



UNIVERSIDADE FEDERAL DO AMAZONAS  
FACULDADE DE TECNOLOGIA  
ENGENHARIA DA COMPUTAÇÃO

## Identificação e Controle de um Levitador a Ar

Mateus Martínez de Lucena

MANAUS-AM

2018

Mateus Martínez de Lucena

## Identificação e Controle de um Levitador a Ar

Monografia apresentada à Coordenação do Curso de Engenharia da Computação da Universidade Federal do Amazonas, como parte dos requisitos necessários à obtenção do título de Engenheiro de Computação.

Orientador: Iury Valente de Bessa

MANAUS-AM

2018

*DEDICATÓRIA AQUI*

# Agradecimentos

AGRADECIMENTOS AQUI.

*EPÍGRAFE AQUI (AUTOR AQUI,  
TÍTULO DA OBRA.)*

# Resumo

A identificação caixa-preta é uma forma de identificar um modelo matemático para um sistema que, em alguns casos, não pode ser modelado de outra forma. O controle de altura de um objeto dentro de um túnel de vento é um desses casos onde a modelagem não é suficiente. Este trabalho estuda a identificação do túnel de vento por mínimos quadrados e por subespaços para aplicar um controle por realimentação de estados.

**Palavras-chave:** Levitador, Túnel de Vento, Controle.

# Abstract

ABSTRACT AQUI

**Keywords:** KEYWORDS HERE.

# Lista de Figuras

2.1	Esquema do amortecedor da roda de um carro . . . . .	5
3.1	Túnel de vento construído . . . . .	21
3.2	Forças atuantes na bola . . . . .	21
3.3	Superfície de $v_{ar}$ em função de altura e pwm . . . . .	23
3.4	Gráfico $v_{ar}$ x altura . . . . .	23
3.5	Simulador do túnel de vento . . . . .	24
3.6	Resposta ao degrau do simulador . . . . .	24
4.1	Gráfico da saída PRBS do sistema . . . . .	27
4.2	Gráfico da saída PRBS do simulador . . . . .	27
4.3	Autocorrelação dos resíduos do modelo $ARX1$ . . . . .	30
4.4	Autocorrelação dos resíduos do modelo $ARX2$ . . . . .	30
4.5	Autocorrelação dos resíduos do modelo $ARXsim$ . . . . .	31
4.6	Autocorrelação dos resíduos do modelo $SUB1$ . . . . .	31
4.7	Resposta ao degrau do modelo $ARX1$ . . . . .	31
4.8	Resposta ao degrau do modelo $ARX2$ . . . . .	32
4.9	Resposta ao degrau do modelo $ARXsim$ . . . . .	32
4.10	Resposta ao degrau do modelo $SUB1$ . . . . .	32
5.1	Resposta ao degrau unitário do modelo $SUB1$ controlado . . . . .	35
5.2	Resposta ao degrau do modelo simulado $ARX1$ controlado . . . . .	35
5.3	Resposta ao degrau do modelo $ARX2$ . . . . .	36
5.4	Resposta ao degrau do modelo simulado $ARXsim$ . . . . .	37
5.5	Simulação do estimador do modelo $SUB1$ . . . . .	38
5.6	Resposta ao degrau do modelo $SUB1$ controlado com observador . . . . .	38



5.7	Simulação do estimador do modelo <i>ARX1</i> . . . . .	39
5.8	Resposta ao degrau do modelo <i>ARX1</i> controlado com observador . . . . .	39
5.9	Simulação do estimador do modelo <i>ARX2</i> . . . . .	40
5.10	Resposta ao degrau do modelo <i>ARX2</i> controlado com observador . . . . .	40
5.11	Simulação do estimador do modelo <i>ARXsim</i> . . . . .	41
5.12	Resposta ao degrau do modelo <i>ARXsim</i> controlado com observador . . . . .	41
6.1	Resposta ao degrau do sistema usando o controlador <i>SUB1</i> . . . . .	43
6.2	Resposta ao degrau do sistema com o controlador do modelo <i>ARX1</i> . . . . .	44
6.3	Resposta ao degrau do sistema com o controlador do modelo <i>ARX2</i> . . . . .	45
6.4	Resposta ao degrau do sistema com o controlador do modelo <i>ARXsim</i> . . . . .	45
6.5	Gráfico ISE dos 4 modelos estimados . . . . .	46
6.6	Resposta à escadaria do sistema com o controlador do modelo <i>SUB1</i> . . . . .	47
6.7	Resposta à escadaria do sistema com o controlador do modelo <i>ARX1</i> . . . . .	47
6.8	Resposta à escadaria do sistema com o controlador do modelo <i>ARX2</i> . . . . .	47
6.9	Resposta à escadaria do sistema com o controlador do modelo <i>ARXsim</i> . . . . .	48
6.10	Erro do estimador do modelo <i>SUB1</i> na resposta à escadaria . . . . .	48
6.11	Erro do estimador do modelo <i>ARX1</i> na resposta à escadaria . . . . .	49
6.12	Erro do estimador do modelo <i>ARX2</i> na resposta à escadaria . . . . .	49
6.13	Resposta do sistema com o controlador do modelo <i>SUB1</i> com mudança de parâmetros . . . . .	50
6.14	Resposta do sistema com o controlador do modelo <i>ARX1</i> com mudança de parâmetros . . . . .	50
6.15	Resposta do sistema com o controlador do modelo <i>ARX2</i> com mudança de parâmetros . . . . .	50
6.16	Resposta do sistema com o controlador do modelo <i>ARXsim</i> com mudança de parâmetros . . . . .	51
6.17	Resposta à perturbação externa do sistema com o controlador do modelo <i>SUB1</i> . . . . .	51
6.18	Resposta à perturbação externa do sistema com o controlador do modelo <i>ARX1</i> . . . . .	52
6.19	Resposta à perturbação externa do sistema com o controlador do modelo <i>ARX1</i> simulado . . . . .	52

---

6.20	Resposta à perturbação externa do sistema com o controlador do modelo	
	<i>ARX2</i> . . . . .	53
6.21	Resposta à perturbação externa do sistema com o controlador do modelo	
	<i>ARXsim</i> . . . . .	53

# Lista de Tabelas

3.1	Tabela da velocidade do ar em função de altura e do PWM aplicado ao motor . . . . .	22
6.1	Tabela dos requisitos do sistema <i>SUB1</i> . . . . .	44
6.2	Tabela dos requisitos do sistema <i>ARX1</i> . . . . .	44
6.3	Tabela dos requisitos do sistema <i>SUB1</i> . . . . .	45
6.4	Tabela de comparação dos requisitos de controle dos modelos e índice ISE .	46
6.5	Tabela com índices ISE do erro de estimação . . . . .	48

# Lista de Abreviaturas e Siglas

<b>PRBS</b>	Sinal Binário Pseudo Aleatório – do inglês Pseudo Random Binary Signal
<b>E.D.O.</b>	Equação Diferencial Ordinária
<b>ARX</b>	Auto regressivo com termos exógenos – do inglês Autoregressive with exogenous terms
<b>SISO</b>	Entrada única saída única – do inglês Single Input Single Output
<b>BIBO</b>	Entrada limitada saída limitada – do inglês Bounded Input Bounded Output
<b>PE Condition</b>	Condição de excitação permanente – do inglês Permanent Excitement Condition
<b>PVC</b>	Policloreto de polivinila
<b>RC</b>	Controle remoto – do inglês Remote Control

# Lista de Símbolos

## Símbolos Matemáticos

$\mathbb{R}$	conjunto dos números reais
$\varphi$	matriz de regressores dos mínimos quadrados
$\{\cdot\}^\dagger$	pseudo inversa

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Levitação a ar . . . . .	1
1.2	Objetivos . . . . .	2
1.2.1	Objetivo Geral . . . . .	2
1.2.2	Objetivos Específicos . . . . .	2
1.3	Organização do trabalho . . . . .	3
<b>2</b>	<b>Revisão Bibliográfica</b>	<b>4</b>
2.1	Fundamentação Teórica . . . . .	4
2.1.1	Modelos de Sistemas Dinâmicos . . . . .	4
2.1.1.1	Equação Diferencial Ordinária . . . . .	4
2.1.1.2	Funções de Transferência . . . . .	5
2.1.1.3	Espaço de Estados . . . . .	6
2.1.1.4	Modelo ARX . . . . .	7
2.1.2	Critérios de Estabilidade de sistemas discretos . . . . .	7
2.1.3	Projeto de Controlador com Realimentação de Estados . . . . .	10
2.1.3.1	Controlador . . . . .	11
2.1.3.2	Estimador de Estados . . . . .	11
2.1.4	Identificação de Sistemas e Estimação de Parâmetros . . . . .	12
2.1.4.1	Visão Geral . . . . .	12
2.1.4.2	Escolha de sinais e estrutura . . . . .	13
2.1.4.3	Identificação por Mínimos Quadrados . . . . .	13
2.1.4.4	Identificação por Subespaços . . . . .	15
2.1.4.5	Validação de modelos identificados . . . . .	18
2.2	Trabalhos Relacionados . . . . .	19

<b>3</b>	<b>Modelagem e Simulação</b>	<b>20</b>
3.1	Descrição da Planta . . . . .	20
3.2	Modelo Matemático . . . . .	20
3.3	Estimação da Velocidade do Vento . . . . .	22
3.4	Simulação do Túnel de Vento no Simulink . . . . .	23
<b>4</b>	<b>Identificação do Túnel de Ar</b>	<b>25</b>
4.1	Escolha de estrutura . . . . .	25
4.1.1	Mínimos Quadrados . . . . .	25
4.1.2	Subespaços . . . . .	26
4.2	Experimento . . . . .	27
4.3	Estimação . . . . .	28
4.3.1	Mínimos Quadrados . . . . .	28
4.3.2	Subespaços . . . . .	28
4.4	Validação . . . . .	29
<b>5</b>	<b>Projeto de Controlador com Alocação de Polos Baseado nos Modelos Identificados</b>	<b>33</b>
5.1	Alocação de Polos com Realimentação de Estados . . . . .	33
5.1.1	Modelo <i>SUB1</i> . . . . .	33
5.1.2	Modelo <i>ARX1</i> . . . . .	35
5.1.3	Modelo <i>ARX2</i> . . . . .	36
5.1.4	Modelo <i>ARXsim</i> . . . . .	36
5.2	Projeto do Estimador de Estados . . . . .	37
5.2.1	Modelo <i>SUB1</i> . . . . .	37
5.2.2	Modelo <i>ARX1</i> . . . . .	38
5.2.3	Modelo <i>ARX2</i> . . . . .	39
5.2.4	Modelo <i>ARXsim</i> . . . . .	40
<b>6</b>	<b>Avaliação Experimental do Sistema de Controle</b>	<b>42</b>
6.1	Descrição e objetivos experimentais . . . . .	42
6.2	Resultados Experimentais . . . . .	43
6.2.1	Resultados da Resposta ao degrau unitário . . . . .	43
6.2.1.1	Modelo <i>SUB1</i> . . . . .	43

6.2.1.2	Modelo <i>ARX1</i> . . . . .	44
6.2.1.3	Modelo <i>ARX2</i> . . . . .	44
6.2.1.4	Modelo <i>ARXsim</i> . . . . .	44
6.2.1.5	ISE da resposta ao degrau dos modelos simulada . . . . .	45
6.2.1.6	Comparação dos modelos . . . . .	46
6.2.2	Resultados da Resposta à Escadaria . . . . .	46
6.2.2.1	Modelo <i>SUB1</i> . . . . .	46
6.2.2.2	Modelo <i>ARX1</i> . . . . .	46
6.2.2.3	Modelo <i>ARX2</i> . . . . .	47
6.2.2.4	Modelo <i>ARXsim</i> . . . . .	47
6.2.2.5	Erro do Estimador . . . . .	48
6.2.3	Resultados do Teste de Robustez à mudança de Parâmetros . . . . .	49
6.2.3.1	Modelo <i>SUB1</i> . . . . .	49
6.2.3.2	Modelo <i>ARX1</i> . . . . .	50
6.2.3.3	Modelo <i>ARX2</i> . . . . .	50
6.2.3.4	Modelo <i>ARXsim</i> . . . . .	51
6.2.4	Resultados do Teste de Robustez à perturbação externa . . . . .	51
6.2.4.1	Modelo <i>SUB1</i> . . . . .	51
6.2.4.2	Modelo <i>ARX1</i> . . . . .	51
6.2.4.3	Modelo <i>ARX2</i> . . . . .	52
6.2.4.4	Modelo <i>ARXsim</i> . . . . .	52
6.3	Discussão . . . . .	53
<b>7</b>	<b>Conclusão</b>	<b>55</b>
7.1	Considerações Finais . . . . .	55
7.2	Trabalhos Futuros . . . . .	56
	<b>Referências Bibliográficas</b>	<b>57</b>



# Capítulo 1

## Introdução

### 1.1 Levitação a ar

O túnel de vento é uma instalação utilizada para estudar o desempenho aerodinâmico de objetos. Ele consiste em um duto de tamanho adequado para o objeto testado pelo qual o ar flui ao ser empurrado por uma turbina. Os testes feitos variam de velocidade, de acordo com o que se deseja estudar.

Para que os testes sobre objetos tenham resultados precisos e replicáveis, a velocidade dos ventos sobre eles deve ser controlável e o fluxo de ar que atinge o objeto não deve ter turbulência, apesar de haver testes onde isso é proposital. O túnel é construído com um dispositivo de fluxo laminar que possibilita reduzir o fluxo turbulento gerado pela turbina [10].

Os testes geralmente são feitos em modelos em escala de aviões, carros, naves espaciais e para estudo da aerodinâmica de objetos na física. Os túneis podem ser verticais ou horizontais, apesar de os túneis horizontais existirem em maior quantidade. Um teste comum é feito em modelos de aviões onde se medem as propriedades das asas como envergadura ao ser submetida a velocidades específicas do vento, força de elevação para diferentes velocidades e, em geral, sua aerodinâmica. Em Dayton, Ohio [8], foi construído um túnel de vento vertical que possibilita que sejam feitos testes específicos para aeronaves. Um objeto é posto em queda livre virtual devido a velocidade do vento impulsionado por uma turbina e se observa o seu movimento na situação. Este teste foi essencial para a prototipagem de aviões militares e comerciais desde os anos 50.

Hoje, os testes ainda são responsáveis por economizar milhões de reais na fase de pro-

totipagem em testes aerodinâmicos usados no desenvolvimento de trens de alta velocidade [3]. Em objetos que se movem em alta velocidade, o vento se torna um fator preocupante. Uma rajada de vento pode ser o fator determinante em um descarrilhamento de um trem. Os testes possibilitam que os cientistas analisem o comportamento de suas criações em situações previsíveis e os ajuda a antecipar possíveis problemas.

No entanto, existe um problema no teste aerodinâmico utilizando túneis de vento. Para os testes de queda livre como o de aeronaves, o controle de altitude do objeto é complexo. É possível obter o número de Reynolds, uma variável sem dimensão que possibilita prever o padrão de fluxo de fluídos, e é possível medir a velocidade do vento atuando em cima do objeto. Porém, a real dificuldade se encontra no fato de que quando a força do vento atua em cima de um objeto, ele tende a girar [4] de acordo com suas características aerodinâmicas, o que não é possível de se controlar sem um atuador no objeto.

Tendo em vista que o controle de altura de um objeto tendo como único atuador o vento, e que um modelo de atuação da força do vento sobre um objeto é complexo e de baixa fidelidade, e que precisamos de um modelo que possibilite um controle de altitude preciso e robusto, temos como opção a modelagem caixa-preta, conhecida como identificação de sistemas.

## 1.2 Objetivos

### 1.2.1 Objetivo Geral

O objetivo deste trabalho é construir uma plataforma de estudos de túnel de ar, aplicar identificação por subespaços e por mínimos quadrados, e projetar um controlador para o sistema.

### 1.2.2 Objetivos Específicos

Para alcançar o objetivo geral deste projeto, devemos primeiro alcançar os seguintes objetivos específicos.

- Construir uma plataforma de experimento no formato de túnel de vento;
- Medir a velocidade do vento em alturas variadas dentro do túnel de vento;

- Criar um simulador para o túnel de vento;
- Executar testes para identificação usando como sinal de controle um sinal binário pseudo aleatório (PRBS);
- Identificar o sistema por mínimos quadrados;
- Identificar o sistema por subespaços;
- Validar os modelos obtidos;
- Obter um controlador para os dois modelos usando alocação de polos.

## 1.3 Organização do trabalho

Este trabalho se encontra organizado em seis capítulos com várias subseções.

- Capítulo 2: é a revisão bibliográfica. A fundamentação teórica dividida em conceitos fundamentais ao entendimento do trabalho, explicações sobre modelos de sistemas dinâmicos, dos critérios de estabilidade de sistemas discretos, e da identificação de sistemas e estimação de parâmetros.
- Capítulo 3: descreve a modelagem e simulação do túnel de vento, onde se explica o processo de modelagem utilizado e a construção de um simulador no Simulink.
- Capítulo 4: trata da identificação do túnel usando mínimos quadrados e subespaços, explicando os procedimentos usados e o método de validação do modelo.
- Capítulo 5: explica como foi projetado o controlador para adequar o sistema aos requisitos de sua resposta ao degrau unitário.
- Capítulo 6: trata da avaliação experimental do sistema de controle, onde o sistema é testado para avaliar sua robustez.
- Capítulo 7: é a conclusão do trabalho onde as considerações finais e trabalhos futuros serão discutidos.

# Capítulo 2

## Revisão Bibliográfica

Nesta seção é apresentada a revisão bibliográfica, que trata de conceitos fundamentais ao entendimento do trabalho.

### 2.1 Fundamentação Teórica

Nesta seção tratamos de conceitos essenciais para o entendimento deste trabalho. Falaremos sobre alguns modelos matemáticos, critérios de estabilidade de sistemas discretos, projeto de controle por realimentação de estados, e métodos de identificação de sistemas e estimação de parâmetros, em especial, identificação por mínimos quadrados e por subespaços.

#### 2.1.1 Modelos de Sistemas Dinâmicos

O modelo matemático de um sistema dinâmico é uma forma de descrever uma parte do fenômeno físico que queremos controlar. Nesta seção, serão apresentados alguns dos vários modelos matemáticos relacionados ao projeto.

##### 2.1.1.1 Equação Diferencial Ordinária

Um sistema dinâmico pode ser modelado usando somente a descrição matemática do fenômeno desejado. Usamos como exemplo a modelagem do amortecimento em uma roda de um carro, visto na figura [2.1](#).

Onde  $X_1$  e  $X_2$  são a altura das massas  $M_1$  e  $M_2$ , massa do carro e massa da roda, respectivamente.  $K_1$  é o efeito elástico do amortecedor e  $K_2$  é o efeito elástico da roda.

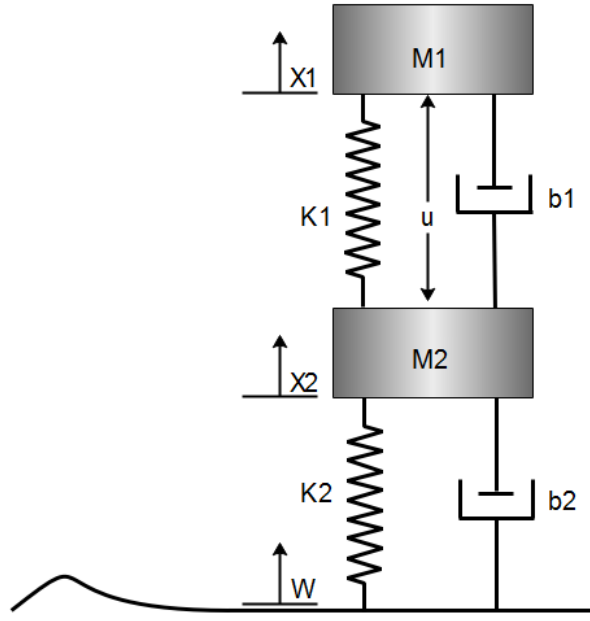


Figura 2.1: Esquema do amortecedor da roda de um carro

$b_1$  é o efeito amortecedor da suspensão e  $b_2$  é o efeito amortecedor da roda. Esse sistema pode ser modelado em uma E.D.O. (Equação Diferencial Ordinária) usando as leis de Newton, visto nas equações (2.1) e (2.2).

$$M_1 \ddot{X}_1 = -b_1(\dot{X}_1 - \dot{X}_2) - K_1(X_1 - X_2) + U \quad (2.1)$$

$$M_2 \ddot{X}_2 = b_1(\dot{X}_1 - \dot{X}_2) + K_1(X_1 - X_2) + b_2(\dot{W} - \dot{X}_2) + K_2(W - X_2) - U \quad (2.2)$$

### 2.1.1.2 Funções de Transferência

A função de transferência é uma equação que descreve o comportamento dinâmico de um sistema relacionando uma entrada com uma saída. Ela é a transformada de Laplace da resposta ao impulso do sistema.

Usando o mesmo sistema da figura 2.1 e aproveitando as equações (2.1) e (2.2), podemos encontrar as equações de transferência referentes às saídas  $X_1$  e  $X_2$ .

Aplicamos a transformada de Laplace nas equações  $X_1$  e  $X_2$  para obter:

$$(M_1 s^2 + b_1 s + K_1)X_1(s) - (b_1 s + K_1)X_2(s) = U(s) \quad (2.3)$$

$$-(b_1 s + K_1)X_1(s) + (M_2 s^2 + (b_1 + b_2)s + (K_1 + K_2))X_2(s) = (b_2 s + K_2)W(s) - U(s) \quad (2.4)$$

Fazendo as devidas manipulações matemáticas chegamos à função de transferência.

$$\begin{aligned} G_1(s) &= \frac{X_1(s) - X_2(s)}{W(s)} = \frac{(M_1 + M_2)s^2 + b_2s + K_2}{\Delta} \\ G_2(s) &= \frac{X_1(s) - X_2(s)}{W(s)} = \frac{-M_1b_2s^3 - M_1K_2s^2}{\Delta} \\ \Delta &= (M_1s^2 + b_1s + K_1) \cdot (M_2s^2 + (b_1 + b_2)s + (K_1 + K_2)) - (b_1s + K_1) \cdot (b_1s + K_1) \end{aligned} \quad (2.5)$$

Com as equações descritas em (2.5), temos uma equação representando a transferência da entrada para a saída.

### 2.1.1.3 Espaço de Estados

A representação em espaço de estados é uma forma mais conveniente para representar sistemas no domínio do tempo quando existe mais de uma entrada ou saída do que a função de transferência. Um modelo linear é representado tipicamente no seguinte formato:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \quad (2.6)$$

Onde  $x \in \mathbb{R}^n$  é o vetor de estado n-dimensional.  $\dot{x} = dx/dt$ ,  $u(t) \in \mathbb{R}^p$  é o vetor de entradas formado por r funções temporais,  $y(t) \in \mathbb{R}^q$  é o vetor m-dimensional de saídas e  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times p}$ ,  $C \in \mathbb{R}^{q \times n}$  e  $D \in \mathbb{R}^{q \times p}$  são as matrizes do sistema constantes.

Podemos gerar uma representação de espaço de estados a partir da E.D.O. do sistema. Vamos utilizar novamente o sistema da figura 2.1. Definimos primeiramente os estados do vetor x,  $x_1 = X_1$ ,  $x_2 = \dot{X}_1$ ,  $x_3 = Y_1$  e  $x_4 = \dot{Y}_1$ . Onde  $Y_1 = X_1 - X_2$ , geramos, então, as matrizes do espaço de estados usando as equações (2.1) e (2.2):

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{-b_1b_2}{M_1M_2} & 0 & \left[ \frac{b_1}{M_1} \left( \frac{b_1}{M_1} + \frac{b_1}{M_2} + \frac{b_2}{M_2} \right) - \frac{K_1}{M_1} \right] & \frac{-b_1}{M_1} \\ \frac{b_2}{M_2} & 0 & - \left( \frac{b_1}{M_1} + \frac{b_1}{M_2} + \frac{b_2}{M_2} \right) & 1 \\ \frac{K_2}{M_2} & 0 & - \left( \frac{K_1}{M_1} + \frac{K_1}{M_2} + \frac{K_2}{M_2} \right) & 0 \end{bmatrix} \quad (2.7)$$

$$B = \begin{bmatrix} 0 & 0 \\ \frac{1}{M_1} & \frac{b_1 b_2}{M_1 M_2} \\ 0 & \frac{-b_2}{M_2} \\ \left(\frac{1}{M_1} + \frac{1}{M_2}\right) & \frac{-\tilde{K}_2}{M_2} \end{bmatrix} \quad (2.8)$$

$$\begin{bmatrix} \dot{X}_1 \\ \ddot{X}_1 \\ \dot{Y}_1 \\ \ddot{Y}_1 \end{bmatrix} = A \begin{bmatrix} X_1 \\ \dot{X}_1 \\ Y_1 \\ \dot{Y}_1 \end{bmatrix} + B \begin{bmatrix} U \\ W \end{bmatrix} \quad (2.9)$$

$$Y = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_1 \\ \dot{X}_1 \\ Y_1 \\ \dot{Y}_1 \end{bmatrix} + \begin{bmatrix} 0 & 0 \end{bmatrix} \begin{bmatrix} U \\ W \end{bmatrix} \quad (2.10)$$

#### 2.1.1.4 Modelo ARX

O modelo ARX SISO linear tem o seguinte formato:

$$\begin{aligned} y(t) + a_1 y(t-1) + a_2 y(t-2) + \dots + a_{na} y(t-na) = \\ b_1 u(t) + b_2 u(t-1) + \dots + b_{nb} u(t-nb+1) + e(t) \end{aligned} \quad (2.11)$$

Onde  $y$  é a entrada,  $u$  é a saída e  $e$  é o ruído. Isso implica que a saída  $y(t)$  é uma predição a partir de uma média ponderada de entradas e saídas passadas. Os parâmetros  $a_{na}$  e  $b_{nb}$  podem ser estimados utilizando mínimos quadrados numa coleção de dados entrada-saída.

### 2.1.2 Critérios de Estabilidade de sistemas discretos

O conceito de estabilidade é extremamente importante para a análise de sistemas dinâmicos. Primeiramente, definimos estabilidade de acordo com mudanças nas condições iniciais. Consideremos a equação (2.12).

$$x(k+1) = f(x(k), k) \quad (2.12)$$

Sejam  $x_0(k)$  e  $x(k)$  suas soluções quando as condições iniciais são  $x_0(k_0)$  e  $x(k_0)$ . Definimos:

- Estabilidade: A solução  $x_0(k)$  é estável, se para um dado  $\epsilon > 0$ , existe um  $\delta(\epsilon, k_0)$  tal que para todas as soluções com  $\|x(k_0) - x_0(k_0)\| < \delta$  são tais que  $\|x(k) - x_0(k)\| < \epsilon$  para todos os  $k \geq k_0$ .

- Estabilidade Assintótica: A solução  $x_0(k)$  é assintoticamente estável se é estável e um  $\delta$  pode ser escolhido tal que  $\|x(k_0) - x_0(k_0)\| < \delta$  que implica que  $\|x(k) - x_0(k)\| \rightarrow 0$  quando  $k \rightarrow \infty$ .

Existem outros tipos de estabilidade que são de interesse:

- Estabilidade BIBO (Bounded-Input Bounded-Output): Um sistema linear invariante no tempo é definido BIBO estável se, dado um sinal de entrada limitado ele produz uma saída limitada para cada valor inicial.

Dado que a estabilidade de um sistema é importante para o seu estudo, os métodos para determinar a sua estabilidade são de grande interesse. Os seguintes são alguns dos métodos utilizados para determinar a estabilidade de um sistema:

- Cálculo dos autovalores da matriz A da representação de espaços de estado.
- Métodos baseados nas propriedades dos polinômios característicos.
- O método do lugar das raízes
- O método de Lyapunov

O cálculo dos autovalores de uma matriz de ordem maior que 2 à mão não é conveniente e, em alguns, casos é mais fácil calcular a equação característica da forma,

$$A(z) = a_0 z^n + a_1 z^{n-1} \cdots + a_n = 0 \quad (2.13)$$

e investigar suas raízes utilizando o método do lugar das raízes, onde o critério de estabilidade muda para sistemas discretos, determinando que para o sistema ser estável, todas as raízes devem estar dentro do círculo unitário.

Outro método para determinar a estabilidade de um sistema é o critério de Jury, versão discreta do critério de Routh-Hurwitz. A tabela de Jury é gerada conforme:

$$H(z) = \frac{b(z)}{a(z)} = \frac{b(z)}{a_0 z^n + a_1 z^{n-1} + \cdots + a_n} \quad (2.14)$$



$$\begin{array}{c|cccc}
a_0 & a_1 & \dots & a_n \\
a_n & a_{n-1} & \dots & a_0 \\
b_0 & b_1 & \dots & \\
b_{n-1} & b_{n-2} & \dots & \\
c_0 & c_1 & \dots & \\
\vdots & \vdots & & 
\end{array} \quad (2.15)$$

onde

$$\begin{aligned}
b_0 &= a_0 - \frac{a_n}{a_0} a_n \\
b_1 &= a_1 - \frac{a_n}{a_0} a_{n-1} \\
&\vdots \\
b_k &= a_k - \frac{a_n}{a_0} a_{n-k} \\
&\vdots \\
c_k &= b_k - \frac{b_{n-1}}{b_0} b_{n-1-k}
\end{aligned} \quad (2.16)$$

Com a tabela formada, aplicamos o critério de Jury que diz: se  $a_0 > 0$ , então todas as raízes estarão dentro do círculo unitário se, e somente se, todos os termos da primeira coluna das linhas ímpares forem positivos. Se nenhum elemento da primeira coluna das linhas ímpares for nulo, o número de raízes fora do círculo unitário é igual ao número de elementos negativos.

O segundo método de Lyapunov é outra ferramenta útil para determinar a estabilidade de sistemas dinâmicos não lineares. A ideia é introduzir uma função de energia generalizada chamada função de Lyapunov que é zero no ponto de equilíbrio e positiva em outras posições. O equilíbrio será estável, se pudermos mostrar que a função de Lyapunov diminui ao longo das trajetórias do sistema.

O primeiro passo é encontrar a função de Lyapunov definida como segue:

$V(x)$  é uma função de Lyapunov do sistema

$$x(k+1) = f(x(k)) \quad f(0) = 0 \quad (2.17)$$

se:

1.  $V(x)$  é contínuo em  $x$  e  $V(0)=0$
2.  $V(x)$  é definida positiva

3.  $\Delta V(x) = V(f(x)) - V(x)$  é definida negativa

Definimos:

• Teorema de estabilidade de Lyapunov: A solução  $x(k) = 0$  é assintoticamente estável, se existir uma função de Lyapunov para o sistema da equação (2.17). Se:

$$0 < \varphi(\|x\|) < V(x) \quad (2.18)$$

onde  $\varphi(\|x\|) \rightarrow \infty$  quando  $\|x\| \rightarrow \infty$ , então a solução é assintoticamente estável para todas as condições iniciais.

A grande dificuldade do teorema de Lyapunov é encontrar uma função de Lyapunov adequada. No entanto, para sistemas lineares como o da equação (2.19), é fácil determinar uma função de Lyapunov quadrática.

$$x_0(k+1) = \Phi x_0(k) \quad x_0(0) = a_0 \quad (2.19)$$

Tomemos  $V(x) = x^T P_x$  como candidato para a função de Lyapunov. O incremento de V é dado por:

$$\begin{aligned} \Delta V(x) &= V(\Phi x) - V(x) = x^T \Phi^T P \Phi x - x^T P x \\ &= x^T (\Phi^T P \Phi - P) x = -x^T Q x \end{aligned} \quad (2.20)$$

Para que V seja uma função de Lyapunov, é necessário e suficiente que exista uma matriz definida positiva P que satisfaça a equação (2.21).

$$\Phi^T P \Phi - P = -Q \quad (2.21)$$

### 2.1.3 Projeto de Controlador com Realimentação de Estados

Uma forma de controlar um sistema é utilizando realimentação de estados. A realimentação de estados é feita em sistemas modelados em espaço de estado, e a cada instante, cada estado individual é realimentado usando um ganho  $K$ . Esse tipo de controle requer que todos os estados sejam observáveis. Entretanto, isso quer dizer que o sistema precisa de um sensor para cada estado conhecido, o que acaba sendo muito custoso em termos de projeto e dinheiro. A solução para isso é projetar, além do controlador, um observador de estados, também chamado estimador, capaz de estimar o estado atual a partir da saída e da entrada do sistema.

### 2.1.3.1 Controlador

O controlador de realimentação de estados faz uma alocação de polos no sistema do tipo  $A - BK$ . Uma das formas de escolher  $K$ , é com o seguinte procedimento explicado por Chen [6]:

---

**Algoritmo 1:** ALOCAÇÃO DE POLOS POR REALIMENTAÇÃO DE ESTADOS

---

**Entrada:** Matrizes  $A$  e  $B$  do sistema, polos escolhidos para serem alocados

**Saída:** Ganho de realimentação de estados  $K$

**início**

- |   |  |
|---|--|
| 1 | Crie uma matriz $F$ com os autovalores (polos) escolhidos.                                 |
| 2 | Selecione um vetor arbitrário $\hat{k}$ de forma que o par $(F, \hat{k})$ seja observável. |
| 3 | Solucione o $T$ único na equação de Lyapunov $AT - TF = B\hat{k}$ .                        |
| 4 | Calcule o ganho de realimentação $K = \hat{k}T^{-1}$ .                                     |

**fim**

**retorna**  $K$

---

### 2.1.3.2 Estimador de Estados

O estimador de estados é um processo que permite que medir os estados de um sistema a partir de sua entrada e saída. Um estimador deve ser projetado para que ele consiga obter os estados de um sistema o mais rápido possível, dado a entrada e saída no momento anterior. Para isso, fazemos a escolha dos polos do estimador de uma forma que eles reajam mais rápido que o sistema real. Fazemos isso multiplicando a parte real dos polos escolhidos para o controlador por um número, efetivamente afastando eles do zero real. Uma forma de calcular o ganho  $L$  do estimador de estados é muito similar ao algoritmo 4:

---

**Algoritmo 2:** ESTIMADOR DE ESTADOS

---

**Entrada:** Matrizes  $A$  e  $C$  do sistema, polos escolhidos para serem alocados**Saída:** Ganho de realimentação de estados  $L$ **início**

- 1 | Crie uma matriz  $F$  com os autovalores (polos) escolhidos.
- 2 | Selecione um vetor arbitrário  $\hat{l}$  de forma que o par  $(F, \hat{l})$  seja controlável.
- 3 | Solucione o  $T$  único na equação de Lyapunov  $AT - TF = \hat{l}C$ .
- 4 | Calcule o ganho de realimentação  $L = \hat{l}T^{-1}$ .

**fim****retorna**  $\underline{L}$ 

---

Por fim, a equação 2.22 nos retorna os estados.

$$\dot{z} = Az + Bu + (y - y_e)L \quad (2.22)$$

## 2.1.4 Identificação de Sistemas e Estimação de Parâmetros

A maior parte das técnicas de controle aplicadas na indústria dependem do conhecimento de um modelo matemático do sistema a ser controlado. Apresentamos alguns dos modelos na seção 2.1.1 e, nesta seção, apresentaremos alguns dos métodos para obtermos os modelos matemáticos utilizados neste trabalho.

### 2.1.4.1 Visão Geral

A identificação de sistemas e estimação de parâmetros trata de métodos e práticas que permitem construir modelos dinâmicos de um sistema real a partir de experimentos. Muitas vezes, um sistema construído que precisa ser controlado não pode ser modelado devido a limitações matemáticas ou imprecisão na interação dos componentes. Nestes casos, utilizamos um método de identificação de sistemas para obter um modelo matemático. A identificação de sistemas se baseia em testar a resposta do sistema a certas entradas e, a partir das respostas, aproximar o modelo matemático de forma satisfatória.

#### 2.1.4.2 Escolha de sinais e estrutura

A escolha dos sinais de entrada para a identificação de um sistema é importante, pois a identificação deve ser capaz de medir todo o alcance dinâmico do sistema. Se o sinal não for adequado, identificamos um modelo que não é capaz de mostrar o funcionamento do sistema real. Uma condição para a escolha desses sinais é a condição PE (permanent excitement) [9], que dita que enquanto o sistema está submetido ao sinal, ele está constantemente em dinâmica, sem chegar a estabilizar em um ponto. Um sinal que atende essa condição, é o PRBS.

É importante escolher os sinais de interação com o sistema a ser identificado, levando em conta os atuadores presentes e os sensores disponíveis. Os sensores e o tipo de sistema estudado influenciam na escolha do tempo de amostragem de identificação. Em sistemas lentos, como o controle de temperatura de uma sala, o tempo de amostragem pode ser longo, mas em sistemas rápidos, como o controle da velocidade de um veículo, o tempo de amostragem deve ser pequeno para possibilitar melhor controle. A escolha errada do tempo de amostragem pode fazer com que a identificação não funcione, pois limita a quantidade de dados obtidos do sistema e pode impedir a coleta de dados cruciais na mensuração da dinâmica do sistema.

A escolha da ordem do sistema também é de grande importância por causa do tipo de dinâmica que a ordem acarreta.

#### 2.1.4.3 Identificação por Mínimos Quadrados

O método de mínimos quadrados é um dos mais conhecidos e utilizados em várias áreas da ciência e tecnologia. A partir de uma coleção de dados experimentais é possível encontrar um modelo matemático ARX.

Para um sistema SISO em que não conhecemos o seu modelo matemático, podemos descrever o seu comportamento durante o teste como um conjunto de funções entrada-saída como na equação (2.11), pois a partir do conhecimento que temos da modelagem física do sistema, podemos deduzir aproximadamente de quantos valores passados de  $y$  e de  $u$  o nosso sistema estudado depende.

Com os vetores entrada e saída em mãos, montamos uma matriz de regressores  $y$  e  $u$ ,

no seguinte formato:

$$Y = \begin{bmatrix} y(n) & y(n-1) & \dots & y(n-i) \\ y(n+1) & y(n) & \dots & y(n-i+1) \\ \vdots & \vdots & \ddots & \vdots \\ y(n+k) & y(n+k-1) & \dots & y(n+k-i) \end{bmatrix} \quad (2.23)$$

$$U = \begin{bmatrix} u(n) & u(n-1) & \dots & u(n-j) \\ u(n+1) & u(n) & \dots & u(n-j+1) \\ \vdots & \vdots & \ddots & \vdots \\ u(n+k) & u(n+k-1) & \dots & u(n+k-j) \end{bmatrix} \quad (2.24)$$

$$\psi = \begin{bmatrix} Y & U \end{bmatrix} \quad (2.25)$$

Onde  $y(n-i)$  é o primeiro valor do vetor de saídas  $y$  e  $u(n-i)$  é o primeiro valor do vetor de entradas  $u$ ,  $i$  é a quantidade de regressores  $y$  que influenciam na saída  $y$  e  $j$  é a quantidade de regressores  $u$  que influenciam na saída  $y$ ,  $n+k$  é o número de valores do vetor de entrada.

Podemos então, relacionar o vetor de saídas com a matriz de regressores  $\psi$ , da seguinte forma:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} = \psi \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix} \quad (2.26)$$

$$\hat{y} = \psi \hat{\theta}$$

Onde  $\hat{y}$  é um vetor que depende da matriz de regressores  $\psi$  e do vetor  $\hat{\theta}$ . Conhecemos  $\hat{y}$  e  $\psi$ , queremos determinar  $\hat{\theta}$ . Desde que  $X$  seja não singular, é possível determinar o vetor de parâmetros invertendo a matriz:

$$\theta = \psi^{-1}y \quad (2.27)$$

A matriz  $\psi$ , no entanto, não é invertível e, para realizarmos  $\psi^{-1}$ , utilizamos a pseudo

inversa, que será utilizada em diversas partes deste trabalho, representada por  $\{\cdot\}^\dagger$ , onde  $\{\cdot\}$  é a matriz que receberá a operação pseudo inversa.

$$A^\dagger = [A^T A]^{-1} A^T \quad (2.28)$$

E a equação (2.27) se torna:

$$\theta = \psi^\dagger y \quad (2.29)$$

A equação (2.27) é a única equação que satisfaz todos os regressores do sistema de equações formado a partir da equação (2.11). Assumindo que conhecemos  $\hat{\theta}$  e que existe um resíduo  $\xi$  entre o valor observado  $y$  e o valor obtido a partir do vetor de regressores  $\psi$  da forma:

$$y = \psi^T \hat{\theta} + \xi \quad (2.30)$$

Este resíduo é o menor possível devido ao fato de que a equação (2.29) ser demonstravelmente minimizada.

Apresentamos aqui um pseudo algoritmo para a identificação por mínimos quadrados:

---

**Algoritmo 3: IDENTIFICAÇÃO POR MÍNIMOS QUADRADOS**

---

**Entrada:** Vetor de entradas  $U[N]$ , vetor de saídas  $Y[N]$ , quantidade de regressores de  $Y$   $n$ , quantidade de regressores de  $U$   $m$

**Saída:** Vetor  $\hat{\theta}$

**início**

    Faça a matriz de regressores  $\psi$  da equação (2.25)  
    Calcule  $\theta$  com a equação (2.29)

**fim**

**retorna**  $\underline{\theta}$

---

#### 2.1.4.4 Identificação por Subespaços

A identificação por subespaços é um algoritmo que, a partir de uma sequência de dados experimentais, é capaz de gerar um modelo em espaço de estados. O procedimento aqui descrito é explicado com mais detalhes por Rico [11].

Com o sistema a ser identificado em mãos, precisamos executar um ou mais experimentos medindo o sinal de entrada e o sinal da saída do sistema. Para que a identificação funcione adequadamente o sinal de entrada deve atender a condição de persistência de excitação [9]. Um sinal binário pseudo aleatório atende esse requisito.

Aplicamos o sinal no sistema e medimos a sua resposta ao longo de um tempo  $t$  com um tempo de amostragem  $t_s$  e obtemos dois vetores,  $u[N]$ , sinal PRBS de entrada aplicados ao sistema, e  $y[N]$  que é a saída medida do sistema,  $N$  é a quantidade de medidas tomadas. Com esses dois vetores geramos um conjunto de 4 matrizes em blocos de Hankel.

$$U_{0|2i-1} = \frac{\begin{pmatrix} u_0 & u_1 & \dots & u_{j-1} \\ u_1 & u_2 & \dots & u_j \\ \vdots & \vdots & \ddots & \vdots \\ u_{i-1} & u_i & \dots & u_{i+j-2} \\ u_i & u_{i+1} & \dots & u_{i+j-1} \\ u_{i+1} & u_{i+2} & \dots & u_{i+j} \\ \vdots & \vdots & \ddots & \vdots \\ u_{2i-1} & u_{2i} & \dots & u_{2i+j-2} \end{pmatrix}}{U_f} = \frac{U_p}{U_f} \quad (2.31)$$

Onde  $i = nm$ ,  $n$  é a ordem da matriz em blocos de Hankel desejada e  $m$  é o número de entradas, e  $j = N - 2i + 1$ ,  $U_f$  é a matriz em blocos das entradas futuras,  $U_p$  é a matriz em blocos das entradas passados. O mesmo é feito para  $y[N]$  gerando  $Y_f$  e  $Y_p$ . Com essas matrizes em mãos, geramos  $W_p = [U_p \ Y_p]^T$  e podemos encontrar  $O_i$ , fazendo:

$$O_i = Y_{f/U_f} W_p \quad (2.32)$$

$Y_{f/U_f} W_p$  denota uma projeção oblíqua de  $Y_f$  sobre  $W_p$  ao longo de  $U_f$  calculado por:

$$O_i = [Y_f/U_f^\perp][W_p/U_f^\perp]^\dagger W_p \quad (2.33)$$

Em seguida, fazemos uma decomposição SVD de  $O_i$ :

$$O_i = USV^T = \begin{pmatrix} U_1 & U_2 \end{pmatrix} \begin{pmatrix} S_1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix} \quad (2.34)$$

Com  $O_i$  decomposto, calculamos  $\Gamma_i$  da seguinte forma:

$$\Gamma_i = U_1 S_1^{1/2} \quad (2.35)$$



Tiramos a raiz quadrada de uma matriz com o seguinte procedimento:

$$A = QPQ^{-1} \quad (2.36)$$

Onde  $Q$  e  $P$  fazem uma decomposição de  $S_1$  e  $P$  é escolhido de forma que se conheça a sua raiz quadrada.

$$S_1 = (QP^{1/2}Q^{-1})^2 \quad (2.37)$$

$$S_1^{1/2} = QP^{1/2}Q^{-1} \quad (2.38)$$

Com  $\Gamma_i$  em mãos, podemos identificar as matrizes  $A$  e  $C$  do sistema.  $\Gamma_i$  é equivalente à matriz de observabilidade do sistema, então  $C$  é a primeira linha de  $\Gamma_i$  e  $A$  pode ser encontrado por:

$$A = \underline{\Gamma_i^\dagger} \overline{\Gamma_i} \quad (2.39)$$

Onde a matriz sublinhada significa que foi removida a última linha e a matriz sobrelinhada significa que foi removida a primeira linha.

Encontramos  $B$  e  $D$  com a equação:

$$\Gamma_i^\perp Y_f U_f^\dagger = \Gamma_i^\perp H_i^d \quad (2.40)$$

Renomeamos os termos dessa equação da seguinte forma:

$$Q = \begin{pmatrix} D & 0 & 0 & \dots & 0 \\ CB & D & 0 & \dots & 0 \\ CAB & CB & D & \ddots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{i-2}B & CA^{i-3}B & CA^{i-4}B & \dots & D \end{pmatrix} \quad (2.41)$$

$$\begin{pmatrix} M_1 & M_2 & \dots & M_i \end{pmatrix} = \begin{pmatrix} L_1 & L_2 & \dots & L_i \end{pmatrix} Q \quad (2.42)$$

Reescrevemos essa equação:

$$\begin{pmatrix} M_1 \\ M_2 \\ \vdots \\ M_i \end{pmatrix} = \begin{pmatrix} L_1 & L_2 & \dots & L_{i-1} & L_i \\ L_2 & L_3 & \dots & L_i & 0 \\ L_3 & L_4 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ L_i & 0 & \dots & 0 & 0 \end{pmatrix} \begin{pmatrix} I_l & 0 \\ 0 & \underline{\Gamma_i} \end{pmatrix} \begin{pmatrix} D \\ B \end{pmatrix} \quad (2.43)$$

E dessa forma obtemos todas as matrizes do sistema identificado.

**Algoritmos** Para facilitar o entendimento, colocamos em forma de algoritmo os passos que devem ser tomados para identificar um sistema por subespaços.

---

**Algoritmo 4: IDENTIFICAÇÃO POR SUBESPAÇOS**

---

**Entrada:** Vetor de entradas  $U[N]$ , vetor de saídas  $Y[N]$ , ordem determinada do sistema  $n$

**Resultado:** Matrizes do espaço de estados do sistema  $A$ ,  $B$ ,  $C$  e  $D$

- 1 Gerar as matrizes de bloco de Hankel  $Y_p$ ,  $Y_f$ ,  $U_p$  e  $U_f$ , usando a equação (2.31)
- 2 Calcular  $O_i$  usando as equações (2.32) e (2.33)
- 3 Decomposição SVD de  $O_i$  (2.34)
- 4 Calcular  $\Gamma_i$  (2.35)
- 5 Obter  $A$  e  $C$  (2.39)
- 6 Obter  $B$  e  $D$  (2.43)

**retorna**  $A, B, C, D$

---

#### 2.1.4.5 Validação de modelos identificados

Não basta identificar o modelo usando os métodos descritos anteriormente para que ele seja adequado ao sistema real, é preciso validar o modelo obtido usando técnicas capazes de medir o desempenho do modelo.

A técnica que será utilizada é descrita por Aguirre [1], chamada análise de resíduos, onde é feita uma auto correlação dos resíduos, diferença entre a saída real do sistema e a saída do modelo obtido para um mesmo sinal, para verificar a brancura desse sinal. Quanto menos auto correlacionado melhor, pois significa que o resíduo se assemelha a ruído branco.

## 2.2 Trabalhos Relacionados

A identificação de sistemas por caixa-preta, especificamente por subespaços é um tema recorrente do estudo da engenharia, dado que é uma forma de identificar um sistema complexo com menos trabalho mental na modelagem matemática dos fenômenos físicos.

Em 2009, Jernigan et al. [7], publicaram um artigo falando sobre um experimento que fizeram na construção e utilização de um túnel de vento como experiência laboratório remota. Eles escolheram como experimento a levitação de uma bola de praia de plástico usando um soprador de ar. Um grupo de alunos foi encarregado de modelar, simular e controlar o sistema usando o laboratório remoto.

Em 2017, Chacon et al. [5], relatam em um artigo a construção de um túnel de vento para o ensino de engenharia de controle. Eles tem o mesmo objetivo deste trabalho em relação à construir uma plataforma de experimentos barata e reproduzível mas diferem porque aplicam a plataforma para aprendizado à distância. No artigo eles descrevem um processo de identificação caixa preta usando um sinal binário pseudo aleatório bastante parecido com o que será utilizado neste trabalho, mas tratam a identificação como um exercício a ser aplicado aos alunos.

# Capítulo 3

## Modelagem e Simulação

Neste capítulo será descrita a modelagem do sistema, o processo de estimação de velocidade do vento e a simulação do túnel de vento no Simulink Matlab.

### 3.1 Descrição da Planta

O sistema em questão, figura 3.1, se trata de um tubo PVC medindo de 40 mm de diâmetro e 120 cm de comprimento, afixado em uma plataforma plástica através de uma base e de elásticos para estabilização. Na extremidade inferior do tubo, encontra-se uma turbina de avião RC que empurra o ar para dentro do tubo elevando uma bola de tênis de mesa. Na extremidade superior, encontra-se um sensor infravermelho de distância. A turbina é controlada por um Arduino Mega 2560.

### 3.2 Modelo Matemático

Na figura 3.2 vemos que há duas forças atuantes na esfera: a gravidade que a puxa para baixo e a força de empuxo gerada pelo vento. Obtemos a seguinte equação do movimento:

$$m\ddot{h} = F = \frac{1}{2} \cdot C_a \cdot \rho \cdot A \cdot (v_{ar} - \dot{h})^2 - m \cdot g \quad (3.1)$$

onde  $m$  é a massa da esfera,  $h$  é a posição vertical da esfera no tubo,  $\rho$  é a densidade do ar,  $A$  é a área da esfera em contato com o fluxo de ar,  $v_{ar}$  é a velocidade do ar dentro do tubo e  $C_a$  é o coeficiente aerodinâmico da esfera. O coeficiente aerodinâmico depende da velocidade relativa entre a esfera e o vento, mas para as velocidades baixas de

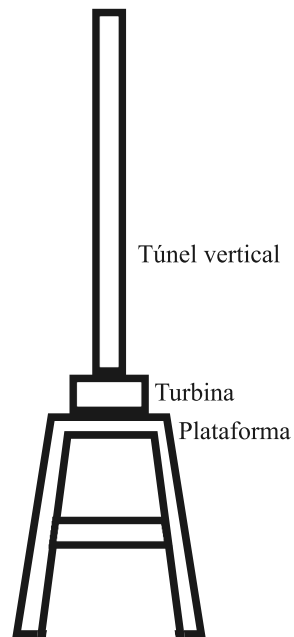


Figura 3.1: Túnel de vento construído

vento que estamos utilizando, esse valor pode ser considerado constante. Consideramos  $\alpha = \frac{1}{2} \cdot C_a \cdot \rho \cdot A$ :

$$\ddot{h} = \frac{\alpha}{m} \cdot (v_{ar} - \dot{h})^2 - g \quad (3.2)$$

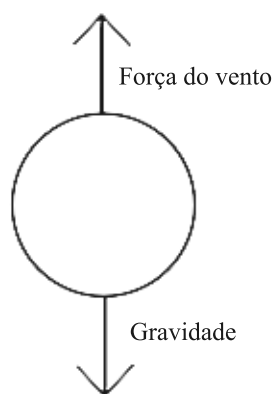


Figura 3.2: Forças atuantes na bola

### 3.3 Estimação da Velocidade do Vento

Para criarmos um simulador do sistema conforme descrito anteriormente, precisamos saber o valor de  $v_{ar}$ . Ele determina a velocidade do vento para diferentes valores de tensão e altura da bola. Não foi possível adquirir um sensor de velocidade do ar devido a problemas de localização geológica, portanto, foi necessário usar diferentes estratégias para adquirir a velocidade do ar em diversas alturas.

Foram adquiridas diversas bolas de tênis de mesa e se injetou uma mistura de cola e água nelas para aumentar o seu peso. Foram escolhidos 6 pesos diferentes e medidos em balança com precisão de duas casas decimais. Então para cada peso foi medida a tensão necessária para que se alcançasse as 6 alturas dentro da região de funcionamento do sensor, 10, 20, 30, 40, 50, 60 cm. Tendo medido a altura e a tensão necessária para alcançar as alturas, bastou utilizar a equação (3.2) fazendo  $\ddot{h} = \dot{h} = 0$ , obtendo-se a equação 3.3 para calcular a velocidade do vento relacionada com uma altura e uma tensão.

$$v_{ar} = \sqrt{\frac{gm}{\alpha}} \quad (3.3)$$

Na figura 3.3, vemos a superfície gerada com os dados medidos. E a partir da função dessa superfície geramos a figura 3.4 e a tabela 3.1.

Podemos ver que a figura 3.4 lembra um gráfico similar ao encontrado em [7] onde o autor utiliza um sensor de velocidade de vento e mostra uma curva similar, para uma mesma tensão aplicada na turbina. Quanto mais elevada se encontra a bola, menor vai ser a velocidade do ar que se choca com ela.

Altura\PWM	1850 ms	1870 ms	1890 ms	1910 ms	1930 ms	1950 ms
10 cm	18.24	21.78	24.94	27.75	30.22	32.37
20 cm	15.02	17.79	20.24	22.38	24.23	25.80
30 cm	13.33	15.52	17.44	19.09	20.49	21.67
40 cm	12.51	14.30	15.86	17.21	18.35	19.30
50 cm	11.88	13.45	14.84	16.05	17.11	18.03
60 cm	10.76	12.29	13.68	14.95	16.10	17.16

Tabela 3.1: Tabela da velocidade do ar em função de altura e do PWM aplicado ao motor

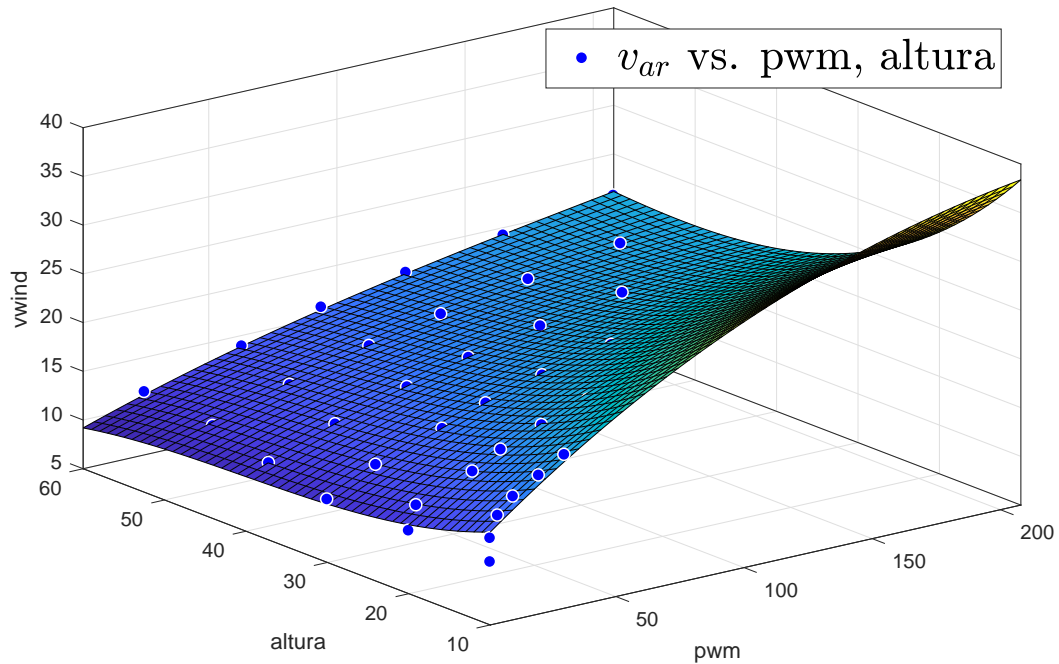


Figura 3.3: Superfície de  $v_{ar}$  em função de altura e pwm

### 3.4 Simulação do Túnel de Vento no Simulink

Criamos um modelo de simulação( figura 3.5) a partir da equação 3.2 e utilizamos os dados adquiridos em 3.4 para fazer um ajuste de curva que alimenta o modelo com os valores da velocidade do vento para determinada altura e tensão no motor.

Na figura 3.6, vemos como a resposta do simulador não representa o sistema real. O tempo de estabilização é 10 vezes maior que o tempo do sistema, mas o sobrevalor está condizente com o medido.

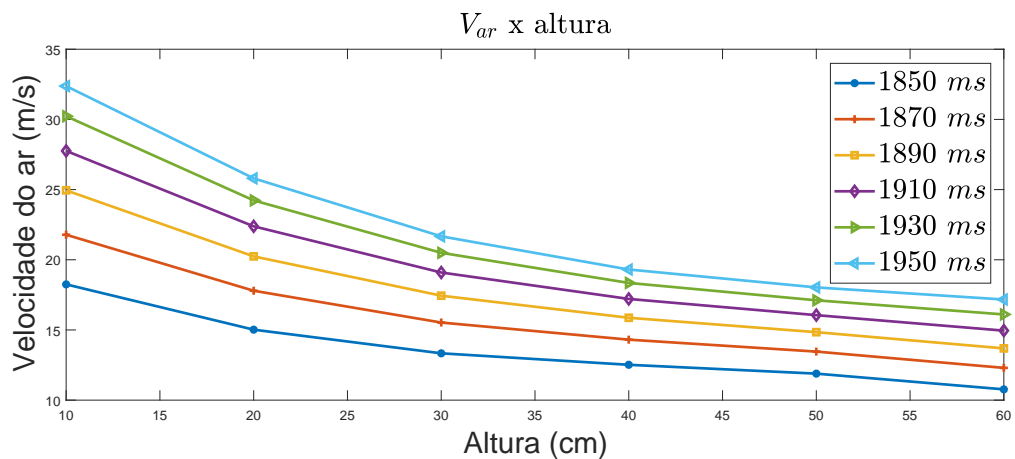


Figura 3.4: Gráfico de  $v_{ar}$  x altura

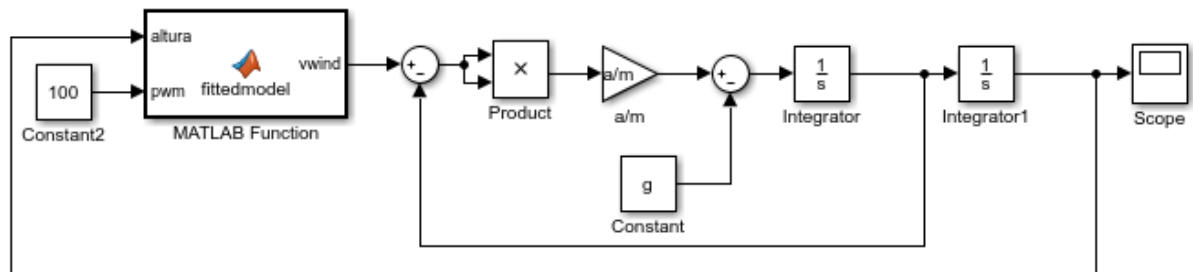


Figura 3.5: Simulador do túnel de vento

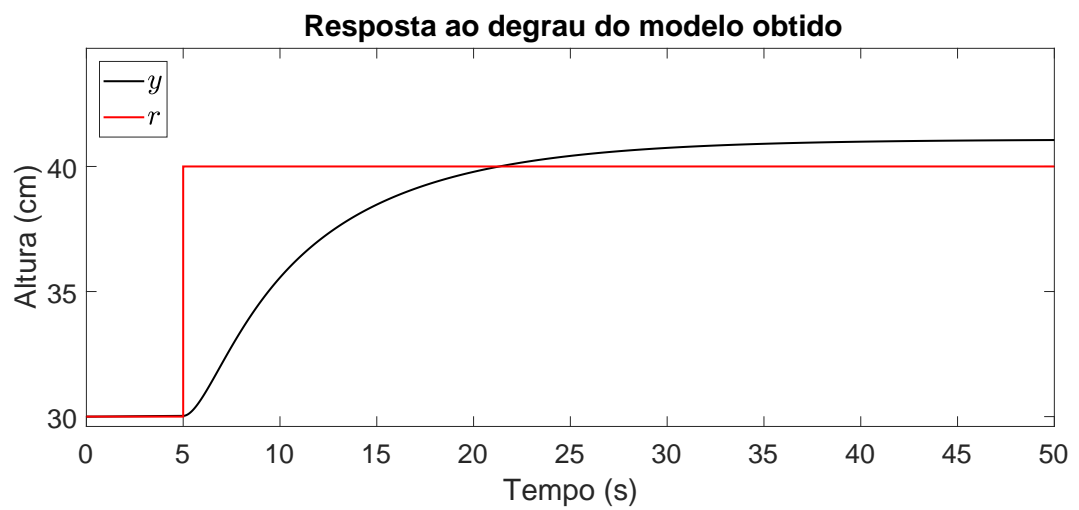


Figura 3.6: Resposta ao degrau do simulador



# Capítulo 4

## Identificação do Túnel de Ar

Neste capítulo, será explicada a identificação do sistema do túnel de ar. Para fazer a identificação caixa-preta de um sistema é necessário fazer um estudo prévio do funcionamento dele, conhecer entradas e saídas, e obter um modelo matemático, se possível, mesmo sem ter todos os parâmetros.

### 4.1 Escolha de estrutura

Para a identificação do sistema, escolhemos fazer um teste que atenda a condição PE [9] usando um sinal de entrada PRBS e, como saída, a altura da bola medida pelo sensor infravermelho. Existem vários métodos para fazer a escolha da estrutura de identificação de um sistema, como o critério de informação de Akaike [2]. Entretanto, decidimos escolher a ordem do sistema fazendo uma análise visual da autocorrelação dos resíduos e da resposta ao degrau do modelo identificado.

Os sinais de entrada e saída do sistema foram normalizados em torno do ponto de operação definido a 30 cm de altura. Onde o -1 da saída corresponde a 20 cm e 1 corresponde a 40 cm. O sinal aplicado ao atuador foi normalizado para que 1 correspondesse ao PWM necessário para que o sistema estabilizasse em 40 cm e o -1 correspondesse ao PWM que estabilizasse o sistema em 20 cm.

#### 4.1.1 Mínimos Quadrados

A identificação por Mínimos Quadrados gera um sistema do tipo ARX como visto na equação (2.11). A escolha da estrutura, neste caso, se dá escolhendo uma quantidade de

regressores da saída e uma quantidade de regressores da entrada.

Para a escolha da estrutura do modelo do sistema, foi feito o seguinte procedimento:

1. Escolha de uma quantidade de regressores de  $y$
2. Escolha de uma quantidade de regressores de  $u$
3. Identificação por Mínimos Quadrados com os regressores de  $y$  e  $u$
4. Análise de autocorrelação dos resíduos  $\xi = y - \Psi\hat{\theta}$

Este procedimento foi repetido para valores de 1 a 10 para ambos os regressores, a fim de escolhermos quantos zeros e polos precisaria para uma estrutura que tem a seguinte forma:

$$Y[k] = \frac{A[k]}{B[k]} \quad (4.1)$$

$$A[k] = a_1U[k-1] + a_2U[k-2] + \cdots + a_nU[k-n] \quad (4.2)$$

$$B[k] = 1 + b_1Y[k-1] + b_2Y[k-2] + \cdots + b_mY[k-m] \quad (4.3)$$

Onde  $n$  é o número de zeros e  $m$  é o número de polos. Ao final, foram identificadas duas ordens de sistema que serão comparadas posteriormente, uma com 3 polos e 3 zeros, modelo *ARX1*, e uma com 5 polos e 5 zeros, modelo *ARX2*. Escolhemos dois modelos *ARX* para fazer uma comparação entre o desempenho de um modelo de ordem menor e um modelo de ordem maior, obtidos do mesmo sistema. Geramos um terceiro modelo *ARX*, aplicando a identificação no simulador, gerando um modelo, *ARXsim*, com 3 polos e 3 zeros para comparação.

### 4.1.2 Subespaços

A identificação por Subespaços gera um sistema em espaço de estados como visto na equação (2.6). Neste tipo de identificação, a escolha da estrutura é feita decidindo a ordem do sistema e a ordem da matriz em blocos de Hankel. E, similarmente ao procedimento feito na seção 4.1.1, foram variadas as ordens do sistema e da matriz em blocos de Hankel. Foi escolhida ordem 3 para o sistema e ordem 15 para a matriz em blocos de Hankel, usada para identificar o sistema. Vemos na figura 4.4 a auto correlação dos resíduos do sistema identificado.

## 4.2 Experimento

O experimento para identificação precisa de um sinal que atenda a condição PE. Para tanto foi gerado um sinal PRBS. O sinal foi aplicado ao sistema através do Arduino e os sinais foram medidos com um tempo de amostragem de 50 ms.

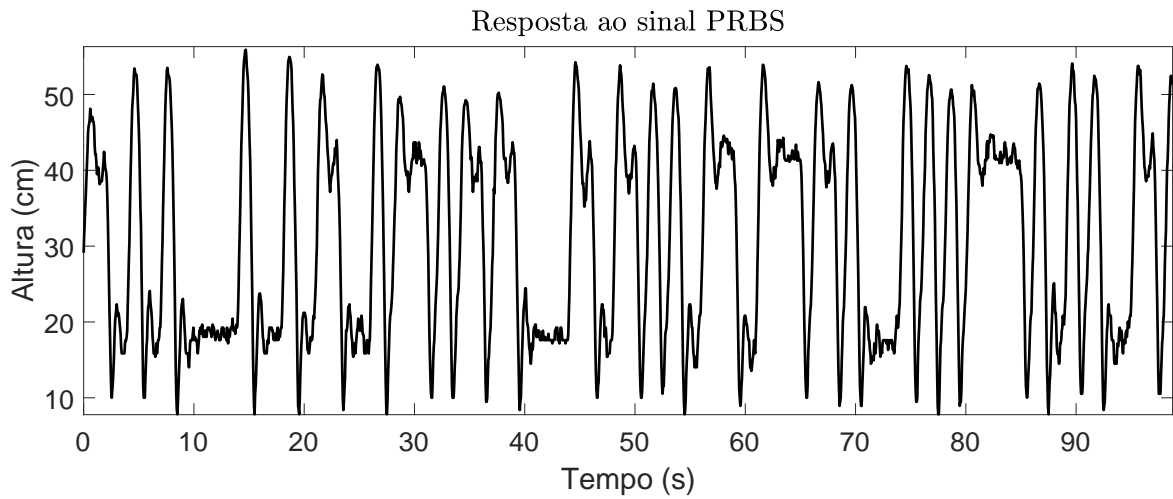


Figura 4.1: Gráfico da saída do sistema ao aplicar o sinal PRBS com tempo de amostragem de 50 ms

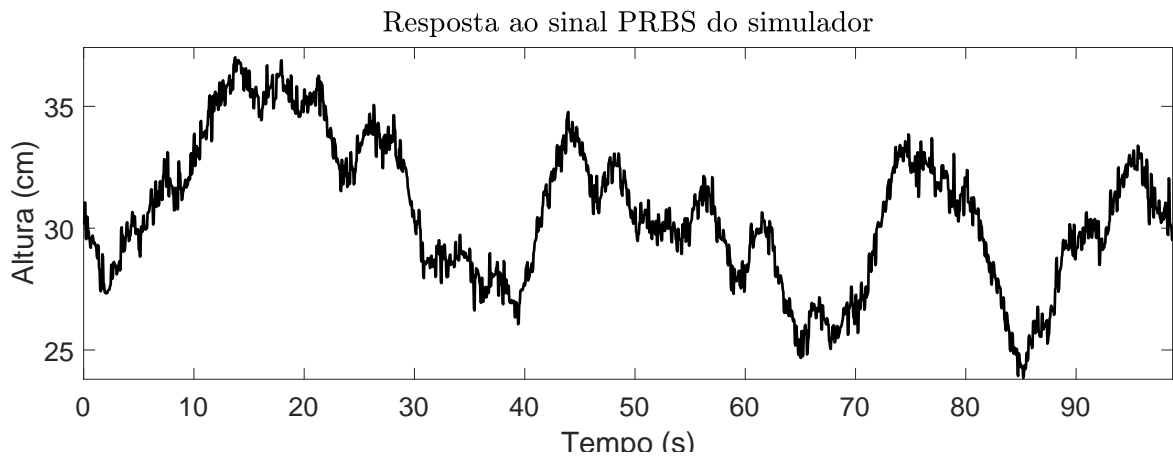


Figura 4.2: Gráfico da saída do simulador ao aplicar o sinal PRBS com tempo de amostragem de 50 ms

A figura 4.1, mostra a resposta do sistema ao sinal PRBS que foi aplicado ao sistema, e a figura 4.2 mostra a resposta do simulador. Vemos que a resposta do modelo simulado é muito diferente do sistema real.

## 4.3 Estimação

Com a resposta ao sinal PRBS em mãos, podemos identificar o sistema, usando o método dos mínimos quadrados e o método de identificação por subespaços.

### 4.3.1 Mínimos Quadrados

Para fazer a identificação por mínimos quadrados geramos a matriz de regressores  $\psi$  da equação (2.25) e usamos a equação (2.27) para obter os coeficientes dos regressores. Obtivemos os coeficientes do modelo *ARX1*:

$$a_1 = 0.0017 \quad a_2 = 0.0011 \quad a_3 = 0.0550 \quad (4.4)$$

$$b_1 = -1.6309 \quad b_2 = 0.5828 \quad b_3 = 0.0965 \quad (4.5)$$

Obtivemos os coeficientes do modelo *ARX2*:

$$a_1 = -0.0029 \quad a_2 = -0.0040 \quad a_3 = 0.0143 \quad a_4 = 0.0057 \quad a_5 = 0.0725 \quad (4.6)$$

$$b_1 = -1.5506 \quad b_2 = 0.7601 \quad b_3 = -0.5321 \quad b_4 = 0.4854 \quad b_5 = -0.0897 \quad (4.7)$$

Obtivemos os coeficientes do modelo *ARXsim*:

$$a_1 = -0.0037 \quad a_2 = 0.0017 \quad a_3 = 0.0066 \quad (4.8)$$

$$b_1 = -1.0108 \quad b_2 = 0.4531 \quad b_3 = -0.4341 \quad (4.9)$$

### 4.3.2 Subespaços

Para identificar o sistema usando subespaços, utilizamos o algoritmo mostrado na seção 2.1.4.4 com uma matriz em blocos de Hankel de ordem 15 para encontrar um sistema de ordem 3. Identificamos o seguinte modelo, *SUB1*, com o formato da equação (2.6):

$$A = \begin{bmatrix} 0.9761 & 0.1933 & -0.0438 \\ -0.1817 & 0.9841 & -0.1489 \\ 0.0840 & 0.3107 & 0.6994 \end{bmatrix} \quad (4.10)$$

$$B = \begin{bmatrix} 0.1466 \\ 0.2515 \\ 0.8460 \end{bmatrix} \quad (4.11)$$

$$C = \begin{bmatrix} -1.0104 & -0.3354 & 0.2496 \end{bmatrix} \quad (4.12)$$

$$D = \begin{bmatrix} 0.0010 \end{bmatrix} \quad (4.13)$$

## 4.4 Validação

Ao identificar um sistema precisamos validar o modelo obtido para garantir que ele seja adequado para representar o sistema real. Para isso, fizemos uma análise da auto correlação dos resíduos  $\xi = y - \Psi\hat{\theta}$  para o ARX e  $\xi = y - y_{sim}$  para o espaço de estados. Vemos nas figuras 4.3, 4.4 e 4.6, que os resíduos do modelo ARX são muito menos correlacionados do que os do modelo em espaço de estados. Isso acontece porque para o modelo ARX estamos analisando os seus regressores com a matriz  $\Psi$  que os gera, o que retorna uma resposta muito mais próxima do sistema. Já para o espaço de estados, estamos analisando a simulação do sistema, que apresenta pequenas diferenças em relação ao sistema real, como a ausência de ruído.

Olhamos também a resposta ao degrau do sistema (figuras 4.7, 4.8, 4.9 e 4.10) e vemos que para os dois sistemas identificados, ela tem uma oscilação similar ao sistema real. Concluimos, através da análise da auto correlação dos resíduos e da análise da resposta ao degrau dos sistemas identificados, que os modelos são adequados para mostrar o funcionamento do sistema.

Vemos nas figuras 4.3, 4.4 e 4.5 que os resíduos dos modelos identificados tem uma autocorrelação que indica que eles assemelham um ruído branco, e que os modelos identificados são fiéis aos dados inseridos. O que não quer dizer que eles são adequados para representar o sistema.

Nas figuras 4.7 e 4.8, vemos que a resposta ao degrau dos modelos *ARX1* e *ARX2* são adequadas para representar o modelo. Ambas tem um tempo de assentamento similar ao observado no sistema e uma dinâmica muito parecida. Já na figura 4.9, vemos que tanto o tempo de assentamento quanto a dinâmica representada, estão diferentes do sistema real.

Para o caso do modelo *SUB1*, a autocorrelação dos resíduos mostrada na figura 4.6,

indica uma certa polarização do modelo. Entretanto, isso ocorre porque estamos fazendo uma análise da resposta simulada do modelo e não dos seus regressores, aos quais não temos acesso.

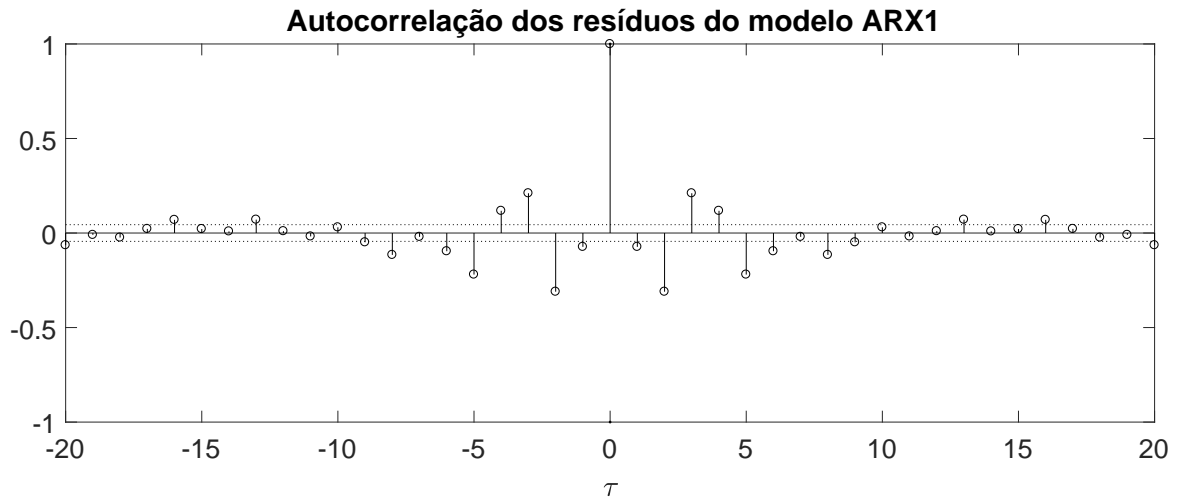


Figura 4.3: Autocorrelação dos resíduos do modelo  $ARX1$

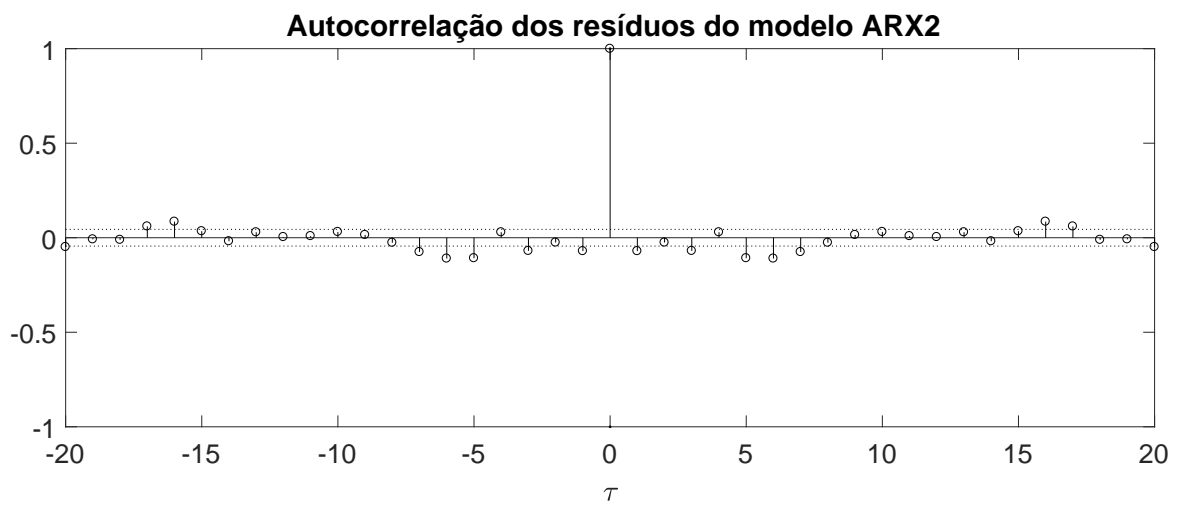
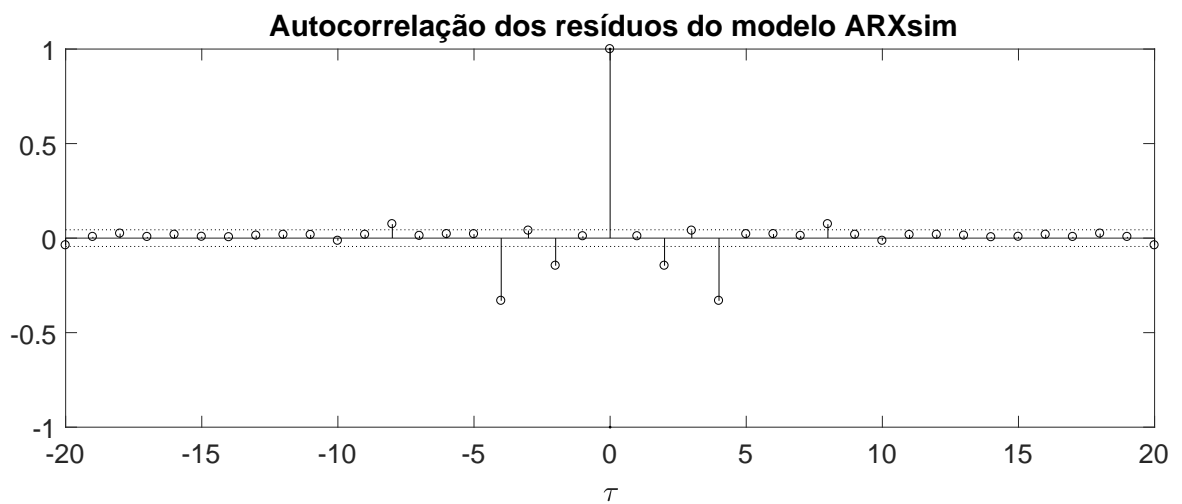
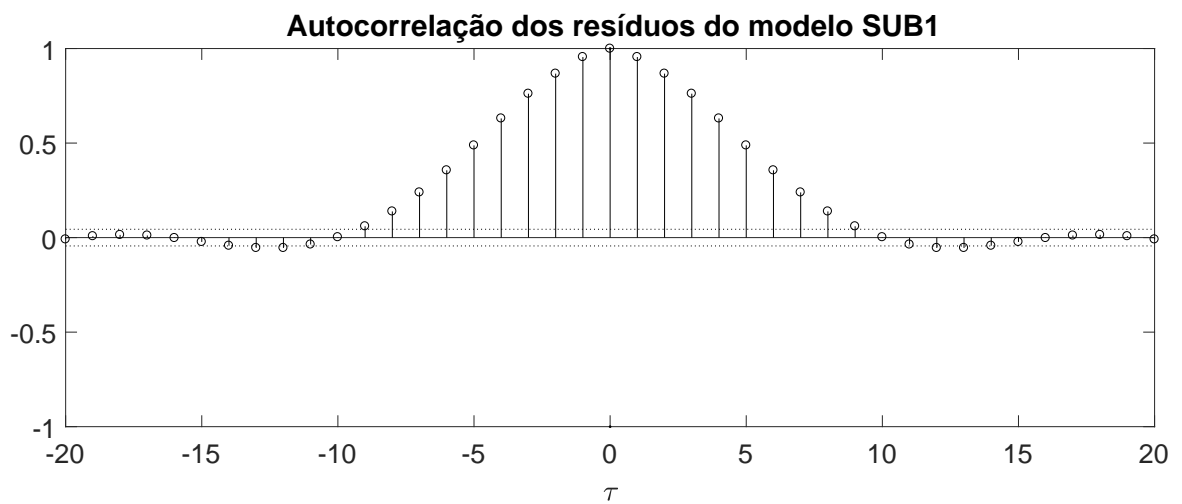
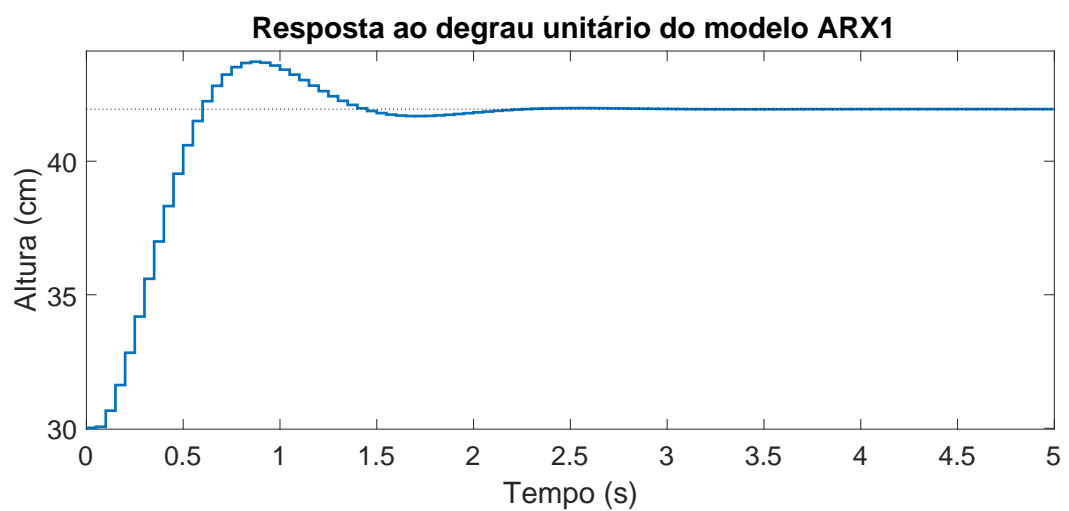
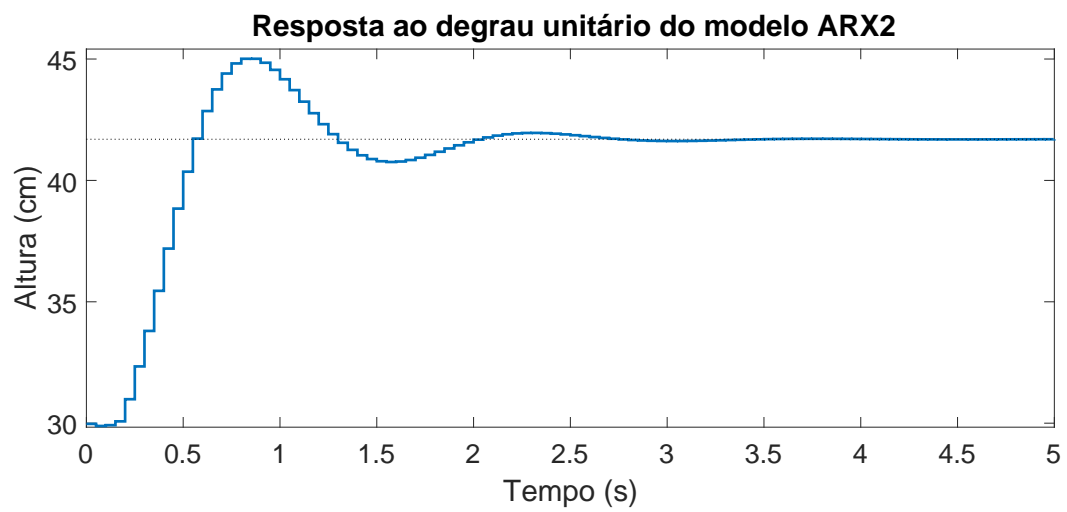
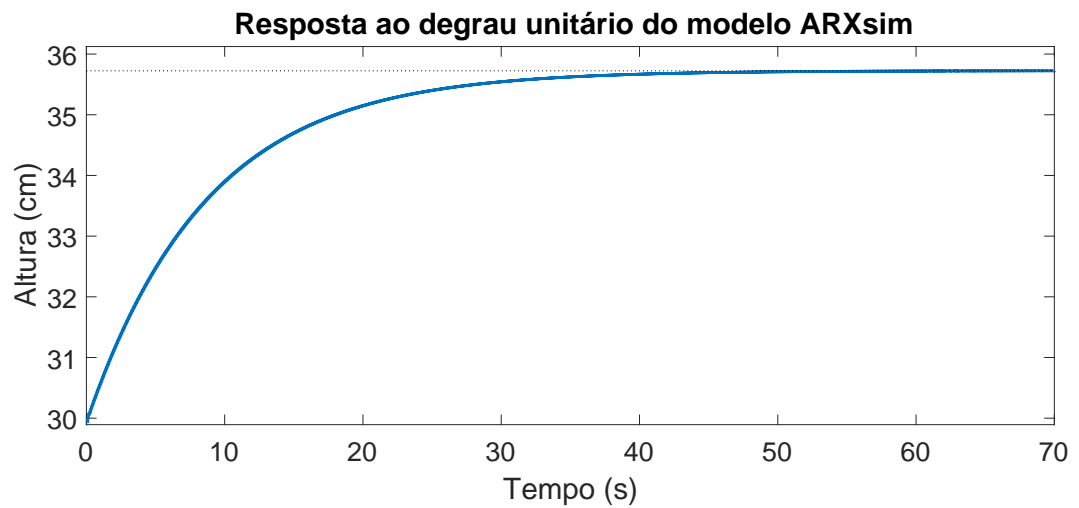
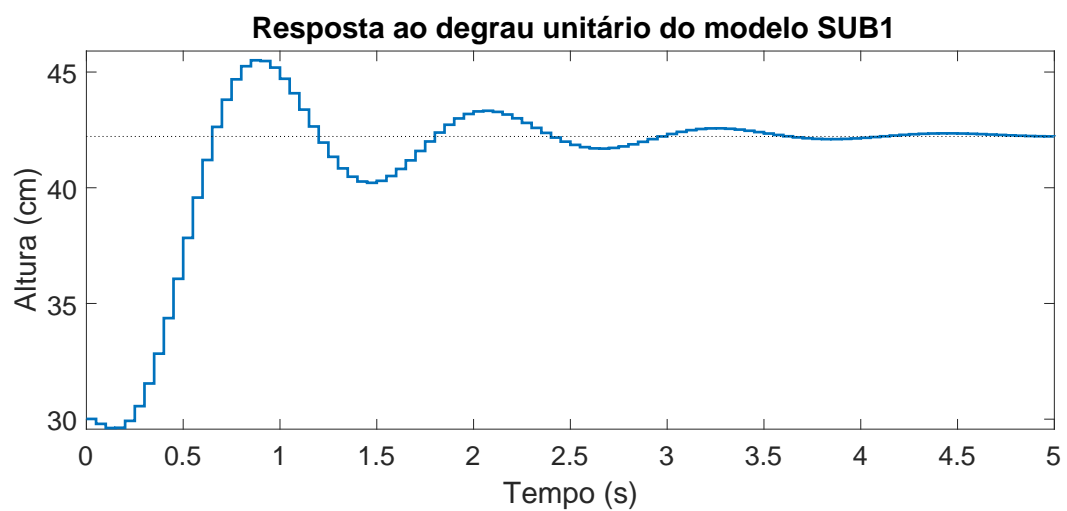


Figura 4.4: Autocorrelação dos resíduos do modelo  $ARX2$

Figura 4.5: Autocorrelação dos resíduos do modelo *ARXsim*Figura 4.6: Autocorrelação dos resíduos do modelo *SUB1*Figura 4.7: Resposta ao degrau unitário do modelo *ARX1*

Figura 4.8: Resposta ao degrau unitário do modelo  $ARX2$ Figura 4.9: Resposta ao degrau unitário do modelo  $ARXsim$ Figura 4.10: Resposta ao degrau unitário do modelo  $SUB1$



## Capítulo 5

# Projeto de Controlador com Alocação de Polos Baseado nos Modelos Identificados

Com o sistema identificado, podemos finalmente controlá-lo. Faremos isso através de realimentação de estados para os quatro modelos identificados.

### 5.1 Alocação de Polos com Realimentação de Estados

Para fazer o controle por realimentação de estados, precisamos passar os 3 modelos ARX obtidos para modelos em espaço de estados. Fazemos isso usando a função do Matlab `tf2ss`, onde entramos com o nominador e denominador do modelo ARX e a função nos retorna as matrizes A, B, C e D do sistema em espaço de estados.

Queremos controlar os 4 modelos fazendo alocação de polos através da realimentação de estados. Projetamos os controladores para que atendam como requisitos um sobrevalor menor que 4% do valor final e tempo de assentamento menor que 2 segundos quando recebe uma entrada do tipo degrau unitário.

#### 5.1.1 Modelo *SUB1*

Para atender os requisitos de tempo de assentamento  $t_s < 2\text{ s}$  e sobrevalor máximo de  $ovs < 4\%$  calculamos os valores de  $\zeta$  e  $\omega_n$  usados na equação característica de um sistema de segunda ordem (5.1), usando as equações (5.2) e (5.3).

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (5.1)$$

$$ovs = e^{\frac{\zeta\pi}{\sqrt{1-\zeta^2}}} < 4\% \quad (5.2)$$

$$t_s = \frac{4}{\zeta\omega_n} < 2 \text{ s} \quad (5.3)$$

Encontramos  $\zeta > 0.7157$  e  $\omega_n > 2.7945$ , e escolhemos os valores de  $\zeta = 0.9$  e  $\omega_n = 4$ . Ao aplicar esses valores na equação (5.1) obtivemos os polos que precisávamos alocar no domínio  $s$ ,  $s_1 = -3.6000 + 1.7436i$  e  $s_2 = -3.6000 - 1.7436i$ . Como o modelo *SUB1* tem ordem 3, precisamos de um terceiro polo distante da parte real dos anteriores. Escolhemos  $s_3 = -8$ . Passamos os três polos para o domínio  $z$  e obtivemos  $z_1 = 0.8321 + 0.0727i$ ,  $z_2 = 0.8321 - 0.0727i$  e  $z_3 = 0.6703$ .

Para fazer a alocação de polos, usamos a equação de Lyapunov (5.4):

$$(A - B\bar{k})T = TF \quad (5.4)$$

Onde  $A$  e  $B$  são matrizes do sistema,  $\bar{k}$  é um vetor arbitrário escolhido de forma que o par  $F$  e  $\bar{k}$  seja observável,  $F$  é uma matriz em blocos diagonais com os polos que se deseja alocar e  $k = \bar{k} * T^{-1}$  nos retorna o vetor de realimentação de estados  $k$ . Obtemos  $k = [0.7552, 0.2637, 0.1750]$ . Com essa realimentação de estados, obtivemos a seguinte resposta ao degrau:

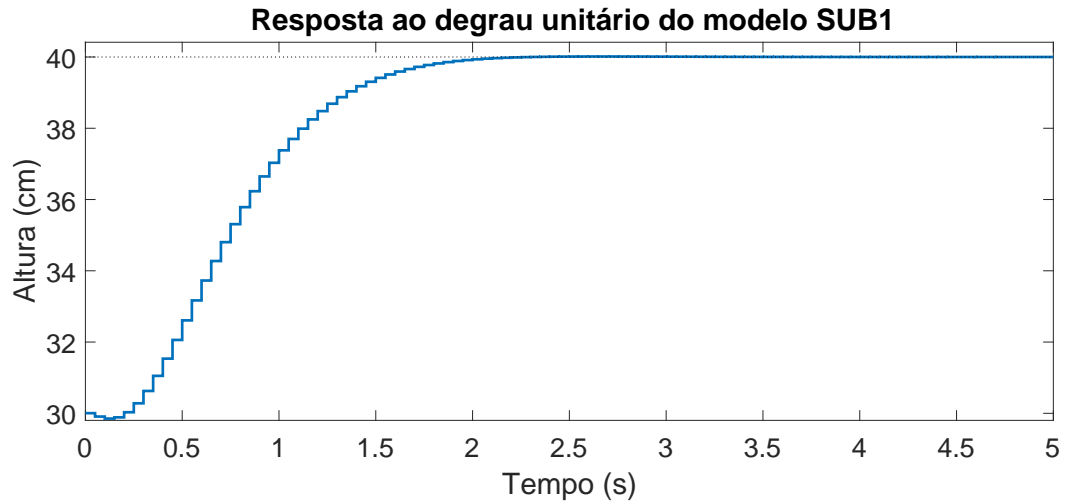


Figura 5.1: Resposta ao degrau unitário do modelo simulado *SUB1* controlado

A resposta ao degrau do sistema, como vista na figura 5.1, apresentou um tempo de assentamento de 1.8 segundos e um sobrevalor de 0.14%.

### 5.1.2 Modelo *ARX1*

Para o modelo *ARX1*, escolhemos os valores de  $\zeta = 0.9$  e  $\omega_n = 7$ . Ao aplicar esses valores na equação (5.1), obtivemos os polos que precisamos alocar no domínio  $s$ ,  $s_1 = -6.3000 + 3.0512i$  e  $s_2 = -6.3000 - 3.0512i$ , para o polo distante alocamos  $s_3 = -15$ . Que no domínio  $z$  são  $z_1 = 0.7213 + 0.1109i$ ,  $z_2 = 0.7213 - 0.1109i$  e  $z_3 = 0.4966$ . Repetimos o mesmo procedimento de alocação de polos usado no modelo *SUB1* e obtivemos  $k = [-0.2840, 0.6313, -0.3481]$ .

O sistema realimentado teve a seguinte resposta ao degrau:

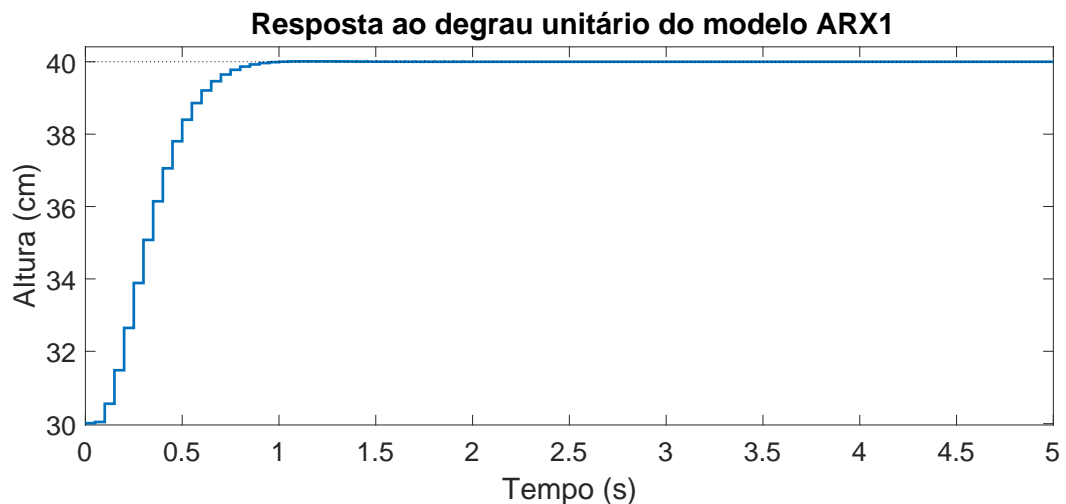


Figura 5.2: Resposta ao degrau do modelo simulado *ARX1* controlado

A resposta ao degrau da figura 5.2 apresenta um tempo de assentamento de 0.8 segundos e um sobrevalor de 0,125%.

### 5.1.3 Modelo $ARX2$

Para o modelo  $ARX2$ , escolhemos  $\zeta = 0.9$  e  $\omega_n = 9$ . Aplicando na equação (5.1) obtivemos os polos para alocar no domínio  $s$ ,  $s_1 = -8.1000 + 3.9230i$  e  $s_2 = -8.1000 - 3.9230i$ . Este modelo ainda tem 3 polos extras que devem ser alocados distantes dos polos  $s_1$  e  $s_2$ , escolhemos  $s_3 = -14$ ,  $s_4 = -14.1$  e  $s_5 = -14.2$ . Que no domínio  $z$  são  $z_1 = 0.6542 + 0.1300i$ ,  $z_2 = 0.6542 - 0.1300i$ ,  $z_3 = 0.4966$ ,  $z_4 = 0.4941$  e  $z_5 = 0.4916$ . Fazendo a alocação de polos da mesma forma que foi feita para o modelo  $SUB1$ , obtivemos  $k = [-1.2401, 2.3566, -1.2062, -0.0018, 0.0360]$ .

O sistema controlado teve a seguinte resposta ao degrau unitário:

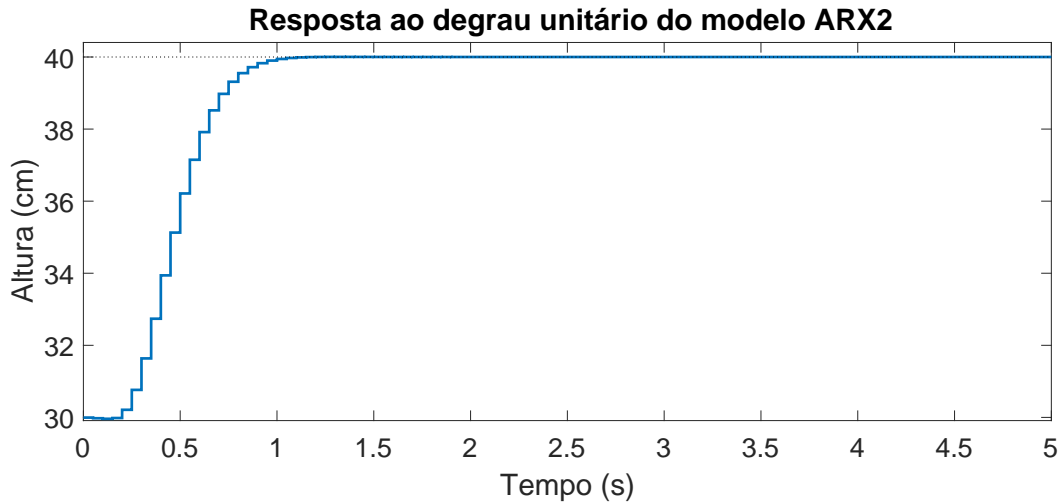


Figura 5.3: Resposta ao degrau do modelo simulado  $ARX2$  controlado

Vemos na figura 5.3, que a resposta ao degrau unitário do sistema controlado teve tempo de assentamento de 0.9 segundos e sobrevalor de 0.0326%.

### 5.1.4 Modelo $ARXsim$

Para o modelo  $ARXsim$ , escolhemos  $\zeta = 0.9$  e  $\omega_n = 9$ . Aplicados na equação (5.1) obtivemos os polos no domínio  $s$ ,  $s_1 = -3.6000 + 1.7436i$  e  $s_2 = -3.6000 - 1.7436i$ , e escolhemos o polo distante  $s_3 = -8$ . No domínio  $z$  os polos são  $z_1 = 0.8321 + 0.0727i$ ,  $z_2 = 0.8321 - 0.0727i$  e  $z_3 = 0.4966$ . Fizemos a alocação de polos como foi feito para o

modelo *SUB1* e obtivemos  $k = [-1.1499, 1.0710, 0.0876]$ . O sistema controlado teve a seguinte resposta ao degrau unitário:

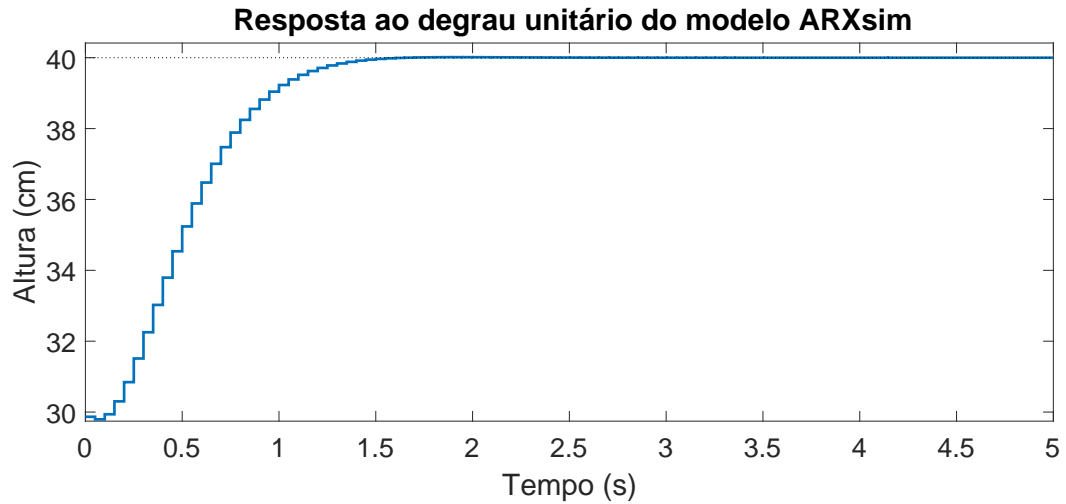


Figura 5.4: Resposta ao degrau do modelo simulado *ARXsim* controlado

Vemos na figura 5.4, que a resposta ao degrau unitário do sistema controlado teve tempo de assentamento de 1.3 segundos e máximo sobrevalor de 0.1487%.

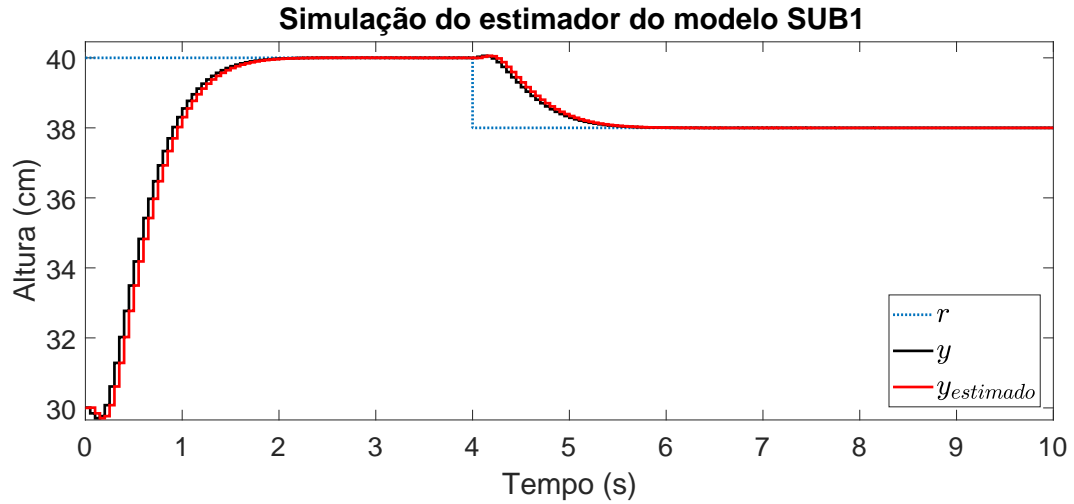
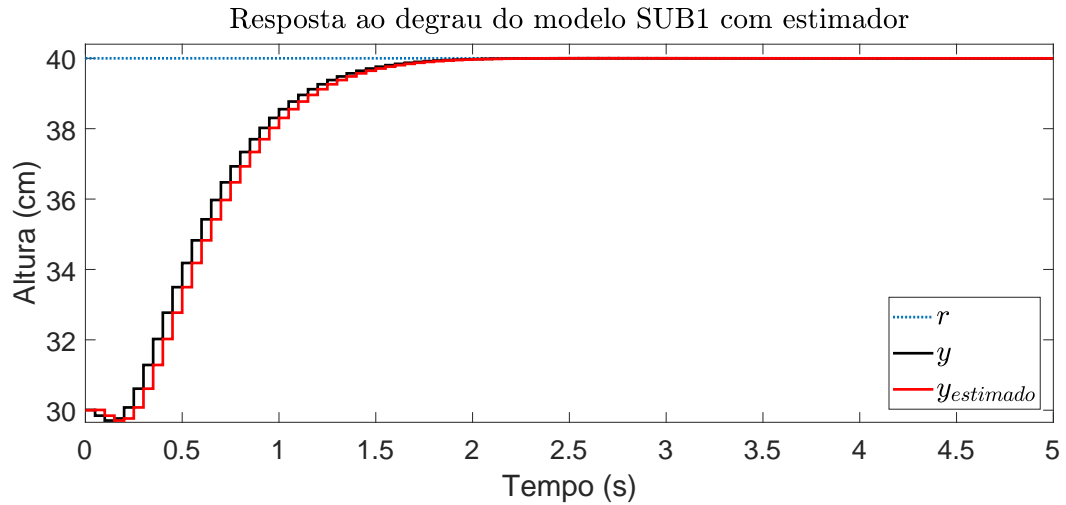
## 5.2 Projeto do Estimador de Estados

Para realizar um controlador por realimentação de estados, precisamos ser capazes de medir todos os estados do nosso modelo a cada tempo de amostragem. O nosso sistema, no entanto, não tem sensores para medir cada um dos estados necessários. A única medida disponível é a saída do sistema na forma da altura da bola. Portanto, precisamos implementar um estimador de estados para cada um dos controladores projetados na subseção anterior.

### 5.2.1 Modelo *SUB1*

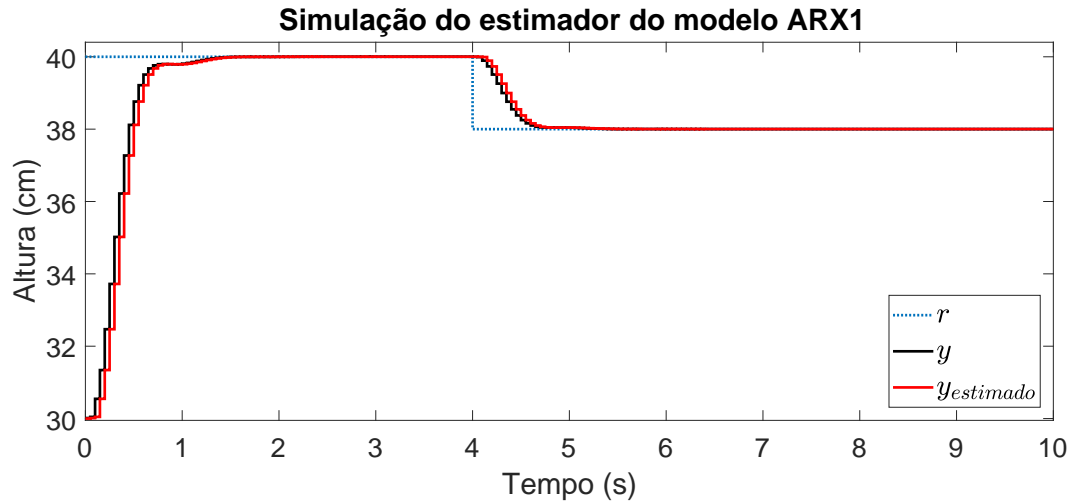
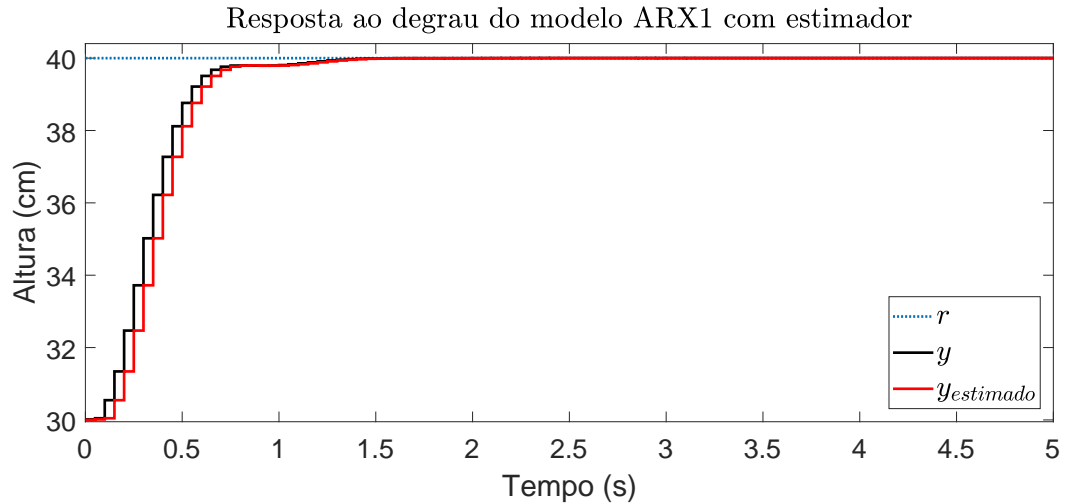
Um estimador de estados é projetado de forma similar ao controlador por realimentação de estados, precisamos escolher polos adequados para que o estimador funcione da forma correta. A forma mais simples de escolher esses polos é fazer com que eles reajam mais rápido que o sistema à entrada recebida. Conseguimos isso, multiplicando a parte real dos polos no domínio  $s$  por um número para que fossem mais rápidos.

Projetamos o estimador de estados com os seguintes polos  $z_1 = 0.1647 + 0.0144i$ ,  $z_2 = 0.1647 - 0.0144i$  e  $z_3 = 0.0183$  e obtivemos o ganho do estimador  $L = [3.5436, -7.2192, 13.9077]^T$ . Nas figuras 5.5 e 5.6, vemos como uma simulação do sistema se comportou.

Figura 5.5: Simulação do estimador do modelo *SUB1*Figura 5.6: Resposta ao degrau do modelo *SUB1* controlado com observador

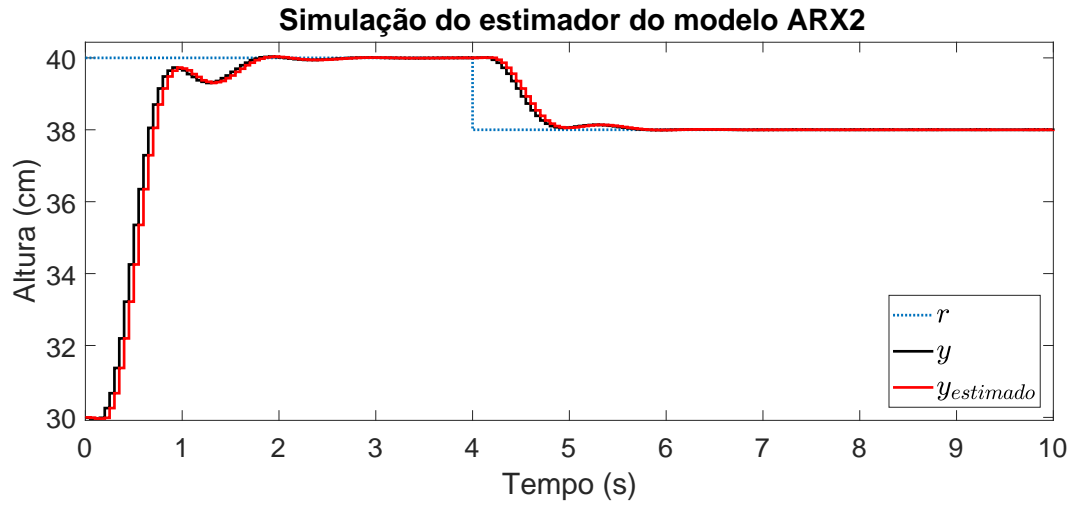
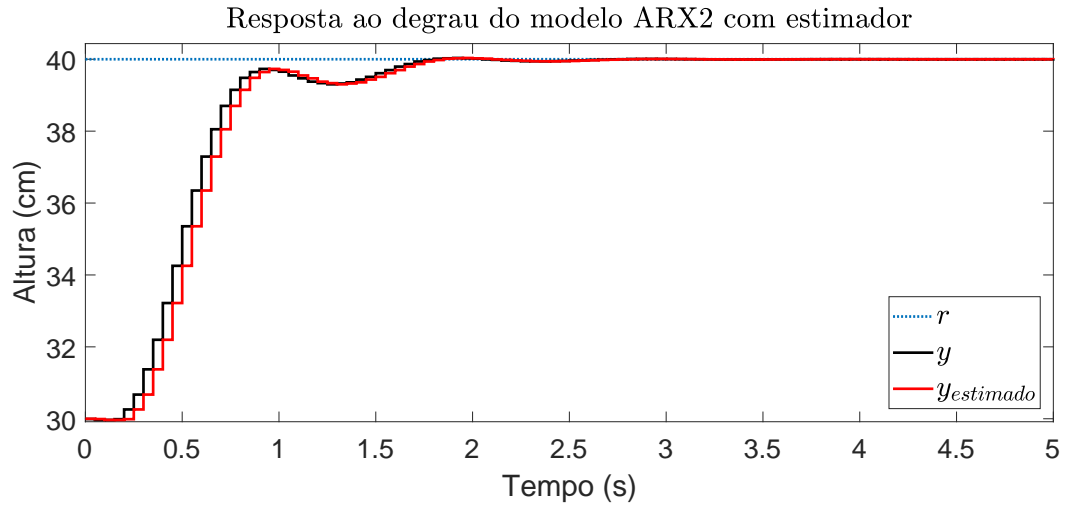
### 5.2.2 Modelo *ARX1*

Para o estimador do modelo *ARX1*, alocamos os polos em  $z_1 = 0.0309 + 0.0048i$ ,  $z_2 = 0.0309 - 0.0048i$  e  $z_3 = 0.0003$ . Obtivemos o ganho do estimador  $L = [33.9534, 26.6709, 16.6024]^T$ . Nas figuras 5.7 e 5.8, vemos como uma simulação do sistema se comportou.

Figura 5.7: Simulação do estimador do modelo  $ARX1$ Figura 5.8: Resposta ao degrau do modelo  $ARX1$  controlado com observador

### 5.2.3 Modelo $ARX2$

Para o estimador do modelo  $ARX2$ , alocamos os polos em  $z_1 = 0.0304 + 0.0074i$ ,  $z_2 = 0.0304 - 0.0074i$ ,  $z_3 = 0.0074$ ,  $z_4 = 0.0072$  e  $z_5 = 0.0069$ . Obtivemos o ganho do estimador  $L = [22.1925, 20.4118, 16.9038, 16.9671, 11.1584]^T$ . Nas figuras 5.9 e 5.10, vemos como uma simulação do sistema se comportou.

Figura 5.9: Simulação do estimador do modelo  $ARX2$ Figura 5.10: Resposta ao degrau do modelo  $ARX2$  controlado com observador

#### 5.2.4 Modelo $ARXsim$

Para o estimador do modelo  $ARX2$ , alocamos os polos em  $z_1 = 0.3383 + 0.0296i$ ,  $z_2 = 0.3383 - 0.0296i$  e  $z_3 = 0.0150$ . Obtivemos o ganho do estimador  $L = [27.5247, 73.1490, 146.4144]^T$ . Nas figuras 5.11 e 5.12, vemos como uma simulação do sistema se comportou.



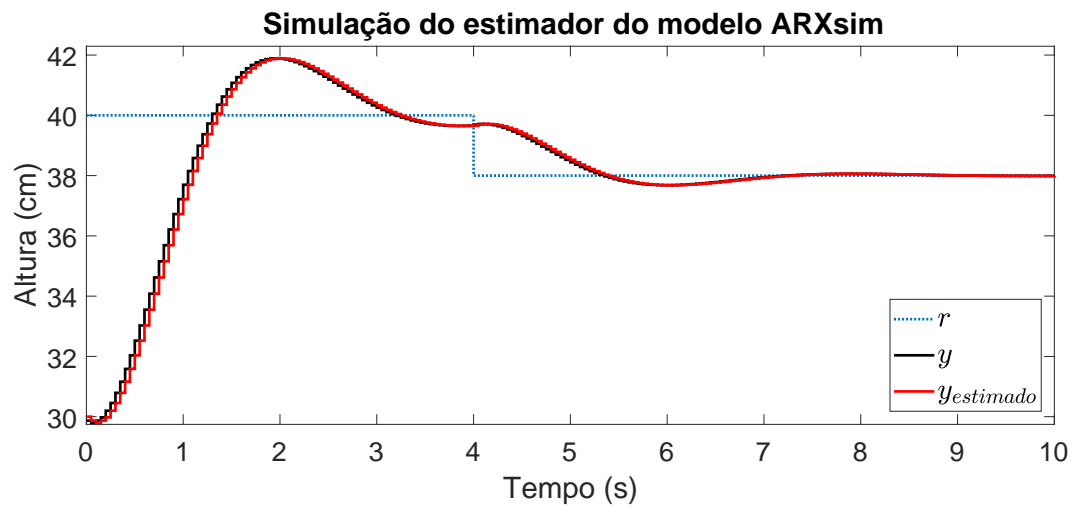


Figura 5.11: Simulação do estimador do modelo *ARXsim*

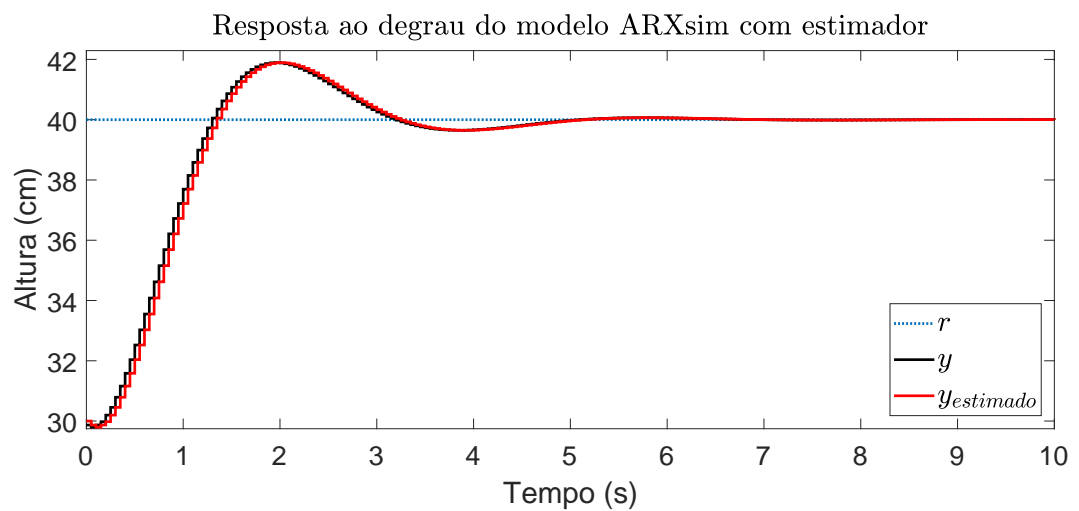


Figura 5.12: Resposta ao degrau do modelo *ARXsim* controlado com observador

## Capítulo 6

# Avaliação Experimental do Sistema de Controle

Após projetar os controladores, fizemos uma avaliação do seu funcionamento, aplicando cada um no sistema real.

### 6.1 Descrição e objetivos experimentais

Os experimentos descritos neste capítulo tiveram como objetivo, verificar e avaliar o funcionamento do sistema ao ser controlado pelos controladores projetados no capítulo 5. Foram executados 4 experimentos no sistema que funcionam da seguinte forma:

**Resposta ao Degrau Unitário** A resposta ao degrau unitário é um experimento onde o sistema recebe uma referência unitária e reage a ela. Com este experimento verificamos se o sistema se mantinha estável ao ser controlado e se ele atendia aos requisitos de tempo de assentamento e máximo sobrevalor definidos quando o controlador foi projetado.

**Resposta à escadaria** A resposta à escadaria é um experimento onde o sistema como entrada uma referência que muda de nível em intervalos de tempo, para que possamos verificar se o controlador é capaz de seguir diferentes níveis de referência.

**Teste de Robustez à variação paramétrica** O teste de robustez à mudança de parâmetros serve para determinar se o controlador é capaz de reagir quando algum

parâmetro é alterado. Neste experimento, alteramos o peso da bola e verificamos se o sistema era capaz de responder ao degrau unitário.

**Teste de Robustez à Perturbação Externa** O teste de robustez à perturbação externa mostra se o sistema controlado é capaz de rejeitar perturbações externas ao sistema. Para executar este teste, mediremos a resposta do sistema enquanto obstruímos a saída de ar através dos buracos inferiores do duto do túnel de vento, efetivamente aumentando o fluxo de ar que eleva a bola.

## 6.2 Resultados Experimentais

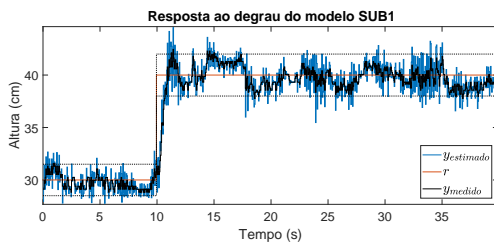
Nesta seção serão apresentados os resultados dos experimentos descritos anteriormente.

### 6.2.1 Resultados da Resposta ao degrau unitário

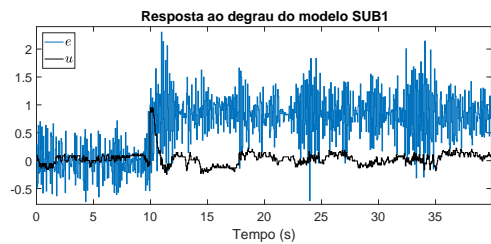
Nas figuras desta seção, as linhas pontilhadas indicam os limites de  $\pm 5\%$  da referência.

#### 6.2.1.1 Modelo *SUB1*

Nos deparamos com uma grande diferença na figura 6.1a, a resposta simulada (figura 5.1), atendeu aos requisitos definidos na seção 5.1.1, mas o sistema real não atendeu, vemos a comparação na tabela 6.1. Com um tempo de assentamento de 8 segundos, muito maior que o tempo projetado, o sistema real também não atende ao requisito de máximo sobrevalor. Era esperado que o sistema real tivesse problemas de atender os requisitos de tempo de assentamento e máximo sobrevalor porque utilizamos um modelo idealizado para projetar o controlador.



(a)  $y_{estimado}$  e  $y_{medido}$  do modelo *SUB1*



(b) erro  $e$  e sinal de controle  $u$  do controlador *SUB1*

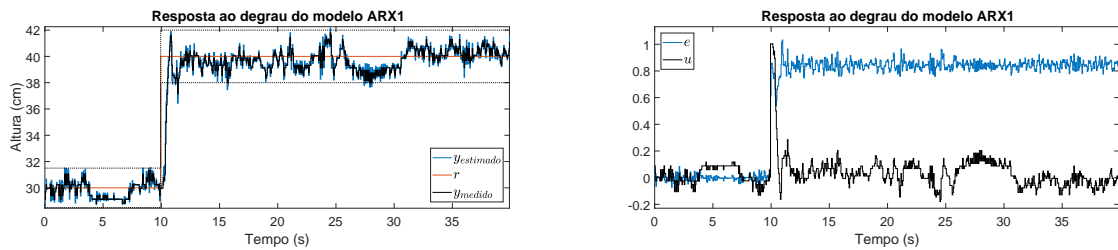
Figura 6.1: Resposta ao degrau do sistema usando o controlador *SUB1*

Requisitos	Projetado	Real
$T_s$ (s)	0.8	8
$ovs\%$	0.125%	6.1164%

Tabela 6.1: Tabela dos requisitos do sistema *SUB1*

### 6.2.1.2 Modelo *ARX1*

A resposta ao degrau com o controlador do modelo *ARX1* apresentou os mesmos problemas do modelo *SUB1* quando aplicado no sistema real: não atendeu aos requisitos de tempo de assentamento e máximo sobrevalor, como vemos na figura 6.2a e na tabela 6.2.

(a)  $y_{estimado}$  e  $y_{medido}$  do modelo *ARX1*(b) erro  $e$  e sinal de controle  $u$  do controlador *ARX1*Figura 6.2: Resposta ao degrau do sistema com o controlador do modelo *ARX1*

Requisitos	Projetado	Real
$T_s$ (s)	0.9	6
$ovs\%$	0.0326%	4.4785%

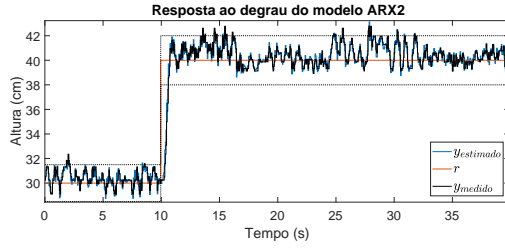
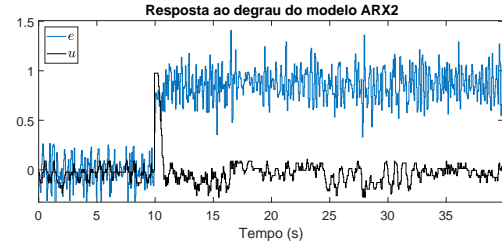
Tabela 6.2: Tabela dos requisitos do sistema *ARX1*

### 6.2.1.3 Modelo *ARX2*

O mesmo problema de não atender aos requisitos apareceu no controlador do modelo *ARX2*, figura 6.3a e na tabela 6.3.

### 6.2.1.4 Modelo *ARXsim*

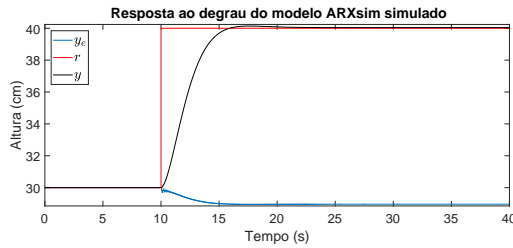
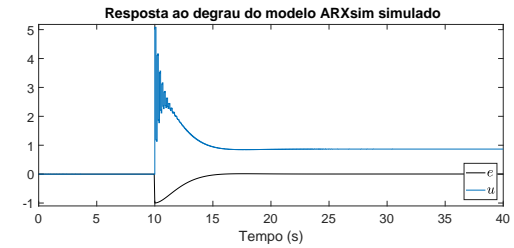
Tentamos aplicar o controlador do modelo *ARXsim* no sistema real, mas o ganho do controlador foi muito alto, o que saturou o sinal de controle do motor. Portanto, simulamos esse modelo. Vemos a sua resposta ao degrau na figura 6.4a e observamos que o sistema é completamente diferente do sistema real, o estimador de estados não

(a)  $y_{estimado}$  e  $y_{medido}$  do modelo  $ARX2$ (b) erro  $e$  e sinal de controle  $u$  do controlador  $ARX2$ Figura 6.3: Resposta ao degrau do sistema com o controlador do modelo  $ARX2$ 

Requisitos	Projetado	Real
$T_s$ (s)	1.3	6
$ovs\%$	0.1487%	6.9156%

Tabela 6.3: Tabela dos requisitos do sistema  $SUB1$ 

está estimando a posição corretamente apesar de ter sido projetado de forma correta. Entretanto, mesmo com o estimador errado, a saída do sistema chega perto do degrau unitário.

(a)  $y_{estimado}$  e  $y_{medido}$  do modelo  $ARXsim$ (b) erro  $e$  e sinal de controle  $u$  do controlador  $ARXsim$ Figura 6.4: Resposta ao degrau do sistema com o controlador do modelo  $ARXsim$ 

### 6.2.1.5 ISE da resposta ao degrau dos modelos simulada

O ISE é um índice de desempenho de sistema que quantifica o desempenho de um sistema para respostas rápidas pela equação (6.1). Vemos na figura 6.5, um gráfico com os 4 modelos identificados,  $SUB1$ ,  $ARX1$ ,  $ARX2$  e  $ARXsim$ , com o ISE calculado para sua resposta ao degrau de -1 à 1. Com esse gráfico, vemos que o controlador que mais minimizou o erro foi o do modelo  $ARX1$  e o que menos minimizou o erro, o do modelo  $ARXsim$ , o que era de se esperar dado que o modelo  $ARXsim$  não é um modelo adequado.

$$ISE = \int e^2 dt \quad (6.1)$$

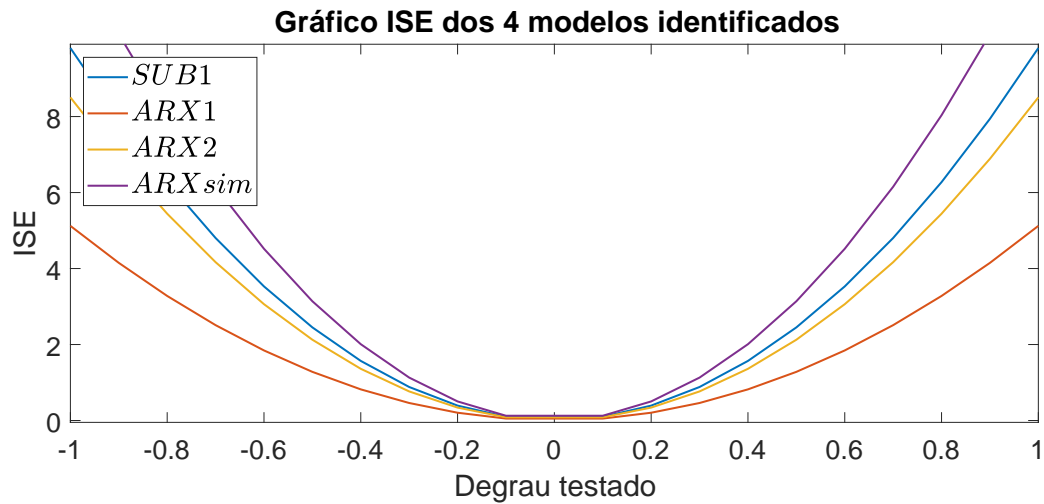


Figura 6.5: Gráfico ISE dos 4 modelos estimados

#### 6.2.1.6 Comparação dos modelos

Na tabela 6.4, mostramos os requisitos de resposta ao degrau definidos. Vemos que no sistema real, nenhum dos modelos foi capaz de atender aos requisitos de sobrevalor e de tempo de assentamento, e o índice ISE foi muito superior ao índice calculado para o modelo simulado.

		Teste			Modelo Simulado	
	$t_s$ (s)	OVS	ISE	$t_s$ (s)	OVS	ISE
<i>SUB1</i>	8	6.125%	14.2801	0.8	0.125%	9.8033
<i>ARX1</i>	6	4.4785%	13.0572	0.9	0.0326%	5.1225
<i>ARX2</i>	6	6.9156%	15.2428	1.3	0.1487%	8.5043
<i>ARXsim</i>	6	0%	8.5043	1.3	0.14%	8.1588

Tabela 6.4: Tabela de comparação dos requisitos de controle dos modelos e índice ISE

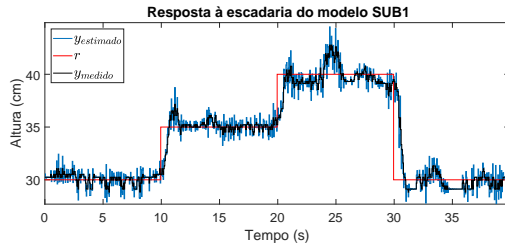
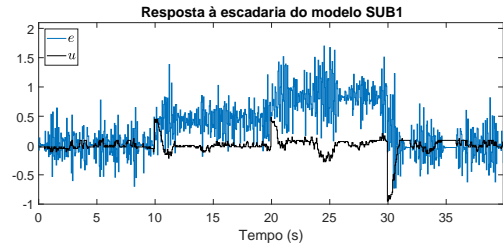
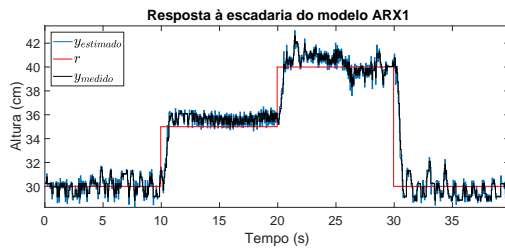
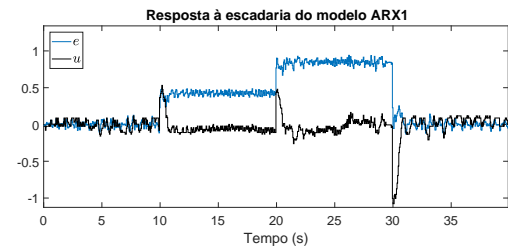
### 6.2.2 Resultados da Resposta à Escadaria

#### 6.2.2.1 Modelo *SUB1*

Testamos a resposta à escadaria do controlador do modelo *SUB1* e constatamos que ele foi capaz de seguir a referência (figura 6.6a).

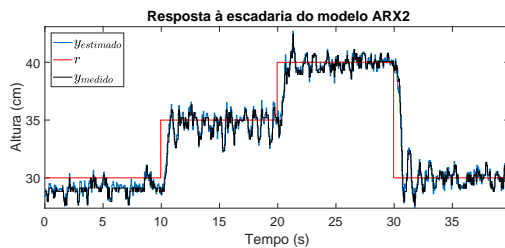
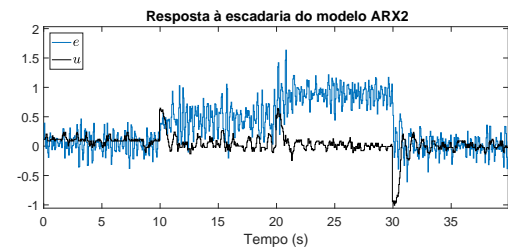
#### 6.2.2.2 Modelo *ARX1*

A resposta à escadaria do sistema real controlado por *ARX1* nos mostra que ele foi capaz de seguir a referência (figura 6.7a).

(a)  $y_{estimado}$  e  $y_{medido}$  do modelo *SUB1*(b) erro  $e$  e sinal de controle  $u$  do controlador *SUB1*Figura 6.6: Resposta à escadaria do sistema com o controlador do modelo *SUB1*(a)  $y_{estimado}$  e  $y_{medido}$  do modelo *ARX1*(b) erro  $e$  e sinal de controle  $u$  do controlador *ARX1*Figura 6.7: Resposta à escadaria do sistema com o controlador do modelo *ARX1*

### 6.2.2.3 Modelo *ARX2*

A resposta à escadaria do modelo *ARX2* também foi capaz de seguir a referência (figura 6.8a).

(a)  $y_{estimado}$  e  $y_{medido}$  do modelo *ARX2*(b) erro  $e$  e sinal de controle  $u$  do controlador *ARX2*Figura 6.8: Resposta à escadaria do sistema com o controlador do modelo *ARX2*

### 6.2.2.4 Modelo *ARXsim*

Testamos o controlador do modelo *ARXsim* para um sinal de referência do tipo escadaria. Vemos na figura 6.9a, que o modelo seguiu a referência.

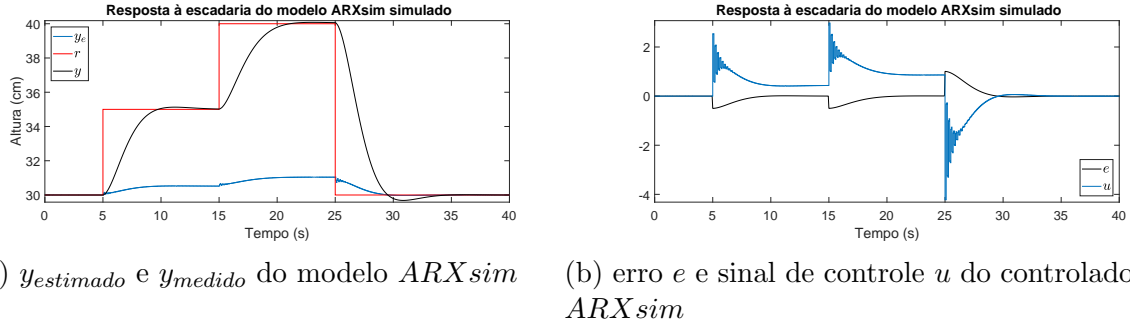


Figura 6.9: Resposta à escadaria do sistema com o controlador do modelo  $ARXsim$

#### 6.2.2.5 Erro do Estimador

É interessante verificar aqui o funcionamento dos estimadores projetados quando aplicados no sistema real. Na figura 6.10, vemos que o estimador do modelo  $SUB1$  teve problemas para estimar a posição da bola, mas os estimadores dos modelos  $ARX1$  e  $ARX2$  (figuras 6.11 e 6.12), tem um erro baixo, o que é de se esperar visto que a identificação por mínimos quadrados é um estimador ótimo, e reduz o resíduo. A tabela 6.5 mostra o índice ISE do erro de estimação e nos confirma que os estimadores dos modelos  $ARX$  foram melhores que o estimador do modelo  $SUB1$ .

	$SUB1$	$ARX1$	$ARX2$
ISE	5.3315	0.5193	0.6134

Tabela 6.5: Tabela com índices ISE do erro de estimação

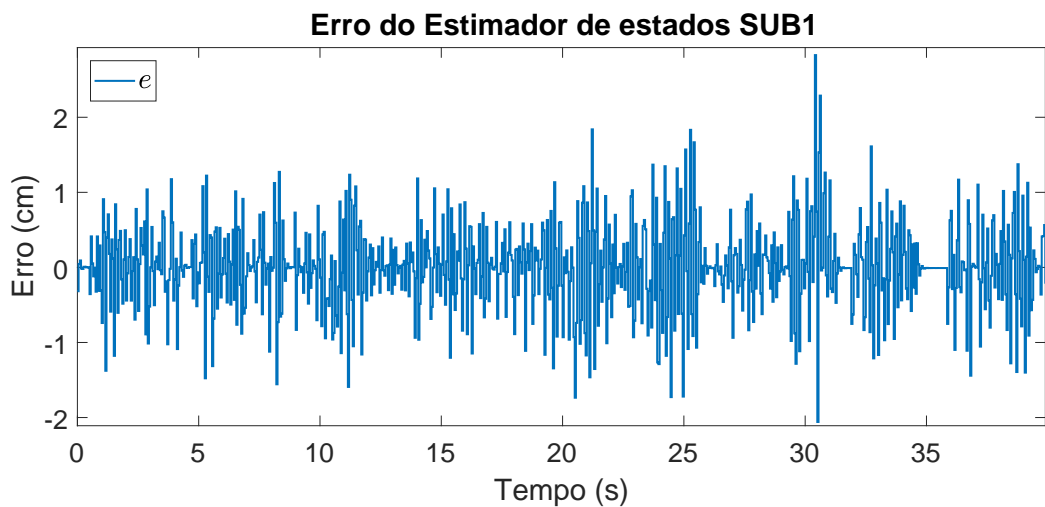


Figura 6.10: Erro do estimador do modelo  $SUB1$  na resposta à escadaria

Vemos que os estimadores dos dois modelos  $ARX$ , ao serem aplicados no sistema, tiveram um bom desempenho (figuras 6.11 e 6.12), permanecendo abaixo de 1 cm de



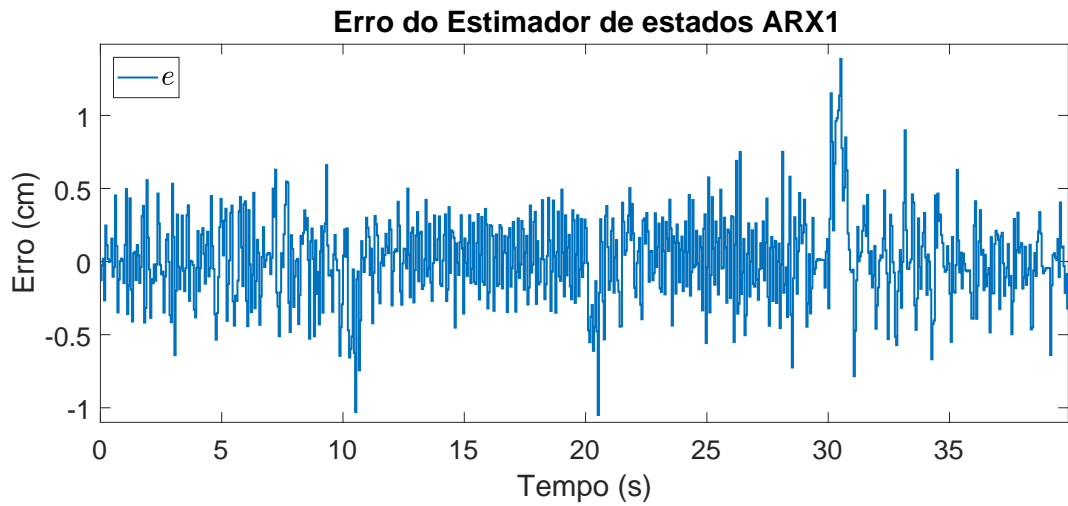


Figura 6.11: Erro do estimador do modelo  $ARX1$  na resposta à escadaria

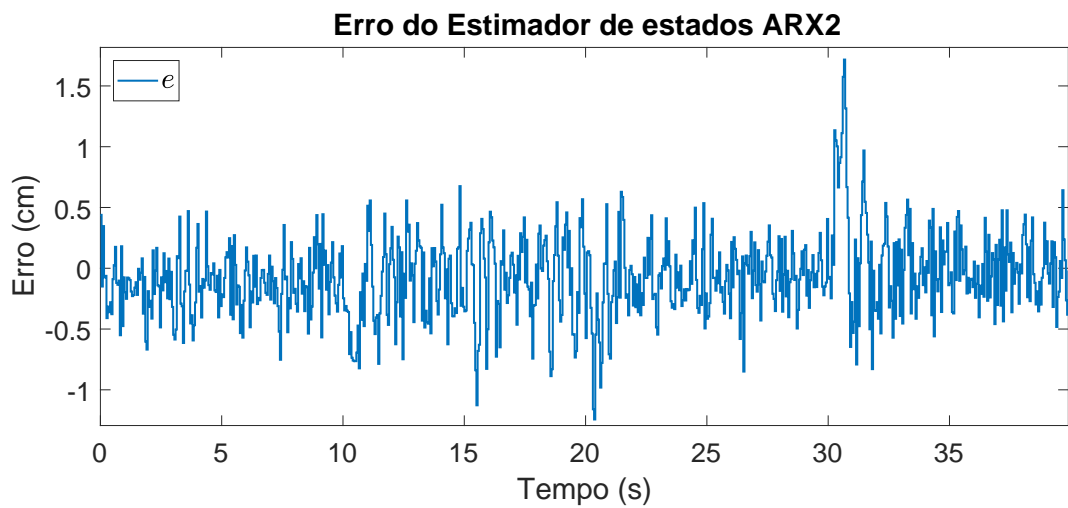


Figura 6.12: Erro do estimador do modelo  $ARX2$  na resposta à escadaria

diferença com algumas exceções, que atribuímos ao ruído do sensor. O estimador do modelo  $SUB1$  não apresentou bom desempenho, mostrando erros superiores a 1 cm com frequência.

## 6.2.3 Resultados do Teste de Robustez à mudança de Parâmetros

### 6.2.3.1 Modelo $SUB1$

Testamos a robustez do modelo  $SUB1$  à mudança de parâmetros com o sistema real e vemos na figura 6.13a que ele não conseguiu seguir a referência quando aumentamos o peso da bola.

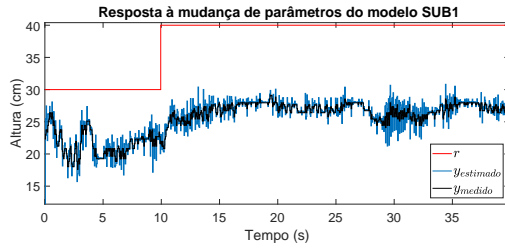
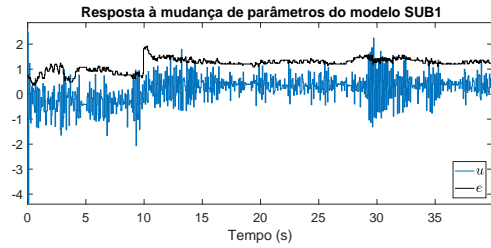
(a)  $y_{estimado}$  e  $y_{medido}$  do modelo  $SUB1$ (b) erro  $e$  e sinal de controle  $u$  do controlador  $SUB1$ 

Figura 6.13: Resposta do sistema com o controlador do modelo  $SUB1$  com mudança de parâmetros

### 6.2.3.2 Modelo $ARX1$

Testamos a robustez do modelo  $ARX1$  à mudança de parâmetros com o sistema real e, como no modelo  $SUB1$ , ele não conseguiu seguir a referência (figura 6.14a).

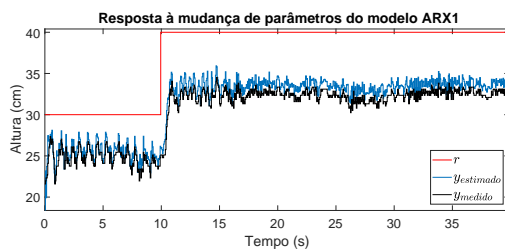
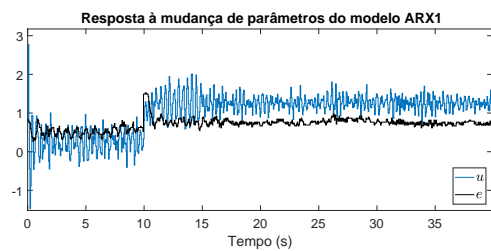
(a)  $y_{estimado}$  e  $y_{medido}$  do modelo  $ARX1$ (b) erro  $e$  e sinal de controle  $u$  do controlador  $ARX1$ 

Figura 6.14: Resposta do sistema com o controlador do modelo  $ARX1$  com mudança de parâmetros

### 6.2.3.3 Modelo $ARX2$

O modelo  $ARX2$  também não conseguiu seguir a referência (figura 6.15a).

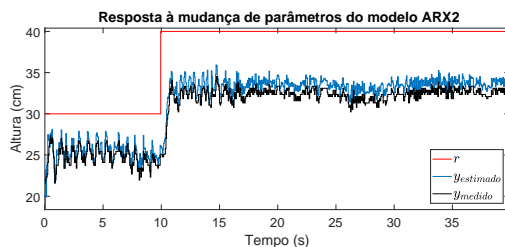
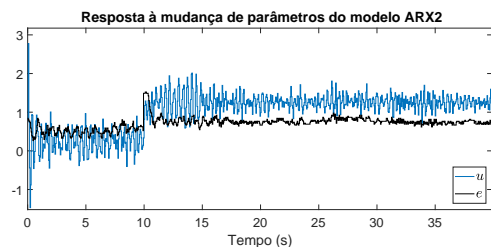
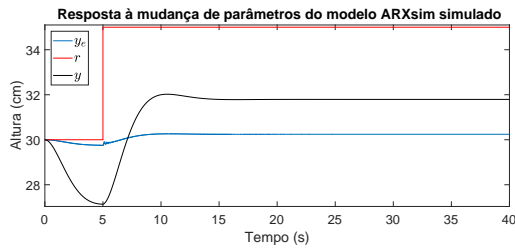
(a)  $y_{estimado}$  e  $y_{medido}$  do modelo  $ARX2$ (b) erro  $e$  e sinal de controle  $u$  do controlador  $ARX2$ 

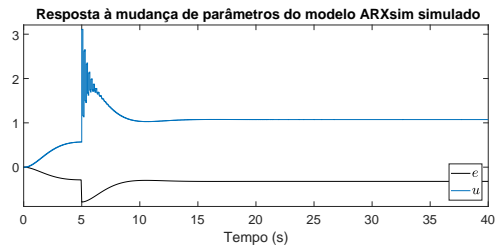
Figura 6.15: Resposta do sistema com o controlador do modelo  $ARX2$  com mudança de parâmetros

### 6.2.3.4 Modelo $ARX_{sim}$

Mudamos o peso da bola para o modelo  $ARX_{sim}$  e testamos a sua resposta ao degrau. Vemos na figura 6.16a, que o sistema simulado não foi capaz de responder à mudança de parâmetros.



(a)  $y_{estimado}$  e  $y_{medido}$  do modelo  $ARX_{sim}$



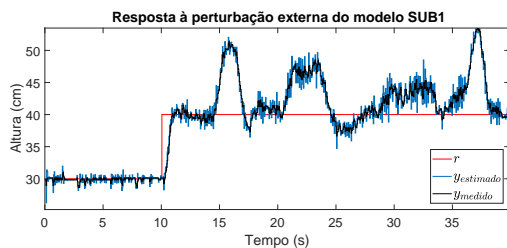
(b) erro  $e$  e sinal de controle  $u$  do controlador  $ARX_{sim}$

Figura 6.16: Resposta do sistema com o controlador do modelo  $ARX_{sim}$  com mudança de parâmetros

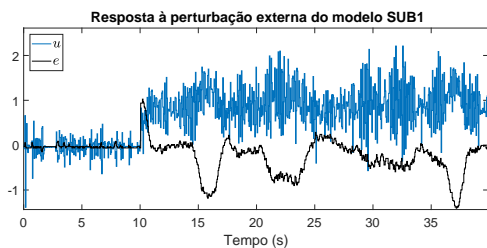
## 6.2.4 Resultados do Teste de Robustez à perturbação externa

### 6.2.4.1 Modelo $SUB1$

Testamos a robustez do modelo  $SUB1$  à perturbação externa com o sistema real e vemos que o sistema também não conseguiu rejeitar perturbação externa (figura 6.17a).



(a)  $y_{estimado}$  e  $y_{medido}$  do modelo  $SUB1$

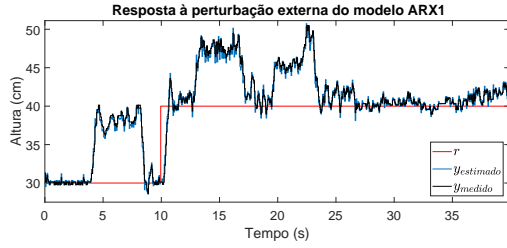
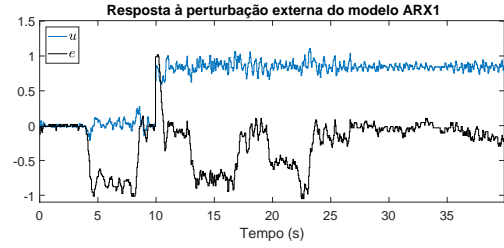


(b) erro  $e$  e sinal de controle  $u$  do controlador  $SUB1$

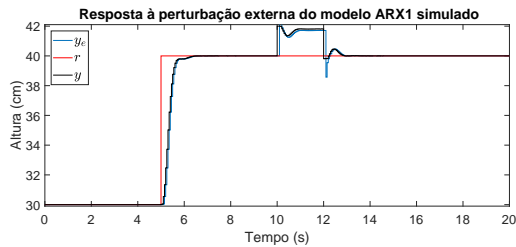
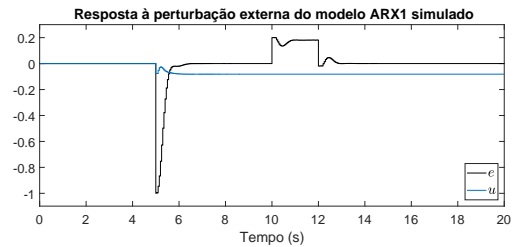
Figura 6.17: Resposta à perturbação externa do sistema com o controlador do modelo  $SUB1$

### 6.2.4.2 Modelo $ARX1$

O modelo  $ARX1$  também não conseguiu seguir a referência com o aumento do peso da bola (figura 6.18a).

(a)  $y_{estimado}$  e  $y_{medido}$  do modelo  $ARX1$ (b) erro  $e$  e sinal de controle  $u$  do controlador  $SUB1$ Figura 6.18: Resposta à perturbação externa do sistema com o controlador do modelo  $ARX1$ 

A resposta à perturbação externa apresentou um resultado curioso. É do nosso entendimento que um controlador seja capaz de corrigir para medidas de erro e reduzir o erro a zero. Entretanto, isso não aconteceu quando o controlador foi aplicado no sistema. Investigamos isso fazendo um teste simulado com o modelo  $ARX1$  e obtivemos a figura 6.19. Com essa simulação, confirmamos que a implementação de nosso controlador não foi defeituosa, apesar de o controlador, de fato, não ter sido capaz de rejeitar perturbação externa.

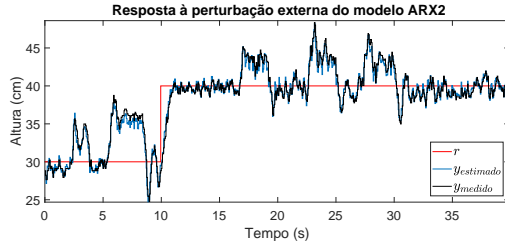
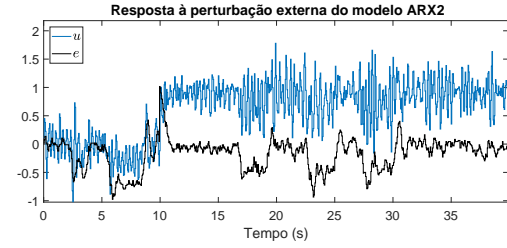
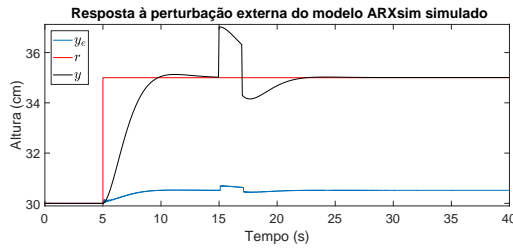
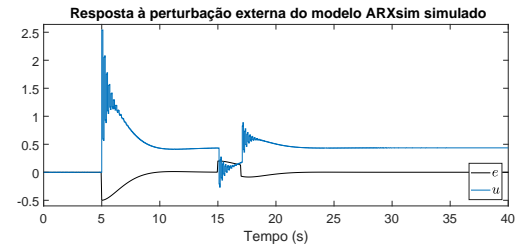
(a)  $y_{estimado}$  e  $y_{medido}$  do modelo  $ARX1$ (b) erro  $e$  e sinal de controle  $u$  do controlador  $SUB1$ Figura 6.19: Resposta à perturbação externa do sistema com o controlador do modelo  $ARX1$  simulado

### 6.2.4.3 Modelo $ARX2$

O modelo  $ARX2$  também não conseguiu seguir a referência com o aumento da bola (figura 6.20a).

### 6.2.4.4 Modelo $ARXsim$

Testamos a resposta do modelo  $ARXsim$  para perturbações externas. Vemos na figura 6.21a, que o sistema não compensou para perturbações externas.

(a)  $y_{estimado}$  e  $y_{medido}$  do modelo  $ARX2$ (b) erro  $e$  e sinal de controle  $u$  do controlador  $SUB1$ Figura 6.20: Resposta à perturbação externa do sistema com o controlador do modelo  $ARX2$ (a)  $y_{estimado}$  e  $y_{medido}$  do modelo  $ARXsim$ (b) erro  $e$  e sinal de controle  $u$  do controlador  $ARXsim$ Figura 6.21: Resposta à perturbação externa do sistema com o controlador do modelo  $ARXsim$ 

### 6.3 Discussão

Ao analisar as respostas obtidas dos experimentos, devemos levar em conta o quão ruidoso é o sensor. Apesar de aplicar um filtro na saída, ele ainda produziu valores que apresentam um alto erro de medida que, em conjunto com o giro da bola devido a força do vento, fizeram com que a leitura do sensor tivesse erros que influenciaram nos resultados.

Na seção 4.4, mostramos que os 3 modelos,  $SUB1$ ,  $ARX1$  e  $ARX2$ , identificados a partir do sistema real, eram adequados para representar o sistema. O modelo  $ARXsim$  identificado a partir do simulador, que continha dados experimentais do sistema sobre a velocidade do vento para diferentes PWM aplicados no motor, não apresentava as características do sistema, o que ficou evidente comparando as figuras 4.1 e 4.2 que mostram as diferenças entre as respostas ao sinal PRBS. A resposta do  $ARXsim$  (figura 4.9), foi discrepante da observada no sistema real e da observada nos outros modelos obtidos.

Essa diferença nos mostra a necessidade da identificação de um sistema usando métodos caixa-preta. Apesar de termos conseguido modelar o sistema de forma matematicamente sólida, o desempenho do modelo obtido foi divergente do sistema real. Enquanto que

os modelos obtidos utilizando métodos de identificação caixa-preta conseguiram gerar modelos satisfatórios do sistema.

Na seção 5.1, projetamos um controlador por realimentação de estados, para cada um dos modelos obtidos, capaz de atender as especificações de tempo de assentamento e máximo sobrevalor definidas. Mas, devido à necessidade de criar um observador de estados na seção 5.2, vimos que a forma como o sistema modelado atende aos requisitos mudou, mesmo se mantendo dentro dos requisitos definidos. No entanto, ao aplicar esse controlador no sistema, vimos que a resposta não estava atendendo aos requisitos estabelecidos. Projetamos ainda um controlador para o modelo obtido do simulador, identificado por caixa-cinza e, ao aplicarmos no sistema real, a resposta foi instável, reforçando a necessidade de obter modelos por identificação caixa-preta.

É interessante notar como a análise de resíduos foi uma medida capaz de prever o quão adequado um sistema é, de acordo com os dados utilizados para gerá-lo. A autocorrelação vista nas figuras 4.3 e 4.6, mostra uma grande diferença entre os sistemas, apesar de a resposta ao degrau dos modelos ser similar. O controle projetado para os modelos referidos, *ARX1* e *SUB1*, mostra que, para que o controle seja bem sucedido, o modelo extraído tem que ser o mais fiel possível ao sistema.

# Capítulo 7

## Conclusão

### 7.1 Considerações Finais

Com este projeto conseguimos evidenciar a importância da identificação de sistemas por métodos caixa-preta para o controle da altura de uma bola dentro de um túnel de vento. Exploramos a possibilidade de usar um modelo caixa branca para modelar o sistema mas percebemos que o seu desempenho estava muito aquém do sistema real. Somente a aplicação de identificação caixa-preta, no entanto, não é suficiente para resolver o problema da modelagem, é preciso validar o modelo escolhido e determinar se o método de identificação é adequado para o tipo de sistema estudado.

O desenvolvimento deste projeto apresentou diversos desafios na construção adequada do sistema estudado. O diâmetro da bola de tênis de mesa foi um fator que ao oscilar dentro do tubo, foi capaz de mudar drasticamente a leitura do sensor. Levando em conta que a oscilação foi da ordem de 2 mm, ainda foi suficiente para interferir no experimento. O motor RC usado, tanto quanto outros motores, não foi capaz de manter uma velocidade constante, o que influenciou nos testes.

A manufatura do tubo do túnel de vento também precisa de certas considerações a serem tomadas como o escape de ar necessário para que a bola possa ser controlada. Foi utilizada uma furadeira de bancada para alinhar os furos verticais e tentar garantir que a distância entre os mesmos fosse igual.

Constatamos que o método de identificação e controle para este tipo de sistema não é adequado para garantir os requisitos de tempo de assentamento e máximo sobrevalor. O sistema estudado é não linear e apresenta uma dinâmica difícil de identificar.

## 7.2 Trabalhos Futuros

Levando em conta os conhecimentos adquiridos durante este trabalho percebemos ser necessário o projeto de um novo túnel de vento, com um duto maior que o atual, com uma bola maior com a mesma relação de diâmetro entre a bola atual e o tubo, para diminuir o erro de medição do sensor. O tubo maior também influencia na medida do sensor diminuindo a chance de haver reflexão nas paredes e nos buracos de escape de ar.



# Referências Bibliográficas

- [1] Aguirre, L. (2015). Introdução à Identificação de Sistemas – Técnicas Lineares e Não-Lineares Aplicadas a Sistemas Reais. Editora UFMG.
- [2] Akaike, H. (1974). A new look at the statistical model identification. IEEE Transactions on Automatic Control, 19(6):716–723.
- [3] bin Kwon, H., Park, Y.-W., ho Lee, D., and Kim, M.-S. (2001). Wind tunnel experiments on korean high-speed trains using various ground simulation techniques. Journal of Wind Engineering and Industrial Aerodynamics, 89(13):1179 – 1195.
- [4] Briggs, L. J. (1959). Effect of spin and speed on the lateral deflection (curve) of a baseball; and the magnus effect for smooth spheres. American Journal of Physics, 27:589–596.
- [5] Chacon, J., Saenz, J., de la Torre, L., Diaz, J., and Esquembre, F. (2017). Design of a low-cost air levitation system for teaching control engineering. Sensors (Basel).
- [6] Chen, C.-T. (1998). Linear System Theory and Design. Oxford University Press, Inc., New York, NY, USA, 3rd edition.
- [7] Jernigan, S. R., Fahmy, Y., and Buckner, G. D. (2009). Implementing a remote laboratory experience into a joint engineering degree program: Aerodynamic levitation of a beach ball. IEEE Transactions on Education, 52(2):205–213.
- [8] journal, E. E. (1946). Vertical wind tunnel. Electrical Engineering, 65(6):265–266.
- [9] Katayama, T. (2005). Subspace Methods for System Identification. Communications and Control Engineering. Springer London.
- [10] McDade, J., Phillips, G., Sivinski, H., and Whitfield, W. (1969). Chapter v principles and applications of laminar-flow devices. In Norris, J. and Ribbons, D., editors,

---

Methods in Microbiology, volume 1 of Methods in Microbiology, pages 137 – 168. Academic Press.

- [11] Ricco, R. (2012). Identificação de sistemas utilizando métodos de subespaços.