

Universidade Federal de Campina Grande - UFCG
Centro de Engenharia Elétrica e Informática - CEEI
Departamento de Engenharia Elétrica - DEE

Bruno Nascimento Gomes de Oliveira

Relatório de Estágio Supervisionado

Campina Grande, Brasil

Outubro de 2024

Bruno Nascimento Gomes de Oliveira

Relatório de Estágio Supervisionado

Relatório de Estágio Supervisionado submetido à Coordenação de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande, Campus Campina Grande, como parte dos requisitos necessários para obtenção do título de Graduado em Engenharia Elétrica.

Universidade Federal de Campina Grande - UFCG
Centro de Engenharia Elétrica e Informática - CEEI
Departamento de Engenharia Elétrica - DEE

Orientador: George Acioli Júnior, D.Sc.

Campina Grande, Brasil

Outubro de 2024

Bruno Nascimento Gomes de Oliveira

Relatório de Estágio Supervisionado

Relatório de Estágio Supervisionado submetido à Coordenação de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande, Campus Campina Grande, como parte dos requisitos necessários para obtenção do título de Graduado em Engenharia Elétrica.

Trabalho aprovado em: / /

George Acioli Júnior, D.Sc.
Orientador

Eisenhauer de Moura Fernandes
Convidado

Campina Grande, Brasil
Outubro de 2024

Agradecimentos

*"Seja forte e corajoso! Não se apavore nem desanime, pois o Senhor, o seu Deus, estará
com você por onde você andar"*

Josué 1:9

Resumo

Este relatório de estágio descreve a comunicação OPC-DA e o desenvolvimento do controle para uma planta em escala laboratorial, desenvolvida no Laboratório de Instrumentação Eletrônica e Controle(LIEC), usando módulos Peltier para aquecimento ou resfriamento das malhas. O estágio foi realizado no LIEC da Universidade Federal de Campina Grande (UFCG), com duração de 196 horas, entre 19/06 a 12/09 de 2024, sob a orientação do professor George Acioli Júnior e supervisão de Victor Luiz Santiago de Oliveira.

Palavras-chaves: OPC-DA; Peltier; COM/DCOM; PID,

Abstract

This internship report describes the OPC-DA communication and the development of control for a laboratory-scale plant, developed at the Laboratory of Electronic Instrumentation and Control (LIEC), using Peltier modules for heating or cooling the loops. The internship took place at LIEC of the Federal University of Campina Grande (UFCG), lasting 227 hours, from July 30 to September 20, 2024, under the guidance of Professor George Acioli Júnior and the supervision of Victor Luiz Santiago de Oliveira.

Key-words: OPC-DA; Peltier; COM/DCOM; PID.

Lista de ilustrações

Figura 1 – Fachada do LIEC	2
Figura 2 – Efeito Seebeck	4
Figura 3 – Módulo Peltier	5
Figura 4 – LM35	5
Figura 5 – Planta	6
Figura 6 – Esquemático do acoplamento do Peltier, <i>Fan</i> e dissipador	7
Figura 7 – Controlador	7
Figura 8 – PWM	7
Figura 9 – <i>Softwares</i>	8
Figura 10 – OPCExpert	10
Figura 11 – Primeira parte do programa	12
Figura 12 – Segunda Parte do Código	13
Figura 13 – Terceira parte do código	13
Figura 14 – Quarta parte do código	14
Figura 15 – PID	14
Figura 16 – Programa para leitura dos dados	15
Figura 17 – Visualização durante o experimento	15
Figura 18 – Visualização das variáveis durante o experimento	16
Figura 19 – Experimento antes das correções	16
Figura 20 – Resposta ao degrau	17
Figura 21 – System Identification	18
Figura 22 – Parâmetros com o <i>PID Tuner</i>	18
Figura 23 – Resposta ao controle PI com os parâmetros obtidos pelo <i>PID Tuner</i>	19
Figura 24 – Parâmetros obtidos pelo <i>BR Tuning</i>	20
Figura 25 – Respostas com os novos parâmetros aplicados	20
Figura 26 – Parâmetros obtidos para malha2	21
Figura 27 – Resultado da malha2	21
Figura 28 – Novos parâmetros com a suposição de uma malha1 mais rápida	22
Figura 29 – Comparativo entre os resultados	23
Figura 30 – Novos parâmetros para malha2	23
Figura 31 – Comparativo entre os resultados	24

Lista de abreviaturas e siglas

LIEC: Laboratório de Instrumentação Eletrônica e Controle

PID: Proporcional-Integral-Derivativo

CLP: Controlador Lógico Programável

MV: Manipulated Variable (Variável Manipulada)

SP: Set Point (Ponto de Ajuste)

TITO: Two-Input Two-Output (Sistema de Duas Entradas e Duas Saídas)

PV: Process Variable (Variável de Processo)

Sumário

1	INTRODUÇÃO	1
2	LOCAL DO ESTÁGIO	2
3	FUNDAMENTAÇÃO TEÓRICA	3
3.1	Efeitos Termoelétricos	3
3.1.1	Efeito Seebeck	3
3.1.2	Efeito Peltier	3
3.1.3	Efeito Thomson	3
3.1.4	Módulo Peltier	4
3.1.5	Sensores de Temperatura LM35	5
3.1.6	Descrição da planta	6
3.1.7	<i>Softwares</i> utilizados	6
4	ATIVIDADES REALIZADAS	9
4.1	Comunicação OPC-DA	9
4.1.1	Configuração COM/DCOM	9
4.1.2	Código no CLP	12
4.1.3	<i>Experimentos</i>	14
4.1.4	Testes para Melhoria do Controle	22
5	CONCLUSÃO	25
6	TRABALHOS FUTUROS	26

1 Introdução

Neste documento são descritas as atividades desenvolvidas pelo discente Bruno Nascimento Gomes de Oliveira durante o Estágio Supervisionado de 196 horas. O estágio foi realizado no Laboratório de Instrumentação Eletrônica e Controle (LIEC), durante 19/06/2024 até 12/09/2024, sob a orientação do professor George Acioli Júnior e a supervisão de Victor Santiago de Oliveira.

Este estágio supervisionado teve como objetivo o cumprimento da disciplina Estágio Curricular, parte integrante da grade do curso de Engenharia Elétrica da Universidade Federal de Campina Grande (UFCG). A disciplina é fundamental para a formação profissional, proporcionando ao aluno a oportunidade de aplicar na prática os conhecimentos adquiridos ao longo do curso.

A principal atividade desenvolvida durante o estágio foi trabalhar com uma planta de escala laboratorial para o controle de temperatura de duas malhas, para isso, foi necessário configuração COM/DCOM na máquina Virtual e na máquina Windows para que fosse possível obter os dados da planta, e, com isso, realizar testes de malha aberta para obter os resultados necessários para modelagem do processo e para as configurações corretas dos controladores PID.

2 Local do Estágio

O Laboratório de Instrumentação Eletrônica e Controle (LIEC) faz parte do Departamento de Engenharia Elétrica (DEE) do Centro de Engenharia Elétrica e Informática (CEEI) da Universidade Federal de Campina Grande (UFCG), localizado no campus de Campina Grande, Paraíba. Fundado em dezembro de 1975, o LIEC desempenha um papel importante no desenvolvimento de pesquisas e projetos inovadores nas áreas de instrumentação eletrônica, controle e automação.

O laboratório ocupa um prédio de aproximadamente $900 m^2$ e conta com uma infraestrutura moderna que inclui oito laboratórios bem equipados, duas salas de apoio técnico, uma sala de reuniões e diversas salas destinadas a alunos e professores. O corpo técnico do LIEC é composto por professores doutores, alunos de pós-graduação e graduação, todos empenhados em atividades de pesquisa e desenvolvimento que contribuem significativamente para o avanço da engenharia elétrica.

A figura 1 apresenta a fachada do prédio onde se localiza o LIEC, destacando sua infraestrutura que oferece um ambiente propício para a realização de atividades acadêmicas e de pesquisa.



Figura 1 – Fachada do LIEC

3 Fundamentação Teórica

Este capítulo apresenta os conceitos teóricos fundamentais que sustentam as atividades desenvolvidas durante o estágio.

3.1 Efeitos Termoelétricos

Os efeitos termoelétricos referem-se à interação entre a calor e eletricidade em materiais condutores ou semicondutores. Dentre eles, existem três principais:

3.1.1 Efeito Seebeck

Descoberto por Thomas Seebeck, este efeito ocorre quando há uma diferença de temperatura ao longo da junção entre dois materiais condutores ou semicondutores que gera uma tensão entre eles. A equação para esse fenômeno é dada

$$V = \int_{T_1}^{T_2} (S_B(T) + S_A(T)) dT$$

Onde S_A e S_B correspondem aos coeficientes de Seebeck dos materiais A e B, respectivamente.

3.1.2 Efeito Peltier

Identificado por Jean Peltier, o efeito Peltier ocorre quando uma corrente elétrica é aplicada a uma junção de dois materiais produzindo um gradiente de temperatura entre os materiais.

$$\frac{dQ}{dt} = (\Pi_B - \Pi_A) \cdot I$$

Onde Π_A e Π_B são os coeficientes de Peltier dos materiais A e B, respectivamente. Existe uma relação entre os coeficientes de Seebeck e Peltier que é dada por:

$$\Pi = S \cdot T$$

3.1.3 Efeito Thomson

Esse efeito, descoberto por Lord Kelvin, é o aquecimento ou resfriamento que ocorre em um material condutor quando há tanto uma corrente elétrica quanto um gra-

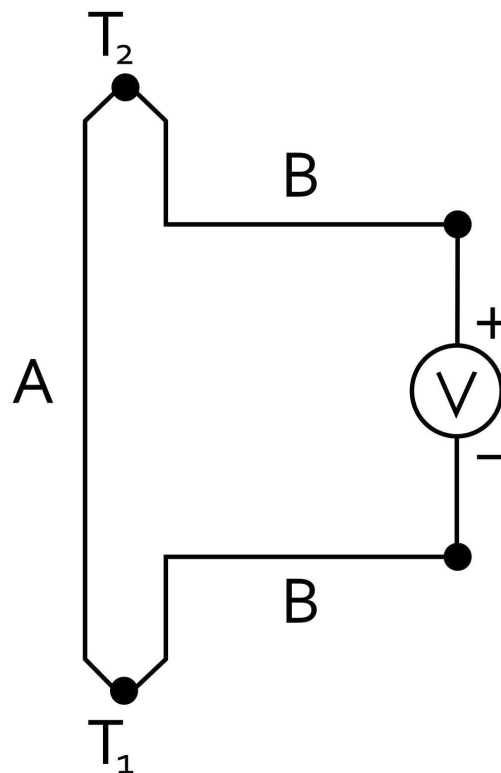


Figura 2 – Efeito Seebeck

diente de temperatura. O efeito Thomson é uma extensão dos efeitos Seebeck e Peltier, sendo mais sutil e menos comumente observado.

$$q = \rho J^2 - \mu J \frac{dT}{dx}$$

Nessa equação, q representa a resistividade do condutor, J representa a densidade de corrente, μ é o coeficiente de Thomson e $\frac{dT}{dx}$ é o gradiente de temperatura sobre o condutor.

3.1.4 Módulo Peltier

O módulo Peltier é composto por duas placas cerâmicas paralelas, entre as quais há um conjunto de semicondutores do tipo P e N conectados em série. Essas placas formam as superfícies fria e quente do módulo, e o conjunto de semicondutores são os responsáveis pelo efeito de bombeamento de calor. Conforme ilustrado na Figura 1, essa disposição permite que a corrente elétrica aplicada ao módulo provoque a transferência de calor entre as superfícies, gerando um lado quente e outro frio.

Logo, fica evidente que seu princípio de funcionamento baseia-se no efeito Peltier. Quando uma tensão é aplicada aos terminais do dispositivo, uma diferença de temperatura é estabelecida, fazendo com que o calor seja transferido de uma interface para

outra. Durante esse processo, a corrente elétrica gerada atravessa os semicondutores internos do tipo P e do tipo N, que atuam como condutores de calor entre as placas. Esses semicondutores aumentam a densidade de corrente, o que resulta em um aumento da potência. Eletricamente, as junções estão conectadas em série, enquanto, termicamente, estão conectadas em paralelo.



Figura 3 – Módulo Peltier

3.1.5 Sensores de Temperatura LM35

O sensor LM35 é um componente de alta precisão que fornece uma saída de tensão linear diretamente relacionada à temperatura do ambiente. Ele funciona com uma alimentação de 4 a 20Vdc e GND, gerando uma saída de 10mV para cada grau Celsius medido. Com uma saída de baixa impedância, linearidade e uma calibração precisa, sendo fácil de integrar em sistemas de leitura. A alimentação necessária varia de acordo com o uso, e o sensor está disponível em diferentes tipos de encapsulamento.

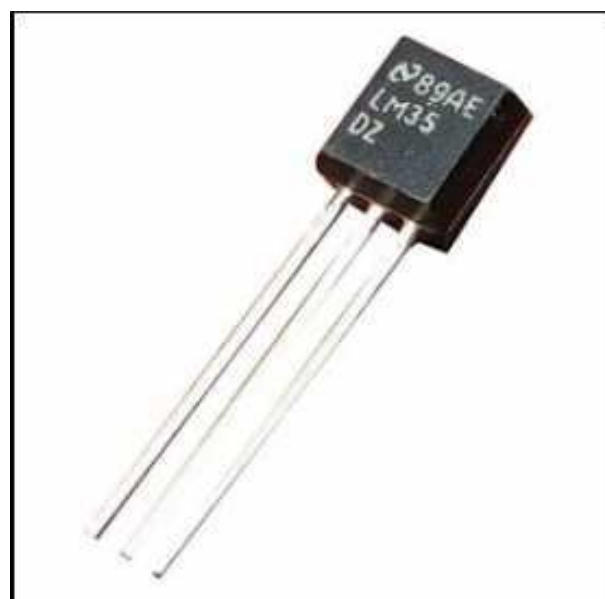


Figura 4 – LM35

3.1.6 Descrição da planta

A planta em escala laboratorial contém duas malhas iguais, onde cada malha é constituída por um módulo Peltier, um sensor de temperatura do tipo LM35, um *vent* e um dissipador, e, para a aplicação de potência ao Peltier, utiliza-se atuadores PWM.

Como duas malhas fazem parte do processo, tem-se um sistema TITO (*Two-inputs two-outputs*). Onde, em cada extremidade da chapa de metal, tem-se um Peltier que vai fornecer calor à chapa metálica. No lado oposto a essa chapa metálica, fica o conjunto de dissipador mais *vent* com o objetivo de retirar o calor. Nesse processo, a temperatura varia de 10°C a 70°C, a uma temperatura ambiente de 24°C, e as temperaturas são medidas em pontos determinados da chapa metálica pelos o sensores LM35.



Figura 5 – Planta

O controlador utilizado nesse processo é um Controllogix da linha Allen-Bradley, que recebe as saídas dos sensores como variáveis de entrada, e manda os sinais de saída para os atuadores, que por sua vez, atuam sobre os módulos.

3.1.7 Softwares utilizados

O RSLogix 5000 e o RSLinx são softwares essenciais desenvolvidos pela Rockwell Automation para programação e CLPs da linha Allen-Bradley, amplamente utilizados na automação industrial. O RSLogix 5000 é uma plataforma de desenvolvimento que permite a criação, monitoramento e depuração de sistemas de controle.

O RSLinx, por sua vez, é o software de comunicação que complementa o RSLogix 5000. Ele atua como um servidor de dados, responsável por gerenciar a comunicação entre

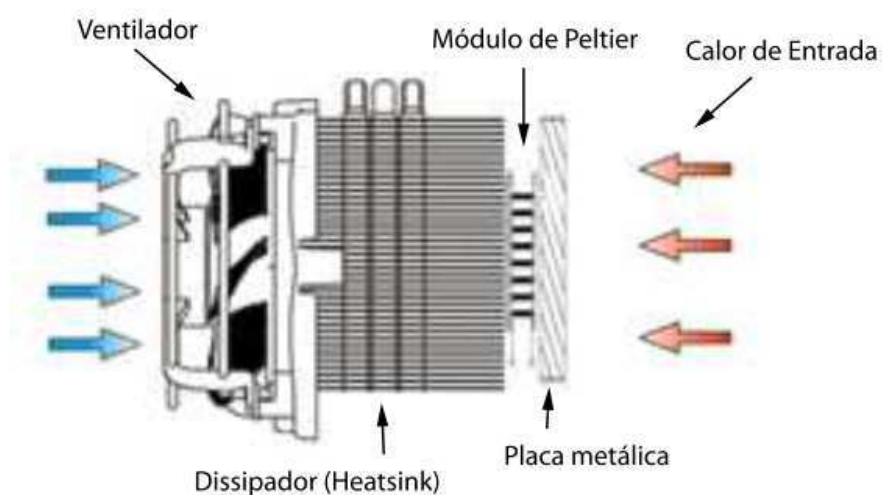


Figura 6 – Esquemático do acoplamento do Peltier, Fan e dissipador

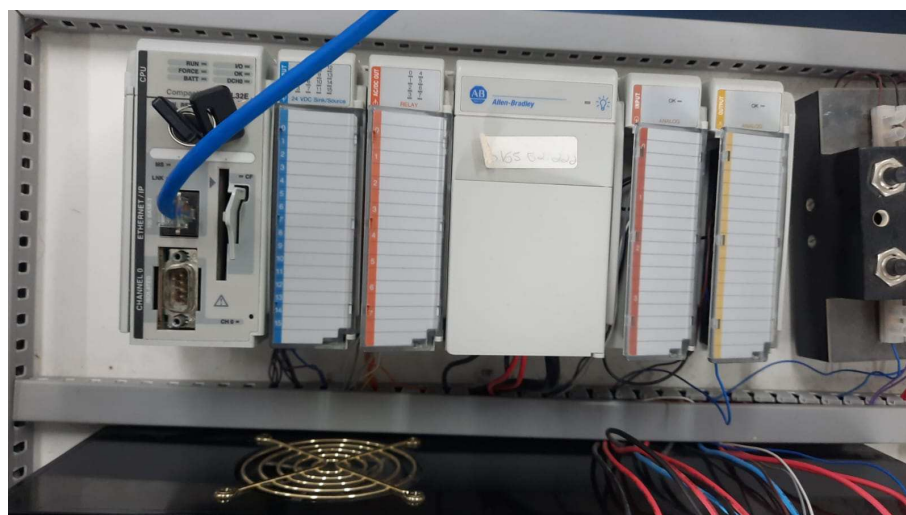


Figura 7 – Controlador

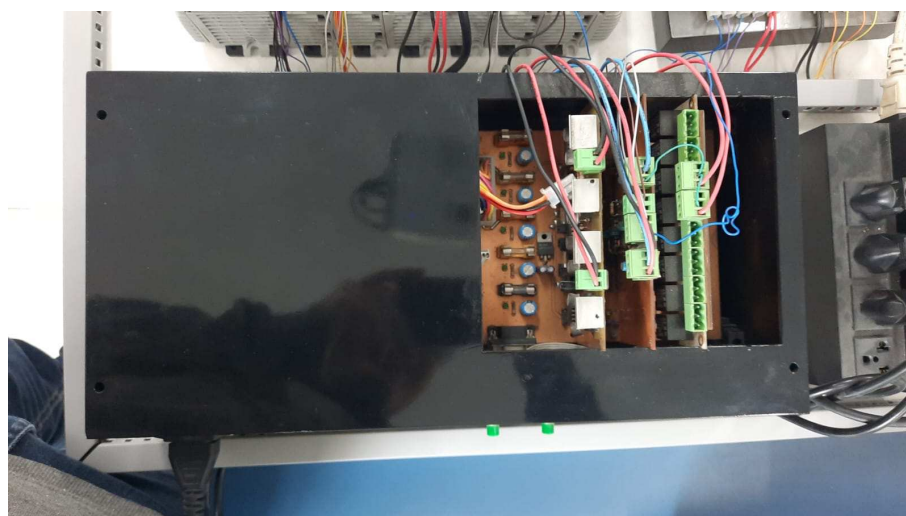


Figura 8 – PWM

os controladores programados e o software de controle ou outros dispositivos conectados à rede. O RSLinx possibilita a troca de informações em tempo real e suporta uma ampla gama de protocolos e redes industriais, como Ethernet/IP, ControlNet e DeviceNet. Graças ao RSLinx, o RSLogix 5000 pode se conectar aos CLPs e realizar transferências de dados, monitoramento e depuração em tempo real.



Figura 9 – *Softwares*

Além desses dois, outros programas para ajudar no desenvolvimento foram o *MatLab* e o *BR-Tuning* que foram usados para ajudar na modelagem do processo e cálculo dos parâmetros do controlador PID que são aplicados para o controle das malhas.

4 Atividades Realizadas

Durante o estágio supervisionado no LIEC, foram desenvolvidos atividades em cima da planta descrita como teste de malha aberta e fechada para desenvolvimento de um modelo e controle de temperatura adequado.

4.1 Comunicação OPC-DA

Uma das soluções mais utilizadas na automação industrial é a comunicação OPC. Sua criação ocorreu na década de 1990, quando a Microsoft introduziu as tecnologias COM e DCOM no setor industrial. Com base nessas tecnologias, foi desenvolvida uma solução chamada OPC. O OPC clássico, conhecido como OPC DA (*Data Access*), foi lançado em 1995 e passou por melhorias ao longo do tempo. Em 2003, surgiu o OPC UA (Unified Architecture), cuja primeira versão foi disponibilizada três anos depois. Entre as principais diferenças entre o OPC DA e o OPC UA, destaca-se o fato de que o OPC DA é compatível apenas com sistemas Windows.

O OPC Data Access (OPC DA) estabelece especificações para a comunicação entre cliente e servidor, permitindo a coleta de dados de processo em tempo real a partir de diferentes dispositivos, mantendo uma comunicação constante. No entanto, por ser baseado no protocolo DCOM, o OPC DA enfrenta algumas limitações herdadas desse protocolo, como a necessidade de configurar permissões de acesso nas máquinas envolvidas (tanto no servidor quanto no cliente OPC), além de requerer ajustes no Firewall para garantir a correta comunicação.

4.1.1 Configuração COM/DCOM

Nessa seção, será mostrado as configurações necessárias para estabelecer a comunicação OPC entre o dispositivo servindo como servidor e os clientes OPC, nesse caso aplicativos como o BR-Tuning e o *Matlab*.

Primeiramente, alguns requisitos para comunicação devem ser atendidos:

- Os dispositivos devem estar na mesma rede grupo de trabalho;
- As contas nos computadores, cliente e servidor, devem ter permissão de administrador, tendo o mesmo nome de usuário e mesma senha.

Um *software* que auxilia para o teste da comunicação e indicação de erros é o *OPCExpert*, com ele, é possível acessar as *tags* da comunicação, além do status da co-

municação em si, disponibilidade do servidor, entre outras coisas. Para o funcionamento adequado, é necessário que o aplicativo seja executado como administrador. Na figura 10, encontra-se a sua tela inicial.

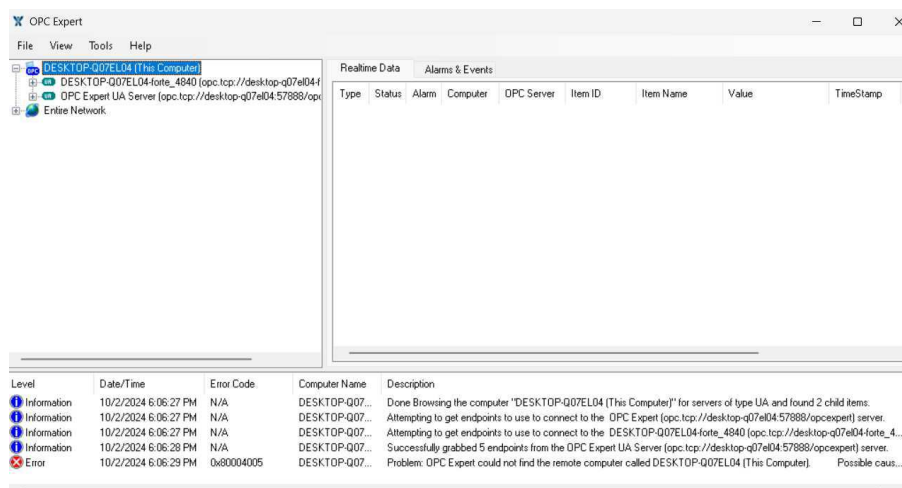


Figura 10 – OPCExpert

Depois da configuração feita, é necessário fazer modificações no *Firewall* para permitir a comunicação. Esse passo pode ser feito de duas maneiras diferentes. A maneira mais simples é desabilitar o *Firewall* nas máquinas, e não é necessário mais nenhuma configuração para o funcionamento do servidor. A segunda maneira envolve a permissão para comunicação de alguns programas pelo *Firewall*, a adição do *Opcenum* para conexão remota de clientes, além da criação de novas regras para liberação das portas TCP e UDP que são usadas pelo DCOM.

Um terceiro passo envolve a configuração DCOM. Porém, antes de ir para essa configuração, é necessário ativar a opção "Ativar a DEP só para programas e serviços essenciais do Windows" nas configurações avançadas do sistema. Feito isso, é ir nas "Opções de Segurança", que está dentro de "Política de Segurança Local", para modificar as configurações DCOM:

- Restrições de Acesso ao Computador na sintaxe de Definição do Descritor de Segurança (SDDL)
 - Clicando em "Editar Segurança";
 - Adiciona "Todos", "Interativo", "REDE", "SISTEMA" e "usuário local";
 - Habilita "Acesso Local" e "Acesso Remoto" de cada.
- Fazer o mesmo procedimento para "Restrições de Inicialização de Computador na sintaxe da Linguagem de Definição do Descritor de Segurança (SDDL)"
 - Para todo usuário adicionado, habilitar "Início Local", "Início Remoto", "Ativação Local" e "Ativação Remota".

- Ir em "Acesso à Rede:deixar que as permissões de todos os usuários sejam aplicadas a usuários anônimos"e habilitar;
- Ir em "Acesso à Rede:modelo de compartilhamento e segurança para contas locais"e selecionar "Clássico – os usuários locais são autenticados como eles próprios".

Com as configurações das "Opções de Segurança"feitas, o próximo passo é ir para "Atribuição de direitos do usuário", e acessar "Acesso a este computador pela rede"e adicionar o usuário local.

Indo para configuração DCOM, é necessário acessar o "Serviço de componente"pelo prompt com o comando "dcomcnfg". Ao entrar aqui, seleciona-se as "Propriedades"no menu de contexto, onde terá a opção "Segurança COM", nessa opção é necessário editar os padrões de "Permissões de Acesso"e "Permissões de Inicialização e Ativação". Na primeira edição, é adicionado o usuário local habilitando as opções de "Acesso Local"e "Acesso Remoto". Na segunda edição, o usuário local também é adicionado, habilitando as opções de "Início Local", "Início Remoto", "Ativação Local"e "Ativação Remota".

Todas as configurações anteriores foram feitas tanto na máquina servidora quanto na máquina cliente. Porém, existem configurações que só são necessárias em uma das máquinas.

Nesse passo as configurações só são feitas na máquina cliente, pois é necessário dar as permissões de acesso, inicialização e ativação da aplicação ao *Matlab*. Para isso, é acessado novamente o "Serviços de Componente", achando a aplicação do *Matlab*, é acessado as "Propriedades", e na "Aba Geral"é selecionado a opção "Conectar-se", na aba "Local"é selecionado "Executar o aplicativo neste computador", na aba "Segurança"é selecionado a opção "Personalizar"em "Permissões de Inicialização e Ativação"e "Permissões de Acesso". Na área de "Permissões de Inicialização e Ativação", são adicionados os usuários "TODOS", "INTERATIVO", "REDE", "SISTEMA"e o usuário local com as opções "Início Local", "Início Remoto", "Ativação Local"e "Ativação Remota"habilitadas em todos usuários adicionados. Na área de "Permissões de Acesso", os mesmos usuários são adicionados com as opções de "Acesso Local"e "Acesso Remoto"habilitadas. Na aba "Ponto de Extremidade", é adicionado "TCP/IP orientado a conexão". E, por fim, na aba "Identidade", é selecionado "Este Usuário", com as informações de usuário e senha.

No próximo passo, as configurações só são feitas na máquina servidora. É necessário acessar a janela de configuração DCOM pelo comando `dcomcnfg` no prompt, com essa janela aberta, e seguindo o caminho de Serviços de Componente-Computadores-Meu Computador, onde é aberto a pasta `configDCOM`. Dentro dessa pasta, é necessário achar o servidor OPC da aplicação(RSLinx) e acessar suas propriedades, nele são feitas todas as configurações realizadas para aplicação "*Matlab*".

4.1.2 Código no CLP

Como mostrado na figura 11, as instruções "MOV" movem dados de dois canais (localizados no módulo Local:3) para variáveis intermediárias Y1 e Y2. Essas variáveis podem representar valores brutos lidos de sensores.

A função CPT (Compute) normaliza os valores de Y1 e Y2, calculando Y1-norm e Y2-norm com base em expressões matemáticas:

$$Y1\text{-norm} = \frac{Y_1}{312.07} \quad \text{e} \quad Y2\text{-norm} = \frac{Y_2}{312.07}$$

Este cálculo é um processo comum para converter os valores de leitura brutos dos sensores em uma unidade real utilizável.

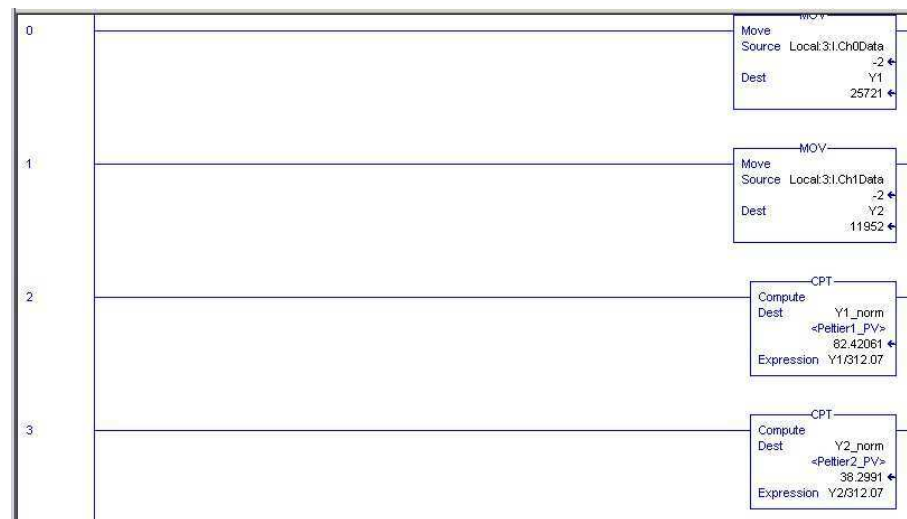


Figura 11 – Primeira parte do programa

Nas linhas 4 a 7, são feitas comparações usando a função CMP para verificar as variáveis "FecharMalha1" e "FecharMalha2", que representam o estado de controle (malha aberta ou malha fechada) das malhas 1 e 2. Caso a comparação com zero seja verdadeira, ativando o bloco "PID1.SWM" (*Switch Manual Mode*) da malha 1, colocando o *loop* PID1 em modo manual e passando o valor da MV escolhido, o mesmo acontece na linha 6 para a malha 2. Já na linha 5, caso "FecharMalha1" = 1, isso indica que a malha 1 deve estar fechada, ativando o bloco "PID1.SWM" novamente, colocando o *loop* PID1 em modo automático, e é calculado um valor para o setpoint, o qual é movido para PID1.SP (Setpoint do PID). Este valor é calculado com base na expressão Referencial*312.07, onde "Referencial" é uma variável que define o valor desejado para a malha, uma referência de temperatura, o mesmo acontece com a linha 7 para malha 2.

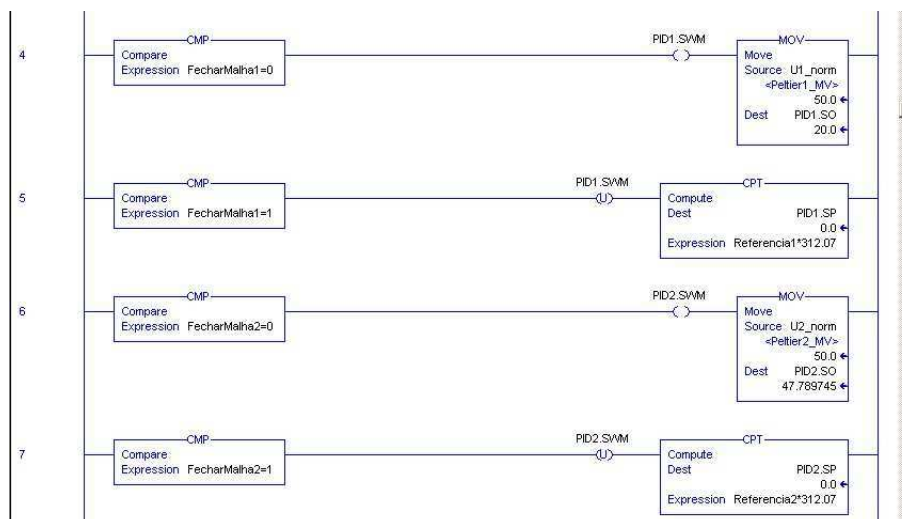


Figura 12 – Segunda Parte do Código

Nas linhas 8 e 9, estão as instruções PID. Os controladores PID1 e PID2 controlam os processos com base nas variáveis de processo Y1 e Y2, que foram obtidas dos sensores, mas também depende do modo de operação que estejam, caso a malha esteja aberta, não é feito um "controle" da temperatura, pois ela vai estar livre.

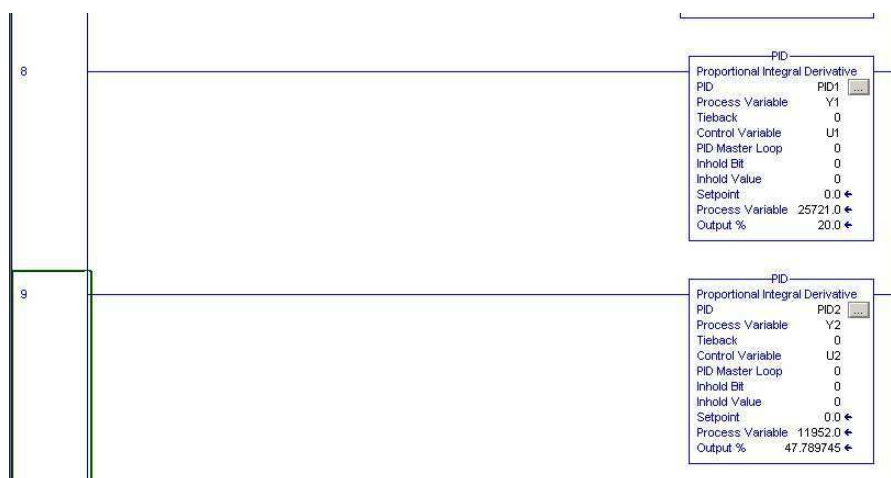


Figura 13 – Terceira parte do código

As linhas 10 a 13 do programa desempenham um papel crucial na lógica de controle de um sistema baseado em loops PID, sendo responsáveis por ajustar as saídas controladas das malhas de controle e enviar esses sinais para o hardware. Este bloco de código lida com o ajuste final da variável de controle das saídas normalizadas, garantindo que as ações necessárias sejam aplicadas corretamente aos atuadores físicos conectados ao controlador.

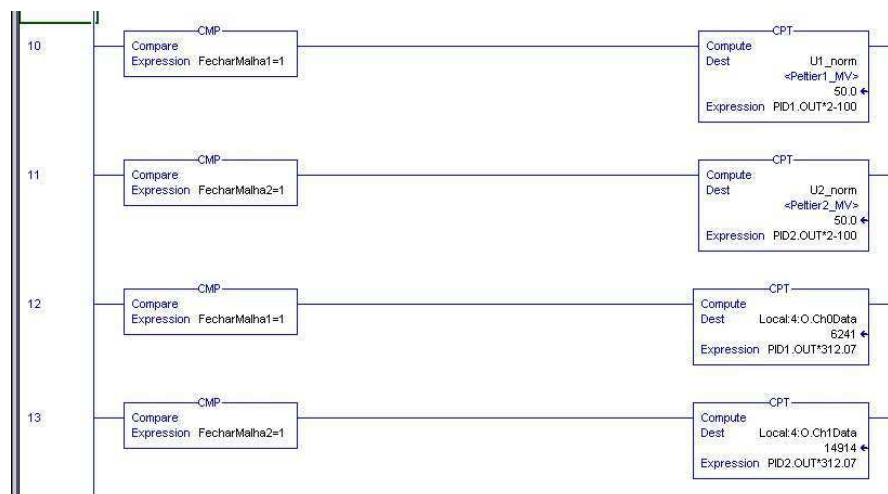


Figura 14 – Quarta parte do código

É possível acessar os valores dos parâmetros do PID ao clicar nos três pontos no seu bloco, como mostra a figura 15.

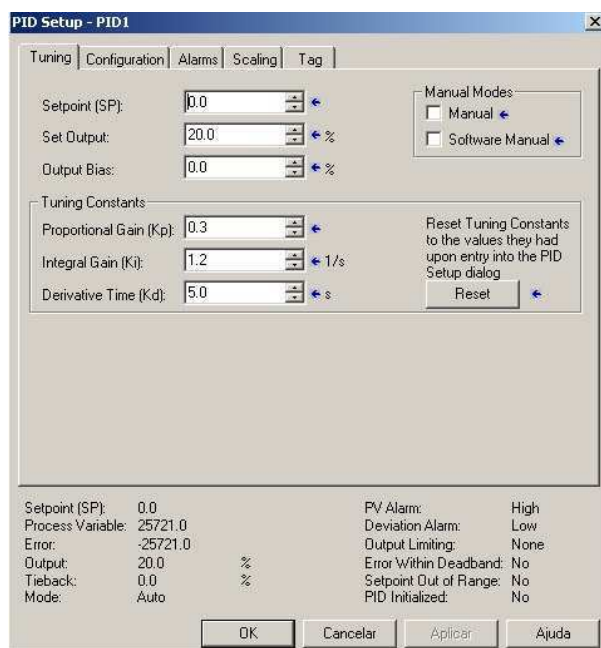


Figura 15 – PID

Nessa imagem, além de mostrar os parâmetros do PID, também é possível modificá-las. Outra informação importante que é possível visualizar é o modo de operação que o PID se encontra, nesse caso, o modo seria "Auto".

4.1.3 Experimentos

Com as configurações de comunicação e com o código feitos, é possível estabelecer uma conexão entre o servidor (*RSLinx*) e o cliente (*Matlab*), e dessa forma, é possível

visualizar graficamente e em tempo real as variáveis de temperatura em cada uma das malhas, além das variáveis manipuláveis que são enviadas para os atuadores. Para essa visualização, é necessário criar um programa para conexão com o servidor e para leitura das variáveis.

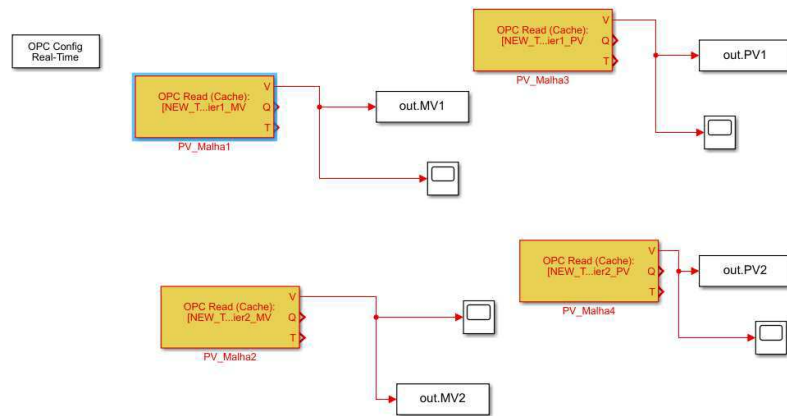


Figura 16 – Programa para leitura dos dados

Esse programa permite a análise gráfica em tempo real dos dados, além do salvamento dos mesmos para análises futuras. Porém, a mudança nos valores das MVs e dos estados do PID são feitas diretamente no *RSLogix 5000*. A imagem 17 mostra a visualização durante o experimento, onde os gráficos são gerados pela aplicação no *Matlab*, mas também é possível ver o valor da variável em tempo real no software onde foi feita a programação.

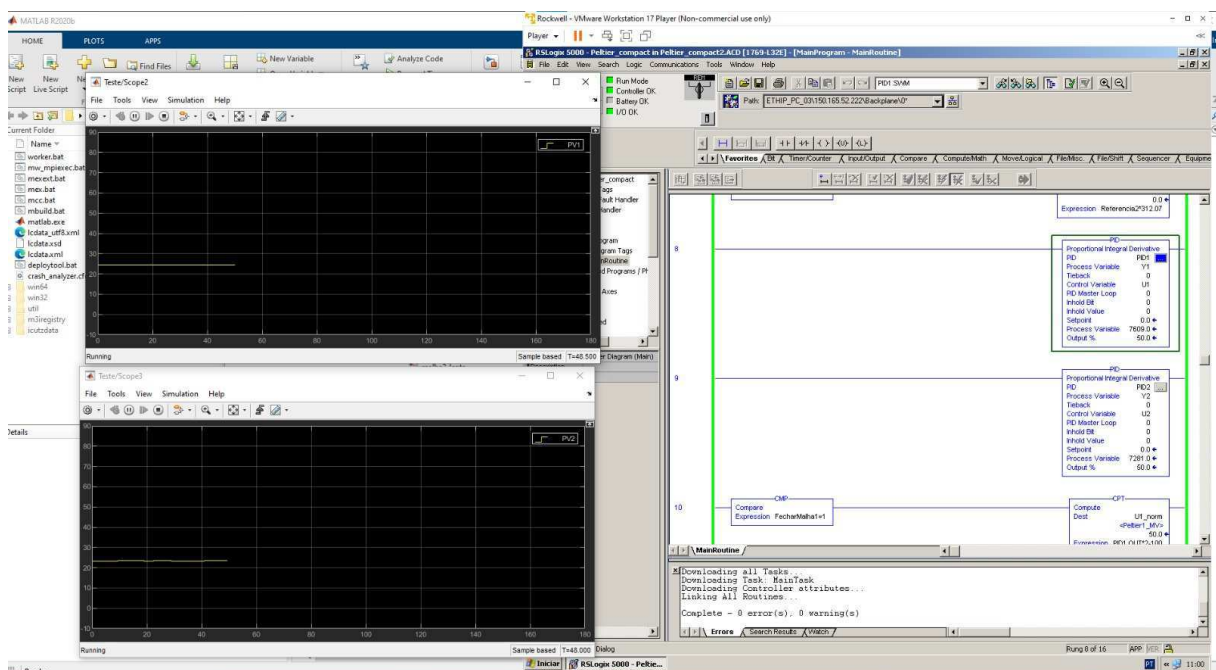


Figura 17 – Visualização durante o experimento

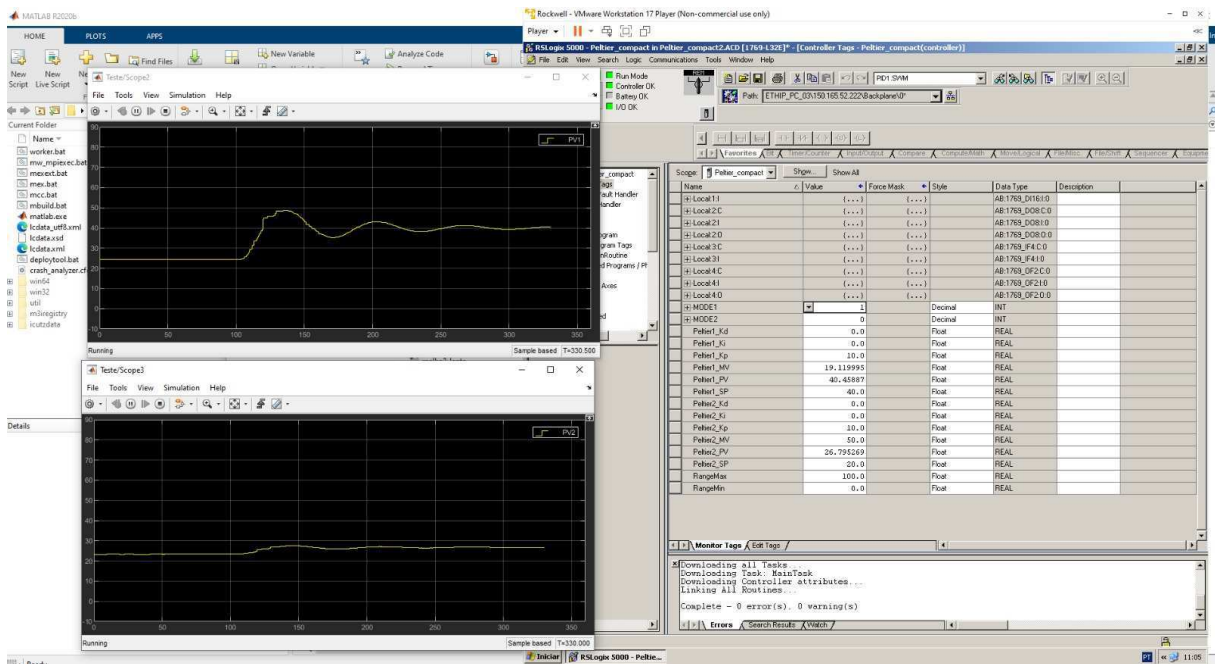


Figura 18 – Visualização das variáveis durante o experimento

Um primeiro teste essencial para extrair o modelo de uma planta é a resposta ao degrau, pois ela permite avaliar o comportamento dinâmico do sistema de forma simples e eficiente. Ao aplicar uma variação repentina (um "degrau") na entrada do sistema, é possível observar como a saída responde ao longo do tempo. A partir dessa resposta, é possível determinar parâmetros importantes, como o ganho estático, a constante de tempo e o tempo de resposta. Esses dados são fundamentais para criar um modelo matemático que represente a dinâmica da planta, permitindo o desenvolvimento de controladores adequados para otimizar seu desempenho. Porém, no primeiro momento, foi possível notar irregularidades na malha 1, as quais foram medições incorretas e ruidosas, além de um comportamento inverso do esperado. A questão do comportamento se resolve com a mudança das polaridades que alimentam o Peltier, visto que é a direção da corrente que dita o aquecimento ou resfriamento, enquanto que as medições incorretas eram um problema de mau contato na saída do sensor.

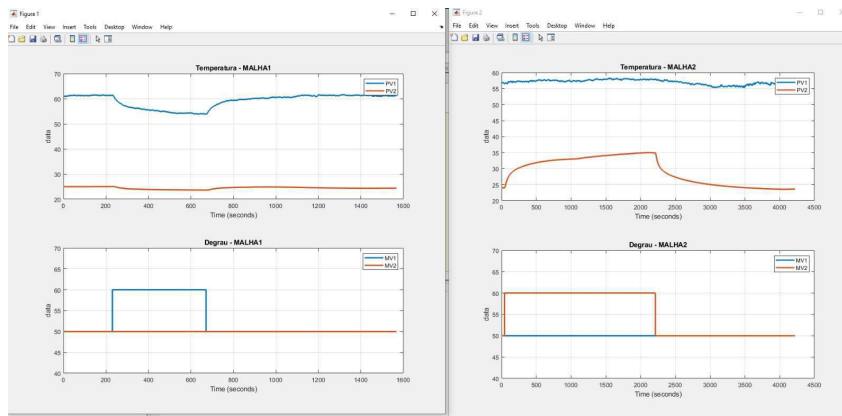


Figura 19 – Experimento antes das correções

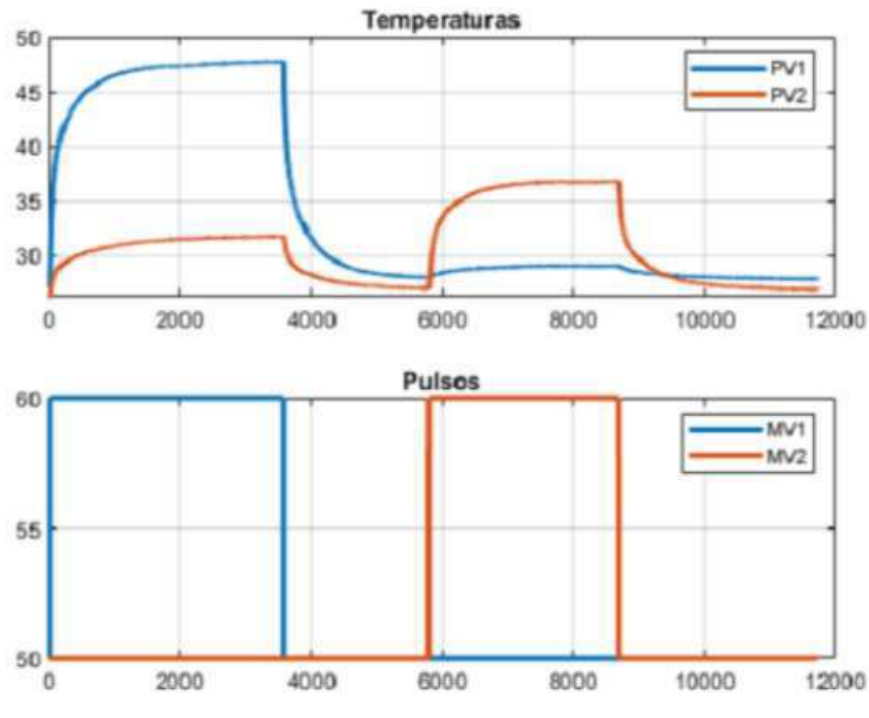


Figura 20 – Resposta ao degrau

Com base nesses dados, é possível elaborar um modelo para o sistema. Observando a curva, nota-se inicialmente um aumento exponencial seguido por uma elevação mais lenta. Isso indica a provável presença de um polo mais distante do eixo real no plano complexo, enquanto outro polo está mais próximo desse eixo. Além disso, é possível inferir que existe algum problema em uma das duas malhas, visto que a dinâmica deveria ser parecida, porém é notório a diferença nas amplitudes das respostas. Nesse ponto, a ferramenta "*System Identification*" do *Matlab* foi usada para estimar os parâmetros do modelo, resultando nos parâmetros da figura 21.

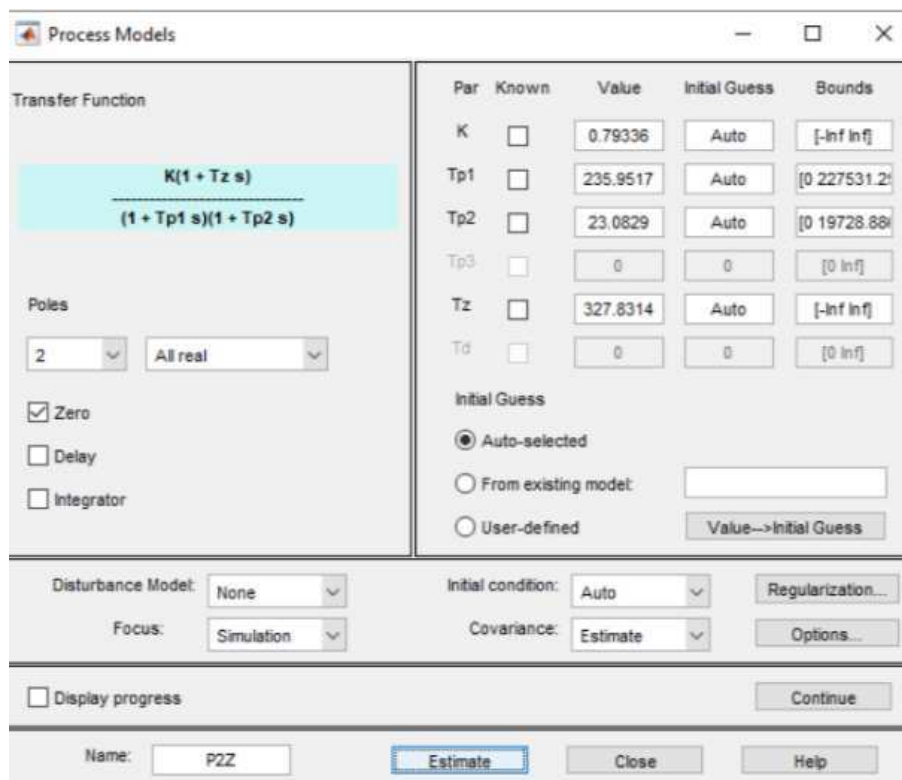


Figura 21 – System Identification

Feita a identificação do modelo, o próximo passo é construir um controlador PID, tal qual o controle seja feito de maneira satisfatória. Nesse ponto, outro aplicativo do *Matlab* foi usado, o *PID Tuner*. O que levou aos resultados dos parâmetros da figura 22.

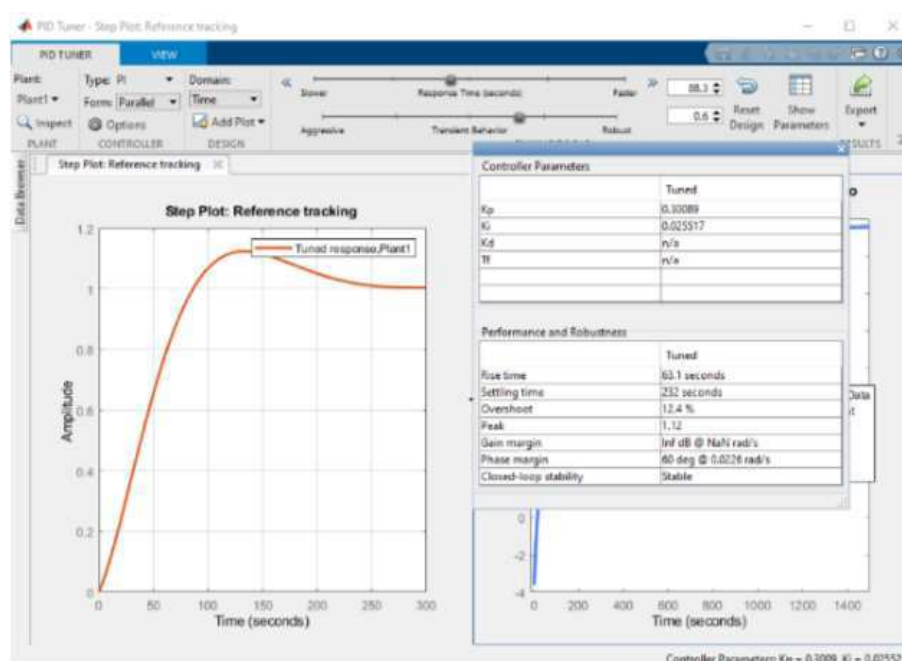


Figura 22 – Parâmetros com o *PID Tuner*

Apesar da resposta mostrada na figura 23, na realidade, ao aplicar esses parâmetros

de controle, tem-se uma resposta insatisfatória, onde a temperatura varia em torno do *set point*, como é possível ver na figura 21.

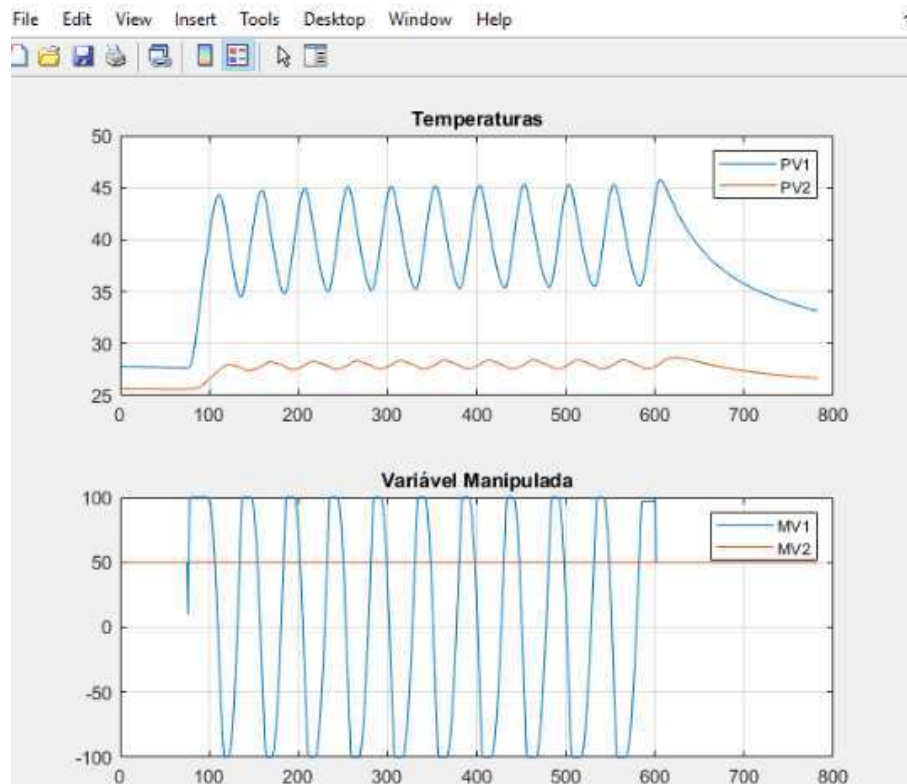


Figura 23 – Resposta ao controle PI com os parâmetros obtidos pelo *PID Tuner*

Como os resultados não foram bons, é necessário refazer a parametrização do controlador, logo, outro *software* foi utilizado, o *BR Tuning*. Nele, é possível escolher o método que o PID será estimado, nessa aplicação, foi usada mesma técnica anterior, a aplicação de um degrau na entrada, o que resultou nos parâmetros da figura 24.

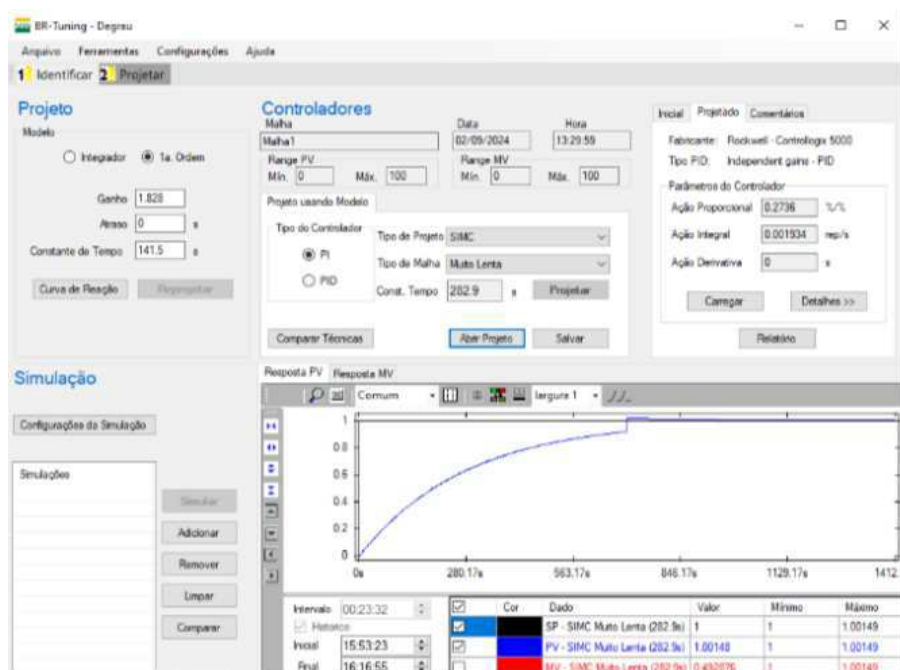


Figura 24 – Parâmetros obtidos pelo *BR Tuning*

Com esses novos parâmetros, o controle se mostrou mais eficaz, com a respostas estabilizando ao decorrer do tempo, como mostra a figura 25. Outro fator importante é que foi feita uma suposição de resposta "Muito Lenta" para malha no *BR Tuning*.

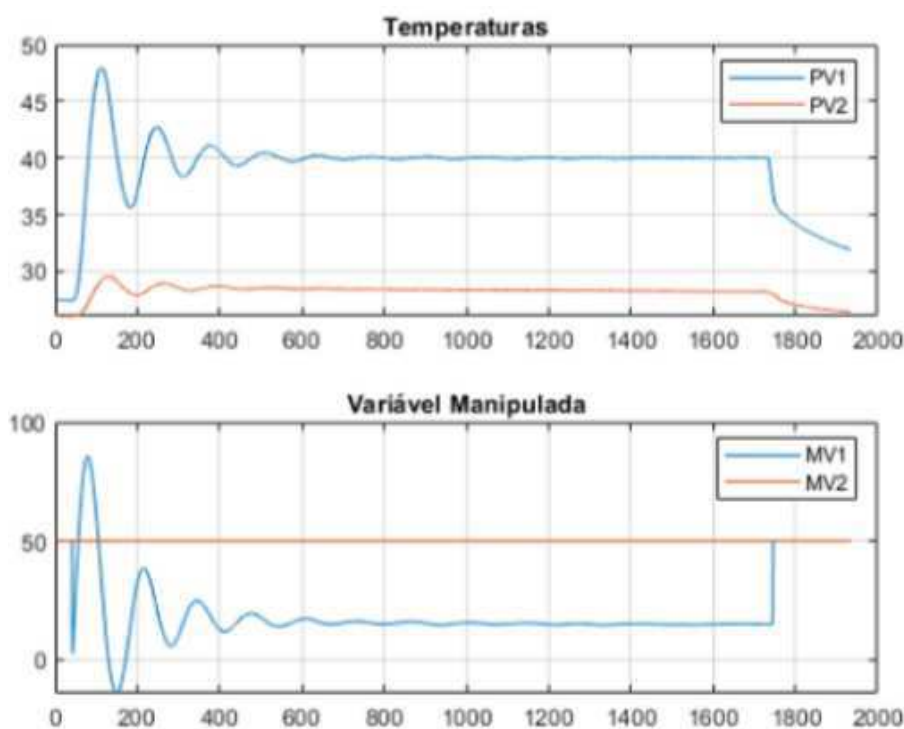


Figura 25 – Respostas com os novos parâmetros aplicados

O mesmo procedimento foi feito para malha 2, resultando em valores em parâ-

metros de controle mostrados pela figura 26, novamente supondo uma malha "Muito Lenta" para gerar esses valores.

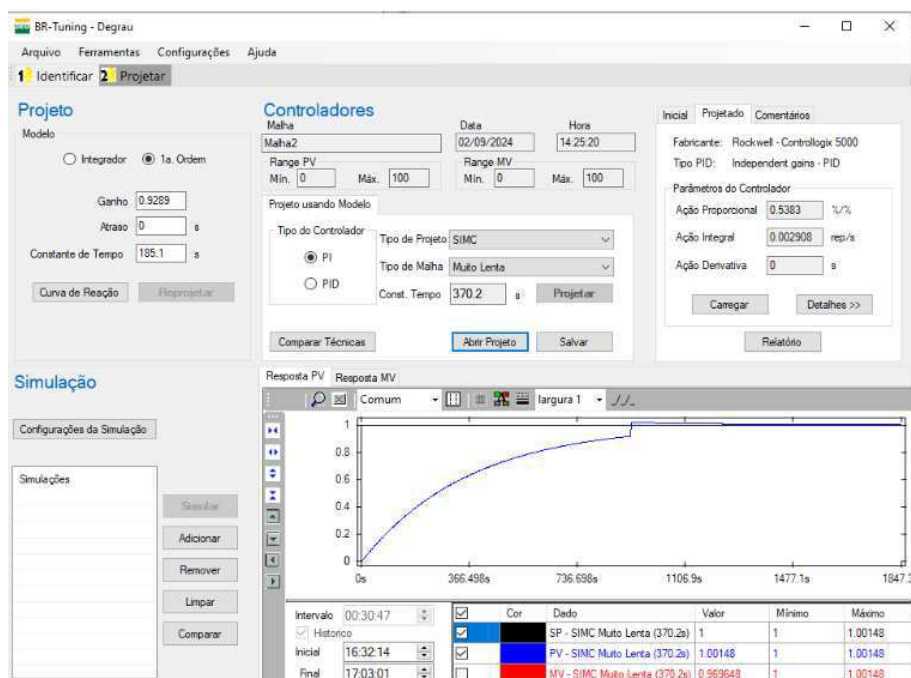


Figura 26 – Parâmetros obtidos para malha2

Aplicando esses valores no PID, é possível visualizar que a ação de controle é bem sucedida, visto que a saída segue o sinal de referência. Como mostrado na figura 27.

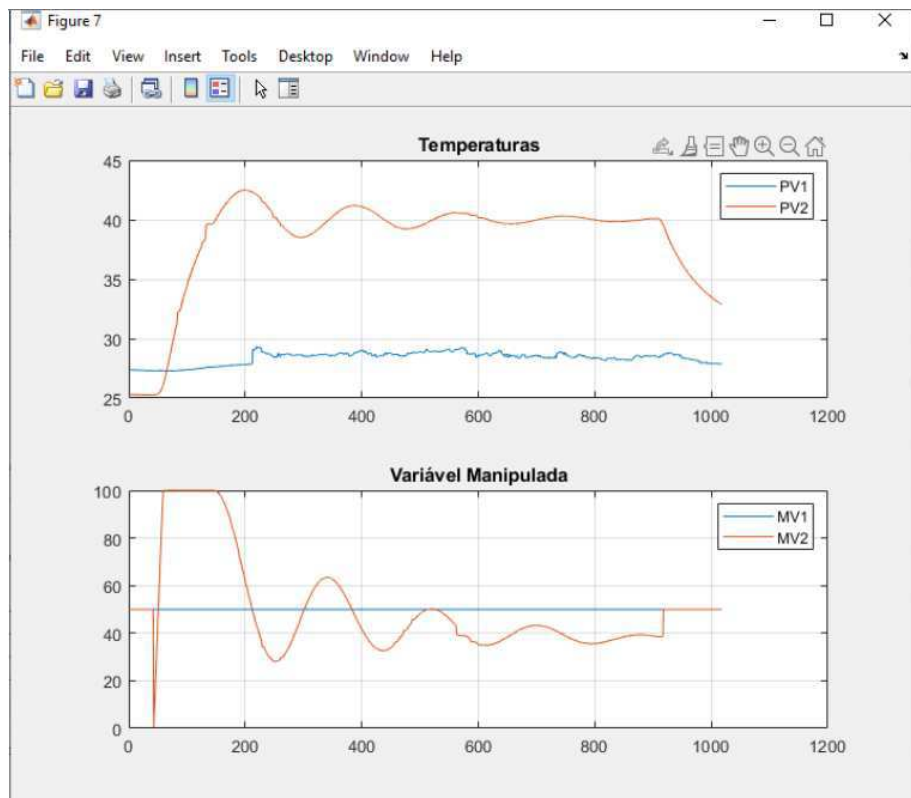


Figura 27 – Resultado da malha2

4.1.4 Testes para Melhoria do Controle

Agora, com a suposição de uma malha mais rápida para obtenção dos parâmetros dos PIDs, é possível novos parâmetros para novos testes para verificar se o controle é melhorado com essa nova suposição.

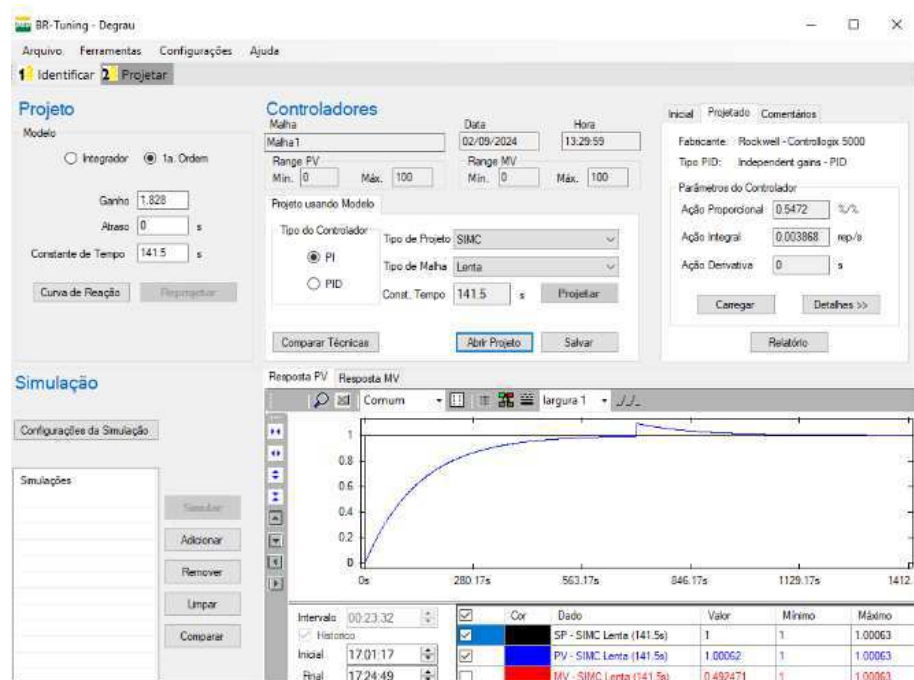


Figura 28 – Novos parâmetros com a suposição de uma malha1 mais rápida

Comparando os resultados, é notável que o primeiro resultado tem uma ação de controle menos brusca, e a PV estabiliza no *set point* mais rapidamente.

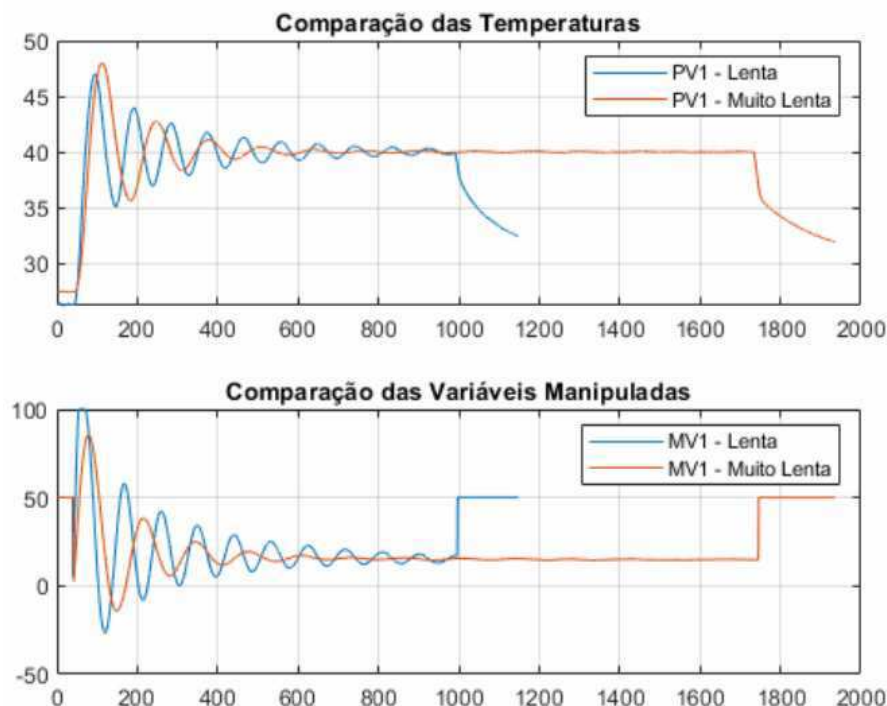


Figura 29 – Comparativo entre os resultados

Olhando para malha 2, agora com a suposição de uma malha mais rápida, o resultado para os novos parâmetros são vistos na imagem 28.

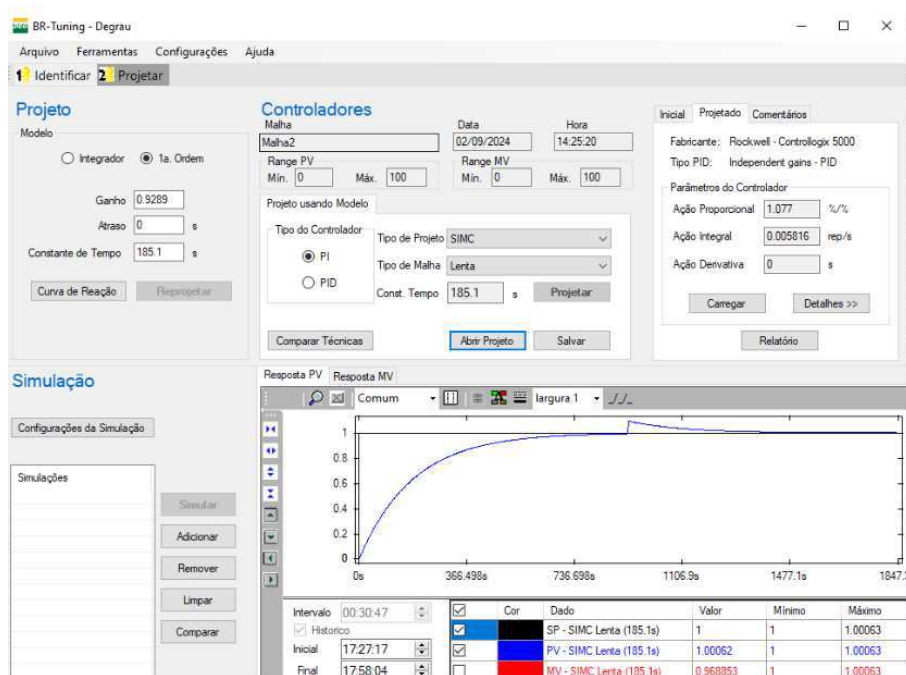


Figura 30 – Novos parâmetros para malha2

Porém, o novo controlador não satisfaz os requisitos de controle, visto que a PV não converge para o valor do *set point* ao longo do tempo, como é notável pela figura 31.

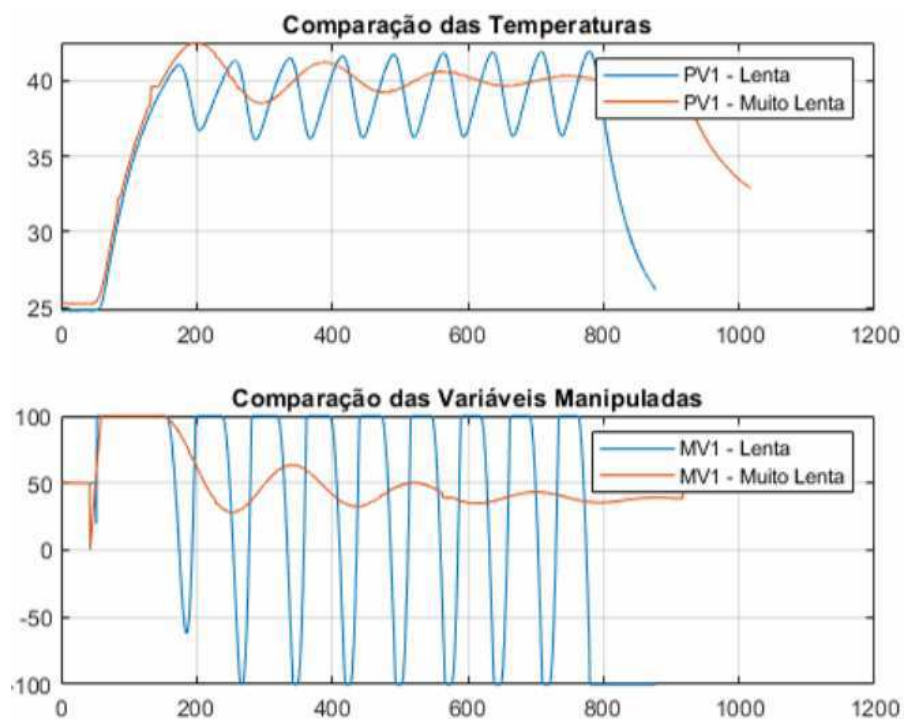


Figura 31 – Comparativo entre os resultados

5 Conclusão

O estágio supervisionado realizado no LIEC da Universidade Federal de Campina Grande proporcionou uma experiência prática valiosa, permitindo a aplicação de conceitos teóricos adquiridos ao longo do curso de Engenharia Elétrica. Através do desenvolvimento de um sistema de controle de temperatura utilizando módulos Peltier e sensores LM35, foi possível explorar as tecnologias de comunicação OPC-DA e configurar sistemas de controle PID utilizando softwares especializados como *Matlab* e *BR-Tuning*.

Durante o estágio, foram realizados testes e experimentos para modelagem do sistema, configurando adequadamente os parâmetros do controlador para obter uma resposta satisfatória. As atividades permitiram o aprimoramento de habilidades práticas, como a configuração de sistemas de automação e o ajuste de controladores, além de aprofundar o conhecimento em tecnologias específicas de comunicação industrial.

Os desafios encontrados ao longo do estágio, como a dificuldade inicial em estabilizar o controle e a necessidade de ajustes finos nos parâmetros dos PIDs, contribuíram significativamente para o aprendizado e para o desenvolvimento de competências cruciais para um engenheiro eletricista.

Por fim, o estágio não apenas complementou a formação acadêmica, mas também ampliou a visão sobre as aplicações práticas da engenharia elétrica para desafios futuros no campo da automação industrial.

6 Trabalhos Futuros

Como a planta ainda enfrenta problemas de funcionamento relacionados às diferenças entre as duas malhas, é essencial realizar um trabalho aprofundado para identificar as causas dessas questões. Para isso, serão necessários testes nos sensores, nos módulos Peltier e nos atuadores PWM.

Além disso, com a implementação dos novos softwares de desenvolvimento do sistema de controle, é viável estabelecer a comunicação com base no protocolo OPC-UA. Outra possibilidade seria desenvolver um sistema supervisor que utilize esse mesmo protocolo para comunicação.