

Daniel Pereira

# **CONTROLADOR DIGITAL**

Sistemas de Controle II

Instituto Federal de Santa Catarina – IFSC  
Departamento de Engenharia Eletrônica

Florianópolis, SC – Brasil 12 de Novembro de 2019

## Sumário

1. Introdução.....	3
2. Dimensionamento e Montagem da Placa.....	4
3. Sistema Escolhido.....	8
4. Análise do Sistema Teórico.....	9
5. Implementação Prática do Sistema.....	12
6. Controlador digital.....	21
6a. Lugar das Raízes.....	21
7. Conclusões.....	32
8. ANEXO I – CÓDIGO MATLAB.....	33
9. ANEXO II – CÓDIGO ARDUINO DUE.....	35

# 1. Introdução

Controladores digitais cada vez mais são utilizados no mundo de hoje. Isso se deve ao fato de serem sistemas confiáveis, baratos e extremamente versáteis sendo, em sua maioria, software e não hardware são fáceis de se programar e implementar, não necessitando de muito espaço físico no produto final pois seus sistemas podem ser colocados inteiramente dentro de microcontroladores assim poupando em espaço e custo.

O surgimento desta nova tecnologia possibilitou projetistas do mundo todo a capacidade de utilizar sistemas de compensação de ordem elevada de uma forma eficiente tanto em tamanho que ocupa quando aos recursos necessários o que facilitou muitas inovações e descobertas e com a evolução constante de microcontroladores (os tornando mais potentes e compactos) a utilidade e utilização dos controladores digitais só cresce.

Este trabalho consiste no planejamento e desenvolvimento de um controlado digital para reduzir o sobressinal e o tempo de acomodação da resposta em degrau do sistema apresentado abaixo na Figura 1.

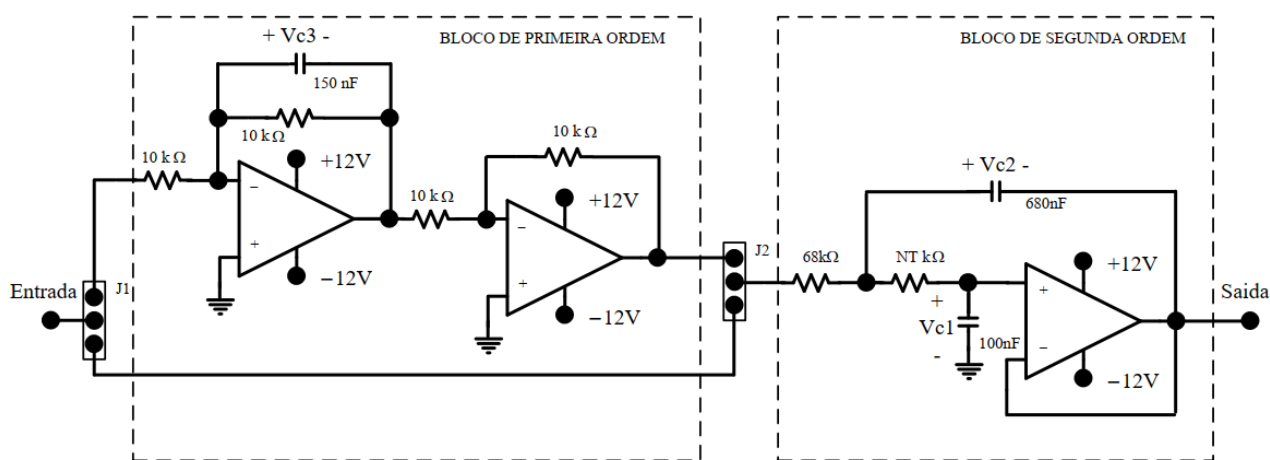


Figura 1 – Sistema proposto para o trabalho.

## 2. Dimensionamento e Montagem da Placa

Para esta etapa do projeto foi utilizado o programa Proteus V8.6 devido à versatilidade de conter um simulador de sistemas (ISIS) e um software para o dimensionamento e montagem de placas (ARES).

Tendo em vista os requisitos de projeto se escolheu o amplificador operacional LM741 para este projeto devido a cada uma de suas unidades possuir apenas um amplificador operacional assim facilitado medidas em pinos mais específicos, permitindo fazer a placa de forma mais separa em blocos e facilidade para a troca de peças.

### LM741 Pinout Diagram

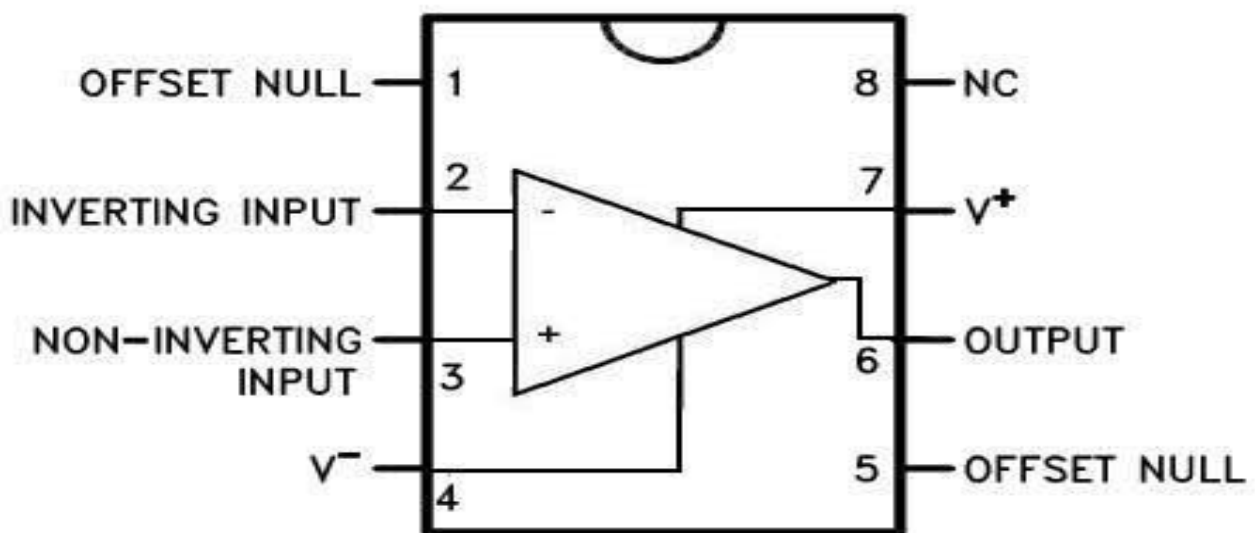


Figura 2 – LM741

Em seguida se focou em projetar a placa em si e nesta etapa o programa de simulação ISIS foi utilizando primariamente por permitia recriar o sistema sugerido identicamente a aquele proposto.

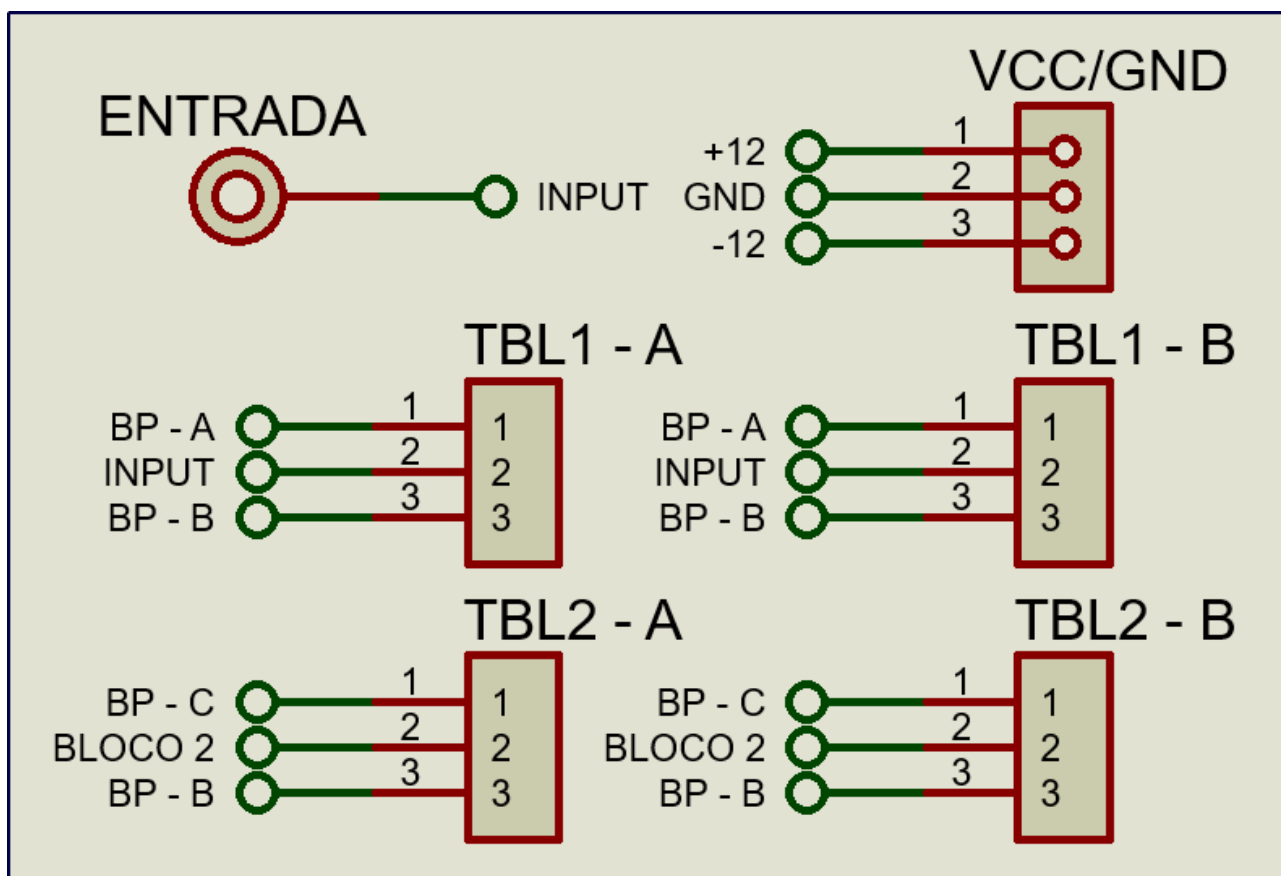


Figura 3 – Bloco de utilitários

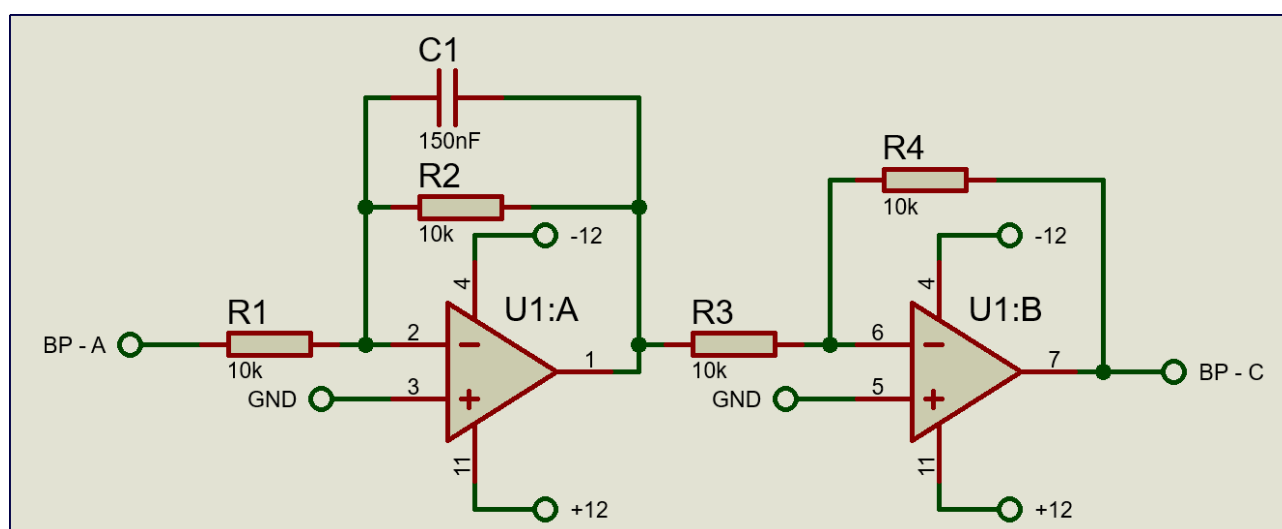


Figura 4 – Bloco de primeiro grau

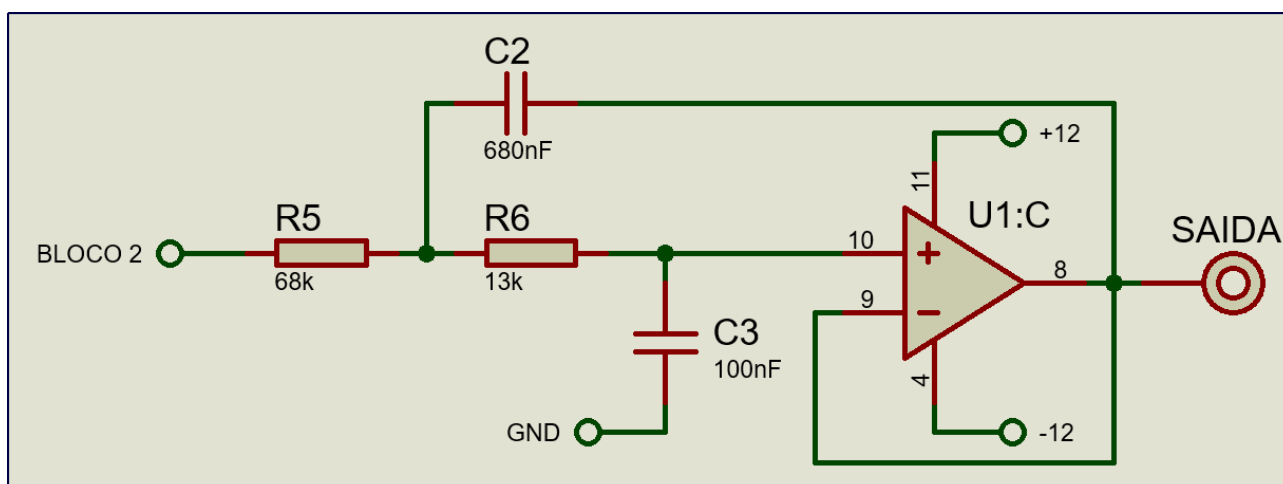


Figura 5 – Bloco de segundo grau

O bloco mencionado como “utilitários” existe somente para providenciar o barro pinos e referências que tornaram a esquematização da placa mais fácil.

A seguir, nas figuras 4 e 5, tem os blocos de primeiro e segundo grau recriados de exatamente como aquele sugerido para este trabalho.

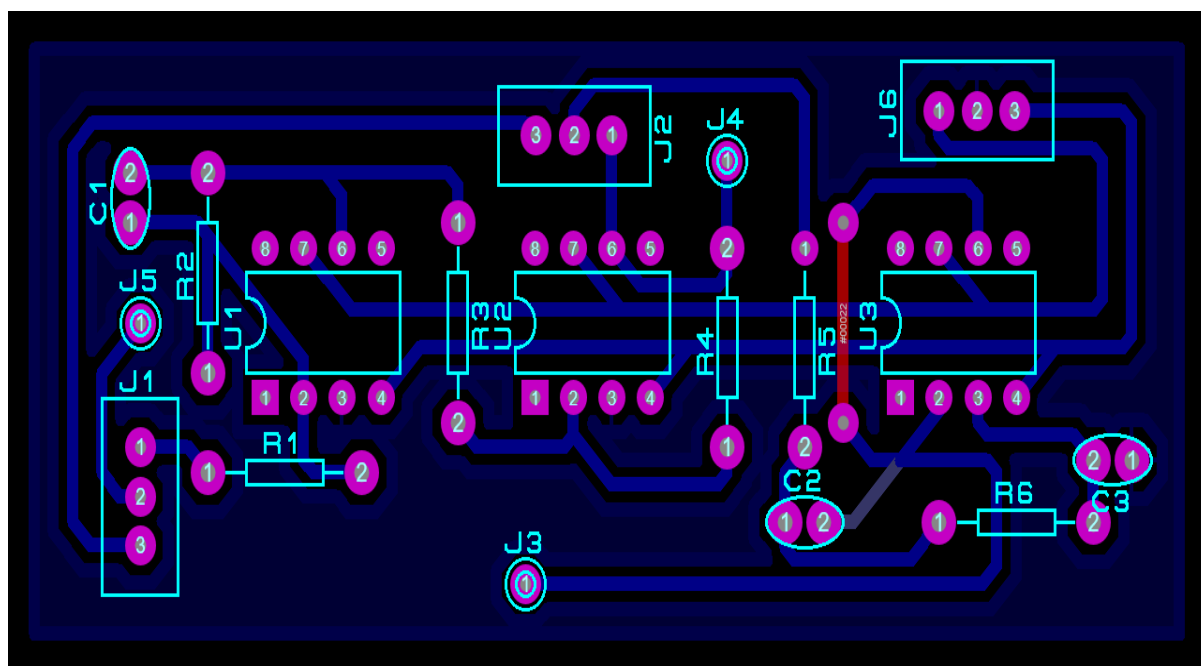


Figura 6 – Projeto da placa

A placa foi projetada visando um tamanho compacto para utilizar menos fenolite e melhor poder transportar a placa alcançando um tamanho de 7,5 cm por 3,1 cm (como visto na figura 6).

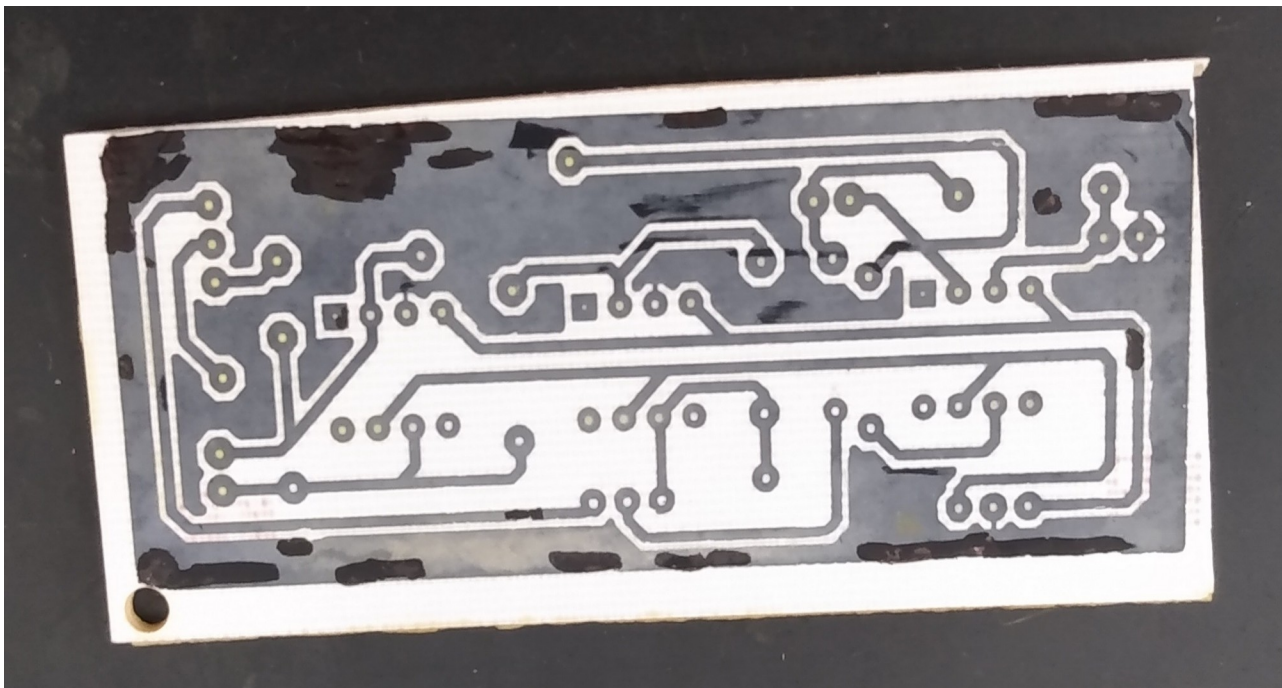


Figura 7 – Placa após ser retirada do ácido.

Foi escolhido utilizar vários pinos ao longo da placa (ligados a entrada, saída do primeiro bloco e saída do segundo bloco) para facilitar medidas e testes durante os experimentos.

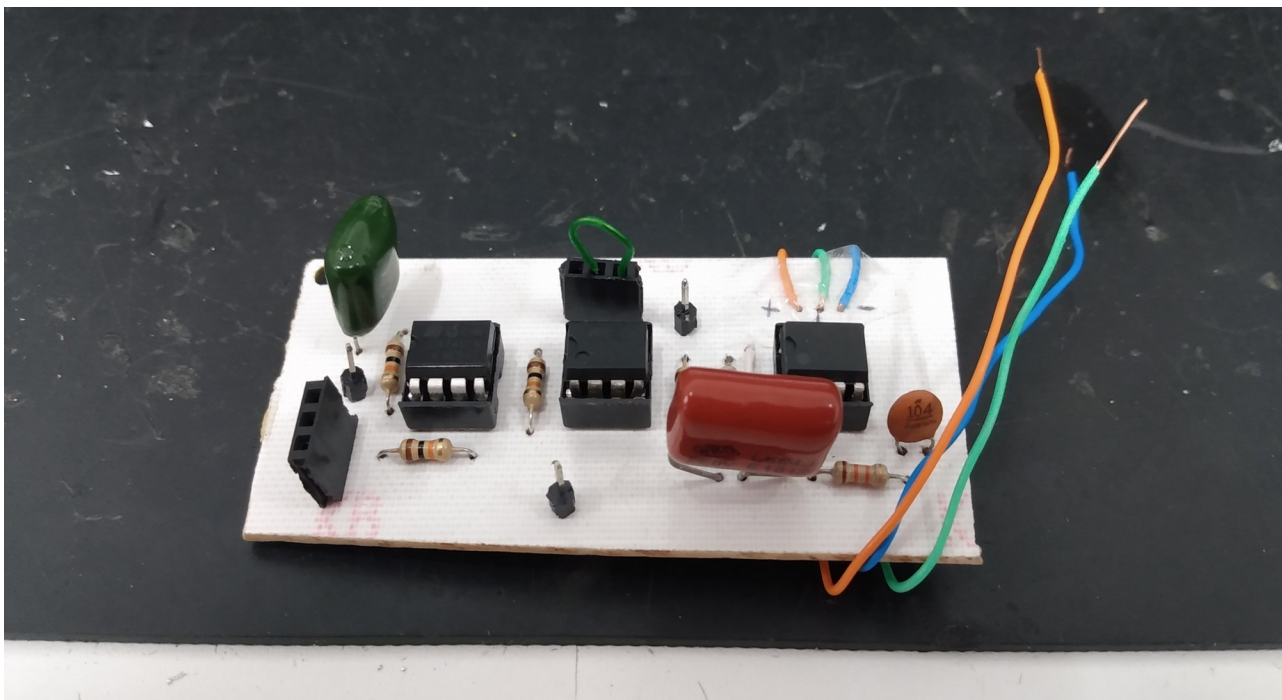


Figura 8 – Placa final.

No lugar do LM741 foi soldado soquetes para evitar perigos de danificar o CI com o calor do ferro de solda e também para não prendê-lo em uma placa assim podendo o reutilizar mais tarde, o capacitor de 150nF teve que ser trocado por um de 180nF devido a falta de disponibilidade de um capacitor deste valor.

### 3. Sistema Escolhido

Para realizar o controle foi escolhido a placa Arduino Due (representado na figura 9) devido a sua disponibilidade e capacidade de gerar sinais PWM e ler sinais analógicos assim tornando o controle do sistema possível.

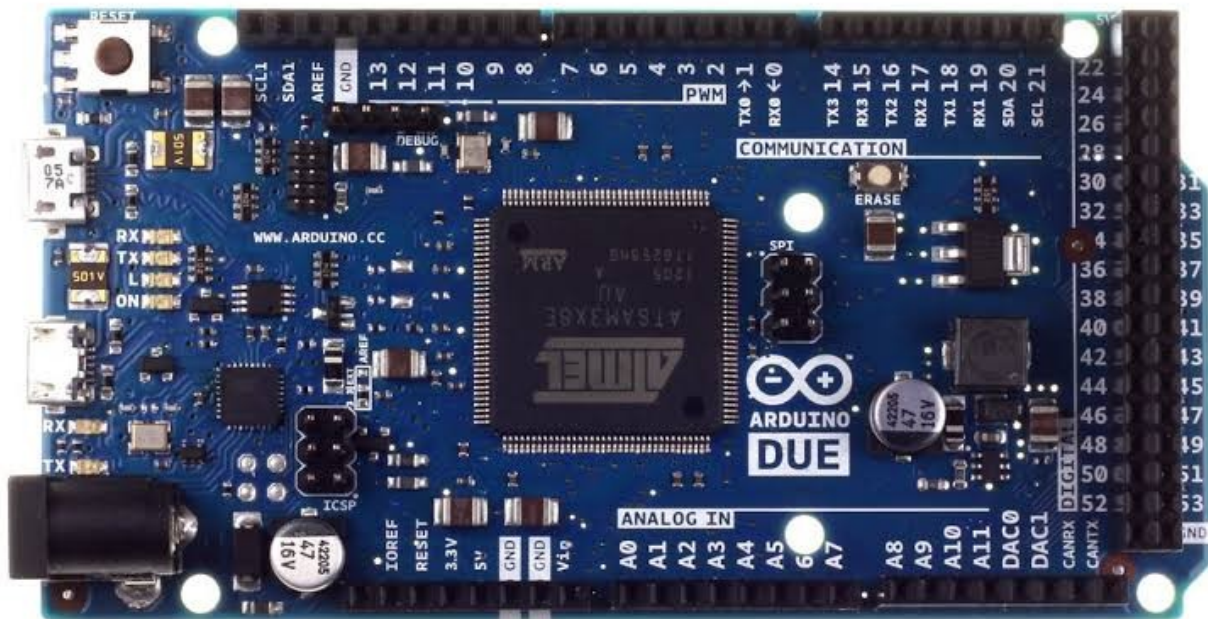


Figura 9 – Arduino Due

Esse Arduino possui as seguintes configurações:

- \* Microcontrolador AT91SAM3X8E;
- \* Tensão de operação: 3.3V;
- \* Pinos I/O Digitais: 54 (dos quais 12 possuem suporte para PWM);
- \* Pinos de entrada analógicos: 12
- \* Pinos de saída analógicas: 2
- \* Memória Flash: 512 KB;
- \* SRAM: 96 KB;
- \* Velocidade do clock: 84 MHz.



## 4. Análise do Sistema Teórico

Tendo em vista os componentes:

$R=10K\Omega$      $R1=68K\Omega$      $R2=13K\Omega$      $C=180nF$      $C1=100nF$      $C2=680nF$

É possível calcular o valor ideal dos blocos de primeira, segunda e terceira ordem e averiguar sua resposta a degrau em condições ideais com a utilização do programa MATLAB.

Para o sistema de primeira ordem os seguintes cálculos foram utilizados e com sua resposta ao degrau se analisou um tempo de subida de 3.95ms.

$$G1(s) := \frac{1}{R \cdot C \cdot s + 1}$$

$$G1(s) := \frac{1}{10 \cdot 10^3 \cdot 180 \cdot 10^{-9} \cdot s + 1}$$

$$G1(s) := \frac{1}{1,8 \cdot 10^{-3} \cdot s + 1}$$

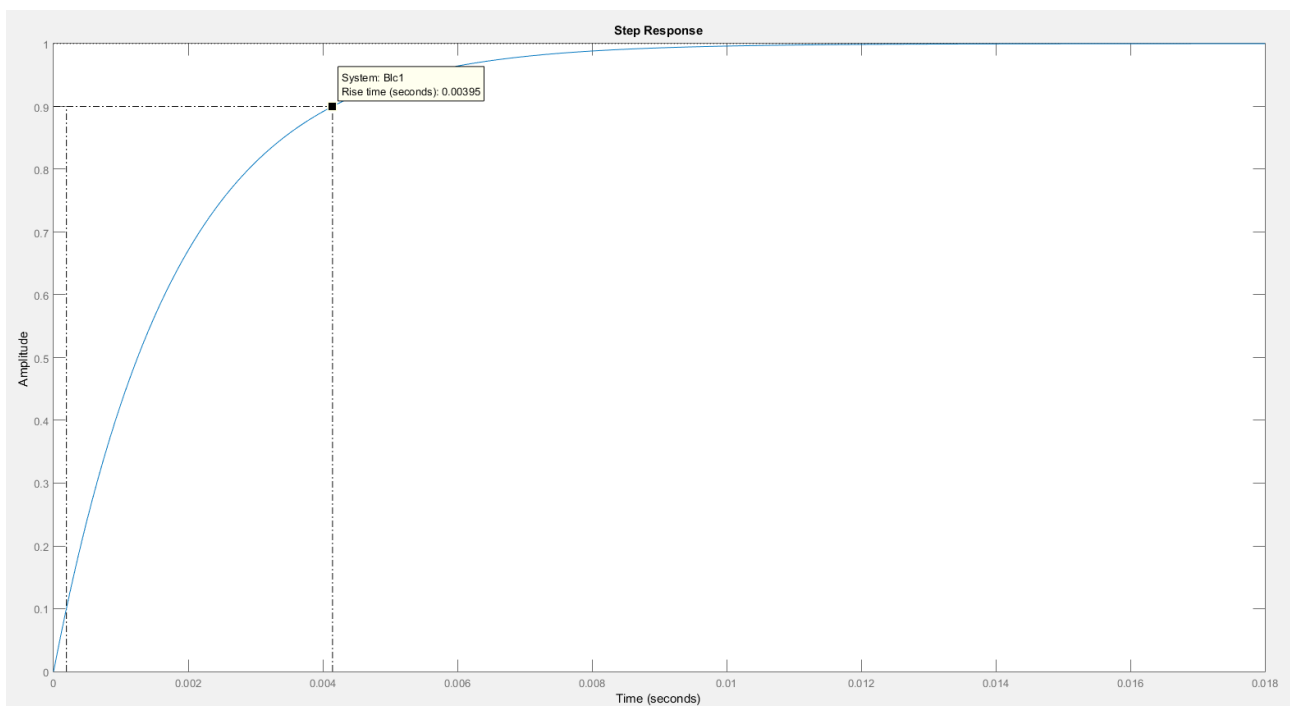


Figura 10 – Resposta ao Degrau do Sistema de Primeira Ordem.

Para o sistema de segunda ordem se realizaram os seguintes cálculos e através da resposta ao degrau em condições ideais se encontrou um sobressinal de 0.15V e um tempo de pico de 28.7ms.

$$G2(s) := \frac{1}{(C1 \cdot C2 \cdot R1 \cdot R2) \cdot s^2 + ((C1 \cdot R2) + (C1 \cdot R1)) \cdot s + 1}$$

$$Temp1 := (C1 \cdot C2 \cdot R1 \cdot R2)$$

$$Temp1 := (100 \cdot 10^{-9}) \cdot (680 \cdot 10^{-9}) \cdot (68 \cdot 10^3) \cdot (13 \cdot 10^3) = 6,0112 \cdot 10^{-5}$$

$$Temp2 := ((C1 \cdot R2) + (C1 \cdot R1))$$

$$Temp2 := ((100 \cdot 10^{-9}) \cdot (13 \cdot 10^3)) + ((100 \cdot 10^{-9}) \cdot (68 \cdot 10^3)) = 0,0081$$

$$G2(s) := \frac{1}{6,0112 \cdot 10^{-5} \cdot s^2 + 0,0081 \cdot s + 1}$$

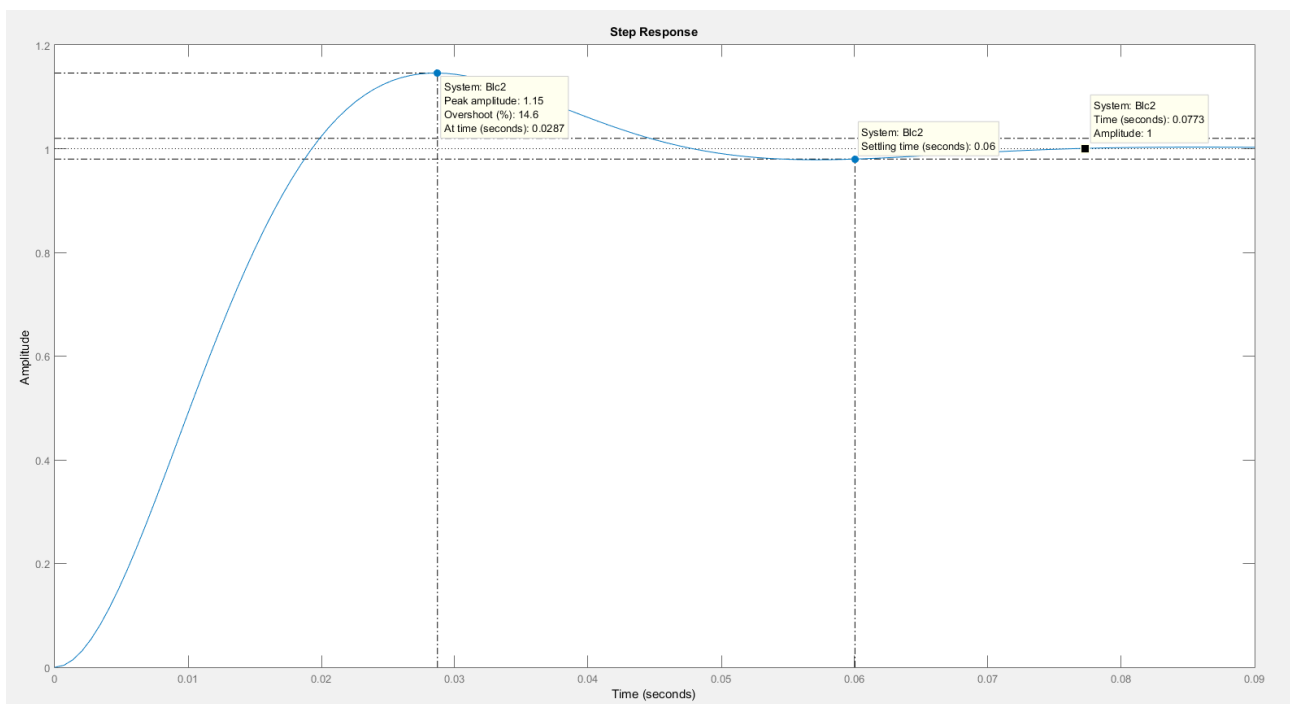


Figura 11 – Resposta ao Degrau do Sistema de Segunda Ordem.

Possuindo os sistemas de primeira e segunda ordem se adquiriu o de terceira ordem os multiplicando e adquirindo os seguintes resultados simulados de tempo de pico de 30.8ms, tempo de acomodação 61ms.

$$G3(s) := G1(s) \cdot G2(s)$$

$$G3(s) := \frac{9,24 \cdot 10^6}{s^3 + 690,3 \cdot s^2 + 9,15 \cdot 10^4 \cdot s + 9,242 \cdot 10^6}$$

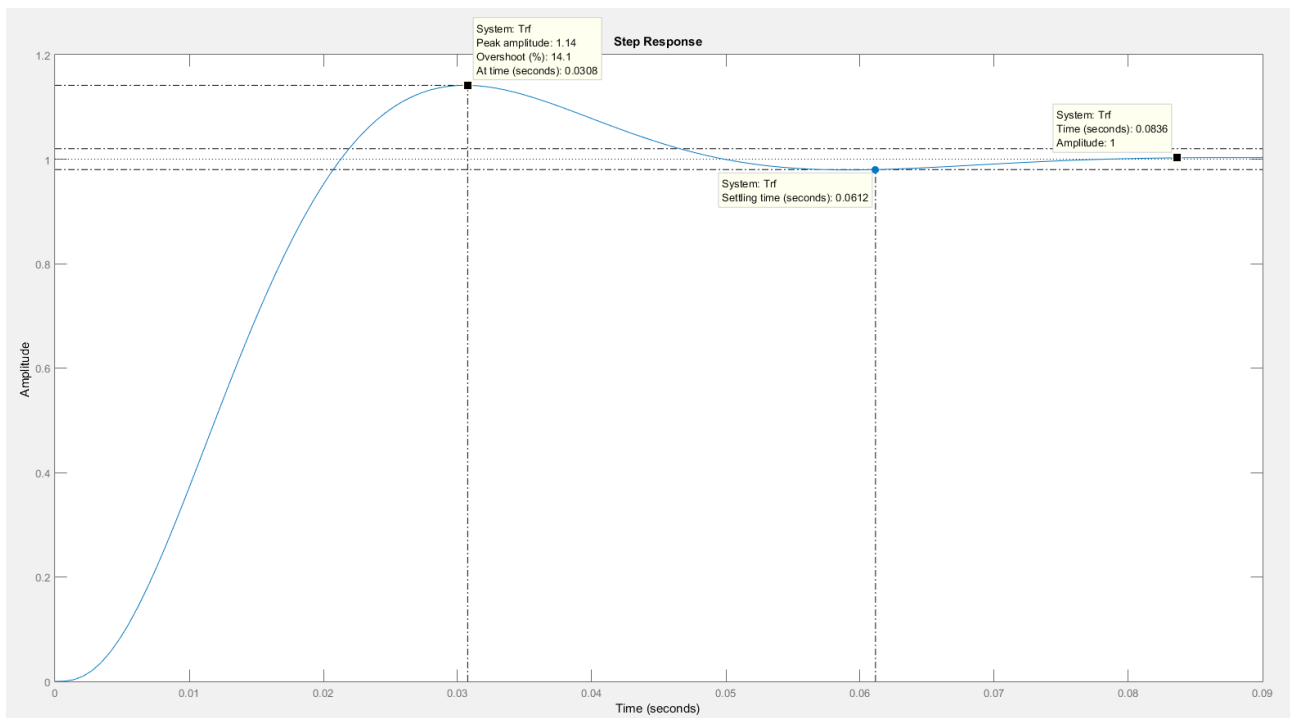


Figura 12 – Resposta ao Degrau do Sistema de Terceira Ordem.

## 5. Implementação Prática do Sistema

Para realizar os testes práticos primeiramente foi estabelecido qual sinal utilizar. De acordo com o que havia disponível foi utilizado um sinal quadrado gerado por um gerador de função (com sua frequência de 5 Hz) porém após adequadamente programar o Arduino Due com um PWM onde os sinais simulam um sinal de degrau que vai de 1 V até 1,5 V (como demonstrado teoricamente na figura 13).

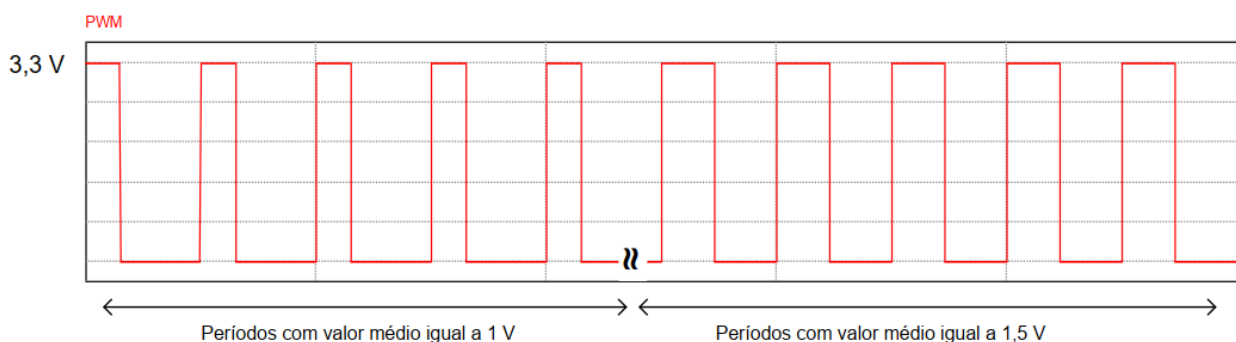


Figura 13 – Sinal de entrada.

Para o bloco de primeira ordem se obteve o sinal mostrado na figura 14.

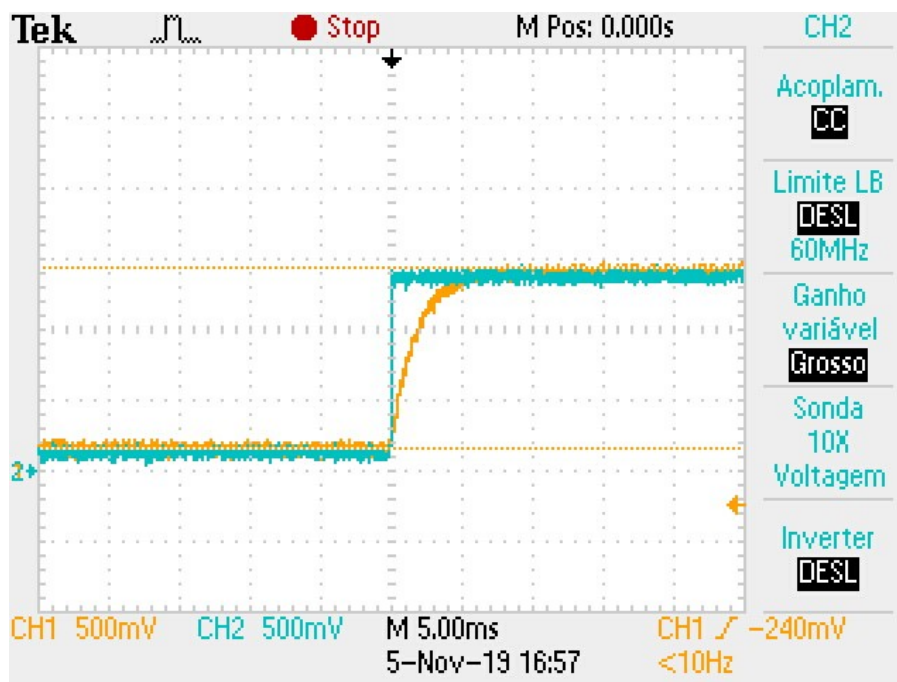


Figura 14 – Resultado do Bloco de Primeira Ordem.

Nesta etapa se mediu o valor de pico se adquirindo o valor máximo de 1,24 V

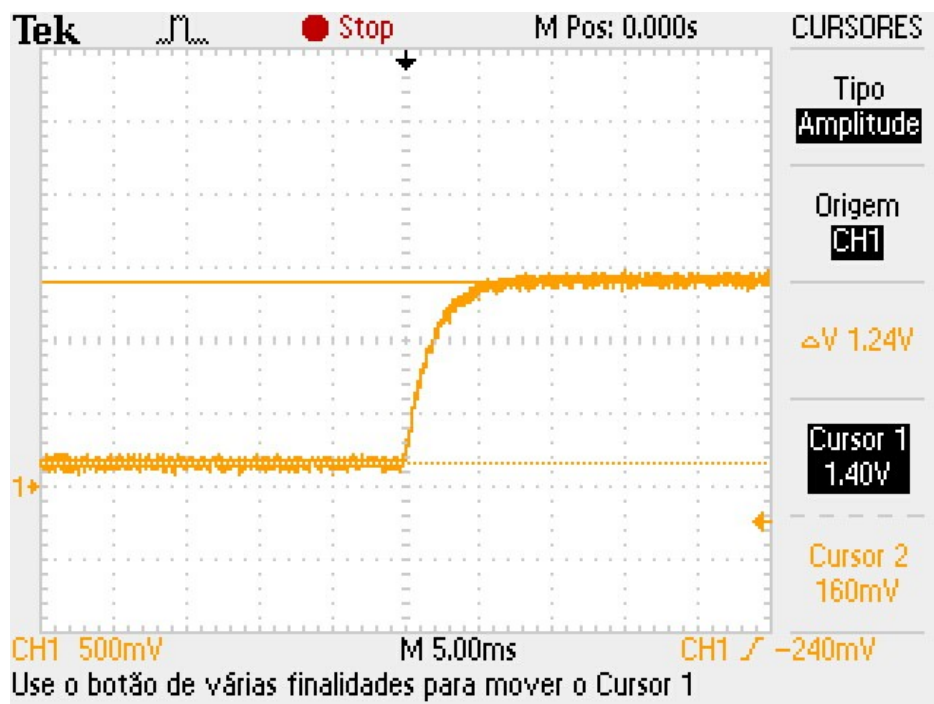


Figura 15 – Resultado do Bloco de Primeira Ordem.

Para se encontrar o valor de  $t_{s5\%}$  se multiplicou o valor de pico encontrado anteriormente e o multiplicou por 0.95 e então se localizou na melhor da capacidade do osciloscópio e então se adquiriu o tempo onde se encontra esta tensão e se adquiriu o tempo de subida ( $3 \cdot \tau$ ) de 5,2ms.

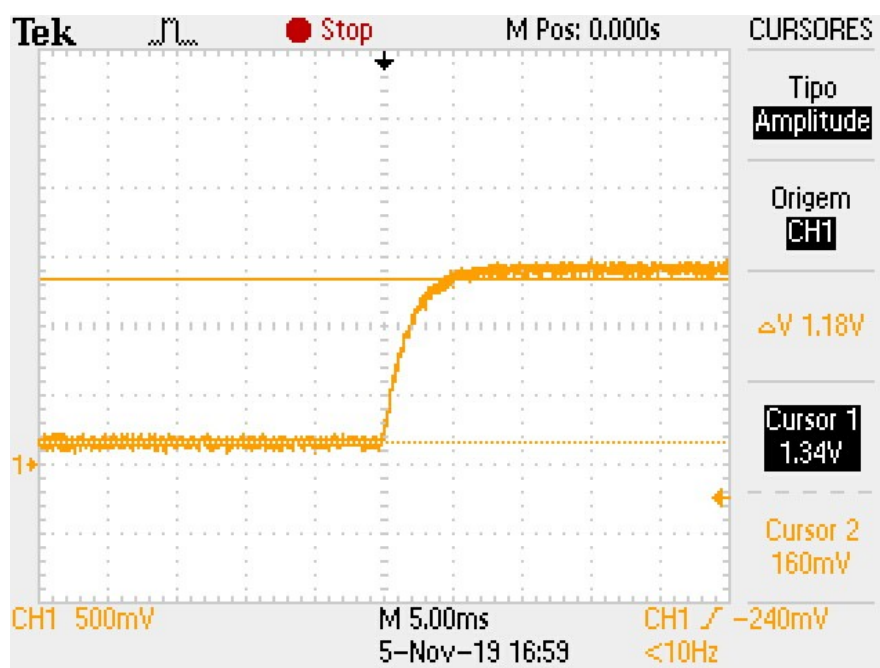


Figura 16 – Resultado do Bloco de Primeira Ordem.

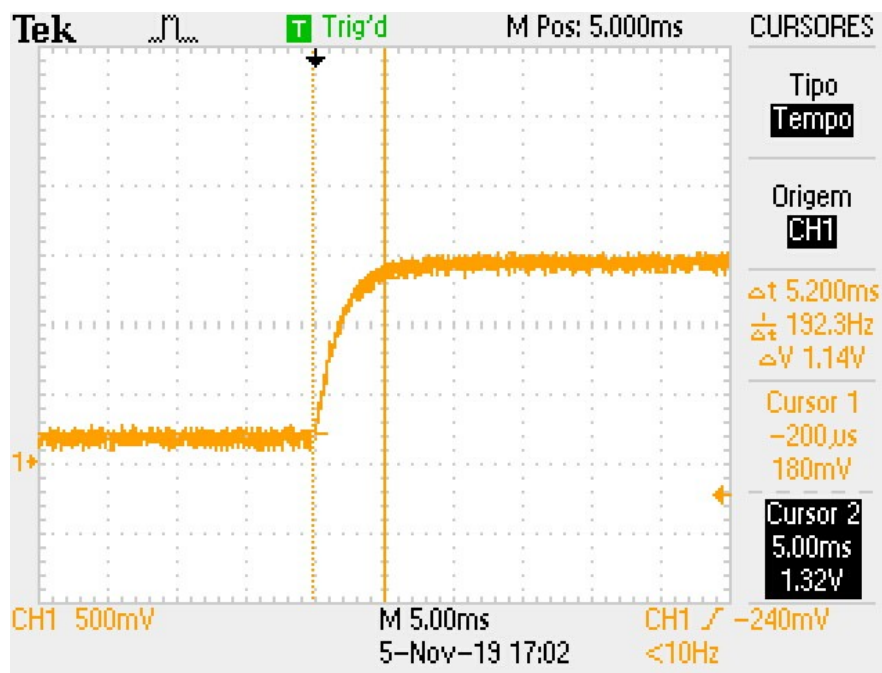


Figura 17 – Resultado do Bloco de Primeira Ordem.

Para se adquirir a equação  $G(s)$  do bloco se utilizou a equação para se adquirir Tau (que vem a ser o tempo de subida medido antes dividido por 3) e a equação de bloco de primeiro grau como mostrado abaixo

$$G1(s) := \frac{1}{\left( \frac{t_{subida}}{3} \right) \cdot s + 1}$$

$$G1(s) := \frac{1}{Tau \cdot s + 1}$$

$$G1(s) := \frac{1}{0,001647 \cdot s + 1}$$

Para o bloco de segunda ordem se adquiriu o sinal demonstrado na figura 18.

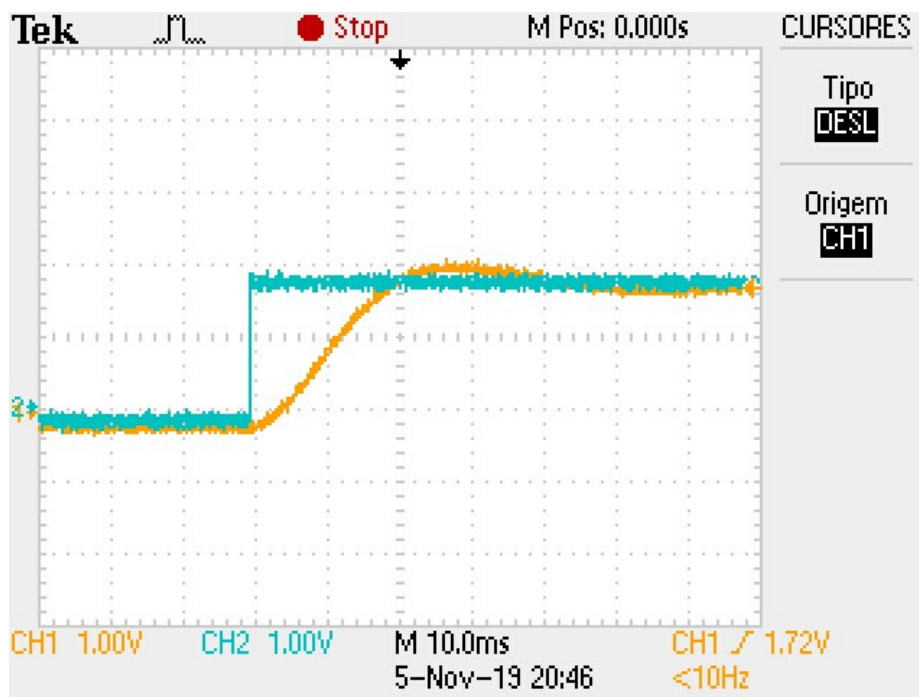


Figura 18 – Resultado do Bloco de Segunda Ordem.

Para a medição do sobressinal é necessário medir a diferença entre a tensão de pico e a tensão de acomodação e dividindo um pelo outro.

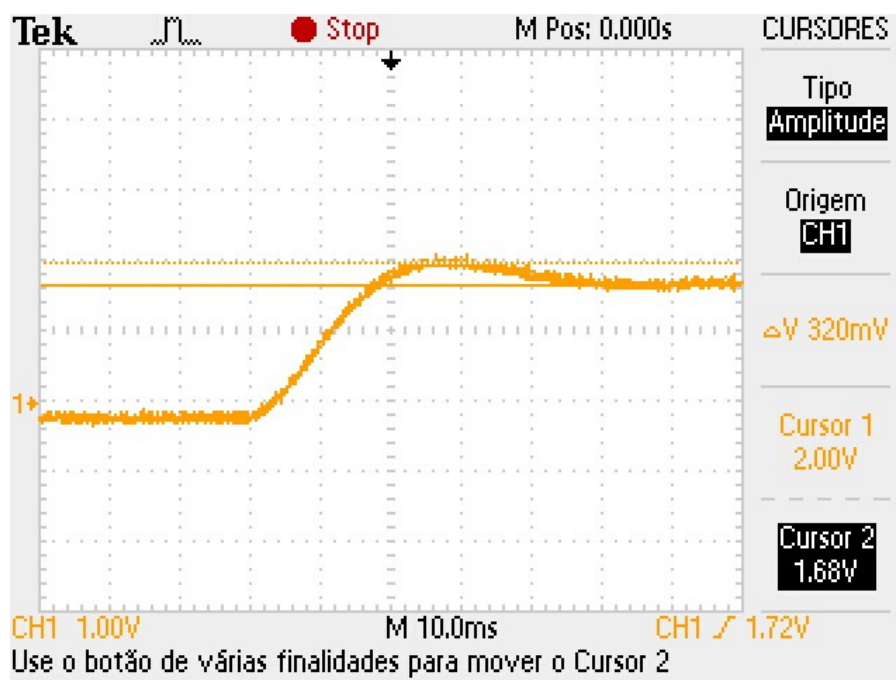


Figura 19 – Medição da tensão de pico.

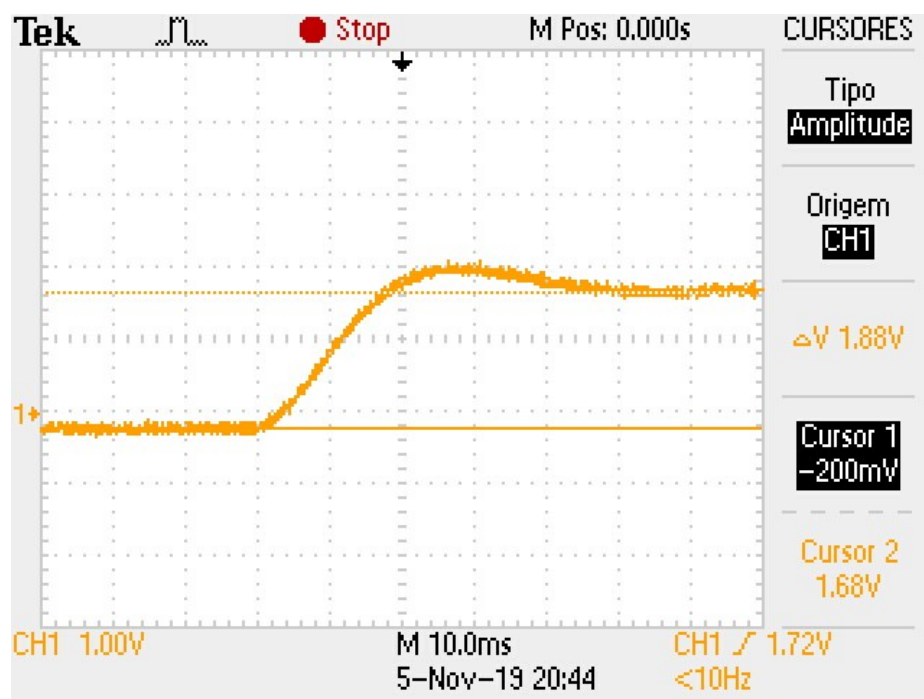


Figura 20 – Medição da tensão de pico.

$$Sobre\_Sinal := \frac{Tensão\_de\_Pico}{Tensão\_de\_Acomodação}$$

$$Sobre\_Sinal := \frac{0,32}{1,88} = 0,1702$$



Então se mediu o tempo até a tensão de pico se adquirindo 27.6ms e o tempo de acomodação 48ms (a medição desta etapa na figura 22 é apenas uma aproximação).

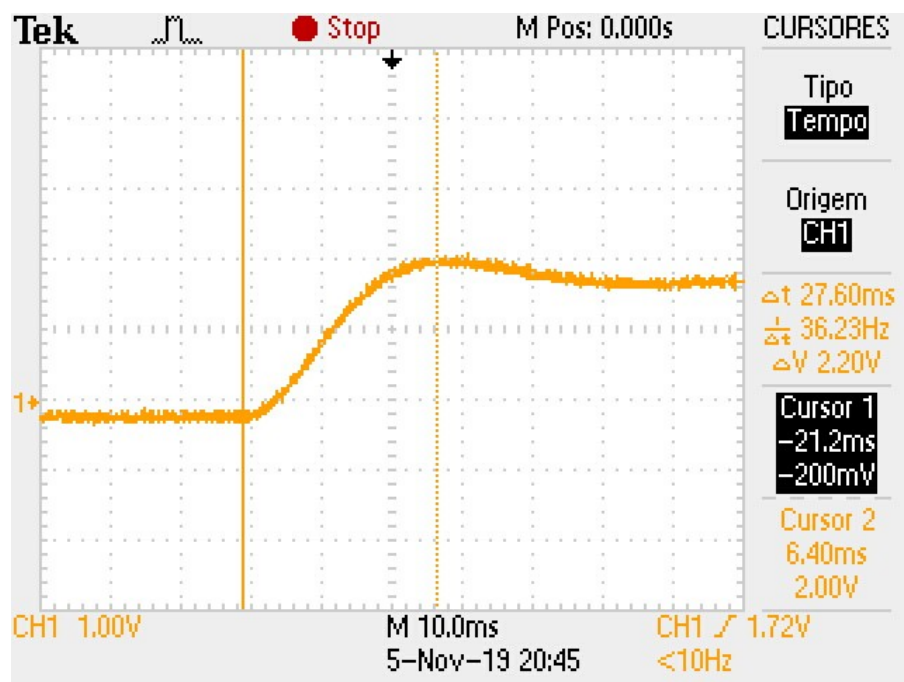


Figura 21 – Medição do tempo de pico.

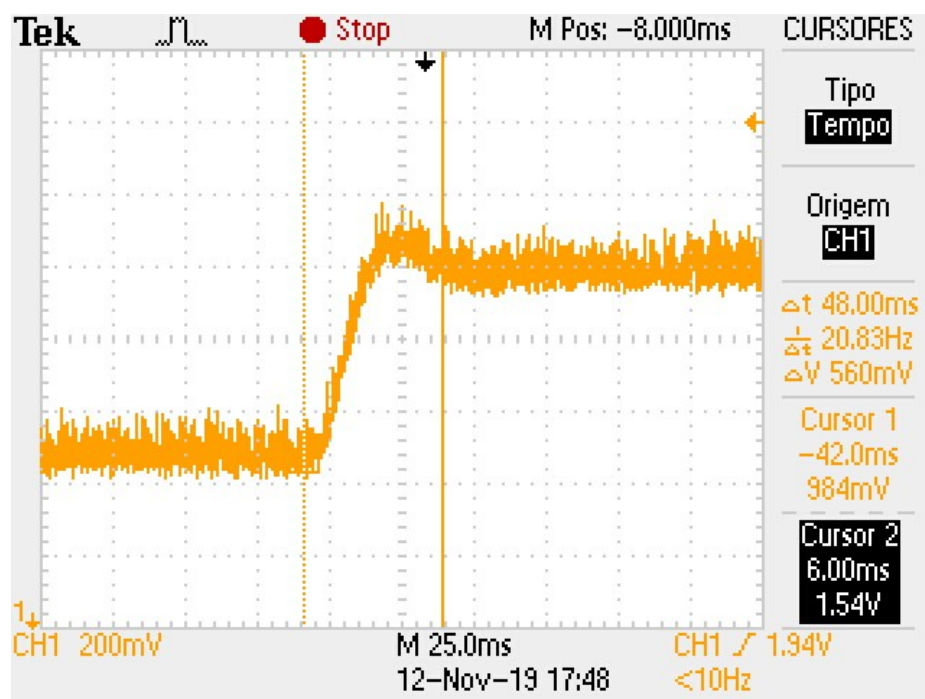


Figura 22 – Medição do tempo de acomodação.

Para calcular o bloco de segunda ordem primeiro se precisa calcular zeta e wn através das seguintes equações (fórmula do sobressinal e fórmula do tempo de pico).

$$Sobre\_Sinal := e^{\left(-\frac{\pi \cdot \zeta}{\sqrt{1-\zeta^2}}\right)}$$

$$\ln(Sobre\_Sinal) := -\frac{\pi \cdot \zeta}{\sqrt{1-\zeta^2}}$$

$$\frac{\ln(Sobre\_Sinal)}{\pi} := -\frac{\zeta}{\sqrt{1-\zeta^2}}$$

$$\left(\frac{\ln(Sobre\_Sinal)}{\pi}\right)^2 := \frac{\zeta^2}{1-\zeta^2}$$

Tendo:  

$$X := \left(\frac{\ln(Sobre\_Sinal)}{\pi}\right)^2$$

$$X \cdot (1 - \zeta^2) := \zeta^2$$

$$X - X \cdot \zeta^2 := \zeta^2$$

$$X := X \cdot \zeta^2 + \zeta^2$$

$$X := (1 + X) \cdot \zeta^2$$

$$\zeta := \sqrt{\frac{X}{1+X}}$$

Portanto:

$$\zeta := \sqrt{\frac{\left(\frac{\ln(Sobre\_Sinal)}{\pi}\right)^2}{1 + \left(\frac{\ln(Sobre\_Sinal)}{\pi}\right)^2}}$$

$$\zeta := \sqrt{\frac{\left(\frac{\ln(0,1702)}{\pi}\right)^2}{1 + \left(\frac{\ln(0,1702)}{\pi}\right)^2}} = 0,491$$

$$Tempo\_de\_Pico := \frac{\pi}{Wn \cdot \sqrt{1-\zeta^2}}$$

$$Wn := \frac{\pi}{Tempo\_de\_Pico \cdot \sqrt{1-\zeta^2}}$$

$$Wn := \frac{\pi}{Tempo\_de\_Pico \cdot \sqrt{1-\zeta^2}} = 129,7224$$

Então se realizou os cálculos para a função do bloco de segunda ordem ( $G_2(s)$ ) segundo a equação padrão para tal circuito como se vê abaixo:

$$G_2(s) := \frac{W_n^2}{s^2 + 2 \cdot \zeta \cdot W_n \cdot s + W_n^2}$$

$$G_2(s) := \frac{129,7224^2}{s^2 + 2 \cdot 0,491 \cdot 129,7224 \cdot s + 129,7224^2}$$

$$G_2(s) := \frac{16827,9011}{s^2 + 128,3105 \cdot s + 16827,9011}$$

A seguida se fez a medida da resposta ao degrau função de transferência de terceira ordem.

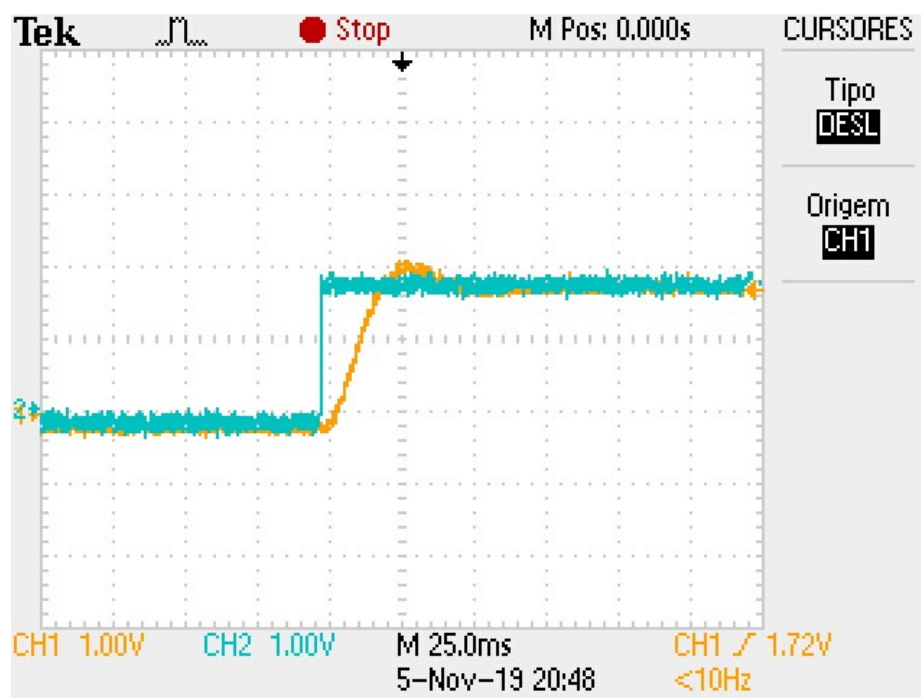


Figura 23 – Resposta ao Degrau do Circuito de Terceira Ordem.

E para o cálculo deste bloco é necessário multiplicar o bloco de primeira ordem com aquele de segunda ordem.

$$G3(s) := G1(s) \cdot G2(s)$$

$$G3(s) := \left( \frac{16827,9011}{s^2 \cdot 128,3105 \cdot s + 16827,9011} \right) \cdot \left( \frac{1}{0,001647 \cdot s + 1} \right)$$

$$G3(s) := \frac{1,022 \cdot 10^7}{s^3 + 737,7 \cdot s^2 + 9,419 \cdot 10^4 \cdot s + 1,022 \cdot 10^7}$$

## 6. Controlador digital

### 6a. Lugar das Raízes

O objetivo deste controlador é de reduzir no mínimo em 50% o valor de pico, tempo acomodação e tempo de valor de pico.

Para os cálculos do controlador se seguiu os valores encontrados a partir da implementação prática do sistema devido, por ser baseado em valores medidos e não calculados em condições ideais, inclui mais fatores e imperfeições.

Na figura 24 pode ser visto uma comparação da resposta ao degrau de ambas as equações adquirida através de programa no MATLAB.

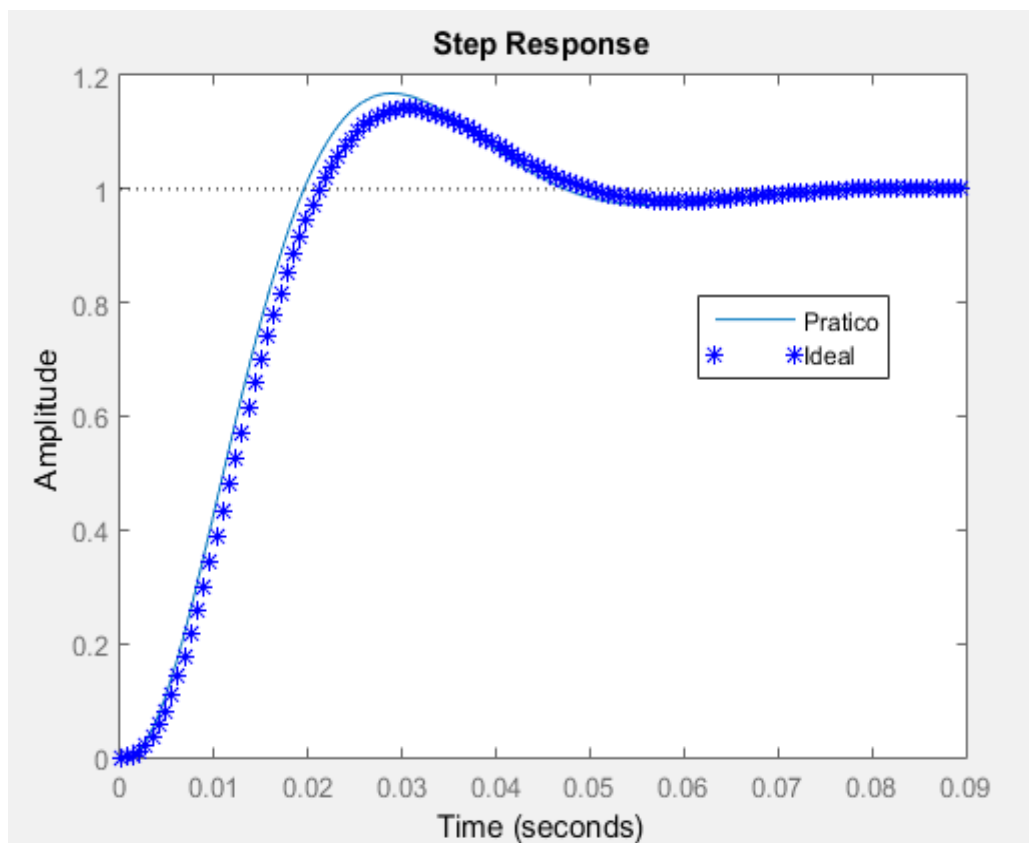


Figura 24 – Comparação da Resposta Ideal a Pratica

Para melhor analisar as características deste sistema se realizou a plotagem dos polos dominantes (como visto na figura 25) se adquirindo os valores de  $-63,7 \pm 113i$ .

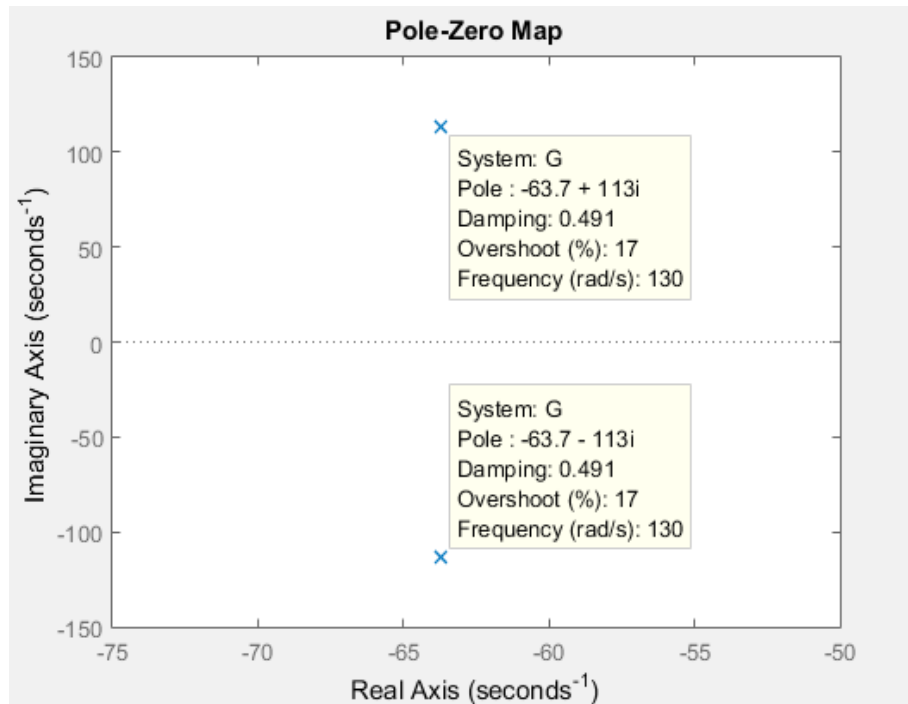


Figura 25 – Polos Dominantes

Para reduzir o sobressinal em 50% precisamos de um valor de  $\zeta > 0.491$ . O sistema Original possuía um tempo de acomodação de  $t_s = 3/(\zeta \cdot \omega_n) = 47.1$  ms. Para reduzirmos em 50% o tempo de acomodação do sistema original prático necessitamos de novos valores para  $\zeta$  e  $\omega_n$ . Foram utilizados os seguintes cálculos para os adquirir.

$$\zeta = 0,491 \quad \omega_n = 129,7224$$

$$\text{Sobre\_Sinal} := 0,1702$$

$$\text{Tempo\_de\_Pico} = 0,0278$$

$$\text{Sobre\_Sinal\_Ideal} := \frac{\text{Sobre\_Sinal}}{2} = 0,0851 \quad \text{Tempo\_de\_Pico\_Ideal} := \frac{\text{Tempo\_de\_Pico}}{2} = 0,0139$$

$$\text{Tempo\_Acomodação\_Ideal} := \frac{\left( \frac{3}{\omega_n \cdot \zeta} \right)}{2} = 0,0235 \quad \zeta\_Ideal := \sqrt{\frac{\left( \frac{\ln(\text{Sobre\_Sinal\_Ideal})}{\pi} \right)^2}{1 + \left( \frac{\ln(\text{Sobre\_Sinal\_Ideal})}{\pi} \right)^2}} = 0,6171$$

$$\omega_n\_Ideal := \frac{3}{\text{Tempo\_Acomodação\_Ideal} \cdot \zeta\_Ideal} = 206,4304$$

E a partir desses valores é possível, utilizando o critério de dez amostras por ciclo de oscilação, adquirir o período de amostragem (representado por T).

$$Fs := \frac{Wn\_Ideal \cdot \sqrt{1 - \zeta\_Ideal^2}}{2 \cdot \pi} \cdot 10 = 258,5187$$

$$T := \frac{1}{Fs} = 0,0039$$

A função de transferência discreta do sistema praticado pelo método *zero order hold* é expressa como:

$$G(z) := \frac{0,0657 \cdot z^2 + 0,1162 \cdot z + 0,008733}{z^3 - 1,439 \cdot z^2 + 0,6444 \cdot z - 0,01469}$$

Para que os polos dominantes estejam na região que satisfaz as duas condições foi calculado o ponto z da seguinte forma:

$$\zeta\_Ideal = 0,6171 \quad Wn\_Ideal = 206,4304$$

$$wd := Wn\_Ideal \cdot \sqrt{1 - \zeta\_Ideal^2} = 162,4321 \quad ws := \frac{2 \cdot \pi}{T} = 1624,3211$$

$$Módulo\_z := \exp\left(-\frac{2 \cdot \pi \cdot \zeta\_Ideal}{\sqrt{1 - \zeta\_Ideal^2}} \cdot \frac{wd}{ws}\right) = 0,6109 \quad Angulo\_z := 2 \cdot \pi \cdot \left(\frac{wd}{ws}\right) = 0,6283$$

$$z := Módulo\_z \cdot e^{j \cdot Angulo\_z}$$

Convertendo

$$z := 0,4933 + 0,3594 \cdot j$$

Resultando em um  $z=0,4933+0,3594i$  como pode ser visto abaixo na localização dos polos desejados (Figura 26).

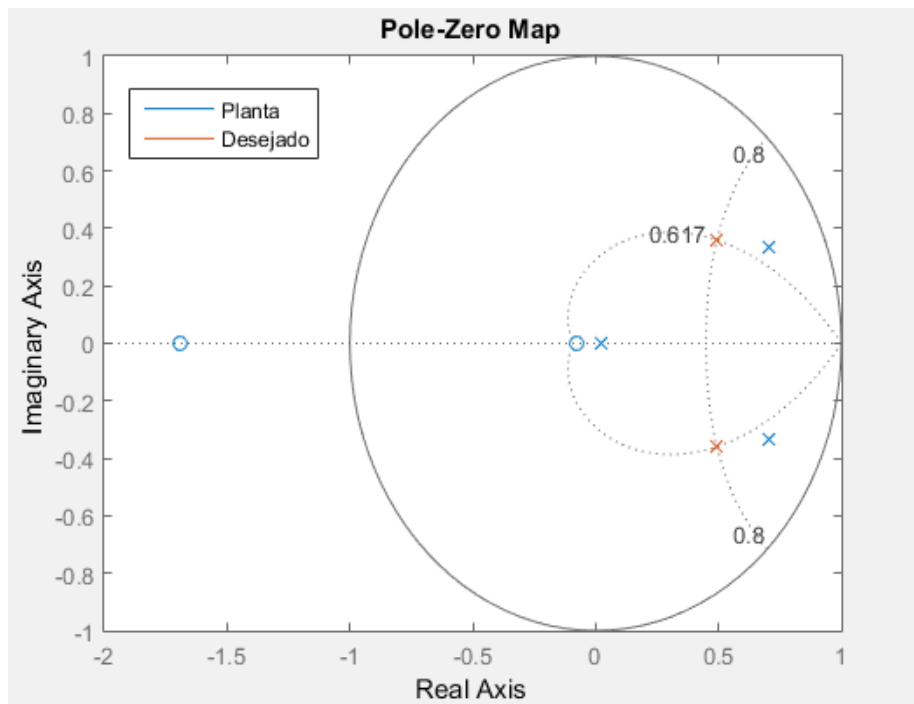


Figura 26 – Localização dos Polos Desejados

Para que o ponto  $z$  escolhido seja parte do lugar das raízes, este deve satisfazer a condição de ângulo e portanto a soma dos ângulos no ponto  $z$  deve ser igual à  $\pm 180$ .

Calculando a função de transferência discreta para o ponto  $z$  resulta em um ângulo complexo de  $16,709^\circ$ . Portanto o controlador deve fornecer  $163,2911^\circ$ .

O calculo pode ser visto a seguir (onde representados abaixo estão os polos e zeros da função de transferência discreta do sistema pratico pelo meto *zero order hold*).

$$Angulo\_1 := \frac{180}{\pi \cdot angulo(Polo1\_Gz(z) + z)} + \frac{180}{\pi \cdot angulo(Polo2\_Gz(z) + z)} + \frac{180}{\pi \cdot angulo(Polo3\_Gz(z) + z)}$$

$$Angulo\_2 := \frac{180}{\pi \cdot angulo(Zero1\_Gz(z) + z)} + \frac{180}{\pi \cdot angulo(Zero2\_Gz(z) + z)}$$

$$Angulo\_Final := Angulo\_1 + Angulo\_2$$

$$Angulo\_Final := 16,709$$

$$Angulo\_Desejado := 180 - Angulo\_Final = 163,2911$$



Para remover completamente o erro de regime permanente ao degrau o controlador tem que possuir um polo em 1. Assim se considera para o controlador anular os polos complexos da função de transferência discreta da planta com zeros complexos e assim encontrar um polo que satisfaça a condição de ângulo do lugar das raízes. Portanto a função de transferência pulsada do controlador tem a seguinte forma:

$$C(z) := K \cdot \left( \frac{z^2 - 1,415 \cdot z + 0,6103}{(z - 1) \cdot (z - \beta)} \right)$$

Encontrando-se o polo  $\beta$  se encontra o valor do polo de mesmo, neste caso sendo precisamente  $z=0.05561$  assim a função de transferência do controlador será descrita como:

$$C(z) := \frac{z^2 - 1,415 \cdot z + 0,6103}{z^2 - 1,056 \cdot z + 0,05561}$$

Para determinar o K se precisa satisfazer a condição de módulo do lugar das raízes ( $|Gz(z)C(z)|_{z=0,4933+0,3594i} = 1$ ) como pode ser visto a seguir:

$$z = 0,4933 + 0,3594 \cdot i$$

$$\left| \frac{z^2 - 1,415 \cdot z + 0,6103}{z^2 - 1,056 \cdot z + 0,05561} \cdot \frac{0,0657 \cdot z^2 + 0,1162 \cdot z + 0,008733}{z^3 - 1,439 \cdot z^2 + 0,6444 \cdot z - 0,01469} \right|_z = 1$$

$$K = 2.4134$$

Encontramos  $K = 2,4134$ . Assim a equação de transferência discreta do controlador será:

$$\text{Controlador} := \frac{2,414 \cdot z^2 - 3,415 \cdot z + 1,473}{z^2 - 0,9117 \cdot z + 0,08834}$$

Se aplicando o controlador na função e fazendo a função de transferência de malha fechada e se aplicando o degrau na equação se obtêm resultados onde a sobretenção é de 0.08, tempo de acomodação de 30ms e tempo de pico de 19ms o que apresenta uma redução teórica que se aproxima daquelas desejadas de 50%.

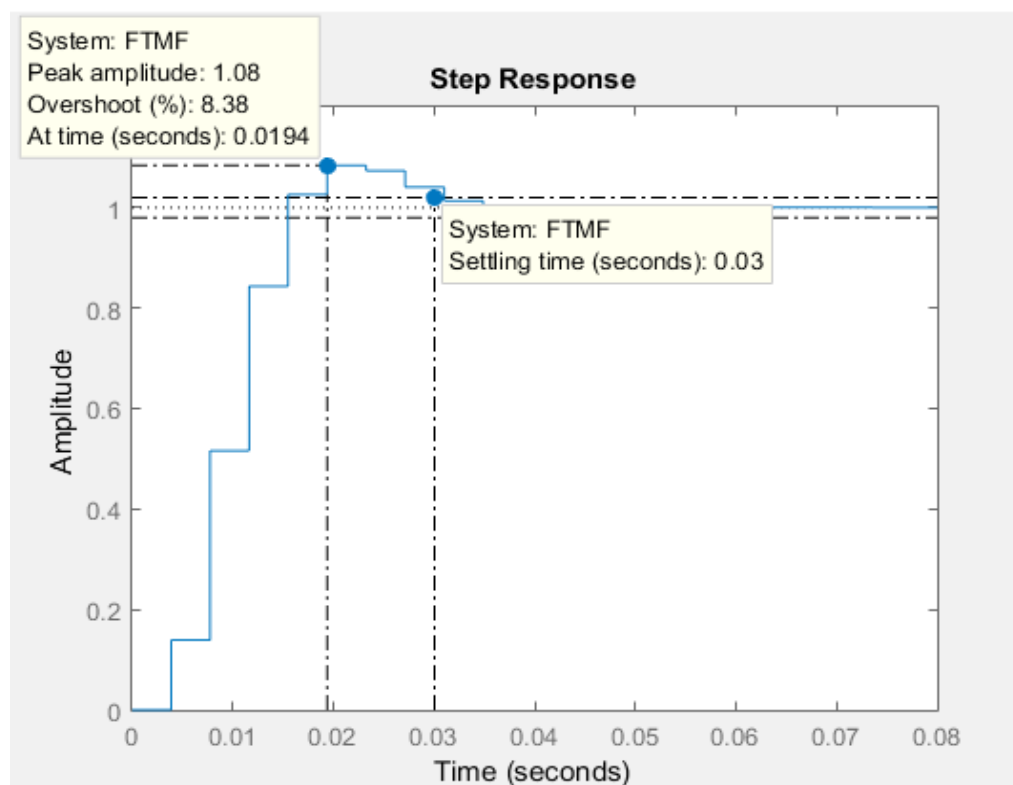


Figura 27 – Resposta Simulada do FTMF

Fazendo um teste por meio do teorema de valor final, utilizando a função de transferência de malha aberta (FTMA) foi verificado um erro a entrada ao degrau unitário.

$$K_p = \lim_{z \rightarrow 1} FTMA(z) = \lim_{z \rightarrow 1} G(z) = 1$$

$$e(\infty) = \frac{1}{1 + K_p} = 0.5$$

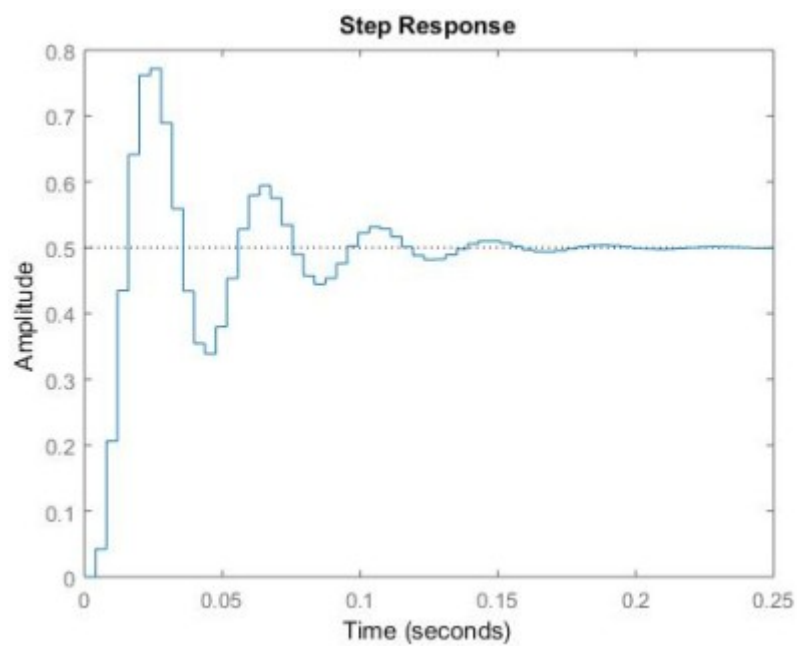


Figura 28 – Erro ao Degrau Unitário.

Com relação aos polos de malha fechada se realizou uma simulação no MATLAB e se adquiriu polos dominantes de  $0.493 \pm 0.359i$  o que se aproxima muito do polo escolhido  $z = 0,4933 + 0,3594i$  assim concluindo que teoricamente se alcançou o resultado desejado.

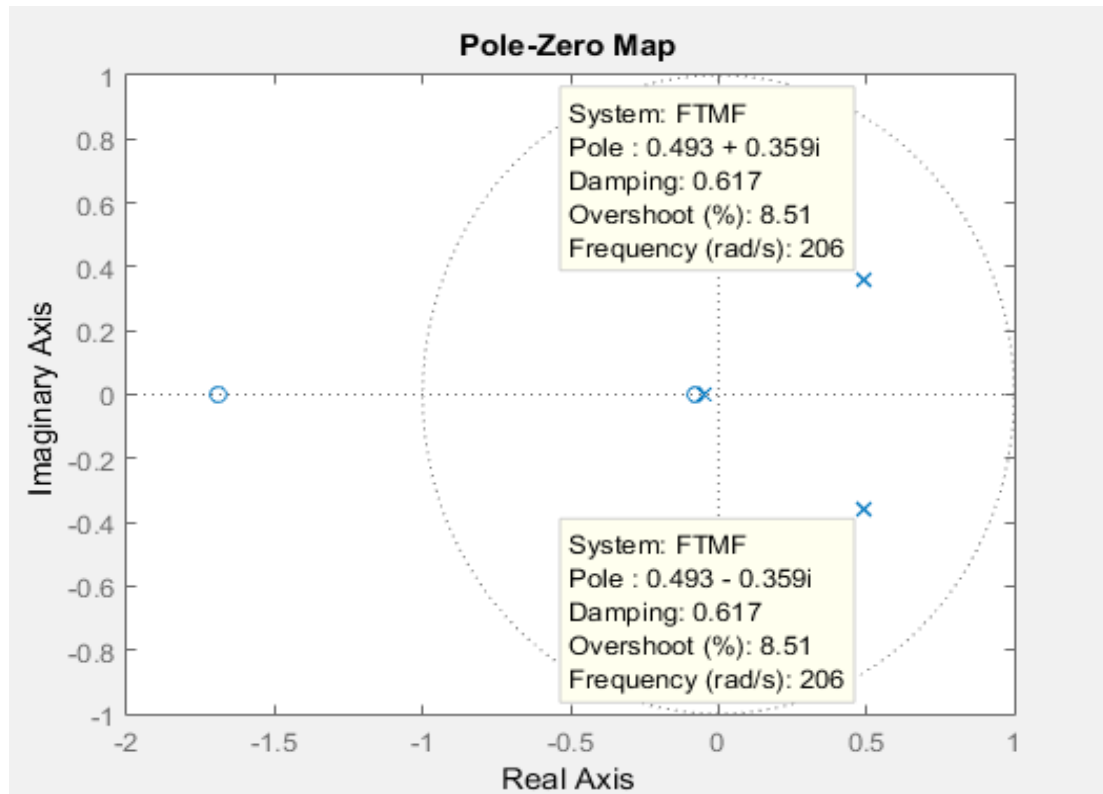


Figura 29 – Polos da FTMF

Apos todas essas etapas se montou o seguinte fluxograma representando o sistema implementado com a inclusão do controlador onde fisicamente o controlador e a realimentação serão feitos utilizando o Arduino Due.

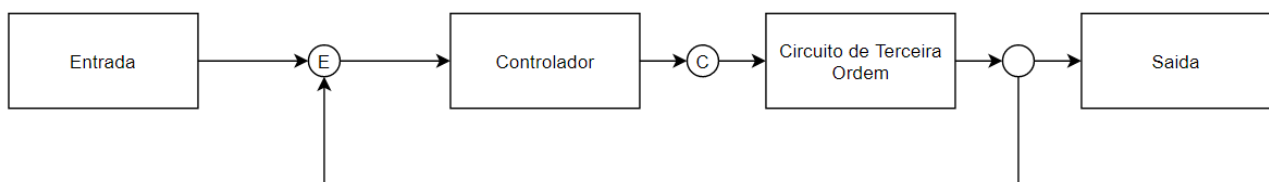


Figura 30 – Fluxograma do Sistema

Para se aplicar o controlador no microcontrolador foi necessário converter a equação do controlador em uma equação recursiva.

$$\frac{C(z)}{E(z)} := \frac{2,414 \cdot z^2 - 3,415 \cdot z + 1,473}{z^2 - 0,9117 \cdot z + 0,08834}$$

$$C(z) \cdot (z^2 - 0,9117 \cdot z + 0,08834) := E(z) \cdot (2,414 \cdot z^2 - 3,415 \cdot z + 1,473)$$

$$C(K+2) := 2,414 \cdot E(K+2) - 3,415 \cdot E(K+1) + 1,473 \cdot E(K) + 0,9117 \cdot C(K+1) - 0,08834 \cdot C(K)$$

Resultando na seguinte equação recursiva que foi implementada no código implementado no anexo II:

$$C(k) := 2,414 \cdot E(k) - 3,415 \cdot E(k-1) + 1,473 \cdot E(k-2) + 0,9117 \cdot C(k-1) - 0,08834 \cdot C(k-2)$$

Nas figuras a seguir pode-se ver os resultados práticos da implementação do controlador.

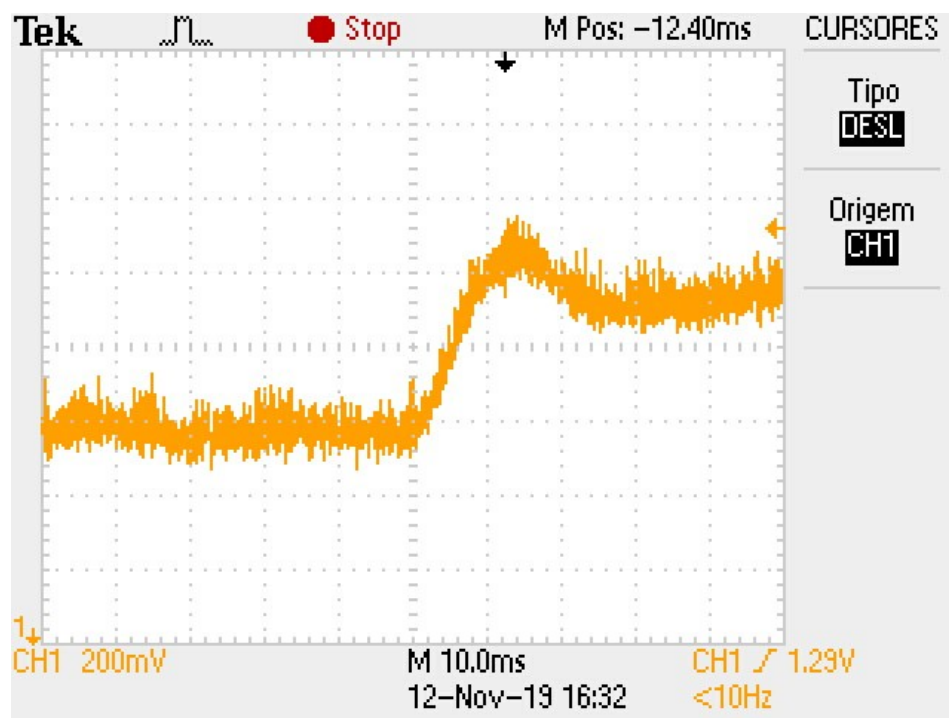


Figura 31 – Sinal Resultante

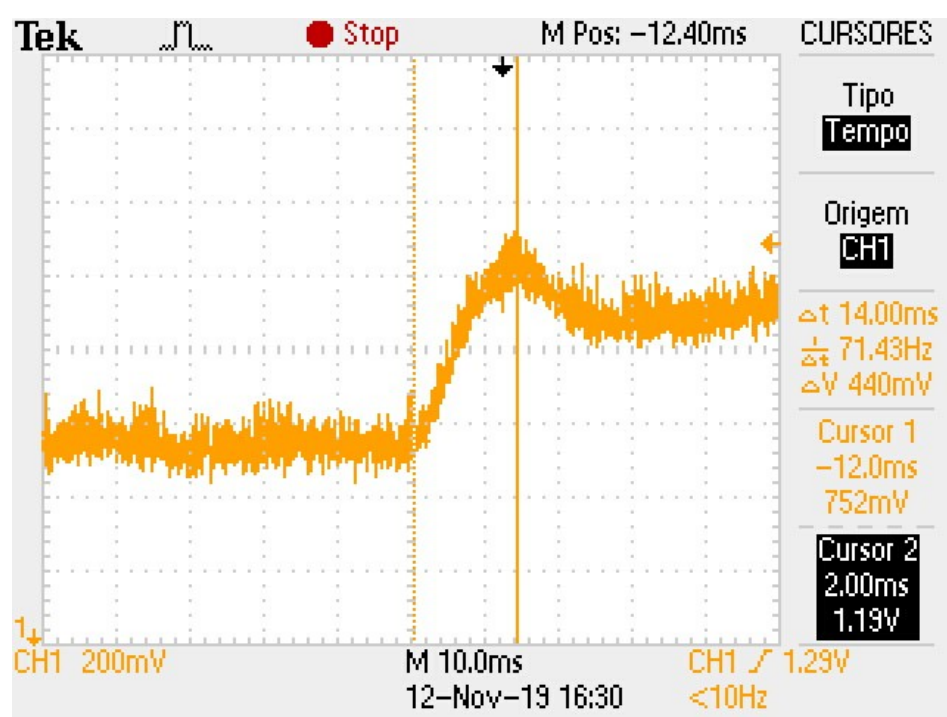


Figura 32 – Sinal Resultante

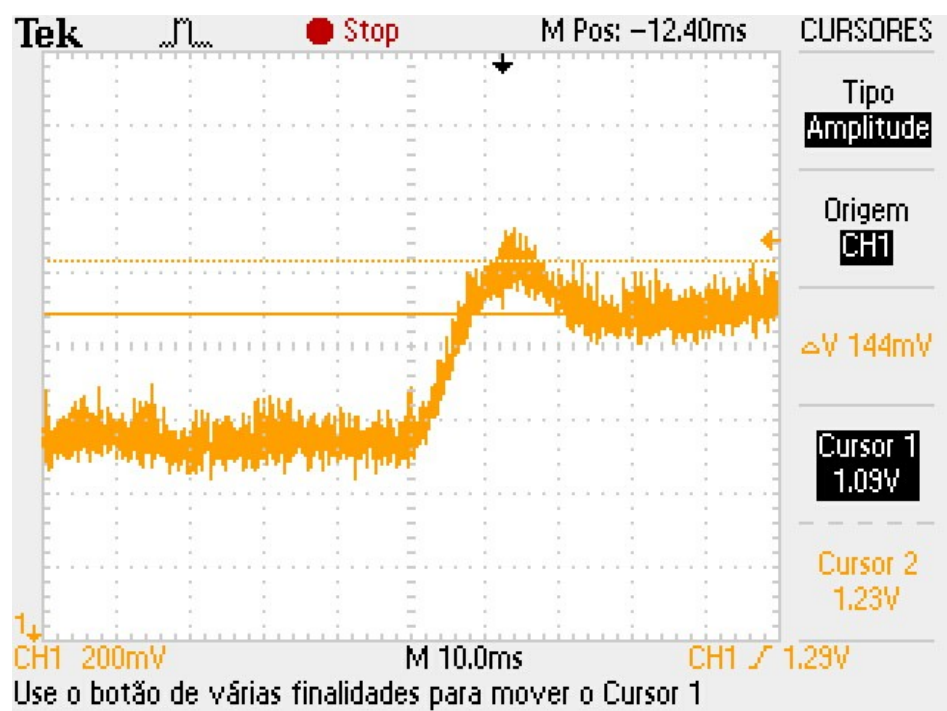


Figura 33 – Sinal Resultante

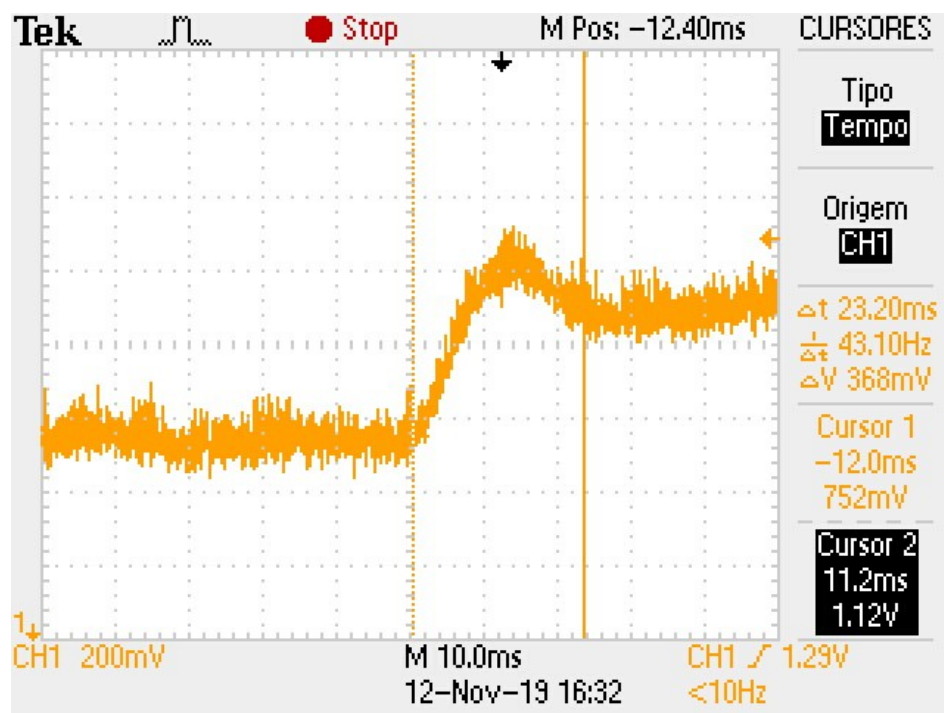


Figura 34 – Sinal Resultante

Analisando os valores pode-se concluir que agora o sobressinal foi 144mV (comparado com 320mV anteriormente medido), tempo de pico de 14ms (onde antes era 28ms), tempo de acomodação de 23,2ms (em relação ao 48ms originais) e erro quase nulo de regime permanente com relação aos referenciais de 1V e 1.5V.

Portanto representando:

- Sobressinal: diminuição de 52%
- Tempo de pico: diminuição de 50%
- Tempo de acomodação: diminuição de 51%

A diminuição de 50% de diminuição desejada foi alcançada e o erro pode não ter sido nulo mas muito próximo (não foi nulo apenas devido a 0,09 V) devido a imperfeições do sistema e interferências (a utilização de fios longos para a ligação do Arduino Due com a placa foi percebido que o sistema ficou muito vulnerável a ruídos pois funcionam como “antenas”).

## 7. Conclusões

Foi verificada a possibilidade de implementar sistemas de controles digitais para plantas analógicas. Apesar da limitação de uma taxa de amostragem finita esses sistemas podem agir em tempo contínuo para o controle e melhor ajustes do sinal.

O maior problema encontrado neste projeto foi aquele com o dimensionamento do controlador e algumas dificuldades com relação a montagem da placa porém todos esses problemas foram superados.

Foi possível projetar e simular os controladores com sucesso tendo seus resultados práticos muito próximos dos teóricos esperados e os problemas encontrados (como interferência e na área de programação) serão levados em conta em futuros projetos para não se repetirem e causarem mais atrasos.



## 8.ANEXO I – CÓDIGO MATLAB

```
clear;
clc;
%% Especifications
s = tf('s');

% Planta Teórica
% Componentes
R = 10*(10^3);    R1 = 68*(10^3);    R2 = 13*(10^3);
C = 180*(10^-9); C1 = 100*(10^-9);    C2 = 680*(10^-9);

% Função de Transferencia Simulada
num = 1;
den1 = [(R*C) 1];
den2 = [(C1*C2*R1*R2) ((C1*R2)+(C1*R1)) 1];

Blc1 = tf(num, den1);    % Bloco 1 (Primeira Ordem)
Blc2 = tf(num, den2);    % Bloco 2 (Segunda Ordem)

Trf = Blc1*Blc2;          % Terceira Ordem
Trf = minreal(Trf);

% Especificações da planta prática
ts1 = 5.2e-3;    %Tempo de subida de primeira ordem
Mp = 0.32/1.88;    %Sobresinal
tp = 27.8e-3;    %Tempo de pico (segunda ordem)

% Bloco primeira ordem
tau = (0.95*ts1)/3;
G1 = 1/(tau*s+1);

% Bloco segunda ordem
zeta = fzero(@(x) ((log(Mp)/pi) + (x/sqrt(1-x^2))), 0.5);
wn = pi/(tp*sqrt(1-zeta^2));
G2 = (wn^2)/(s^2 + 2*zeta*wn*s + wn^2);
ts2 = 3/(zeta*wn);

% Planta Prática Contínua
G = G1*G2;
G = minreal(G);

%% Controlador
% Requisitos
Mp = Mp/2;
ts = ts2/2;
zeta = fzero(@(x) ((log(Mp)/pi) + (x/sqrt(1-x^2))), 0.5);
wn = 3/(ts*zeta);
Fs_required = (10*wn*sqrt(1-zeta^2))/(2*pi);
Fs = 258;
Ts = 1/Fs;
z = tf('z', Ts);

wd = wn*sqrt(1-(zeta^2));
ws = (2*pi)/Ts;

mod_z = exp((-2*pi*zeta)/(sqrt(1-(zeta^2))))*(wd/ws);
ang_g = 2*pi*(wd/ws);

[zx, zy] = pol2cart(ang_g, mod_z);
```

```

z1 = complex(zx, zy);

% Planta Prática Discreta
Gz = c2d(G,Ts);

angle_required = -(angle(evalfr(Gz,z1)) - pi);
zeros_planta = pole(Gz);
angulo_polo = angle(z1 - zeros_planta(1)) + angle(z1 - zeros_planta(2)) -
angle(z1 - 1) - angle_required;
polo_desejado = real(z1) - imag(z1)/tan(angulo_polo);
Cz_semK = ((z-zeros_planta(1))*(z-zeros_planta(2)))/((z-polo_desejado)*(z-1));
K = abs(1/(evalfr(minreal(Cz_semK*Gz), z1)));
Cz = K*Cz_semK;
FTMA = minreal(Cz*Gz);
FTMF = minreal(feedback(FTMA,1));

```

## 9.ANEXO II – CÓDIGO ARDUINO DUE

```
#include <DueTimer.h>
```

```
#define F_Amostragem 258
```

```
#define N_Fatores      5
```

```
#define PWM_1V        4095/3.3
```

```
#define PWM_1v5        (4095/3.3)*1.5
```

```
int Pino_Entrada = A1;
```

```
int Pino_Saida = 8;
```

```
int Pino_PWM = 9;
```

```
int value = 0;
```

```
volatile float coefs[N_Fatores] = {2.414, -3.415, 1.473, 0.9117, -0.08834};
```

```
volatile float r1 = 0, e = 0, e1 = 0, e2 = 0, u = 0, u1 = 0, u2 = 0, sample = 0;
```

```
void Controle(){
```

```
    sample = analogRead(Pino_Entrada)*(3.3/4095);
```

```
    e = r1 - sample;
```

```
    u = coefs[0]*e + coefs[1]*e1 + coefs[2]*e2 + coefs[3]*u1 + coefs[4]*u2;
```

```
    analogWrite(Pino_Saida, u*(4095/3.3)); // Ação de comando
```

```
    u2 = u1;
```

```
    u1 = u;
```

```
    e2 = e1;
```

```
    e1 = e;}
```

```
void PWM_Classic(){
  switch(value){
    case 0:
      analogWrite(Pino_PWM, PWM_1V); // 1V tensão média
      r1 = 1.0;
      value = 1;
      break;
    case 1:
      analogWrite(Pino_PWM, PWM_1v5); // 1.5V tensão média
      r1 = 1.5;
      value = 0;
      break;}}

void setup() {
  pinMode(Pino_Saida, OUTPUT);
  pinMode(Pino_PWM, OUTPUT);
  Timer0.attachInterrupt(PWM_Classic).setFrequency(5).start();
  Timer1.attachInterrupt(Controle).setFrequency(F_Amostragem).start();}

void loop() {}
```