



**Universidade de Brasília  
Faculdade de Tecnologia**

**Validação de observador de distúrbio com  
filtro adaptativo em sistema de tanques**

Felipe Calassa Albuquerque  
Gabriel Vieira de Arimatéa

TRABALHO DE GRADUAÇÃO 2  
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Brasília  
2022

**Universidade de Brasília**  
**Faculdade de Tecnologia**

**Validação de observador de distúrbio com  
filtro adaptativo em sistema de tanques**

Felipe Calassa Albuquerque

Gabriel Vieira de Arimatéa

Trabalho de Graduação submetido como re-  
quisito parcial para obtenção do grau de Enge-  
nheiro de Controle e Automação

Orientador: Prof. Dr. Eduardo Stockler Tognetti

Coorientador: Prof. Dr. Eugênio Libório Feitosa Fortaleza

Brasília

2022

C143v Calassa Albuquerque, Felipe.  
Validação de observador de distúrbio com filtro adaptativo em sistema de tanques / Felipe Calassa Albuquerque; Gabriel Vieira de Arimatéia; orientador Eduardo Stockler Tognetti; coorientador Eugênio Libório Feitosa Fortaleza. -- Brasília, 2022.  
94 p.

Trabalho de Graduação 2 em Engenharia de Controle e Automação -- Universidade de Brasília, 2022.

1. Observador de Distúrbio. 2. Tanques em Cascata. 3. Controle de Nível. 4. Controle de Processos. I. Vieira de Arimatéia, Gabriel. II. Stockler Tognetti, Eduardo, orient. III. Libório Feitosa Fortaleza, Eugênio, coorient. IV. Título

**Universidade de Brasília  
Faculdade de Tecnologia**

**Validação de observador de distúrbio com filtro  
adaptativo em sistema de tanques**

Felipe Calassa Albuquerque  
Gabriel Vieira de Arimatéa

Trabalho de Graduação submetido como re-  
quisito parcial para obtenção do grau de Enge-  
nheiro de Controle e Automação

Trabalho aprovado. Brasília, 23 de setembro de 2022:

---

**Prof. Dr. Eduardo Stockler Tognetti,**  
UnB/FT/ENE  
Orientador

---

**Prof. Dr. Guilherme Caribe de Carvalho,**  
UnB/FT/ENM  
Examinador interno

---

**Prof. Dr. Lélío Ribeiro Soares Júnior,**  
UnB/FT/ENE  
Examinador interno

---

**Prof. Dr. Eugênio Libório Feitosa**  
Fortaleza, UnB/FT/ENM  
Coorientador

Brasília  
2022

*Dedico esta produção acadêmica ao desenvolvimento  
da ciência e à educação como ferramenta social inclusiva*

Felipe Calassa Albuquerque

*Dedico este trabalho à educação e conhecimento científico,  
áreas que ameaçam quem deseja dominar através da desinformação.*

Gabriel Vieira de Arimatéa

# Agradecimentos

Agradeço principalmente à educação e criação que recebi dos meus pais, Maria da Conceição e Cristóvão Madeira, que fizeram de mim a pessoa que sou hoje. Toda e qualquer conquista sempre serão para vocês.

Agradeço também à minha melhor amiga, porto seguro e irmã Samara Calassa, pelo amor, cuidado e companheirismo em todos esses anos.

Agradeço à minha amiga Cintia Isla, que além de ser uma irmã para mim, foi a maior responsável pelo meu ingresso na UNB.

Agradecimento imenso a todas amigadas que são, para mim, família e rede de apoio. Obrigado a todos vocês pelo amor, carinho, paciência, sorrisos, aventuras, conversas e cuidado. Vocês são o motivo que me fazem continuar.

Por fim, um agradecimento especial ao Gabriel Vieira, amigo e companheiro de curso. Obrigado pela cumplicidade e pela ajuda nos momentos mais difíceis. Você foi a pessoa que mais me motivou para a conclusão desse trabalho.

Felipe Calassa Albuquerque

Agradeço primeiramente à minha mãe, Silmara Vieira, por sempre batalhar para que não faltasse nada para mim e para minha irmã, e por sempre correr atrás para que tivéssemos o melhor acesso à educação. Sem seus esforços e dedicação, eu não seria o homem que sou hoje. Agradeço à minha irmã, Dandara, por ser minha parceira e uma pessoa que posso confiar. Agradeço também ao meu pai, Marcos Arimatéa, pelo cuidado e atenção, pelos finais de semana que passamos juntos e pelos ensinamentos de como levar a vida com mais alegria e leveza. Agradeço ao meu namorado Raphael pelo carinho, pelo cuidado e pelo amor nos momentos que eu mais precisava, você tornou esses dias de estresse mais leves, te amo muito. Agradeço também à Regina Lúcia, minha boadrasta, pela atenção, carinho e acolhimento que você tem comigo, com minha irmã e com minha mãe.

Agradeço aos meus colegas e amigos de faculdade, sem vocês essa jornada seria muito menos divertida e prazerosa. Agradeço especialmente ao Abdullah, Ian e Kaynã por estarem ao meu lado desde os primeiros momentos na universidade. Agradeço imensamente à DROID, equipe na qual formei uma outra família e onde aprendi e cresci muito, tanto na área técnica, quanto como pessoa, sou eternamente grato pelas madrugadas viradas remontando os robôs no momento das competições, as memórias das pessoas, das situações, das alegrias e das tensões estarão sempre guardadas com um carinho imenso. Por fim, quero agradecer ao meu colega e também parceiro neste trabalho, Felipe Calassa, pela confiança

em dividir esta responsabilidade e por todas as conversas e conselhos que você me deu nesses anos de amizade.

Gabriel Vieira de Arimatéa

*“Acima de tudo, nunca pense que você não é bom o suficiente. Um homem nunca deveria pensar isso”*  
*(Isaac Asimov)*

# Resumo

Este trabalho apresenta os estudos para a integração de um observador de distúrbio para um sistema de tanques em cascata, que mantém um fluxo constante de líquido. Foram estudadas novas propostas de arquitetura em malha fechada com a inserção de observador de distúrbio com o objetivo de verificar se existe uma arquitetura que melhore o controle do nível do líquido no tanque. Para lidar com a amplificação de ruídos pelo observador de distúrbio, é proposta a utilização de um filtro adaptativo. Além dos testes de diversas arquiteturas, foram analisados ajustes no filtro presente dentro do observador de distúrbio para verificar sua melhor performance. Os resultados foram validados e comparados através de indicadores-chave de performance da saída e da entrada.

**Palavras-chave:** Observador de Distúrbio. Tanques em Cascata. Controle de Nível. Controle de Processos.

# Abstract

This work presents the studies for the integration of a disturbance observer for a cascade tank system, which maintains a constant flow of liquid. New closed loop architecture proposals were studied with the insertion of a disturbance observer in order to verify if there is an architecture that improves the control of the liquid level in the tank. To deal with noise amplification by the disturbance observer, an adaptive filter is proposed. Besides the tests of several architectures, adjustments in the filter present inside the disturbance observer were analyzed to verify its best performance. The results were validated and compared through output and input key performance indicators. This is the english abstract.

**Keywords:** Disturbance Observator. Cascade Tanks. Level Controle. Process Control.

# Lista de ilustrações

Figura 1 – Sistema de controle baseado em observador de distúrbio . . . . .	21
Figura 2 – Representação de um filtro estatístico . . . . .	22
Figura 3 – Arquitetura 1 referência da malha somente com controlador . . . . .	24
Figura 4 – Arquitetura 2: Sistema com observador de distúrbio simples e realimentação unitária . . . . .	24
Figura 5 – Arquitetura 3: Sistema com filtro adaptativo na entrada do observador de distúrbio e realimentação unitária . . . . .	25
Figura 6 – Arquitetura 4: Sistema com observador de distúrbio e filtro adaptativo na realimentação . . . . .	25
Figura 7 – Arquitetura 5: Sistema com filtro adaptativo na entrada do observador de distúrbio e na realimentação . . . . .	25
Figura 8 – Arquitetura 6: Sistema com filtro adaptativo como entrada do observador de distúrbio pela saída do controlador e realimentação unitária . . . . .	26
Figura 9 – Arquitetura 7: Sistema com filtro adaptativo como entrada do observador de distúrbio pela realimentação e realimentação unitária . . . . .	26
Figura 10 – Arquitetura 8: Sistema com filtro adaptativo no observador de distúrbio (sem a função $Q(s)$ ) e realimentação unitária . . . . .	27
Figura 11 – Arquitetura 9: Sistema com filtro adaptativo na entrada do observador de distúrbio, na realimentação e como entrada do observador de distúrbio pela saída do controlador. . . . .	27
Figura 12 – Arquitetura 10: Sistema com observador de distúrbio modificado pelo filtro adaptativo e realimentação pelo mesmo. . . . .	28
Figura 13 – Arquitetura com Observador de Distúrbio para fase não-mínima . . . . .	28
Figura 14 – Fonte de ruído . . . . .	29
Figura 15 – Ensaio I: Saídas $y$ do sistema sem ruído, para todas as arquiteturas . . . . .	30
Figura 16 – Ensaio I: Saídas $y_n$ do sistema com ruído, para todas as arquiteturas . . . . .	31
Figura 17 – Ensaio I: Saídas $y_n$ do sistema com ruído, ampliada no momento do distúrbio . . . . .	31
Figura 18 – Ensaio I: Sinal $u$ do controlador, para todas as arquiteturas . . . . .	32
Figura 19 – Ensaio II: Saídas $y$ do sistema sem ruído, para as arquiteturas 1, 2, 5 e FE . . . . .	34
Figura 20 – Ensaio II: Sinal $u$ do controlador, para as arquiteturas 1, 2, 5 e FE . . . . .	35
Figura 21 – Esquemático do sistema com dois tanques . . . . .	36
Figura 22 – Ensaio III: Saídas $y$ do sistema sem ruído, para as arquiteturas 1, 2, 5 e FE . . . . .	37
Figura 23 – Ensaio III: Saídas $y_n$ do sistema com ruído para as arquiteturas 1, 2, 5 e FE . . . . .	37
Figura 24 – Ensaio III: Sinais $u$ do controlador para as arquiteturas 1, 2, 5 e FE . . . . .	38
Figura 25 – Foto do sistema real em laboratório . . . . .	40
Figura 26 – Esquemático da planta real . . . . .	41

Figura 27 – Foto do painel com CLP (Controlador Lógico Programável) e suas conexões elétricas . . . . .	41
Figura 28 – Malha aberta conectada ao sistema real . . . . .	42
Figura 29 – Tela do programa <i>Automation Studio</i> . . . . .	43
Figura 30 – Resposta malha aberta . . . . .	44
Figura 31 – Arquitetura Malha Fechada Base (MFB) . . . . .	45
Figura 32 – Arquitetura Malha Fechada somente com Filtro Estatístico (MFE) . . . . .	45
Figura 33 – Resposta arquitetura MFB . . . . .	46
Figura 34 – Sinal de controle da arquitetura MFB . . . . .	46
Figura 35 – Resposta arquitetura MFE . . . . .	47
Figura 36 – Sinal de controle da arquitetura MFE . . . . .	47
Figura 37 – Resposta arquitetura MFB no ponto de operação . . . . .	48
Figura 38 – Sinal de controle da arquitetura MFB no ponto de operação . . . . .	49
Figura 39 – Resposta arquitetura MFE no ponto de operação . . . . .	49
Figura 40 – Sinal de controle da arquitetura MFE no ponto de operação . . . . .	50
Figura 41 – Resposta do sistema com $T_d = 2\% \cdot T_b$ . . . . .	51
Figura 42 – Estrutura interna do bloco Observador de Distúrbio . . . . .	52
Figura 43 – Arquitetura base com Observador de Distúrbio (MOD) . . . . .	52
Figura 44 – Arquitetura base com Observador de Distúrbio e Filtro Estatístico (MOF) . . . . .	53
Figura 45 – Parâmetro IAE para cada teste na arquitetura MOD. . . . .	54
Figura 46 – Parâmetro ISE para cada teste na arquitetura MOD. . . . .	55
Figura 47 – Parâmetro TV para cada teste na arquitetura MOD. . . . .	55
Figura 48 – Resposta do sistema para $T_d = 55\% \cdot T_b$ . . . . .	56
Figura 49 – Sinal do controlador para $T_d = 55\% \cdot T_b$ . . . . .	56
Figura 50 – Resposta do sistema para $T_d = 70\% \cdot T_b$ . . . . .	57
Figura 51 – Sinal do controlador para $T_d = 70\% \cdot T_b$ . . . . .	57
Figura 52 – Sistema MOD. Sinais do nível da água em porcentagem. . . . .	58
Figura 53 – Sistema MOD: Sinais da saída do controlador. . . . .	58
Figura 54 – Sistema MOD: Sinais da saída do Observador. . . . .	59
Figura 55 – Parâmetro IAE para cada teste na arquitetura MOF. . . . .	60
Figura 56 – Parâmetro ISE para cada teste na arquitetura MOF. . . . .	61
Figura 57 – Parâmetro TV para cada teste na arquitetura MOF. . . . .	61
Figura 58 – Resposta do sistema para $T_d = 45\% \cdot T_b$ . . . . .	62
Figura 59 – Sinal de controle para $T_d = 45\% \cdot T_b$ . . . . .	62
Figura 60 – Resposta do sistema para $T_d = 55\% \cdot T_b$ . . . . .	63
Figura 61 – Sinal de controle para $T_d = 55\% \cdot T_b$ . . . . .	63
Figura 62 – Sistema MOF: Sinais do nível da água em porcentagem e saída do observador de distúrbio. . . . .	64
Figura 63 – Sistema MOF: Sinais do sistema. . . . .	64

Figura 64 – Sistema MOF: Sinais da saída do Observador. . . . .	65
Figura 65 – Comparação do parâmetro IAE dos testes em MOD e MOF. . . . .	66
Figura 66 – Comparação do parâmetro ISE dos testes em MOD e MOF. . . . .	66
Figura 67 – Comparação do parâmetro TV dos testes em MOD e MOF. . . . .	67
Figura 68 – Melhores sinais de saída encontrados . . . . .	67
Figura A.1 – Saída $y$ das arquiteturas 8 e 10 com $\sigma_n = 0.5$ . . . . .	74
Figura A.2 – Controlador das arquiteturas 8 e 10 com $\sigma_n = 0.5$ . . . . .	74
Figura A.3 – Saída $y$ da arquitetura 9 com $\sigma_n = 0.5$ . . . . .	75
Figura A.4 – Controlador da arquitetura 9 com $\sigma_n = 0.5$ . . . . .	75
Figura A.5 – Saída $y$ das arquiteturas 8 e 10 com $\sigma_n = 1$ . . . . .	76
Figura A.6 – Sinal de controle $u$ das arquiteturas 1, 2, 3, 4 e 5 . . . . .	76
Figura A.7 – Sinal de controle $u$ das arquiteturas 1, 3 e 5 . . . . .	77
Figura A.8 – Sinal de controle das arquiteturas 8 e 10 com $\sigma_n = 1$ . . . . .	77
Figura A.9 – Saída $y$ da arquitetura 9 com $\sigma_n = 1$ . . . . .	78
Figura A.10 – Controlador da arquitetura 9 com $\sigma_n = 0.5$ . . . . .	78
Figura A.11 – Ensaio I: Saída $y$ da arquitetura 2 . . . . .	79
Figura A.12 – Ensaio I: Saída $y$ da arquitetura 3 . . . . .	79
Figura A.13 – Ensaio I: Saída $y$ da arquitetura 4 . . . . .	80
Figura A.14 – Ensaio I: Saída $y$ da arquitetura 5 . . . . .	80
Figura A.15 – Ensaio I: Saída $yn$ das arquiteturas 4, 5 e 10 . . . . .	81
Figura A.16 – Ensaio I: Sinal $u$ do controlador 2 . . . . .	81
Figura A.17 – Ensaio I: Sinal $u$ do controlador 3 . . . . .	82
Figura A.18 – Ensaio I: Sinal $u$ do controlador 4 . . . . .	82
Figura A.19 – Ensaio I: Sinal $u$ do controlador 5 . . . . .	83
Figura A.20 – Ensaio II: Saídas $y$ do sistema sem ruído, com $\sigma_n = 3.765$ para arquitetura 5	83
Figura A.21 – Ensaio II: Saídas $yn$ do sistema com ruído, com $\sigma_n = 7.53$ para arquitetura 5	84
Figura A.22 – Ensaio II: Saídas $yn$ do sistema com ruído, com $\sigma_n = 3.765$ para arquitetura 5 . . . . .	84
Figura A.23 – Ensaio II: Sinal $u$ da arquitetura 1 . . . . .	85
Figura A.24 – Ensaio II: Sinal $u$ para as arquiteturas 1 e 2 . . . . .	85
Figura A.25 – Ensaio II: Sinal $u$ para as arquiteturas 1 e FE . . . . .	86
Figura A.26 – Ensaio II: Sinal $u$ entre as arquiteturas 1 e 5 com $\sigma_n = 7.53$ . . . . .	86
Figura A.27 – Ensaio II: Sinal $u$ entre as arquiteturas 1 e 5 com $\sigma_n = 3.765$ . . . . .	87
Figura A.28 – Ensaio III: Aproximação dos sinais $u$ do sistema sem a arquitetura 2 . . . . .	87
Figura A.29 – Ensaio III: Aproximação dos sinais $u$ do sistema sem as arquiteturas 1 e 2	88

# Lista de tabelas

Tabela 1 – Indicadores de desempenho Ensaio I . . . . .	33
Tabela 2 – Indicadores de desempenho Ensaio II . . . . .	35
Tabela 3 – Indicadores de desempenho Ensaio III . . . . .	38
Tabela 4 – Indicadores de desempenho das arquiteturas MFB e MFE no ponto de operação . . . . .	50
Tabela 5 – Indicadores de performance para arquitetura base com observador de distúrbio (MOD) . . . . .	54
Tabela 6 – Indicadores de performance para arquitetura com filtro estatístico e ob- servador de distúrbio . . . . .	60
Tabela 7 – Comparação final dos indicadores . . . . .	68

# Lista de abreviaturas e siglas

$\alpha$	Frequência de corte do filtro estatístico . . . . .	23
$\eta$	Ruído . . . . .	21
$\sigma_n$	Derivação do padrão de ruído . . . . .	22
$C(s)$	Controlador da planta . . . . .	21
$e$	Erro absoluto ( $y - r$ ) . . . . .	23
$G_d(s)$	Função de transferência que representa a dinâmica do distúrbio . . . . .	21
$G_n(s)$	Modelo nominal do processo $G_p(s)$ . . . . .	21
$G_p(s)$	Função de transferência que representa a planta estudada . . . . .	21
$Q(s)$	Filtro pertencente ao Observador de Distúrbio . . . . .	21
$Q_E(s)$	Filtro estatístico . . . . .	22
$r$	Sinal de entrada/referência . . . . .	21
$T_d$	Constante de tempo do filtro $Q(s)$ . . . . .	21
$u$	Sinal de controle . . . . .	21
$y$	Sinal de saída . . . . .	21
$y_n$	Sinal de saída com ruído . . . . .	21
CLP	Controlador Lógico Programável . . . . .	41
IAE	Integral of Absolute Error . . . . .	23
ISE	Integral of Squared Error . . . . .	23
MFB	Malha Fechada Base . . . . .	45
MFE	Malha Fechada somente com Filtro Estatístico . . . . .	45
MOD	Malha Fechada somente com Observador de Distúrbio . . . . .	52
MOF	Malha Fechada com Observador de Distúrbio e Filtro Estatístico . . . . .	53
$T_a$	Constante de tempo do polo mais rápido de $G(s)$ . . . . .	50
$T_b$	Constante de tempo do polo mais lento de $G(s)$ . . . . .	50
TV	Total Variation of input . . . . .	23

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>17</b>
1.1	Contextualização	17
1.2	Revisão bibliográfica	18
1.3	Definição do problema	18
1.4	Objetivo	19
1.4.1	Objetivos específicos	19
1.5	Metodologia	19
1.6	Organização do trabalho	20
1.7	Organização do repositório	20
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>21</b>
2.1	Observador de distúrbio	21
2.2	Filtro estatístico	22
2.3	Indicadores-chave de desempenho	23
2.4	Arquiteturas de controle	23
2.4.1	Arquiteturas testadas	24
2.4.2	Sistemas de fase não mínima	28
<b>3</b>	<b>ENSAIOS COMPUTACIONAIS</b>	<b>29</b>
3.1	Sistema de primeira ordem	29
3.2	Sistemas com atraso	33
3.3	Sistema tanques em cascata	36
3.4	Conclusão das simulações	39
<b>4</b>	<b>ENSAIOS EXPERIMENTAIS</b>	<b>40</b>
4.1	Sistema de tanques em cascata	40
4.2	Sistemas de automação e comunicação	41
4.3	Comparação entre o sistema simulado e o real	43
4.4	Análise das arquiteturas sem Observador de Distúrbio	44
4.5	Inclusão do Observador de Distúrbio	50
4.6	Análise dos dados - Arquitetura sem filtro estatístico (MOD)	53
4.7	Análise dos dados - Arquitetura com filtro estatístico (MOF)	59
4.8	Comparação entre sistemas	65
<b>5</b>	<b>CONCLUSÕES</b>	<b>69</b>
5.1	Perspectivas futuras	70

	<b>REFERÊNCIAS . . . . .</b>	<b>71</b>
	<b>APÊNDICES</b>	<b>73</b>
	<b>APÊNDICE A – ENSAIOS . . . . .</b>	<b>74</b>
<b>A.1</b>	<b>Sistema de primeira ordem . . . . .</b>	<b>74</b>
A.1.1	Calibração do padrão de ruído . . . . .	74
A.1.2	Respostas do sistema . . . . .	78
<b>A.2</b>	<b>Função de transferência com atraso . . . . .</b>	<b>83</b>
<b>A.3</b>	<b>Sistema tanques em cascata . . . . .</b>	<b>87</b>
	<b>APÊNDICE B – CÓDIGOS DE PROGRAMAÇÃO . . . . .</b>	<b>89</b>
<b>B.1</b>	<b>Script que carrega as variáveis necessárias . . . . .</b>	<b>89</b>
<b>B.2</b>	<b>Scripts para calculo dos indicadores-chave de performance . . . . .</b>	<b>90</b>
<b>B.3</b>	<b>Scripts para automatizar testes com os diversos valores de <math>T_d</math> . . . . .</b>	<b>91</b>
<b>B.4</b>	<b>Script para selecionar os melhores valores de <math>T_d</math> . . . . .</b>	<b>92</b>

# 1 Introdução

## 1.1 Contextualização

O setor industrial se preocupa cada vez mais com a perda dos seus insumos produzidos, sempre buscando novas formas de otimizar o meio de produção, evitando gastos excessivos, desperdício desnecessário e um maior controle no fluxo de produção. Baseado nessas premissas, tecnologias de controle e automação são implantadas nos chãos de fábrica para adaptar a produção em termos de eficiência, rapidez e segurança, otimizando o tempo de produção.

Um sistema de controle deve garantir que as variáveis de interesse permaneçam nos valores de referências especificados. Em um problema de seguimento de referência, também chamado de rastreamento, a dinâmica controlada deve ser capaz de acompanhar a mudança do ponto de operação desejado da maneira mais precisa possível. Entretanto, na indústria de processos, o problema mais crítico a ser enfrentado é a rejeição de distúrbios que fazem com que as variáveis controladas se desviem do valor desejado.

Em geral, a sintonia de controladores é feita de forma a ter uma resposta satisfatória para o problema de seguimento de referência. Uma vez que o projeto tenha uma boa especificação na velocidade de resposta, espera-se que a rejeição aos distúrbios também seja eficaz. Contudo, nem sempre a rejeição ao distúrbio é satisfatória e estratégias avançadas são necessárias.

Duas estratégias se destacam para a rejeição à distúrbios: o controle em cascata e o de alimentação. O controle em cascata, largamente utilizado industrialmente inclusive para além da rejeição ao distúrbio, utiliza uma malha interna mais rápida para responder à perturbações na entrada antes que ela afete a variável controlada. O controle de alimentação propõe uma compensação do efeito do distúrbio no sinal de controle por meio de um modelo de como a perturbação influencia a variável controlada. Ambas as estratégias demandam a medição de variáveis adicionais, seja uma variável secundária no caso da cascata ou do distúrbio no caso da alimentação, e portanto requerem um investimento na instrumentação. Muitas vezes, o distúrbio não pode ser medido ou o modelo de sua influência é difícil de ser obtido, por isso o controle de alimentação não é muito utilizado na prática.

Uma estratégia ainda pouco adotada mas com grandes vantagens sobre as mencionada acima é o controle por meio de observadores de distúrbio. A ideia é utilizar o modelo do processo, o mesmo que é utilizado para a sintonia do controlador e por isso acessível, para estimar (observar) o distúrbio e fazer uma compensação que anule sua influência.

## 1.2 Revisão bibliográfica

Com uma nova proposta para controle, o observador de distúrbio foi inicialmente apresentado por Kyioshi Ohishi (OHISHI, 1983), baseado na teoria de observadores para reduzir o erro na resposta transitória e estacionária, aplicado em um sistema que controlaria a velocidade de um motor DC. Esta proposta serviu como base de estudos para aplicação em demandas industriais, com o objetivo de estimar os distúrbios e remover sua influência em cima do sistema controlado.

Ao longo de décadas, estudos que focavam o aprimoramento desse modelo foram intensificados. Chen (2004), implementou uma topologia de controle baseado em observador de perturbações não lineares em sistemas SISO (sigla em inglês para *single-input and single-output*). Em 2005 Guo e Chen propuseram novos controles que estimam perturbações de sistemas não-lineares em sistemas MIMO (sigla em inglês para *multiple-input and multiple-output*) (GUO; CHEN, 2005).

Até os dias atuais, com o avanço dessa linha de pesquisas e desenvolvimento de tecnologias de controle, a aplicação do observador de distúrbio se estendeu para diversos campos de atuação. Como exemplo, existem estudos para compensar as vibrações em um controlador de frequência de carga (MI et al., 2016), regular a velocidade de veículos hipersônicos sujeitos à distúrbios na entrada (AN et al., 2016) ou então controlar sistemas estáveis de fase não mínimas e com atraso (WANG; SU, 2015).

## 1.3 Definição do problema

Distúrbios são bem comuns no setor industrial, podendo ter diversas origens, como vibrações das máquinas, diferença de temperatura entre processos e o ambiente que o sistema está inserido, escoamento turbulento de fluidos, entradas não observadas pela malha de controle, dentre outros. Essas interferências causadas pelo distúrbio podem diminuir a qualidade do produto (CHEN et al., 2015).

Mesmo com um controlador com a sintonia bem ajustada, os processos ainda sofrem influência de distúrbios, que acabam reduzindo a eficácia do controle. Esses distúrbios podem acabar se propagando para outras malhas de controle. Uma solução para impedir a influência de distúrbios na saída da malha de controle é a inclusão de um Observador de Distúrbio.

Este trabalho apresenta uma estratégia de implementação do observador em cenário muito ruidoso, como normalmente ocorre nas indústrias. Esse cenário é desafiador, pois uma das desvantagens conhecidas do observador de distúrbio é a amplificação de ruído de alta frequência. A solução comumente adotada nesses casos é o *detuning* do observador, ou seja, fazê-lo menos reativo, mas também menos eficiente em compensar as perturbações.

---

A solução adotada nesse trabalho é o uso de um filtro estatístico que não compromete a eficiência do observador e não requer informações em frequência da perturbação, como é o caso de filtros passa-baixa.

## 1.4 Objetivo

Este trabalho tem por objetivo propor arquiteturas de controle que sejam eficientes em mitigar o efeito de distúrbios nas variáveis controladas em malhas monovariáveis.

### 1.4.1 Objetivos específicos

Por sua vez, os objetivos específicos são:

- Propor arquiteturas utilizando observadores de distúrbio e filtros para reduzir o efeito do distúrbio segundo indicadores de desempenho escolhidos.
- Verificar como o uso de filtros adaptativos, junto com observador de distúrbio, pode ajudar a reduzir uma das principais desvantagens de observadores de distúrbio: amplificação de ruídos de alta frequência.
- Aplicar o filtro estatístico em diversas posições na arquitetura de controle comparando os resultados.
- Validar experimentalmente os benefícios de observadores de distúrbio e a dificuldade em sintonizá-los em um processo real de escala de laboratório.

## 1.5 Metodologia

Primeiro foram modeladas arquiteturas de controle com observador de distúrbio em diversas configurações que, combinado com o filtro estatístico, poderia trazer melhores benefícios na resposta da malha analisada.

Em seguida, ensaios computacionais foram realizados para verificar as arquiteturas viáveis e que apresentam melhores indicadores de desempenho, quando comparados com o controle de realimentação clássico, observando também o impacto do filtro estatístico.

Por fim, foram realizados ensaios experimentais em bancada para avaliar as arquiteturas selecionadas na etapa anterior, com o objetivo de propor quais as configurações minimizam o efeito dos distúrbios.

---

## 1.6 Organização do trabalho

O trabalho está organizado em cinco capítulos e dois apêndices conforme descrito abaixo:

- Capítulo 1: Introdução - Apresenta uma introdução, contextualização e revisão bibliográfica do trabalho.
- Capítulo 2: Fundamentos teóricos - Define o observador de distúrbio, filtro estatístico e o cálculo dos indicadores de desempenho.
- Capítulo 3: Ensaio computacionais - Apresenta os ensaios realizados no *Matlab*, com as melhores arquiteturas e resultados obtidos.
- Capítulo 4: Ensaio experimentais - Traz as análises e resultados dos ensaios realizados em bancada.
- Capítulo 5: Conclusão - Encerra a análise dos dados obtidos e propõe os passos seguintes a serem tomados no projeto.

## 1.7 Organização do repositório

Os dados obtidos nos ensaios experimentais, os códigos e arquivos utilizados no *Matlab* foram armazenados em um repositório no site *Github*, que pode ser acessado através do link <https://github.com/vieira-gabriel/TG2>

## 2 Fundamentação teórica

Neste capítulo serão descritos os fundamentos teóricos necessários ao entendimento do trabalho.

### 2.1 Observador de distúrbio

Para lidar com a compensação de distúrbios no processo, Li et al. (2014) propôs a seguinte arquitetura mostrada na Figura 1.

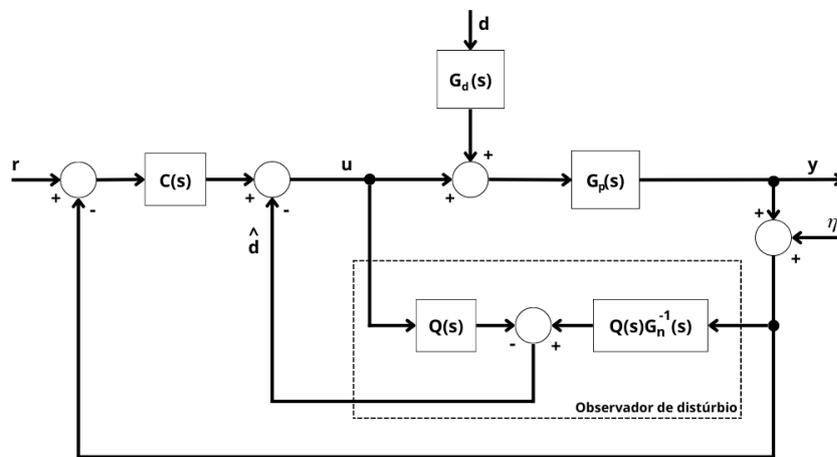


Figura 1 – Sistema de controle baseado em observador de distúrbio

Nesta representação,  $u$  é o sinal de controle,  $y$  é a saída da planta,  $r$  é o sinal de referência,  $d$  é o distúrbio,  $\eta$  o ruído de medição,  $G_p(s)$  o modelo do processo a ser controlado,  $G_n(s)$  o modelo nominal do processo,  $C(s)$  um controlador que atende as especificações de rastreamento e  $Q(s)$  o filtro do observador de distúrbio dado por

$$Q(s) = \frac{1}{(T_d s + 1)^q} \quad (2.1)$$

sendo  $T_d$  a constante de tempo do filtro e  $q$  a ordem relativa do filtro, escolhida de forma que  $Q(s)G_n^{-1}$  seja um produto causal.

Na arquitetura da Figura 1, a relação entrada-saída é dada pela seguinte expressão no domínio da frequência:

$$Y(s) = T_{yr}(s)R(s) + T_{yd}(s)D(s)G_d(s) \quad (2.2)$$

onde

$$T_{yr}(s) = \frac{G_n(s)G_p(s)C(s)}{G_n(s)[1 + C(s)G_p(s)] + Q(s)[G_p(s) - G_n(s)]} \quad (2.3)$$

$$T_{yd}(s) = \frac{G_n(s)G_p(s)[1 - Q(s)]}{G_n(s)[1 + C(s)G_p(s)] + Q(s)[G_p(s) - G_n(s)]} \quad (2.4)$$

Dado que  $Q(s)$  é um filtro passa-baixa com ganho igual a um em regime permanente ( $\lim_{\omega \rightarrow 0} Q(j\omega) = 1$ ) e que as Equações (2.3) e (2.4) possuem grande dependência dele, tem-se que

$$\lim_{\omega \rightarrow 0} T_{yr}(j\omega) = \lim_{\omega \rightarrow 0} \frac{G_n(j\omega)C(j\omega)}{1 + G_n(j\omega)C(j\omega)} \quad (2.5)$$

e

$$\lim_{\omega \rightarrow 0} T_{yd}(j\omega) = 0 \quad (2.6)$$

É importante comentar que para sistemas em que  $G_n = G_p$  o comportamento da resposta ao degrau não é alterada. E para altas frequências, quando o modelo do sistema é menos preciso,  $Q(s)$  atua como um filtro na diferença entre  $G_n$  e  $G_p$ .

Portanto, para distúrbios de baixa-frequência, sua influência na malha de controle pode ser atenuada assintoticamente pelo observador de distúrbio.

## 2.2 Filtro estatístico

Outro parâmetro para prosseguir este estudo, é quanto à utilização de filtros estatísticos em sistemas de controle. Os filtros estatísticos tem como objetivo, através de comparações estatísticas, amenizar os efeitos de ruídos ou distúrbios por ter uma certa predicabilidade. Na figura abaixo é apresentada a forma que o filtro estatístico é introduzido na malha fechada (Figura 2).

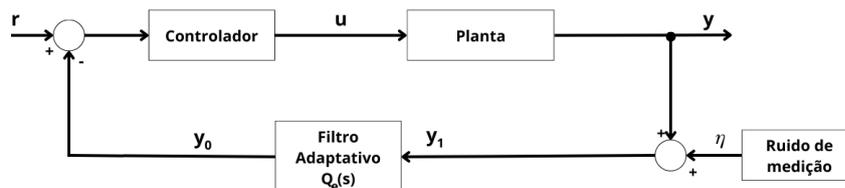


Figura 2 – Representação de um filtro estatístico

Para os ensaios deste trabalho será usado um filtro estatístico  $Q_E(s)$ , o qual será dimensionado para cada sistema. Esse filtro receberá como parâmetro um valor  $\sigma_n$ , apresentado também na Equação (2.8), com o desvio padrão do ruído.

Neste trabalho, foram utilizados o filtro e os cálculos já obtidos no artigo [Fortaleza et al. \(2022\)](#), sendo o filtro definido pelas Equações (2.7) e (2.8).

$$\begin{cases} y_0[0] = y_1[0] \\ y_0[k] = y_0[k-1] + \alpha[k](y_1[k] - y_0[k-1]), \text{ para } k > 0 \end{cases} \quad (2.7)$$

em que  $0 \leq \alpha \leq 1$ , o  $\alpha$  define a frequência de corte (ou constante de tempo) do filtro e afeta a suavidade do sinal  $y_1$ , o qual é calculado por:

$$\alpha[k] = \operatorname{erf} \left( \frac{|y_0[k-1] - y_1[k]|}{2\sigma_n\sqrt{2}} \right) \quad (2.8)$$

É observado que  $\alpha[k]$  é atualizado em função da medição e o único parâmetro que deve ser informado é o desvio padrão  $\sigma_n$  do ruído de medição, que é obtido através dos dados ou das características fornecidas pelo fabricante do instrumento de medição.

## 2.3 Indicadores-chave de desempenho

Ao fim das simulações, são extraídos os indicadores de desempenho IAE (*integral of absolute error*), ISE (*integral of squared error*) e TV (*total variation of input*), para diferentes saídas da malha. Eles são usados como uma forma mais prática de comparar alguns parâmetros entre os diferentes tipos de ensaios (SHAMSUZZOHA; LEE, 2007).

- Desempenho da saída de controle: para analisar a saída do controlador é computado o erro absoluto da referência  $e = r - y$  (deve tender a zero). O indicador ISE penaliza erros maiores, já o indicador IAE é analisado para obter uma boa configuração que penalize erros maiores e erros a longo prazo.

$$IAE = \int_0^{\infty} |e(t)| dt \quad (2.9)$$

$$ISE = \int e(t)^2 dt$$

- Desempenho da entrada de controle: avaliar a entrada da planta, computando a variação total da entrada  $u(t)$ . O indicador TV sinaliza a suavidade do sinal, quanto menor, mais suave.

$$TV = \sum_{i=1}^{\infty} |u_{i+1} - u_i| \quad (2.10)$$

em que  $u_{i+1}$  é o valor do sinal de controle no instante seguinte a  $u_i$ .

## 2.4 Arquiteturas de controle

Com intuito de explorar e comparar as diferentes possibilidades de controle da malha, foram selecionadas algumas opções de arquiteturas, cuja diferença entre elas está na posição do observador de distúrbio e do filtro adaptativo  $Q_E$ .

### 2.4.1 Arquiteturas testadas

Com base na malha simples de realimentação (Figura 3), foram modeladas e analisadas outras nove arquiteturas. Os principais sinais analisados foram: o sinal de saída da malha sem ruído ( $y$ ), o sinal de saída da malha com ruído ( $y_n$ ) e o sinal de controle ( $u$ ).

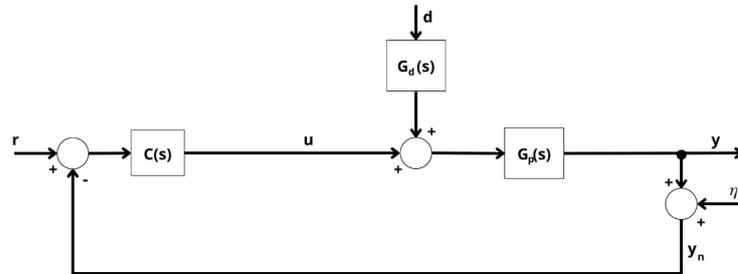


Figura 3 – Arquitetura 1 referência da malha somente com controlador

A arquitetura 2 (Figura 4) tem o observador de distúrbio inserido na malha, que recebe como entrada tanto o sinal de controle  $u$  como a saída da malha com ruído  $y_n$ .

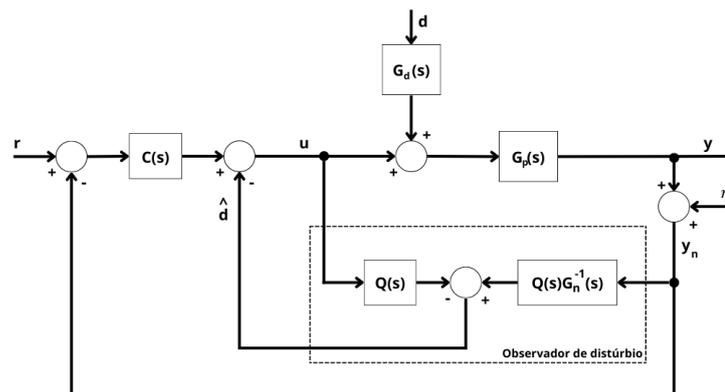


Figura 4 – Arquitetura 2: Sistema com observador de distúrbio simples e realimentação unitária

A arquitetura 3 (Figura 5) tem o observador de distúrbio e o filtro estatístico  $Q_E(s)$  inseridos na malha. Nesse caso, o filtro estatístico recebe o sinal de saída com ruído  $y_n$  e o observador de distúrbio recebe tanto o sinal de controle  $u$  como a saída  $y_n$  filtrada por  $Q_E(s)$ . A realimentação principal dessa malha não é alterada.

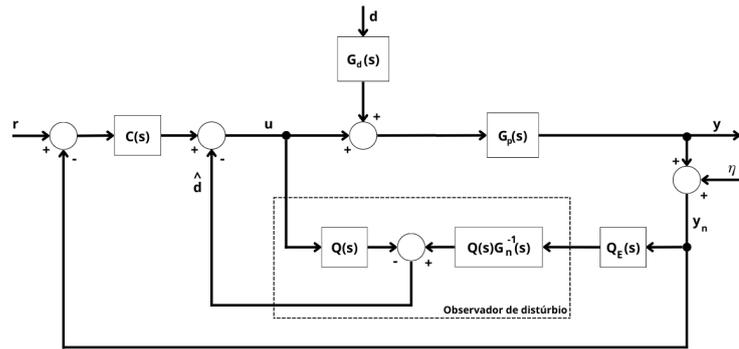


Figura 5 – Arquitetura 3: Sistema com filtro adaptativo na entrada do observador de distúrbio e realimentação unitária

A arquitetura 4 (Figura 6) tem o observador de distúrbio e o filtro estatístico  $Q_E(s)$  inseridos na malha. Nesse caso, o filtro estatístico  $Q_E(s)$  é usado apenas na realimentação principal da malha.

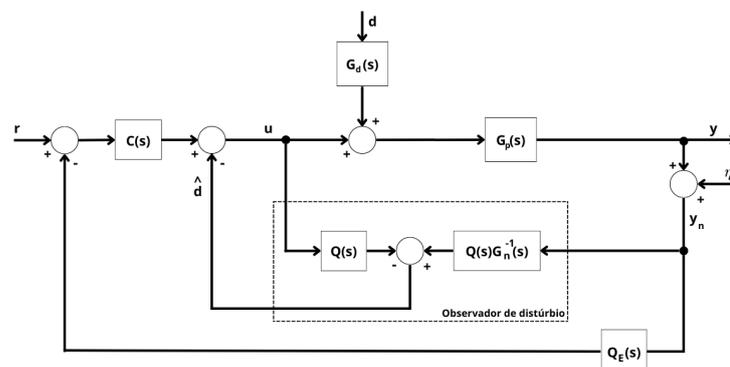


Figura 6 – Arquitetura 4: Sistema com observador de distúrbio e filtro adaptativo na realimentação

A arquitetura 5 (Figura 7) tem o observador de distúrbio e o filtro estatístico  $Q_E(s)$  inseridos na malha. Nesse caso, o sinal da saída com ruído,  $y_n$ , passa por  $Q_E(s)$  e segue tanto para a entrada do observador quanto para a realimentação da malha.

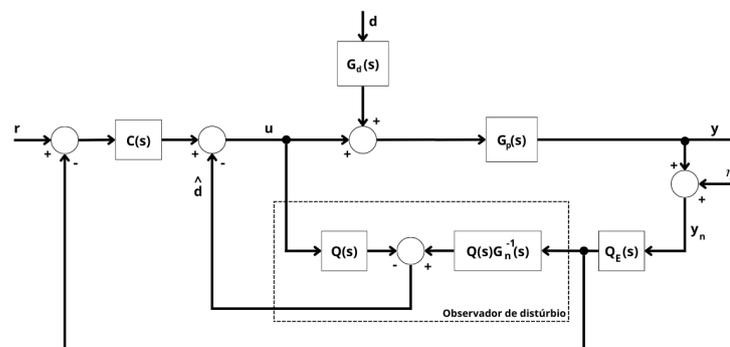


Figura 7 – Arquitetura 5: Sistema com filtro adaptativo na entrada do observador de distúrbio e na realimentação

Na arquitetura 6 (Figura 8), o observador de distúrbio foi alterado de forma que o filtro

$Q(s)$  que recebe o sinal de controle  $u$  passa a ser o filtro estatístico  $Q_E(s)$ . A realimentação principal desse sistema não é filtrada.

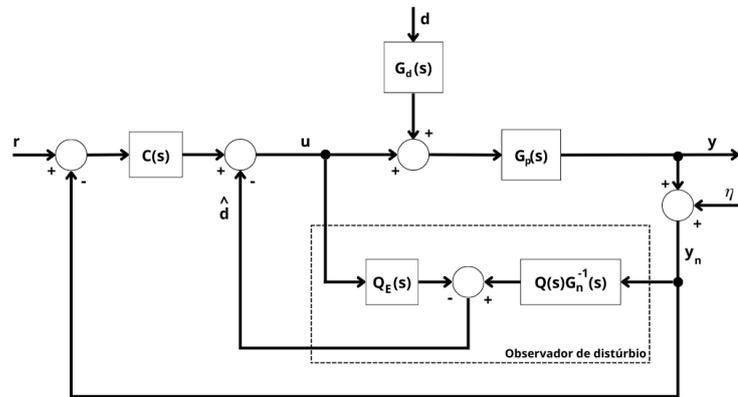


Figura 8 – Arquitetura 6: Sistema com filtro adaptativo como entrada do observador de distúrbio pela saída do controlador e realimentação unitária

Assim como na anterior, a arquitetura 7 (Figura 9) é uma modificação da malha tradicional com observador de distúrbio (Figura 4). Nesse caso, o observador de distúrbio foi modificado de forma que a função de transferência que recebe a saída  $y_n$  (antes  $Q(s)G_n^{-1}(s)$ ) passou a ter o filtro estatístico multiplicado pela função de transferência do modelo nominal ( $Q_E(s)G_n^{-1}(s)$ ).

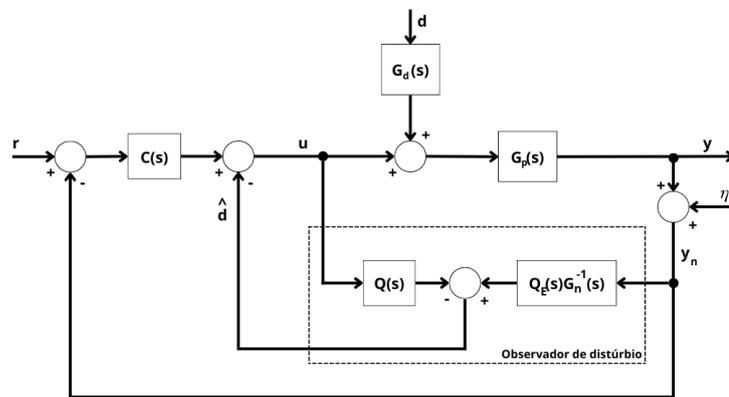


Figura 9 – Arquitetura 7: Sistema com filtro adaptativo como entrada do observador de distúrbio pela realimentação e realimentação unitária

A arquitetura 8 (Figura 10) é uma modificação da arquitetura 4 (Figura 4), na qual os filtros do observador  $Q(s)$  são substituídos pelo filtro estatístico  $Q_E(s)$ . Nesse caso a realimentação principal desse sistema não é filtrada.

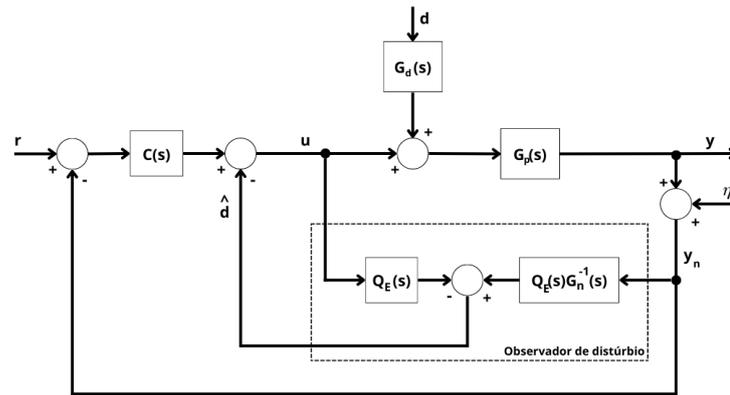


Figura 10 – Arquitetura 8: Sistema com filtro adaptativo no observador de distúrbio (sem a função  $Q(s)$ ) e realimentação unitária

A arquitetura 9 (Figura 11) tem o filtro estatístico  $Q_E(s)$  inseridos na malha e tem o mesmo observador de distúrbio modificado presente na arquitetura 6 (filtro  $Q(s)$  passou a ser o filtro estatístico  $Q_E(s)$ ). Nela o sinal  $y_n$  passa por  $Q_E(s)$  e segue para entrada do observador de distúrbio e realimentação da malha.

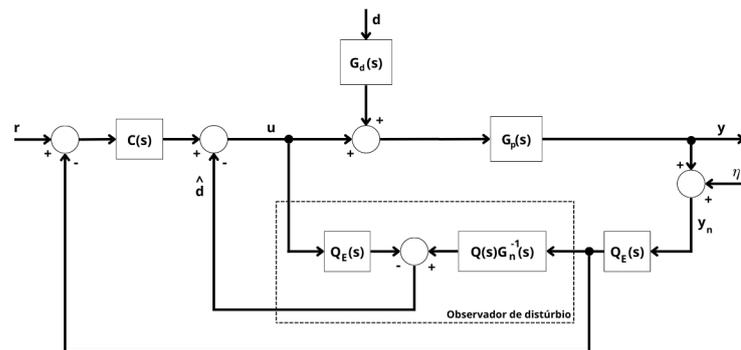


Figura 11 – Arquitetura 9: Sistema com filtro adaptativo na entrada do observador de distúrbio, na realimentação e como entrada do observador de distúrbio pela saída do controlador.

A arquitetura 10 (Figura 12) tem o observador de distúrbio modificado e o filtro estatístico  $Q_E(s)$  inseridos na malha. Nesse caso, o sinal  $y_n$  passa por  $Q_E(s)$  e segue para entrada do observador de distúrbio e para a realimentação da malha. O observador de distúrbio teve filtro  $Q(s)$  substituído pelo filtro estatístico  $Q_E(s)$ , além da função de transferência que recebe a saída  $y_n$  (filtrada por  $Q_E(s)$ ) passar a ser somente  $G_n^{-1}(s)$ .

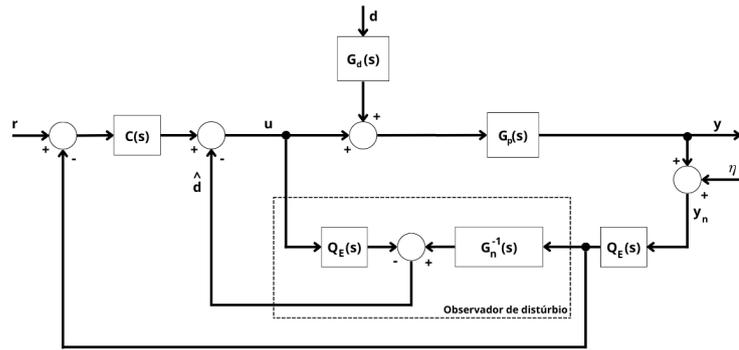


Figura 12 – Arquitetura 10: Sistema com observador de distúrbio modificado pelo filtro adaptativo e realimentação pelo mesmo.

## 2.4.2 Sistemas de fase não mínima

Para o sistema apresentado pela arquitetura 2 (Figura 4) e todas as arquiteturas subsequentes que se baseiam no modelo básico com observador de distúrbio, é necessário que  $G_n$  seja de fase mínima, já que é utilizada sua inversa. Caso a fase seja não mínima, é necessário realizar uma adaptação da estrutura do observador.

A arquitetura para fase não mínima é utilizada quando  $Q(s)G_n^{-1}(s)$  não é realizável (possuir mais zeros que polos) ou é instável. Para isso, é necessário fatorar a função de transferência  $G_n(s)$ , de tal forma que  $G_n(s) = G_{n-}(s) \cdot G_{n+}(s)$ . Deste modo,  $G_{n-}(s)$  deve representar o sistema com aproximação dos polos no semiplano à esquerda e  $G_{n+}(s)$  à direita.

Desta forma,  $G_{n-}(s)$  deve ser tal que  $Q(s)G_{n-}^{-1}(s)$  seja de fase mínima.

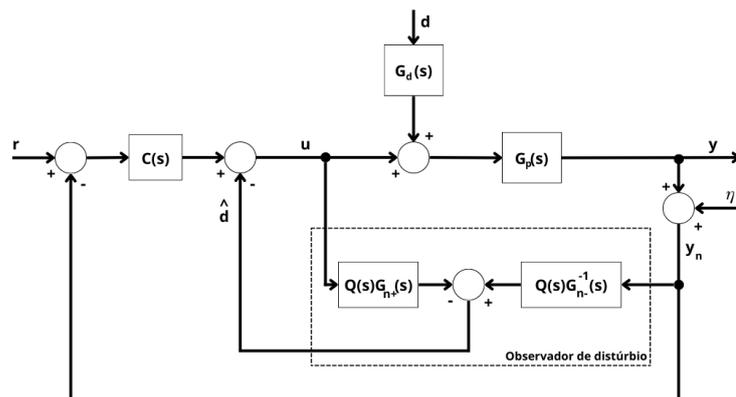


Figura 13 – Arquitetura com Observador de Distúrbio para fase não-mínima

## 3 Ensaaios computacionais

### 3.1 Sistema de primeira ordem

Neste primeiro ensaio, foram utilizados os seguintes valores do sistema:

$$G_p(s) = \frac{2.6}{62s + 1} \quad (3.1)$$

$$C(s) = 3 \cdot \frac{30s + 1}{30s} \quad (3.2)$$

$$T_d = 1.24s \quad (3.3)$$

$$Q(s) = \frac{1}{T_d s + 1} = \frac{1}{1.24s + 1} \quad (3.4)$$

$$\sigma_n = 0.5 \quad (3.5)$$

$$G_d(s) = \frac{1}{23s + 1} \quad (3.6)$$

Para o sistema de primeira ordem, o valor de  $T_d$  foi determinado como 2% da constante de tempo de  $G_p(s)$ . O valor de  $\sigma_n$  para o filtro adaptativo foi um valor empírico obtido em ensaios anteriores. Para a planta do sistema (Equação (3.1)) e o controlador em questão (Equação (3.2)), foi utilizado como referência o modelo de estudo apresentado por [Johansson \(2000\)](#).

A função de transferência  $G_d(s)$  é utilizada para simular a dinâmica de um distúrbio.

#### RUÍDO

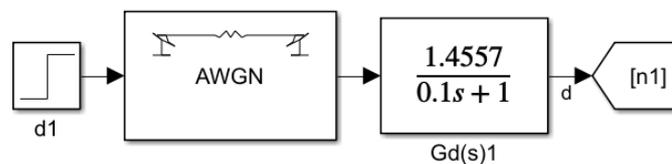


Figura 14 – Fonte de ruído

A fonte de ruído acima (Figura 14) foi utilizada para simular o ruído causado pela planta. Ele possui como entrada um sinal do tipo degrau de amplitude 0, e tempo de amostragem de  $T_s = 0.1$  segundos, somente para introduzir o ruído no sistema. O bloco AWGN adiciona um ruído gaussiano branco ao sinal que passa por ele, no qual os parâmetros são "Initial seed", com o valor de 67, e a variância do ruído, estipulado como  $8 \cdot 10^{-4}$ .

Usando os indicadores como parâmetro para o *tuning* das constantes de tempo do filtro e para avaliação das melhores arquiteturas, tem-se os seguintes preceitos:

- Se o objetivo for priorizar desempenho na resposta ao distúrbio, escolher a arquitetura com melhor IAE ou ISE;
- Se o objetivo for priorizar filtragem, escolher a arquitetura com melhor  $TV$ ;
- Se o objetivo for priorizar o compromisso do filtro, escolher a arquitetura com indicadores intermediários.

Na simulação, os sinais das arquiteturas 8, 9 e 10 (Figuras 10, 11 e 12, respectivamente) foram insatisfatórios, ainda contendo ruídos muito altos na resposta (Figuras A.1 e A.3 no Apêndice A) e sinais controlados também ruidosos (Figuras A.2, e A.4). Foi alterado o valor do desvio padrão do ruído para  $\sigma_n = 1$  nas arquiteturas citadas. Mesmo com essa modificação, os sinais da arquitetura 9 foram insatisfatórios (Figuras A.9 e A.10), portanto ela foi descartada para as análises seguintes.

Para efeitos de comparação, são apresentados os sinais  $y$ ,  $y_n$  e  $u$  (Figuras 15, 16, 18) para todas as arquiteturas.

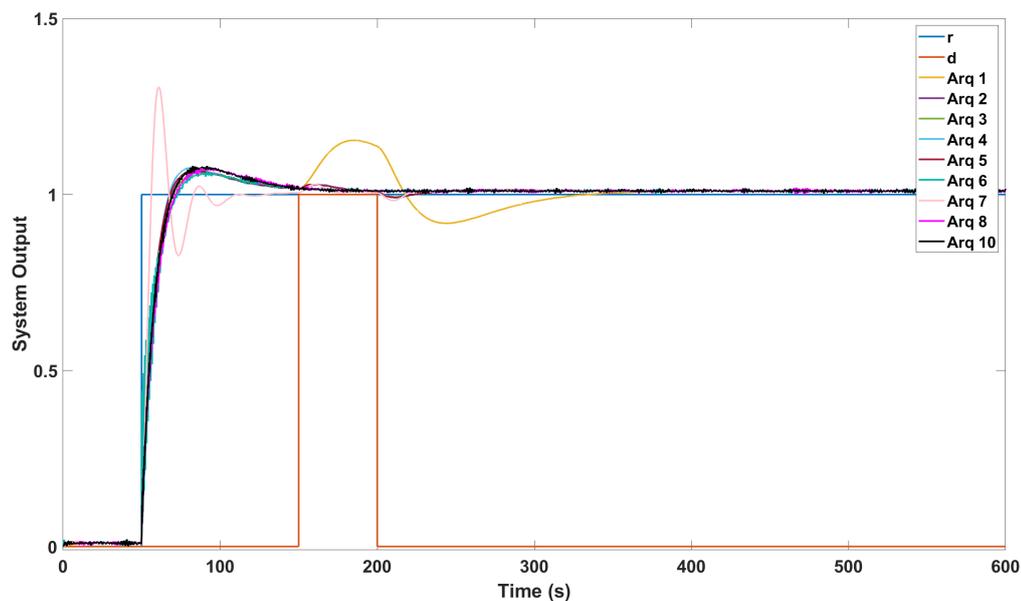


Figura 15 – Ensaio I: Saídas  $y$  do sistema sem ruído, para todas as arquiteturas

Observam-se na Figura 15 as respostas do sistema tanto a um degrau unitário na referência quanto a um distúrbio aplicado. Nota-se a diferença da resposta ao distúrbio (no instante  $T = 150s$ ) entre a arquitetura 1 e as demais arquiteturas. Na arquitetura 7 houve uma degradação do sinal na resposta ao degrau de referência

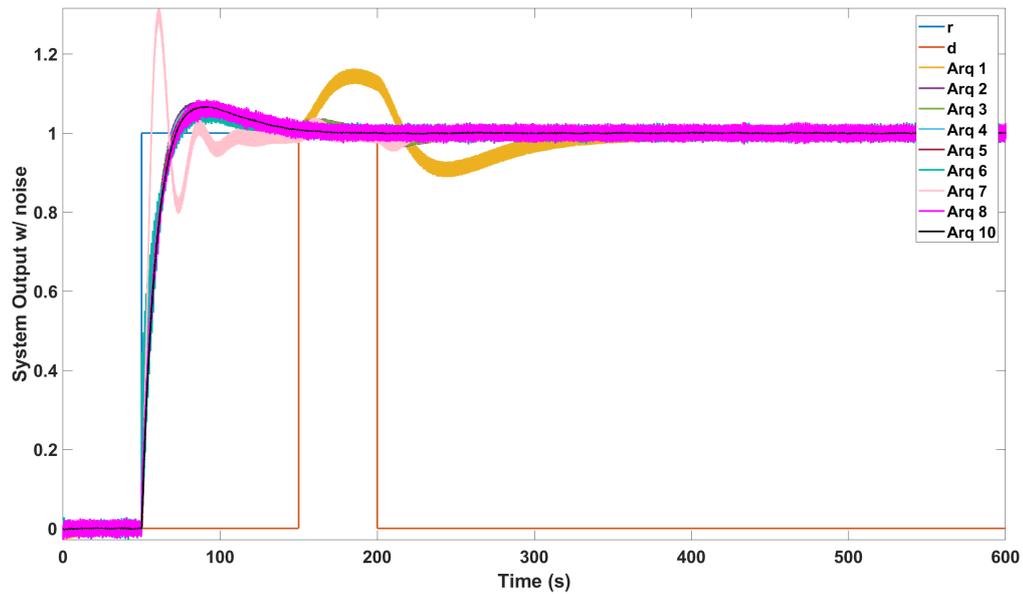


Figura 16 – Ensaio I: Saídas  $y_n$  do sistema com ruído, para todas as arquiteturas

Na Figura 16 são observadas as saídas  $y_n$  do sistema, com as linhas do gráfico mais preenchidas devido ao ruído. Nas arquiteturas 5 e 10, os sinais não apresentam ruídos tão grandes quanto às outras. Isso se dá pela presença do filtro estatístico  $Q_E$ , que atenua o ruído tanto na entrada do observador quanto na realimentação da malha.

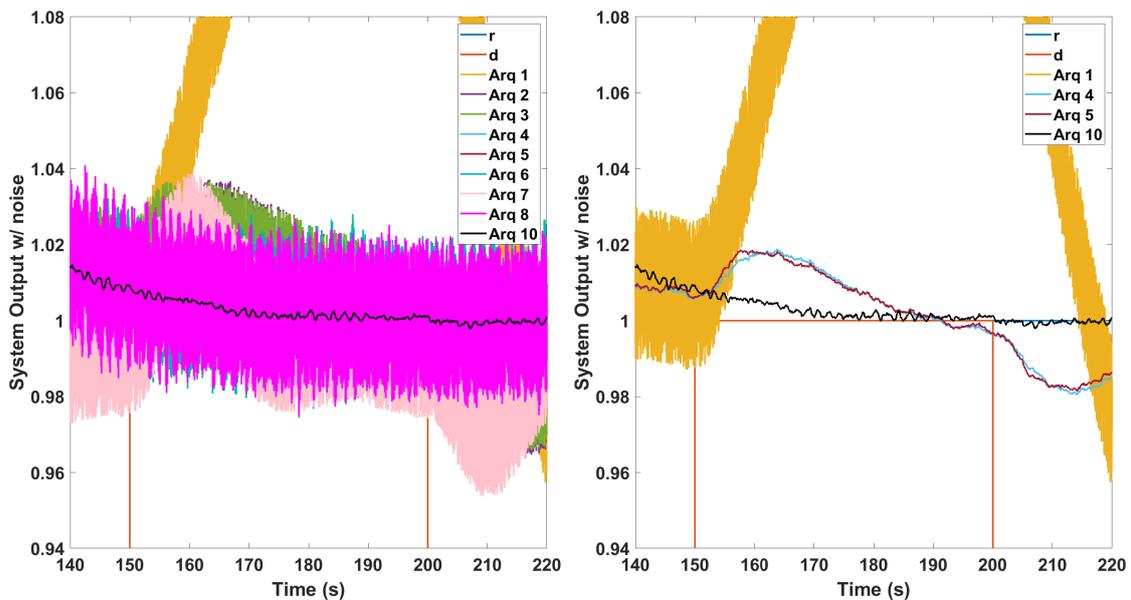


Figura 17 – Ensaio I: Saídas  $y_n$  do sistema com ruído, ampliada no momento do distúrbio

A Figura 17 apresenta os mesmos sinais de saída da Figura 16, mas ampliado no momento do distúrbio. No gráfico da esquerda são mostradas as saídas de todas as arquiteturas.

ras e no da direita, foram removidos os sinais mais ruidosos para visualizar os sinais que estavam sobrepostos.

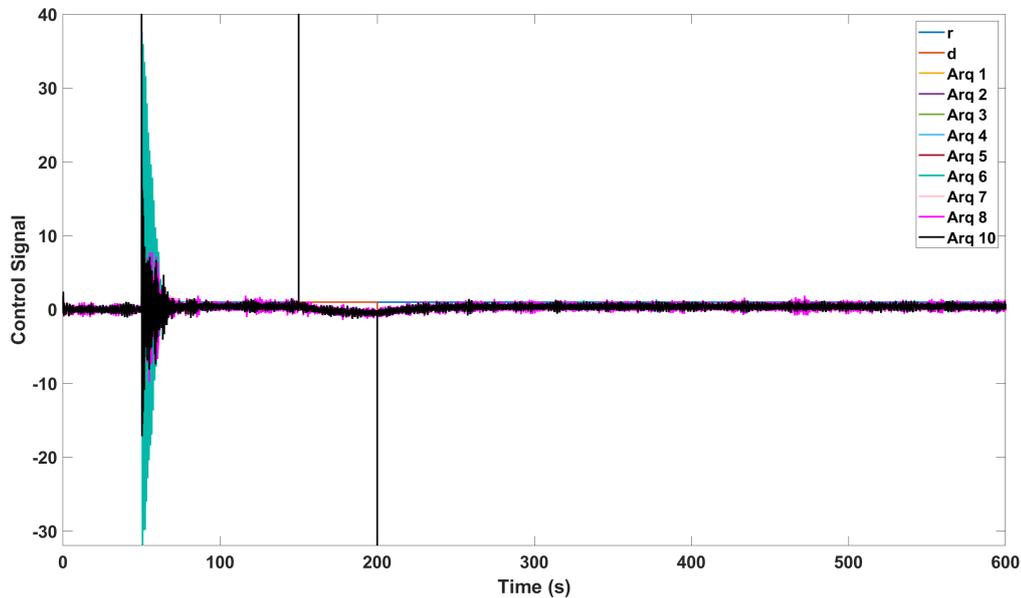


Figura 18 – Ensaio I: Sinal  $u$  do controlador, para todas as arquiteturas

O gráfico da Figura 18 mostra o comportamento do sinal de controle que compensa o degrau de entrada (instante  $T = 50s$ ). Nota-se também as diferenças das ações de controle de cada arquitetura para o mesmo comportamento. A ação de controle das arquiteturas 6, 7, 8 e 10, por exemplo, não estabilizam o sinal tão rápido no momento que o degrau da referência é aplicado quanto as arquiteturas 1, 2, 3, 4 e 5, cujos sinais de controle podem ser melhor visualizados nas Figuras A.6 e A.7 do Apêndice A.

Além dos sinais comentados, também foram calculados os indicadores de desempenho. Para esse cálculo, só foram considerados os sinais no intervalo de tempo  $100s > T > 400s$  para que a resposta ao degrau de referência não influencie no resultado, já que a análise é direcionada à influência do distúrbio em cima de cada arquitetura.

Arquitetura	IAE ( $10^4$ )	ISE ( $10^4$ )	TV ( $10^4$ )
Arq 1	82.3255	7.3308	71.6480
Arq 2	26.4593	0.5663	258.7114
Arq 3	26.5927	0.5741	69.8133
Arq 4	26.1814	0.5465	285.3585
Arq 5	26.3270	0.5555	2.4380
Arq 6	25.9831	0.5461	239.0089
Arq 7	17.4622	0.2136	$3.1575 \cdot 10^8$
Arq 8	27.9709	0.7347	$3.6836 \cdot 10^8$
Arq 9	$3.9889 \cdot 10^{11}$	$8.2309 \cdot 10^{20}$	$5.9387 \cdot 10^{12}$
Arq 10	27.2260	0.6879	$3.7902 \cdot 10^8$

Tabela 1 – Indicadores de desempenho Ensaio I

Após a análise das saídas  $y$ ,  $y_n$  e  $u$  e os indicadores de desempenho, percebeu-se que as arquiteturas 2, 3, 4 e 5 (Figuras 4, 5, 6 e 7, respectivamente) possuíam comportamentos mais próximos do desejado, sofrendo pouca influência dos distúrbios.

Na Tabela 1, é possível ver que as arquiteturas 5, 6 e 7 resultam em melhores IAE, e as arquiteturas 6,7 (Figuras 8 e 9, respectivamente) resultam em um melhor ISE. Nota-se que para o erro absoluto integrado, ambas as arquiteturas 6 e 7 apresentam bons resultados, ou seja, representam bons modelos para redução tanto do erro estacionário como erro transitório. É possível observar, também, que as arquiteturas 1, 3 e 5 apresentam melhores resultados de TV sendo sistemas com o sinal mais suavizado.

No apêndice A, sessão A.1, é possível verificar algumas saídas selecionadas com mais detalhe.

## 3.2 Sistemas com atraso

O estudo seguinte é dado por um sistema com atraso representado pela função de transferência  $G_p(s)$  (Equação (3.7)):

$$G_p(s) = \frac{1}{s(0.5s + 1)} e^{-0.5s} \quad (3.7)$$

Para o controlador foi usado um PID com os ganhos  $K_p = 0.866$ ,  $K_I = 0.073$  e  $K_D = 0.435$  obtidos no artigo Fortaleza et al. (2022).

Foram utilizadas as arquiteturas 2, 5 (Figuras 4 e 7, respectivamente) e a arquitetura somente com filtro estatístico (baseada na Figura 2 e nomeada Arq FE neste ensaio), nas quais foi utilizado do ruído  $\eta$  (sistema na Figura 14) com uma variância de  $\sigma^2 = 8 \cdot 10^{-4}$  e  $T_d$  com o valor de 10% da constante de tempo da função de transferência na Equação (3.7), utilizado no observador de distúrbio.

$$G_d(s) = \frac{1}{23s + 1} \quad (3.8)$$

$$T_d = 0.05s \quad (3.9)$$

$$Q(s) = \frac{1}{(0.05s + 1)^2} \quad (3.10)$$

$$\sigma_n = 7.53 \quad (3.11)$$

Como o sistema possui atraso e o sistemas de fase não-mínima  $Q(s)G_n^{-1}(s)$  não é realizável/instável, o Observador de Distúrbio (como mostrado na Figura 1) teve de sofrer modificações para se adequar ao modelo da Figura 13. Fatorando  $G_n(s)$ , tem-se:

$$G_n(s) = \frac{1}{s(0.5s + 1)} e^{-0.5s} = G_{n-}(s)G_{n+}(s) \quad (3.12)$$

onde

$$G_{n-}(s) = \frac{1}{s(0.5s + 1)} \quad (3.13)$$

$$G_{n+}(s) = e^{-0.5s} \quad (3.14)$$

Para efeitos de comparação, foram apresentados nas Figuras 19 e 20 o resultados da simulação. É possível verificar o sinal de saída do sistema  $y$  e o sinal de controle  $u$  para todas as arquiteturas apresentadas.

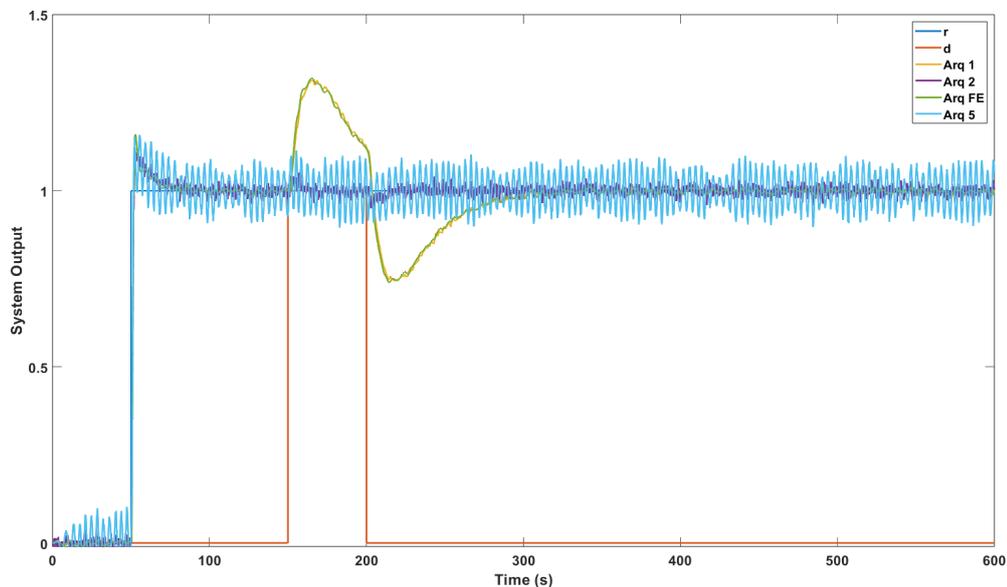


Figura 19 – Ensaio II: Saídas  $y$  do sistema sem ruído, para as arquiteturas 1, 2, 5 e FE

É possível notar que, assim como analisado na Sessão 3.1, o comportamento das arquiteturas na Figura 19 continuam com o comportamento esperado. O sinal de saída da arquitetura 5 é muito ruidoso, e esse comportamento pode ser justificado por o filtro

estatístico não estar bem calibrado para essa planta em questão. Para efeitos comparativos, foi realizado outro teste com a metade do valor de  $\sigma_n$  para a arquitetura 5 (Figura A.20), onde é observado uma redução do ruído.

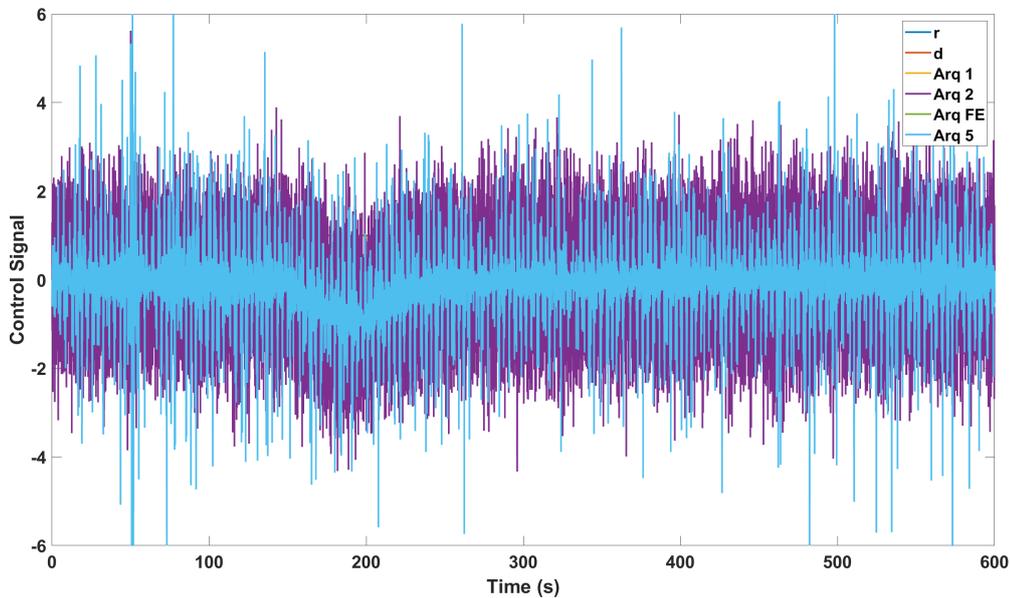


Figura 20 – Ensaio II: Sinal  $u$  do controlador, para as arquiteturas 1, 2, 5 e FE

Os sinais de controle observados na Figura 20 para as arquiteturas 2 e 5 são bastante ruidosos. No caso da arquitetura 2, esse ruído foi amplificado pelo observador de distúrbio, já na arquitetura 5, a amplificação resultante do observador soma com o filtro estatístico mal calibrado, como citado anteriormente. É possível ver resultado das simulações, separado para cada sinal de controle, na sessão A.2 do apêndice A.

Com os resultados da simulação, foram calculados os indicadores calculados para o intervalo de tempo  $100s > T > 400s$ , analisando somente a influência do distúrbio no sistema (Tabela 2).

Arquitetura	IAE ( $10^4$ )	ISE ( $10^4$ )	TV ( $10^4$ )
Arq 1	77.1049	15.5295	15.0135
Arq 2	10.1252	0.1640	154.5911
Arq FE	77.1637	15.5437	1.1031
Arq 5	45.8860	2.6676	437.2315

Tabela 2 – Indicadores de desempenho Ensaio II

Realizando as simulações e cálculo dos indicadores, foi notado que a arquitetura FE possui sinal  $y$  com comportamento próximo ao sistema original sem observador e filtro estatístico. Desse modo, a análise continua em cima das arquiteturas 2 e 5, que não são comprometidas pelo distúrbio.

### 3.3 Sistema tanques em cascata

Os tanques em cascata são conectados entre si por uma válvula ajustável, que regula o fluxo constante de água do reservatório de cima para o reservatório de baixo. Do tanque inferior, a água segue para um reservatório que, através de uma bomba, injeta o líquido no tanque superior novamente. A Figura 21 abaixo apresenta um esquemático de como os tanques estão conectados.

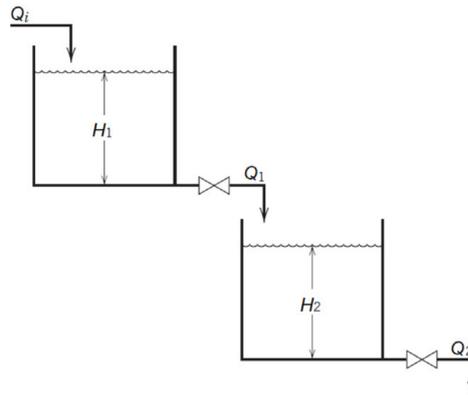


Figura 21 – Esquemático do sistema com dois tanques

Para a simulação computacional do par de tanques em cascata, tem-se a função de transferência da planta  $G_p(s)$  identificada através da ferramenta *MATLAB's System Identification Toolbox*, o que resultou na função de segunda ordem linearizada da Equação (3.15) abaixo que representa a dinâmica do sistema.

$$G_p(s) = \frac{0.002677}{(s + 0.0565)(s + 0.0192)} \quad (3.15)$$

Para o controlador PID foram usados os ganhos obtidos anteriormente no artigo [Fortaleza et al. \(2022\)](#), com  $K_P = 0.3105$ ,  $K_I = 0.0044$  e  $K_D = 4.1019$ .

Nos ensaios, foram utilizados apenas as arquiteturas 2, 5 (Figuras 4 e 7, respectivamente) e a arquitetura somente com filtro estatístico (baseada na Figura 2 e nomeada Arq FE neste ensaio). No Matlab, houve a manipulação de alguns parâmetros para obtenção de melhor resposta, dentre eles: o que diz respeito à variação de amplitude do ruído  $\eta$  e os valores de  $T_d$  para o observador de distúrbio. Nesta etapa, utilizou-se  $T_d$  como 2% da constante de tempo referente ao polo mais lento da planta  $G_p(s)$ .

$$T_d = 0.02 \cdot \frac{1}{0.0192} = 1.04167 \quad (3.16)$$

$$Q(s) = \frac{1}{(1.04167s + 1)^2} \quad (3.17)$$

$$\sigma_n = 0.4703 \quad (3.18)$$

$$G_d(s) = \frac{1}{23s + 1} \quad (3.19)$$

Para este ensaio, utilizou-se o mesmo bloco de ruído da Figura 14.

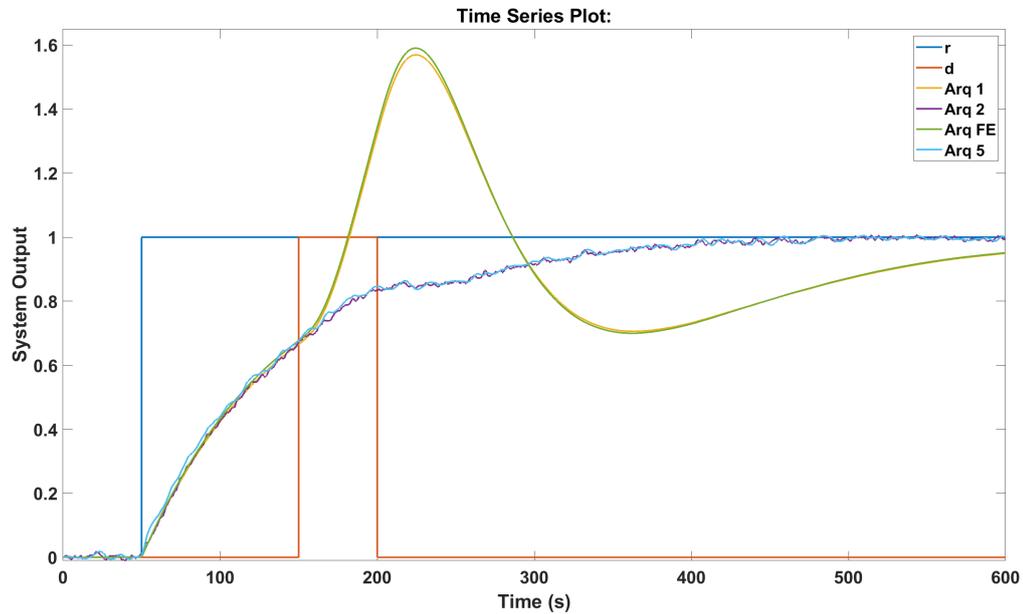


Figura 22 – Ensaio III: Saídas  $y$  do sistema sem ruído, para as arquiteturas 1, 2, 5 e FE

Entre as saídas dos sistemas mostrados na Figura 22, é possível notar comportamento semelhante de ambos sistemas com observador de distúrbio (arquiteturas 2 e 5), apresentando comportamento subamortecido com tempo de acomodação menor e uma resposta significativamente menor comparados com os sistemas sem observador.

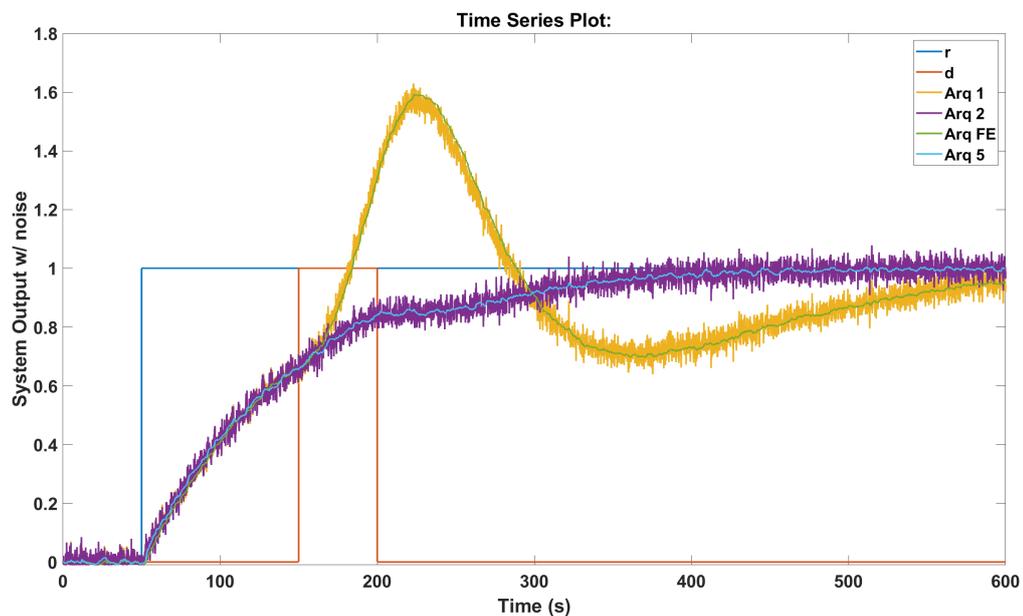


Figura 23 – Ensaio III: Saídas  $y_n$  do sistema com ruído para as arquiteturas 1, 2, 5 e FE

Após a simulação e análise dos sinais  $y$ ,  $y_n$  e  $u$  para  $G_p(s)$  (eq. 3.15), notam-se os sinais das arquiteturas 3 e 5 menos ruidosas (Figura 23), quando comparados com as outras duas arquiteturas. Este resultado era esperado, já que ambas arquiteturas possuem o filtro estatístico.

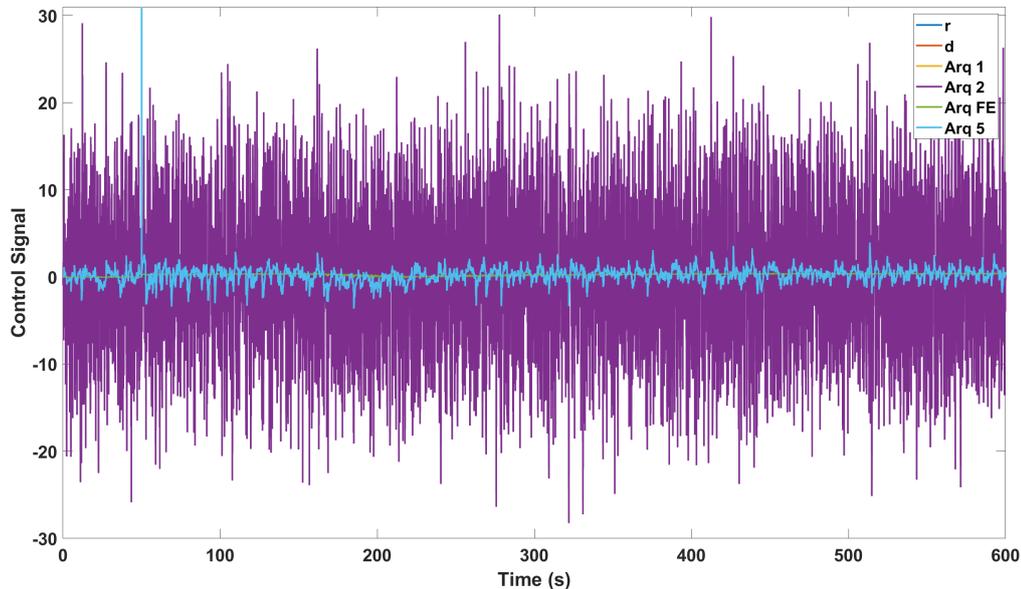


Figura 24 – Ensaio III: Sinais  $u$  do controlador para as arquiteturas 1, 2, 5 e FE

É possível perceber também pelo sinal de controle (Figura 24) que o sinal da arquitetura 2 é mais ruidoso, sendo o sinal de controle da arquitetura 5 o mais suave. Na seção A.3 do apêndice A, os sinais das arquitetura 1 e 2 foram removidos para ser possível observar os sinais de controle que estavam sobrepostos por elas.

Analisando os indicadores de desempenho no intervalo  $100s > T > 400s$ , em que o sinal sofre influência do distúrbio, chega-se à seguinte Tabela 3:

Arquitetura	IAE ( $10^4$ )	ISE ( $10^4$ )	TV ( $10^4$ )
Arq 1	83.0084	31.0159	254.8965
Arq 2	48.6942	14.2235	1096.5389
Arq FE	83.9816	31.7632	6.8974
Arq 5	47.2271	13.3671	35.1549

Tabela 3 – Indicadores de desempenho Ensaio III

Então pode-se dizer que as arquiteturas 1 e FE possuem um desempenho na resposta ao distúrbio relativamente parecida (notável quando analisada a Figura 22), assim como as arquiteturas 2 e 5. Porém, quando  $TV$  é analisado, as arquiteturas 2 e 5 possuem sinal de saída mais satisfatório, tem-se que a arquitetura 5 possui uma melhor filtragem.

---

## 3.4 Conclusão das simulações

Com a simulação dos três sistemas é constatado que as arquiteturas 2 e 5 apresentam os melhores resultados em relação à resposta ao distúrbio, sofrendo quase nenhuma influência dele. Quanto à não amplificação dos ruídos nos sinais de controles, as arquiteturas FE e 5 tiveram melhor desempenho. Assim, a arquitetura 5 é selecionada como a melhor abordagem. Uma forma de melhorar sua aplicabilidade ainda mais seria na calibração mais precisa do filtro estatístico.

Quanto ao sinal de controle, as arquiteturas FE e 5, possuem um melhor desempenho. Como o sinal medido nas simulações foram os sinais  $u$ , ou seja, sinal do controlador menos sinal do observador, não foi possível conferir a dinâmica do controlador separadamente no momento que o distúrbio foi aplicado.

## 4 Ensaio Experimentais

Neste capítulo serão apresentados os ensaios e os resultados das arquiteturas antes e depois da inserção do observador de distúrbio, analisando qual a melhor arquitetura e calibração para o observador.

### 4.1 Sistema de tanques em cascata

A planta real utilizada no laboratório (mostrada na Figura 25) é composta por quatro reservatórios, em que cada par segue o esquemático da Figura 21 do capítulo anterior. Os tanques superiores (1 e 2) são alimentados pelas bombas, que retiram a água do reservatório. Os inferiores (3 e 4) são alimentados tanto pelas bombas quanto pelos tanques superiores, seguindo o esquemático abaixo na Figura 26.

A bancada é controlada por um controlador lógico programável, presente no painel elétrico da Figura 27, onde são feitas as conexões com os sensores e atuadores, além de dispositivos de proteção. As bombas (atuadores) estão conectadas ao controlador através de inversores, que permitem ligá-las de maneira mais apropriada e oferecem uma proteção maior contra sobrecargas.



Figura 25 – Foto do sistema real em laboratório

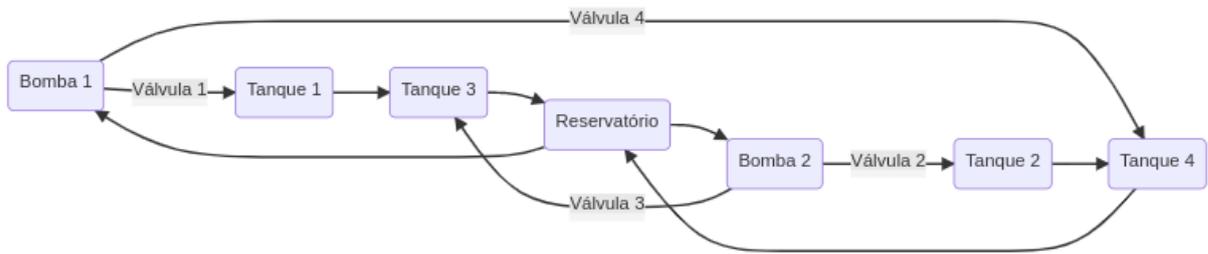


Figura 26 – Esquemático da planta real



Figura 27 – Foto do painel com CLP (Controlador Lógico Programável) e suas conexões elétricas

## 4.2 Sistemas de automação e comunicação

Para implementar o que foi estudado nas simulações anteriores na planta real presente no laboratório, foram utilizados os programas *Automation Studio*, para fazer a conexão entre o computador da bancada e o controlador lógico programável (CLP) do sistema de tanques, e *Matlab*, para implementar as malhas de controle e direcionar os sinais para planta real através da ferramenta *Simulink*, e gerar os gráficos e cálculos através de códigos. Primeiro foi necessário adaptar a malha montada no *Simulink*, utilizada durante a fase de simulação, para que haja a comunicação com o sistema real. Para isso, o bloco *OPC Write* para enviar o sinal da bomba e *OPC Read* para receber os dados do sensor do tanque (visíveis na Figura 28).

É importante citar que para o sistema real, tanto o sinal de entrada da planta quanto o sinal de saída são corrigidos para seus pontos de operação, sendo eles 52% para a bomba

e 24% para a saída esperada do sensor no ponto de operação da bomba. Estes pontos de operação foram definidos em estudos anteriores, quando realizada a identificação da função de transferência que representa a dinâmica da bancada. O sinal de entrada da bomba se refere à porcentagem da potência que a bomba será ligada e o sinal de saída do sensor se refere à porcentagem do nível da água no tanque.

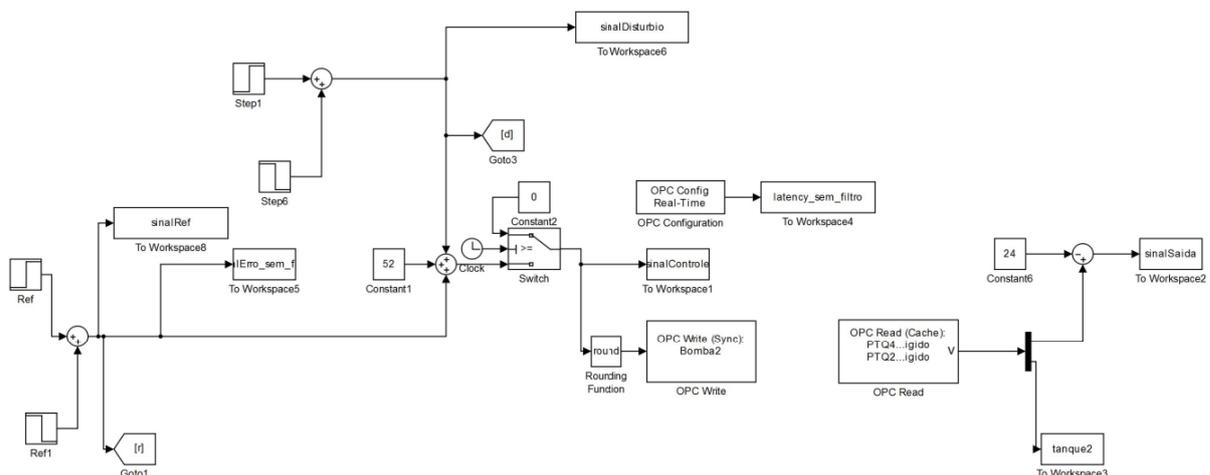


Figura 28 – Malha aberta conectada ao sistema real

É possível notar que antes do sinal que chega no bloco "OPC Write" (responsável por enviar o sinal desejado para a bomba, passando pelo inversor), ele segue por um bloco chamado "Switch". Este bloco é responsável por nos últimos cinco segundos do teste, chavear o sinal de entrada para a constante zero, desta forma, a bomba para e o tanque se esvazia sem a necessidade de o operador ter que desligar manualmente a bomba pelo *Automation Studio*.

O *Automation Studio* (Figura 29) é um programa utilizado para programar e realizar a comunicação com o CLP. O sistema no laboratório tem seu CLP programado utilizando a linguagem *Ladder* para realizar os links dos sensores e atuadores com as tags definidas pelo programa, desta forma, é possível realizar a comunicação pelo *Matlab* citada anteriormente.

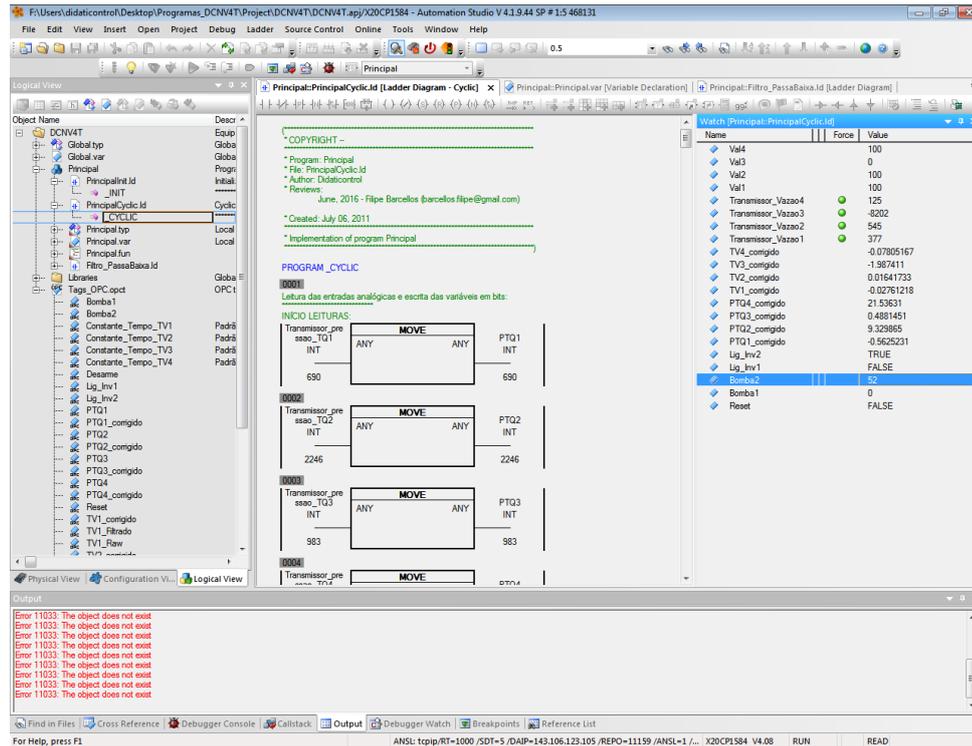


Figura 29 – Tela do programa *Automation Studio*

### 4.3 Comparação entre o sistema simulado e o real

Em um primeiro momento, foi verificado se a função de transferência  $G(s)$  utilizada na etapa de simulação estava condizente com a dinâmica real do sistema de tanques no laboratório. Para isso, aplicou-se um degrau na entrada depois que o sistema se estabilizou no ponto de operação para conferir se a resposta ao degrau da planta real era próxima à resposta da simulação.

Para este teste, foram utilizados a função de transferência (3.15) para  $G(s)$  e as mesmas constantes para o controlador citadas na Seção 3.3.

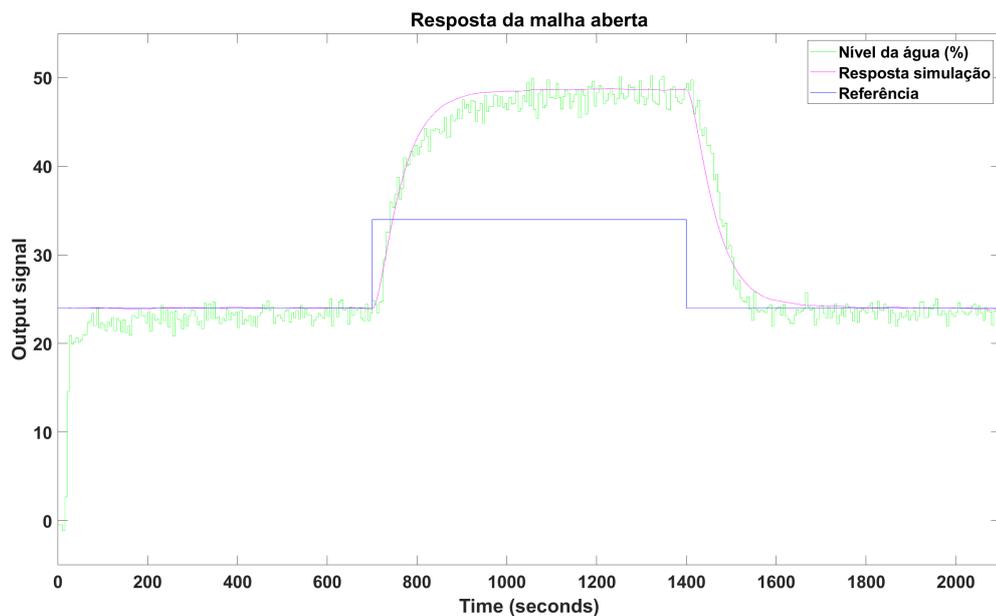


Figura 30 – Resposta malha aberta

Na Figura 30 acima, nota-se que a resposta ao degrau apresentado pelo sistema de tanques em cascata do sistema real em laboratório condiz com a resposta ao degrau do sistema estudada na simulação. Com isso, é possível dar seguimento aos estudos para a aplicação do observador de distúrbio.

## 4.4 Análise das arquiteturas sem Observador de Distúrbio

Para uma futura análise da aplicabilidade do observador de distúrbio, é necessário obter os resultados das arquiteturas já existentes sem o observador, as quais serão referidas como MFB (Malha Fechada Base, análoga à Figura 3) e MFE (Malha Fechada Somente com Filtro Estatístico, análoga à Figura 2). Elas foram montadas no *Simulink* e podem ser vistas nas Figuras 31 e 32.

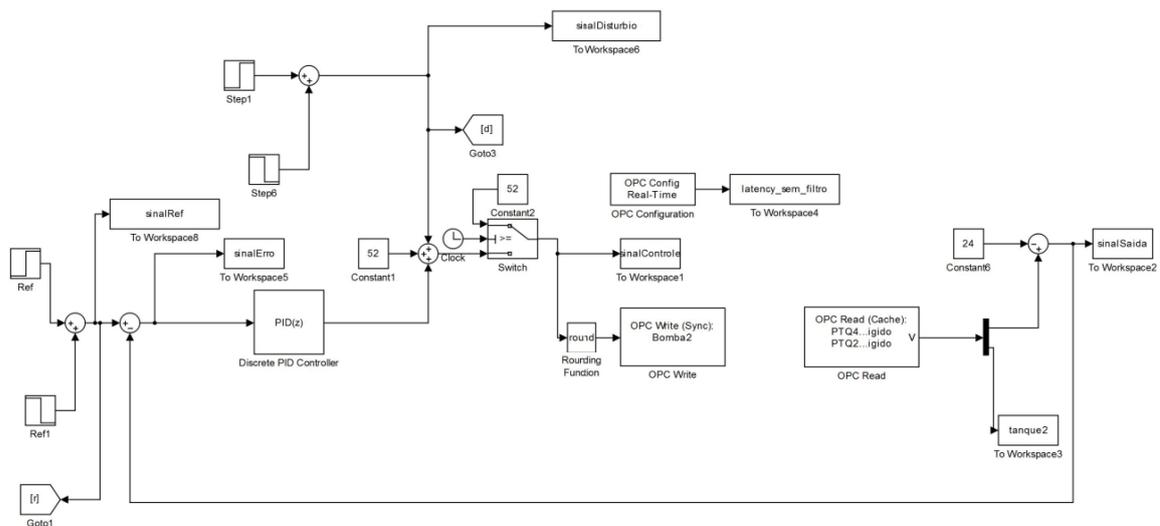


Figura 31 – Arquitetura Malha Fechada Base (MFB)

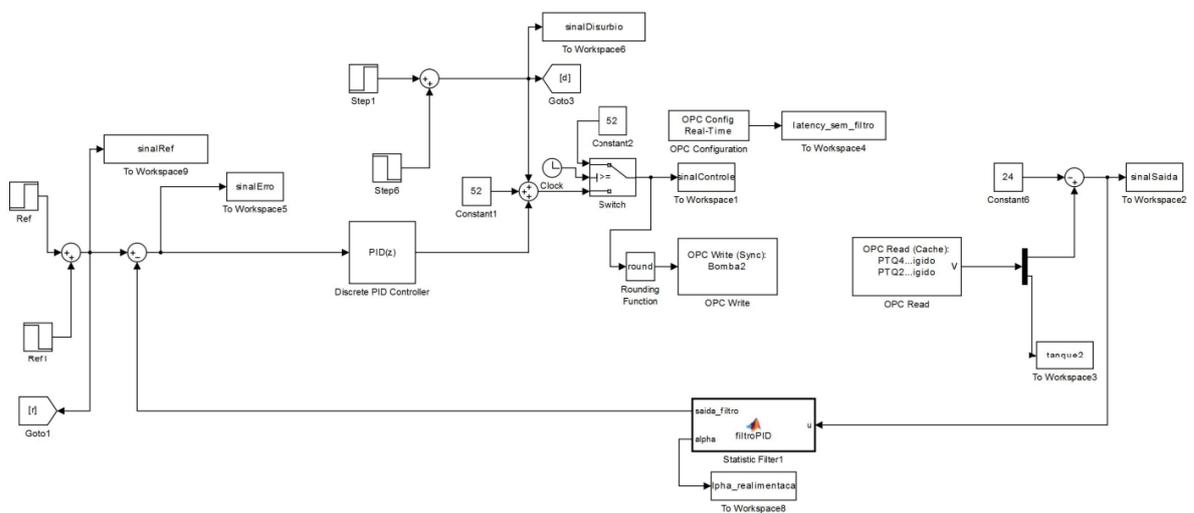


Figura 32 – Arquitetura Malha Fechada somente com Filtro Estatístico (MFE)

Para testar as arquiteturas no *Simulink*, são necessários estabelecer os valores das variáveis utilizadas como parâmetros do sistema. Suas definições são carregadas para o *Workspace* do *Matlab* através do Código B.1. O tempo de amostragem do sistema é alto, isso se deve ao sistema real ser lento e a taxa de atualização não precisar ser tão baixa, portanto foi estabelecida a variável  $T_s = 5s$  para utilizar no *Simulink*. Também pode-se observar que as funções de transferência da planta simulada e do controlador estão de acordo com as apresentadas na Seção 3.3.

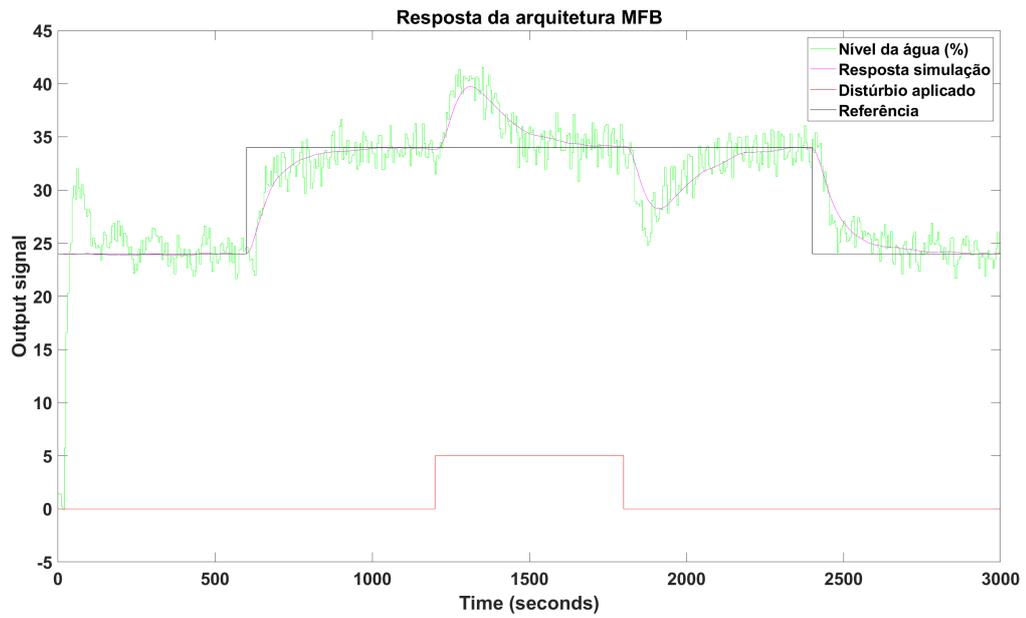


Figura 33 – Resposta arquitetura MFB

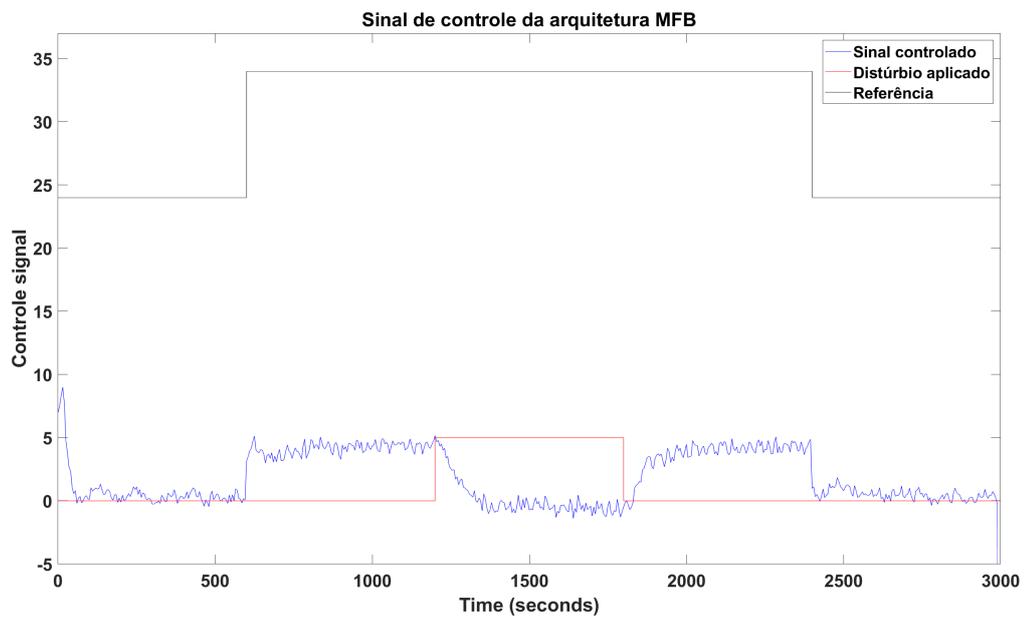


Figura 34 – Sinal de controle da arquitetura MFB

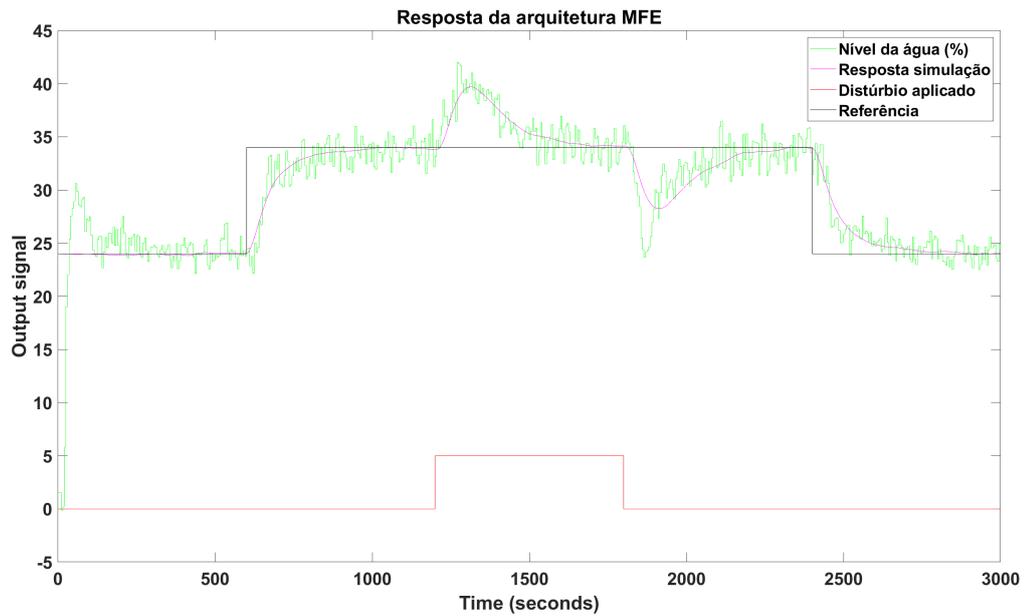


Figura 35 – Resposta arquitetura MFE

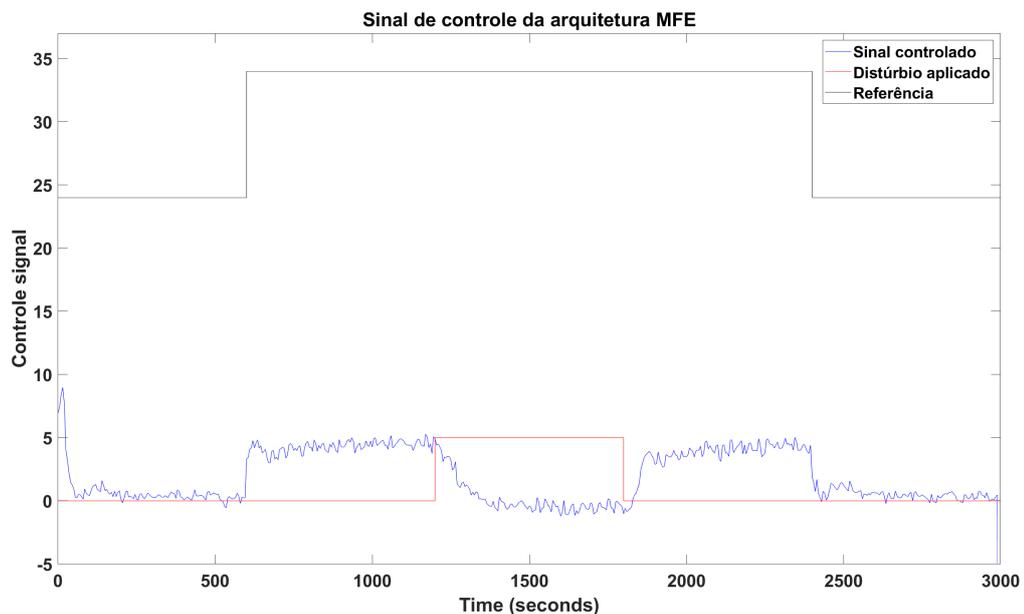


Figura 36 – Sinal de controle da arquitetura MFE

Observando as Figuras 33 e 35, é bem notável a influência que o distúrbio causa na resposta do sistema, e o mesmo acontece nos sinais de controle das Figuras 34 e 36, que tenta corrigir o sinal. No início do teste, há um *overshoot* devido à subida do nível de água no tanque para atingir o ponto de operação. Este comportamento se deve ao sistema ter inicializado com o tanque completamente vazio. Para implementações a nível industrial, a malha fechada com o controlador só é ativada quando o sistema atinge o ponto de operação na malha aberta, portanto o pico que ocorre nos primeiros segundos do teste não são significativos. No entanto,

para fins de melhor análise dos indicadores, um novo teste foi realizado com o nível da água no tanque estando no ponto de operação no primeiro instante.

Para os testes seguindo no ponto de operação, a potência da bomba foi introduzida manualmente no *Automation Studio* para garantir que os testes iniciariam com o nível de água no ponto de operação esperado.

Foi alterada a constante zero conectada ao bloco "Switch" para 52%, desta forma não seria necessário introduzir manualmente a potência da bomba pelo operador toda vez que realizasse testes em sequência. O que garante que quando a análise no *Simulink* se encerra, a planta já está no ponto desejado para iniciar o teste seguinte.

Seguindo os testes com o tanque iniciando no ponto de operação, foram obtidos os seguintes respostas (Figuras 37, 38, 39 e 40) :

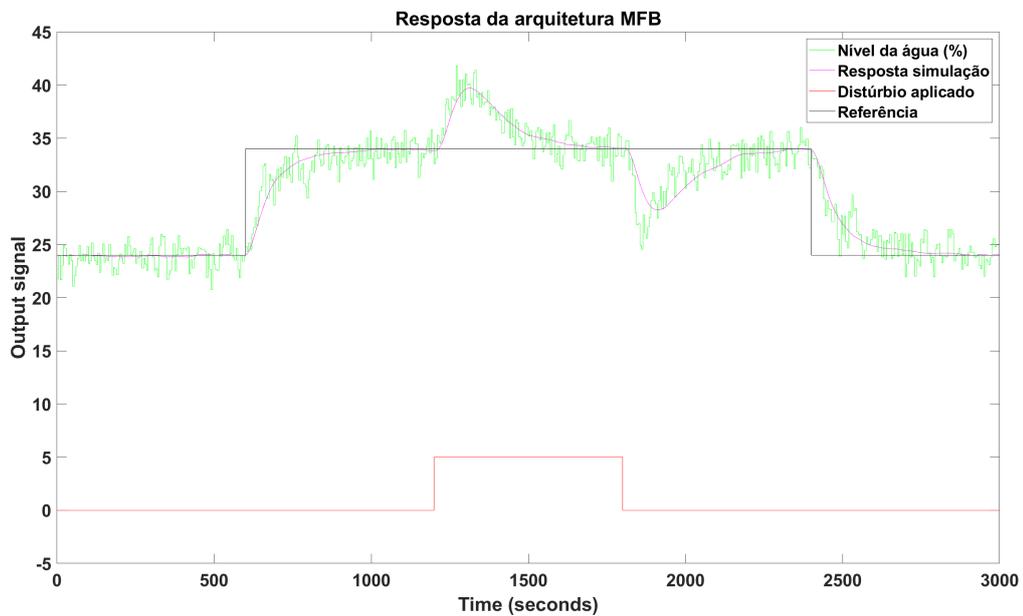


Figura 37 – Resposta arquitetura MFB no ponto de operação

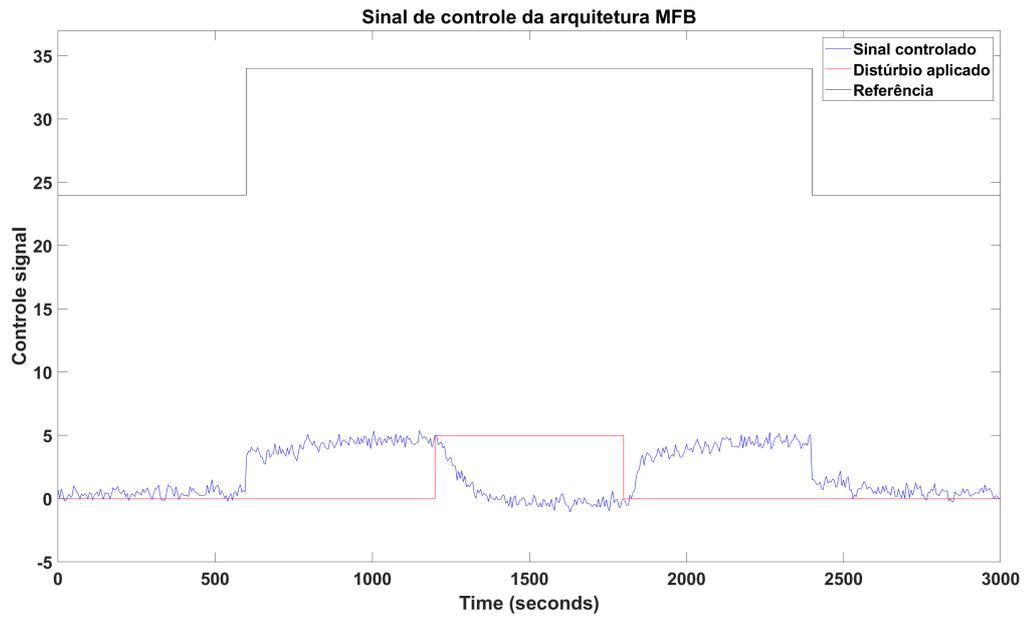


Figura 38 – Sinal de controle da arquitetura MFB no ponto de operação

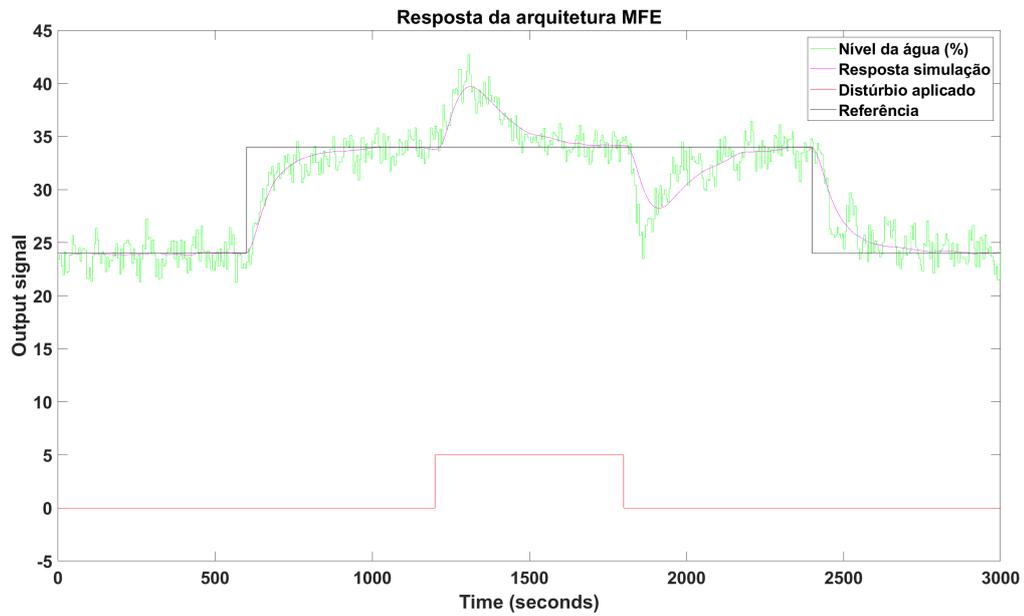


Figura 39 – Resposta arquitetura MFE no ponto de operação

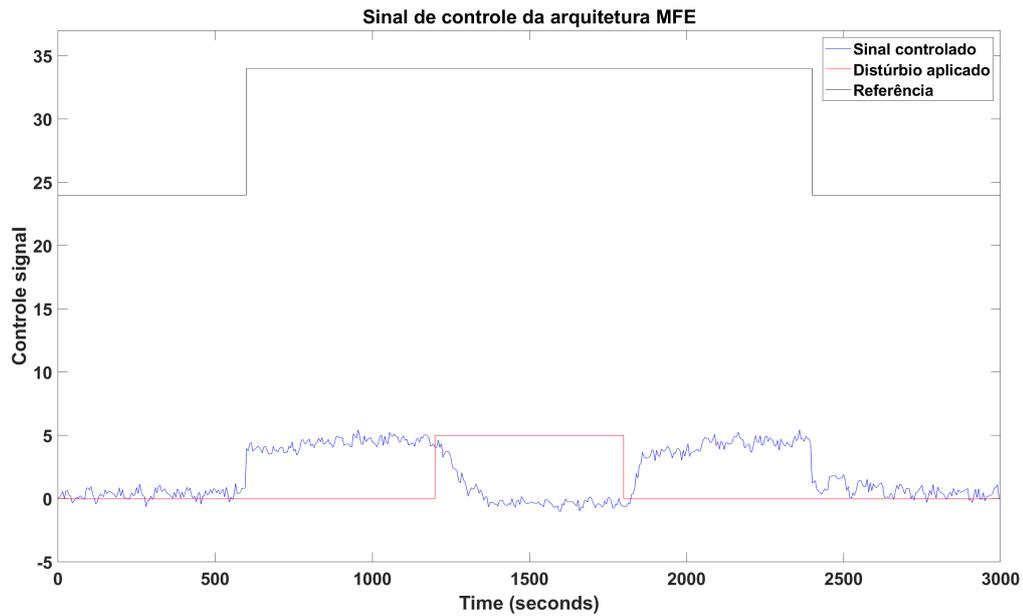


Figura 40 – Sinal de controle da arquitetura MFE no ponto de operação

Com esses dados coletados, é possível calcular os indicadores de desempenho, e para esse cálculo, utilizaram-se os Códigos B.2, B.3 e B.4 presentes no apêndice B. Para este cálculo foram utilizados os dados dos sinais no intervalo  $1100s > t > 2300s$  por ser um intervalo que a saída já se estabilizou do degrau de referência e ser o intervalo que o distúrbio influencia o sistema. Logo, os parâmetros de desempenho das arquiteturas MFB e MFE são:

Arquitetura	IAE ( $10^5$ )	ISE ( $10^5$ )	TV ( $10^5$ )
MFB	16.029664421	64.835842034	2.930483393
MFE	15.914412861	68.10660862	2.249243663

Tabela 4 – Indicadores de desempenho das arquiteturas MFB e MFE no ponto de operação

Os indicadores da Tabela 4 serão utilizados como parâmetro de comparação nos resultados da inserção do observador de distúrbio na malha fechada.

## 4.5 Inclusão do Observador de Distúrbio

Para análises futuras, o sistema de malha fechada com observador de distúrbio é referido por MOD (análogo à Figura 4) e o sistema de malha fechada com observador de distúrbio e com filtro estatístico é referido por MOF (análogo à Figura 7).

Em relação ao estudo da inserção do observador no sistema, inicialmente foi testado com o valor de  $T_d$  igual a 2% da constante de tempo do polo mais lento da planta ( $T_b$ ), porém o teste foi interrompido logo no começo, pois o sinal enviado com a potência da bomba estava acima de 1000% (a bomba opera com o sinal entre 0% e 100%).

Para não ocorrer danos à bomba, foi introduzido um saturador na entrada da bomba para que o sinal de potência enviado permanecesse entre o intervalo de 0 a 80. Optou-se por testar com o mesmo valor de  $T_d$  apenas para capturar alguns dados da saída do sistema.

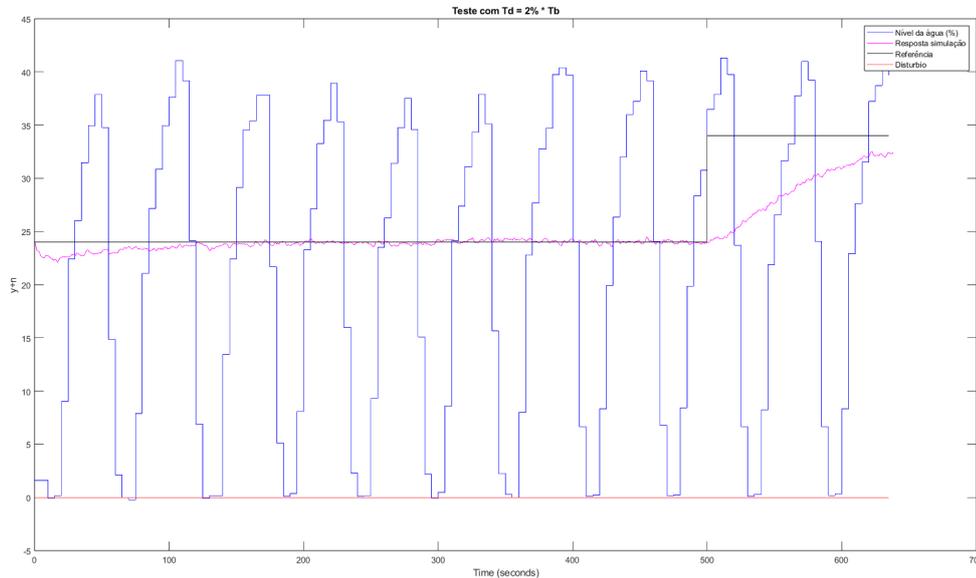


Figura 41 – Resposta do sistema com  $T_d = 2\% \cdot T_b$

Ficou evidente que seria necessário calibrar melhor o filtro  $Q(s)$  para chegar a um observador aplicável, então foram realizados diversos testes com valores de  $T_d$ , analisando a resposta do sistema, o sinal de controle e os indicadores de desempenho TV, IAE e ISE.

Para automatizar o teste com diferentes valores de  $T_d$ , foi criado o Código B.5 no qual o vetor "*p\_vector*" armazena todos os valores das porcentagens a serem testadas, sendo usado  $T_d$  (Equação (4.1)) para obter o valor do parâmetro  $T_d$  do filtro  $Q(s)$  do observador de distúrbio.

$$T_d = \frac{p\_vector[i]}{100} \cdot T_b \quad (4.1)$$

Na função 4.1,  $i$  indica a posição no vetor em que a porcentagem desejada está localizada. Após obter o valor de  $T_d$ ,  $Q(s)$  é calculado com a Equação (2.1) com  $q = 2$ .

O bloco do observador de distúrbio incluído nas arquiteturas (representado no *Simulink* pelo bloco *subsystem*), segue o mesmo formato apresentado na Figura 1, mas com uma correção de 1.5 para a saída da malha, como é possível ver abaixo (Figura 42).

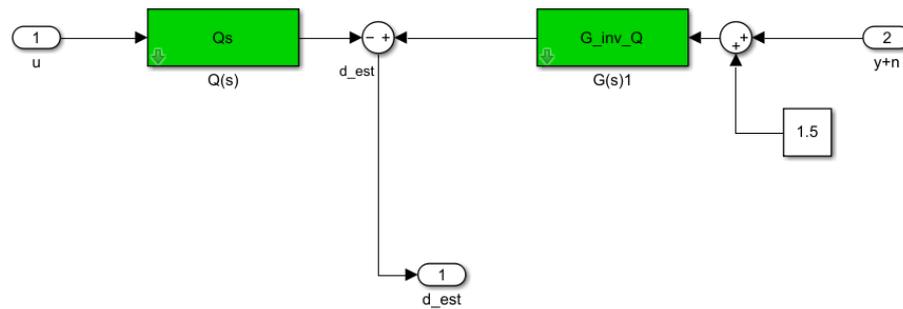


Figura 42 – Estrutura interna do bloco Observador de Distúrbio

Esta correção é necessária, pois, no ponto de operação, o sinal  $y + n$  possuía valores mínimos abaixo de zero.

Adicionando o bloco com o observador (Figura 42), tem-se os seguintes esquemáticos montados no *Simulink* para as arquiteturas:

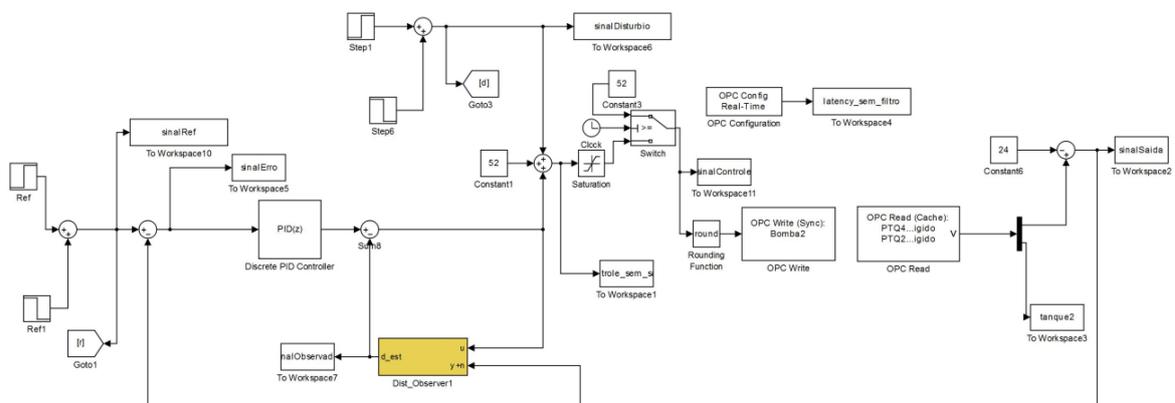


Figura 43 – Arquitetura base com Observador de Distúrbio (MOD)

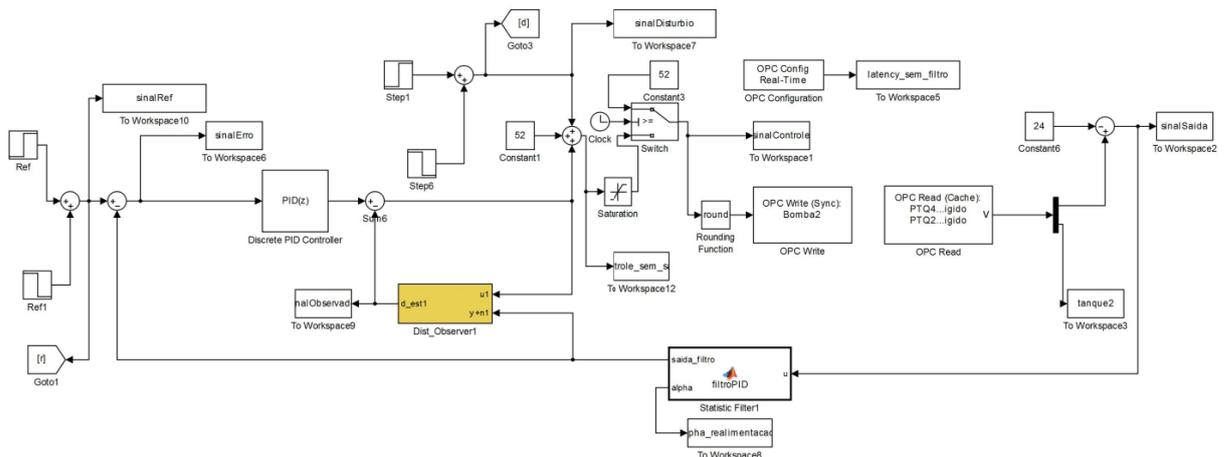


Figura 44 – Arquitetura base com Observador de Distúrbio e Filtro Estatístico (MOF)

É importante reparar que nessas arquiteturas, a constante conectada com o "Switch" já está modificada para quando um teste finalizar, o tanque já estar no ponto de operação para que o teste seguinte seja iniciado com o nível de água correto.

## 4.6 Análise dos dados - Arquitetura sem filtro estatístico (MOD)

Após coletar os dados com os ajustes feitos em Q(s), foi criado o Código B.6 com o objetivo de automatizar a análise de TV, IAE e ISE para todos os valores de  $T_d$  testados. De início, foi utilizado  $filter = columns$ , sendo  $columns$  o número de elementos no vetor  $p\_vector$ , para se coletarem os valores dos indicadores para cada porcentagem utilizada.

A Tabela 5 apresenta os valores dos indicadores calculados para o intervalo  $1100s > t > 2300s$  de todos os ensaios realizados. A partir dela foi possível montar os gráficos nas Figuras 45, 46 e 47 abaixo, tornando mais visível a diferença entre os resultados para cada porcentagem utilizada no cálculo de  $T_d$ .

Porcentagem de $T_b$	IAE	ISE	TV
30	1051069.8338	2545515.9118	814639.9246
35	1093136.8103	3550859.3398	784454.9072
40	1134251.3809	3079202.7349	821730.2386
45	1137354.7268	3131417.8197	938195.5935
50	1049237.2856	3245728.2953	814211.6702
55	1014885.3893	2737791.7211	786756.1439
60	1105675.9071	3204012.4705	892383.7508
65	1097795.9862	2917745.0379	826364.9996
70	1057393.7588	2878922.9592	761393.9739
75	1126896.6293	3068661.6558	818715.5803
80	1100525.0301	3047947.6588	783629.3595
85	1123030.7121	2978687.7536	860222.4424
90	1120639.6637	3100849.7138	788734.2012
95	1181377.0065	3152573.4071	825319.8707
100	1220335.144	3549284.3395	783012.6167
110	1118544.7025	3045315.6998	723539.2758
120	1284731.8783	3990043.967	814509.6252

Tabela 5 – Indicadores de performance para arquitetura base com observador de distúrbio (MOD)

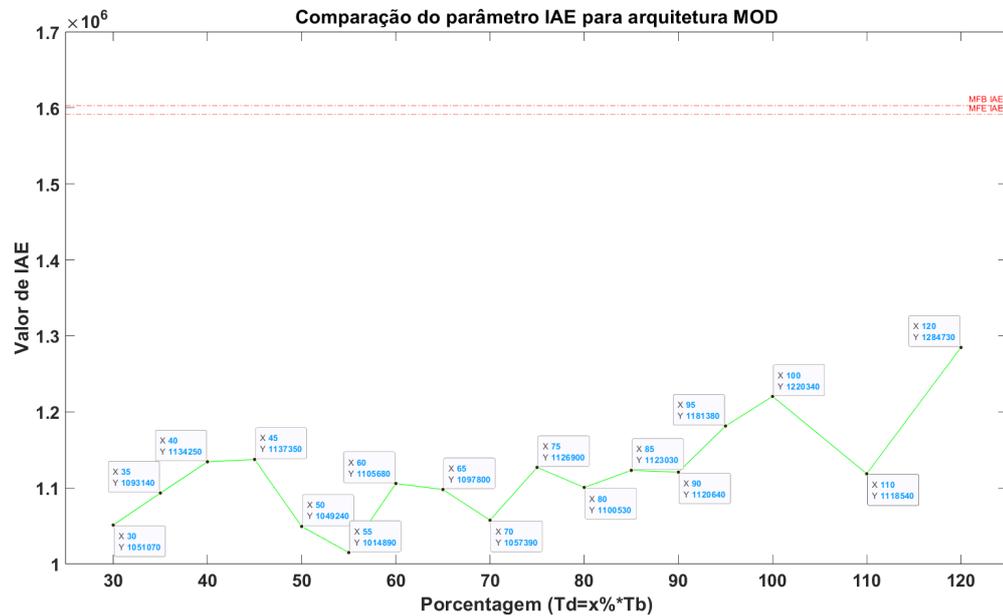


Figura 45 – Parâmetro IAE para cada teste na arquitetura MOD.

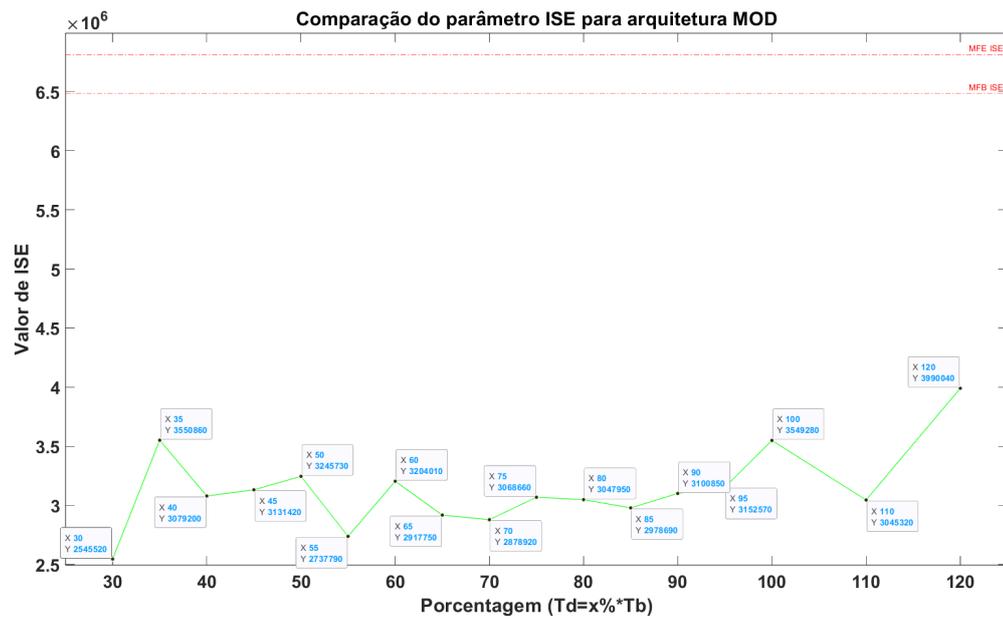


Figura 46 – Parâmetro ISE para cada teste na arquitetura MOD.

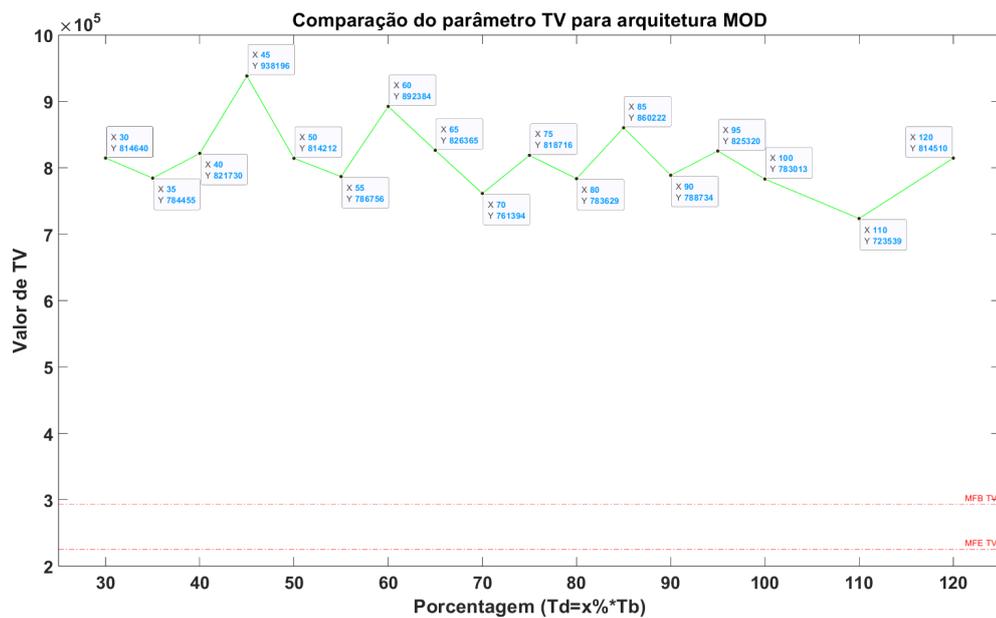


Figura 47 – Parâmetro TV para cada teste na arquitetura MOD.

Utilizando novamente o Código B.6, mas desta vez com *filter* = 6, é possível selecionar as cinco melhores porcentagens olhando para cada indicador.

Seis melhores IAE (ordem crescente): 55%, 50%, 30%, 70%, 35%, 65%

Seis melhores ISE (ordem crescente): 30%, 55%, 70%, 65%, 85%, 110%

Seis melhores TV (ordem crescente): 110%, 70%, 100%, 80%, 35%, 55%

Ao final, o Código B.6 exibe os melhores valores dentre os filtrados que aparecem tanto dentre os com melhores TV quanto nos com melhores IAE e ISE. Com a análise das Figuras 45, 46 e 47 e dos melhores valores de cada indicador, obtém-se dois candidatos a melhor observador de distúrbio que são os observadores construídos com  $T_d = 55\% \cdot T_b$  e  $T_d = 70\% \cdot T_b$ . Para o valor de 55%, tem-se as Figuras 48 e 49 com a saída do ensaio e sinal de controle, respectivamente, e para 70%, as Figuras 50 e 51.

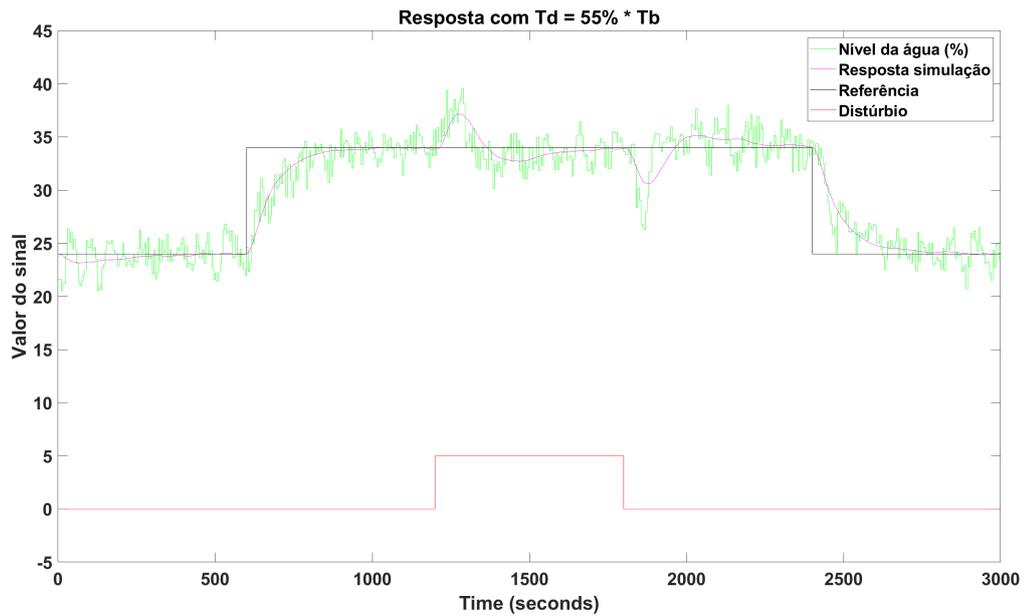


Figura 48 – Resposta do sistema para  $T_d = 55\% \cdot T_b$

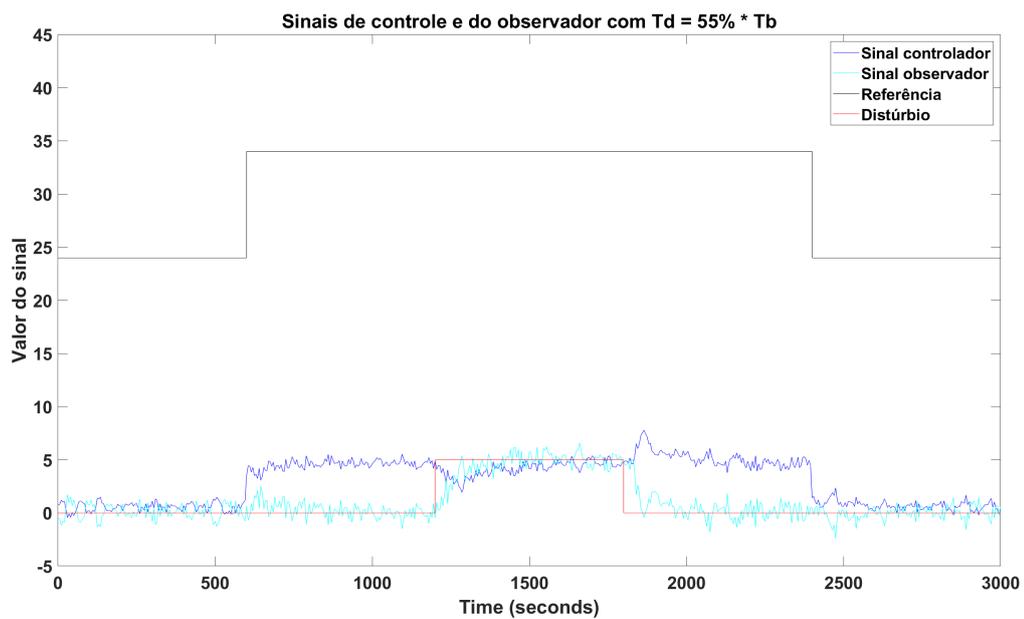
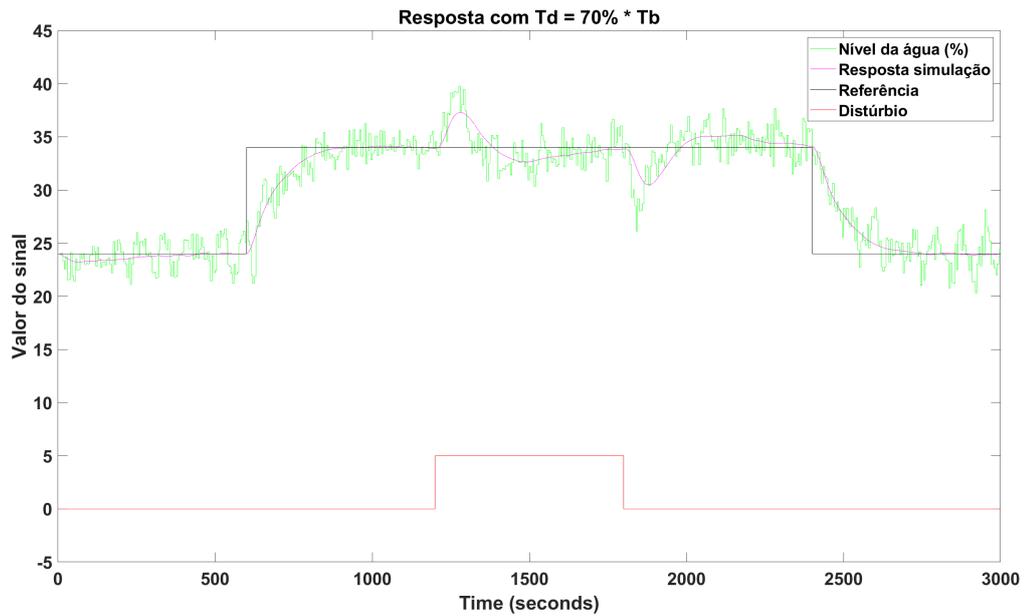
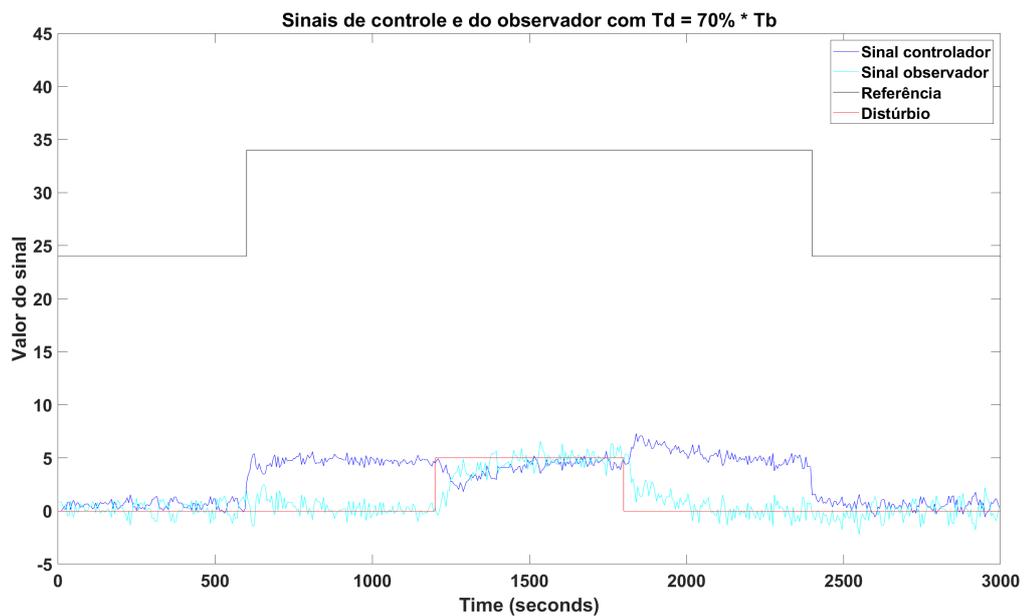


Figura 49 – Sinal do controlador para  $T_d = 55\% \cdot T_b$

Figura 50 – Resposta do sistema para  $T_d = 70\% \cdot T_b$ Figura 51 – Sinal do controlador para  $T_d = 70\% \cdot T_b$ 

Analisando comparativamente agora as melhores porcentagens para obter  $T_d$  (55% e 70%) com a arquitetura sem observador de distúrbio MFB, têm-se os gráficos abaixo, comparando o nível de água dado na saída do sistema (Figura 52), o sinal de controle (Figura 53) e comparando os dois sinais do observador (Figura 54)

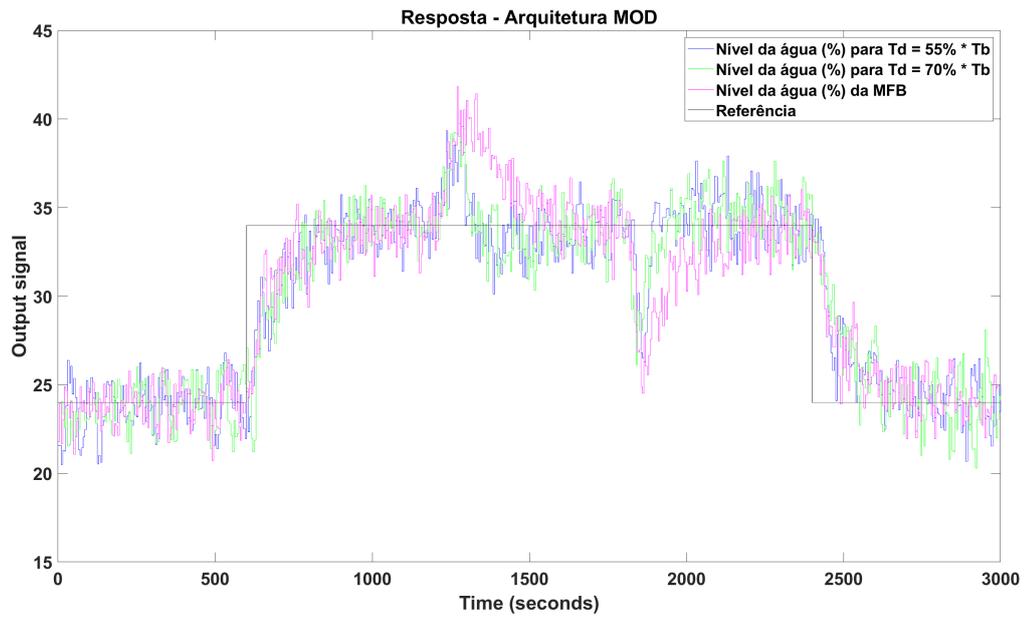


Figura 52 – Sistema MOD. Sinais do nível da água em porcentagem.

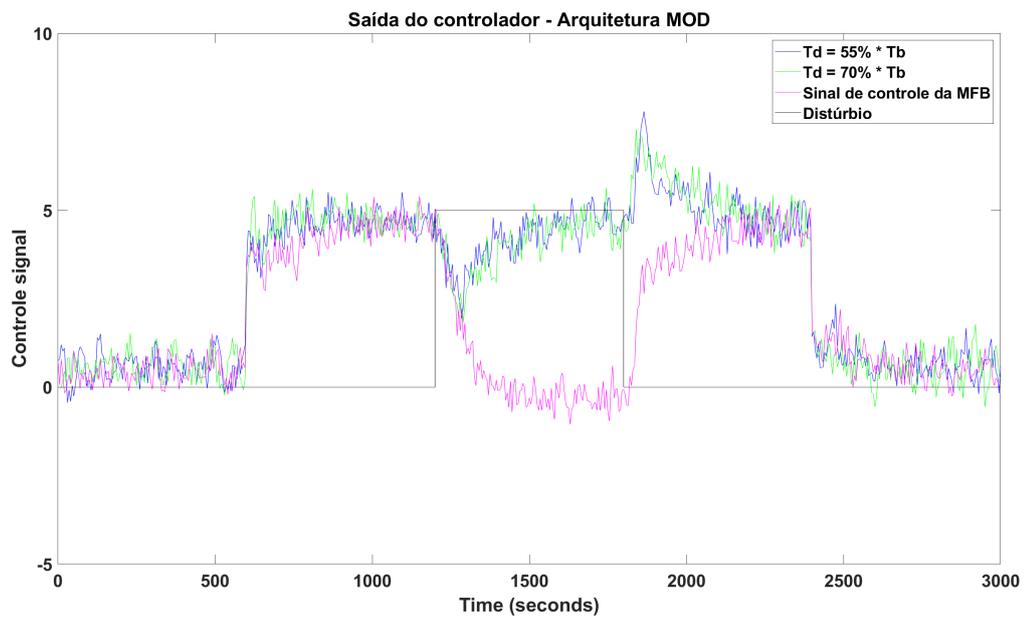


Figura 53 – Sistema MOD: Sinais da saída do controlador.

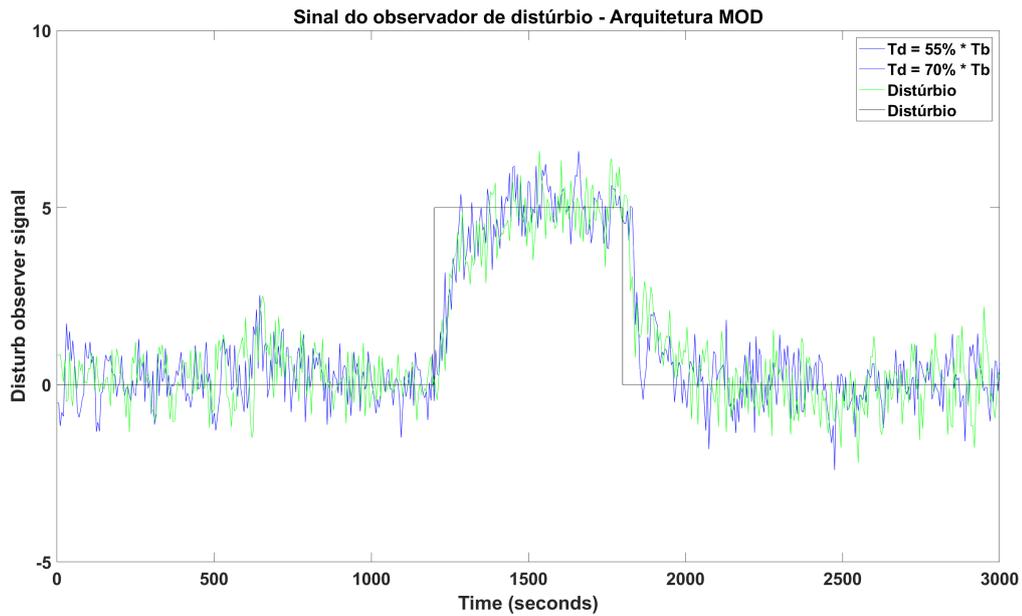


Figura 54 – Sistema MOD: Sinais da saída do Observador.

Verificados os gráficos das figuras acima e também os indicadores de desempenho da saída (IAE e ISE), foi selecionado o valor de  $T_d = 55\% \cdot T_b$  para a construção do observador de distúrbio para arquitetura MOD.

## 4.7 Análise dos dados - Arquitetura com filtro estatístico (MOF)

Seguindo os mesmos passos realizados para a arquitetura MOD, os dados obtidos dos testes com a arquitetura MOF foram analisados através do mesmo Código B.6. Assim como na etapa anterior, primeiro foi utilizado o valor  $filter = columns$ .

Com os valores de TV, IAE e ISE, calculados para o intervalo  $1100s > t > 2300s$ , a Tabela 6 foi preenchida e os gráficos nas Figuras 55, 56 e 57 abaixo foram gerados.

Porcentagem de $T_b$	IAE	ISE	TV
30	1212983.1562	3814574.9919	805444.2691
35	1235793.3083	3329532.6103	708481.2449
40	1095039.0759	2770682.1244	678412.3032
45	1016139.5702	2291633.5838	686362.2171
50	1047136.7912	2759928.2429	739438.3515
55	1032228.9791	2675874.314	660575.475
60	1066298.8873	2617862.7262	695131.0716
65	1091055.9139	2892751.1568	717157.0972
70	1123013.9008	3059605.9081	706912.6658
75	1108293.2739	3069642.6276	629475.6322
80	1165594.8658	3341605.8475	786518.0863
85	1165237.3867	2945895.0568	674666.3575
90	1142390.9283	3660384.573	648299.6736
95	1151326.7612	3224073.905	690337.1195
100	1172533.4644	3358754.0367	718456.4501
110	1169994.421	3345174.7418	652767.1861
120	1193536.2625	3431182.2127	618159.7297

Tabela 6 – Indicadores de performance para arquitetura com filtro estatístico e observador de distúrbio

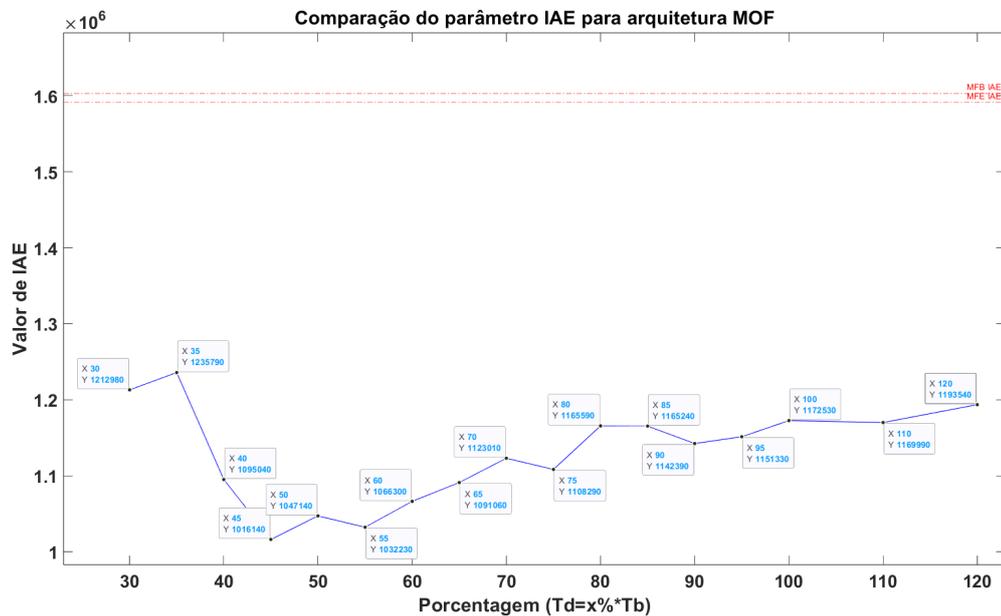


Figura 55 – Parâmetro IAE para cada teste na arquitetura MOF.

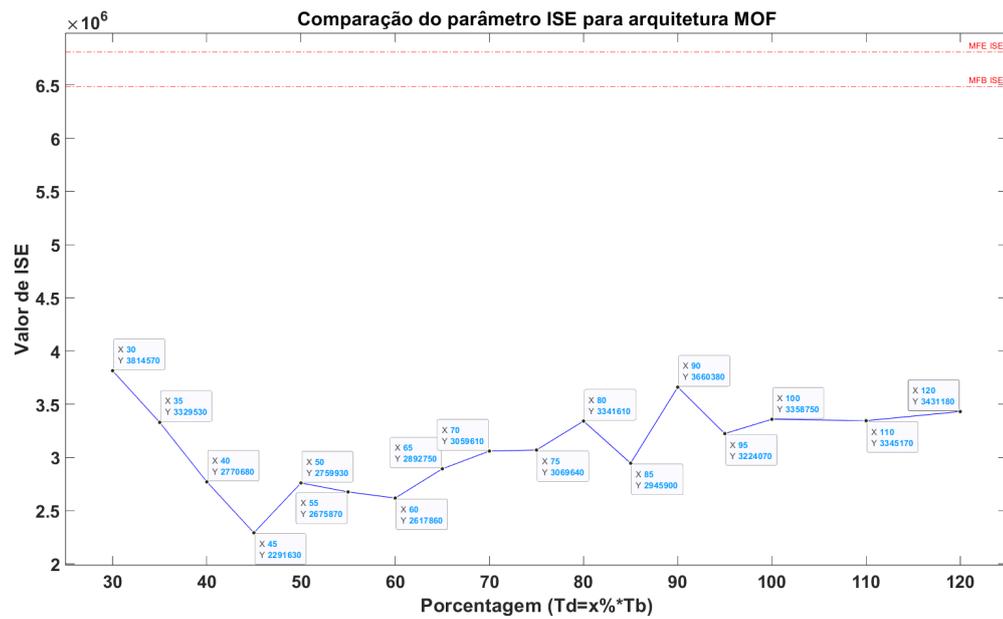


Figura 56 – Parâmetro ISE para cada teste na arquitetura MOF.

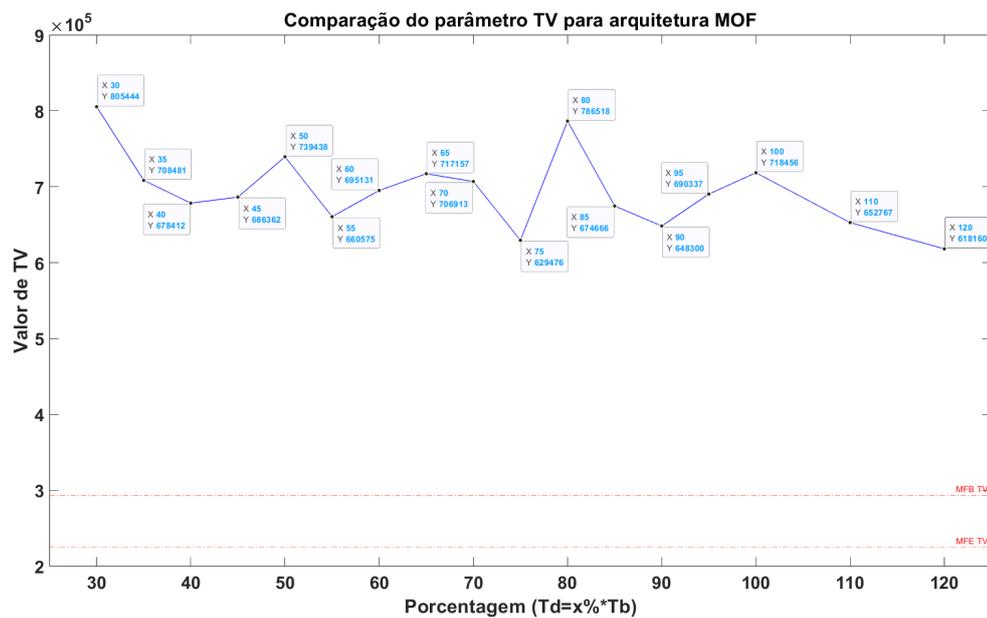


Figura 57 – Parâmetro TV para cada teste na arquitetura MOF.

Filtradas as seis melhores porcentagens analisando cada indicador, tem-se:

Seis melhores IAE (ordem crescente): 45%, 55%, 50%, 60%, 65%, 40%

Seis melhores ISE (ordem crescente): 45%, 60%, 55%, 50%, 40%, 65%

Seis melhores TV (ordem crescente): 120%, 75%, 90%, 110%, 55%, 85%

Após analisar o resultado dado pelo Código B.6, das Figuras 55, 56 e 57 e dos cinco melhores valores de cada indicador, foram selecionados como candidatos a melhor observador de distúrbio para a arquitetura MOF as arquiteturas com  $T_d = 45\% \cdot T_b$  (com o nível de água na saída e sinal de controle vistos nas Figuras 58 e 59 respectivamente) e  $T_d = 55\%$  (com o nível de água na saída e sinal de controle vistos nas Figuras 60 e 61).

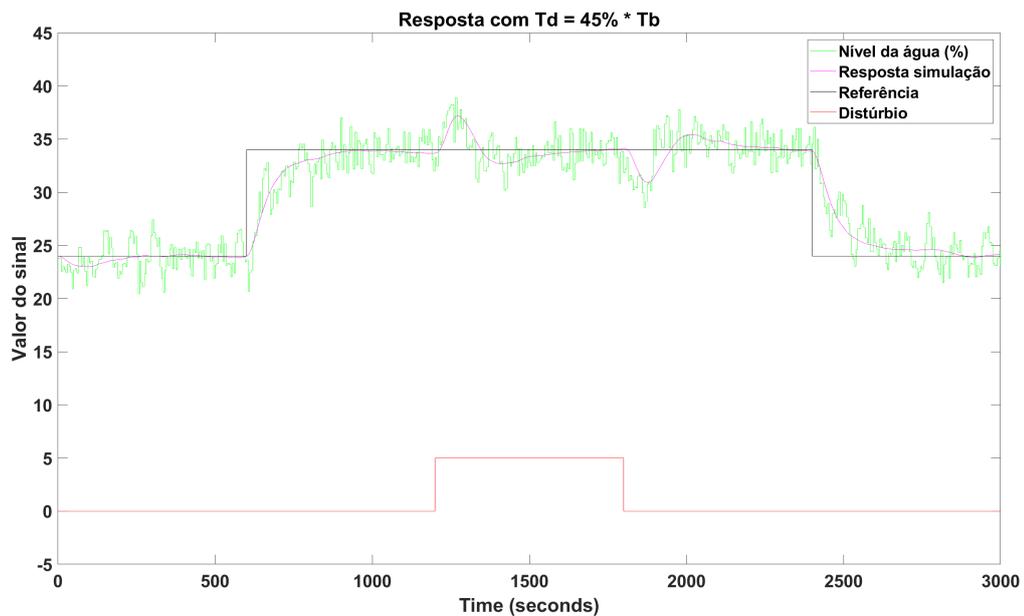


Figura 58 – Resposta do sistema para  $T_d = 45\% \cdot T_b$

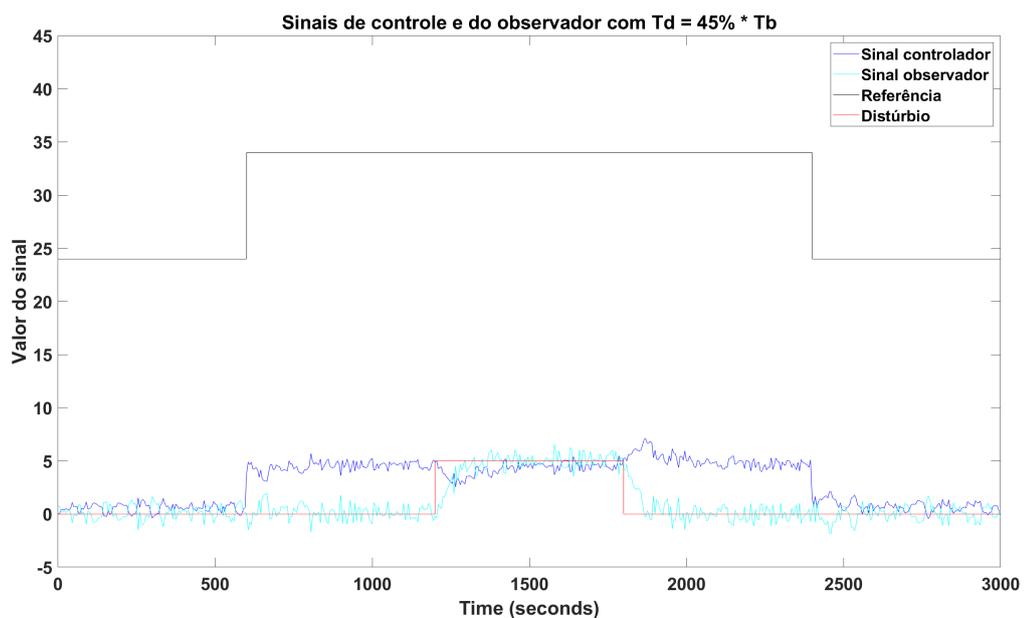


Figura 59 – Sinal de controle para  $T_d = 45\% \cdot T_b$

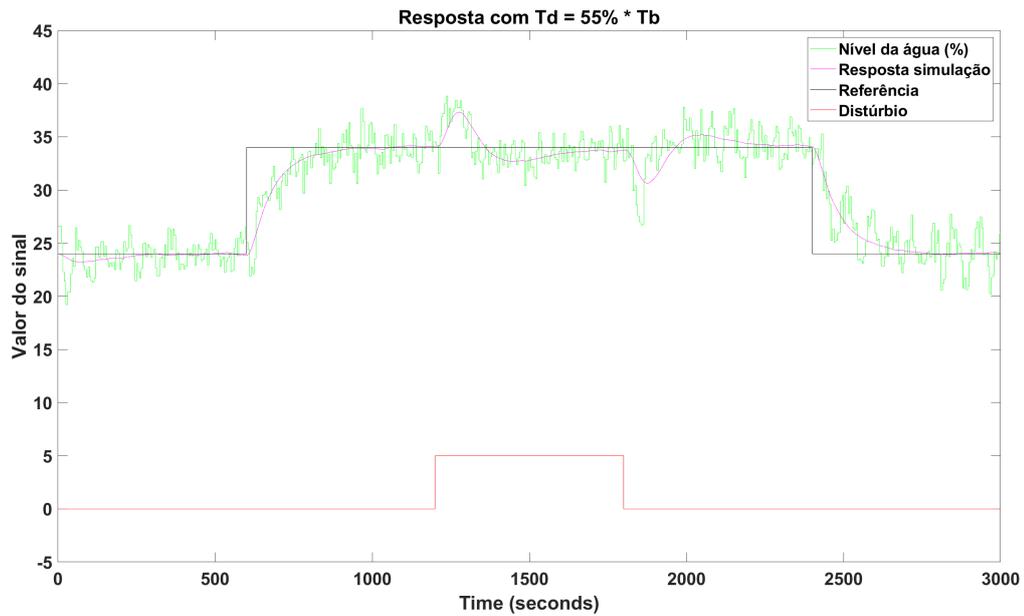


Figura 60 – Resposta do sistema para  $T_d = 55\% \cdot T_b$

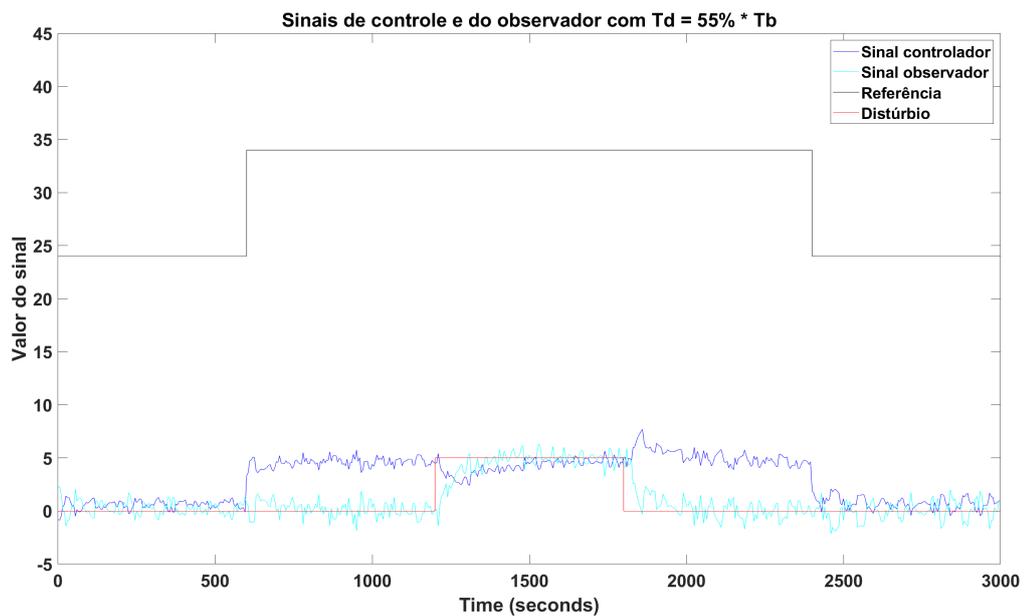


Figura 61 – Sinal de controle para  $T_d = 55\% \cdot T_b$

Feita a análise comparativa das melhores porcentagens para obter  $T_d$  (45% e 55%) com a arquitetura sem observador de distúrbio MFE, têm-se os gráficos abaixo, comparando o nível de água dado na saída do sistema (Figura 62), o sinal de controle (Figura 63) e comparando os dois sinais do observador (Figura 64).

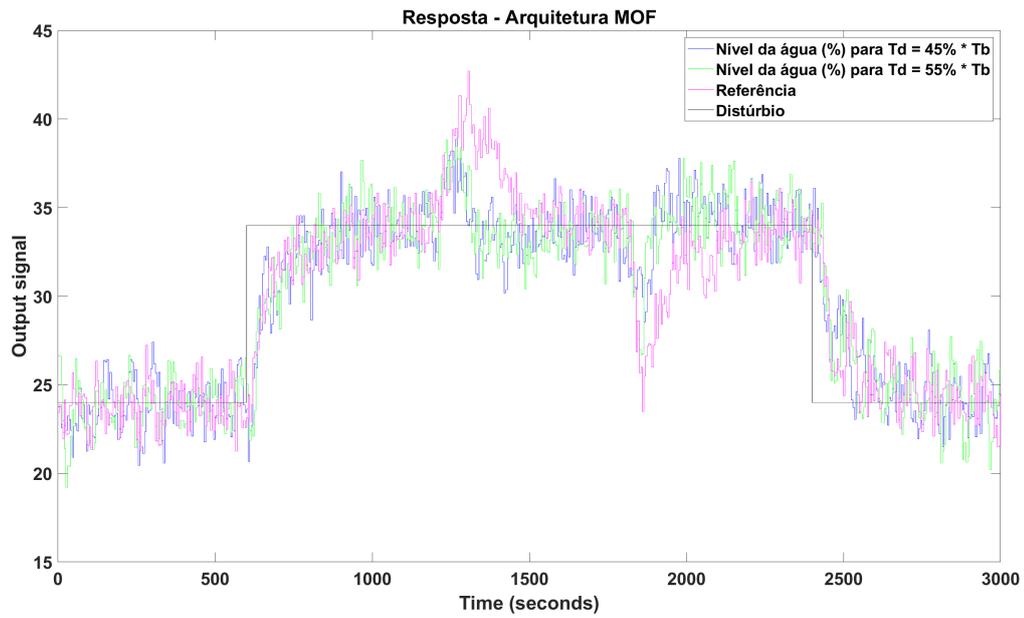


Figura 62 – Sistema MOF: Sinais do nível da água em porcentagem e saída do observador de distúrbio.

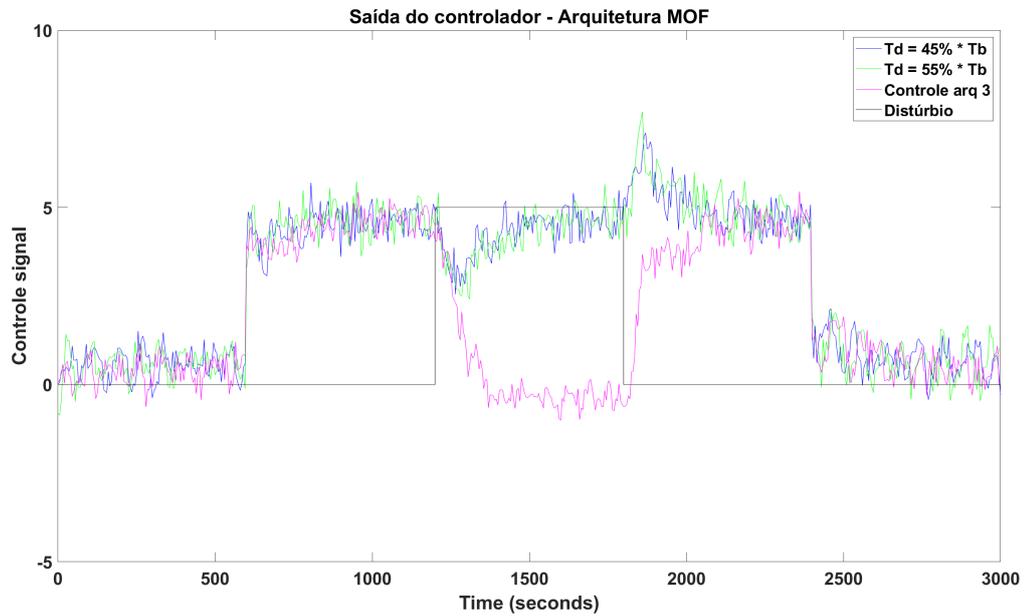


Figura 63 – Sistema MOF: Sinais do sistema.

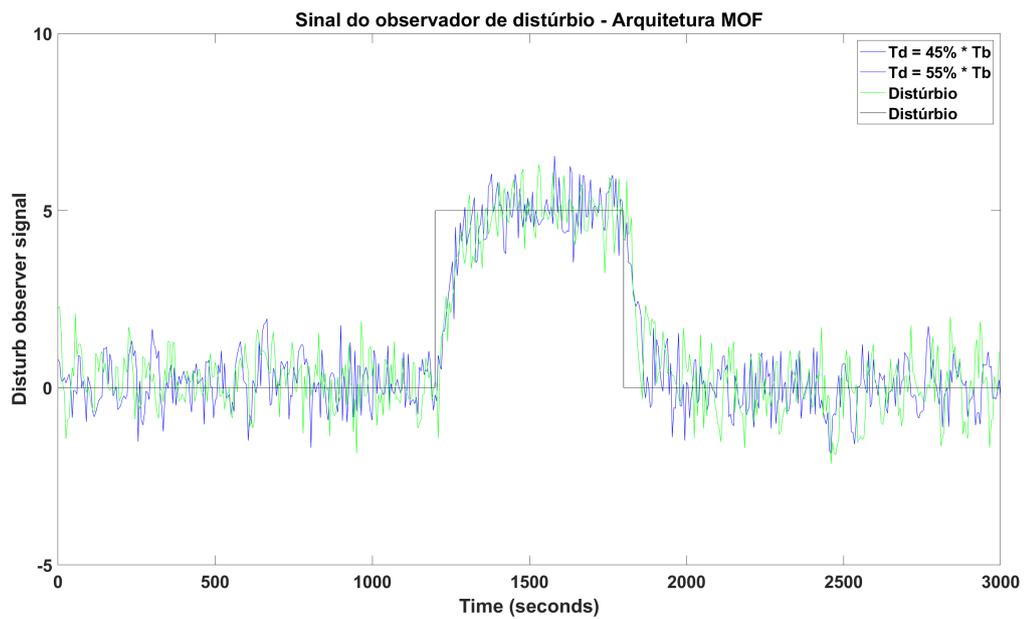


Figura 64 – Sistema MOF: Sinais da saída do Observador.

Após conferir os gráficos das figuras acima e também os indicadores de desempenho da saída (IAE e ISE), o valor de  $T_d = 45\% \cdot T_b$  foi selecionado como o melhor resultado para arquitetura MOD.

## 4.8 Comparação entre sistemas

Sobrepondo as Figuras dos indicadores de TV, IAE e ISE, é possível analisar visualmente qual arquitetura com o observador de distúrbio possui o melhor desempenho. Primeiro são observados os indicadores IAE (Figura 65) e ISE (Figura 66).

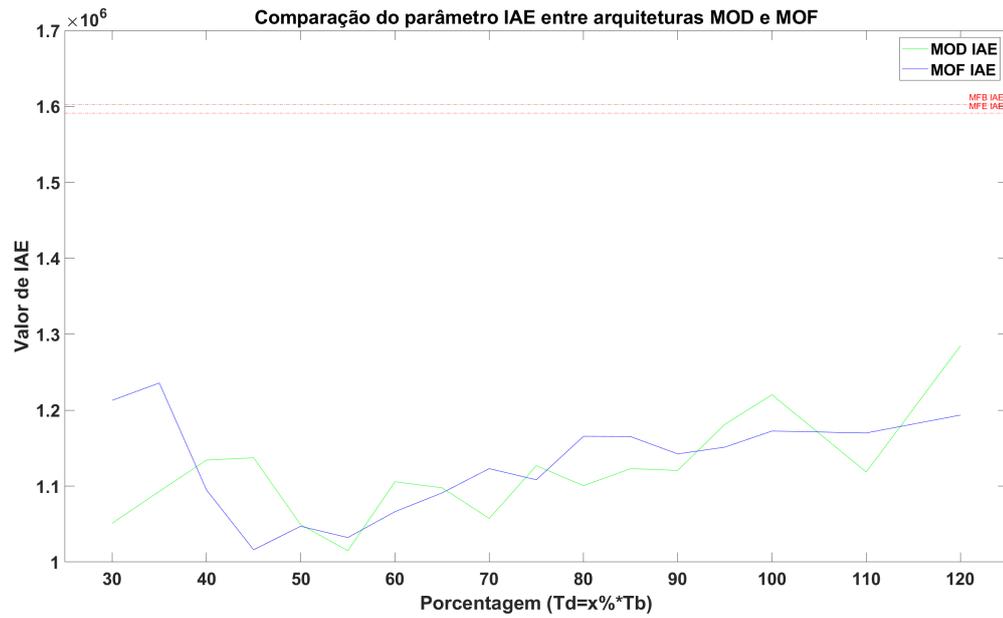


Figura 65 – Comparação do parâmetro IAE dos testes em MOD e MOF.

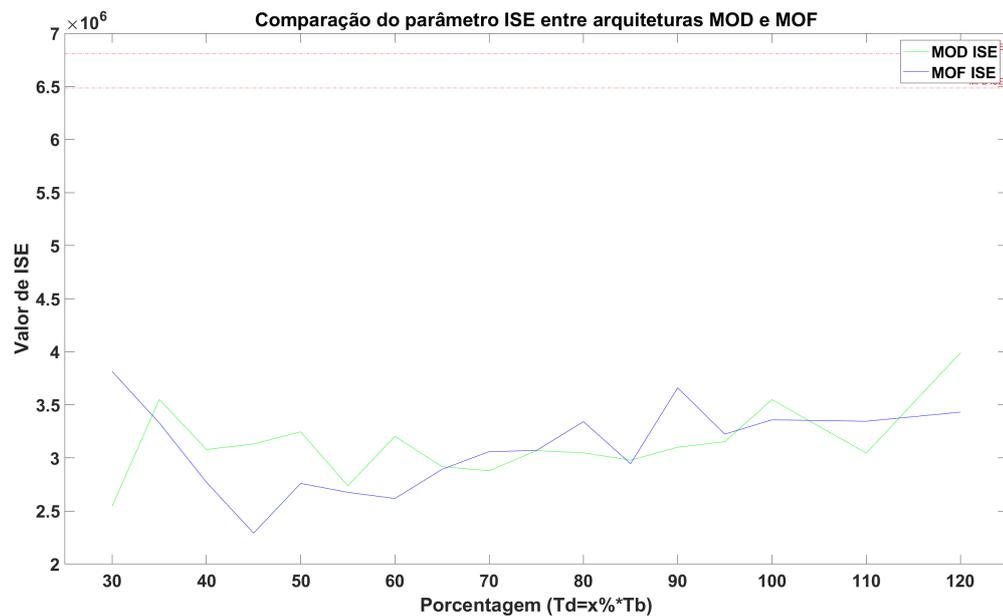


Figura 66 – Comparação do parâmetro ISE dos testes em MOD e MOF.

É notável após análise da performance da saída, que a combinação do observador com o filtro estatístico apresenta uma melhor resposta, indicando um menor erro, principalmente quando influenciado pelo distúrbio.

Observando agora o indicador TV, temos o seguinte gráfico na Figura 67.

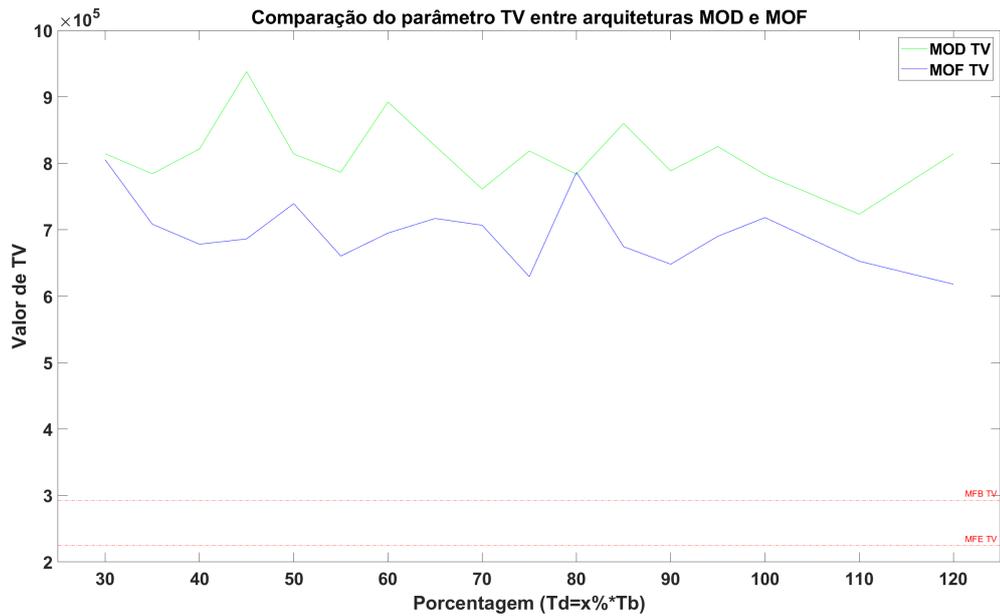


Figura 67 – Comparação do parâmetro TV dos testes em MOD e MOF.

As respostas das duas arquiteturas (MOD e MOF) para todos os observadores testados possuem um comportamento pior quando comparados com os valores de TV das arquiteturas MFB e MFE, por mais que seja notada uma melhora quando a malha fechada possui tanto o observador quanto o filtro estatístico.

Por fim, é apresentada na Figura 68 a comparação das respostas da malha fechada das melhores arquiteturas selecionadas com as melhores configurações do observador de distúrbio

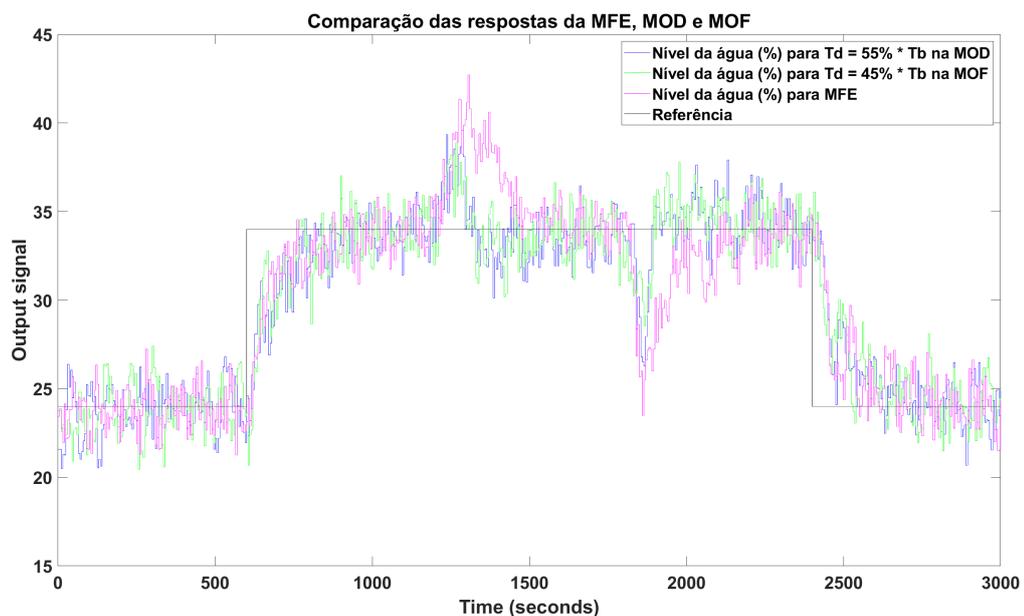


Figura 68 – Melhores sinais de saída encontrados

Comparando os resultados obtidos na simulação e no ensaio experimental do sistema de tanques, é percebida uma similaridade no comportamento de cada arquitetura analisada. Assim como na conclusão da simulação, a arquitetura 5 se destaca como uma boa candidata a melhor modelo, necessitando de alguns ajustes no filtro estatístico para diminuir o ruído da saída. Analisando a Tabela 7 abaixo, a arquitetura 5 (MOF) possui um valor de IAE um pouco maior que o para a arquitetura 3 (MOD), mas nos outros dois indicadores, MOF possui melhores valores com uma certa folga comparando com a MOD.

<b>Arquitetura</b>	<b>IAE (<math>10^5</math>)</b>	<b>ISE (<math>10^5</math>)</b>	<b>TV (<math>10^5</math>)</b>
Arq 1 (MFB)	16.029664421	64.835842034	2.930483393
Arq FE (MFE)	15.914412861	68.10660862	2.249243663
Arq 3 (MOD) com $T_d = 55\% \cdot T_b$	10.148853893	27.377917211	7.867561439
Arq 5 (MOF) com $T_d = 45\% \cdot T_b$	10.161395702	22.916335838	6.863622171

Tabela 7 – Comparação final dos indicadores

## 5 Conclusões

Este trabalho propõe malhas de controle com o objetivo de mitigar a influência que distúrbios causam na resposta do sistema. Para isso foram desenvolvidas arquiteturas com intuito de aumentar a eficiência do observador de distúrbio, com seus desempenhos analisados através de indicadores de performance que avaliam tanto os erros quanto a suavidade do sinal de controle e apresentam resultados específicos do sistema em questão.

É constatado que o observador consegue diminuir os efeitos sofridos pela planta quando é aplicado um distúrbio, conseguindo, de forma mais rápida que uma arquitetura sem o observador, levar a saída do sistema para a referência. É notável que a inserção do observador por si só na malha fechada não modifica a dinâmica original do sistema de forma drástica, mas dependendo de como for combinado com o filtro estatístico, como na arquitetura 7 da Figura 9, altera a forma que o sistema responde à mudança na referência. Quanto a sua desvantagem, é notado que o observador amplifica o ruído inerente ao sistema, sendo necessário acrescentar um filtro para tratar dessa perturbação.

Quanto ao filtro estatístico, é observado, principalmente quando comparados o indicador TV, que ele possui um papel importante no tratamento dos ruídos no sistema. Ele é mais eficiente quando filtra a saída  $y$  da malha, tanto antes desse sinal ser encaminhado para o observador de distúrbio, quanto antes do somatório que resultará no erro da malha, conforme a arquitetura 5 da Figura 7, mesma utilizada para a arquitetura MOF.

Analisando a combinação do filtro estatístico com o observador de distúrbio, ao observar os indicadores TV, não é percebida uma relação da influência do filtro estatístico com a constante de tempo  $T_d$  do filtro  $Q(s)$  do observador, mas ao observar os indicadores IAE e ISE, que verificam a performance quanto ao erro, em conjunto com TV e utilizando o método do Código B.6 para selecionar as melhores constantes de tempo, é observada uma redução nos valores selecionados quando comparadas as arquiteturas MOD (sem filtro estatístico) e MOF (com filtro estatístico). Isso fica mais evidente quando no código é utilizado o valor  $filter = 8$ , que seleciona como melhores constantes de tempo:

Para MOD:  $T_d = 55\% \cdot T_b$  ,  $T_d = 70\% \cdot T_b$  ,  $T_d = 80\% \cdot T_b$

Para MOF:  $T_d = 40\% \cdot T_b$  ,  $T_d = 45\% \cdot T_b$  ,  $T_d = 55\% \cdot T_b$

Para os ensaios feitos, sem realizar uma calibração detalhada nos filtros, principalmente no filtro estatístico, foi selecionada a arquitetura 5 com o valor de  $T_d = 45\% \cdot T_b$  para o filtro  $Q(s)$  como a com o melhor desempenho, por ser a que consegue lidar bem com o distúrbio introduzido no sistema e que tem um valor do indicador TV abaixo da média dos valores calculados para os outros  $Q(s)$  nessa arquitetura.

## 5.1 Perspectivas futuras

Para trabalhos futuros, é interessante um foco maior na calibração da constante de tempo do filtro  $Q(s)$  do observador e na busca de um procedimento sistemático de ajuste. Desta forma é esperada um melhor desempenho quanto a suavidade do sinal e, por consequência, no indicador TV.

Outra sugestão seria analisar a possibilidade da construção de um observador de distúrbio adaptativo, ou seja, que utilizasse os dados em tempo real para a adaptação do modelo interno, o que diminui sua dependência do modelo nominal  $G_n(s)$  da planta.

Por fim, outra possibilidade é o projeto do observador no tempo discreto por meio de um problema de otimização.

## Referências

- AN, H.; LIU, J.; WANG, C.; WU, L. Disturbance observer-based antiwindup control for air-breathing hypersonic vehicles. **IEEE Transactions on Industrial Electronics**, IEEE, v. 63, n. 5, p. 3038–3049, 2016. Citado na p. 18.
- CHEN, W.-H. Disturbance observer based control for nonlinear systems. **IEEE/ASME transactions on mechatronics**, IEEE, v. 9, n. 4, p. 706–710, 2004. Citado na p. 18.
- CHEN, W.-H.; YANG, J.; GUO, L.; LI, S. Disturbance-observer-based control and related methods—An overview. **IEEE Transactions on industrial electronics**, IEEE, v. 63, n. 2, p. 1083–1095, 2015. Citado na p. 18.
- FORTALEZA, E. L.; GOMES, L. M.; DE A. LIMAVERDE FILHO, J. O.; DE CAMPOS, M. C.; LONGHI, L. G. S.; DE LIMA, M. L.; TOGNETTI, E. S. Model-free adaptive filter to mitigate actuator wear. **ISA Transactions**, 2022. ISSN 0019-0578. DOI: <https://doi.org/10.1016/j.isatra.2022.02.026>. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0019057822000829>>. Citado nas pp. 22, 33, 36.
- GUO, L.; CHEN, W.-H. Disturbance attenuation and rejection for systems with nonlinearity via DOBC approach. **International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal**, Wiley Online Library, v. 15, n. 3, p. 109–125, 2005. Citado na p. 18.
- JOHANSSON, K. H. The quadruple-tank process: A multivariable laboratory process with an adjustable zero. **IEEE Transactions on control systems technology**, IEEE, v. 8, n. 3, p. 456–465, 2000. Citado na p. 29.
- LI, S.; YANG, J.; CHEN, W.-H.; CHEN, X. **Disturbance observer-based control: methods and applications**. CRC press, 2014. Citado na p. 21.
- MI, Y.; FU, Y.; LI, D.; WANG, C.; LOH, P. C.; WANG, P. The sliding mode load frequency control for hybrid power system based on disturbance observer. **International Journal of Electrical Power Energy Systems**, v. 74, p. 446–452, 2016. ISSN 0142-0615. DOI: <https://doi.org/10.1016/j.ijepes.2015.07.014>. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0142061515003002>>. Citado na p. 18.
- OHISHI, K. Torque-Speed Regulation of DC Motor Based on Load Torque Estimation. **IEEJ International Power Electronics Conference, IPEC-TOKYO, 1983-3**, v. 2, p. 1209–1216, 1983. Disponível em: <<https://cir.nii.ac.jp/crid/1570291225534338944>>. Citado na p. 18.

- SHAMSUZZOHA, M.; LEE, M. IMC- PID controller design for improved disturbance rejection of time-delayed processes. **Industrial & Engineering Chemistry Research**, ACS Publications, v. 46, n. 7, p. 2077–2091, 2007. Citado na p. 23.
- WANG, L.; SU, J. Disturbance rejection control for non-minimum phase systems with optimal disturbance observer. **ISA Transactions**, v. 57, p. 1–9, 2015. ISSN 0019-0578. DOI: <https://doi.org/10.1016/j.isatra.2014.08.003>. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0019057814002006>>. Citado na p. 18.

# Apêndices

# APÊNDICE A – Ensaio

## A.1 Sistema de primeira ordem

### A.1.1 Calibração do padrão de ruído

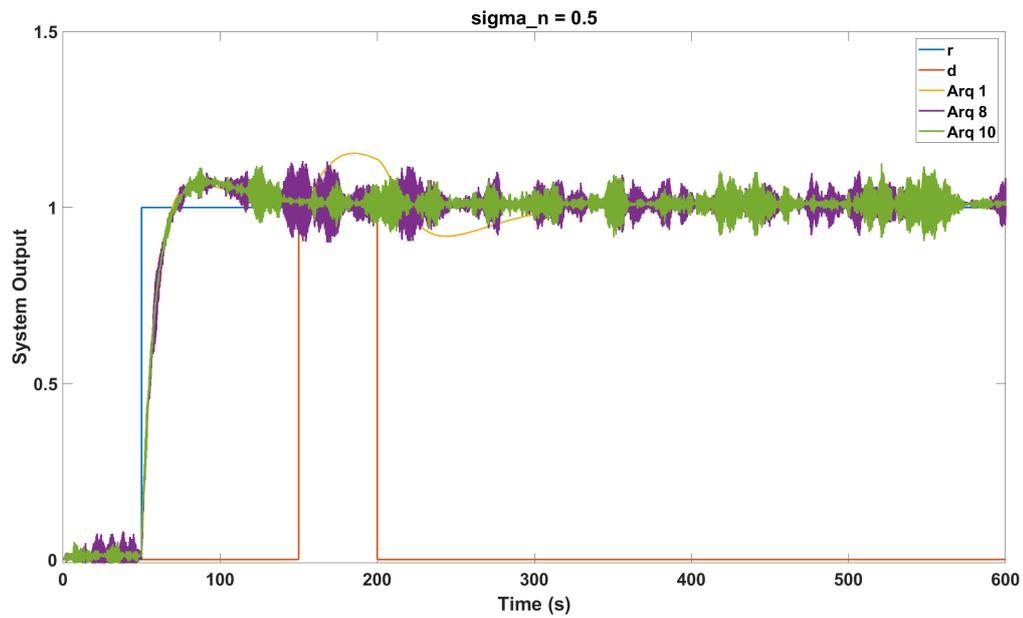


Figura A.1 – Saída y das arquiteturas 8 e 10 com  $\sigma_n = 0.5$

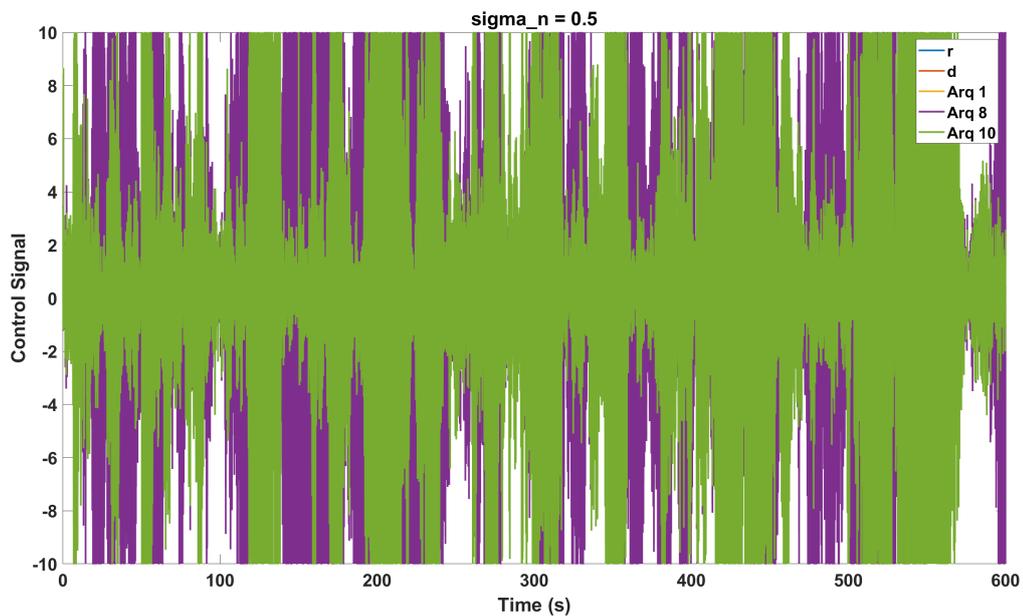
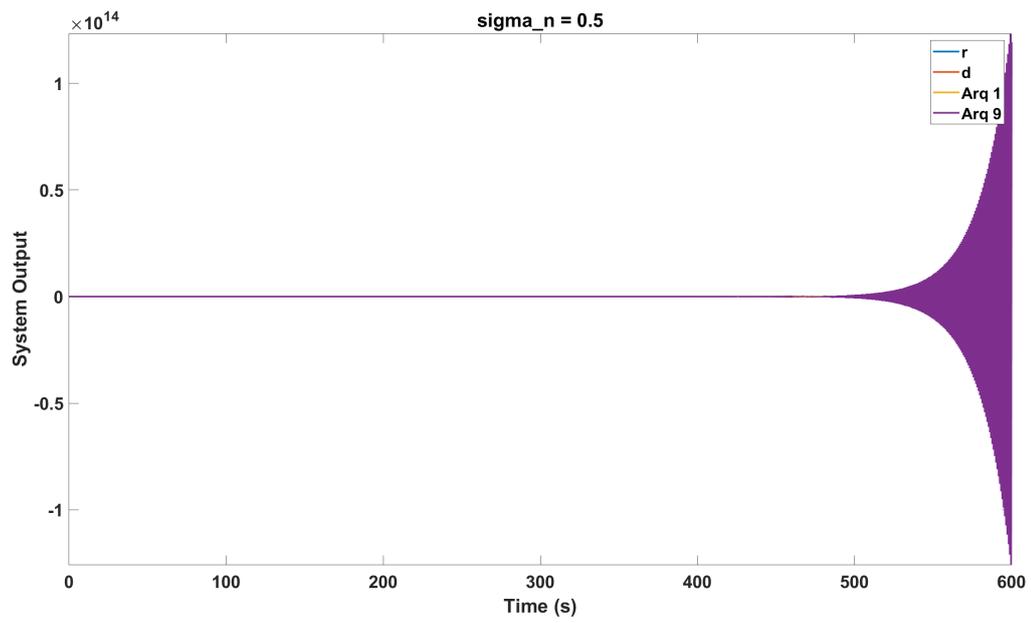
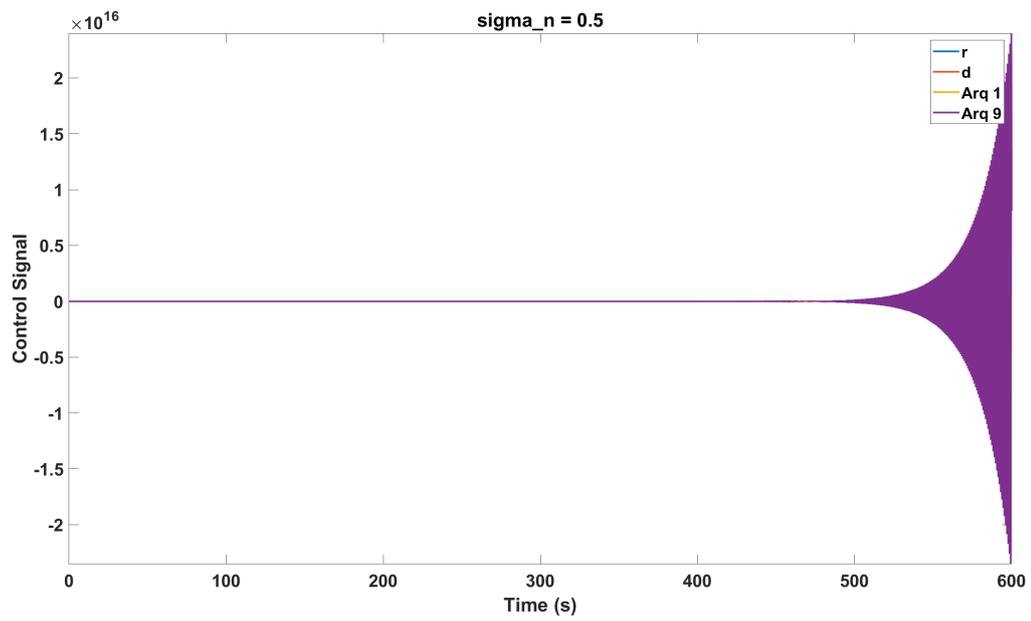


Figura A.2 – Controlador das arquiteturas 8 e 10 com  $\sigma_n = 0.5$

Figura A.3 – Saída y da arquitetura 9 com  $\sigma_n = 0.5$ Figura A.4 – Controlador da arquitetura 9 com  $\sigma_n = 0.5$

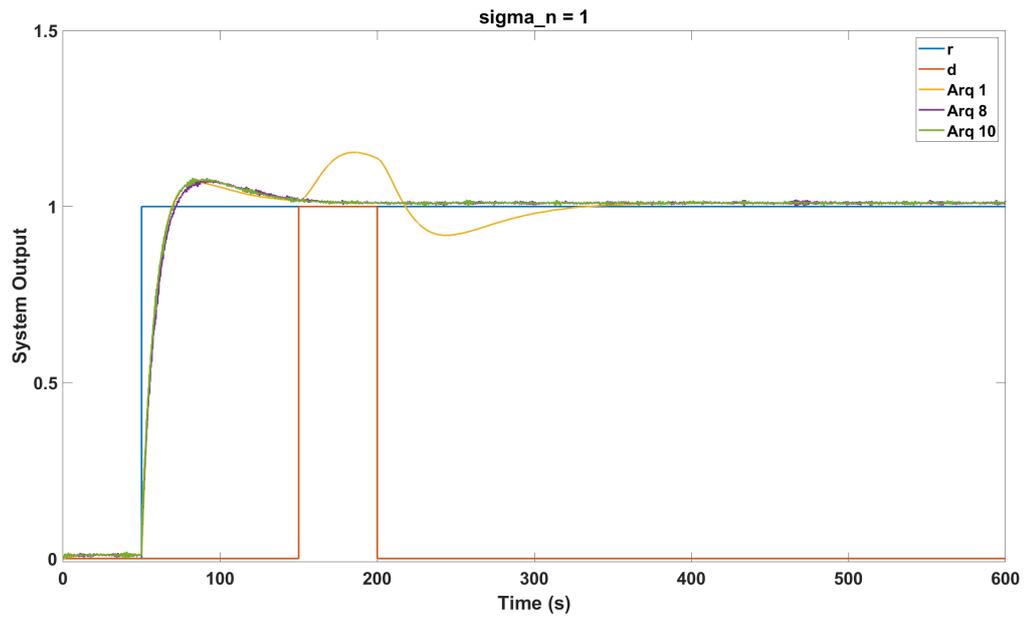


Figura A.5 – Saída  $y$  das arquiteturas 8 e 10 com  $\sigma_n = 1$

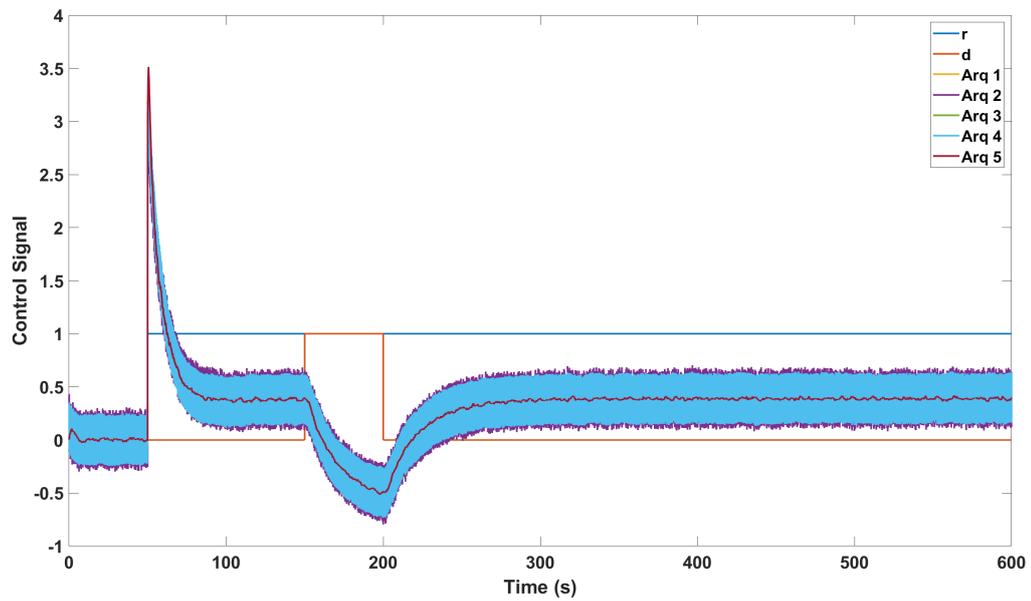


Figura A.6 – Sinal de controle  $u$  das arquiteturas 1, 2, 3, 4 e 5

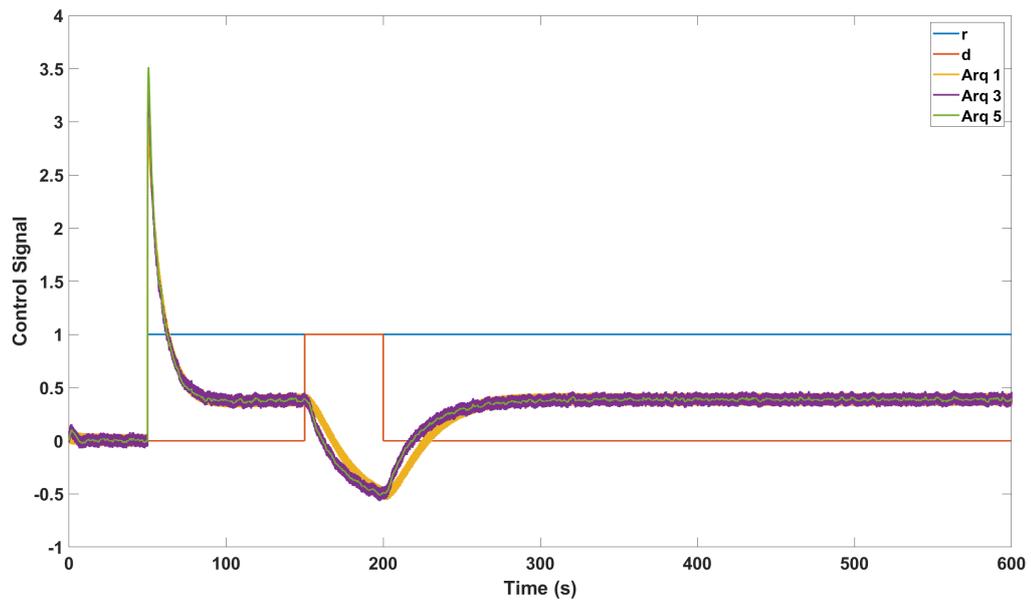


Figura A.7 – Sinal de controle  $u$  das arquiteturas 1, 3 e 5

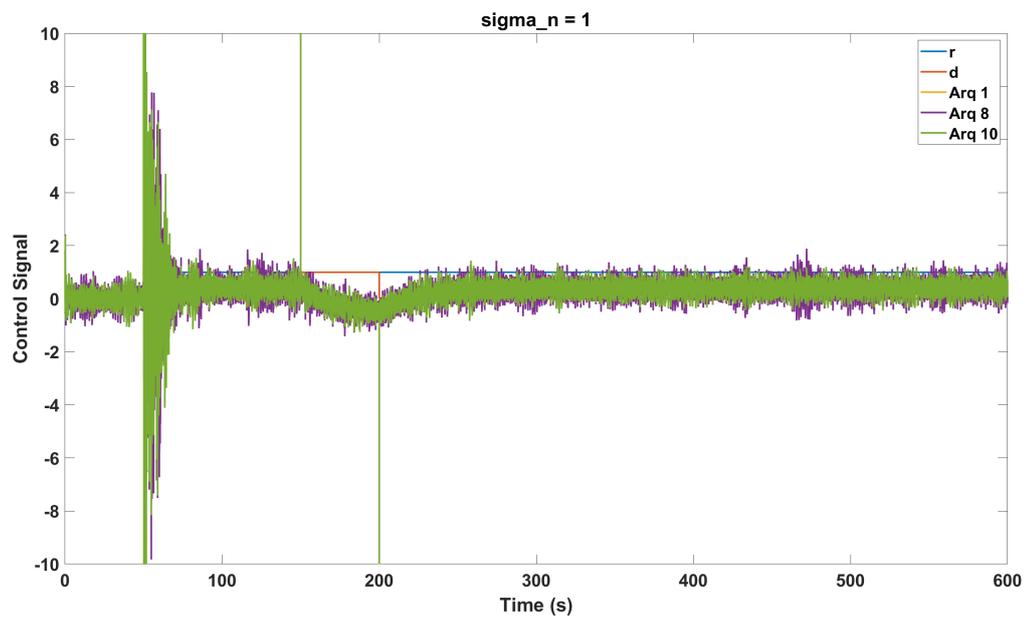


Figura A.8 – Sinal de controle das arquiteturas 8 e 10 com  $\sigma_n = 1$

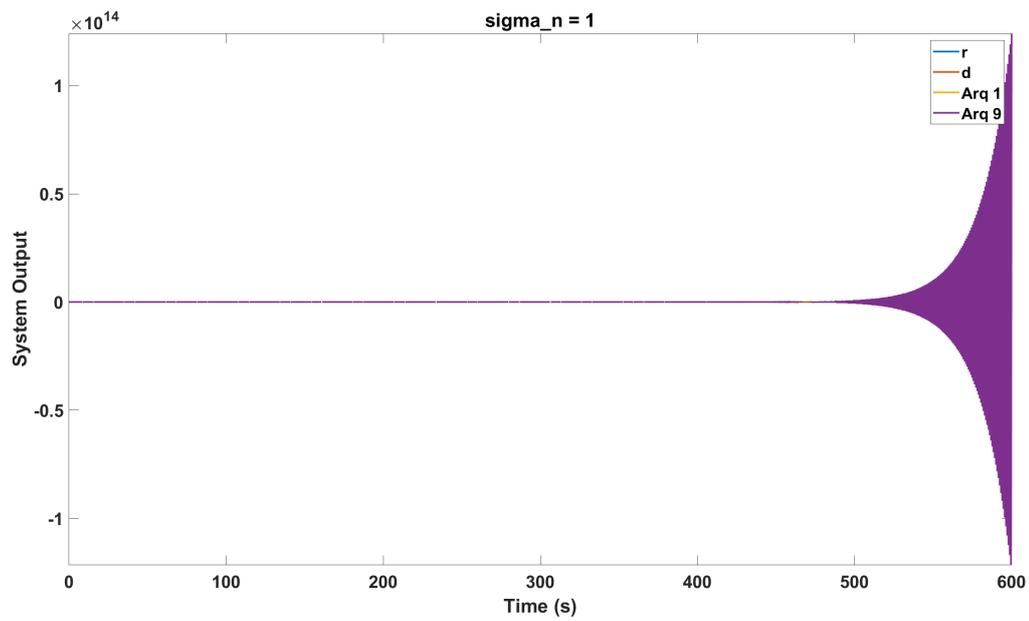


Figura A.9 – Saída y da arquitetura 9 com  $\sigma_n = 1$

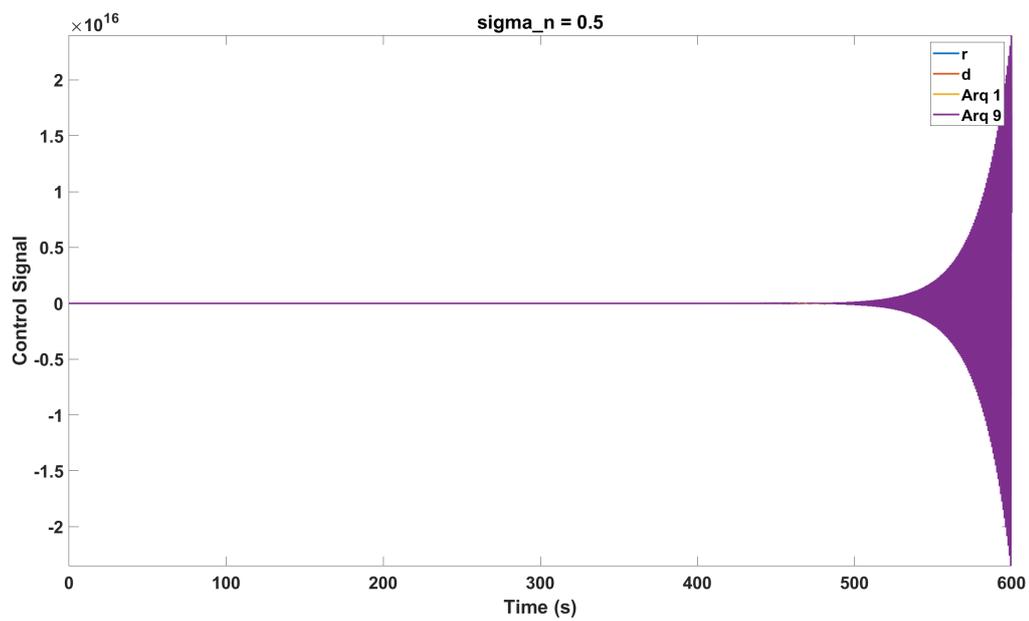


Figura A.10 – Controlador da arquitetura 9 com  $\sigma_n = 0.5$

## A.1.2 Respostas do sistema

- Saída y do sistema

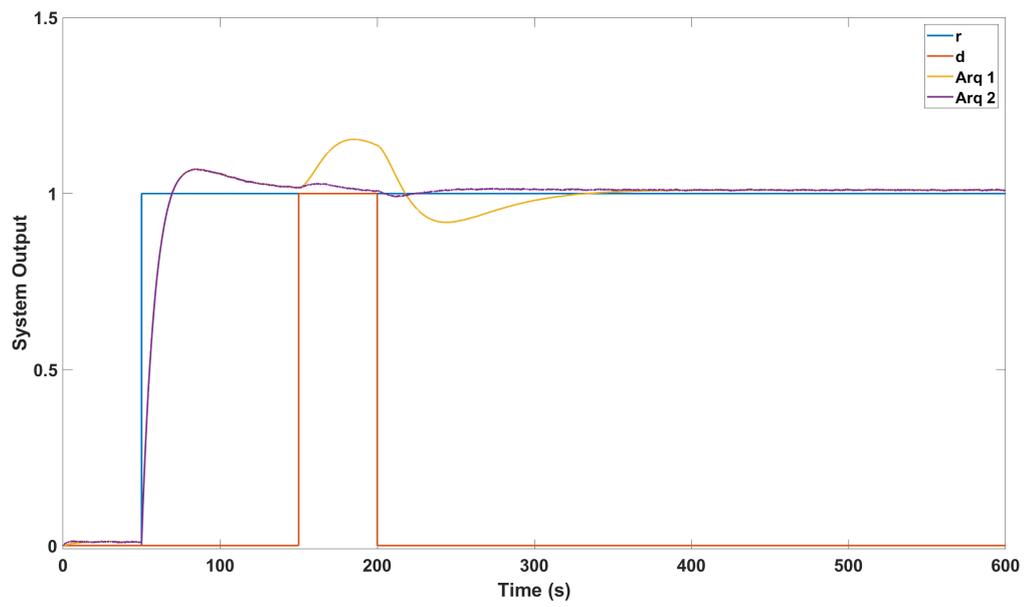


Figura A.11 – Ensaio I: Saída y da arquitetura 2

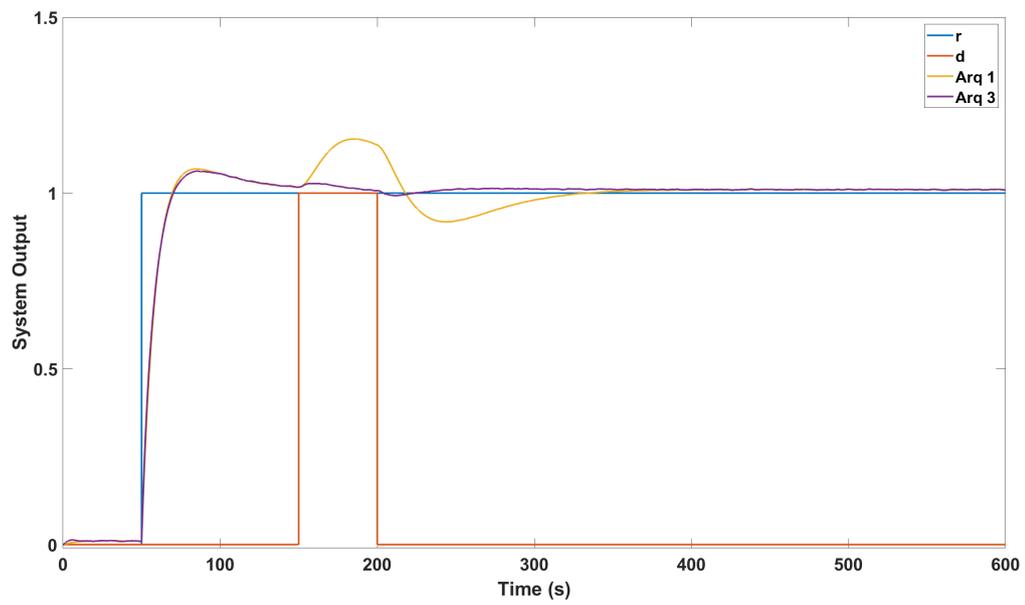


Figura A.12 – Ensaio I: Saída y da arquitetura 3

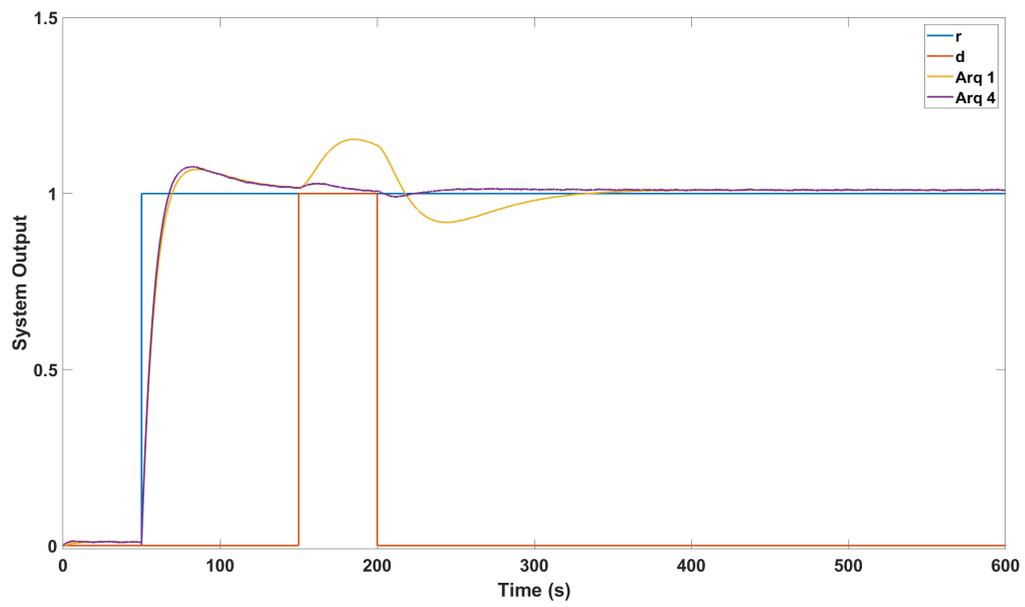


Figura A.13 – Ensaio I: Saída y da arquitetura 4

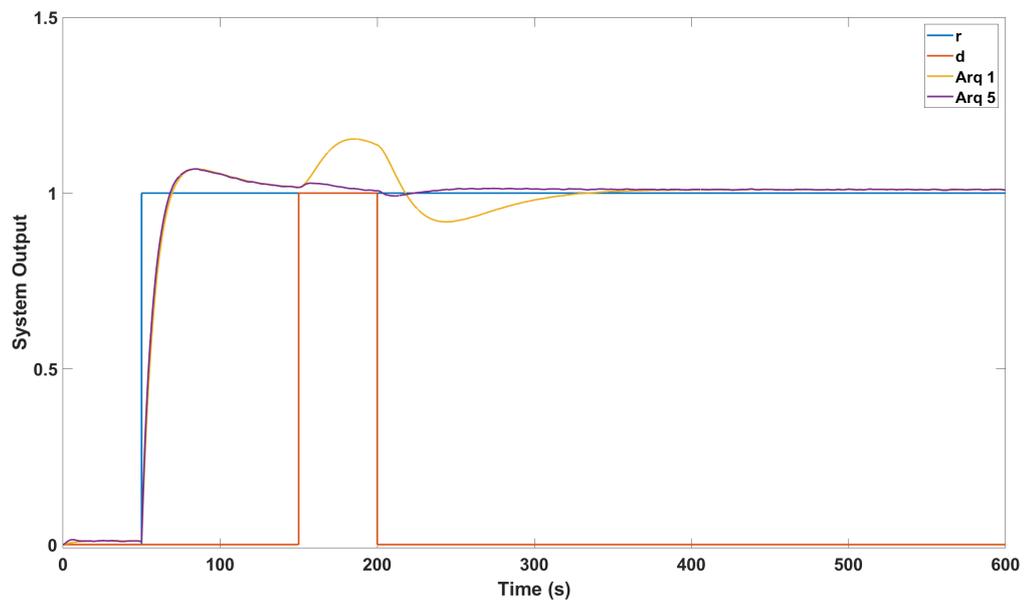


Figura A.14 – Ensaio I: Saída y da arquitetura 5

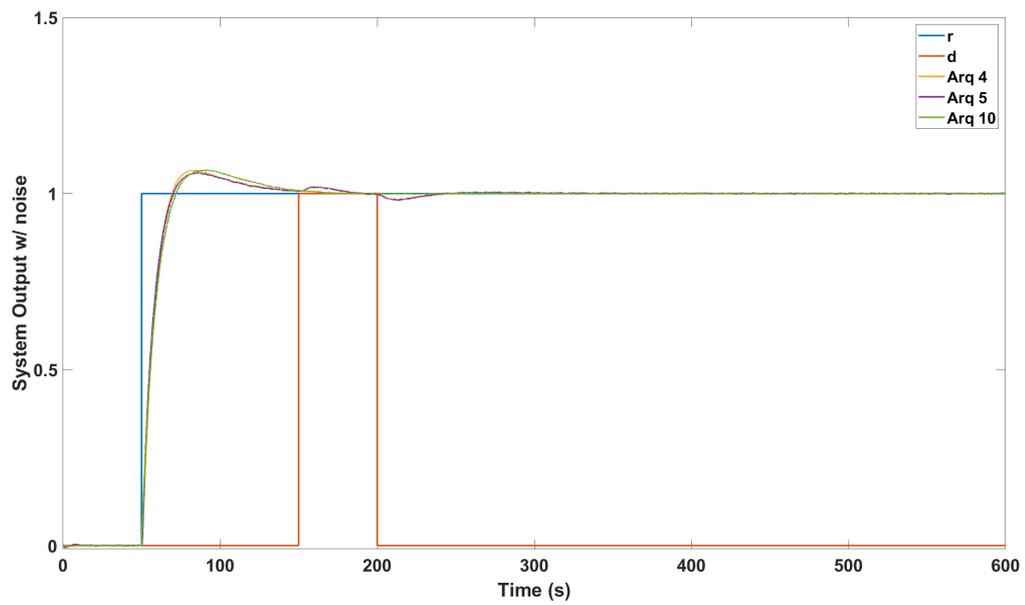


Figura A.15 – Ensaio I: Saída  $y_n$  das arquiteturas 4, 5 e 10

- Saída de controle  $u$

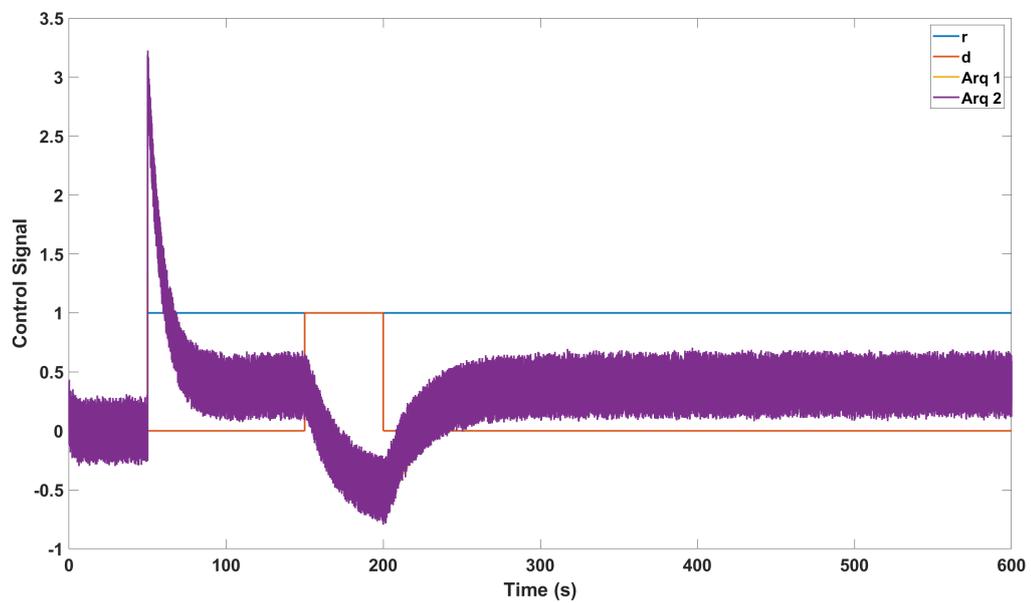
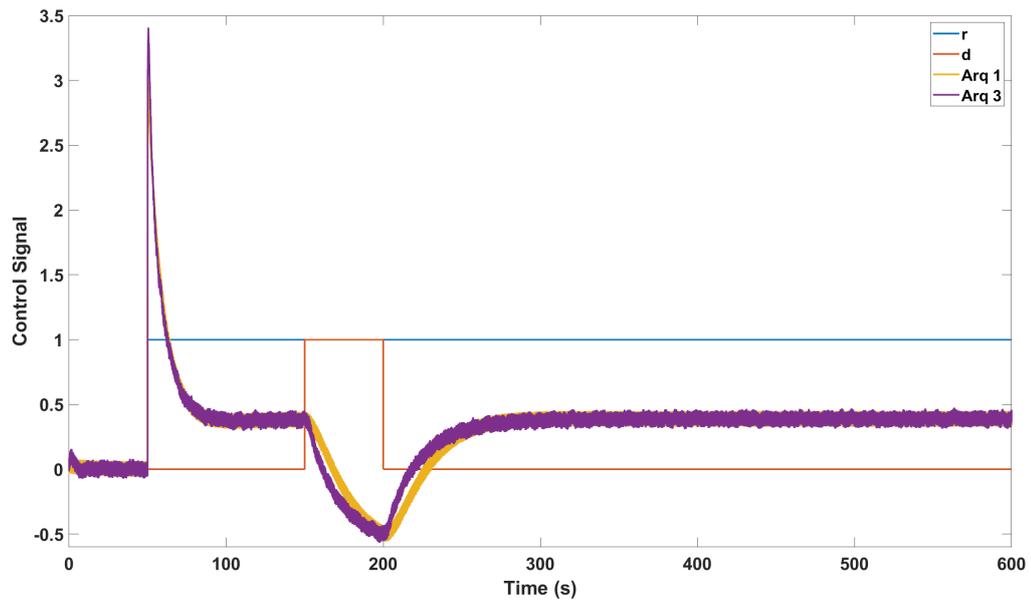
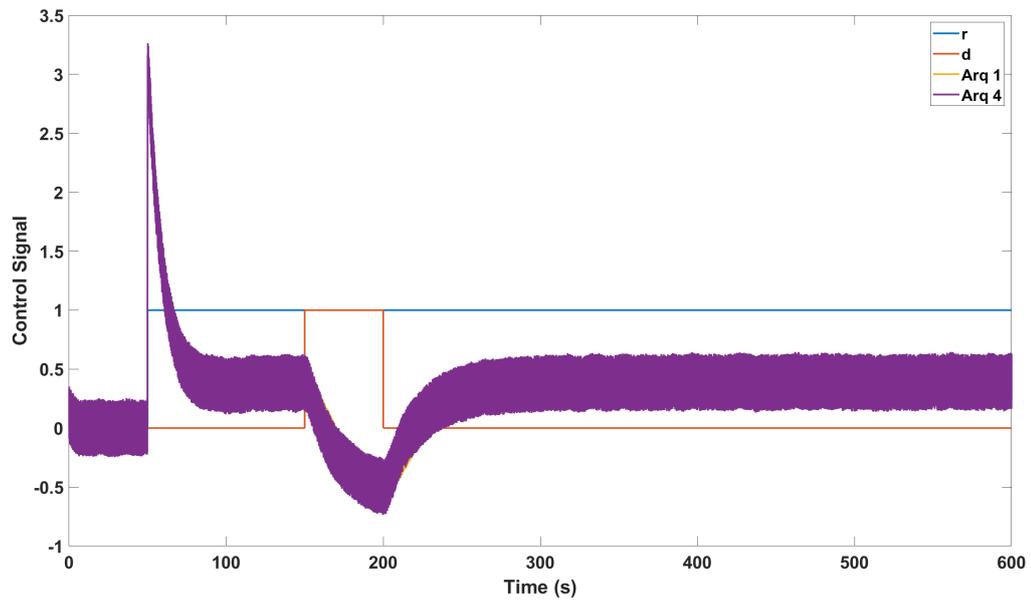
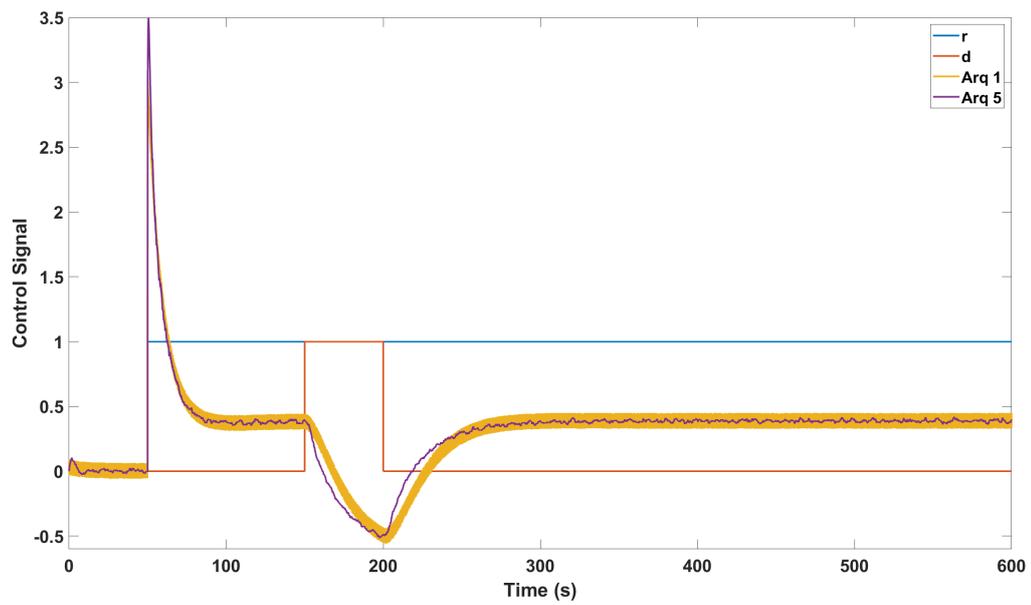
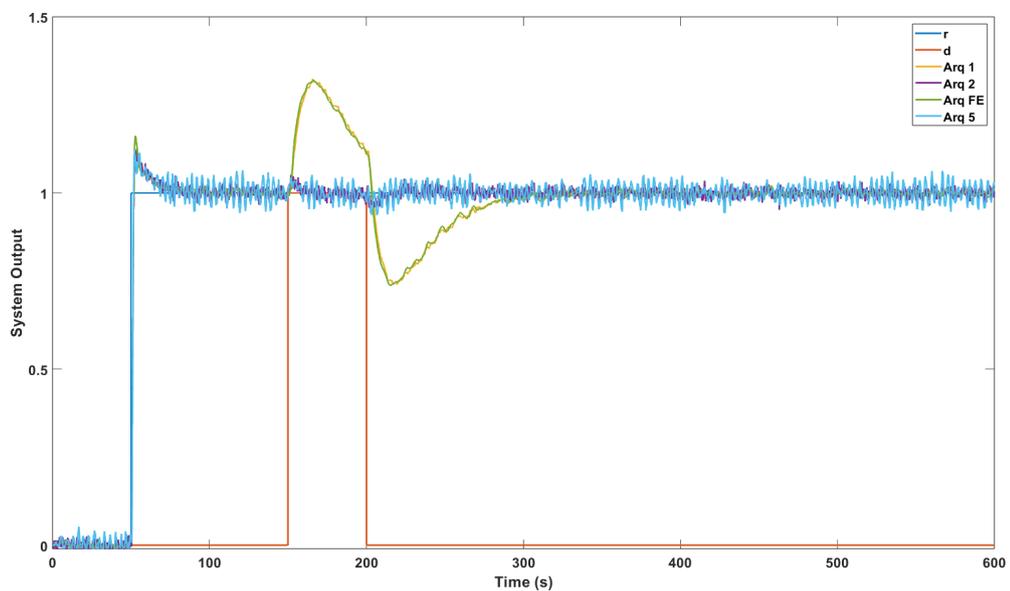


Figura A.16 – Ensaio I: Sinal  $u$  do controlador 2

Figura A.17 – Ensaio I: Sinal  $u$  do controlador 3Figura A.18 – Ensaio I: Sinal  $u$  do controlador 4

Figura A.19 – Ensaio I: Sinal  $u$  do controlador 5

## A.2 Função de transferência com atraso

Figura A.20 – Ensaio II: Saídas  $y$  do sistema sem ruído, com  $\sigma_n = 3.765$  para arquitetura 5

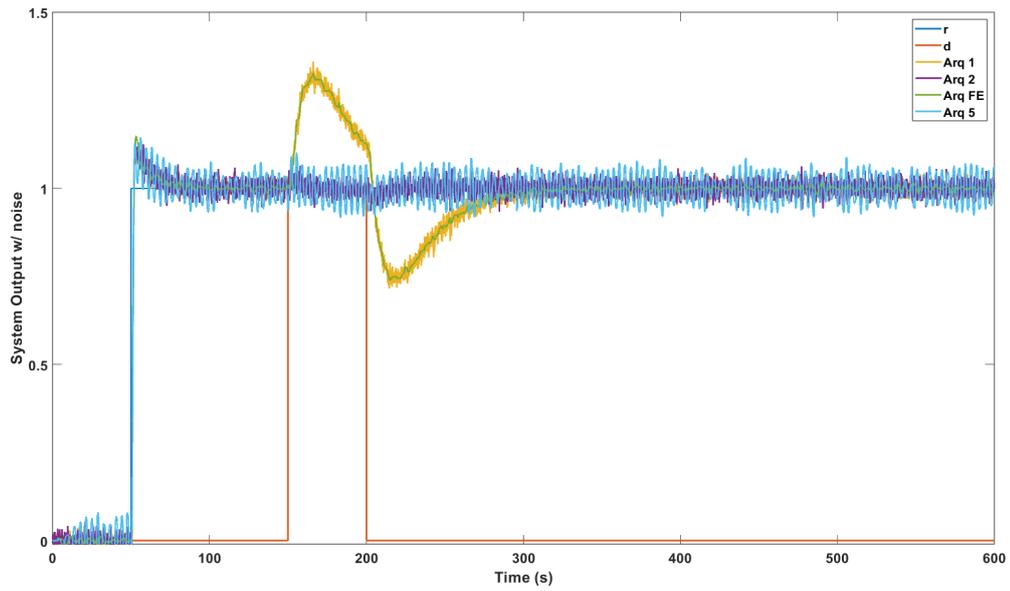


Figura A.21 – Ensaio II: Saídas  $y_n$  do sistema com ruído, com  $\sigma_n = 7.53$  para arquitetura 5

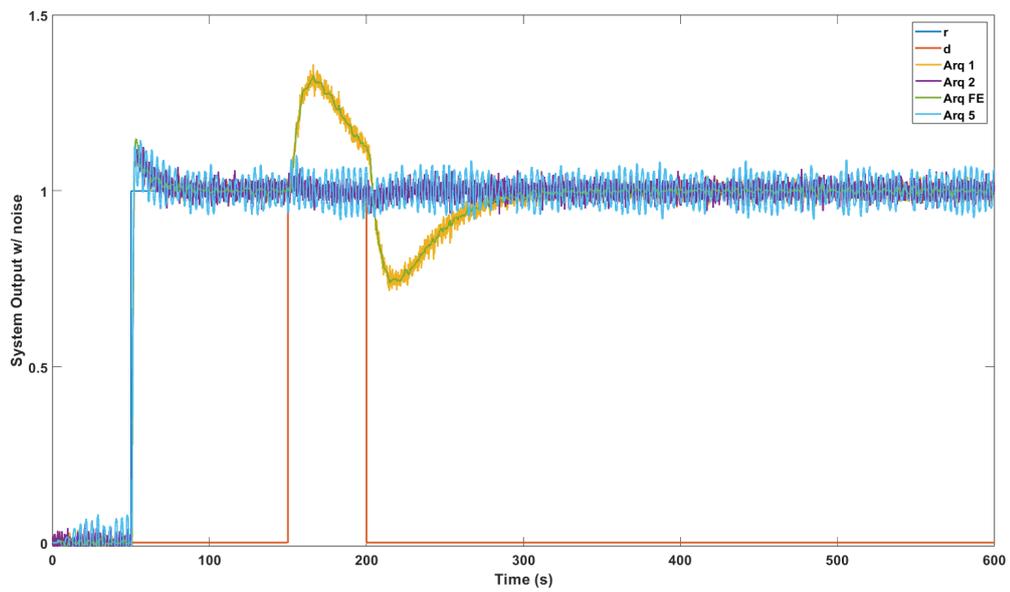
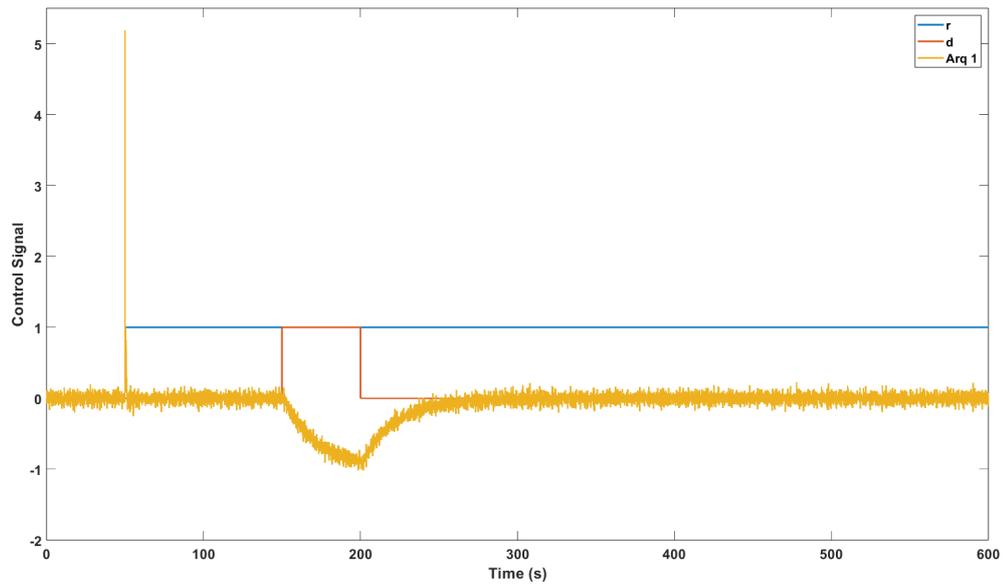
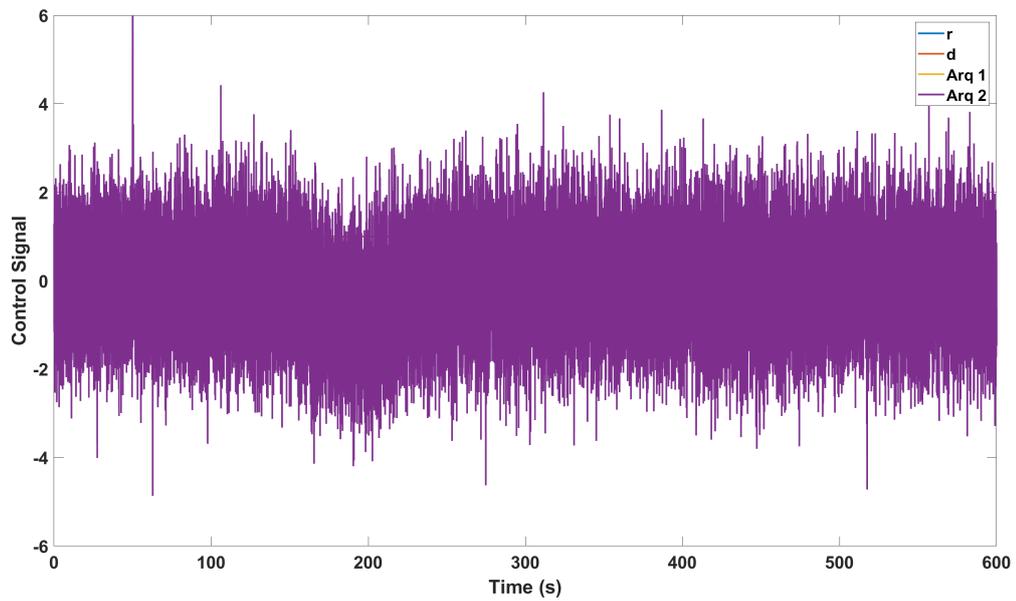


Figura A.22 – Ensaio II: Saídas  $y_n$  do sistema com ruído, com  $\sigma_n = 3.765$  para arquitetura 5

Figura A.23 – Ensaio II: Sinal  $u$  da arquitetura 1Figura A.24 – Ensaio II: Sinal  $u$  para as arquiteturas 1 e 2

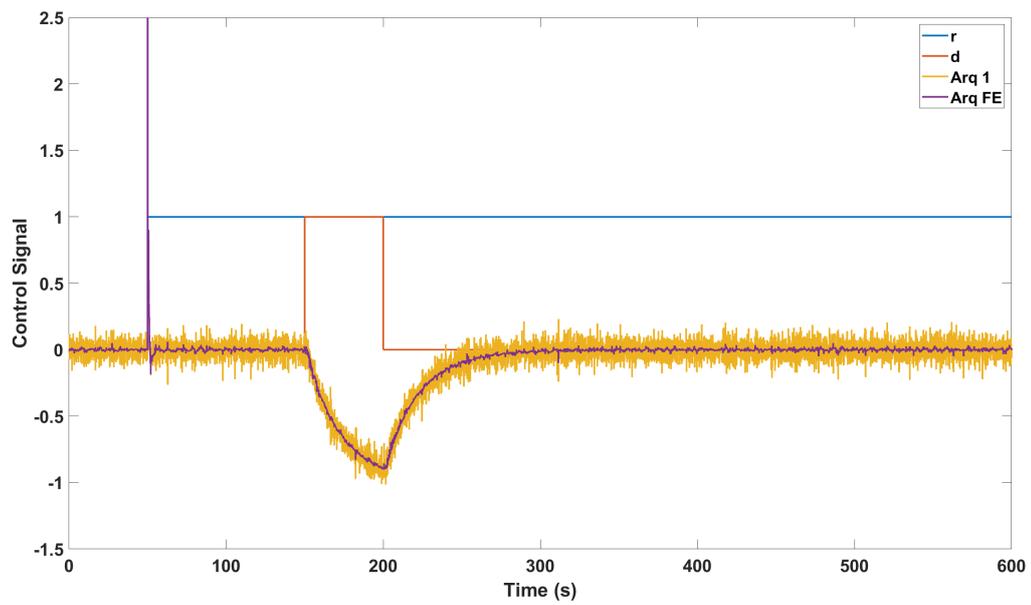


Figura A.25 – Ensaio II: Sinal  $u$  para as arquiteturas 1 e FE

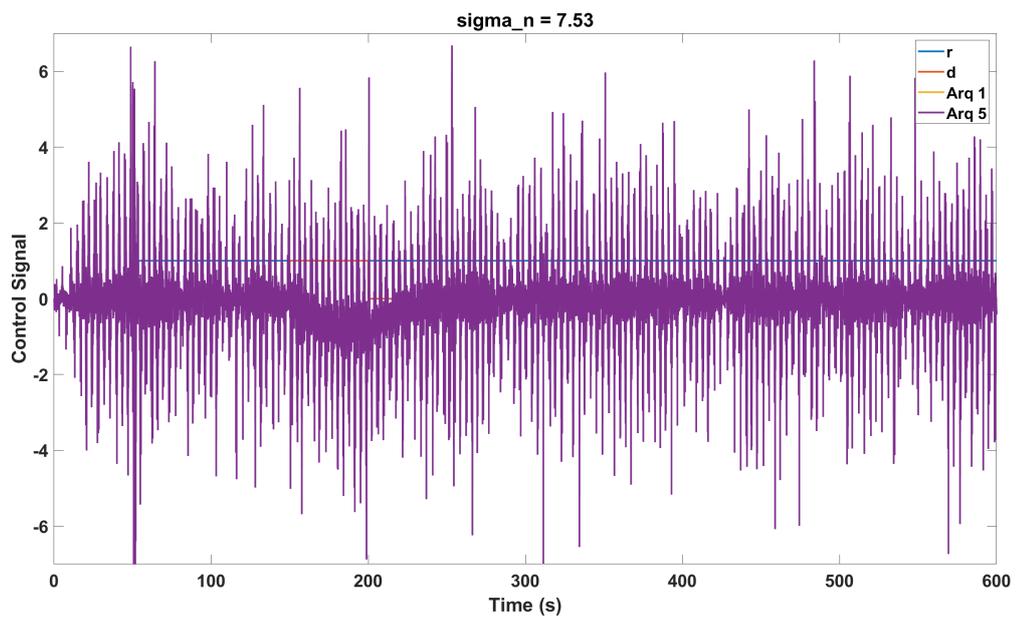


Figura A.26 – Ensaio II: Sinal  $u$  entre as arquiteturas 1 e 5 com  $\sigma_n = 7.53$

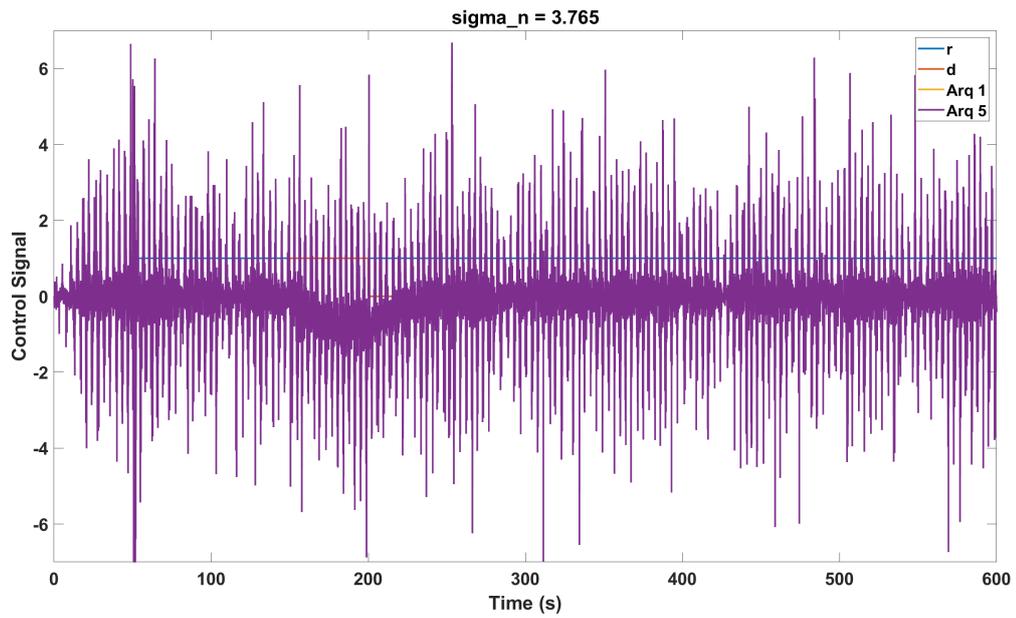


Figura A.27 – Ensaio II: Sinal  $u$  entre as arquiteturas 1 e 5 com  $\sigma_n = 3.765$

### A.3 Sistema tanques em cascata

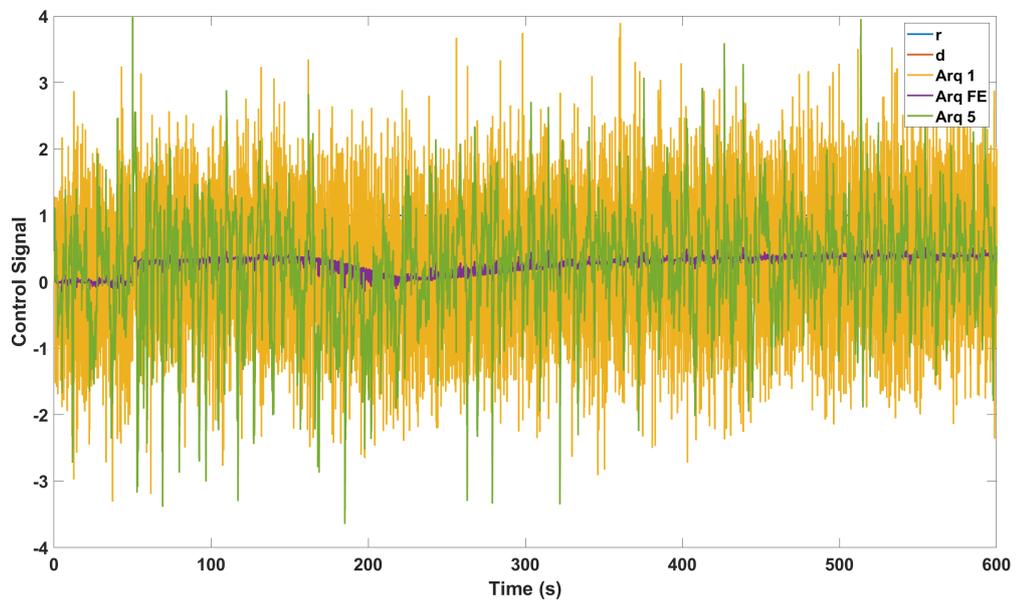


Figura A.28 – Ensaio III: Aproximação dos sinais  $u$  do sistema sem a arquitetura 2

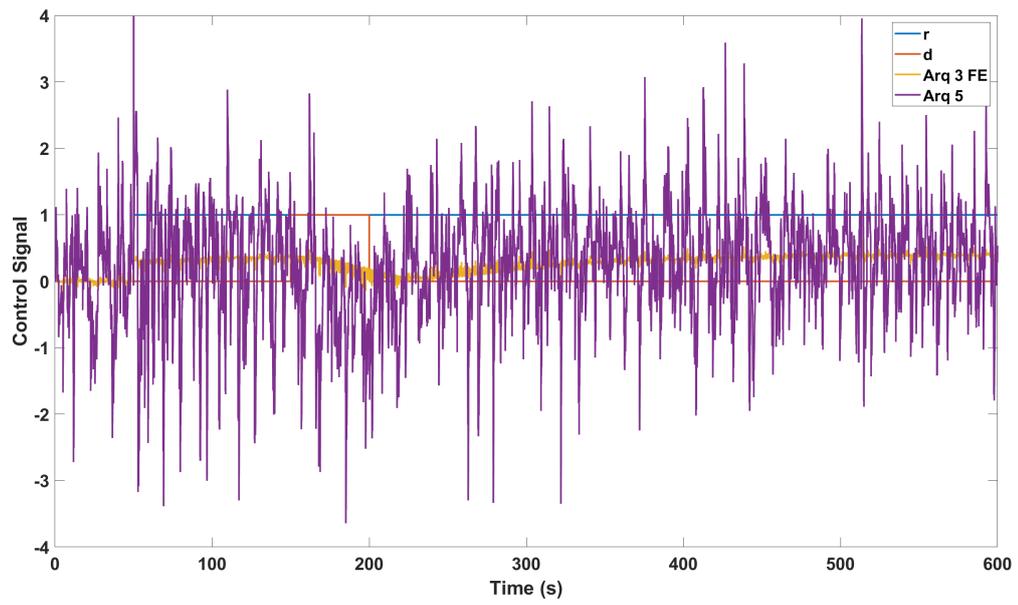


Figura A.29 – Ensaio III: Aproximação dos sinais  $u$  do sistema sem as arquiteturas 1 e 2

# APÊNDICE B – Códigos de programação

## B.1 Script que carrega as variáveis necessárias

Código B.1 – Código "load\_variables" para carregar valores utilizados pelo *Simulink*

```

1 % Variaveis para teste em bancada
2
3 clear
4
5 %% Parametros para simular ruido
6 var_ruido = 0.47;
7 erro_tipico_medicao = sqrt(var_ruido);
8
9 %% Parâmetro para o filtro estatístico
10
11 std_dev = 0.4703;
12
13 %% Degraus e instantes de tempo
14
15 ref = 10;           % Amplitude do degrau de entrada do sistema
16 Tref = 600;        % Instante em que o degrau de referencia eh
    aplicado
17 dist = 5;          % Amplitude do degrau do disturbio
18 Tdist = 2*Tref;    % Momento em que o disturbio eh aplicado
19 Thold = Tref;      % Tempo em que deseja que o disturbio seja
    aplicado
20 Tdistdown = Tdist+Thold; % Momento em que o disturbio eh
    retirado
21 Trefdown = Tref+Tdistdown; % Momento em que a referencia retorna
    para zero
22 Tsim = Tref+Trefdown; % Tempo total de execucao no Simulink
23 Ts = 5;            % Tempo de amostragem
24
25 %% Discretizacao do sinal
26
27 s = tf([1 0],1);
28
29 K = 2.4673;
30 t1 = 17.6991;
31 t2 = 52.0833;
32 tau = sqrt(t1*t2);
33 lambda = 3*tau;
34
35 sysBot = minreal(K/((t1*s+1)*(t2*s+1)));

```

```

36 Z_sysBot = c2d(sysBot, Ts, 'zoh');
37
38 %% Controlador e planta
39
40 P = 0.3105; % Valor Kp do controlador
41 I = 0.0044; % Valor Ki do controlador
42 D = 4.1019; % Valor Kd do controlador
43 C = pid(P, I, D);
44
45 Pa = 0.0565;
46 Pb = 0.0192;
47 Ta = 1/Pa;
48 Tb = 1/Pb;
49 Td = Tb*(50/100); % Td a 50% da constante de tempo do polo mais
    lento
50 Qs = 1/((Td*s+1)^2); % Filtro do observador de disturbio
51
52 Kg = 0.002677;
53
54 Gs = Kg/((s+Pa)*(s+Pb));
55 G_inv_Q = Gs^(-1) * Qs;

```

## B.2 Scripts para calculo dos indicadores-chave de performance

Código B.2 – Função "calcTV" para calcular TV

```

1 function [TVC_U] = calcTV(U,dt)
2     tFinal = length(U);
3     TVC_U = 0;
4     for i = (1+1):1:tFinal %% t0 = 1
5         TVC_U = TVC_U + abs(U(i) - U(i-1))*dt;
6     end

```

Código B.3 – Função "calcIAE" para calcular IAE

```

1 function IAE_y = calcIAE(Y, Yd, dt)
2
3     tFinal = length(Y);
4     t = 1: dt: tFinal*dt; % process time vector
5
6     IAE_y = 0;
7     for i = 1 :1: tFinal %% t0 = 1
8         IAE_y = IAE_y + abs(Y(i)-Yd(i))*dt;
9     end

```

Código B.4 – Função "calcISE" para calcular ISE

```

1 function IAE_y = calcISE(Y, Yd, dt)
2
3     tFinal = length(Y);
4     t = 1: dt: tFinal*dt; % process time vector
5
6     IAE_y = 0;
7     for i = 1 :1: tFinal %% t0 = 1
8         IAE_y = IAE_y + ((Y(i)-Yd(i))^2)*dt;
9     end

```

### B.3 Scripts para automatizar testes com os diversos valores de $T_d$

Código B.5 – Script para testes na arquitetura base com observador

```

1 clear
2
3 load_variables
4
5 p_vector = [30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90,
6            95, 100, 110, 120]; % Porcentagens a serem testadas
7 [rows,columns] = size(p_vector);
8
9 for c = 1:columns
10     disp(['Runing Ts ', num2str(p_vector(1,c)), '%'])
11     Td = Tb*(p_vector(1,c)/100);
12     Qs = 1/((Td*s+1)^2);
13
14     disp('Runing arq2 (MOD)')
15     workspace_name = ['arq2_tm', num2str(p_vector(1,c))];
16
17     sim('arq2', Tsim); % Roda arquitetura com observador de
18                       % disturbio e sem filtro estatistico (MOD)
19
20     save(workspace_name);
21
22     pause(20);
23
24     disp('Runing arq4 (MOF)')
25     workspace_name = ['arq4_tm', num2str(p_vector(1,c))];
26
27     sim('arq4', Tsim); % Roda arquitetura com observador de
28                       % disturbio e com filtro estatistico (MOF)
29
30     save(workspace_name);
31     pause(20);
32 end

```

## B.4 Script para selecionar os melhores valores de $T_d$

Código B.6 – Script que usa as funções da seção B.2 para selecionar o melhor observador de distúrbio

```

1  addpath('.');
2  addpath('./functions/');
3
4  clear
5  close all
6  path = 'Final\'; % Pasta com os arquivos dos testes finais
7
8  p_vector = [30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90,
9            95, 100, 110, 120]; % Vetor com todas as porcentagens testadas
9  [r,columns] = size(p_vector);
10 filter = 6; % Filtro para obter os x melhores valores de cada
    parâmetro
11
12 Tmin = 1100;
13 Tmax = 2300;
14
15
16 load([path, 'arq1_result'], 'sinalControle', 'sinalSaida', 'Tsim',
    'sinalRef') % Arq1 é o nome dado durante os testes para a
    arquitetura MFB
17 name = 'MFB';
18 idx = (sinalRef.time > Tmin & sinalRef.time < Tmax);
19
20 tv_1 = calcTV(sinalControle.data(idx),Tsim);
21 ise_1 = calcISE(sinalSaida.data(idx),sinalRef.data(idx),Tsim);
22 iae_1 = calcIAE(sinalSaida.data(idx),sinalRef.data(idx),Tsim);
23 disp([name, ' : tv = ', num2str(tv_1), ' | iae = ',
    num2str(iae_1), ' | ise = ', num2str(ise_1)])
24
25 load([path, 'arq3_result'], 'sinalControle', 'sinalSaida', 'Tsim',
    'sinalRef') % Arq3 é o nome dado durante os testes para a
    arquitetura MFE
26 name = 'MFE';
27 idx = (sinalRef.time > Tmin & sinalRef.time < Tmax);
28
29 tv_3 = calcTV(sinalControle.data(idx),Tsim);
30 ise_3 = calcISE(sinalSaida.data(idx),sinalRef.data(idx),Tsim);
31 iae_3 = calcIAE(sinalSaida.data(idx),sinalRef.data(idx),Tsim);
32 disp([name, ' : tv = ', num2str(tv_3), ' | iae = ',
    num2str(iae_3), ' | ise = ', num2str(ise_3)])
33
34 for arq = 1:2
35     for c = 1:columns
36         if arq == 1
37             filename = [path, 'arq2_tm', num2str(p_vector(1,c)),
                '.mat']; % Arq2 é o nome dado durante os testes
                para a arquitetura MOD
38             arq_name = 'MOD';

```

```

39     else
40         filename = [path, 'arq4_tm', num2str(p_vector(1,c)),
41                     '.mat']; % Arq4 é o nome dado durante os testes para
42                               a arquitetura MOF
43         arq_name = 'MOF';
44     end
45
46     load(filename, 'sinalControle', 'sinalSaida', 'Tsim',
47           'sinalRef')
48     idx = (sinalRef.time > Tmin & sinalRef.time < Tmax);
49     tv = calcTV(sinalControle.data(idx),Tsim);
50     ise =
51         calcISE(sinalSaida.data(idx),sinalRef.data(idx),Tsim);
52     iae =
53         calcIAE(sinalSaida.data(idx),sinalRef.data(idx),Tsim);
54     if exist('tv_vec','var') == 0
55         tv_vec = [tv];
56         ise_vec = [ise];
57         iae_vec = [iae];
58     else
59         tv_vec = [tv_vec tv];
60         ise_vec = [ise_vec ise];
61         iae_vec = [iae_vec iae];
62     end
63 end
64
65 % Ordena o o vetor com os indicadores do menor valor para o
66 maior
67 [sorted_tv,idx_tv] = sort(tv_vec);
68 [sorted_iae,idx_iae] = sort(iae_vec);
69 [sorted_ise,idx_ise] = sort(ise_vec);
70
71 chosen_values = intersect(idx_tv(1:filter),
72                           idx_iae(1:filter)); % Confere os melhores valores
73 correspondentes de TV e IAE (quantidade de valores
74 selecionados é determinada por filter) e armazena em
75 chosen_values
76
77 chosen_values = intersect(chosen_values, idx_ise(1:filter)); %
78 Confere os melhores valores correspondentes de ISE e
79 chosen_values (que já possui os correspondentes de TV e
80 IAE) e armazena em chosen_values
81
82 best = [];
83
84 % Identifica qual o valor da porcentagem corresponde aos
85 melhores valores armazenados em chosen_values
86 for i = chosen_values
87     best = [best p_vector(1,i)];
88     disp([num2str(p_vector(1,i)), ' : tv = ',
89          num2str(tv_vec(i)), ' | iae = ', num2str(iae_vec(i)), '
90          | ise = ', num2str(ise_vec(i))])
91 end

```

```
75     disp(['Valores selecionados como melhores para arq ',
76          arq_name, ': '])
77     disp(best)
78
79     bestTV = [];
80     bestIAE = [];
81     bestISE = [];
82     for i = 1:filter
83         bestTV = [bestTV p_vector(1,idx_tv(i))];
84         bestIAE = [bestIAE p_vector(1,idx_iae(i))];
85         bestISE = [bestISE p_vector(1,idx_ise(i))];
86     end
87     disp(['Melhores TV para arq ', arq_name])
88     disp(bestTV)
89     disp(['Melhores IAE para arq ', arq_name])
90     disp(bestIAE)
91     disp(['Melhores ISE para arq ', arq_name])
92     disp(bestISE)
93
94     clear tv_vec ise_vec iae_vec
95 end
```