bit 1 = Synchronous mode 0 = Asynchronous mode						
bit 3	<b>SENDB:</b> Send Break Character bit <u>Asynchronous mode:</u> 1 = Send Sync Break on next transmission (cleared by hardware upon completion) 0 = Sync Break transmission completed <u>Synchronous mode:</u> Don't care.						
bit 2	<b>BRGH:</b> High Baud Rate Select bit <u>Asynchronous mode:</u> 1 = High speed 0 = Low speed <u>Synchronous mode:</u> Unused in this mode.						
bit 1	<b>TRMT:</b> Transmit Shift Register Status bit 1 = TSR empty 0 = TSR full						
bit 0	<b>TX9D:</b> 9th bit of Transmit Data Can be address/data bit or a parity bit.						