

# PROJETO E IMPLEMENTAÇÃO DE CONTROLADOR DIGITAL NO ESPAÇO DE ESTADOS

Aluno: João Antônio Cardoso

Prof. Flábio Alberto Bardemaker Batista

# SUMÁRIO

<b>Lista de ilustrações</b>	2
<b>Lista de tabelas</b>	3
<b>Introdução</b>	4
<b>1 Objetivos</b>	4
<b>2 Desenvolvimento</b>	4
2.1 Modelagem da Planta	5
2.1.1 Resposta ao degrau da planta modelada	6
2.2 Identificação da Planta	8
2.3 Planta em Malha Aberta	9
2.4 Requisitos de Projeto	10
2.5 Projeto do Controlador	10
2.5.1 Período de Integração para o Controlador	11
2.5.2 Cálculo da Matriz de Ganhos do Controlador	12
2.6 Projeto do Observador	14
2.6.1 Cálculo do Observador de Ordem Mínima	15
2.7 Simulação Completa - Implementação Recursiva	16
2.7.1 Resultados	18
<b>3 Conclusões</b>	20
<b>REFERÊNCIAS</b>	21
<b>APÊNDICES</b>	22
<b>APÊNDICE A – RESULTADOS DA SIMULAÇÃO COMPLETA</b>	23
<b>APÊNDICE B – CÓDIGOS FONTE DO PROJETO</b>	28
<b>APÊNDICE C – CAPTURAS DA PLANTA EM MALHA ABERTA</b>	29
<b>APÊNDICE D – CAPTURAS DA PLANTA EM MALHA FECHADA</b>	31

# LISTA DE ILUSTRAÇÕES

Figura 1 – Planta . . . . .	5
Figura 2 – Resposta ao Degrau da Planta Modelada . . . . .	7
Figura 3 – Resposta ao Degrau da Planta Modelada - Implementação Recursiva . . . . .	7
Figura 4 – Resposta ao Degrau na Planta em Malha Aberta . . . . .	9
Figura 5 – Diagrama de blocos do projeto de servossistemas sem observador . . . . .	10
Figura 6 – Comparação das Estratégias de Posicionamentos dos Polos Não Dominantes . . . . .	11
Figura 7 – Simulação com Controlador em Malha Fechada . . . . .	13
Figura 8 – Simulação com Controlador em Malha Fechada - Implementação Recursiva . . . . .	13
Figura 9 – Simulação com Controlador em Malha Fechada - Implementação Recursiva - Ampliado . . . . .	14
Figura 10 – Diagrama de Blocos do Projeto de Servossistemas com Observador de Ordem Mínima . . . . .	15
Figura 11 – Simulação com Controlador e Observador em Malha Fechada . . . . .	17
Figura 12 – Simulação com Controlador e Observador em Malha Fechada, com Ruído . . . . .	18
Figura 13 – Resposta ao Degrau na Planta em Malha Fechada e Aberta . . . . .	19
Figura 14 – Simulação com Controlador e Observador em Malha Fechada . . . . .	23
Figura 15 – Simulação com Controlador e Observador em Malha Fechada . . . . .	23
Figura 16 – Simulação com Controlador e Observador em Malha Fechada - Estado 1	24
Figura 17 – Simulação com Controlador e Observador em Malha Fechada - Estado 2	24
Figura 18 – Simulação com Controlador e Observador em Malha Fechada - Estado 3	25
Figura 19 – Simulação com Controlador e Observador em Malha Fechada, com Ruído	25
Figura 20 – Simulação com Controlador e Observador em Malha Fechada, com Ruído	26
Figura 21 – Simulação com Controlador e Observador em Malha Fechada - Estado 1, com Ruído . . . . .	26
Figura 22 – Simulação com Controlador e Observador em Malha Fechada - Estado 2, com Ruído . . . . .	27
Figura 23 – Simulação com Controlador e Observador em Malha Fechada - Estado 3, com Ruído . . . . .	27
Figura 24 – Tempo de Subida . . . . .	29
Figura 25 – Tempo de Acomodação . . . . .	30
Figura 26 – Magnitude e Tempo de Pico . . . . .	30
Figura 27 – Tempo de Acomodação . . . . .	31
Figura 28 – Magnitude e Tempo de Pico . . . . .	32
Figura 29 – Tempo de Execução do Controle . . . . .	32
Figura 30 – Resposta ao degrau - degrau positivo e negativo . . . . .	33

## LISTA DE TABELAS

Tabela 1 – Comparaço dos resultados . . . . .	20
--	----

# INTRODUÇÃO

Este trabalho apresenta um relatório do processo de solução do primeiro trabalho da disciplina de Sistemas de Controle II, no qual se pretende projetar e implementar um controlador digital, da teoria à prática laboratorial.

## 1 OBJETIVOS

A partir de um enunciado explicando a proposta de trabalho disponibilizada pelo professor, o autor propõe os seguintes itens como objetivos deste trabalho:

- Fabricar uma planta para ser controlada.
- Identificar a função de transferência equivalente da planta.
- Especificar os requisitos do projeto.
- Modelar a planta no espaço de estados.
- Projetar um controlador no espaço de estados.
- Projetar um observador de ordem mínima no espaço de estados.
- Simular o sistema controlado e observador utilizando equações recursivas.
- Implementar o controlador e observador em um *microcontrolador*.
- Avaliar o funcionamento da solução implementada.
- (opção do autor) Desenvolver o projeto utilizando somente softwares gratuitos, respeitando suas licenças de uso, dando preferência para softwares de código aberto.

## 2 DESENVOLVIMENTO

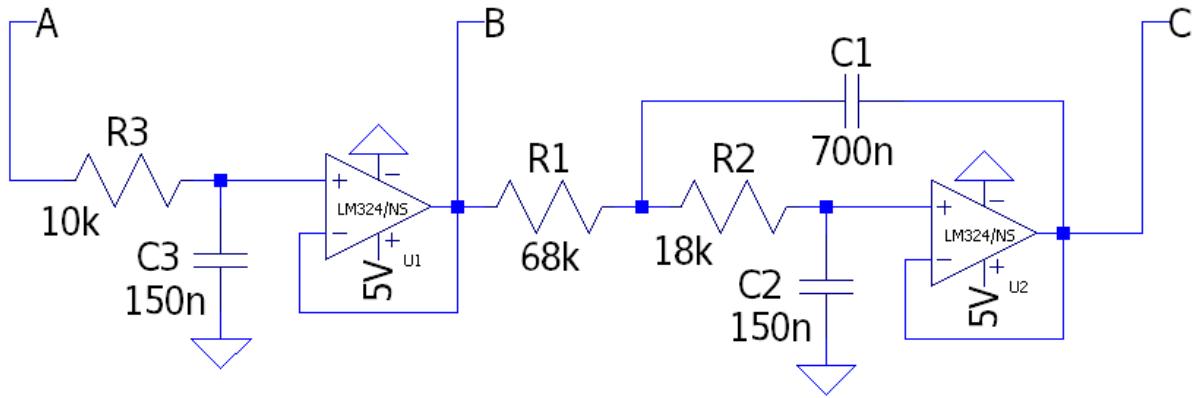
Como objeto de estudo, uma planta para ser controlada foi proposta pelo professor, e modificada de acordo com a disponibilidade de componentes e também modificada para a utilização de alimentação simples ao invés de alimentação simétrica, facilitando a montagem, que desta forma pode ocorrer diretamente por uma porta USB 1.0, comum em nossas ferramentas de trabalho como osciloscópios e computadores.

Por se tratar de um projeto de controle digital, o autor optou por utilizar um *DSP TMS320F28335* da *Texas Instruments* como controlador, imbutido em um *Control Card*, conectado à um kit de desenvolvimento chamado *C2000 Peripheral Explorer Kit*.

## 2.1 Modelagem da Planta

A planta resultante (Figura 1) consiste em três nós que são interconectados por meio de dois blocos. O bloco entre os nós **A** e **B** pode ser compreendido como um filtro passa-baixa de primeira ordem ativo de ganho unitário. O segundo bloco, entre os nós **B** e **C**, pode ser compreendido como um filtro passa-baixa de segunda ordem ativo de ganho unitário, utilizando uma topologia conhecida como *Sallen-key*.

Figura 1 – Planta



Fonte: do autor.

A partir do esquemático da planta (Figura 1), um novo esquemático foi desenhado no software de projeto de placas de circuito impresso *KiCAD*, contendo pontos de teste, conectores e *jumpers* para facilitar o teste dos blocos isoladamente. Para tal esquemático, *layout* foi desenhado, e a placa manufaturada.

Como o amplificador operacional do primeiro bloco fornece a característica de alta impedância na entrada e baixa impedância na saída, podemos caracterizar os dois blocos separadamente. Sendo assim, as equações no espaço de estados para o bloco 1 pode ser vista em 1 e do segundo bloco, na 2.

$$\begin{aligned} \dot{x}_1 &= \left[ \frac{-1}{R_3 C_3} \right] x_1 + \left[ \frac{0}{R_3 C_3} \right] u_1 \\ y_1 &= [1] x_1 + [0] u_1 \end{aligned} \quad (1)$$

$$\begin{aligned} \dot{x}_2 &= \left[ \begin{array}{cc} 0 & \frac{1}{R_2 C_1} \\ \frac{-1}{R_1 C_2} & -\left(\frac{1}{R_1 C_2} + \frac{1}{R_2 C_2}\right) \end{array} \right] x_2 + \left[ \begin{array}{c} 0 \\ \frac{1}{R_1 C_2} \end{array} \right] u_2 \\ y_2 &= [1 \ 0] x_2 + [0] u_2 \end{aligned} \quad (2)$$

Para modelar o sistema completo podemos conectar as saídas do primeiro bloco nas entradas do segundo bloco e re-arranjar as matrizes de estados para obtermos um sistema equivalente, conforme explicado por [Lanari \(2014\)](#), chegamos em 3:

$$\begin{aligned}\dot{x} &= \begin{bmatrix} \frac{-1}{R_3C_3} & 0 & 0 \\ 0 & 0 & \frac{1}{R_2C_1} \\ \frac{1}{R_1C_2} & \frac{-1}{R_1C_2} & -\left(\frac{1}{R_1C_2} + \frac{1}{R_2C_2}\right) \end{bmatrix} x + \begin{bmatrix} \frac{1}{R_3C_3} \\ 0 \\ 0 \end{bmatrix} u \\ y &= [0 \ 1 \ 0] x + [0] u\end{aligned}\quad (3)$$

Reorganizando nossa matriz para facilitar a utilização da metodologia do observador de estados de ordem mínima, mantendo o estado  $x_1$  como nosso estado a ser medido (e não observado)  $v_{C_1}$ , e também para ficar mais intuitivo, podemos fazer o estado  $x_2$  coincidir com  $v_{C_2}$  e  $x_3$  com  $v_{C_3}$ . Para tal, podemos trocar as posições da primeira e terceira linhas/colunas de todas as matrizes e consequentemente trocar as posições da primeira e e segunda, ficando com 4:

$$\begin{aligned}\dot{x} &= \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} v_{C_1} \\ v_{C_2} \\ v_{C_3} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & \frac{1}{R_2C_1} & 0 \\ \frac{-1}{R_1C_2} & -\left(\frac{1}{R_1C_2} + \frac{1}{R_2C_2}\right) & \frac{1}{R_1C_2} \\ 0 & 0 & \frac{-1}{R_3C_3} \end{bmatrix}}_A x + \underbrace{\begin{bmatrix} 0 \\ 0 \\ \frac{1}{R_3C_3} \end{bmatrix}}_B u \\ y &= \underbrace{\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}}_C x + \underbrace{\begin{bmatrix} 0 \end{bmatrix}}_D u\end{aligned}\quad (4)$$

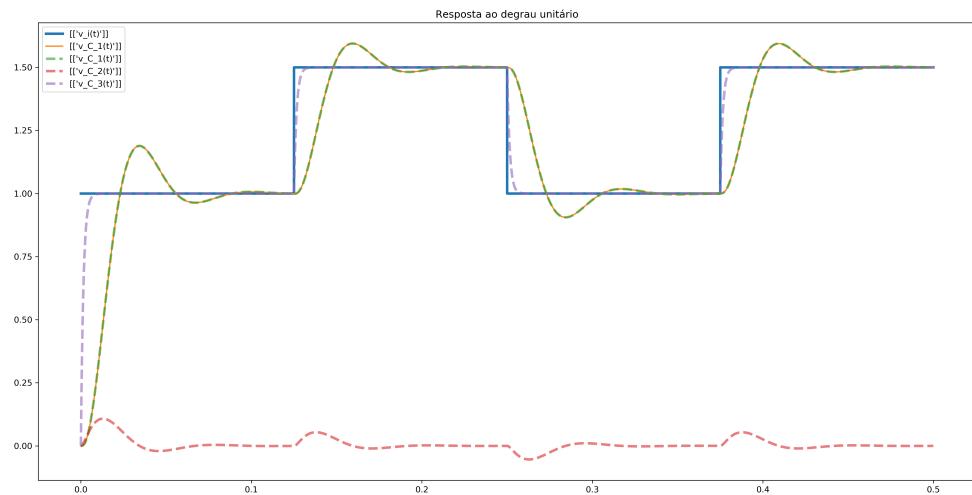
Como projetar um controlador que funcione para toda a gama de tolerância dos componentes está fora do contexto desta disciplina, vamos projetar um controlador considerando apenas os valores nominais dos componentes, resultando em 5:

$$\begin{aligned}\dot{x} &= \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} v_{C_1} \\ v_{C_2} \\ v_{C_3} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 559.44 & 0 \\ -21.01 & -100.93 & 21.01 \\ 0 & 0 & -666.67 \end{bmatrix}}_A x + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 666.67 \end{bmatrix}}_B u \\ y &= \underbrace{\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}}_C x + \underbrace{\begin{bmatrix} 0 \end{bmatrix}}_D u\end{aligned}\quad (5)$$

### 2.1.1 Resposta ao degrau da planta modelada

Utilizando um software de simulação matemática podemos aplicar um sinal arbitrário na entrada de nosso espaço de estados com uma entrada e uma saída (*SISO*) e plotar todos os seus estados, além de sua entrada e saída, conforme a [Figura 2](#):

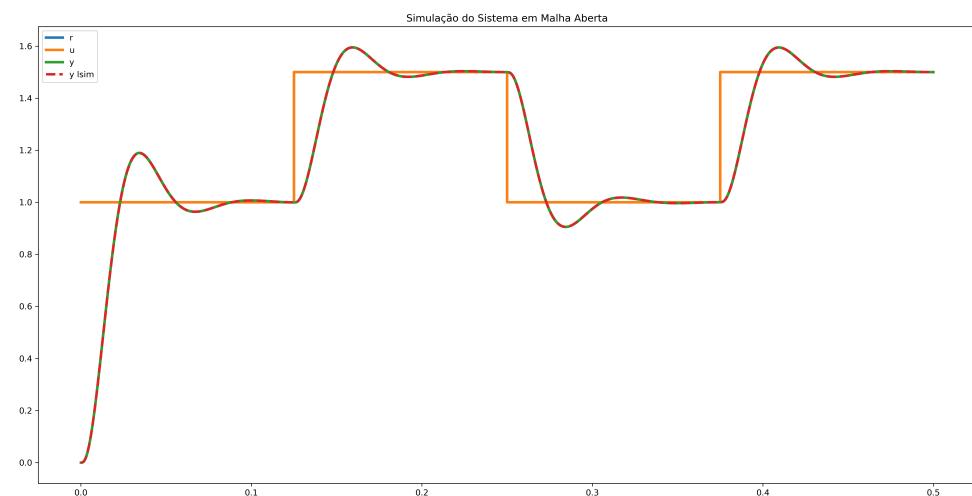
Figura 2 – Resposta ao Degrau da Planta Modelada



Fonte: do autor.

Podemos também realizar uma simulação para o sistema em malha aberto resolvendo as equações do espaço de estados da planta modelada recursivamente, obtendo a Figura 3, no qual pode ser percebida a grande semelhança entre as saídas realizadas na simulação utilizando a função `control.forced_response()`, aqui chamada de `lsim`, por ser equivalente à função do *MATLAB*.

Figura 3 – Resposta ao Degrau da Planta Modelada - Implementação Recursiva



Fonte: do autor.

## 2.2 Identificação da Planta

Uma técnica para a identificação da planta pode ser aplicar um sinal conhecido e a partir de figuras de mérito, levantar seus parâmetros e equacionar uma função de transferência de segundo grau equivalente.

Por se tratar de uma planta que trabalha em uma região de tensão positiva na entrada e saída, e também por sabermos que o circuito possui não linearidades próximo aos valores de sua alimentação, aqui iremos aplicar um degrau de 1 à 1.5 [V], ou seja, não é um degrau unitário, mas é um sinal conhecido e tomando alguns cuidados, toda a teoria que utiliza degraus unitários pode ser utilizada.

Para deixar explícito como foram computadas as figuras de mérito, as Equações 6 à 7 descrevem o modo com que foram calculados os valores de *Tempo de Subida* (*Rise Time* -  $t_r$ ) *Sobressinal* (*Overshoot* -  $M_p$ ), o *Tempo de Acomodação em 5%* (*Settling Time* -  $t_{s5\%}$ ).

$$\begin{aligned} V_{10\%} &= 0,1 \cdot (V_\infty - V_0) \\ V_{90\%} &= 0,9 \cdot (V_\infty - V_0) \\ t_r &= t_{(V_{90\%})} - t_{(V_{10\%})} \end{aligned} \quad (6)$$

$$\begin{aligned} V_{max_{5\%}} &\leq (1 + 0.05) \cdot (V_\infty - V_0) + V_0 \\ V_{min_{5\%}} &\leq (1 - 0,05) \cdot (V_\infty - V_0) + V_0 \\ t_{s5\%} &= max(t_{(V_{max_{5\%}})}, t_{(V_{min_{5\%}})}) \end{aligned} \quad (7)$$

Antes de realizar as medições, a primeira etapa foi executar a auto-calibração do osciloscópio e calibrar manualmente as ponteiras, para garantir que não houvessem erros técnicos nas medições.

Com o osciloscópio devidamente calibrado, verificou-se o sinal *PWM* de entrada, se estava na frequência de operação correta, e se estava nos ciclos-tarefa como foi calculado.

O gatilho do osciloscópio foi configurado no canal 2, que mostra o momento em que o degrau é aplicado, portanto, todas as medidas de tempo com os cursores são relativos ao tempo em que o degrau foi aplicado, que será considerado como o tempo inicial ( $t_0 = 0 [s]$ ).

A primeira ponteira do osciloscópio foi então conectada na saída da planta. O resultado é mostrado na [Figura 4](#).

Para medir com maior precisão, optou-se por trabalhar com o diferentes escalas do osciloscópio para focar em cada parte da forma de onda. A fim de alcançar uma precisão ainda melhor no osciloscópio, foi utilizado o *trigger* no *modo normal*, que permitiu que a função de média de 128 amostras na aquisição atuasse como uma sobre-amostragem,

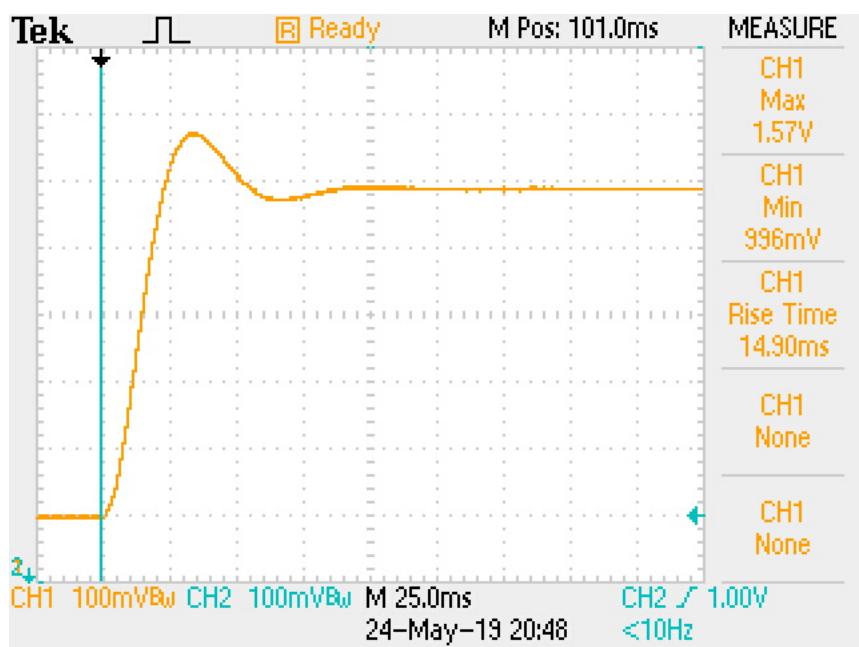
adicionando pelo menos três bits de precisão no desenho da forma de onda, reduzindo consideravelmente o ruído aleatório.

Para que o relatório não fique demasiadamente longo, as capturas do osciloscópio detalhadas para cada aquisição de tempo de subida, tempo de acomodação e tempo de pico constam no [Apêndice C](#) para a planta completa em malha aberta.

### 2.3 Planta em Malha Aberta

Podemos considerar a planta em malha aberta como uma *caixa preta* e obter suas figuras de mérito aplicando um sinal arbitrário em sua entrada.

Figura 4 – Resposta ao Degrau na Planta em Malha Aberta



Resposta ao degrau de 0,5 [V] aplicado pela tensão média do *PWM*. No Canal 1 (Amarelo) é a saída da planta e o canal 2 (azul), o *trigger*. Fonte: do autor.

No [C](#) pode ser verificado a captura para a obtenção dos parâmetros, que foram:

$$V_0 = 0 [V]$$

$$V_\infty = 1,49 [V]$$

$$t_{subida} = 15,0 [ms]$$

$$t_{50\%} = 12,0 [ms]$$

$$t_{85\%} = 47,6 [ms]$$

$$V_{pk} = 1.57 [V]$$

$$t_{pk} = 34,0 [ms]$$

$$M_p = 16,3265\%$$

$$\zeta = 0,4997062$$

$$w_n = 103,330164 [rad/s]$$

Tendo a planta caracterizada, precisamos definir nossos requisitos de projeto.

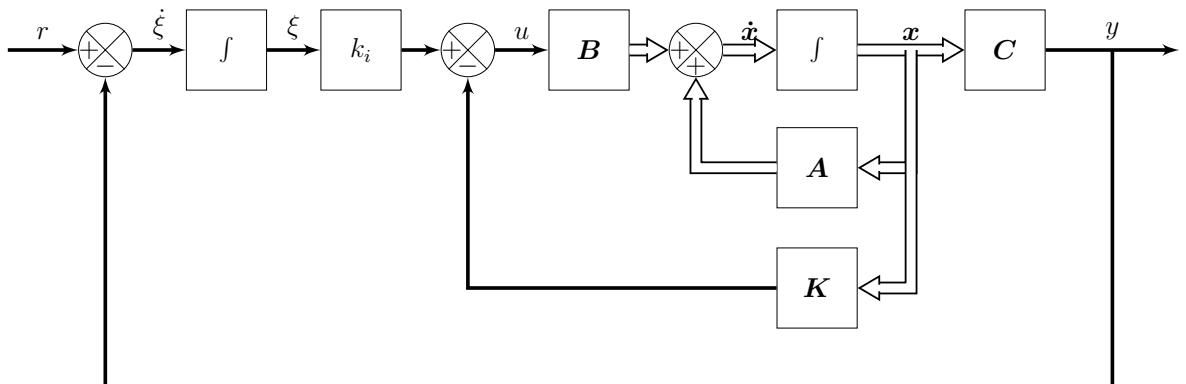
## 2.4 Requisitos de Projeto

Como objetivo do projeto, o professor propôs um controlador que modifique o *Tempo de Acomodação* de 5% e o *Sobressinal* para a metade de seus valores em malha aberta, ou seja, neste caso ficamos com  $t_{s5\%} \leq 25,4 [ms]$  e  $M_p \leq 8,05\%$ . Outra exigência é que haja erro nulo em regime permanente e que o sistema controlado seja estável.

## 2.5 Projeto do Controlador

A topologia para o sistema em malha fechada utilizada aqui é de um controlador de servossistema com a inserção de um integrador, como mostrado em [Figura 5](#).

Figura 5 – Diagrama de blocos do projeto de servossistemas sem observador



Fonte: do autor.

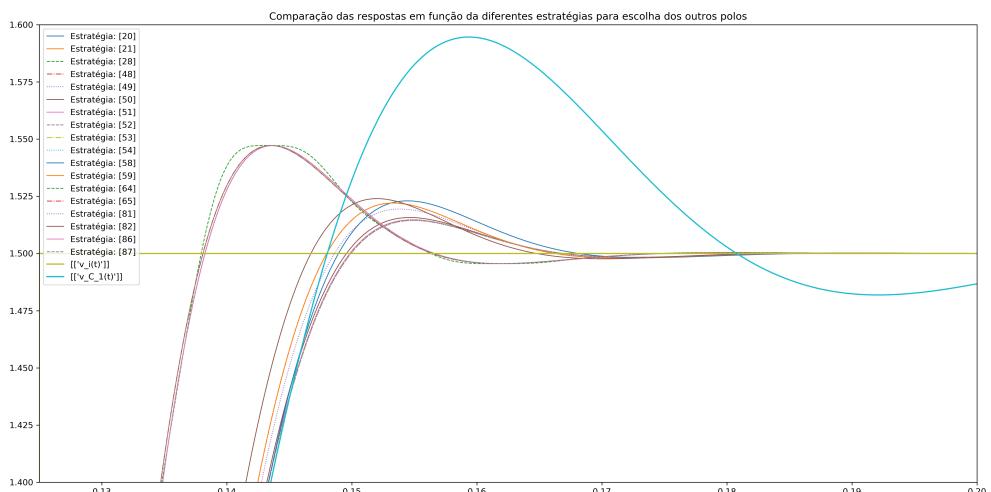
Para alcançar os requisito de projeto, precisamos definir a localização dos polos dominantes quando fecharmos a malha.

Para tal, pode-se utilizar uma relação analítica aproximada entre as figuras de mérito de um sistema de segunda ordem e os parâmetros de um sistema de segundo grau conhecidos como fator de amortecimento ( $\zeta$ ) e frequência natural ( $w_n$ ). É importante salientar que são aproximações e que ao utilizarmos essas equações estamos admitindo uma margem de erro nos resultados do projeto, e também, que serão necessários quatro polos para o sistema e a localização dos dois polos não dominantes pode influenciar drasticamente nos resultados.

O projetista escreveu uma rotina em *Python* que variava a posição dos polos não dominantes seguindo algumas estratégias de posicionamento, testava o resultado dos sistemas (Figura 6) e selecionava o de menor sobressinal, resultando nos seguintes polos para o projeto do controlador conforme 8, que foram posicionados utilizando o segundo e terceiro polos como um par complexo conjugado, reduzindo o valor de sua parte imaginária por um fator ampliando a parte real para seu dobro. Na Figura 6 a estratégia escolhida é mostrada como *Estratégia [54]*, tendo um  $\zeta$  de 0.7484 e um  $w_n$  de  $158.1 \text{ rad/s}$ , alcançando um sobressinal de 2.89% e um tempo de acomodação de  $22,4 \text{ ms}$ .

$$j = \begin{bmatrix} -128.52101782 + 171.08783376i & -128.52101782 - 171.08783376i \\ -257.04203564 + 1.71087834i & -257.04203564 - 1.71087834i \end{bmatrix} \quad (8)$$

Figura 6 – Comparação das Estratégias de Posicionamentos dos Polos Não Dominantes



Comparação de resposta ao degrau dos sistemas controlados com o diferentes estratégias para o posicionamento dos polos não dominantes. Fonte: do autor.

### 2.5.1 Período de Integração para o Controlador

Como projetaremos um controlador digital, precisamos definir o período de integração que será empregado nos integradores do sistema. Para que o controlador alcance a resposta do sistema, consideramos aqui uma frequência de no mínimo duas vezes a frequência natural amortecida do sistema, que resultaria em aproximadamente  $1320 \text{ Hz}$ .

Neste ponto, o autor gostaria de deixar claro que houveram diversas iterações utilizando *scripts* automatizados para projetar e testar o controle utilizando algumas taxas

diferentes e chegou à conclusão que valores acima de  $2kHz$  funcionavam bem, optando por ficar em **3kHz**.

Utilizando tal frequência para o integrador do controlador, temos um período aproximadamente 4,5 vezes menor que o período da frequência natural amortecida da planta.

### 2.5.2 Cálculo da Matriz de Ganhos do Controlador

Primeiramente verificamos se a nossa planta pode ser controlada verificando se sua matriz de controlabilidade tem o mesmo número de equações linearmente independentes que a quantidade de estados modelados. Para o sistema modelado o número de equações linearmente independentes coincide com seu número de estados, logo, é controlável.

O controlador pode ser calculado a partir do método de *Ackermann* para o sistema da [Equação 9](#) utilizando os polos definidos anteriormente em [Equação 8](#), obtendo a matriz de ganhos do controlador  $\mathbf{K}$ , mostrado na [Equação 10](#).

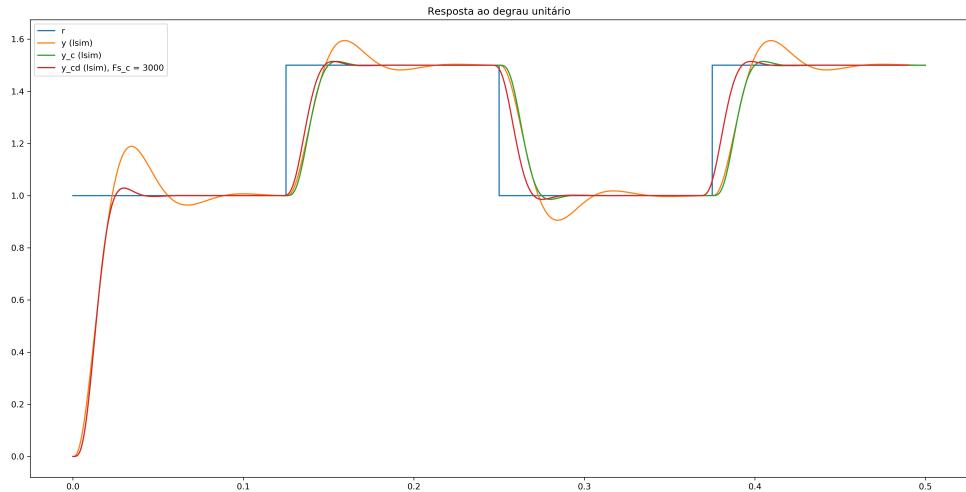
$$\dot{e} = (\hat{\mathbf{A}} - \hat{\mathbf{B}} \hat{\mathbf{K}}) e \quad (9)$$

$$\begin{aligned} \hat{\mathbf{A}} &= \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ -\mathbf{C} & 0 \end{bmatrix} = \begin{bmatrix} 0. & 559.44055944 & 0. & 0. \\ -21.00840336 & -100.92848328 & 21.00840336 & 0. \\ 0. & 0. & -666.66666667 & 0. \\ -1. & 0. & 0. & 0. \end{bmatrix} \\ \hat{\mathbf{B}} &= [\mathbf{B}] \begin{bmatrix} 0. \\ 0. \\ 666.66666667 \\ 0. \end{bmatrix} \\ \hat{\mathbf{C}} &= [\mathbf{C} \ 0.] = [1. \ 0. \ 0. \ 0.] \\ \hat{\mathbf{D}} &= \mathbf{D} = [0] \\ \hat{\mathbf{K}} &= [\mathbf{K} \ -K_i] \\ \mathbf{K} &= [4.16654212e + 00 \ 1.17530446e + 01 \ 5.29643546e - 03] \quad (10) \end{aligned}$$

$$K_i = 386.12694507724626 \quad (11)$$

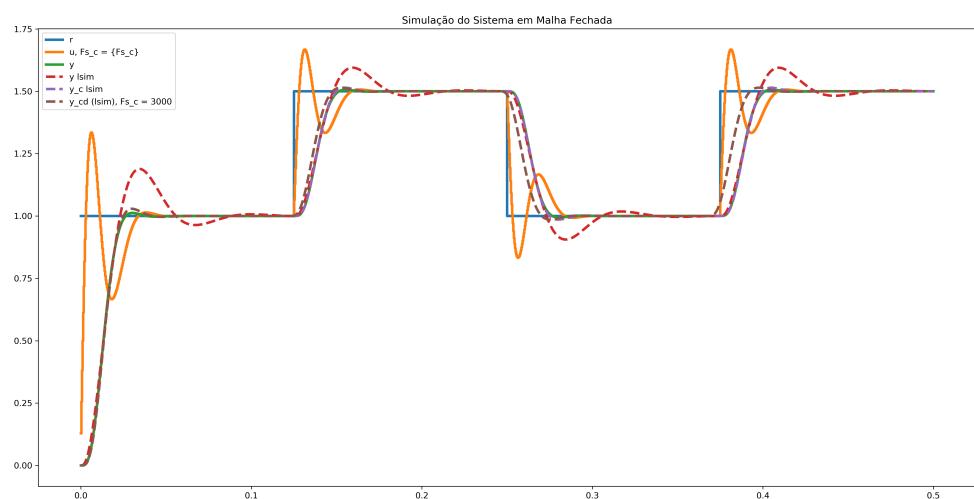
Utilizando as matrizes de estados do sistema controlado  $\hat{\mathbf{A}}$ ,  $\hat{\mathbf{B}}$ ,  $\hat{\mathbf{C}}$  e  $\hat{\mathbf{D}}$  podemos simular o sistema controlado e também o sistema controlado amostrado na frequência de integração escolhida, como mostrado na [Figura 7](#), coerente com sua implementação recursiva, que pode ser visto nas Figuras [8](#) e [9](#).

Figura 7 – Simulação com Controlador em Malha Fechada



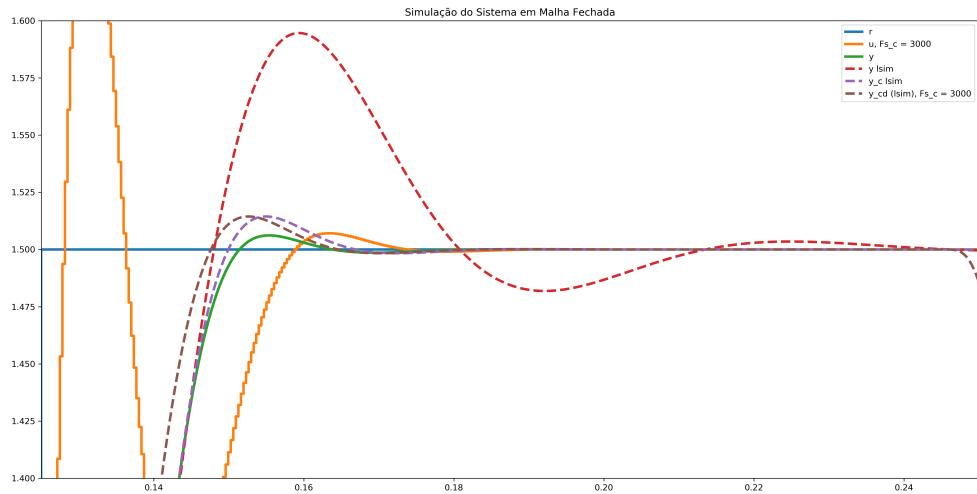
Fonte: do autor.

Figura 8 – Simulação com Controlador em Malha Fechada - Implementação Recursiva



Fonte: do autor.

Figura 9 – Simulação com Controlador em Malha Fechada - Implementação Recursiva - Ampliado



Fonte: do autor.

## 2.6 Projeto do Observador

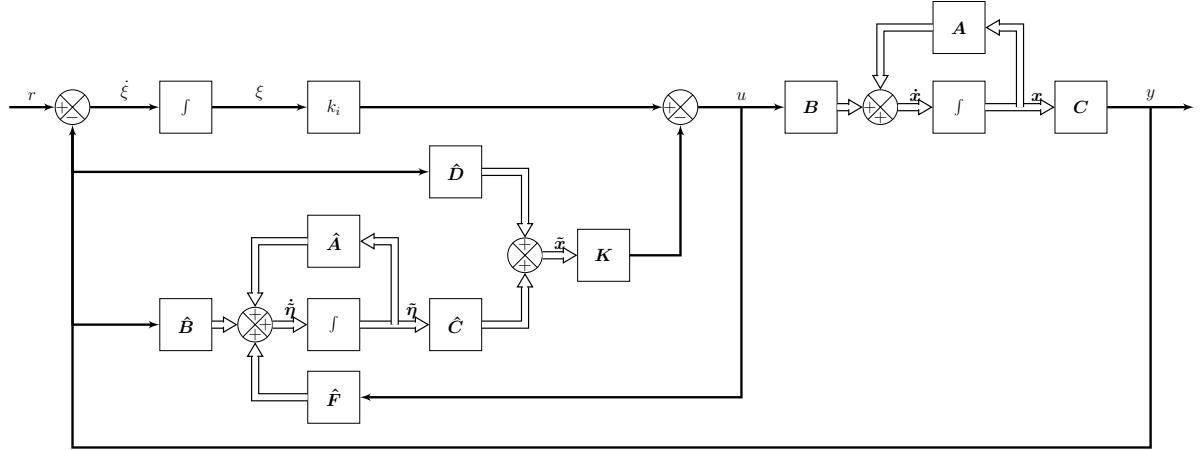
Para que seja possível obter os estados da planta sem a necessidade de implementar um sensor para cada um dos três estados, pode-se fazer o uso de um observador, que pode ser entendido como um modelo da planta simplificado do qual se utiliza para obter estimativas dos estados ( $\tilde{x}$ ) a partir de uma implementação realimentada. Aqui vamos utilizar um observador de ordem mínima, que no nosso caso, irá estimar os estados de  $x_2$  e  $x_3$  a partir da medição da saída  $y$ , que coincide com o estado  $x_1$ . A topologia do controlador de servossistema utilizando um observador pode ser visto na [Figura 10](#).

Para que o observador consiga estimar os estados da planta controlada, podemos definir seus polos como o dobro da parte real dos polos dominantes do controlador, ficando com os polos mostrados na [Equação 12](#).

$$l = \begin{bmatrix} -257.04203564 & -257.04203564 \end{bmatrix} \quad (12)$$

O autor achou interessante utilizar um período de integração para o observador duas vezes menor que o período de integração do controlador, e ainda realizar uma sobre-amostragem, realizando 4 amostragens da variável de saída para cada observação de estados, criando assim, uma melhoria na precisão dos valores adquiridos.

Figura 10 – Diagrama de Blocos do Projeto de Servossistemas com Observador de Ordem Mínima



Fonte: do autor.

### 2.6.1 Cálculo do Observador de Ordem Mínima

Primeiramente verificamos se a nossa planta pode ser observada verificando se sua matriz de observabilidade tem o mesmo número de equações linearmente independentes que a quantidade de estados modelados a serem observados. Para o sistema modelado o número de equações linearmente independentes coincide com seu número de estados -1, logo, é observável.

O observador pode ser calculado a partir do método de *Ackermann* para o sistema da [Equação 13](#) utilizando os polos definidos anteriormente em [Equação 12](#), obtendo a matriz  $\mathbf{K}_e$  e as matrizes de espaço de estados do observador  $\hat{\mathbf{A}}$ ,  $\hat{\mathbf{B}}$ ,  $\hat{\mathbf{C}}$ ,  $\hat{\mathbf{D}}$  e  $\hat{\mathbf{F}}$ .

$$\dot{e} = (\mathbf{A}_{bb}' - \mathbf{K}_e \mathbf{A}_{ab})' e \quad (13)$$

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} A_{aa} & A_{ab} \\ A_{ba} & A_{bb} \end{bmatrix} \\ \mathbf{B} &= \begin{bmatrix} B_a & B_b \end{bmatrix} \end{aligned}$$

$$\begin{aligned}
A_{aa} &= 0 \\
\mathbf{A}_{ab} &= \begin{bmatrix} 559.44055944 & 0. \end{bmatrix} \\
\mathbf{A}_{ba} &= \begin{bmatrix} -21.00840336 \\ 0. \end{bmatrix} \\
\mathbf{A}_{bb} &= \begin{bmatrix} -100.92848328 & 21.00840336 \\ 0. & -666.66666667 \end{bmatrix} \\
B_a &= 0 \\
\mathbf{B}_b &= \begin{bmatrix} 0 \\ 666.66666667 \end{bmatrix} \\
\\
\hat{\mathbf{A}} &= \mathbf{A}_{bb} - \mathbf{K}_e \mathbf{A}_{ab} = \begin{bmatrix} 152.58259539 & 21.00840336. \\ -7986.91530517 & -666.66666667 \end{bmatrix} \\
\hat{\mathbf{B}} &= \hat{\mathbf{A}} \mathbf{K}_e + \mathbf{A}_{ba} - \mathbf{K}_e A_{aa} = \begin{bmatrix} 209.77743764 \\ -5898.46165694 \end{bmatrix} \\
\hat{\mathbf{C}} &= \begin{bmatrix} \mathbf{0} \\ \mathbf{I}. \end{bmatrix} = \begin{bmatrix} 0. & 0. \\ 1. & 0. \\ 0. & 1. \end{bmatrix} \\
\hat{\mathbf{D}} &= \begin{bmatrix} 0 \\ \mathbf{K}_e \end{bmatrix} = \begin{bmatrix} 1. \\ -0.45315105 \\ 14.27661111 \end{bmatrix} \\
\hat{\mathbf{F}} &= \mathbf{B}_b - \mathbf{K}_e B_a = \begin{bmatrix} 0. \\ 666.66666667 \end{bmatrix} \\
\mathbf{K}_e &= \begin{bmatrix} -0.45315105 \\ 14.27661111 \end{bmatrix} \tag{14}
\end{aligned}$$

## 2.7 Simulação Completa - Implementação Recursiva

Utilizando as matrizes de estados da planta, a matriz de ganho do controlador e as matrizes de espaço de estados do observador podemos simular o sistema em recursividade para um sinal arbitrário e assim comparar os estados obtido através do observador e a partir da planta simulada através do comando *control.forced\_response()*, que é equivalente ao *lsim* do *MATLAB*.

Para melhorar a simulação, o autor considerou:

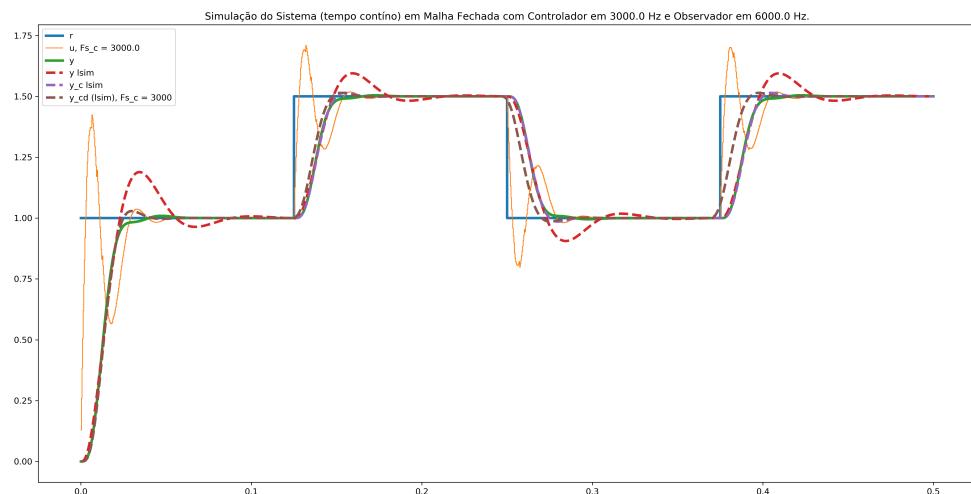
- Quantização no ADC de 12 bits
- Quantização no DAC de 12 bits

- Frequência de integração do Observador em 6kHz
- Frequência de integração do Controlador em 3kHz
- Frequência de integração da planta (passo de simulação) em 50kHz
- Ruído branco na saída planta de 100 mV

Os resultados gerais podem ser vistos na [Figura 11](#) sem ruído, e na [Figura 12](#), com o ruído.

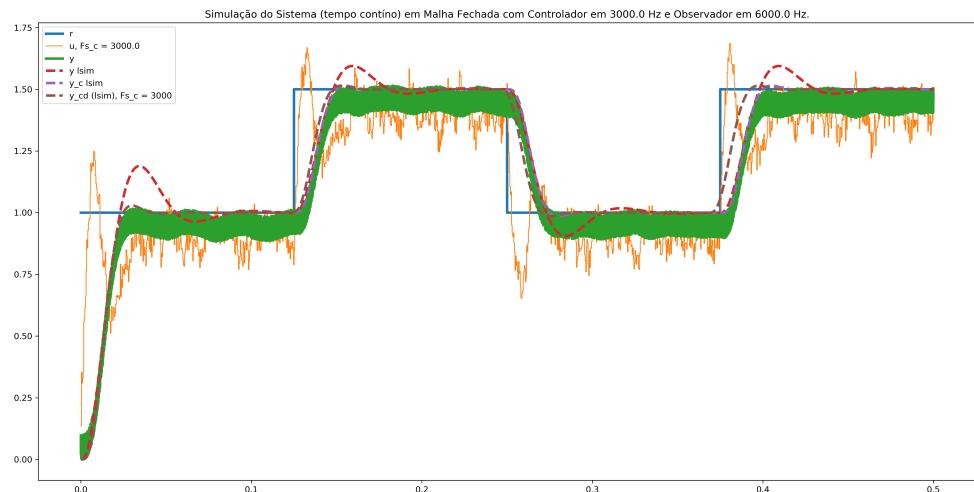
Figuras mais detalhadas com a comparação de cada estado do observador, assim como ampliação com detalhes nos degraus de subida e de descida, podem ser vistos no [Apêndice A](#).

Figura 11 – Simulação com Controlador e Observador em Malha Fechada



Fonte: do autor.

Figura 12 – Simulação com Controlador e Observador em Malha Fechada, com Ruído



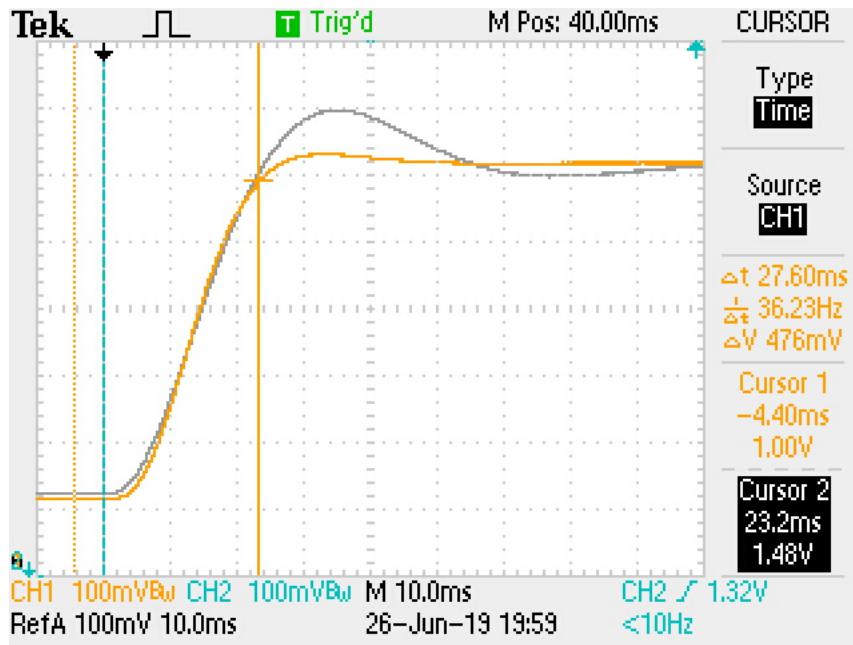
Fonte: do autor.

### 2.7.1 Resultados

Um *script*(Apêndice B) gerou as constantes utilizadas no *firmware* implementado no *DSP*. A resposta ao degrau para o sistema em malha fechada em comparação ao mesmo sistema em malha aberta pode ser visto na Figura 13. As formas de onda apresentam resultados similares aos esperados pela simulação da Figura 8.

O sinal da ação de controle **u** se manteve dentro dos valores de simulação, verificado através de ferramentas de *debug* (entre 0,7 e 1,8 V).

Figura 13 – Resposta ao Degrau na Planta em Malha Fechada e Aberta



Resposta ao degrau de 0,5 [V] aplicado pela tensão média do *PWM*. No Canal 1 (Amarelo) é a saída do sistema em malha fechada, o canal de referência A (Branco), representando a resposta ao degrau do sistema em malha aberta, e o canal 2 (azul), o *trigger*. Fonte: do autor.

No [Apêndice D](#) pode ser verificado a captura do osciloscópio para a obtenção de cada um dos parâmetros a seguir:

$$V_0 = 1,0 \text{ [V]}$$

$$V_\infty = 1,5 \text{ [V]}$$

$$t_{subida} = 14,8 \text{ [ms]}$$

$$t_{50\%} = 13,6 \text{ [ms]}$$

$$t_{s5\%} = 23,8 \text{ [ms]}$$

$$V_{pk} = 1.51 \text{ [V]}$$

$$t_{pk} = 33,3 \text{ [ms]}$$

$$M_p = 2\%$$

$$\zeta = 0.7797$$

$$w_n = 150.22[\text{rad/s}]$$

A [Figura 2.7.1](#) apresenta uma comparação de todos os dados obtidos durante o processo.

Tabela 1 – Comparaço dos resultados

	$Mp\%$	$t_{s5\%}[ms]$	$t_r [ms]$	$v_{inf}$	$v_0$
Malha Aberta simulação	18,88	50,1	14,8	1,50	1,00
Malha Aberta prático	16,32	49,2	15,2	1,49	1,01
Malha Fechada simulação	2,88	22,4	13,8	1,50	1,00
Malha Fechada prático	2,00	23,8	14,8	1,50	1,00
Requisitos	8,05	25,0	N/A	N/A	N/A

Fonte: do autor.

### 3 CONCLUSES

Neste trabalho surgiram algumas dificuldades no âmbito algébrico, mas que foram aos poucos sendo superados com o decorrer do projeto.

Para facilitar a implementação do projeto de controle no microcontrolador, o autor desenvolveu uma biblioteca em código aberto em *C89*, utilizando somente macros, sendo compatível com a maior parte dos microcontroladores, possibilitando a descrição matemática do sistema facilmente seguindo o fluxo dos diagramas de blocos. Alguns outros trabalhos de colegas também tiveram sucesso ao utilizarem esta implementação em seus projetos, então o autor considera que por facilitar não só este projeto, que foi um tempo bem investido.

Neste projeto o autor optou por programar uma rotina que minimizasse os possíveis problemas causados pelos polos não dominantes do sistema, executando uma série de projetos de maneira automatizada, seguindo uma simples regra de seleção e o controlador que foi obtido se comportou de maneira estável.

Os *scripts* de simulação recursiva previram razoavelmente bem os resultados práticos, possibilitando uma série de iterações de testes e ajustes antes de propriamente implementar o controlador no microcontrolador.

Além disso, o autor cumpriu seu objetivo pessoal de realizar o trabalho inteiramente com software gratuitos, inclusive o sistema operacional (*Linux*), utilizando principalmente o *KiCAD* para o projeto da placa de circuito impresso, o pacote de controle *python-control* da linguagem de programação *Python*, o qual inclusive contribuiu adicionando a funcionalidade *stepinfo*, equivalente à função de mesmo nome do *MATLAB*, e por fim o *LaTeX* para compilação deste relatório.

Deste modo, o autor considera este projeto um processo de grande valor para a formação do estudante de engenharia eletrônica, que em exercício, atua com a interface entre a informação teórica e o que é concebível com a tecnologia e recursos disponíveis.

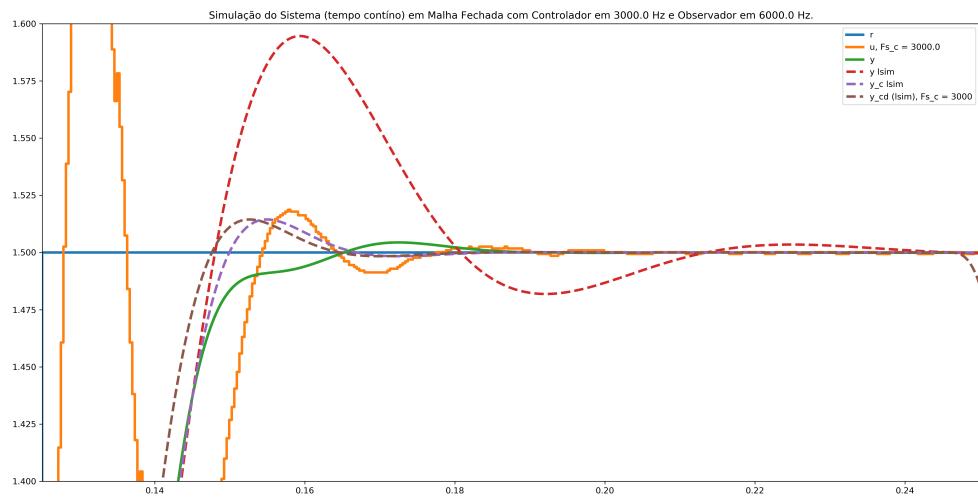
## REFERÊNCIAS

LANARI, L. *Control Systems - Interconnected systems*. Università di Roma, 2014.  
Disponível em: <[http://www.diag.uniroma1.it/~lanari/ControlSystems/CS%20-%20Lectures/2014\\_Lec\\_12\\_Interconnected\\_Systems.pdf](http://www.diag.uniroma1.it/~lanari/ControlSystems/CS%20-%20Lectures/2014_Lec_12_Interconnected_Systems.pdf)>.

# Apêndices

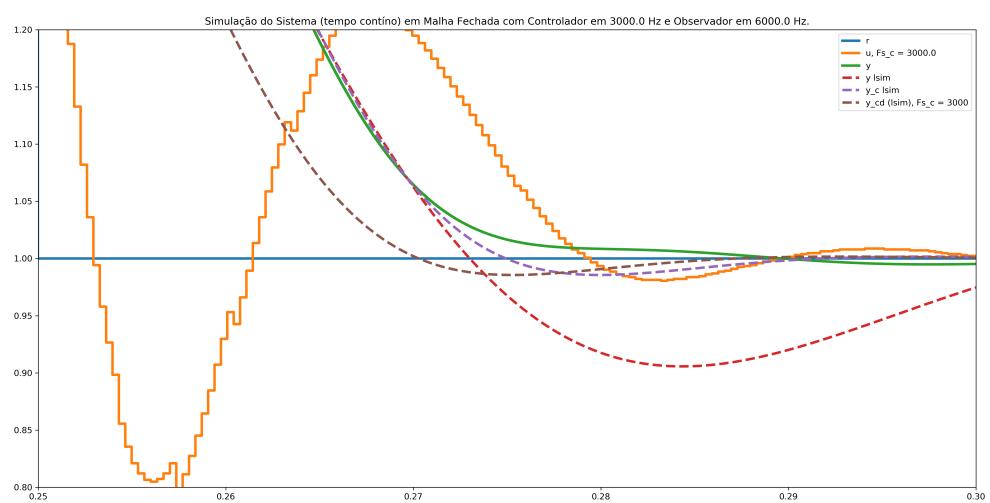
# APÊNDICE A – RESULTADOS DA SIMULAÇÃO COMPLETA

Figura 14 – Simulação com Controlador e Observador em Malha Fechada



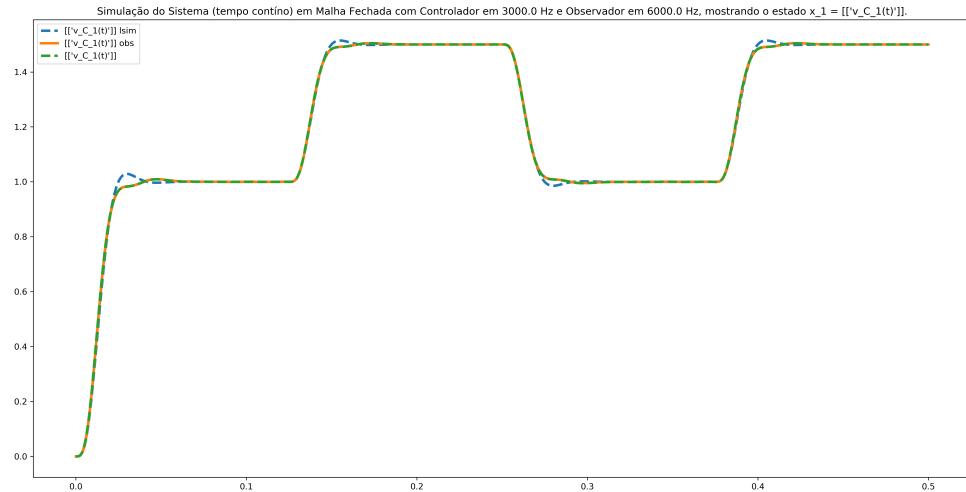
Fonte: do autor.

Figura 15 – Simulação com Controlador e Observador em Malha Fechada



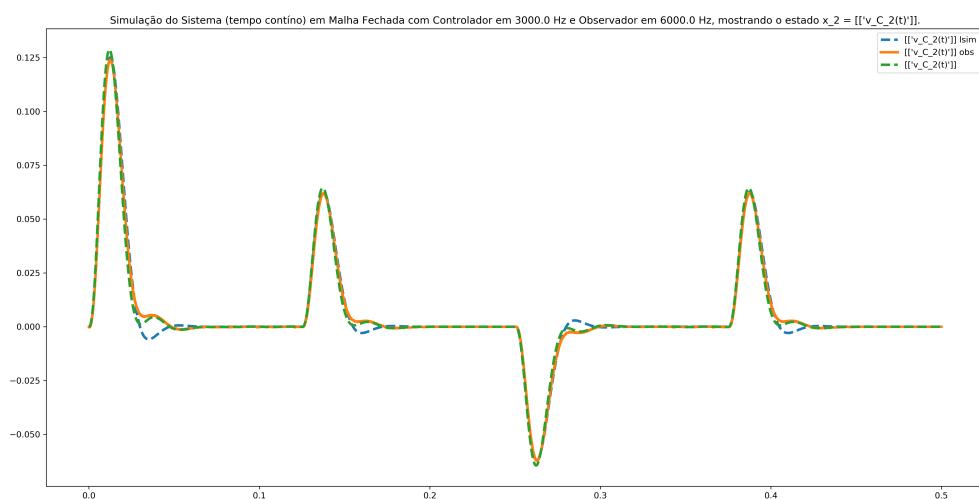
Fonte: do autor.

Figura 16 – Simulação com Controlador e Observador em Malha Fechada - Estado 1



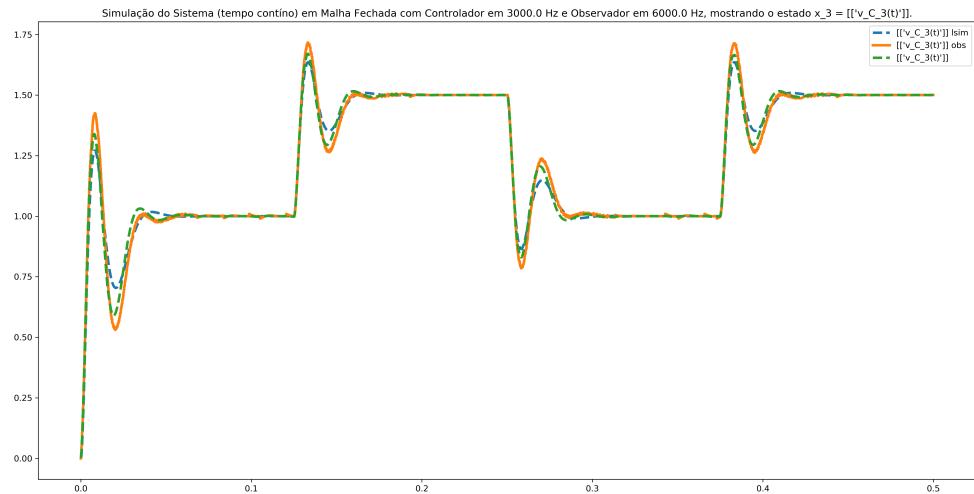
Fonte: do autor.

Figura 17 – Simulação com Controlador e Observador em Malha Fechada - Estado 2



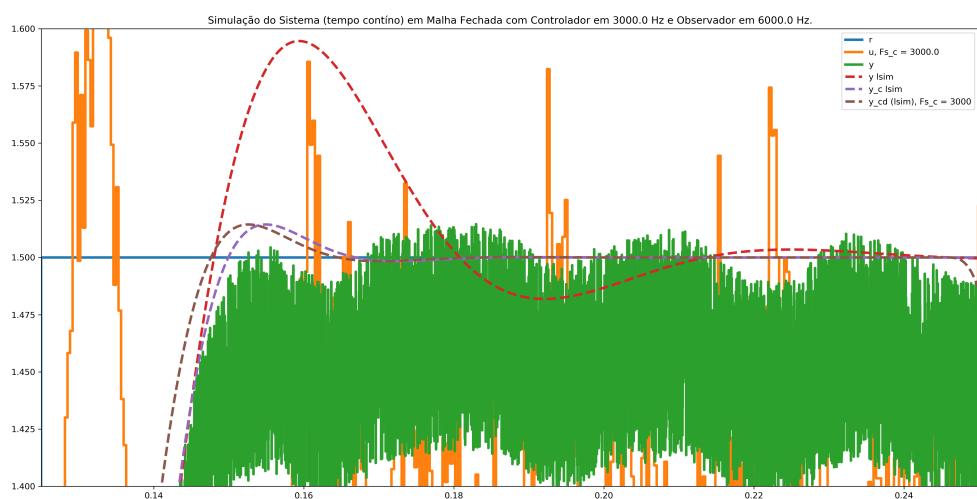
Fonte: do autor.

Figura 18 – Simulação com Controlador e Observador em Malha Fechada - Estado 3



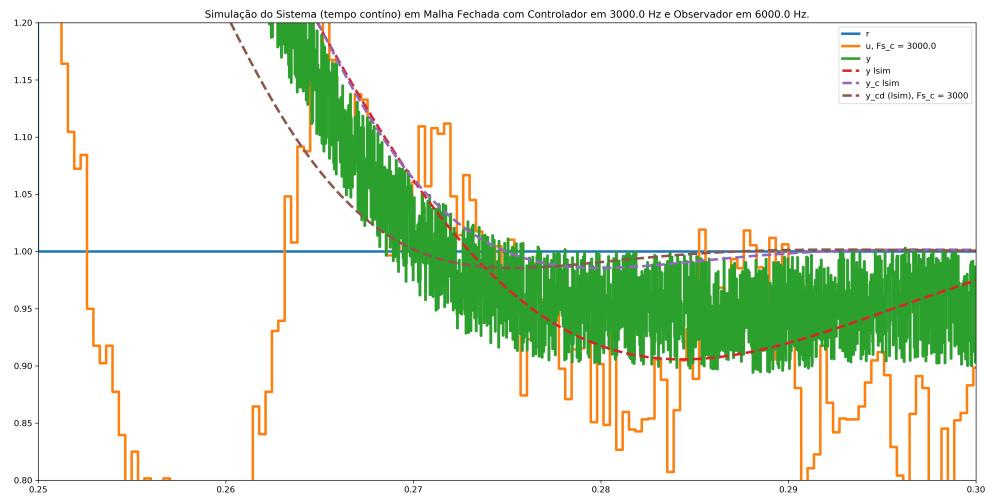
Fonte: do autor.

Figura 19 – Simulação com Controlador e Observador em Malha Fechada, com Ruído



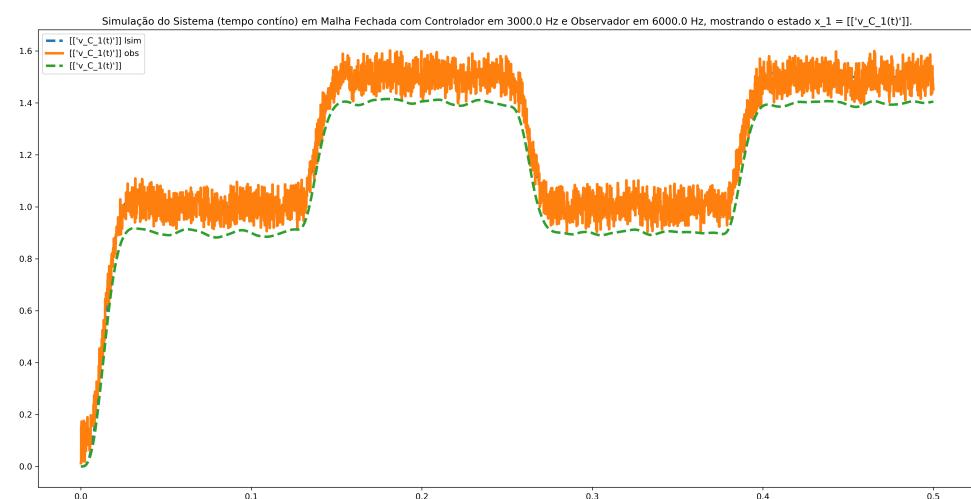
Fonte: do autor.

Figura 20 – Simulação com Controlador e Observador em Malha Fechada, com Ruído



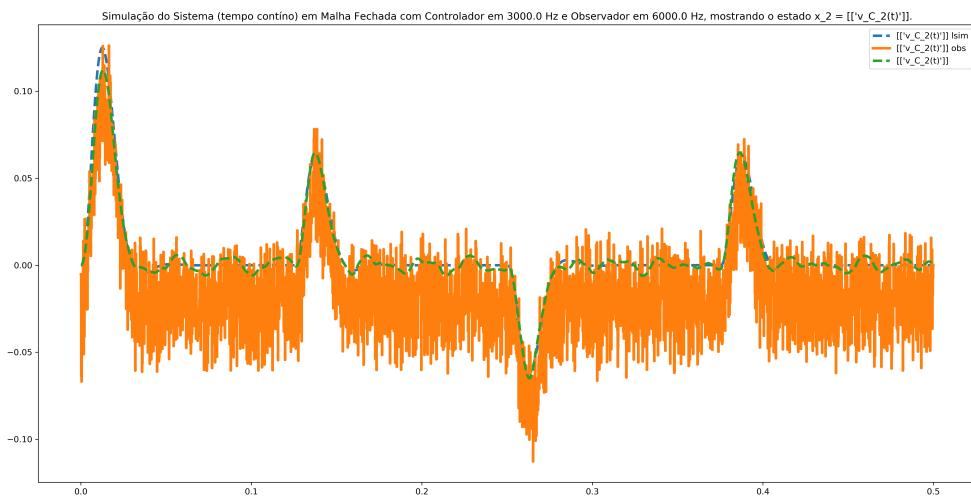
Fonte: do autor.

Figura 21 – Simulação com Controlador e Observador em Malha Fechada - Estado 1, com Ruído



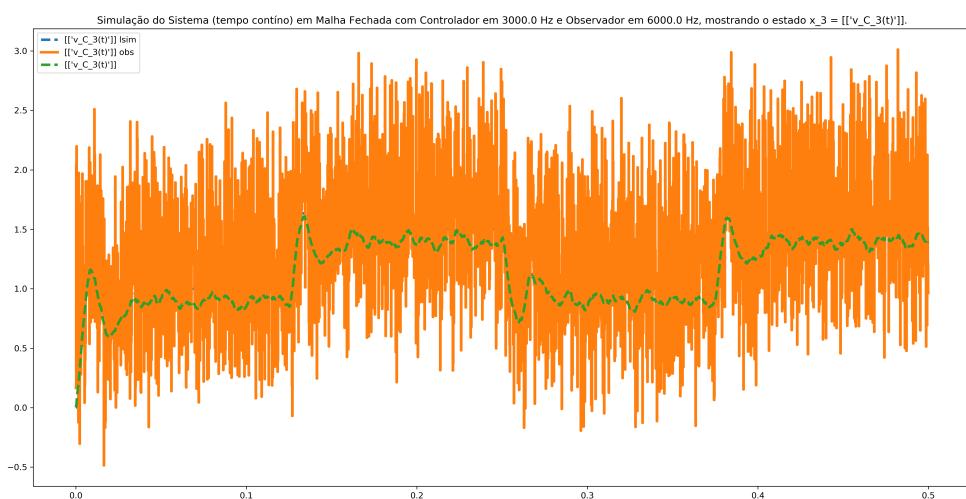
Fonte: do autor.

Figura 22 – Simulação com Controlador e Observador em Malha Fechada - Estado 2, com Ruído



Fonte: do autor.

Figura 23 – Simulação com Controlador e Observador em Malha Fechada - Estado 3, com Ruído



Fonte: do autor.

## APÊNDICE B – CÓDIGOS FONTE DO PROJETO

Por somar mais de 50 páginas, o autor opta por não incluir a listagem dos códigos fonte no relatório, deixando todos eles abertos em licença GPL3, disponíveis em um repositório público no *github* do autor.

Biblioteca desenvolvida para operações matriciais em C - C Arraay Macros:

[<https://github.com/joaoantoniocardoso/c\\_array\\_macros>](https://github.com/joaoantoniocardoso/c_array_macros)

Diretório raiz do repositório deste projeto de controle:

[<http://github.com/joaoantoniocardoso/sct2\\_projeto2>](http://github.com/joaoantoniocardoso/sct2_projeto2)

Para facilitar o acesso, o código fonte deste documento *LaTeX* podem ser acessados em:

[<https://github.com/joaoantoniocardoso/sct2\\_projeto2/tree/master/tex>](https://github.com/joaoantoniocardoso/sct2_projeto2/tree/master/tex)

O código fonte do *firmware* podem ser acessados em:

[<https://github.com/joaoantoniocardoso/sct2\\_projeto2/tree/master/f28335\\_ccs7/basic\\_F28335>](https://github.com/joaoantoniocardoso/sct2_projeto2/tree/master/f28335_ccs7/basic_F28335)

O código fonte do *hardware* podem ser acessados em:

[<https://github.com/joaoantoniocardoso/sct2\\_projeto2/tree/master/hardware/main>](https://github.com/joaoantoniocardoso/sct2_projeto2/tree/master/hardware/main)

O código fonte para as simulações no *LTS spice* podem ser acessados em:

[<https://github.com/joaoantoniocardoso/sct2\\_projeto2/tree/master/hardware/simulations>](https://github.com/joaoantoniocardoso/sct2_projeto2/tree/master/hardware/simulations)

O código fonte dos *scripts* em *Python* e *Jupyter Notebook* podem ser acessados em:

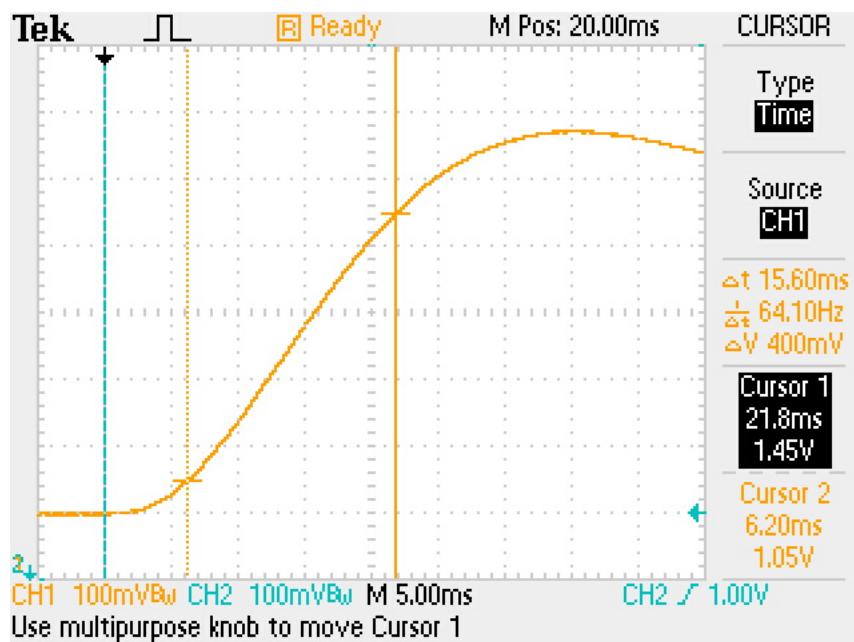
[<https://github.com/joaoantoniocardoso/sct2\\_projeto2/tree/master/controle>](https://github.com/joaoantoniocardoso/sct2_projeto2/tree/master/controle)

Por fim, o repositório pode ser baixado como um arquivo compactado em extensão *.zip* em:

[<https://github.com/joaoantoniocardoso/sct2\\_projeto2/archive/master.zip>](https://github.com/joaoantoniocardoso/sct2_projeto2/archive/master.zip)

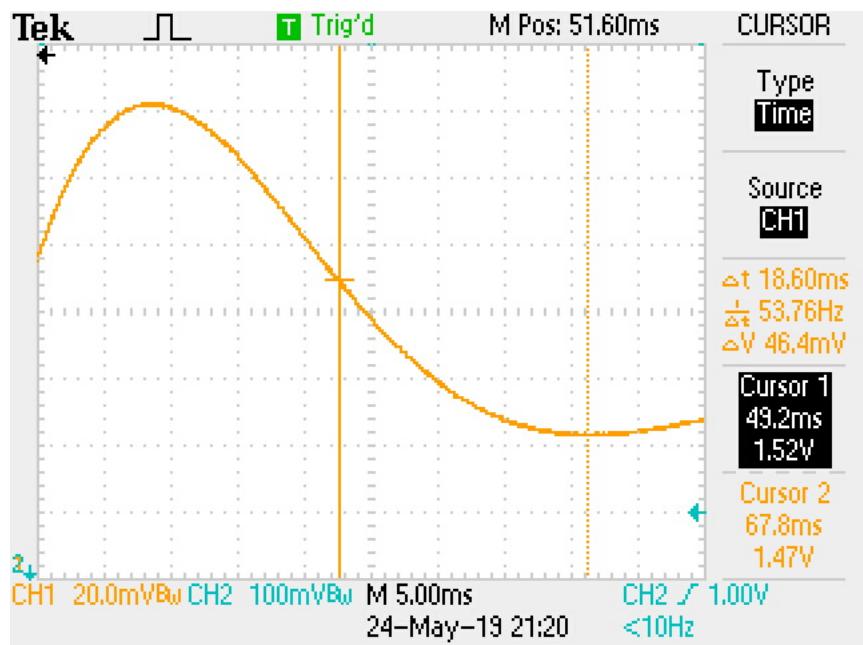
## APÊNDICE C – CAPTURAS DA PLANTA EM MALHA ABERTA

Figura 24 – Tempo de Subida



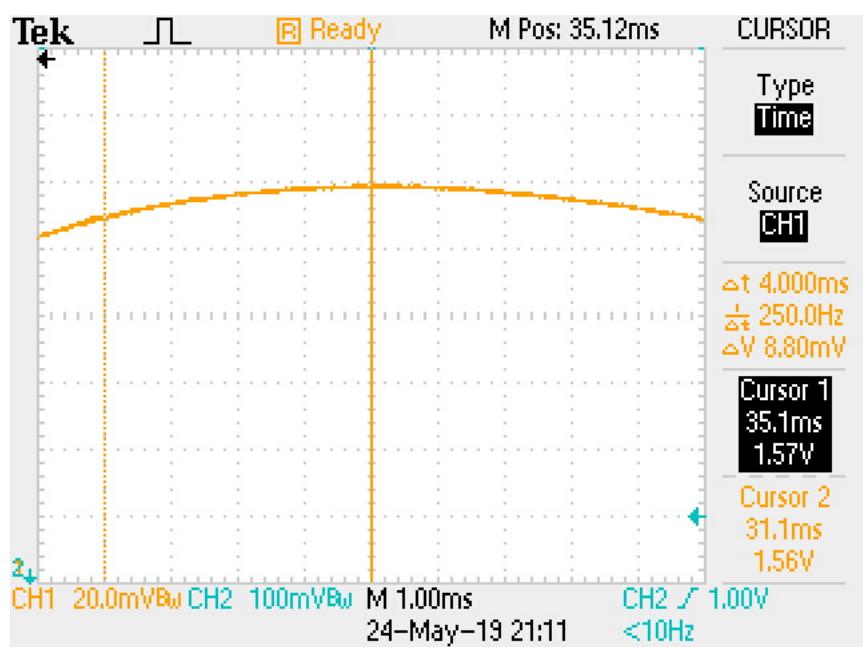
Resposta ao degrau de 0,5 [V] aplicado pela tensão média do PWM. No Canal 1 (Amarelo) é a saída da planta e o canal 2 (azul), o trigger. Fonte: do autor.

Figura 25 – Tempo de Acomodação



Resposta ao degrau de 0,5 [V] aplicado pela tensão média do PWM. No Canal 1 (Amarelo) é a saída da planta e o canal 2 (azul), o trigger. Fonte: do autor.

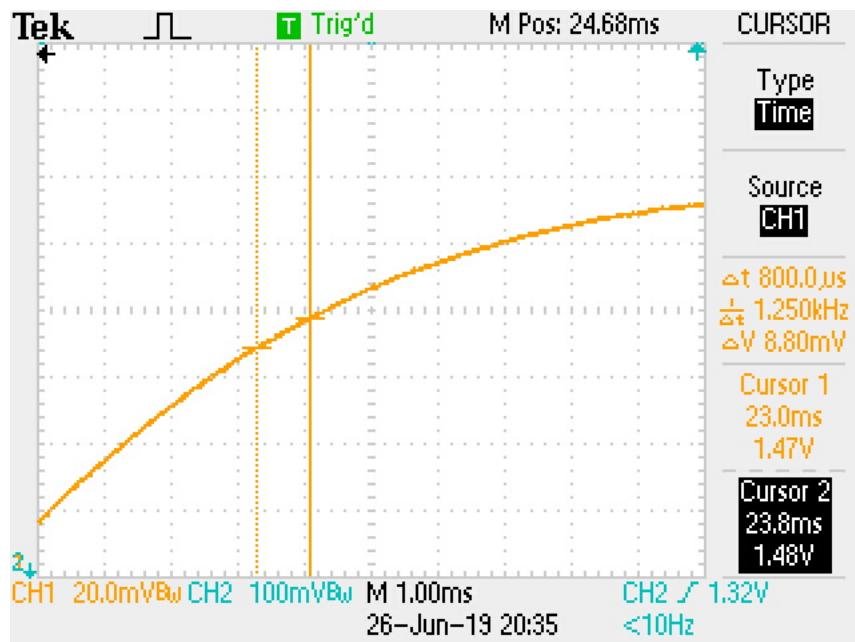
Figura 26 – Magnitude e Tempo de Pico



Resposta ao degrau de 0,5 [V] aplicado pela tensão média do PWM. No Canal 1 (Amarelo) é a saída da planta e o canal 2 (azul), o trigger. Fonte: do autor.

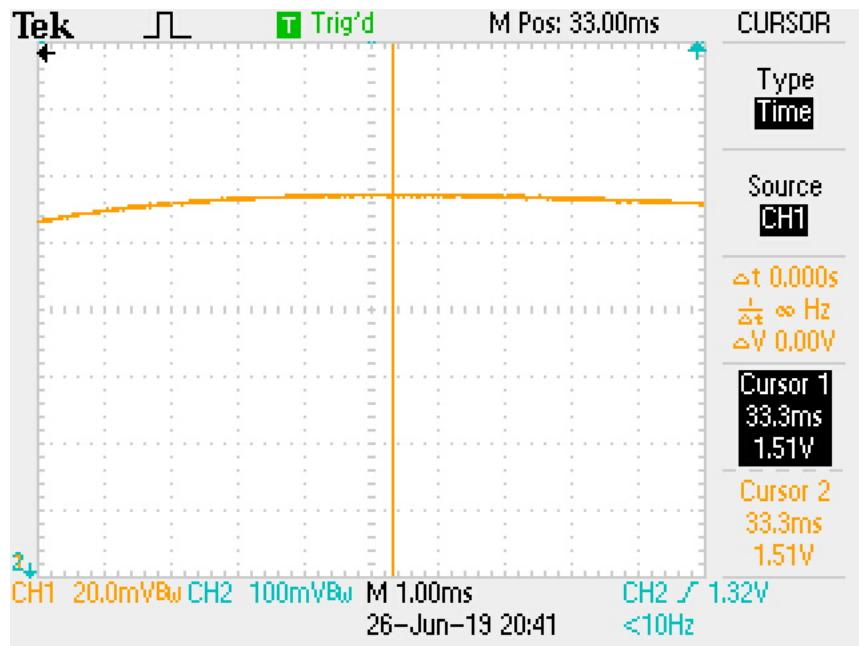
## APÊNDICE D – CAPTURAS DA PLANTA EM MALHA FECHADA

Figura 27 – Tempo de Acomodação



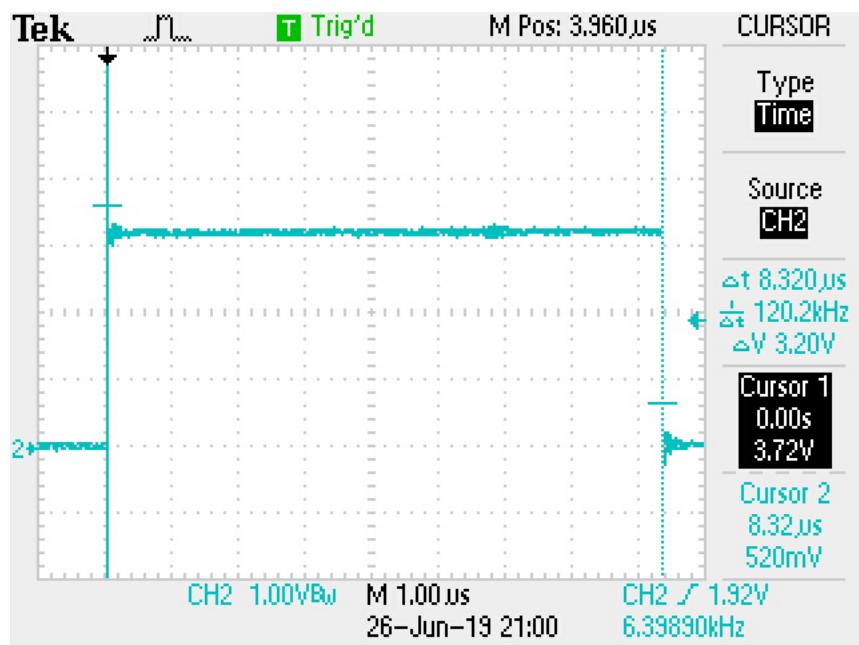
Resposta ao degrau de 0,5 [V] aplicado pela tensão média do *PWM*. No Canal 1 (Amarelo) é a saída da planta e o canal 2 (azul), o *trigger*. Fonte: do autor.

Figura 28 – Magnitude e Tempo de Pico



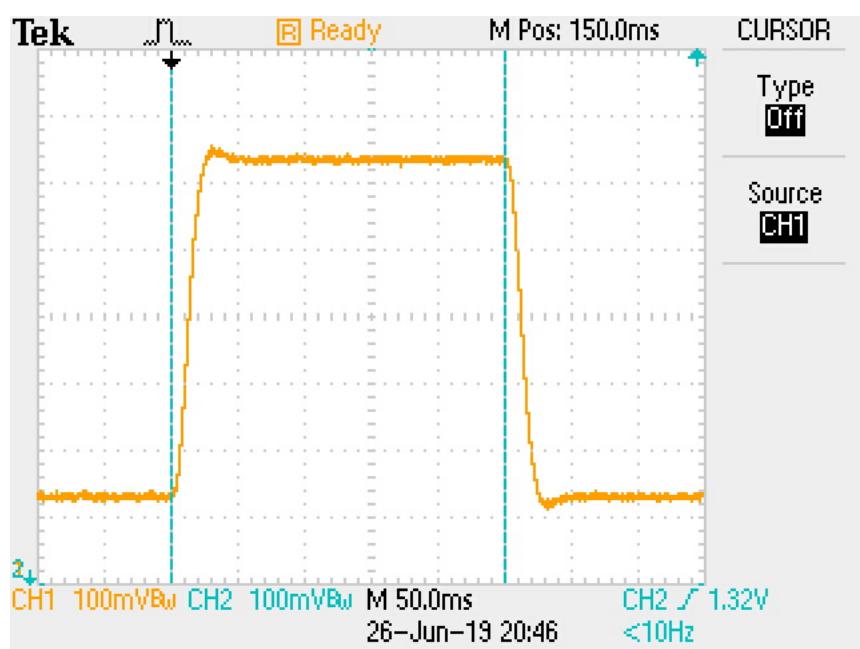
Resposta ao degrau de 0,5 [V] aplicado pela tensão média do *PWM*. No Canal 1 (Amarelo) é a saída da planta e o canal 2 (azul), o *trigger*. Fonte: do autor.

Figura 29 – Tempo de Execução do Controle



Tempo de execução em 8,32  $\mu$ s. Fonte: do autor.

Figura 30 – Resposta ao degrau - degrau positivo e negativo



Tempo de execução em  $8,32 \mu s$  Fonte: do autor.