

UNIVERSIDADE FEDERAL DE LAVRAS
DEPARTAMENTO DE ENGENHARIA - DEG
LABORATÓRIO INTEGRADOR - GAT125

Professor: Daniel Furtado Leite

Modelagem e controle de temperatura de uma laminadora caseira

Bruna de Paula Faria - 201520782

Nicolas Iulius Lussari Brison - 201510916

Vitor Barbosa Nunes Costa - 201610290

6 de dezembro de 2019

Lavras – MG

Sumário

1. Introdução.....	2
1.1 Objetivo.....	2
1.2 Fluxograma Geral.....	3
1.3 Sobre Hardware, Software, Comunicação.....	4
2. Modelagem.....	5
2.1 Calibração do sensor	6
2.2 Aquisição de dados	7
2.3 Modelos Obtidos (Matlab/ident).....	8
2.4 Simulações em Malha Aberta	10
3. Controle.....	11
3.1 Metodologia para Escolha de Parâmetros PID.....	12
3.2 Projeto de Controle PID via Ziegler- Nichols.....	13
3.3 Projeto de Controle PID via Cohen- Coon.....	15
3.4 Projeto de Controle PID via Lugar das Raízes.....	16
3.5 Comparação da Malha Fechada em Simulação Matlab e Planta Física.....	17
4. Conclusão.....	25
4.1. Síntese do Trabalho.....	25
4.2. Dificuldades.....	25
5. Referências.....	25

1. Introdução

Um grande desafio das pessoas que tem como hobbie a eletrônica ou prototipagem é a impressão de placas de circuito impresso. A dificuldade não se encontra na modelagem da placa, mas sim na impressão dos desenhos que devem ser transferidos para a placa, a fim de corrobô-la de maneira correta. Antigamente, eram feitos desenhos à mão na placa, porém os resultados podiam ficar imprecisos muitas vezes.

Com a evolução e barateamento dos componentes e máquinas, hoje é possível fazer de forma mais fácil e eficiente a transferência do desenho para a placa: por meio de transferência térmica. Muitos amadores da eletrônicas utilizam ferros de passar roupa ou outras formas semelhantes para transferir a tinta do toner impresso no papel glossy, no entanto isso caracteriza muitas vezes um problemas, afinal a temperatura varia muito devido a ser um processo manual, aplicação de pressão variável no papel, dentre outras variáveis.

Uma solução muito viável para os hobbistas é utilizar uma laminadora, que é uma máquina com um sistema de dois cilindros rotativos, no qual um deles atinge a temperatura necessária para fundir o toner que foi despejado no papel glossy e transferi-lo para a placa de circuito impresso (PCI). Desta forma, se torna muito prática, rápida e barata a fabricação de PCI's.

1.1. Objetivos

O objetivo deste trabalho é adquirir dados suficientes para obter um modelo matemático do sistema térmico, e utilizá-lo para controlar a temperatura da laminadora caseira, feita pelo grupo, testando e validando os modelos e o projeto do controlador.

1.2. Fluxograma Geral

Pode-se ver pela figura 1.1 a seguir o fluxograma geral do sistema. De maneira simplificada e visual ele mostra como os sinais são transmitidos dentro do sistema, apresenta também os principais elementos presentes nesta bancada.

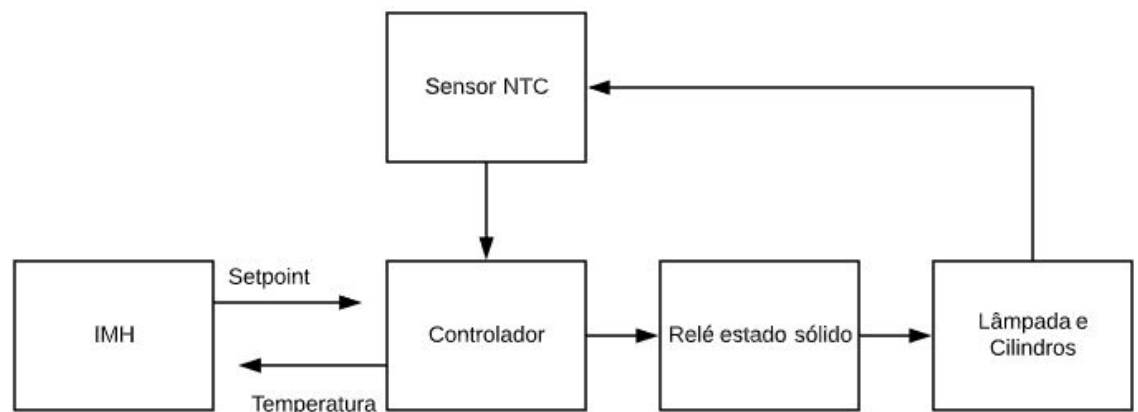


Figura 1.1 - Fluxograma Geral do Sistema

1.3. Sobre Hardware e Software

Os principais componentes da planta são:

- Par de cilindros
- Conjunto de lâmpadas para aquecimento
- Relé de estado sólido
- Microcontrolador STM-32
- Circuito eletrônico de condicionamento de sinal
- Sensor de temperatura
- Motor para rotação dos cilindros
- Fonte
- Tela para supervisão do usuário
- Encoder para definição do setpoint

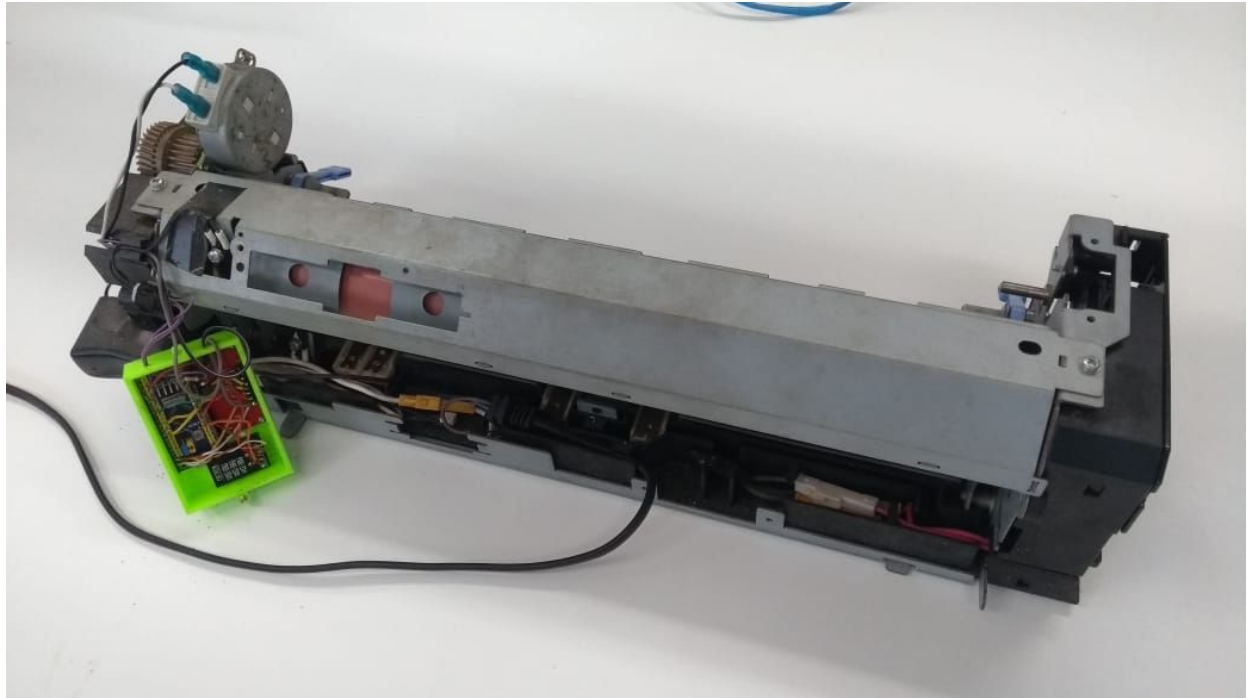


Figura 1.2 - Imagem real do sistema (frente)

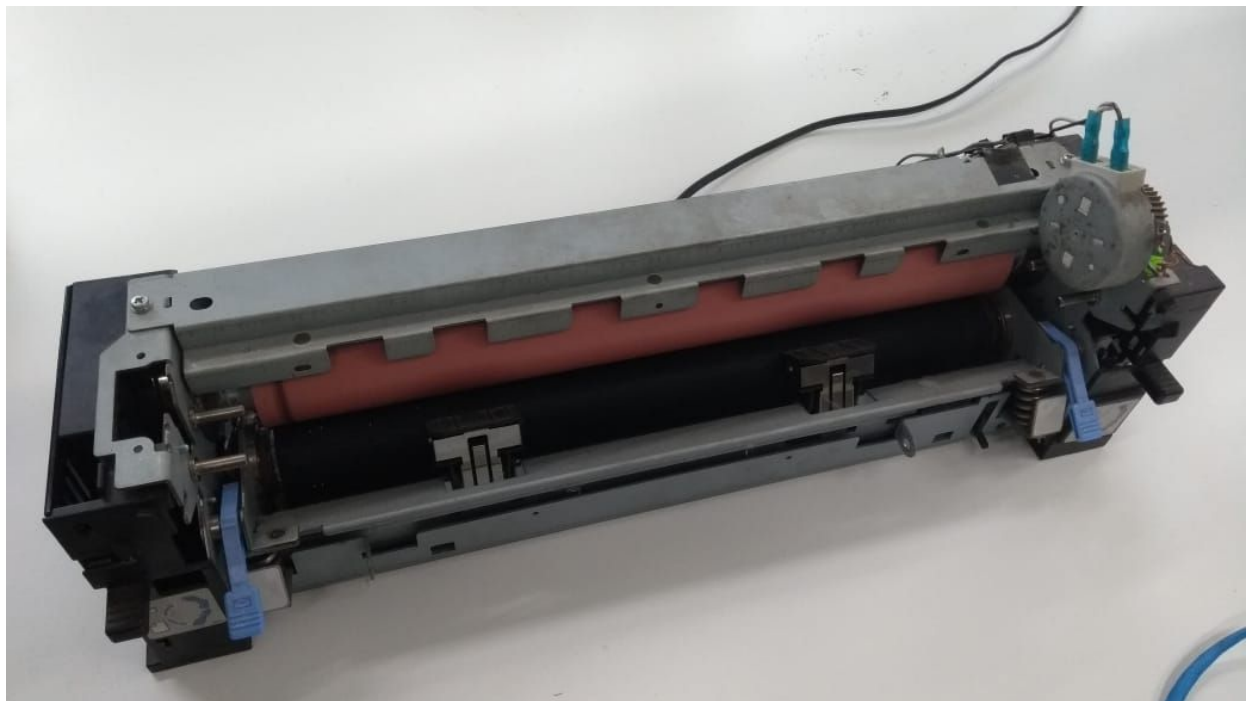


Figura 1.3 - Imagem real do sistema (verso)

O cilindro que esquentava a laminadora tem 3 lâmpadas em seu interior, e o controle será feito em cima destas lâmpadas. O distúrbio que o sistema sofre é quando a placa é inserida para que seja impresso o circuito, porém como a área da placa é muito pequena em relação à área dos cilindros, o distúrbio é irrelevante. O relé é utilizado para converter os sinais enviados pelo

microcontrolador STM-32, para que a laminadoras seja alimentada com a tensão adequada. O microcontrolador faz a comunicação planta/ computador, ele faz a aquisição e controle dos sinais. A interface de comunicação entre computador e microcontrolador foi a ARDUINO IDE, na qual é permitida a comunicação com o STM-32.

A seguir, é apresentado o circuito elétrico da planta.

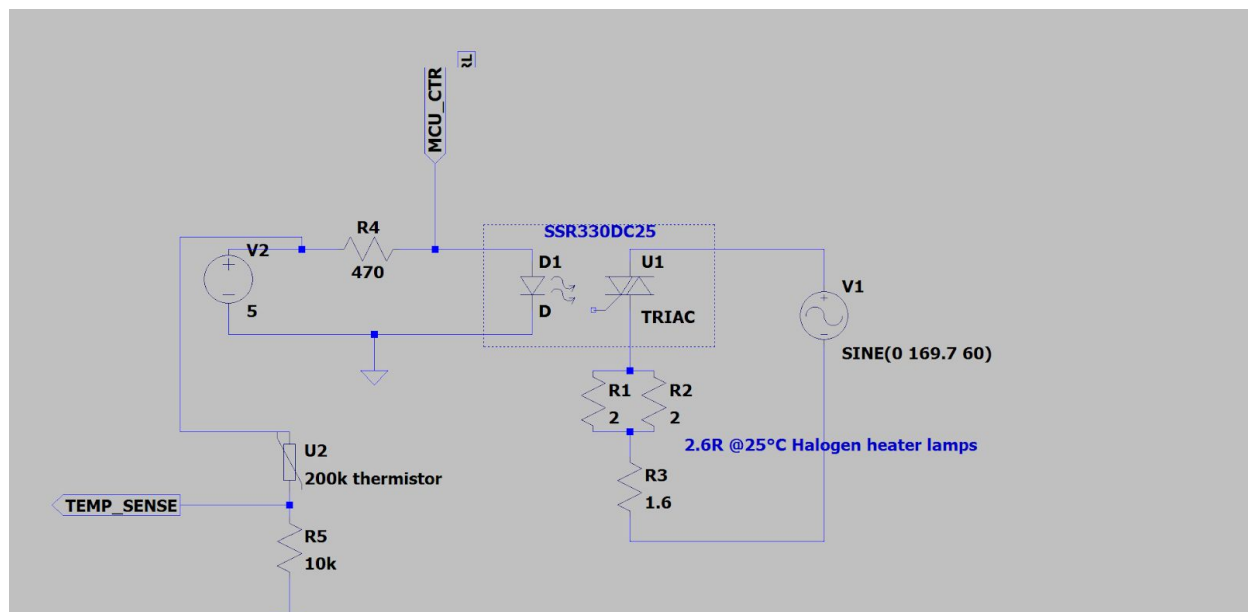


Figura 1.4 -Representação do circuito elétrico

No circuito podem ser notados o conjunto de lâmpadas, representados pelos resistores R1 a R3, o relé de estado sólido e o sensor de temperatura U2, que forma um divisor de tensão com o resistor R5.

2. Modelagem

2.1 Calibração do Sensor

Para fazer aquisição dos dados de temperatura, inicialmente era necessário conhecer a curva de calibração do sensor. Como não havia datasheet disponível, a curva de calibração foi determinada experimentalmente, e os resultados estão disponíveis no gráfico abaixo.

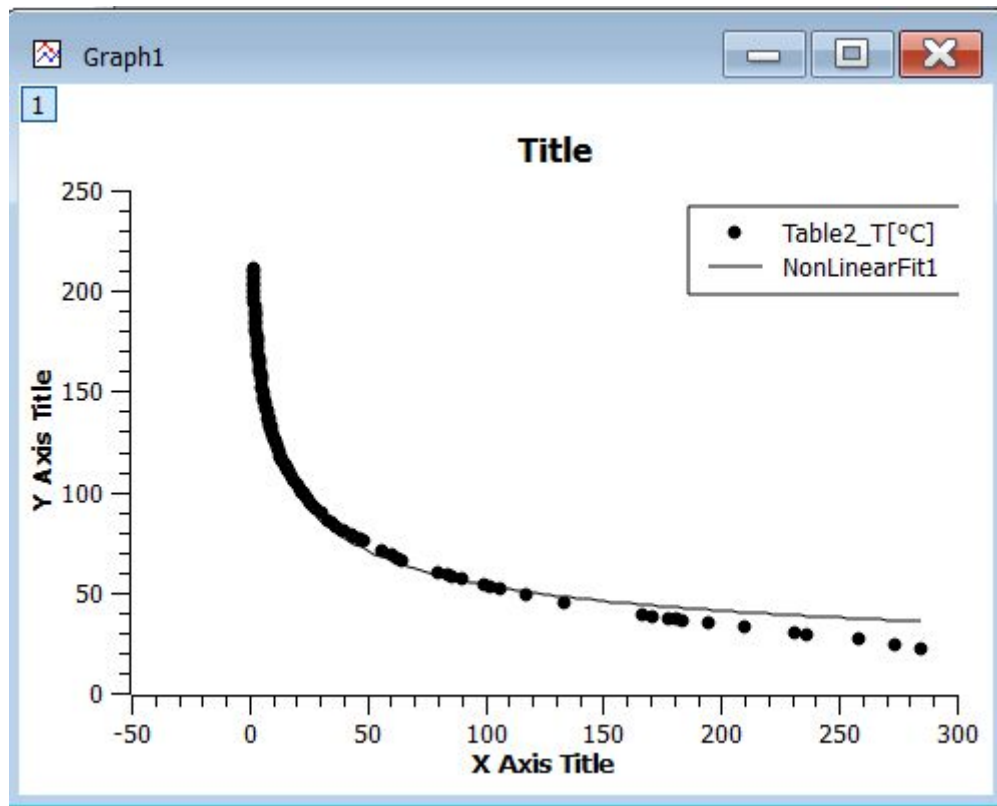


Figura 2.1 -Fit com a equação de Steinhart-Hart

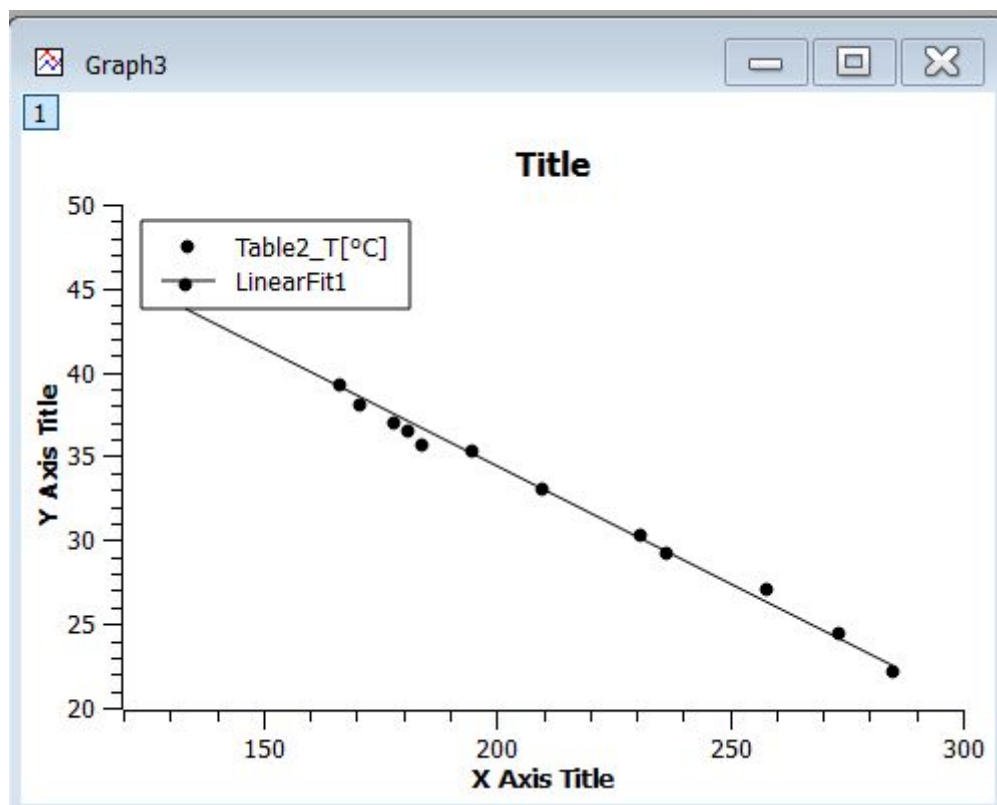


Figura 2.2 -Fit com a equação Linear

As duas equações foram utilizadas, sendo que o software do microcontrolador alterna entre elas de acordo com a faixa de temperatura, garantir que a saída do sensor se aproxima da real temperatura do sistema.

2.2. Aquisição dos dados

Para determinar o modelo da planta, foi dado um degrau de máximo de ciclo de trabalho na planta, e os dados foram coletados a cada 100 ms pelo cabo USB que conecta à porta serial, com auxílio de um adaptador USB-Serial RS232.

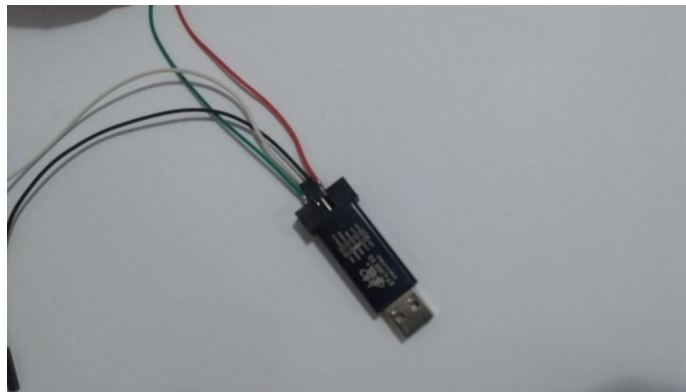


Figura 2.3 - Cabo para conexão

A partir dos dados, que foram coletados em formato .txt, foi possível trabalhar com eles no MatLab Ident.

```
#Amostra (N) , Temperatura (T)
#Tempo de amostragem 100ms
#Degrau com relé de estado sólido ligado (110V) em N = 100
0,38.80
1,39.50
2,39.40
3,39.00
4,38.80
5,39.00
6,38.60
7,39.00
8,39.00
9,38.80
10,39.00
11,40.20
12,39.50
13,38.80
14,40.70
15,38.90
16,38.40
17,38.60
18,38.30
19,38.80
20,39.50
21,39.30
22,38.80
23,38.70
```

Figura 2.4 - Dados adquiridos

2.3. Modelos Obtidos (Matlab/Ident)

Foi levantado o modelo do sistema com a toolbox de identificação de sistemas do MatLab. Para poder realizar posteriormente o controle, um sistema de primeira ordem com tempo morto foi utilizado como modelo para o Ziegler Nichols e Cohen Coon, porém para o projeto com lugar das raízes foi escolhido um modelo de terceira ordem.

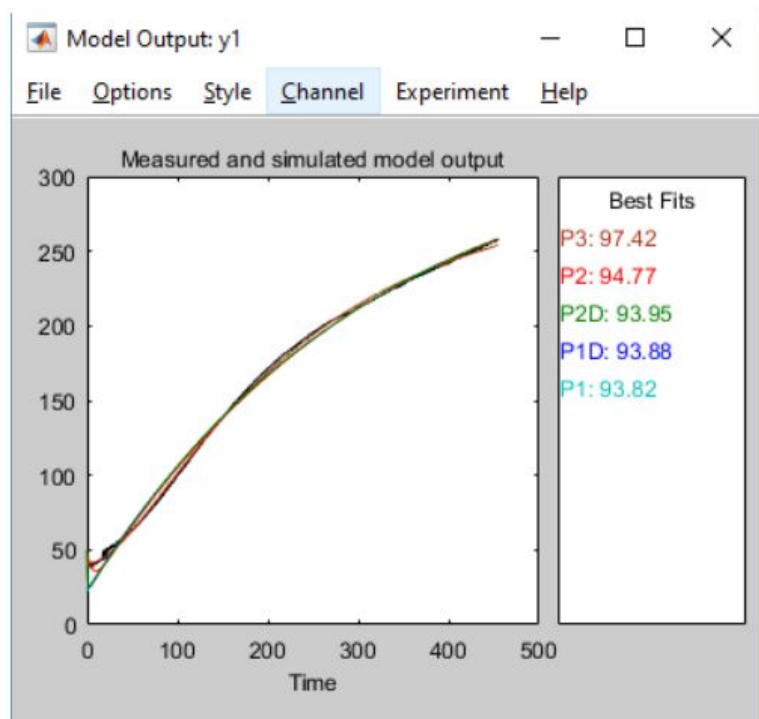


Figura 2.5 - Fits dos vários modelos levantados para o sistema

Como mostrado na figura acima, para o lugar das raízes foi utilizado o modelo P3 (terceira ordem) que teve fit de 97,42% e para o ZN e Cohen Coon o modelo P1D (primeira ordem), que teve fit de 93,88%.

Para fins de comparação, foram testados diversos algoritmos no MatLab.

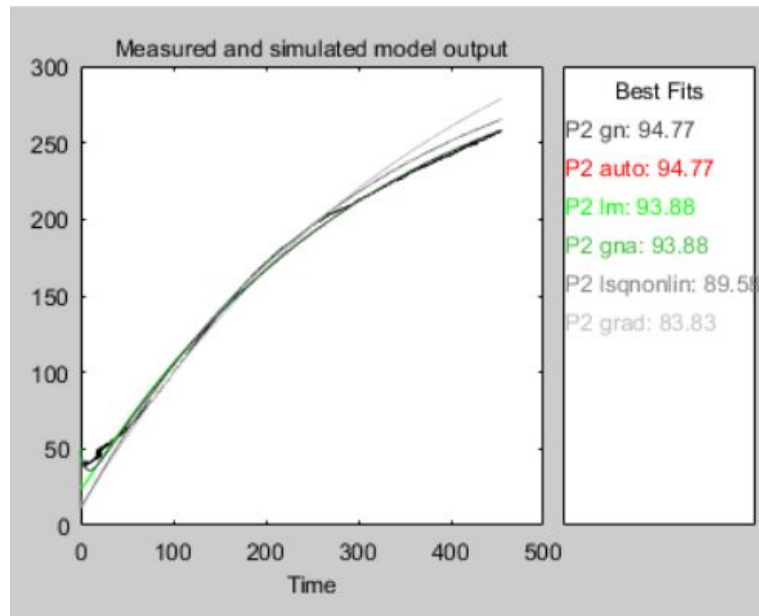


Figura 2.6 - Fits dos sistemas de segunda ordem

Os modelos acima se encaixam melhor dos que o de primeira ordem, porém como para o método Ziegler Nichols e Cohen Coon são utilizados modelo de primeira ordem, seria necessário aproximá-las, deixando o resultado parecido, logo optamos por usar as originais

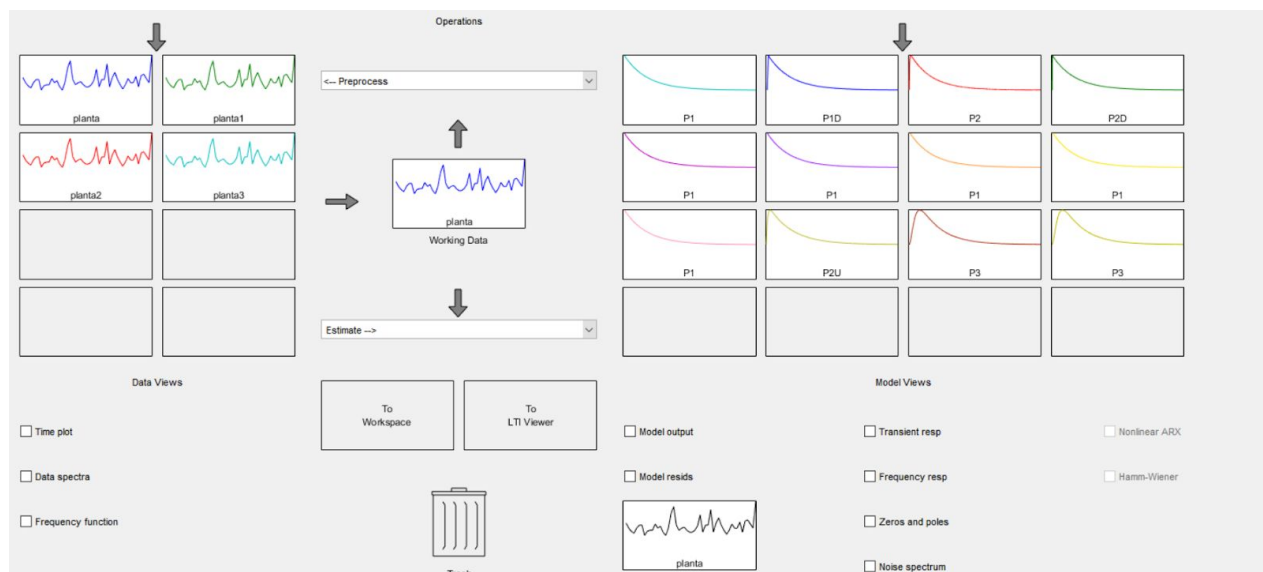


Figura 2.7 - Janela da Toolbox de Identificação de Sistemas

O modelo de primeira ordem adotado tem a forma da equação abaixo:

$$G(s) = \frac{K_p}{1 + T_{p1} * s} * e^{-T_d * s}$$

$$K_p = 333.02 \pm 0.87761$$

$$T_{p1} = 319.21 \pm 1.7338$$

$$T_d = 3 \pm 0.75671$$

Figura 2.8 - Modelo de primeira ordem com atraso

O modelo de terceira ordem adotado tem a seguinte forma:

$$G(s) = \frac{K_p}{(1 + T_{p1} * s)(1 + T_{p2} * s)(1 + T_{p3} * s)}$$

$$K_p = 292.61 \pm 13.413$$

$$T_{p1} = 218.89 \pm 37.852$$

$$T_{p2} = 30.713 \pm 864.3$$

$$T_{p3} = 28.38 \pm 964.08$$

Figura 2.9 - Modelo de terceira ordem

Os algoritmos disponibilizados pelo MATLAB para ajuste de curvas também foram comparados. Para o modelo de segunda ordem, tanto o ajuste pelo Algoritmo de Levenberg-Marquardt quanto o algoritmo Gauss-Newton Adaptativo convergiram para os mesmos valores, e o algoritmo que melhor explicou os dados foi o algoritmo de Gauss Newton comum.

2.4. Simulações em Malha Aberta

Com os modelos determinados, procedeu-se à observação do comportamento do sistema em malha aberta. Os parâmetros foram inseridos no MatLab e a resposta ao degrau unitário foi simulada. Os resultados podem ser vistos nas duas figuras abaixo.

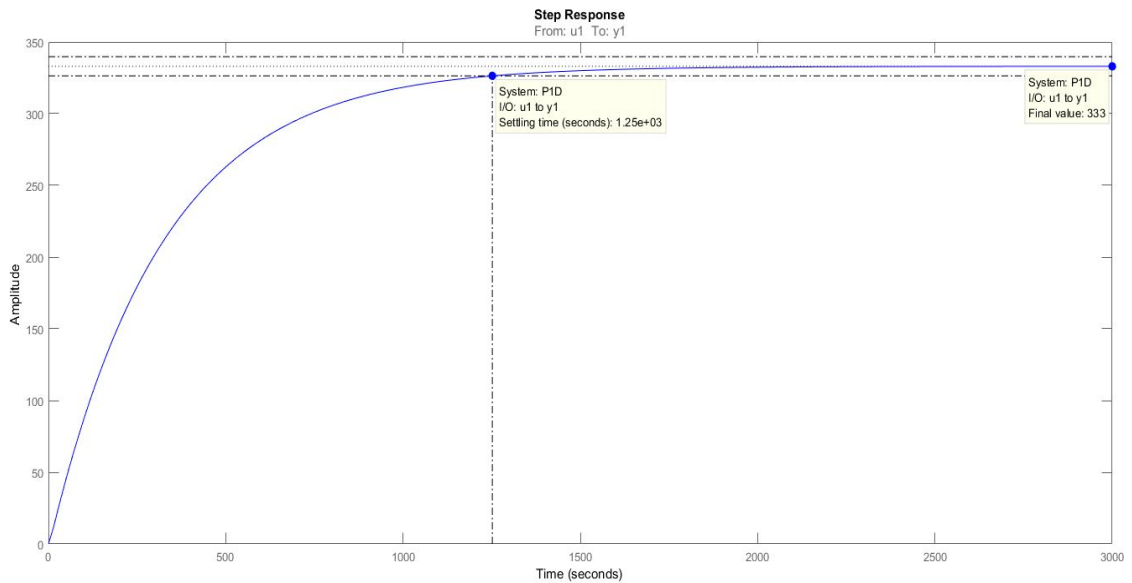


Figura 2.10 - Resposta ao degrau do modelo de primeira ordem com atraso

O modelo de primeira ordem apresentou tempo de assentamento de 1250 segundos, bastante longo, e um erro absoluto de 332 ao degrau unitário, valor também muito elevado.

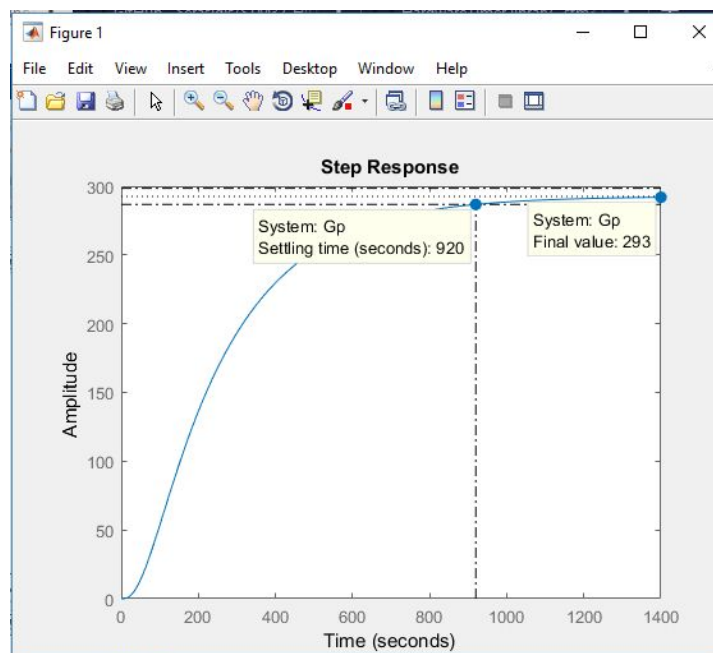


Figura 2.11 - Resposta ao degrau do modelo de terceira ordem com pólos reais

O modelo de terceira ordem será discutido melhor na próxima seção, dedicada ao Controle.

3. Controle

3.1 Metodologia para Escolha de Parâmetros PID

Repete-se aqui o gráfico de resposta ao degrau em malha aberta do modelo de terceira ordem, desta vez destacando as características relevantes para o projeto.

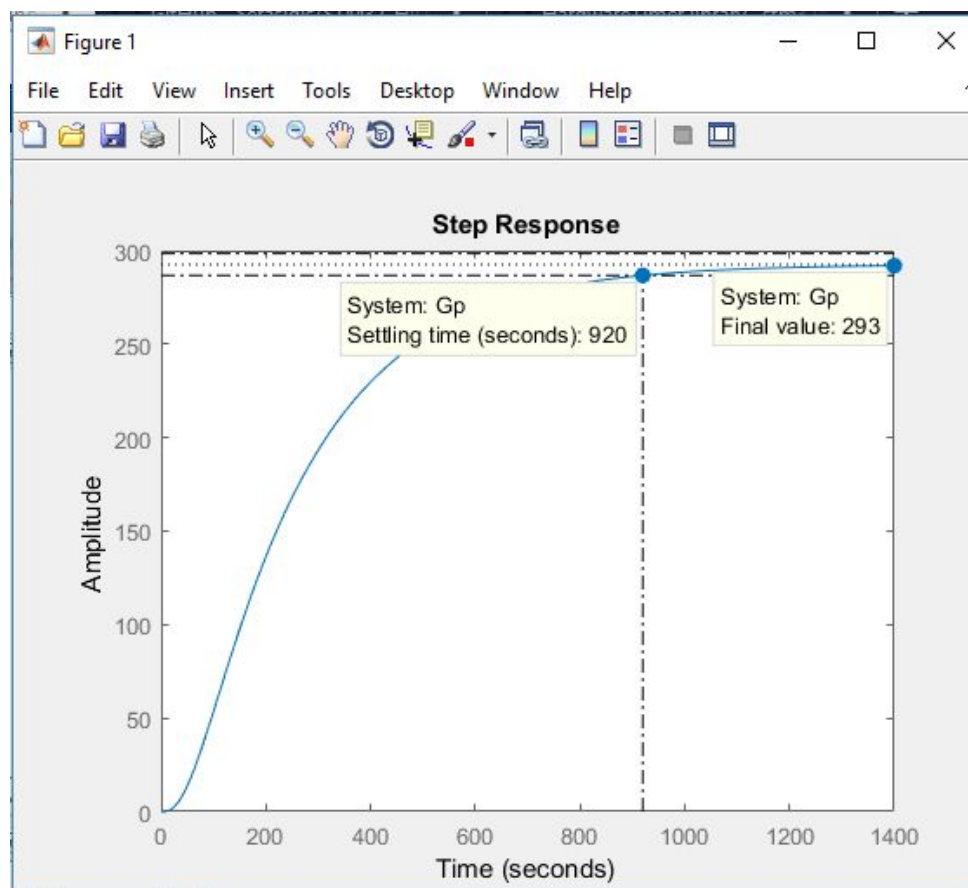


Figura 3.1 - Resposta ao degrau do modelo de terceira ordem

Como é possível notar no gráfico de resposta do degrau, o sistema apresentou erro absoluto em regime permanente de 292 e um tempo de assentamento bem longo de 920 segundos.

A partir destes dados, foi montado o projeto de um compensador capaz de zerar o erro ao degrau e melhorar o tempo de assentamento. Foi necessário um compensador PID, sendo que o integrador atuará para zerar o erro, enquanto o controle derivativo irá melhorar a resposta transiente, isto é, o longo assentamento do sistema.

3.2 Projeto de Controle PID via Ziegler-Nichols

Assim, o projeto do compensador PID foi feito utilizando o tradicional e testado método de Ziegler-Nichols para sintonia de controladores.

Inicialmente, a topologia foi definida e um diagrama de blocos para simulação do controle do sistema em malha fechada foi criado com auxílio do Simulink.

A partir das tabelas de Ziegler-Nichols para Sistemas Auto-Ajustáveis (que é o caso deste sistema, já que ele se estabiliza em um valor final determinado), da função de transferência de malha aberta e do gráfico de resposta ao degrau, definiram-se os ganhos adequados para o controlador.

Tabela 1 - Parâmetros do Controlador por Ziegler-Nichols para Sistemas Auto-Ajustáveis

Tipo de controlador	Kp	Ti	Td
P	$\frac{\tau}{T}$	∞	0
PI	$0,9 \cdot \frac{\tau}{T}$	$\frac{T}{0,3}$	0
PID	$1,2 \cdot \frac{\tau}{T}$	2.T	0,5.T

Tabela 2 - Parâmetros do Controlador Determinados por Ziegler-Nichols

Kd	Kp	Ki	Td[s] = Kd/Kp	Ti[s] = Kp/Ki
191.52	127.68	21.28	1.5	6

3.3 Projeto de Controle PID via Cohen e Coon

Similarmente, foi projetado um controlador adicional através das técnicas de sintonia definidas por Cohen e Coon. O método é bastante parecido com Ziegler-Nichols, no sentido de que ambos utilizam uma aproximação de primeira

ordem com atraso para o sistemas, no entanto, o procedimento de Cohen e Coon leva em conta o ganho estático do sistema.

Seguindo as tabelas de ajuste, analisando a resposta ao degrau em malha aberta e o modelo do sistema, os ganhos foram determinados.

Tabela 3 - Parâmetros do Controlador por Cohen e Coon para Sistemas Auto-Ajustáveis

Tipo de controlador	Kp	Ti	Td
P	$\frac{\tau}{T.K} \cdot \left(1 + 0,33 \cdot \frac{T}{\tau}\right)$	∞	0
PI	$\frac{\tau}{T.K} \cdot \left(0,9 + 0,082 \cdot \frac{T}{\tau}\right)$	$T \cdot \left(\frac{3,33 + 0,3 \cdot \frac{T}{\tau}}{1 + 2 \cdot \frac{T}{\tau}}\right)$	0
PID	$\frac{\tau}{T.K} \cdot \left(1,35 + 0,27 \cdot \frac{T}{\tau}\right)$	$T \cdot \left(\frac{2,5 + 0,5 \cdot \frac{T}{\tau}}{1 + 0,6 \cdot \frac{T}{\tau}}\right)$	$T \cdot \left(\frac{0,37}{1 + 0,2 \cdot \frac{T}{\tau}}\right)$

Tabela 4 - Parâmetros do Controlador Determinados por Cohen e Coon

Kd	Kp	Ki	Td[s] = Kd/Kp	Ti[s] = Kp/Ki
0.1596	0.4321	0.0578	0.369	7.472

O código MATLAB usado para computar o modelo utilizado e os ganhos foi o seguinte:

```
s = tf('s')

%Modelo Ordem 1
K = 333.02;
tau = 319.21;
T = 3;

G1= K*exp(-T*s)/(1+tau*s)
```

```

figure(2)
step(G1)

%Ziegler-Nichols PID
Kp_zn = 1.2*tau/T
Ti_zn = 2*T
Td_zn = 0.5*T

Ki_zn = Kp_zn/Ti_zn
Kd_zn = Kp_zn*Td_zn

%Cohen-Coon PID
Kp_cc = tau/(K*T)*(1.35+0.27*T/tau)
Ti_cc = T*((2.5+0.5*T/tau)/(1+0.6*T/tau))
Td_cc = 0.37/(1+0.2*T/tau)

Ki_cc = Kp_cc/Ti_cc
Kd_cc = Kp_cc*Td_cc

%Controladores PID
%G_pid = Kd*s+ Kp +Ki/s
Gc_zn = Kp_zn + Ki_zn/s + Kd_zn*s
Gc_cc = Kp_cc + Ki_cc/s + Kd_cc*s

%Sistemas em malha fechada
T_zn = Gc_zn*G1
T_cc = Gc_cc*G1
T_zn = T_zn/(1+T_zn)
T_cc = T_cc/(1+T_cc)

%Degrau em malha fechada
figure(4)
subplot(1,2,1)

```



```

step(T_zn)
subplot(1,2,2)
step(T_cc)

```

3.4 Projeto de Controle PID via Lugar das Raízes

Para o modelo de terceira ordem, que melhor havia explicado os dados, procedeu-se com o projeto de um controlador pelo método de lugar das raízes.

Os parâmetros de projeto foram overshoot nulo ou de até 3% e tempo de assentamento próximo de 150s.

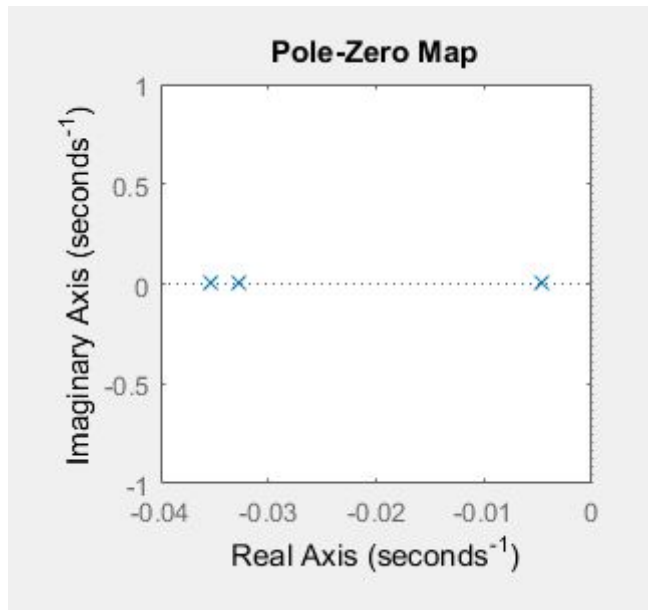


Figura 3.2 - Pólos do Sistema em Malha aberta

Para projetar o controlador, determinou-se que os pólos adicionais deveriam ser cancelados pelos zeros do compensador para manter a validade da aproximação de segunda ordem do sistema. O pólo intermediário foi mantido já que ele atendia de imediato as especificações de projeto, conforme a equação abaixo:

$$T_s = 4/\sigma = 4/0.0326 = 122.7s$$

O ganho do compensador poderia ser determinado também pela condição de fase para os pólos de malha fechada desejados, mas optou-se por usar o gráfico de lugar das raízes do Matlab para determinar o maior ganho possível com mínimo de overshoot.

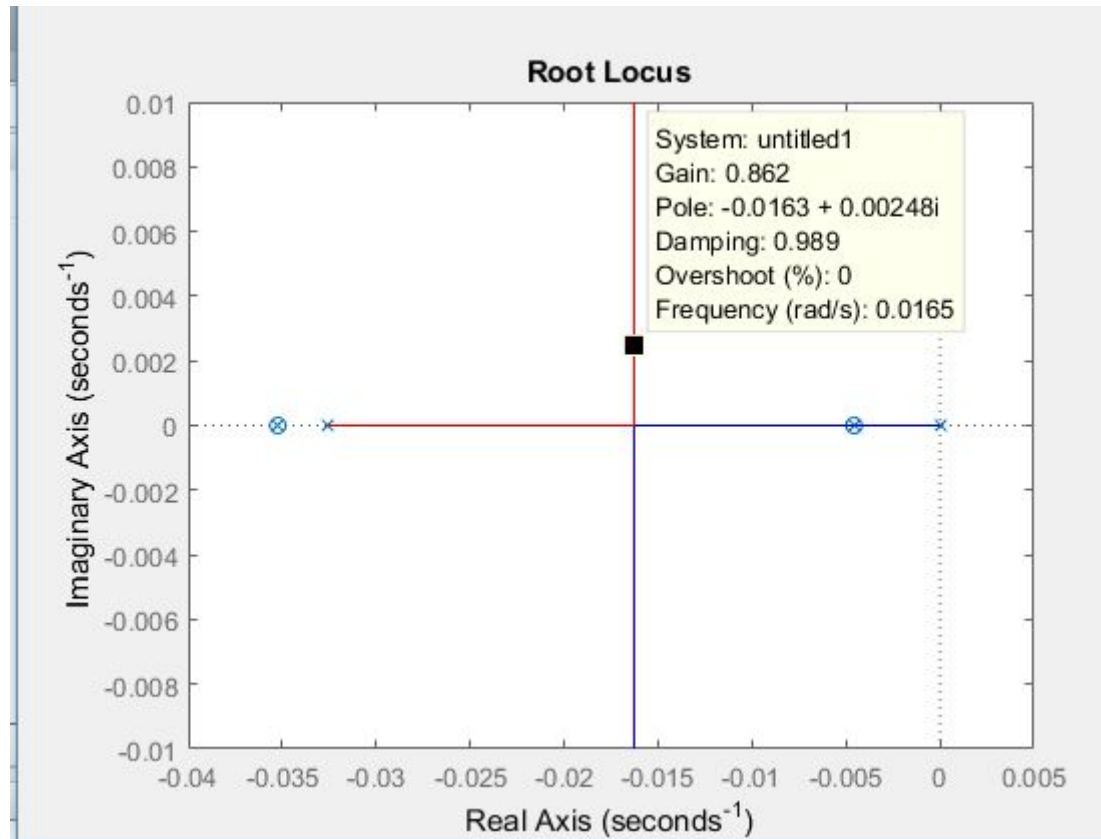


Figura 3.3 -Determinação dos Ganhos do Controlador

Tabela 5 - Parâmetros do Controlador por Lugar das Raízes

Kd	Kp	Ki	Td[s] = Kd/Kp	Ti[s] = Kp/Ki
0.205	0.0082	0.000033	25	248,48

3.5 Comparação da Malha Fechada em Simulação Matlab e Planta Física

Para verificar a validade parâmetros calculados, fez-se uma simulação no MATLAB do sistema em malha fechada com ambos os controladores, através do seguinte código:

```
%Controladores PID
%G_pid = Kd*s+ Kp +Ki/s
Gc_zn = Kp_zn + Ki_zn/s + Kd_zn*s
Gc_cc = Kp_cc + Ki_cc/s + Kd_cc*s
%Sistemas em malha fechada
```

```

T_zn = Gc_zn*G1
T_cc = Gc_cc*G1
T_zn = T_zn/(1+T_zn)
T_cc = T_cc/(1+T_cc)
%Degrau em malha fechada
figure(4)
subplot(1,2,1)
step(T_zn)
subplot(1,2,2)
step(T_cc)

```

O resultado pode ser visto no gráfico abaixo.

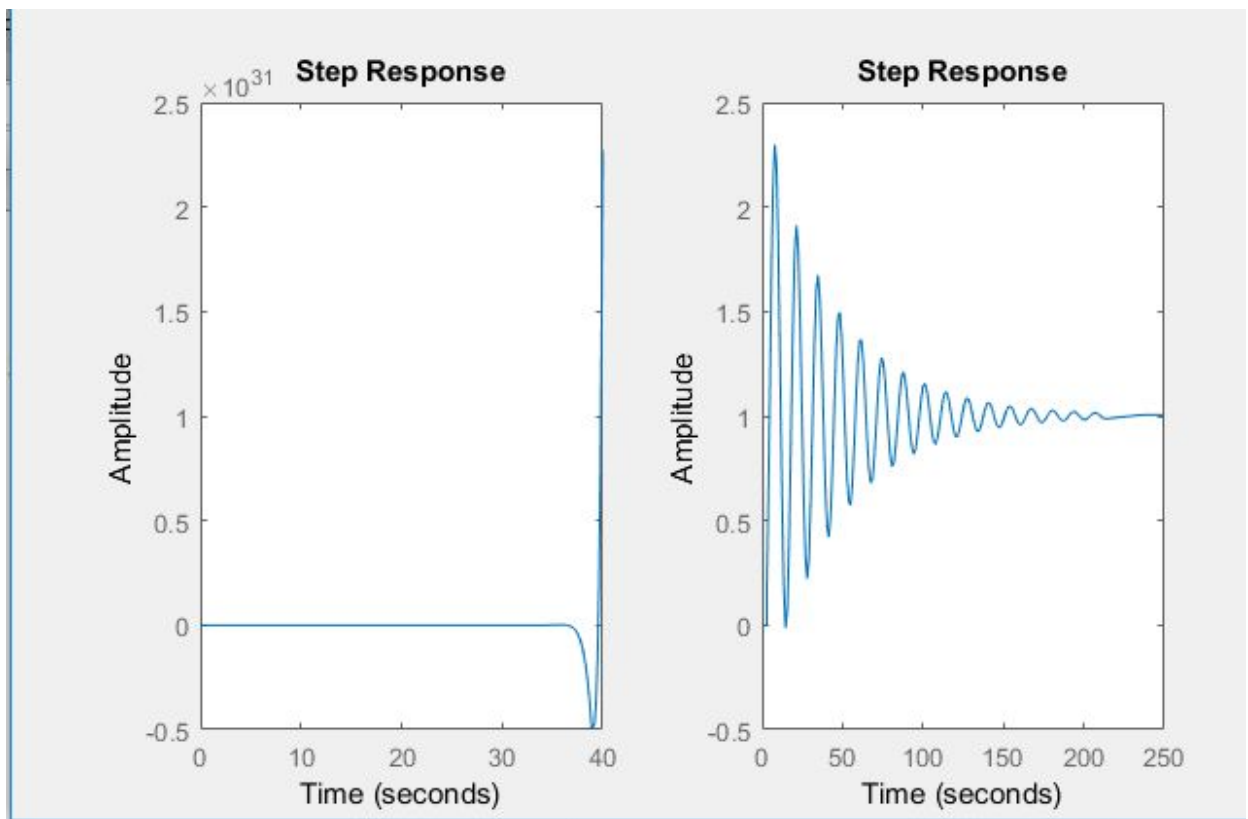


Figura 3.4 - Simulação para Ziegler-Nichols e Cohen e Coon, respectivamente

O resultado da simulação não foi satisfatório para as sintonias com modelo de primeira ordem com atraso. O controle com Ziegler-Nichols apresentou uma resposta ao degrau extremamente agressiva, e a saída foi a valores muito altos. O controle com Cohen-Coon atingiu o tempo de estabelecimento esperado e zerou o erro, mas com overshoot de mais de 100%.

Para o modelo de terceira ordem, o código de projeto e simulação utilizado no Matlab foi o seguinte:

```
s = tf('s');
%Modelo ordem 3
Gp = 292.61/((1+218.89*s)*(1+30.713*s)*(1+28.38*s))
subplot(2,2,1)
step(Gp)
Tplanta = feedback(Gp,1)
subplot(2,2,3)
step(Tplanta)

%Controlador PID
%Métricas desejadas, especificações realistas
%Eliminar erro em regime permanente-> PI
%Ts aprox = 150s, máx atingível para 100 graus com atuador
saturado -> PID
%Max overshoot = 1%, o controle ON/OFF tem cerca de 11%
%PID raiz por lugar das raízes:
% zeros-> no polos de malha aberta
%Ts = 4/sigma -> sigma = 4/Ts
%Um dos pólos fica em zero (integrador)
%Segundo polo, parte real em 2*sigma, para que o root locus
fique na metade
plant_poles = pole(Gp)
pzmap(Gp)
Gc = (s-plant_poles(1))*(s-plant_poles(3))/s
rlocus(Gp*Gc)
%Ganho escolhido pelo lugar das raízes, maior ganho com mínimo
de overshoot (0.7%)
K = 0.205
Gc = K*Gc
```

```

T = feedback(Gp*Gc,1)
subplot(2,2,2)
step(T)
%Comparando coeficientes com PID canônico
%syms Kp Ki Kd S
%Gpid = Kp+Kd*S+Ki/S
%Gpid = (Kd*S^2+Kp*S+Ki)/S
Kd = 0.205;
Kp = 0.00816;
Ki = 3.3e-5;

```

O resultado da simulação pode ser visto na figura abaixo.

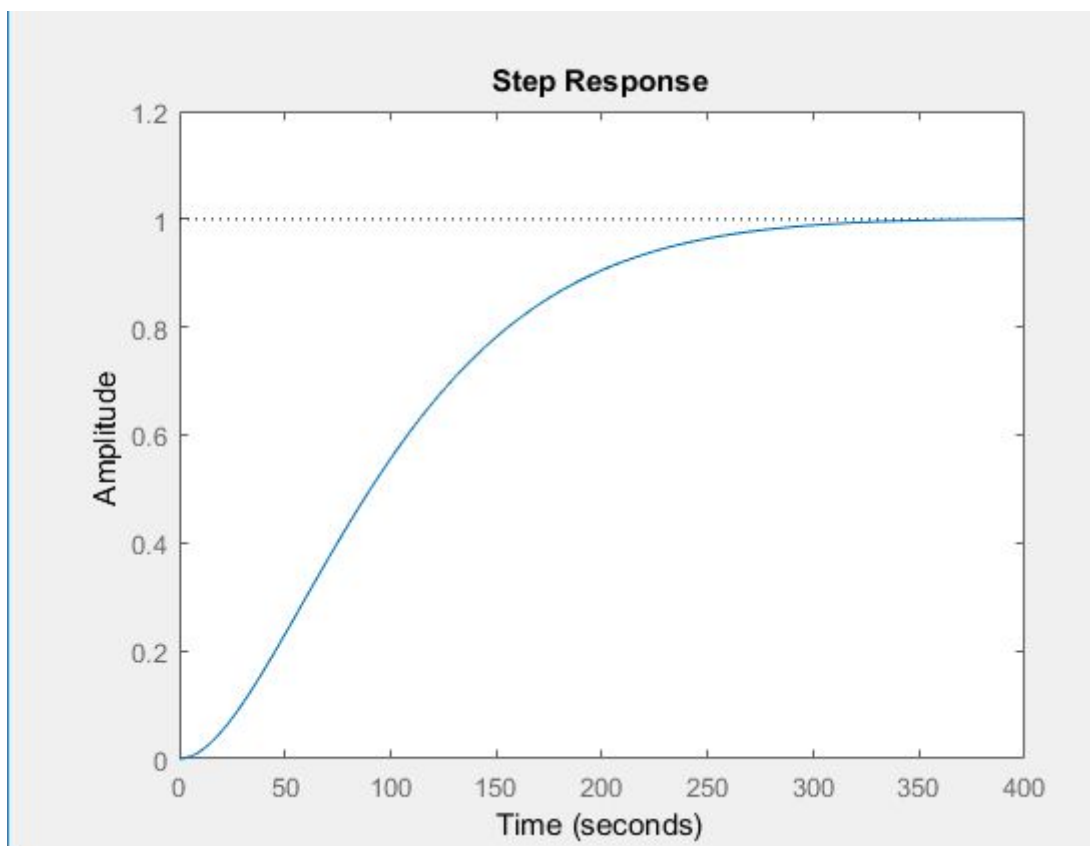


Figura 3.5 - Simulação para Lugar das Raízes

Para o controle projetado por lugar das raízes, a simulação foi muito satisfatória. Não houve qualquer ultrapassagem ou erro, e o tempo de estabelecimento foi dentro

das especificações.

Após a simulação, o passo seguinte foi testar o controlador PID na planta, o que foi feito aplicando os ganhos calculados ao algoritmo de controle pela porta serial. Cada conjunto de ganhos foi aplicado uma vez para a aquisição dos dados, e então aguardava-se o resfriamento do sistema até que o novo conjunto de ganhos pudesse ser testado. Os ganhos testados, que são os mesmos calculados anteriormente, estão dispostos na tabela abaixo.

Tabela 6 - Parâmetros Finais

Metodologia	Kd	Kp	Ki	$T_d[\text{min}] = K_d/K_p$	$T_i[\text{min}] = K_p/K_i$
Cohen e Coon	0.1596	0.4321	0.0578	0.369	7.472
Ziegler e Nichols	191.52	127.68	21.28	1.5	6
Lugar das Raízes	0.205	0.0082	0.000033	25	248,48

Na figura seguinte, é apresentado a resposta da planta com o método root-locus, onde se obteve um overshoot de 38%, o que mostrou não ser o método mais adequado para o controle. O tempo de assentamento foi muito longo, portanto foi outro motivo para que não fosse utilizado esse modelo.

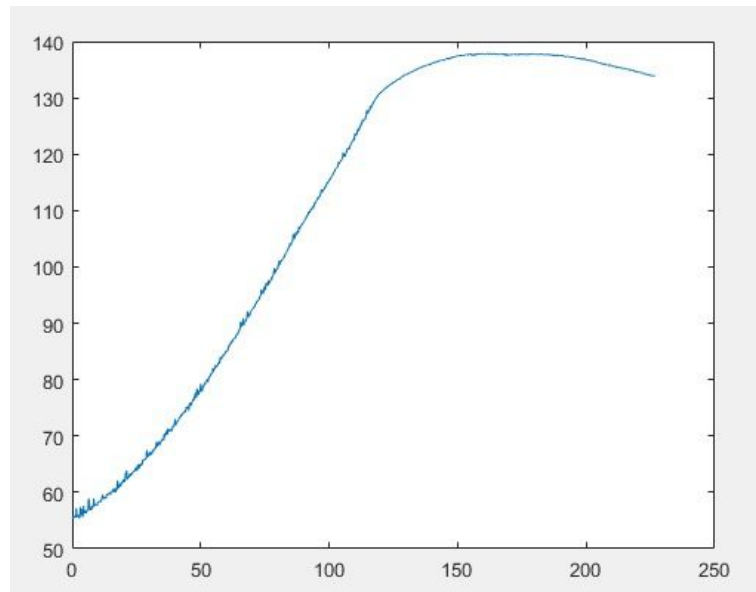


Figura 3.6 - Teste na planta com Root-locus, setpoint=100°C

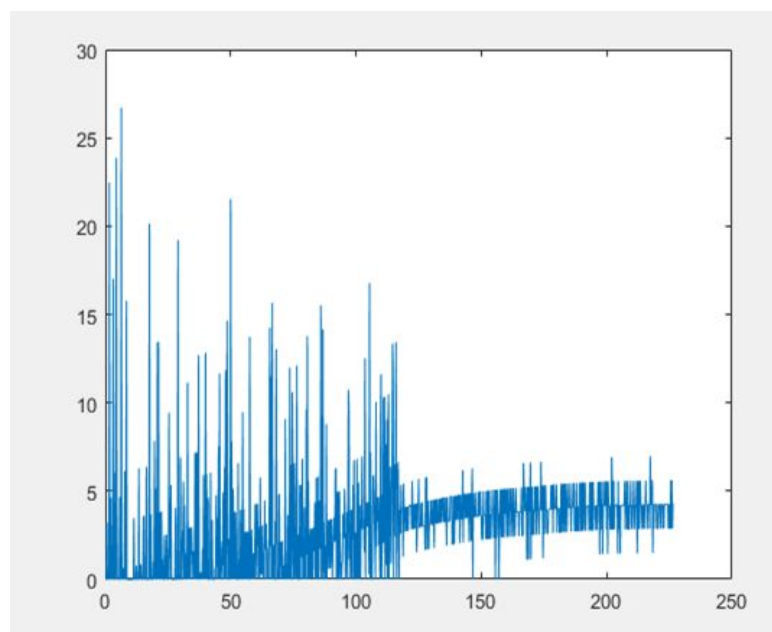


Figura 3.7 - Sinal de controle do controlador com Root-locus, setpoint=100°C

Já neste caso abaixo, obteve-se pelo método de Cohen Coon, um overshoot menor, na faixa de 25%, o que ainda não é um resultado satisfatório. O tempo de assentamento apesar de menor, ainda não foi satisfatório.

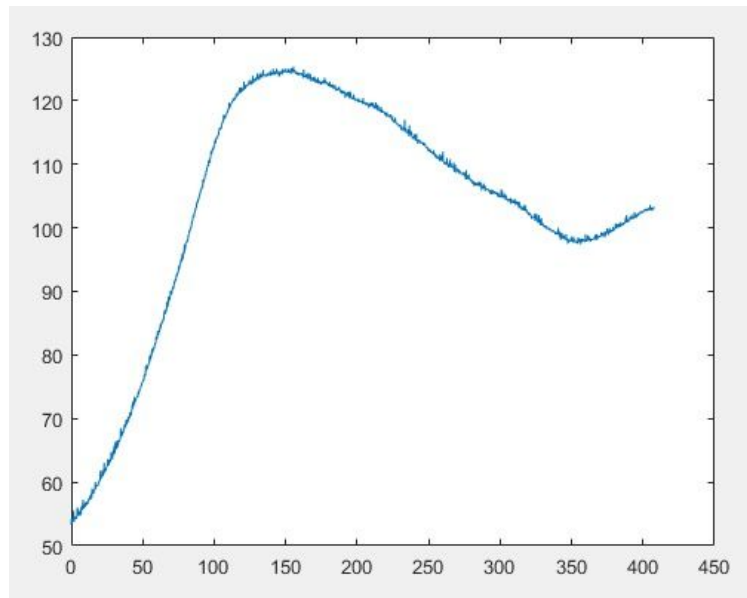


Figura 3.8 - Teste na planta com Cohen e Coon, setpoint=100°C

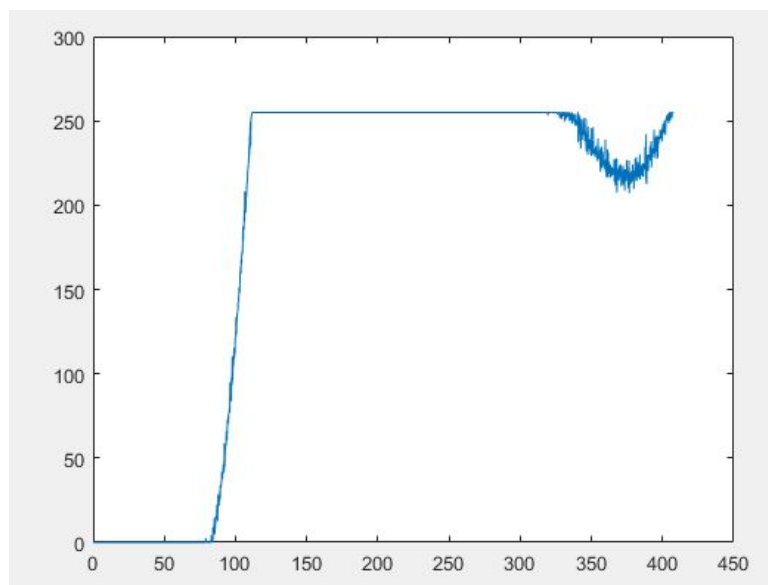


Figura 3.9 - Sinal de controle do controlador com Cohen-Coon, setpoint=100°C

Finalmente, para o método de Ziegler-Nichols, obteve-se um overshoot bom, no valor de 10%, o qual definiu-se como a melhor situação para a planta. Também é possível observar que o tempo de assentamento de 250 segundos.

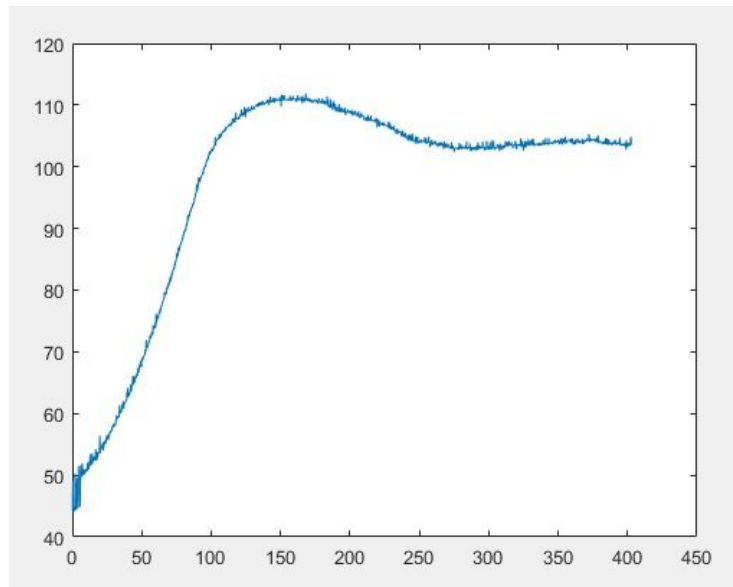


Figura 3.10 - Fim do teste na planta com Ziegler- Nichols, setpoint=105°C

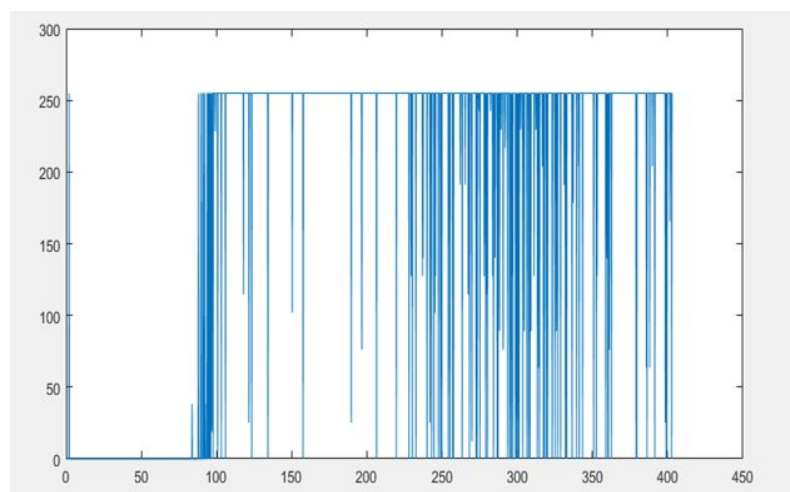


Figura 3.11 - Sinal de controle do controlador com Ziegler-Nichols, setpoint=105°C

Os resultados da implementação foram inesperados. O controlador Ziegler-Nichols foi o que apresentou ação de controle mais agressiva dentre os dois, o sinal de saída do compensador oscilou bastante, mas o tempo de estabelecimento foi menor e houve ultrapassagem de cerca de 10%, dentro do esperado.

Com o controlador Cohen e Coon o sistema chegou a atingir o valor final após a ultrapassagem, mas depois o erro aumentou novamente.

O controlador do lugar das raízes foi o mais passivo dentre todos, e acabou levando a um overshoot muito grande, sem corrigi-lo com eficácia, acabando por não zerar o erro em regime permanente.

4. Conclusão

4.1. Síntese do Trabalho

O trabalho propôs-se a coletar dados para levantar um modelo teórico da planta de uma laminadora, que pode ser caracterizada como um sistema térmico, e portanto utilizar o modelo para projetar um compensador que seria aplicado na planta real para controle de temperatura.

Os modelos criados a partir das manipulações e utilização das ferramentas do MatLab apresentaram um alto grau de compatibilidade com os dados experimentais, com qualidade de ajuste (*fit*) de pelo menos 92%.

Embora a simulação dos modelos tenha sido satisfatória, o comportamento no sistema real se deu um pouco pior do que as simulações executadas, porém o overshoot, tempo de assentamento e erro permanente se apresentaram com bons valores, sendo assim um bom resultado para o projeto.

4.2. Dificuldades

Foram encontradas dificuldades para a concepção e montagem do projeto, e principalmente para encontrar os materiais necessários.

Além disso, também houve dificuldades para fazer o teste na planta, uma vez que era bem demorado o resfriamento da laminadora, e portanto, era gasto certo tempo para que novos testes fossem realizados.

Nos códigos também houveram empecilhos, pois para a implementação foi necessário algum tempo de pesquisa e aprendizado.

5. Referências

Laminadora para placa de circuito impresso -Disponível em
[http://www.luccas.com.br/index.php/eletronica/montagens/
8-laminadora-para-placa-de-circuito-impresso](http://www.luccas.com.br/index.php/eletronica/montagens/8-laminadora-para-placa-de-circuito-impresso)>

<

Laminador de PCI feito com sucata - Disponível em < <https://www.py2bbs.qsl>.

br/laminador.php >

Tutorial: Estimating a transfer function model from random input using MATLAB
- Disponível em < <https://www.youtube.com/watch?v=gFZNNe1qsR8> >

NISE, Norman S. Control Systems Engineering. Edição 6