

Pedro Cunha Garcia Lopes

# **Controle e supervisão de um deslocador pneumático utilizando controladores lógicos programáveis e rede Modbus**

Uberlândia, MG

2025

Pedro Cunha Garcia Lopes

# **Controle e supervisão de um deslocador pneumático utilizando controladores lógicos programáveis e rede Modbus**

Trabalho de Conclusão de Curso da Engenharia de Controle e Automação da Universidade Federal de Uberlândia - UFU - Campus Santa Mônica, como requisito para a obtenção do título de Graduação em Engenharia de Controle e Automação.

Universidade Federal de Uberlândia - UFU  
Faculdade de Engenharia Elétrica - FEELT

Orientador Prof. Dr. Renato Ferreira Fernandes

Uberlândia, MG

2025

*Este trabalho é dedicado aos meus pais Cecília e José Eduardo e à toda a minha família.  
Dedico também à minha namorada Sofia e aos meus amigos da “barraquinha”.*

# Agradecimentos

Gostaria de agradecer primeiramente aos meus pais, Cecília Cunha Garcia Lopes e José Eduardo Ferreira Lopes, por todo o amor e dedicação que sempre tiveram por mim. Agradeço por todas as oportunidades que me proporcionaram para estar onde estou hoje e todo o apoio nas minhas decisões acompanhado de todas as conversas e conselhos. Sei que nunca mediram esforços para me ver feliz e por isso sou muito grato.

À toda a minha família, meu irmão André, meus avós Fernando, Wilma, Helena e José Inês, minha tia Fernanda e minha prima Luiza, agradeço por todo o carinho e apoio demonstrado durante minha trajetória.

Expresso meu sincero apreço pelo meu professor e orientador Renato Ferreira Fernandes. Seu conhecimento e aconselhamento foram fundamentais para a realização desse projeto.

Aos demais professores da Faculdade de Engenharia Elétrica e em especial aos professores da graduação de Engenharia de Controle e Automação, manifesto minha gratidão por todo o ensino proporcionado e pela contribuição ao longo da minha graduação.

Agradeço também aos meus amigos que fiz durante o curso desde a primeira semana até o último período, foram todos muito especiais e tornaram esta etapa da minha vida mais proveitosa e mais leve. Alexandre, Gabriel, Gustavo, Leonardo Gotardo, Pedro Tizzo, Rodrigo e Vinícius, agradeço por todo o companheirismo e pelas amizades que vão se perpetuar durante toda a vida.

Por fim, agradeço a minha namorada Sofia Lopes Pinto Fernandes, por ter sido uma das maiores felicidades que a faculdade me proporcionou. Obrigado pelos momentos compartilhados, pelo amor demonstrado e o apoio incondicional sempre.

# Resumo

Os sistemas pneumáticos industriais, apesar de estarem há décadas sendo utilizados, ainda são de extrema importância em diversas aplicações na Indústria. Este trabalho teve como objetivo realizar o controle e supervisão de uma planta pneumática, através da comunicação com CLP e seu monitoramento através de um sistema supervisório com o intuito de reabilitar a planta que não estava em condições de funcionamento e aplicação de sua função didática. Esta planta, assim como os dispositivos e softwares industriais, se encontram no laboratório de Redes Industriais II do curso de Engenharia de Controle e Automação da Universidade Federal de Uberlândia. Desta forma, foi desenvolvido, a nível de controle, a conexão do CLP (WAGO Compact Controller 100), responsável pela leitura das variáveis de entrada e saída da planta pneumática, assim como pela lógica de controle do sistema. Além disso, através do protocolo de comunicação Modbus RTU, foi realizada toda a supervisão do sistema utilizando o sistema SCADA ELIPSE E3, assim como a obtenção dos dados necessários para a realização da identificação do sistema que será feita através do software MATLAB. Por fim, foi realizada a sintonização do PID, utilizando o método mais eficiente para esse tipo de aplicação, através da modelagem resultante da identificação da planta.

**Palavras-chaves:** Servopneumático; Controladores; Modbus; SCADA; Redes Industriais; PID.

# Abstract

Industrial pneumatic systems, although used for decades, remain highly relevant in a wide range of industrial applications. This work aimed to implement the control and supervision of a pneumatic plant by establishing communication with a PLC and monitoring through a supervisory system, with the goal of restoring the operation of the plant, which was previously non-functional and unable to fulfill its didactic purpose. The plant, along with its associated industrial devices and software, is located in the Industrial Networks II laboratory of the Control and Automation Engineering program at the Federal University of Uberlândia. At the control level, the WAGO Compact Controller 100 PLC was integrated, responsible for reading the plant's input and output variables and executing the control logic. Furthermore, system supervision was carried out using the ELIPSE E3 SCADA system via the Modbus RTU communication protocol, enabling data acquisition for system identification using MATLAB. Finally, the PID controller have been tuned based on the resulting model, employing the most suitable method for this type of application.

**Key-words:** Servo-Pneumatic; Controllers; Modbus; SCADA; Industrial Networks; PID.

# Listas de ilustrações

Figura 1 – Sistema pneumático genérico.	16
Figura 2 – Sistema de produção, armazenamento e condicionamento de ar comprimido.	17
Figura 3 – Esquemático de um sistema de consumo de ar.	18
Figura 4 – Pirâmide de automação.	19
Figura 5 – CLP - WAGO Compact Controller 100.	20
Figura 6 – Ciclo de varredura.	20
Figura 7 – Arquitetura básica do CLP.	21
Figura 8 – Exemplo de lógica Ladder.	22
Figura 9 – Exemplo de linguagem ST.	23
Figura 10 – Tela de supervisão do ELIPSE E3.	24
Figura 11 – Gráfico com o resultado da pesquisa sobre os protocolos mais utilizados em 2023.	25
Figura 12 – Diagrama exemplificando uma solicitação do tipo unicast.	26
Figura 13 – Diagrama exemplificando uma solicitação do tipo broadcast.	26
Figura 14 – Estrutura de um Frame Modbus no modo de transmissão RTU.	27
Figura 15 – Estrutura de um Frame Modbus no modo de transmissão TCP/IP.	28
Figura 16 – Tipos de funções Modbus.	29
Figura 17 – Sistema de controle em malha fechada com realimentação negativa.	31
Figura 18 – Controlador PID.	31
Figura 19 – Processos para identificação de um sistema.	33
Figura 20 – System Identification Tool – Matlab.	35
Figura 21 – Sistema com oscilação periódica.	36
Figura 22 – Modelo de um processo segundo o método IMC.	39
Figura 23 – Visão Geral do Sistema.	43
Figura 24 – Visão real da planta.	44
Figura 25 – Visão técnica da válvula.	44
Figura 26 – Curva característica de vazão por tensão.	45
Figura 27 – Sensor de posição FESTO TLH225.	45
Figura 28 – Entradas e saídas do CLP WAGO Compact Controller 100.	47
Figura 29 – Mapeamento das Variáveis Físicas.	48
Figura 30 – Configuração Parâmetros Modbus.	48
Figura 31 – Mapeamento das Variáveis Físicas Modbus.	49
Figura 32 – Variáveis declaradas no código Ladder.	49
Figura 33 – Primeiro Bloco de Execução em ST.	50
Figura 34 – Bloco PID da Linguagem Ladder.	51

Figura 35 – Segundo Bloco de Execução em ST. . . . .	51
Figura 36 – Conexão Cliente Servidor do Supervisório com o CLP. . . . .	52
Figura 37 – Tags do Sistema Supervisório. . . . .	52
Figura 38 – Tela do Sistema Supervisório. . . . .	53
Figura 39 – Leitura da saída do sensor de posição pelo multímetro. . . . .	55
Figura 40 – Leitura da saída do sensor de posição pelo CLP. . . . .	55
Figura 41 – Gráfico do controle realizado com parâmetros empíricos. . . . .	56
Figura 42 – Qualidade das variáveis utilizadas no supervisório. . . . .	57
Figura 43 – Leitura da entrada da válvula pelo multímetro. . . . .	57
Figura 44 – Leitura da entrada da válvula pelo supervisório. . . . .	58
Figura 45 – Dados coletados para identificação do sistema com degrau 5300 e 4700.	59
Figura 46 – Ferramenta de Identificação de Sistemas. . . . .	59
Figura 47 – Comparação do modelo estimado com os dados reais. . . . .	60
Figura 48 – Resposta ao degrau do controlador com $\lambda = 1$ . . . . .	61
Figura 49 – Resposta ao degrau do controlador com $\lambda = 0,5; 1; 2; 3; 4; 5$ . . . . .	62
Figura 50 – Gráfico do controle realizado com parâmetro de sintonia onde $\lambda = 1$ . .	63
Figura 51 – Gráfico do controle realizado com parâmetro de sintonia onde $\lambda = 2$ . .	63

# **Lista de tabelas**

Tabela 1 – Parâmetros do método em malha fechada de Ziegler e Nichols. . . . .	36
Tabela 2 – Parâmetros do método em malha aberta de Ziegler e Nichols. . . . .	37
Tabela 3 – Método de integral do erro. . . . .	37
Tabela 4 – Valores das constantes da equação. . . . .	38
Tabela 5 – Valores das constantes da equação para sintonia do PID. . . . .	38
Tabela 6 – Sintonia de controlados com tempo morto segundo Skogestad. . . . .	40
Tabela 7 – Sintonia de controlados sem tempo morto segundo Rivera. . . . .	41

# **Lista de abreviaturas e siglas**

ASCII	American Standard Code for Information Interchange
bps	Bits por segundo
CLP	Controladores Lógico Programáveis
CPU	Central Processing Unit
CRC	Cyclic redundancy check
FT	Função de transferência
Gbps	Gigabits por segundo
IAE	Integral do erro absoluto
IHM	Interface Homem Máquina
IMC	Internal Model Control
ISE	Integral do erro quadrático
ITAE	Integral do erro absoluto ponderado pelo tempo
ITSE	Integral do erro quadrático ponderado pelo tempo
I/O	Input / Output (Entrada / Saída)
kbps	Quilobit por segundo
Kc	Ganho estático do controlador
Kp	Ganho proporcional do controlador
LD	Ladder Diagram
MBAP	MODBUS Application Protocol
Mbps	Megabits por segundo
MV	Manipulated variable
PID	Proporcional Integral Derivativo
POU	Program Organization Unit

PV	Process Value
RS-232	Recommended Standard 232
RS-485	Recommended Standard 485
RTU	Remote Terminal Unit
SCADA	Supervisory Control and Data Acquisition
SIMC	Skogestad Internal Model Control
SP	Set Point
ST	Structured Text
TCP	Transmission Control Protocol
Z&N	Ziegler-Nichols

# Listas de símbolos

$\omega_n$  Frequência natural de um sistema

$\zeta$  Fator de amortecimento do sistema

$K_c$  Ganho proporcional

$T_i$  Constante de tempo integral

$T_d$  Constante de tempo derivativa

$\tau$  Constante de tempo

$\theta$  Tempo morto

$\lambda$  Critério de desempenho

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>14</b>
1.1	Justificativas	15
1.2	Objetivos	15
1.2.1	Objetivos Específicos	15
<b>2</b>	<b>REFERENCIAIS TEÓRICOS</b>	<b>16</b>
2.1	<b>Sistemas Pneumáticos</b>	<b>16</b>
2.1.1	Sistemas de Produção e Distribuição de Ar Comprimido	17
2.1.2	Sistemas de Consumo de Ar	17
2.2	<b>Automação Industrial</b>	<b>18</b>
2.2.1	Nível de chão de fábrica	19
2.2.2	Nível de controle	19
2.2.2.1	Funcionamento do Controlador Lógico Programável	20
2.2.2.2	Arquitetura do Controlador Lógico Programável	21
2.2.2.3	Linguagem de programação Ladder	21
2.2.2.4	Linguagem de Programação ST (STRUCTURED TEXT)	22
2.2.3	Nível de supervisão	23
2.3	<b>Redes Industriais</b>	<b>24</b>
2.3.1	Protocolo Modbus	25
2.3.1.1	Modbus RTU	27
2.3.1.2	Modbus TCP/IP	27
2.3.1.3	Tipos de Dados Lógicos e Funções do Modbus	28
2.4	<b>Sistemas de Segunda Ordem</b>	<b>29</b>
2.5	<b>Controlador PID</b>	<b>30</b>
2.5.1	Ação Proporcional	32
2.5.2	Ação Integral	32
2.5.3	Ação Derivativa	32
2.6	<b>Identificação de Sistemas</b>	<b>32</b>
2.7	<b>Sintonia de controladores PID</b>	<b>35</b>
2.7.1	Método de Ziegler e Nichols	35
2.7.2	Método de Integral do Erro	37
2.7.3	Método IMC (Internal Model Control)	38
2.7.3.1	Método IMC para Plantas de 2 <sup>a</sup> Ordem	41
2.7.3.2	Adaptação para Plantas em Forma Polinomial	41
<b>3</b>	<b>METODOLOGIA</b>	<b>43</b>

<b>3.1</b>	<b>Setup Experimental</b>	<b>43</b>
3.1.1	Atuadores e Sensores	44
3.1.2	Sistema de Controle	46
3.1.3	Sistema de Supervisão	47
<b>3.2</b>	<b>Configuração Controlador Lógico Programável</b>	<b>47</b>
3.2.1	Lógica de Controle	49
<b>3.3</b>	<b>Sistema Supervisório</b>	<b>51</b>
3.3.1	Comunicação Modbus	51
3.3.2	Tela do Sistema Supervisório	53
<b>4</b>	<b>RESULTADOS E DISCUSSÕES</b>	<b>54</b>
<b>4.1</b>	<b>Aquisição e Envio de Dados</b>	<b>54</b>
4.1.1	Validação Comunicação CLP e Planta	54
<b>4.2</b>	<b>Controle Local</b>	<b>56</b>
<b>4.3</b>	<b>Validação da Comunicação com o Supervisório</b>	<b>56</b>
<b>4.4</b>	<b>Identificação do Sistema e Sintonia do Controlador</b>	<b>58</b>
<b>4.5</b>	<b>Resultados da Sintonia no Controlador PID</b>	<b>62</b>
<b>5</b>	<b>CONCLUSÃO</b>	<b>64</b>
	<b>REFERÊNCIAS BIBLIOGRÁFICAS</b>	<b>65</b>

# 1 Introdução

O avanço tecnológico tem impulsionado o uso de aplicações que exigem alta precisão e confiabilidade, características essenciais em diversos segmentos da indústria. Na área da automação e da robótica, destacam-se os sistemas pneumáticos. Esses sistemas se sobressaem por oferecerem baixo custo de implantação, manutenção simples, componentes robustos, além de serem de fácil aplicação e, principalmente, representarem uma alternativa limpa e ambientalmente sustentável ([DOMINGUES, 2019](#)).

Quando falamos sobre sistemas pneumáticos de posicionamento, sua função é realizada através de ar comprimido e de alguns componentes básicos como um atuador e uma ou várias válvulas de controle. Outros dispositivos que geralmente encontramos nesse tipo de sistema, são os sensores, de posição por exemplo, e unidades de controle, para que se possa realizar a supervisão e controle do sistema.

Os motivos para se instalar um sistema pneumático de posicionamento em uma linha industrial são variados, mas se deve principalmente aos seus benefícios, os quais podemos citar: a possibilidade de incremento da produção de maneira fácil e com baixo investimento, aumento na produtividade ao se comparar com a mesma atividade sendo realizada por pessoas, baixo custo, leves e práticos para montagem e desmontagem, uma ótima relação peso potência, entre outros. Também devemos considerar as desvantagens, que podemos citar: os sistemas pneumáticos possuem não linearidade causada principalmente pelo ar comprimido, ocorrência de vazamentos no sistema e pela zona morta.

Podemos encontrar diversos estudos relacionados a sistemas pneumáticos, principalmente relacionados a dispositivos de posicionamento, estudando seu comportamento, maneiras de supervisionar e maneiras de realizar a sintonia de controladores. Alguns autores que podemos destacar são: ([SANTOS, 2019](#)), ([MENDOZA et al., 2006](#)), ([EURICH, 2014](#)).

Apesar de vários estudos já presentes ao longo de vários anos, ainda existem questões que podemos explorar e desenvolver algum estudo sobre, principalmente no que se trata em relação a utilização de controladores cada vez mais modernos e a identificação de sistemas e a sintonia de controladores. Dentro desse contexto, que esse trabalho entra, buscando contribuir com este tema nas questões de identificação do sistema de posicionamento servo pneumático, sintonia do controlador PID utilizado, comunicação entre dispositivos e supervisionamento.

## 1.1 Justificativas

Este trabalho tem por motivação entender o funcionamento de uma planta pneumática e como podemos utilizar este tipo de dispositivo em um âmbito industrial, uma vez que, como citado anteriormente, este tipo de sistema está amplamente presente em aplicações práticas no mercado. Entendendo e monitorando seu comportamento, de modo a projetar seu acionamento de forma adequada. Apesar de se tratar de uma planta didática, seu funcionamento se assemelha muito ao que encontramos em indústrias atualmente, alguns exemplos são indústrias farmacêuticas, alimentícias, automotivas.

No âmbito acadêmico, o estudo realizado nesse projeto incrementa discussões relacionadas a este tipo de equipamento. Ademais, por se tratar de uma planta instalada dentro de uma universidade, este estudo procura incrementar as possibilidades de aplicação da teoria no contexto das aulas práticas e auxiliar nas disciplinas ligadas ao tema. Uma vez que a planta havia sido, em partes, desmontada e não estava sendo utilizada com o propósito de aprendizado nas disciplinas de sistemas de controle, informática industrial e redes industriais, viu-se necessário reabilitar a planta para atender essa função didática. Por envolver CLPs, sistema supervisório, modelagem e identificação de sistema e controle, se torna um estudo rico no âmbito multidisciplinar no curso de engenharia de controle e automação.

## 1.2 Objetivos

O objetivo deste trabalho é realizar o controle e supervisão de uma planta pneumática, através da comunicação com CLP e seu monitoramento através de um sistema supervisório com o intuito de reabilitar a planta que não estava em condições de funcionamento e aplicação de sua função didática.

### 1.2.1 Objetivos Específicos

- Montagem da planta e atestar seu funcionamento;
- Conexão da planta ao CLP e aquisição e envio de dados;
- Desenvolver a lógica de controle e aplicá-lo localmente;
- Comunicação com o sistema supervisório e desenvolvimento do mesmo;
- Identificação do sistema;
- Sintonizar controlador PID e realizar ajustes;

## 2 Referenciais Teóricos

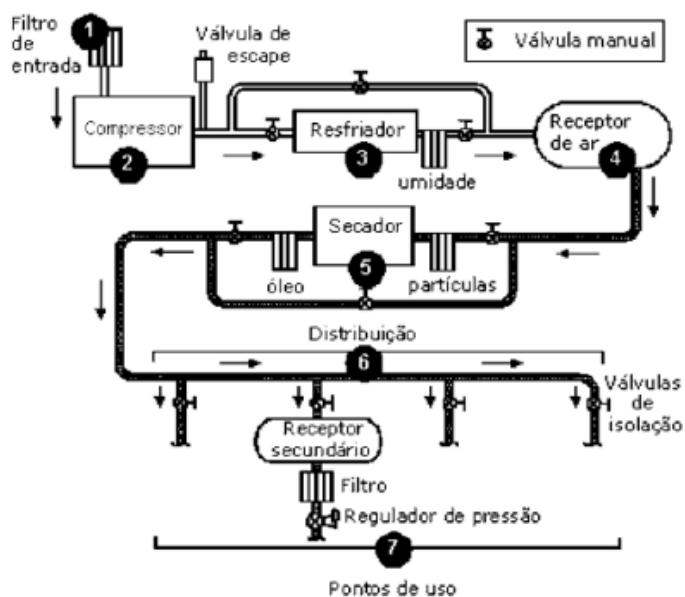
Neste capítulo serão apresentados alguns fundamentos teóricos cujos conhecimentos são necessários para uma melhor compreensão do desenvolvimento a ser realizado, abordando desde o nível físico até o nível lógico do projeto. Serão abordados os fundamentos de sistemas pneumáticos, automação industrial, modelagem de sistemas e controladores PID.

### 2.1 Sistemas Pneumáticos

O ramo da pneumática é responsável pelo estudo sistemático da utilização do ar comprimido na tecnologia de acionamento, comando e controle de sistemas automáticos. Esses sistemas são responsáveis por transformar a energia proveniente do ar comprimido em alguma forma de trabalho a ser realizado, através de dispositivos como válvulas ou atuadores pneumáticos (SANTOS, 2019).

Segundo (FABIÁN, 2017) a composição dos sistemas pneumáticos varia de acordo com sua finalidade, podendo assumir diferentes configurações. Entre seus elementos, alguns são indispensáveis para garantir o funcionamento adequado do circuito, como os filtros. Outros têm a função de controlar o fluxo e a direção do ar comprimido, a exemplo das válvulas. Já os atuadores são os componentes que efetivamente realizam o trabalho mecânico, convertendo a energia pneumática em movimento na máquina onde estão instalados. Na Figura 1 a seguir apresenta-se um circuito pneumático genérico.

Figura 1 – Sistema pneumático genérico.

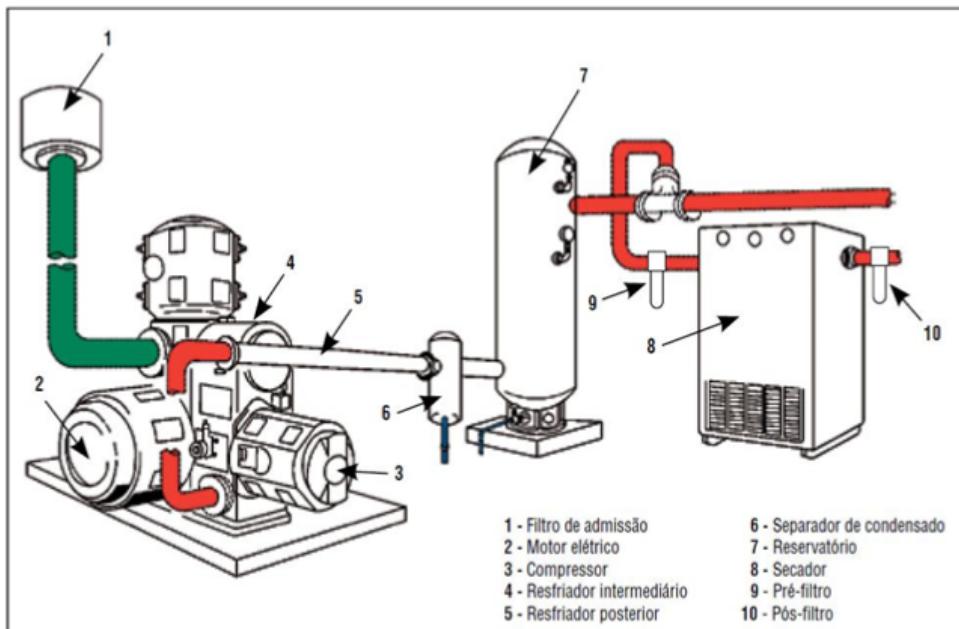


Fonte: (FABIÁN, 2017).

### 2.1.1 Sistemas de Produção e Distribuição de Ar Comprimido

Os circuitos pneumáticos dependem de uma fonte de ar comprimido que seja capaz de manter uma pressão constante e atender à vazão exigida pelos seus componentes. Essa fonte é composta por três unidades principais: produção, condicionamento do ar comprimido e distribuição. Sendo estas unidades representadas na figura 2.

Figura 2 – Sistema de produção, armazenamento e condicionamento de ar comprimido.



Fonte: ([TAFFAREL, 2015](#)).

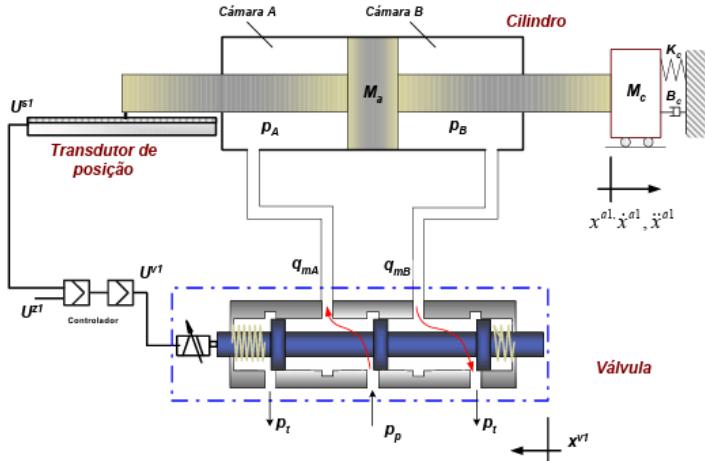
### 2.1.2 Sistemas de Consumo de Ar

Os sistemas pneumáticos de consumos de ar são muito diversos, sendo utilizados em diversos tipos de aplicações industriais, alguns exemplos são válvulas, sensores pneumáticos, cilindros, pistões e mesas pneumáticas.

Dependendo da finalidade, esses sistemas podem assumir diversas configurações dependendo dos componentes a serem utilizados e da aplicação final, algumas delas que estão em destaque atualmente são máquinas agrícolas, robôs manipuladores, lemes de aeronaves e suspensões ativas ([EURICH, 2014](#)). A Figura 3 representa um sistema pneumático de posicionamento utilizado em diversos tipos de aplicações industriais.

O funcionamento do sistema se inicia com a aplicação de uma tensão na solenoide da servoválvula, gerando assim um deslocamento em seu carretel. O carretel se desloca de forma proporcional ao valor da tensão aplicada, gerando assim a abertura ou fechamento dos orifícios de passagem de ar, permitindo que o ar passe para uma das cavidades do cilindro ([EURICH, 2014](#)).

Figura 3 – Esquemático de um sistema de consumo de ar.



Fonte: ([MENDOZA et al., 2006](#)).

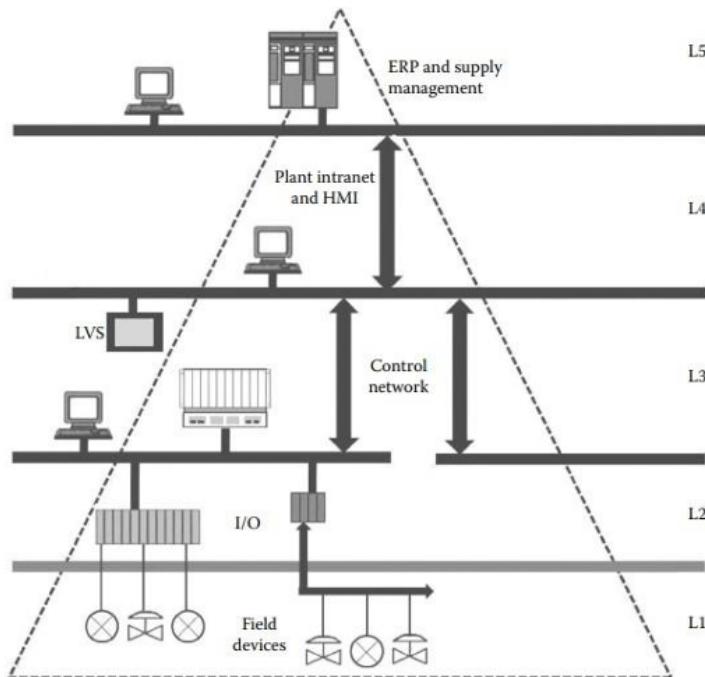
Como resultado do movimento do carretel, é gerado uma diferença de pressão entre as camadas do cilindro, provocando o deslocamento de seu êmbolo. O transdutor de posição, acoplado ao êmbolo do cilindro, registra o deslocamento do cilindro e transmite um sinal elétrico, sendo este sinal proporcional ao deslocamento. O sinal é utilizado para definir a posição do pistão, e podendo ser utilizado em lógicas de controle para controlar o movimento do sistema.

## 2.2 Automação Industrial

A automação industrial normalmente é classificada em diferentes níveis de automação, sendo estes desde o nível de chão de fábrica, o mais baixo, até o mais alto, denominado nível corporativo. Essa classificação pode ser vista representada na Figura 4 ([SEN, 2021](#)).

De acordo com a pirâmide da Figura 4, podemos observar que no nível mais baixo, normalmente denominado chão de fábrica, da pirâmide se encontram os sensores e atuadores responsáveis pelo processo de produção. Logo acima, temos o nível de I/O, onde é realizado o gerenciamento e troca dos dados dos sensores e atuadores com os controladores. Acima deste, se encontra nível de controle, correspondendo aos Programmable Logic Controller (PLC ou, em português, CLP), que atuam no controle de toda a planta, conectando o nível 2 com os níveis superiores através do envio e recebimento de sinais. Ferramentas e softwares de engenharia e programas de manutenção programada são exemplos do que é de responsabilidade do nível 4, denominado Human-Machine Interface (HMI ou, em português, IHM). Neste nível, qualquer variável de importância deve estar à visualização do operador ou responsável da planta, assim como alertá-lo em caso de undershoot/overshoot de alguma variável do processo. Por último, o quinto nível, de gerenciamento da empresa ou da cadeia de suprimentos, onde é possível ter uma visualização geral de todo o sistema produtivo, de modo a auxiliar na tomada de decisões ([SEN, 2021](#)).

Figura 4 – Pirâmide de automação.



Fonte: ([SEN, 2021](#)).

### 2.2.1 Nível de chão de fábrica

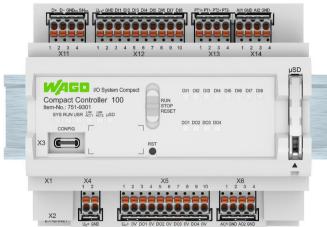
Segundo ([JÚNIOR; BRANDÃO, 2009](#)), no nível de chão de fábrica, os sistemas de automação são formados por dispositivos de campo que interagem diretamente no processo, essa interação é realizada através de leitura de dados, como sensores transmitindo valores de temperatura, e indo até elementos de controle, como atuadores.

Em um ambiente industrial, é extremamente importante que a comunicação seja determinística e em tempo real, garantindo o funcionamento correto dos sistemas de controle. Os sensores e atuadores presentes neste nível são interligados através de barramentos de campo, chamado fieldbuses. As principais vantagens de utilizar os fieldbuses são: controle distribuído, facilidade de expansão, interoperabilidade e intercambialidade, redução de custo e tempo na instalação de novos equipamentos ([SEN, 2021](#)).

### 2.2.2 Nível de controle

Segundo ([ROGGIA; FUENTES, 2016](#)) o Controlador Lógico Programável (CLP) é um equipamento eletrônico digital, com hardware e software compatível com as aplicações industriais, podendo ser programado através de uma linguagem de programação, de modo a executar tarefas lógicas, temporais, contagem, entre outras, controlando por módulos de entradas e saídas vários tipos de sistemas. A principal vantagem é a possibilidade de reprogramação e adaptação ao projeto, já que é possível adaptar as mudanças no hardware também no software.

Figura 5 – CLP - WAGO Compact Controller 100.

Fonte: ([WAGO, 2025](#)).

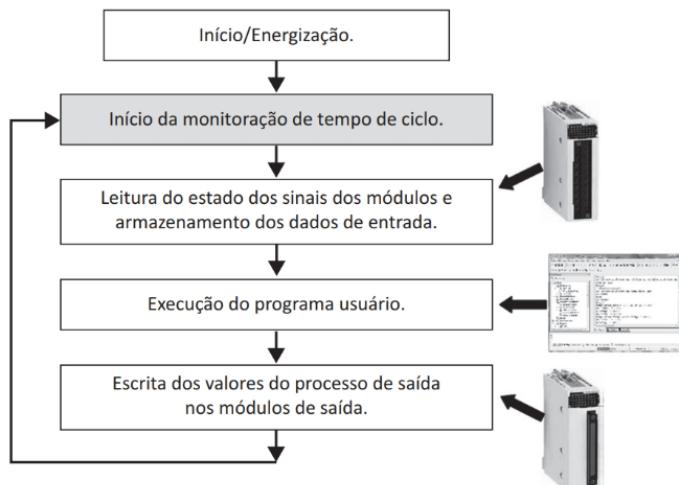
### 2.2.2.1 Funcionamento do Controlador Lógico Programável

O CLP substitui comandos elétricos convencionais simples por um sistema baseado em microcontrolador que processa sinais binários de entrada. Em ciclos discretos, o CLP lê as entradas, executa o programa de controle e atualiza as saídas. As leituras são armazenadas na memória de entrada e utilizadas durante a execução do programa, enquanto os resultados são enviados à memória de saída e, em seguida, às saídas físicas.

O tempo de ciclo depende da complexidade e da configuração do programa, devendo estar alinhado ao tempo de resposta do processo controlado. O funcionamento segue etapas como: energização, limpeza de memórias, testes de memória e programa, inicialização e execução da varredura cíclica.

O controlador executa continuamente um ciclo de varredura, no qual lê o estado das entradas, armazena esses valores na memória e os utiliza para processar o programa do usuário. Com base nessas informações, a CPU atualiza as saídas conforme as instruções programadas. Esse ciclo, que dura apenas microsssegundos, reinicia logo após a atualização das saídas, permitindo o controle em tempo quase real do processo automatizado. Esse processo é representado conforme a figura 6 a seguir.

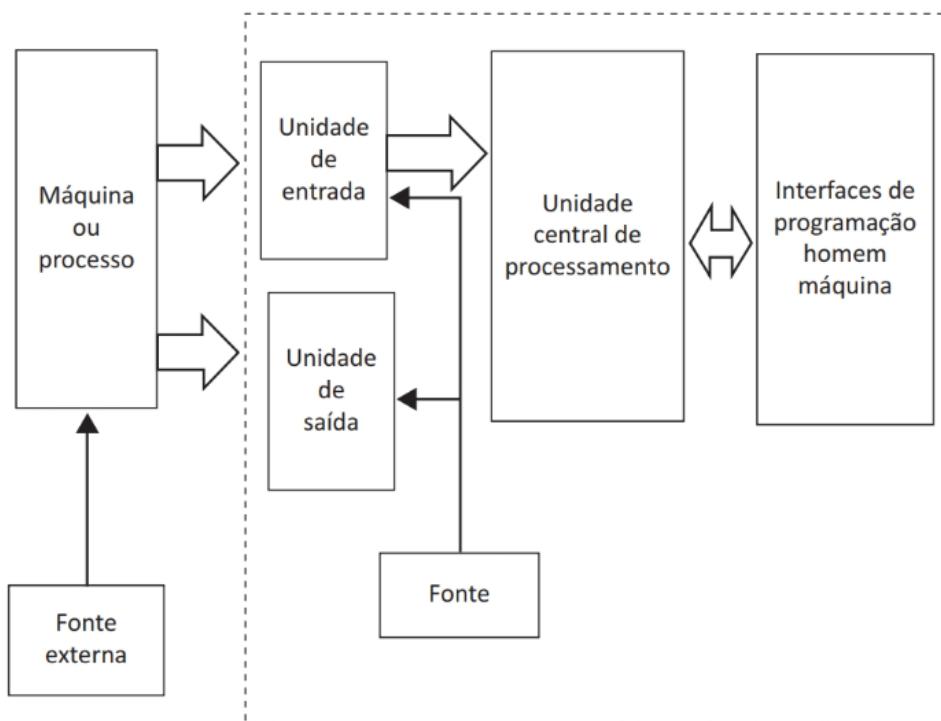
Figura 6 – Ciclo de varredura.

Fonte: ([SILVA, 2021](#)).

### 2.2.2.2 Arquitetura do Controlador Lógico Programável

A arquitetura básica de um CLP é composta por quatro unidades principais. A unidade de entrada recebe sinais elétricos provenientes da máquina ou processo, compatíveis com o circuito eletrônico do CLP. A unidade de saída envia os sinais processados para os dispositivos de campo, permitindo a atuação no processo. A unidade de processamento, considerada o “cérebro” do sistema, coordena todas as funções: recebe os dados das entradas, executa a lógica do programa e encaminha os resultados para as saídas. Por fim, a fonte de alimentação fornece a energia elétrica necessária para o funcionamento de todo o conjunto.

Figura 7 – Arquitetura básica do CLP.



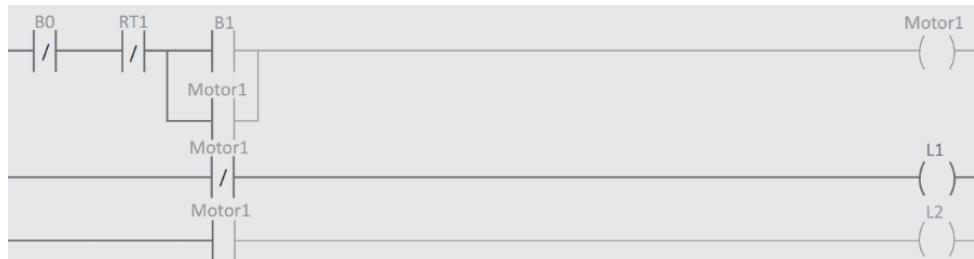
Fonte: (SILVA, 2021).

### 2.2.2.3 Linguagem de programação Ladder

Ladder é a linguagem de programação mais utilizada entre os CLPs, se assemelhando a um diagrama elétrico que possui bobinas e relés. Os diagramas são compostos de duas linhas verticais e de linhas horizontais, sob as quais são colocadas as instruções a serem executadas. Essa programação é feita utilizando uma combinação lógica dos contatos de entrada sendo eles contatos abertos ou contatos fechados. Essas linhas simulam um fluxo de corrente elétrica, fluindo da esquerda para a direita (ROGGIA; FUENTES, 2016).

Além dos contatos, existem diversas outras funções avançadas que podem ser programadas através da lógica Ladder. Sendo algumas delas (ROGGIA; FUENTES, 2016):

Figura 8 – Exemplo de lógica Ladder.

Fonte: ([SILVA, 2021](#)).

- Função set: finalidade de acionar uma bobina e mantê-la acionada mesmo após cessar o estímulo.
- Função reset: finalidade de desligar uma bobina acionada pela função set.
- Temporizador: finalidade de acionar ou desligar uma memória ou saída de acordo com um tempo programado.
- Contador: tem por finalidade ativar ou desativar uma memória ou saída a partir de uma determinada quantidade de eventos.

#### 2.2.2.4 Linguagem de Programação ST (STRUCTURED TEXT)

Ao contrário da linguagem ladder, o ST é uma linguagem textual de programação de CLPs. Esta é uma linguagem de alto nível semelhante a linguagem Pascal. A linguagem permite a invocação de blocos funcionais, funções e atribuições, além de possibilitar a execução condicional de instruções e tarefas repetitivas. Sua flexibilidade é facilmente compreendida, facilitando a interpretação do programa do processo industrial por programadores e desenvolvedores de software. Criada para controle industrial, a linguagem de programação ST utiliza “expressões” formadas por operadores e operandos que retornam um valor quando executados. ([SILVA, 2021](#))

A linguagem oferece suporte para o desenvolvimento de: declarações; configurações (configuration); recursos (resource); unidades de organização de programa (Program Organization Unit, POU); tarefas (task); tomada de decisões (IF...THEN...ELSE, FOR...DO etc.); cálculos diversos; implementação de algoritmos; e uso de literais.

A linguagem de texto estruturado é amplamente empregada na programação de funções e blocos funcionais, visando ocultar os detalhes de sua implementação interna dos usuários do programa. Os usuários entendem quais operações do objeto podem ser solicitadas, mas não têm conhecimento dos detalhes de como essas operações são executadas.

Figura 9 – Exemplo de linguagem ST.

```

(*Chave de partida direta*)
(*Comando para desligar*)
IF (B0=true or RT1=true) and (B1=false or C1=true)
THEN
C1:=false;
END_IF;
(*Comando para ligar*)
IF (B0=false and RT1=false) and (B1=true or C1=true)
THEN
C1:=true;
END_IF;
(*Sinalização L1*)
IF (C1=false) THEN L1:=true;
END_IF;
IF (C1=true) THEN L1:=false;
END_IF;
(*Sinalização L2*)
IF (C1=false)
THEN L2:=false;
END_IF;
IF (C1=true) THEN L2:=true;
END_IF;

```

Fonte: ([SILVA, 2021](#)).

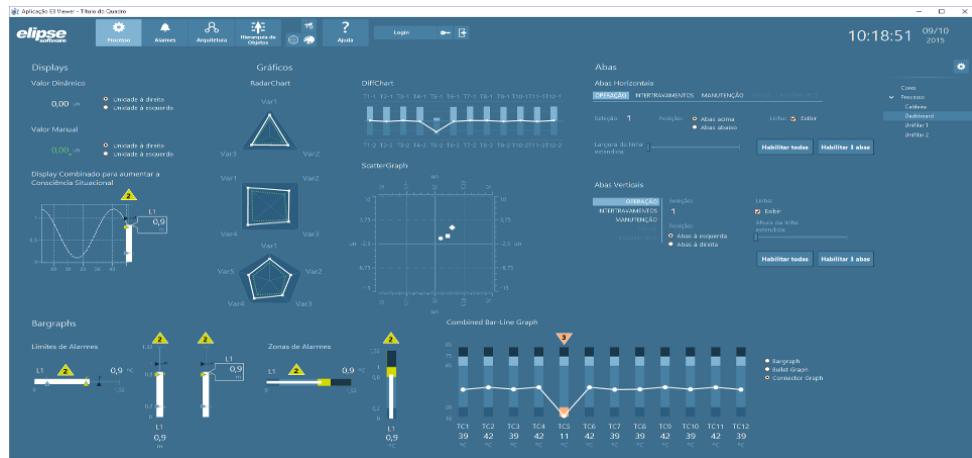
### 2.2.3 Nível de supervisão

Segundo ([SEN, 2021](#)) no segundo nível mais alto da pirâmide de automação estão localizados os sistemas de supervisão, controle e manutenção da planta de responsabilidade dos operadores e engenheiros. Esse nível é formado principalmente pelos sistemas SCADA (Sistemas de Supervisão e Aquisição de Dados). O SCADA é um sistema que usa um software para monitorar, supervisionar e controlar as variáveis e os equipamentos presentes em um processo industrial.

Este nível, também conhecido como HMI, é onde são programadas as telas de supervisão para controle de variáveis do processo, alarmes para alertar os operadores e/ou engenheiros sobre algum comportamento anormal do sistema, softwares para configuração de equipamentos para alterar algum parâmetro de seu funcionamento, programas de manutenções preditivas, entre outros.

Um dos principais sistemas SCADA é o ELIPSE E3, que é um software com componentes para: configuração de sistemas de supervisão personalizados, monitoramento de dados em tempo real através de diagramas ou gráficos, gerenciamento das variáveis do sistema para armazenamento em banco de dados com redundância e sincronia, configuração de alarmes e eventos, integração com sistemas corporativos e de gestão, entre outros ([ELIPSE, 2025](#)).

Figura 10 – Tela de supervisão do ELIPSE E3.



Fonte: (ELIPSE, 2025).

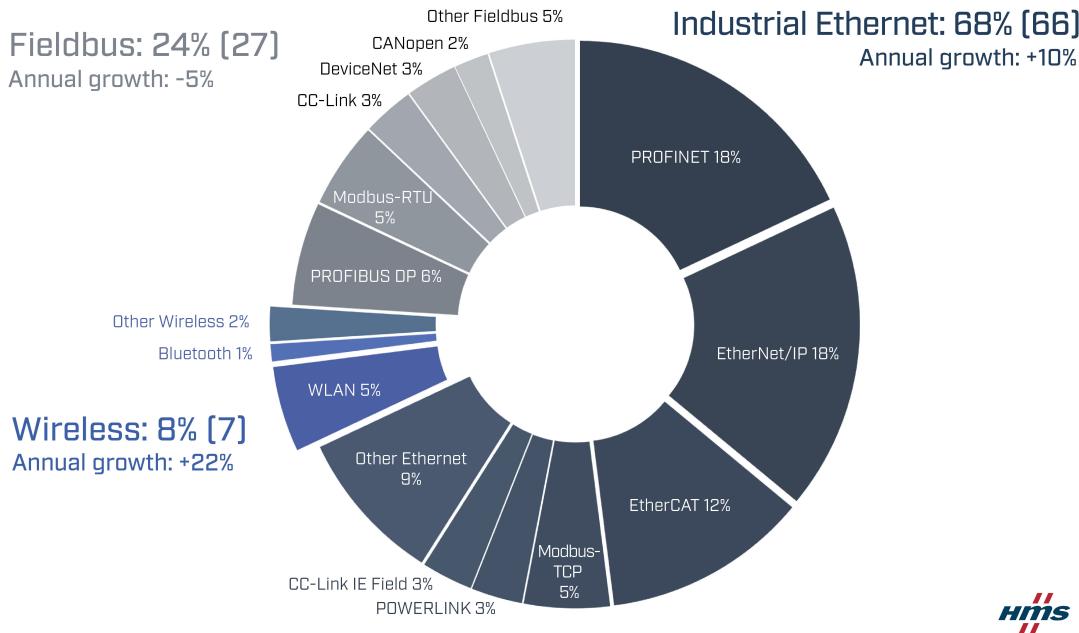
## 2.3 Redes Industriais

As redes dentro de um processo industrial também tendem a diferir de acordo com o seu nível de automação e posicionamento na pirâmide de automação, que abrange desde o nível de chão de fábrica ao nível corporativo. Como dito anteriormente, no nível de chão de fábrica é formado por transmissores que comunicam com os controladores através de sub redes, podendo essas estarem a distância de quilômetros. Essas redes são denominadas fieldbuses (SEN, 2021).

Os diferentes níveis possuem suas peculiaridades, já que lidam com diferentes tipos de informações, por exemplo, no nível corporativo temos um grande volume de dados, mas a velocidade não é crítica, e a disponibilidade dos dados não necessita ser constante. Diferentemente, de como é a nível de chão de fábrica, onde o volume de dados é pequeno, mas precisa ser rápido e determinístico dependendo do tipo de aplicação implementada (SEN, 2021).

Os protocolos de redes industriais de campo, vem evoluindo muito nas últimas décadas, tendo seu início da década de 80, com os protocolos HART e Modbus. Hoje existem dezenas de diferentes protocolos, alguns exemplos que são utilizados em redes de manufatura, onde normalmente os equipamentos são discretos (solenoides, sensores de posição, entre outros), com a velocidade do ciclo de varredura na ordem de milissegundos, são: Asi, Devicenet, CAN. Quando se trata de controle de processos, que predomina variáveis analógicas, e com a velocidade do ciclo de varredura na ordem de dezenas de milissegundos, os principais exemplos de protocolos são: Profibus DP, Profibus PA, Foundation Fieldbus. Para a comunicação entre o nível de campo para os níveis de controle e de supervisão (IHM), normalmente são utilizados os protocolos Modbus TCP, Ethernet IP, Profinet, HSE (SEN, 2021). A Figura 11 mostra os resultados de uma pesquisa sobre quais os protocolos mais utilizados.

Figura 11 – Gráfico com o resultado da pesquisa sobre os protocolos mais utilizados em 2023.



Fonte: ([NETWORKS, 2023](#)).

### 2.3.1 Protocolo Modbus

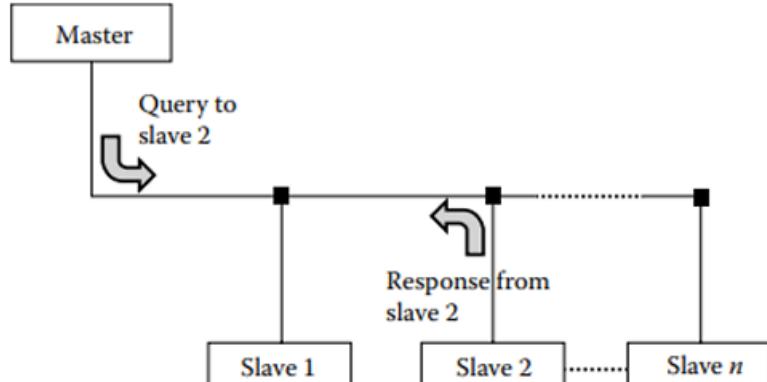
O protocolo Modbus foi criado pela Modicon Industrial Automation Systems, atual Schneider, com a finalidade de estabelecer uma comunicação de um dispositivo mestre com um ou mais dispositivos escravos.

O protocolo Modbus se baseia no modelo de comunicação cliente-servidor (mestre-escravo), onde um dos equipamentos é o cliente (mestre), que deve ser único, e é responsável por estabelecer e iniciar a comunicação (query) com os outros equipamentos. Os outros equipamentos, os servidores (escravos), respondem a solicitação do cliente, realizando alguma tarefa ou transmitindo dados.

O dispositivo cliente pode enviar uma solicitação endereçada unicamente (unicast) a um dispositivo ou para todos os servidores da rede (broadcast). A resposta ao cliente somente ocorre quando a solicitação é da forma unicast, no caso do broadcast, o cliente não recebe uma resposta ([SEN, 2021](#)).

A Figura 12 demonstra o tipo de solicitação de endereçamento único:

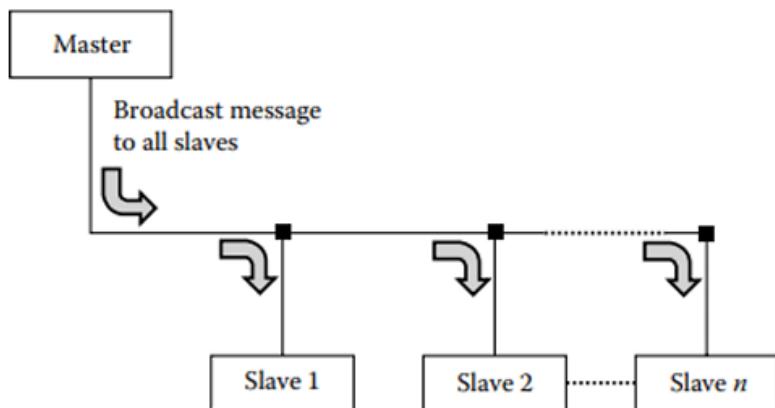
Figura 12 – Diagrama exemplificando uma solicitação do tipo unicast.



Fonte: ([SEN, 2021](#)).

A Figura 13 demonstra o tipo de solicitação de endereçamento múltiplo:

Figura 13 – Diagrama exemplificando uma solicitação do tipo broadcast.



Fonte: ([SEN, 2021](#)).

O protocolo possui uma estrutura de comunicação composta por bytes, assimilada pela maioria dos equipamentos. Devido a versatilidade pode ser utilizado em diversos padrões de meio físico como ([FREITAS, 2014](#)):

- Padrão RS-232: utilizado apenas em comunicação ponto a ponto, só admite dois dispositivos, sendo um mestre e um escravo. A velocidade máxima está em torno de 115kbps e a distância máxima entre os dispositivos é aproximadamente de 30m.
- Padrão RS-485: muito utilizado na indústria e um dos padrões mais utilizados pelo protocolo Modbus. O número máximo de dispositivos é 32, a velocidade máxima é em torno de 12 Mbps, podendo alcançar até os 50 Mbps, reforçando que quanto maior o comprimento da rede, menor a velocidade. A distância máxima está em torno de 1200m.
- Padrão TCP/IP (Ethernet): A velocidade desse padrão pode variar da casa dos 12Mbps até 10Gbps e a distância máxima variar de 100m a 200m, dependendo do

tipo de cabo e conexão utilizada. Também é possível utilizar redes de fibra ótica, que aumentam a velocidade assim como um maior comprimento da rede.

### 2.3.1.1 Modbus RTU

Para a transmissão de mensagens, o quadro de mensagens (frame) é construído antes mesmo da sua transmissão. O frame é constituído por um ou mais caractere(s) de início e fim, endereço do dispositivo em caso de comunicação unicast ou o endereço dos dispositivos em caso de comunicação broadcast, o código da função e o código de checagem de dados e erro.

No modo de transmissão RTU (Remote Terminal Unit), para cada palavra de dados da mensagem, é enviado somente um caractere no formato Hexadecimal. O frame inicia com um intervalo de silêncio de pelo menos 3,5 vezes o tamanho da palavra de dados. Esse intervalo é denominado silent. Após o tempo de silêncio, indicando o início da transmissão, o campo de endereço é monitorado pelos dispositivos que a recebem, verificando a quem a mensagem foi destinada ([SEN, 2021](#)).

A transmissão de um frame deve ocorrer em fluxo contínuo; caso haja uma interrupção superior a 1,5 vezes o tempo correspondente ao tamanho da palavra de dados, os dispositivos receptores interpretarão como erro e descartarão qualquer informação recebida. Da mesma forma, se uma nova mensagem for recebida em um intervalo inferior a 3,5 vezes o tempo da palavra de dados após o término da anterior, o dispositivo poderá associá-la incorretamente à mensagem anterior, resultando em um erro de verificação de redundância cíclica (CRC).

Figura 14 – Estrutura de um Frame Modbus no modo de transmissão RTU.

<b>Start 3.5 char times</b>	<b>Address 8 bits</b>	<b>Function code 8 bits</b>	<b>Data <math>N \times 8</math> bits</b>	<b>CRC 16 bits</b>	<b>End 3.5 char times</b>
-------------------------------------	---------------------------	-------------------------------------	--	------------------------	-----------------------------------

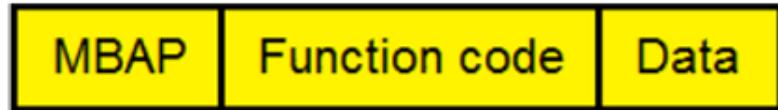
Fonte: ([SEN, 2021](#)).

### 2.3.1.2 Modbus TCP/IP

O padrão Modbus TCP/IP foi introduzido em 1999. Ele possui diversas vantagens como: simplicidade, uso do padrão Ethernet. Taxas de transferência acima de 1 kB/s podem ser facilmente atingidas ([SEN, 2021](#)).

Segundo ([FREITAS, 2014](#)) o Modbus TCP/IP é uma implementação TCP/IP para o protocolo Modbus. Esse protocolo adiciona ao frame Modbus um cabeçalho específico denominado MBAP (MODBUS Application Protocol). Assim, o modelo de frame do protocolo Modbus TCP/IP pode ser representado da seguinte forma:

Figura 15 – Estrutura de um Frame Modbus no modo de transmissão TCP/IP.



Fonte: ([FREITAS, 2014](#)).

O cabeçalho MBAP possui o tamanho de 7 bytes e é constituído por:

- Transaction Identifier: identificação da resposta para a transação (2 bytes).
- Protocol Identifier: 0(zero) se refere a Modbus (2 bytes).
- Length : contagem dos próximos bytes (2 bytes).
- Unit Identifier: identifica o escravo remoto em uma rede Modbus RTU (1 byte).

O padrão Modbus TCP/IP não acrescenta um quadro referente a checagem de erros ao frame, mas o frame Ethernet utiliza o CRC de 32 bits, assim não é necessário outro tipo de checagem de erro. Assim, o cliente Modbus TCP deve iniciar uma conexão TCP com o servidor para o envio das requisições de comunicação, utilizando, por padrão, a porta TCP 502.

### 2.3.1.3 Tipos de Dados Lógicos e Funções do Modbus

No frame de transmissão é definido o tipo de dados e o operando, em que cada tipo pode contar com 9999 operandos. Com isso é possível ter os seguintes tipos de endereço:

1. Saída discreta (coils) 00001 a 09999
2. Entrada discreta (Inputs) 10001 a 19999
3. Entradas Analógicas (Input Registers) 30001 a 39999
4. Saídas Analógicas (Holding Registers) 40001 a 49999

Além disso, é definida a função Modbus a ser executada, sendo as funções disponíveis no protocolo Modbus, de leitura e escrita em um escravo, ou mensagem de broadcasting, que pode ser leitura e escrita simultânea em todos os dispositivos escravos da rede. Com isso, dependendo do tipo de dado é determinado o número do comando.

Figura 16 – Tipos de funções Modbus.

Código da função	Descrição
1	Leitura de bloco de bits do tipo coil(saída discreta).
2	Leitura de bloco de bits do tipo entradas discretas.
3	Leitura de bloco de registradores do tipo holding.
4	Leitura de bloco de registradores do tipo input.
5	Escrita em um único bit do tipo coil(saída discreta).
6	Escrita em um único registrador do tipo holding.
7	Ler o conteúdo de 8 estados de exceção.
8	Prover uma série de testes para verificação da comunicação e erros internos.
11	Obter o contador de eventos.
12	Obter um relatório de eventos.
15	Escrita em bloco de bits do tipo coil(saída discreta).
16	Escrita em bloco de registradores do tipo holding.
17	Ler algumas informações do dispositivo.
20	Ler informações de um arquivo.
21	Escrever informações em um arquivo.
22	Modificar o conteúdo de registradores de espera através de operações lógicas.
23	Combina ler e escrever em registradores numa única transação.
24	Ler o conteúdo da fila FIFO de registradores.
43	Identificação do modelo do dispositivo.

Fonte: ([FREITAS, 2014](#)).

## 2.4 Sistemas de Segunda Ordem

Segundo ([NISE, 2023](#)), enquanto sistemas de primeira ordem possuem comportamento dinâmico mais previsível e limitado, os sistemas de segunda ordem se destacam por sua complexidade e variedade de respostas. Em um sistema de primeira ordem, a variação de parâmetros geralmente influencia apenas a rapidez da resposta. Já em sistemas de segunda ordem, essas variações podem transformar profundamente o perfil da resposta, resultando em características como oscilações, sobrepasso e diferentes níveis de amortecimento, exigindo uma análise mais cuidadosa de seu comportamento dinâmico.

A função de transferência de segunda ordem geral apresenta a seguinte forma:

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (2.1)$$

Onde:

- $\omega_n$  é A frequência natural de um sistema de segunda ordem corresponde à frequência com que ele oscilaria livremente, ou seja, na ausência de qualquer tipo de amortecimento.

- $\zeta$  é o fator de amortecimento do sistema, definido como uma grandeza baseada na razão entre a frequência de decaimento exponencial da envoltória e a frequência natural do sistema. Essa razão se mantém constante, independentemente da escala de tempo considerada. Da mesma forma, seu inverso, que é proporcional à relação entre o período natural e a constante de tempo do decaimento exponencial, também permanece inalterado, independentemente da base temporal adotada.

O fato de amortecimento permite classificar o tipo de resposta do sistema:

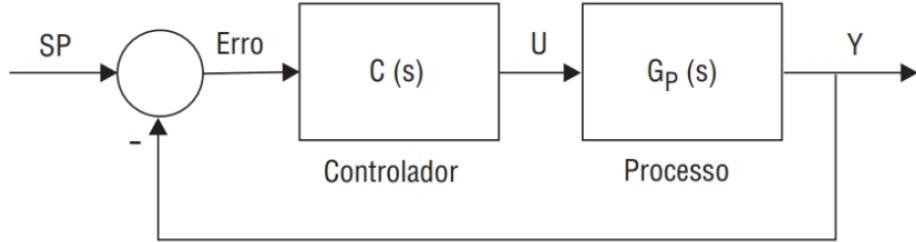
- $\zeta = 0$ : sistema não amortecido (oscilação permanente);
- $0 < \zeta < 1$ : sistema subamortecido (oscilação com amortecimento);
- $\zeta = 1$ : sistema criticamente amortecido (sem oscilação, resposta mais rápida sem ultrapassagem);
- $\zeta > 1$ : sistema superamortecido (resposta lenta e sem ultrapassagem);

## 2.5 Controlador PID

O controlador Proporcional-Integral-Derivativo (PID), mesmo com os avanços tecnológicos obtidos nas últimas décadas, continua oferecendo uma solução simples e eficiente para a grande maioria de aplicações de controle industrial. Devido a ser uma proposta existente há décadas, hoje possuímos diversos estudos e soluções referente ao controlador PID ([CAMPOS; TEIXEIRA, 2010](#)).

Quando falamos de sistemas de controle, uma malha de controle é formada pelos seguintes itens: processo controlado, elemento de medição, controlador e atuador. Esse controle pode ser realizado com a malha aberta ou fechada, o que diferencia é que, no caso da malha fechada, o sinal do elemento de medição ou a variável do processo é utilizada para corrigir o erro na saída da malha. O controle em malha fechada atua considerando o valor de saída do sistema, realimentando-o na entrada. Além disso, o sinal de saída é comparado a um valor de referência (setpoint), e a diferença entre esses dois sinais (erro) determina que tipo de sinal o controlador efetuará ([CAMPOS; TEIXEIRA, 2010](#)).

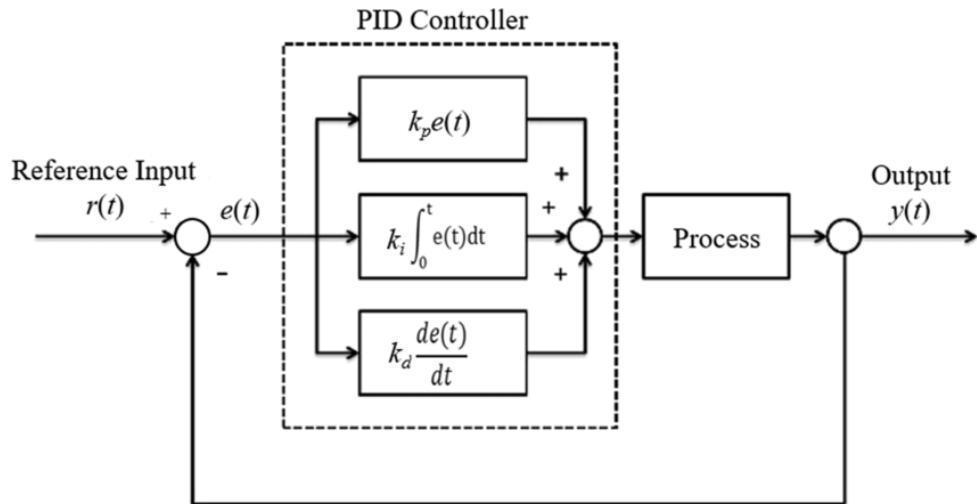
Figura 17 – Sistema de controle em malha fechada com realimentação negativa.



Fonte: (CAMPOS; TEIXEIRA, 2010).

O controlador realiza o cálculo usando o sinal de erro, de forma a determinar o sinal a ser aplicado na entrada do sistema. O controlador possui três parâmetros de configuração denominados: ação proporcional, ação integrativa e ação derivativa.

Figura 18 – Controlador PID.



Fonte: (CAMPOS; TEIXEIRA, 2010).

Normalmente, o algoritmo do controlador PID é encontrado em três diferentes estruturas na literatura: a forma padrão ou ISA, a forma paralela e a forma série. A seguir, podemos visualizar as fórmulas de cada equação, referente a cada tipo de estrutura no domínio de Laplace (JÚNIOR; BRANDÃO, 2009):

$$G_c = \frac{Y_{(s)}}{U_{(s)}} = K_c \left( 1 + \frac{1}{T_i s} + T_d s \right) \quad (\text{ISA}) \quad (2.2)$$

$$G_c = \frac{Y_{(s)}}{U_{(s)}} = K_c + \frac{1}{T_i s} + T_d s \quad (\text{PARALELO}) \quad (2.3)$$

$$G_c = \frac{Y_{(s)}}{U_{(s)}} = K_c + \left( \frac{T_i s + 1}{T_i s} \right) + (T_d s + 1) \quad (\text{SÉRIE}) \quad (2.4)$$

Sendo  $K_c$  o ganho proporcional,  $T_i$  a constante de tempo integral,  $T_d$  a constante de tempo derivativa.

### 2.5.1 Ação Proporcional

A ação proporcional, depende somente do erro calculado, sendo esse a diferença do setpoint e a variável do processo. O ganho do controlador proporcional ( $K_p$ ) é a taxa de resposta da saída do sistema para o valor do erro em função do tempo ( $e(t)$ ). Assim, quando aumentamos o ganho proporcional, obtemos uma maior velocidade de resposta do sistema, assim como ao reduzi-lo, deixamos a resposta do sistema mais lenta.

$$y(t) = K_p \cdot e(t) \quad (2.5)$$

### 2.5.2 Ação Integral

A ação integral produz um sinal de saída proporcional ao valor e duração do erro, melhor dizendo, ao erro acumulado ( $\int e(t) dt$ ). Assim, é possível corrigir o erro de offset que é gerado pela ação proporcional, de forma a reduzir o erro em regime permanente para zero. Sendo a constante de tempo de integral ( $\tau_i$ ).

$$y(t) = \frac{1}{\tau_i} \cdot \int e(t) dt \quad (2.6)$$

### 2.5.3 Ação Derivativa

A ação derivativa realiza uma correção antecipada do erro, reduzindo o tempo de resposta e melhorando a estabilidade do sistema. A constante de tempo derivativa ( $T_d$ ), tem a função de determinar a velocidade de ação do sistema caso ocorra uma variação de erro no tempo, aumentando a velocidade de resposta global do sistema, através da taxa de variação do erro em função do tempo  $\left(\frac{de(t)}{dt}\right)$ .

$$y(t) = T_d \cdot \frac{de(t)}{dt} \quad (2.7)$$

Como existem diferenças nos parâmetros dos algoritmos ISA, paralelo e série, assim como diferenças nos outros algoritmos existentes, é necessário saber com qual algoritmo está trabalhando a sua aplicação. Os diversos métodos de sintonia presentes atualmente serão demonstrados nas próximas seções.

## 2.6 Identificação de Sistemas

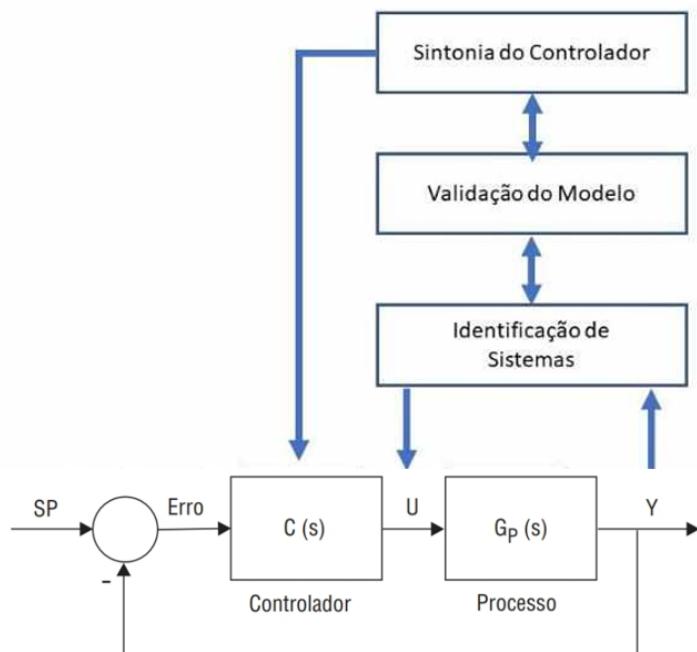
Antes de apresentar os métodos de sintonia dos controladores, é necessário, primeiramente, entender as características do sistema a ser controlado. Para obtermos um modelo que represente o funcionamento de um processo ou sistema real é possível através

da utilização de modelagem matemática sendo essa próxima as características do sistema real ([CAMPOS; TEIXEIRA, 2010](#)).

Podemos classificar essa modelagem matemática em caixa branca ou caixa preta. A modelagem caixa branca, ou teórica, se baseia no modelo teórico do sistema, sendo necessário conhecer todas as características dinâmicas, físicas e químicas do processo. No caso da modelagem em caixa preta, também conhecida como modelagem experimental, o modelo é obtido através do estudo dos dados obtidos diretamente do processo a partir de uma excitação.

Segundo ([JÚNIOR; BRANDÃO, 2009](#)) para obtermos a sintonia de um controlador é preciso identificar o sistema, validar o modelo obtido, propor uma sintonia baseada no modelo e então efetivar a sintonia do controlador. A identificação de sistemas é uma maneira de obtermos um modelo experimental, através da coleta de dados dos sinais de entrada  $u$  e os sinais de saída  $y$  do sistema a ser identificado, como representado na Figura 19. Após essa etapa, é estimado a função de transferência continua ou discreta do sistema, que representa seu comportamento.

Figura 19 – Processos para identificação de um sistema.



Fonte: Adaptado de ([SANTOS, 2019](#)).

Quando se trata de um modelo dinâmico, podemos analisar o sistema através das relações entre as entradas e saídas do sistema durante uma resposta transiente. Isto se aplica a sistemas lineares invariantes no tempo. A partir da resposta transiente, os processos podem obter distintas classificações, sendo elas: processos autorregulatórios e processos integradores. Para cada classificação é necessário um processo distinto de sintonia ([SANTOS, 2019](#)).

O processo autorregulatório se identifica por sua saída  $y$  se estabilizar em um outro patamar após uma variação na entrada  $u$ . Sistemas de temperatura, de pressão e de vazão são exemplos de processos autorregulatórios e são geralmente representados por um modelo de primeira ordem com tempo morto (JÚNIOR; BRANDÃO, 2009):

$$G_p(s) = \frac{K_p}{\tau s + 1} \cdot e^{-\theta s} \quad (2.8)$$

Onde:

- Ganho estático ( $K_p$ ): representa a sensibilidade do processo a uma mudança na entrada.
- Constante de tempo ( $\tau$ ): mede a velocidade em que a variável de processo (PV) atingirá seu valor final.
- Tempo morto ( $\theta$ ): tempo necessário para a PV começar a sofrer alterações, por causa de uma mudança na variável manipulada (MV).

O processo integrador pode diminuir ou aumentar o valor da saída  $y$  indefinidamente, depois de ocorrer uma variação transiente. Sistemas de controle de posição e controle de nível são alguns exemplos de um processo integrador. A equação abaixo exemplifica o comportamento de um sistema integrador (JÚNIOR; BRANDÃO, 2009):

$$G_p(s) = \frac{K_p}{s(\tau s + 1)} \cdot e^{-\theta s} \quad (2.9)$$

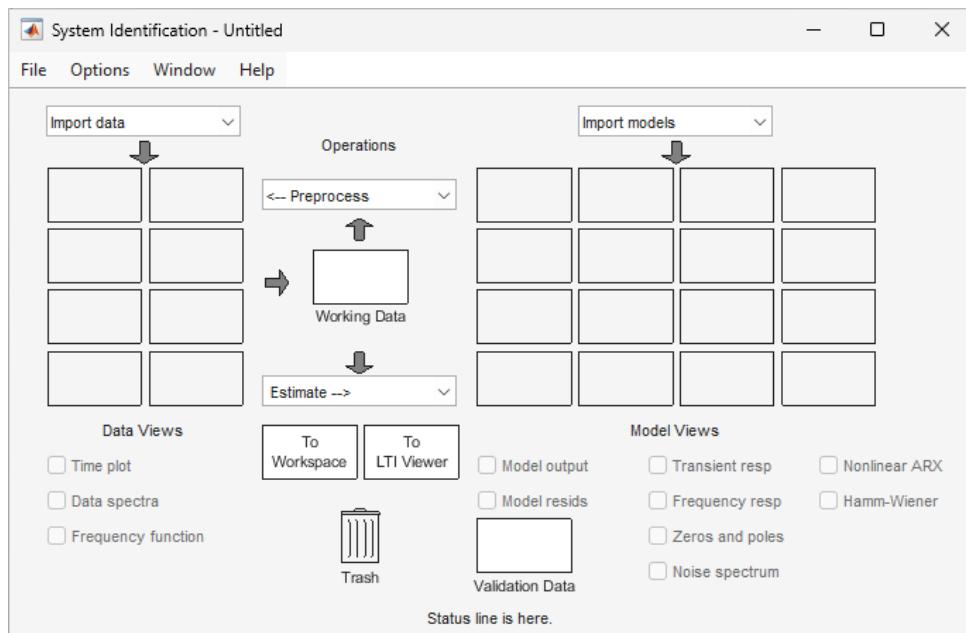
Podemos dividir a identificação de sistemas em algumas etapas (JÚNIOR; BRANDÃO, 2009):

- Testes dinâmicos e coletas de dados: para se obter os valores de entrada e saída do sistema, é necessário realizar testes para gerar perturbações na entrada do sistema, assim revelando o comportamento dinâmico da saída do sistema.
- Escolha do modelo matemático que representará o sistema: determinar como será obtida a função transferência do sistema. Geralmente, a maioria dos processos industriais podem ser aproximados por um sistema de primeira ou segunda ordem com tempo morto.
- Estimação do modelo: são estimados os valores dos parâmetros do modelo matemático escolhido que representará o sistema.

- Validação da estimativa: nesta etapa é verificado se realmente o sistema estimado representa fielmente o sistema original. Caso não, é necessário reestimar os parâmetros ou até mesmo alterar o modelo matemático escolhido.

A identificação de sistema pode ser feita, normalmente, por algoritmos de programação e por softwares. O software Matlab possui uma ferramenta de identificação de sistemas eficiente, onde é necessário somente inserir os dados de entrada  $u$  e saída  $y$  do sistema e o software fornece diversas opções de estimativa.

Figura 20 – System Identification Tool – Matlab.



Fonte: ([MATLAB, 2025](#)).

## 2.7 Sintonia de controladores PID

Segundo ([SANTOS, 2019](#)) o controlador PID é o algoritmo de controle mais utilizado na indústria, por já existir a várias décadas, assim muitos estudos foram feitos a partir dele, criando vários métodos de sintonia. Serão apresentados nessa seção três métodos de sintonia: Ziegler e Nichols (Z&N), IMC e os métodos baseados em critério de desempenho da integral do erro. O método de Ziegler e Nichols é o pioneiro, já os outros dois são conhecidos pelo baixo overshoot e bom tempo de acomodação.

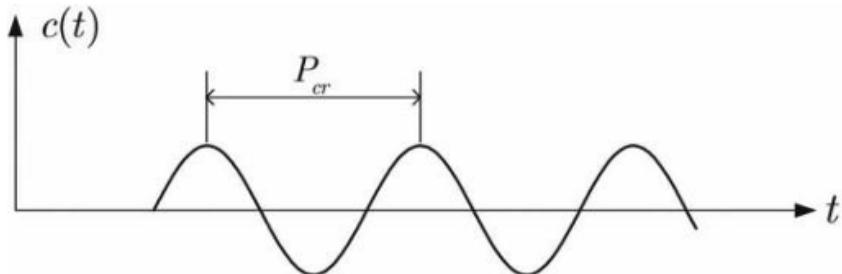
### 2.7.1 Método de Ziegler e Nichols

Ziegler e Nichols (1942) elaboraram dois métodos de sintonia de controladores PID, um deles baseado em um sistema de malha fechada e o segundo realizado em um sistema de malha aberta ([CAMPOS; TEIXEIRA, 2010](#)).

Os passos para o primeiro método, em malha fechada, são:

1. Ajuste  $T_i$  e  $T_d$  do controlador PID em zero;
2. Aumente vagarosamente o ganho  $K_c$  até que ocorra uma oscilação periódica na saída do sistema;
3. Sendo  $K_u$  o ganho final e  $P_u$  o período de oscilação.

Figura 21 – Sistema com oscilação periódica.



Fonte: ([CAMPOS; TEIXEIRA, 2010](#)).

4. Realize o ajuste dos parâmetros do controlador de acordo com a Tabela 1.

Tabela 1 – Parâmetros do método em malha fechada de Ziegler e Nichols.

Tipo de controlador	$K_c$	$T_d$	$1/T_i$
P	$0,5K_u$	0	0
PD	$0,6K_u$	$P_u/8$	0
PI	$0,45K_u$	0	$1,2/P_u$
PID	$0,6K_u$	$P_u/8$	$2/P_u$

O outro método é baseado em um sistema de malha aberta. Com isso, após deixar o sistema em malha aberta, deve-se excitar a entrada do processo  $u$  e com a saída do sistema  $y$  obtém-se um modelo de primeira ordem com tempo morto.

Sendo:

- $K$  : o ganho do processo;
- $\tau$  : a constante de tempo do processo;
- $\theta$  : o tempo morto.

Temos abaixo a Tabela 2, com os parâmetros para o método em malha aberta:

O problema presente nestes dois métodos é um overshoot alto ao utilizar esse tipo de sintonia em um controlador, o que nem sempre é desejado.

Tabela 2 – Parâmetros do método em malha aberta de Ziegler e Nichols.

Tipo de controlador	$K_p$	$T_i$	$T_d$
P	$\tau/(K \theta)$	—	—
PI	$0,9 \frac{\tau}{(K \theta)}$	$3,33 \theta$	—
PID	$1,2 \frac{\tau}{(K \theta)}$	$2 \theta$	$0,5 \theta$

### 2.7.2 Método de Integral do Erro

O método de integral do erro foi proposto inicialmente por Lopez e Rovira (1969) e ele possui a vantagem de considerar toda a curva de resposta, ao contrário de somente dois pontos como proposto por Ziegler e Nichols (Z&N) (JÚNIOR; BRANDÃO, 2009).

Considerando a entrada do sistema ( $SP$ ) e a saída do sistema ( $y(t)$ ), o método é baseado na integral do erro e em critérios de desempenho. Foi estudado o algoritmo padrão (ISA) utilizando um sistema de primeira ordem e tempo morto. Existem quatro critérios: ISE, ITSE, IAE e ITAE. O quadro abaixo descreve cada um deles.

Tabela 3 – Método de integral do erro.

Critério	Fórmula	Descrição
ISE – Integral do erro quadrático	$\int_0^{\infty} e(t)^2 dt$	Proporciona grande peso para erros grandes e pequenos pesos para erros pequenos. Possui pouca variação dos parâmetros para o sistema escolhido. Um sistema projetado utilizando esse critério possui: rápida diminuição em um erro inicial grande, resposta pode ser rápida e oscilatória e o sistema possui baixa estabilidade relativa.
ITSE – Integral do erro quadrático ponderado pelo tempo	$\int_0^{\infty} t e(t)^2 dt$	Neste critério, um grande erro inicial possui o peso baixo, enquanto os erros em regime permanente possuem grande peso. Possui uma seletividade melhor que o critério ISE.
IAE – Integral do erro absoluto	$\int_0^{\infty}  e(t)  dt$	Sistemas que são altamente subamortecidos ou sobreamortecidos apresentam amortecimento aceitável e uma resposta transitória satisfatória.
ITAE – Integral do erro absoluto ponderado pelo tempo	$\int_0^{\infty} t  e(t)  dt$	Erro inicial grande em uma resposta a degrau unitário possui peso pequeno, e erros que ocorrem posteriormente são muito penalizados. Possui pequenas oscilações no sobre-sinal.

No caso de variação de carga, o principal objetivo é rejeitar as perturbações (problema regulatório), foi estudado para a razão de incontrolabilidade ( $\theta/\tau$ ) entre 0 e 1, o ganho proporcional do controlador ( $K_p$ ), o tempo de integração ( $T_i$ ) e o tempo derivativo ( $T_d$ ), obtendo assim as seguintes equações:

$$K_p = \frac{1}{K} \cdot \left( A \left( \frac{\theta}{\tau} \right)^B \right) \quad (2.10)$$

$$T_i = \frac{\tau}{\left( C \left( \frac{\theta}{\tau} \right)^D \right)} \quad (2.11)$$

$$T_d = \tau \cdot \left( E \left( \frac{\theta}{\tau} \right)^F \right) \quad (2.12)$$

Abaixo, a tabela com os valores das constantes A, B, C, D, E, F:

Tabela 4 – Valores das constantes da equação.

Controlador	Critério	A	B	C	D	E	F
PI	IAE	0,984	-0,986	0,608	-0,707	—	—
PI	ITAE	0,859	-0,977	0,674	-0,68	—	—
PID	IAE	1,435	-0,921	0,878	-0,749	0,482	1,137
PID	ITAE	1,357	-0,947	0,842	-0,738	0,381	0,995

No caso de variação de  $SP$ , foi obtido as seguintes equações:

$$K_p = \frac{1}{K} \cdot \left( A^* \left( \frac{\theta}{\tau} \right)^{B^*} \right) \quad (2.13)$$

$$T_i = \frac{\tau}{\left( C^* \left( \frac{\theta}{\tau} \right)^{D^*} \right)} \quad (2.14)$$

$$T_d = \tau \cdot \left( E^* \left( \frac{\theta}{\tau} \right)^{F^*} \right) \quad (2.15)$$

Onde as constantes  $A^*$ ,  $B^*$ ,  $C^*$ ,  $D^*$ ,  $E^*$ ,  $F^*$  são:

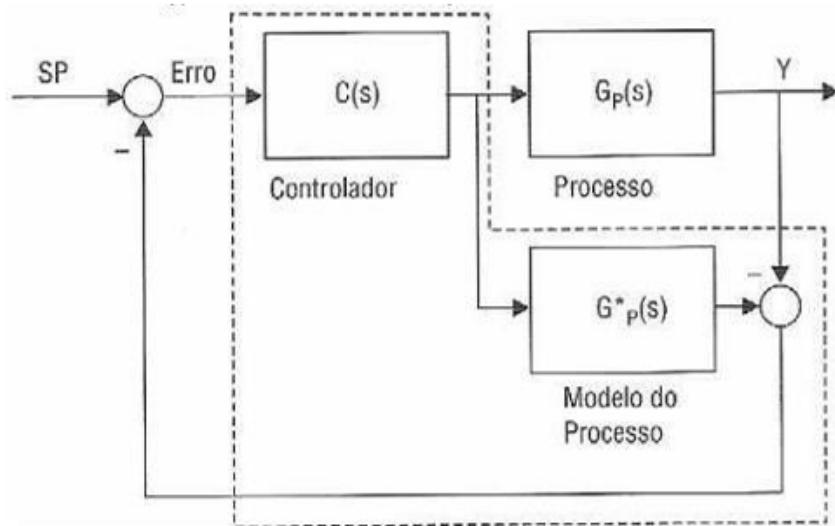
Tabela 5 – Valores das constantes da equação para sintonia do PID.

Controlador	Critério	A	B	C	D	E	F
PI	IAE	0,758	-0,861	1,02	-0,323	—	—
PI	ITAE	0,586	-0,916	1,03	-0,165	—	—
PID	IAE	1,086	-0,869	0,740	-0,130	0,348	0,914
PID	ITAE	0,965	-0,850	0,796	-0,147	0,308	0,929

### 2.7.3 Método IMC (Internal Model Control)

O método IMC tem como objetivo, a partir do modelo do processo e de uma especificação de desempenho, sintonizar o controlador desejado (CAMPOS; TEIXEIRA, 2010).

Figura 22 – Modelo de um processo segundo o método IMC.



Fonte: (CAMPOS; TEIXEIRA, 2010).

Assim, esse método requer um modelo de processo, podendo ser obtido através de identificação experimental, e após isso obter a função de transferência de malha fechada do sistema. Considerando  $C(s)$  o controlador,  $G_p(s)$  o processo físico, e  $G_p^*(s)$  o modelo do processo, a função de transferência em malha fechada se dá por:

$$\frac{Y(s)}{SP(s)} = \frac{G_p(s) \cdot C(s)}{(1 + G_p(s) \cdot C(s))} \quad (2.16)$$

Onde o controlador é:

$$C(s) = K_p \cdot \left( 1 + \frac{1}{T_i \cdot s} + T_d \cdot s \right) \quad (2.17)$$

Temos o objetivo de que a resposta do sistema a um degrau no  $SP$  tenha uma dinâmica conhecida e que é fornecida como critério de ajuste. Abaixo, um exemplo de como podemos sintonizar o controlador pelo método IMC, considerando uma FT (função transferência) de 1ª ordem com critério de desempenho ( $\lambda$ ):

$$\frac{Y(s)}{SP(s)} = \frac{1}{(\lambda s + 1)^n} \quad (2.18)$$

Lembrando que o Lambda deve ser escolhido respeitando as restrições dinâmicas do processo. Não se deve escolher um ( $\lambda$ ) menor que o tempo morto ( $\theta$ ) pois a sintonia pode ficar extremamente agressiva. Ao igualarmos as equações (16) e (18):

$$\frac{Y(s)}{SP(s)} = \frac{1}{(\lambda s + 1)} = \frac{G_p(s) \cdot C(s)}{(1 + G_p(s) \cdot C(s))} \quad (2.19)$$

Assim, obtemos o controlador que satisfaz o critério:

$$C(s) = \frac{1}{(G_p(s) \cdot \lambda s)} \quad (2.20)$$

Algumas observações que devemos ter ao utilizarmos o método IMC:

- Uma vantagem de utilizar esse método de sintonia é que o desempenho do controlador está associado com a razão de constante de tempo em malha fechada ( $\lambda$ ) ou em malha aberta ( $\tau$ ), possibilitando definirmos de forma clara o desempenho do sistema.
- O único parâmetro que ajustamos no método IMC é o Lambda ( $\lambda$ ), no método robusto, o Lambda pode ser escolhido igual a constante de tempo dominante do processo.
- Caso o sistema possua muitas não linearidades, como histereses, bandas mortas e saturações, a sintonia deve ser mais conservadora. O mesmo se aplica caso possua um erro grande de modelagem.
- O tempo morto também dificulta a sintonia, assim acaba refletindo em um ( $\lambda$ ) maior.

Considerando um sistema de 1ª ordem com tempo morto, a tabela abaixo mostra as sintonias sugeridas:

Tabela 6 – Sintonia de controlados com tempo morto segundo Skogestad.

Modelo do processo	$K_p$	$T_i$	$T_d$
$\frac{K}{\tau s + 1} \cdot e^{-\theta s}$	$\frac{\tau}{K(\lambda + \theta)}$	$\min[\tau, 4(\lambda + \theta)]$	—
$\frac{K}{(\tau_1 \cdot s + 1)(\tau_2 \cdot s + 1)} \cdot e^{-\theta s}$	$\frac{\tau}{K(\lambda + \theta)}$	$\min[\tau, 4(\lambda + \theta)]$	$\tau_2$
$K \cdot e^{-\theta s}$	$\frac{1}{K}$	$\lambda + \theta$	—
$\frac{K}{s} \cdot e^{-\theta s}$	$\frac{\tau}{K(\lambda + \theta)}$	$4(\lambda + \theta)$	—
$\frac{K}{s(\tau_2 + 1)} \cdot e^{-\theta s}$	$\frac{\tau}{K(\lambda + \theta)}$	$4(\lambda + \theta)$	$\tau_2$

Segue abaixo mais algumas sintonias para diferentes tipos de sistemas analisados:

Tabela 7 – Sintonia de controlados sem tempo morto segundo Rivera.

Modelo do processo	$K_p$	$T_i$	$T_d$
$\frac{K}{\tau s + 1}$	$\frac{\tau}{K \cdot \lambda}$	$\tau$	—
$\frac{K}{(\tau_1 \cdot s + 1)(\tau_2 \cdot s + 1)}$	$\frac{\tau_1 + \tau_2}{K \cdot \lambda}$	$\tau_1 + \tau_2$	$\frac{\tau_1 \cdot \tau_2}{\tau_1 + \tau_2}$
$\frac{K}{s}$	$\frac{1}{K \cdot \lambda}$	—	—
$\frac{K}{s(\tau s + 1)}$	$\frac{1}{K \cdot \lambda}$	—	$\tau$

### 2.7.3.1 Método IMC para Plantas de 2<sup>a</sup> Ordem

(PEREIRA, 2011) baseia-se no artigo de (SKOGESTAD, 2003) para dizer que, embora muitos processos possam ser satisfatoriamente modelados por funções de transferência de primeira ordem, essa abordagem se torna limitada conforme a complexidade da planta aumenta. Nesses casos, os erros de modelagem se intensificam a ponto de inviabilizar a aplicação de técnicas simples de sintonia de controladores. Diante desse cenário, é fundamental recorrer a métodos de ajuste de controladores PID baseados em modelos de segunda ordem, que oferecem uma representação mais precisa do comportamento dinâmico do sistema.

No caso de plantas com comportamento de segunda ordem e sem atraso de tempo, (SKOGESTAD, 2003) propôs regras analíticas simples para a sintonia de controladores PID a partir de uma planta na forma:

$$G(s) = \frac{K(\tau_z s + 1)}{(\tau_1 s + 1)(\tau_2 s + 1)} \quad (2.21)$$

Para essa estrutura, as regras SIMC (Skogestad Internal Model Control) para o controlador PID resultante são:

$$K_p = \frac{\tau_1 + \tau_2}{K \cdot \lambda}, \quad T_i = \tau_1 + \tau_2, \quad T_d = \frac{\tau_1 \cdot \tau_2}{\tau_1 + \tau_2} \quad (2.22)$$

onde  $K$  é o ganho da planta,  $\tau_1$  e  $\tau_2$  são constantes de tempo dos pólos e  $\lambda$  é o parâmetro de suavização escolhido para o filtro.

### 2.7.3.2 Adaptação para Plantas em Forma Polinomial

Em muitos casos práticos, especialmente quando a planta é obtida por identificação experimental, sua função de transferência é expressa na forma polinomial:

$$G(s) = \frac{b_1 s + b_0}{s^2 + a_1 s + a_0} \quad (2.23)$$

Essa forma dificulta a aplicação direta das regras SIMC, que dependem da decomposição da planta em pólos ( $\tau_1, \tau_2$ ) e ganho ( $K$ ). Para contornar essa limitação, diversos autores propuseram fórmulas alternativas, compatíveis com a lógica do IMC, para estimar os parâmetros do controlador PID diretamente a partir dos coeficientes polinomiais.

As fórmulas aproximadas comumente utilizadas são:

$$K_p = \frac{a_0}{b_1}, \quad T_i = 2\lambda, \quad T_d = \frac{a_1}{a_0} \quad (2.24)$$

Essas expressões são obtidas por meio do método de “matching” de coeficientes da função de transferência da malha fechada, buscando manter a coerência entre a dinâmica da planta e a ação do controlador. A fórmula de  $K_p$  visa alinhar o ganho estático do sistema, enquanto  $T_i$  e  $T_d$  atuam sobre os termos de primeira e segunda ordem, garantindo uma resposta coerente com o modelo de referência. A escolha de  $T_i = 2\lambda$  é uma recomendação prática inspirada nas regras SIMC para casos sem fatoração explícita dos pólos.

Embora essas fórmulas não estejam formalmente derivadas nos artigos originais de (SKOGESTAD, 2003) ou em trabalhos que se apoiam nessas teorias como (PEREIRA, 2011), elas possuem bases teóricas e seus princípios estão presentes nos seu desenvolvimento. Além disso, são consideradas adequadas para aplicações em que o modelo da planta foi identificado diretamente, e têm se mostrado eficazes na prática. Essa abordagem é adotada em diversos materiais técnicos, tutoriais e implementações computacionais, como aulas práticas de controle automático, apostilas universitárias e blocos de sintonia automática em ambientes como *MATLAB/Simulink*.

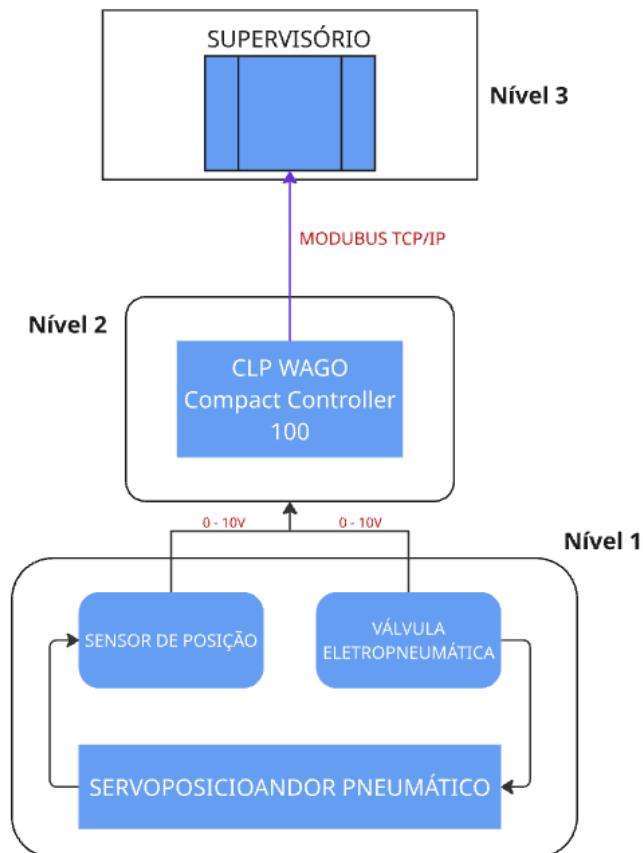
# 3 Metodologia

O trabalho proposto é uma continuação do que foi desenvolvido por (SANTOS, 2019) buscando outras abordagens do que foi executado anteriormente e com o intuito de propor possíveis melhorias a partir disso. Dessa maneira, a metodologia utilizada inicia-se do mesmo ponto de partida do que foi realizado no trabalho anterior.

## 3.1 Setup Experimental

O setup experimental do projeto foi composto por três níveis da pirâmide de automação, a parte dos dispositivos de campo que é a região de nível 1, composta por sensores e atuadores, o sistema de controle no nível 2, na qual foi feita toda a lógica de controle, e o sistema de supervisão no nível 3, onde é possível visualizar os dados e interagir com o sistema. A figura 23 apresenta a visão geral do sistema, contemplando todos os níveis citados acima.

Figura 23 – Visão Geral do Sistema.



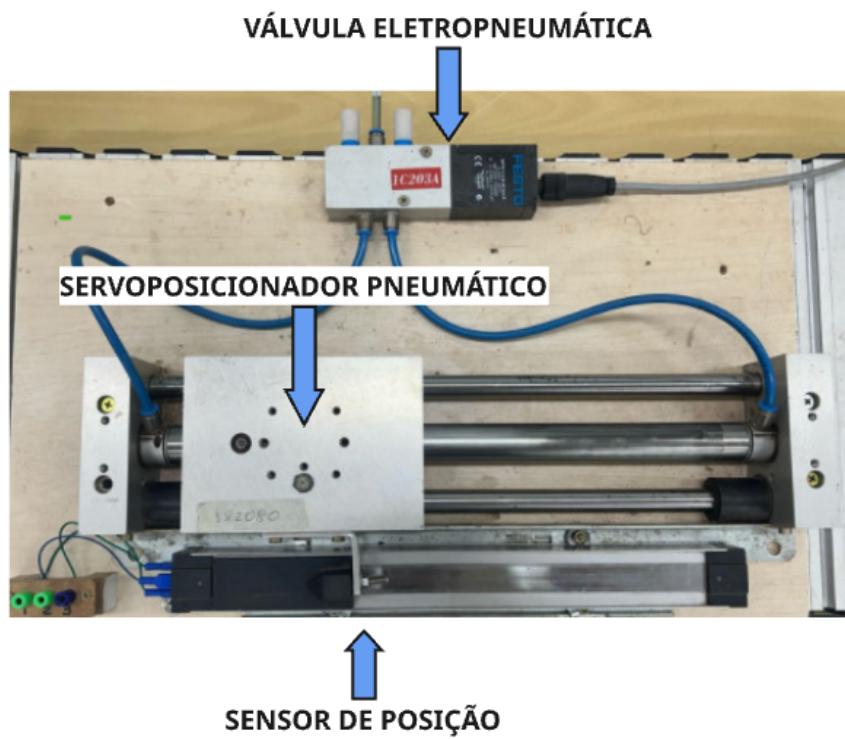
Fonte: Elaborado pelo autor.

### 3.1.1 Atuadores e Sensores

A estrutura do nível 1 que será responsável pela aquisição de sinais e execução dos comandos recebidos pelo controle é composta por um servoposicionador pneumático FESTO, uma válvula eletropneumática de controle proporcional FESTO MPYE-5-1/8-LF-010 B e um sensor de posição FESTO TLH 225.

O servoposicionador pneumático FESTO é um cilindro de dupla ação pneumático, com apenas um eixo de liberdade, com aproximadamente 40 centímetros de comprimento, contendo duas vias de alimentação de ar comprimido, cujo acionamento é realizado pela liberação de pressão da válvula proporcional para suas duas câmaras.

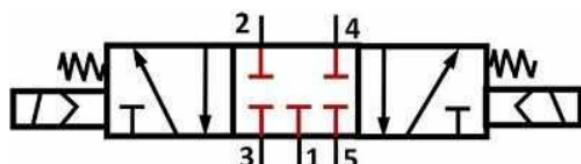
Figura 24 – Visão real da planta.



Fonte: Elaborado pelo autor.

A válvula utilizada é do tipo 5/3 vias de centro fechado, acionada por sinal eletrônico, onde a medida que é aplicada tensão elétrica no sinal de entrada, seu carretel interno é deslocado, liberando as passagens de ar em ambos os sentidos, porém com pressões proporcionais. A figura a seguir representa esta visão técnica da válvula.

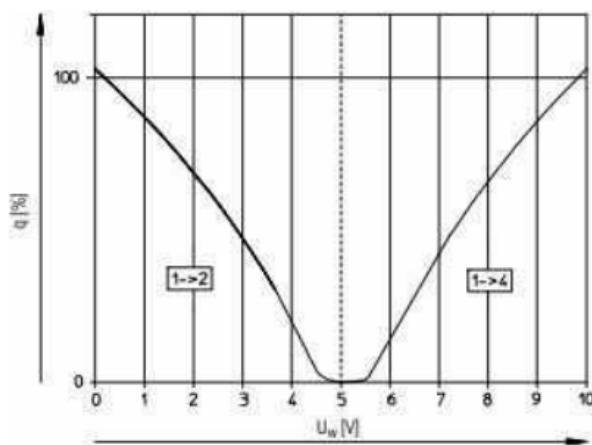
Figura 25 – Visão técnica da válvula.



Fonte: (FESTO, 2025b).

Para funcionamento, é necessário alimentá-la com fonte de tensão contínua de 24 volts. O sinal de controle a ser enviado para a válvula possui uma faixa de tensão entre 0 e 10 V, em que na faixa entre 0 e 5 V, o cilindro apresenta movimento de recuo, alternando sua velocidade, ficando mais rápido conforme a tensão for próxima de zero. Na faixa entre 5 e 10 V, apresenta movimento de avanço, possuindo maiores velocidades conforme se aproxima dos 10 volts. A válvula não enviará ar para nenhum canal quando for alimentada com 5 volts, ou seja o sistema estará no seu ponto estável. O gráfico presente na figura 26 representa a curva característica de vazão em relação a tensão aplicada na válvula e demonstra o comportamento descrito previamente:

Figura 26 – Curva característica de vazão por tensão.



Fonte: ([FESTO, 2025b](#)).

Por fim, o conjunto atuador/sensor é composto por um sensor de posição FESTO TLH225, que indica a posição atual do cilindro. Este sensor é um potenciômetro linear com comprimento aproximado de 30 centímetros, alimentado com 10 volts e fornecerá um sinal de saída entre 0 a 10 V, sendo 0 a posição mais a esquerda e 10 mais a direita. O sensor é representado na figura a seguir.

Figura 27 – Sensor de posição FESTO TLH225.



Fonte: ([FESTO, 2025a](#)).

### 3.1.2 Sistema de Controle

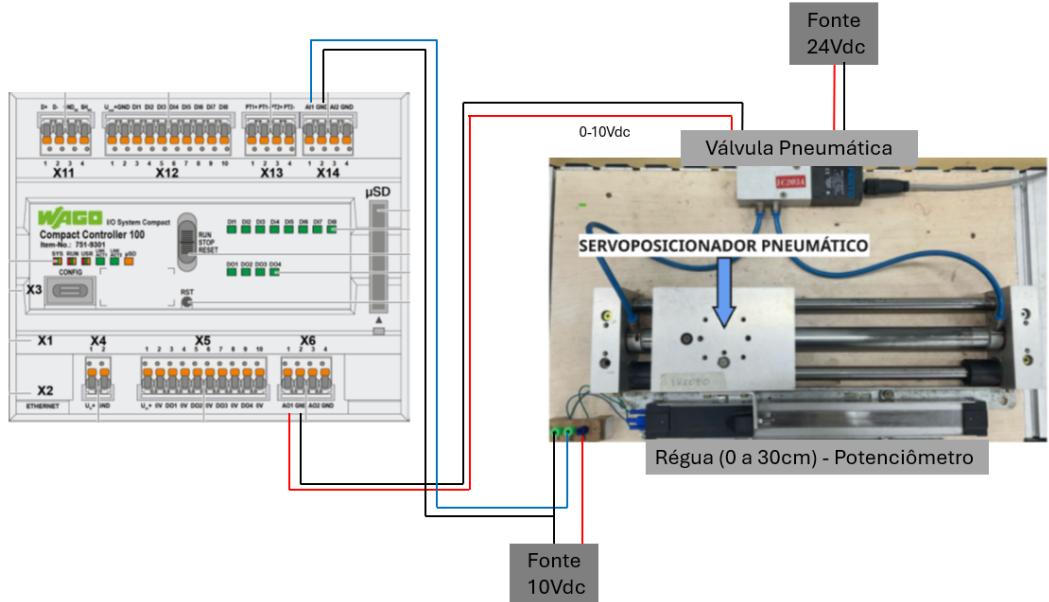
A princípio, o projeto proposto previa o uso de dois CLPs para realizar o controle do processo, visto que seria utilizado o CLP Siemens S7-1200 para realizar o controle e uma vez que este não possui um módulo de entrada e saída analógica seria necessário utilizar o TWDLCAA24DRF da Schneider, um controlador mais antigo mas que possui entradas e saídas analógicas e portando faria o papel de um remota. Esse processo seria semelhante ao realizado no trabalho de (SANTOS, 2019).

Contudo, no inicio da execução do projeto proposto, após vários testes iniciais, foi identificado que o módulo analógico do CLP TWDLCAA24DRF da Schneider não estava funcionando, tornando necessário encontrar outra alternativa para aquisição e escrita de dados e posterior envio para o CLP S7-1200 realizar o controle.

A alternativa encontrada foi a utilização do CLP WAGO Compact Controller 100, que havia sido recentemente adquirido pelo laboratório de Controle e Automação e que, apesar de não ter sido utilizado anteriormente, havia entradas e saídas analógicas. Após realização de teste confirmou-se a viabilidade do uso deste controlador e uma vez que este é um controlador moderno, robusto e de alto desempenho, além de possuir uma conectividade versátil, suportando diversos protocolos de comunicação industrial e uma programação flexível, sendo compatível com CODESYS V3, optou-se por prosseguir com o desenvolvimento do projeto apenas com este CLP, fazendo o papel de aquisição e envio de sinais, além do controle geral do processo. Essa escolha permitiu encurtar a linha de comunicação, eliminando o intermédio de uma remota e diminuindo o tempo de resposta do processo.

Desse modo, o nível 2 do setup experimental ficou composto pelo CLP WAGO Compact Controller 100, estando este conectado a uma fonte de alimentação de 24 V. Ademais, no conjunto de conexões “X14”, identificado pela figura 28, na entrada “AI1” foi conectado a saída do sensor de posição e no conjunto “X6”, na saída “AO1”, foi conectado a entrada da válvula. E para que fosse possível realizar a configuração do controlador, foi feita a conexão entre ele e o computador utilizado através de um cabo Ethernet. As ligações são demostradas na figura 28.

Figura 28 – Entradas e saídas do CLP WAGO Compact Controller 100.



Fonte: Adaptado de ([WAGO, 2024](#)).

### 3.1.3 Sistema de Supervisão

O sistema de supervisão foi feito utilizando o programa Elipse E3, em que foi desenvolvida uma interface que possibilita observar um gráfico de variáveis do sistema, tais como:

1. Variável controlada ou de processo (PV);
2. Variável manipulada (MV);
3. Setpoint do sistema (SP);

O supervisório foi conectado ao CLP utilizando o protocolo Modbus TCP/IP, desse modo, ele é o cliente e o CLP é o servidor, pois, apesar do controle ser realizado localmente no controlador, o supervisório estará fazendo solicitações através da mudança de variáveis como o Setpoint e os parâmetros do PID.

## 3.2 Configuração Controlador Lógico Programável

Com as conexões entre a planta e o controlador estabelecidas, foi feita a configuração para posterior programação do CLP. Com o software CODESYS instalado juntamente com as bibliotecas necessárias disponibilizadas pela fabricante do controlador, iniciou-se por atribuir um IP ao equipamento, uma vez estabelecida a conexão com o notebook através de um cabo ethernet, foi definido o endereço 192.168.0.10.

Após isso, foram definidas as entradas e saídas físicas e seus respectivos endereços, utilizou-se a entrada analógica 1 com endereço %IW3 nomeada de “leituraSensor” e a saída analógica 1 com o endereço %QW1 nomeada de “comandoValvula”. Após essa definição, já foi possível acompanhar a leitura e escrita de variáveis. Por definição padrão do CODESYS, as variáveis analógicas são definidas como tipo WORD, possuindo um range de 0 a 32767, ou seja, a leitura de dados vindo do sensor de posição e a escrita na válvula eletropneumática que no meio físico variam de 0 a 10V, são lidas pelo programa dentro do range padrão, sendo 0V equivalente a 0 e 10V equivalente a 32767.

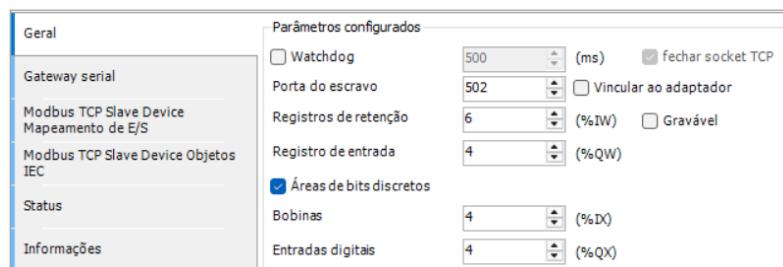
Figura 29 – Mapeamento das Variáveis Físicas.

Onboard-IO Parâmetros		Buscar						
		Filtro		Mostrar tudo				
		Variável	Mapeamento	Canal	Endereço	Tipo	Valor Padrão	Valor atual
	Onboard-IO Mapeamento de E/S			X12.DI	%IW0	BYTE		Não atualizado (consulte a dica de ferramenta)
	Status			X13.PT1+/X13.PT1-	%IW1	INT		Não atualizado (consulte a dica de ferramenta)
	Informações			X13.PT2+/X13.PT2-	%IW2	INT		Não atualizado (consulte a dica de ferramenta)
		leituraSensor		X14.AI1	%IW3	WORD	64	Não atualizado (consulte a dica de ferramenta)
				X14.AI2	%IW4	WORD		Não atualizado (consulte a dica de ferramenta)
				X5.DO	%Q0	BYTE		Não atualizado (consulte a dica de ferramenta)
		comandoValvula		X6.AO1	%QW1	WORD	0	16383
				X6.AO2	%QW2	WORD		Não atualizado (consulte a dica de ferramenta)

Fonte: Elaborado pelo autor.

Para que houvesse comunicação do controlador com o sistema supervisório, foi preciso adicionar um novo *device* dentro do programa no CODESYS, primeiramente foi adicionada uma placa de rede ethernet e depois o *device* “ModbusTCP Slave Device”, uma vez que, como foi definido anteriormente, o CLP será o servidor da comunicação Modbus com o supervisório. Tendo habilitado a comunicação modbus no nosso programa, foi preciso configurar os parâmetros do protocolo. Foi mantida a porta padrão do escravo como “502”, adicionou-se 6 registros de retenção (%IW), 4 registros de entrada (%QW), 4 bobinas (%IX) e 4 entradas digitais (%QX).

Figura 30 – Configuração Parâmetros Modbus.

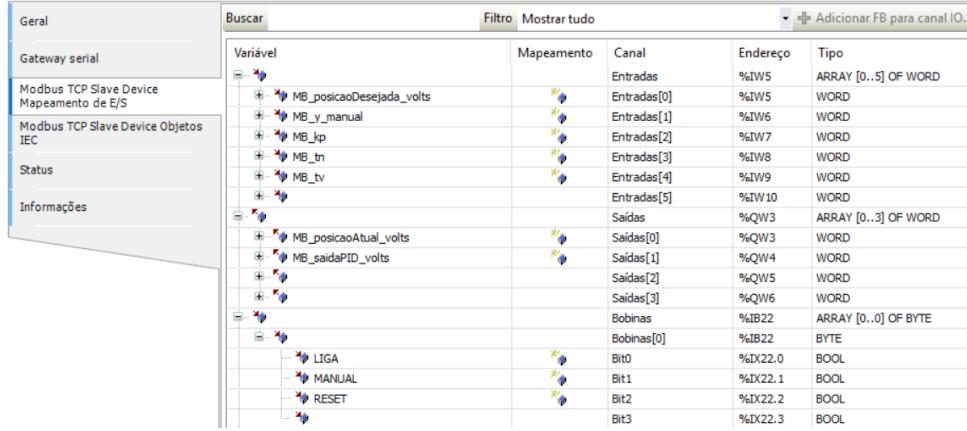


Fonte: Elaborado pelo autor.

Além disso, foram definidas as variáveis a serem utilizadas na comunicação entre o controlador e o ELIPSE E3 e seus respectivos endereços modbus. Como bobinas, foi adicionado o botão “LIGA” no bit 0, “MANUAL” no bit 1 e “RESET” no bit 2. Já para as variáveis analógicas, que assim como as do meio físico, são definidas como tipo WORD, definiu-se, como entradas: “MB\_posicaoDesejada\_volts” no endereço %IW5, “MB\_y\_manual”

em %IW6, “MB\_kp” em %IW7, “MB\_tn” em %IW8 e “MB\_tv” em %IW9. Como saídas: “MB\_posicaoAtual\_volts” no endereço %QW3 e “MB\_saidaPID\_volts” em %QW4.

Figura 31 – Mapeamento das Variáveis Físicas Modbus.



Geral	Buscar	Filtro	Mostrar tudo		Adicionar FB para canal IO...
	Variável	Mapeamento	Canal	Endereço	Tipo
Gateway serial			Entradas	%IW5	ARRAY [0..5] OF WORD
Modbus TCP Slave Device	MB_posicaoDesejada_volts		Entradas[0]	%IW5	WORD
Mapeamento de E/S	MB_y_manual		Entradas[1]	%IW6	WORD
Modbus TCP Slave Device Objetos IEC	MB_kp		Entradas[2]	%IW7	WORD
Status	MB_tn		Entradas[3]	%IW8	WORD
Informações	MB_tv		Entradas[4]	%IW9	WORD
			Entradas[5]	%IW10	WORD
			Saídas	%QW3	ARRAY [0..3] OF WORD
	MB_posicaoAtual_volts		Saídas[0]	%QW3	WORD
	MB_saidaPID_volts		Saídas[1]	%QW4	WORD
			Saídas[2]	%QW5	WORD
			Saídas[3]	%QW6	WORD
			Bobinas	%IB22	ARRAY [0..0] OF BYTE
			Bobinas[0]	%IB22	BYTE
	LIGA		Bit0	%IX22.0	BOOL
	MANUAL		Bit1	%IX22.1	BOOL
	RESET		Bit2	%IX22.2	BOOL
			Bit3	%IX22.3	BOOL

Fonte: Elaborado pelo autor.

### 3.2.1 Lógica de Controle

Com as configurações necessárias concluídas, prosseguiu-se com a implementação da lógica de controle do CLP. Foi incluída uma POU do tipo programa e linguagem de implementação “Gráfico da linguagem Ladder (LD)” na aplicação e definida como “MainTask”. Com o intuito de facilitar o acompanhamento dos valores das variáveis durante a execução do programa, optou-se por realizar a declaração das variáveis previamente definidas no mapeamento diretamente no programa. Além das variáveis com endereço, foram definidas algumas outras variáveis auxiliares à serem utilizadas, como “posicaoAtual\_volts”, “saidaPID\_volts”, “kp”, “tn” e “tv”, que serão utilizadas para realizar conversões.

Figura 32 – Variáveis declaradas no código Ladder.

```

1 PROGRAM IDENT_PID
2 VAR
3   MB_posicaoDesejada_volts AT %IW5 : WORD;
4   MB_y_manual AT %IW6 : WORD;
5   MB_kp AT %IW7 : WORD;
6   MB_tn AT %IW8 : WORD;
7   MB_tv AT %IW9 : WORD;
8   MB_posicaoAtual_volts AT %QW3 : WORD;
9   MB_saidaPID_volts AT %QW4 : WORD;
10
11 leituraSensor AT %IW3 : WORD;
12 comandoValvula AT %QW1 : WORD;
13
14 posicaoAtual_volts : REAL;
15 saidaPID_volts : REAL;
16
17 kp : REAL;
18 tn : REAL;
19 tv : REAL;
20
21 yMin : REAL := 0;
22 yMax : REAL := 10000;
23
24 y_manual : REAL;
25
26 PID_1: PID;
27 END_VAR

```

Fonte: Elaborado pelo autor.

Como explicado anteriormente, por definição padrão do CODESYS, as variáveis analógicas são definidas como tipo WORD, possuindo um range de 0 a 32767. Dessa maneira, com o intuito de facilitar a compreensão dos dados, optou-se por linearizar essas variáveis. Para isso, foi utilizado um bloco "EXECUTE", que permite escrever códigos na linguagem ST a serem executados mesmo na linguagem Ladder.

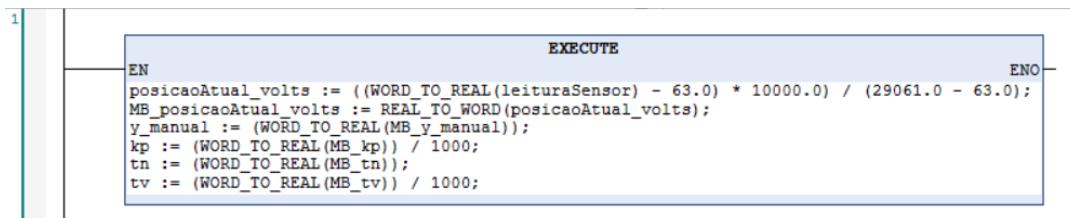
Iniciando pela linearização da variável de entrada "leituraSensor", que corresponde a leitura da tensão equivalente à posição do sensor. Como descrito na etapa do setup experimental, o sensor é um potenciômetro linear que informa a posição do servoposicionador através de um sinal de saída entre 0 a 10V, contudo, o trilho do posicionador tem uma limitação física que não deixa que a peça alcance as duas extremidades, sendo assim, a leitura do sensor varia entre aproximadamente 0,2 até 8,9V, o que, na leitura do CODESYS corresponde de 63 a 29061. Com o intuito de facilitar a manipulação dos dados, foi feita a opção de obter através da linearização um valor de 0 a 10000. Sendo assim, foi obtida a fórmula:

$$posicaoAtual\_volts = \frac{(leituraSensor - 63) \cdot 10000}{29061 - 63} \quad (3.1)$$

Além disso, foi necessário fazer uma transformação no tipo da variável, passando de WORD para REAL, já que é o tipo aceito pelo bloco PID que será usado posteriormente.

Ainda neste bloco "EXECUTE", foi feita a atribuição das variáveis "posicaoAtual.volts" para a variável "MB\_posicaoAtual.volts", que será por onde vai ser obtida a informação da posição do sensor no sistema supervisório, da variável "MB\_y\_manual" para a variável "y\_manual", "MB\_kp" para "kp", "MB\_tn" para "tn" e "MB\_tv" para "tv", sendo essas por onde serão escritos os parâmetros do PID através do supervisório.

Figura 33 – Primeiro Bloco de Execução em ST.

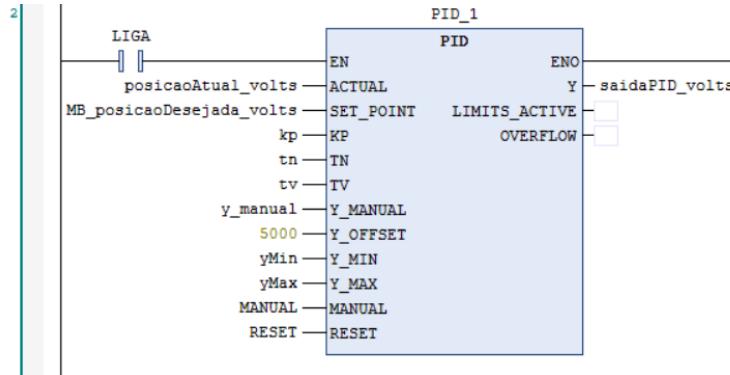


Fonte: Elaborado pelo autor.

Após o bloco de linearização da entrada e atribuição de algumas variáveis da comunicação modbus, tem-se o bloco PID. Para esta aplicação, optou-se por utilizar o bloco do função PID da biblioteca "Util". Com o bloco inserido na lógica, foram realizadas as atribuições das variáveis previamente definidas nos parâmetros do bloco, "posicaoAtual.volts" como "ACTUAL", "MB\_posicaoDesejada.volts" como "SET\_POINT" e "saidaPID.volts" como "Y". Ademais, foi adicionado o botão "LIGA" entrada "EN",

definiu-se o “Y\_MIN” como 0 e “Y\_MAX” como 10000 e atribuído um “Y\_OFFSET” de 5000, já que o ponto de estabilidade do sistema é neste valor.

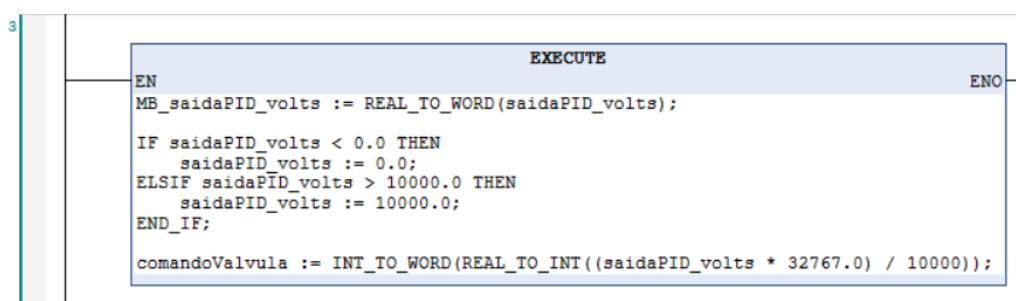
Figura 34 – Bloco PID da Linguagem Ladder.



Fonte: Elaborado pelo autor.

Por último, foi utilizado outro bloco “EXECUTE” na linguagem ST onde, primeiramente atribuiu-se o valor de “saidaPID\_volts” para a variável “MB\_saidaPID\_volts”, em seguida, o valor da saída do bloco foi limitado entre 0 e 10000 e por último, o valor que havia sido linearizado retornou para estar entre 0 e 10000 para do valor padrão das variáveis do tipo WORD do CODESYS, entre 0 e 32767 e atribuiu-se esse novo valor para a variável física do sistema “comandoValvula”, que irá resultar em um sinal enviado para a válvula entre 0 e 10V.

Figura 35 – Segundo Bloco de Execução em ST.



Fonte: Elaborado pelo autor.

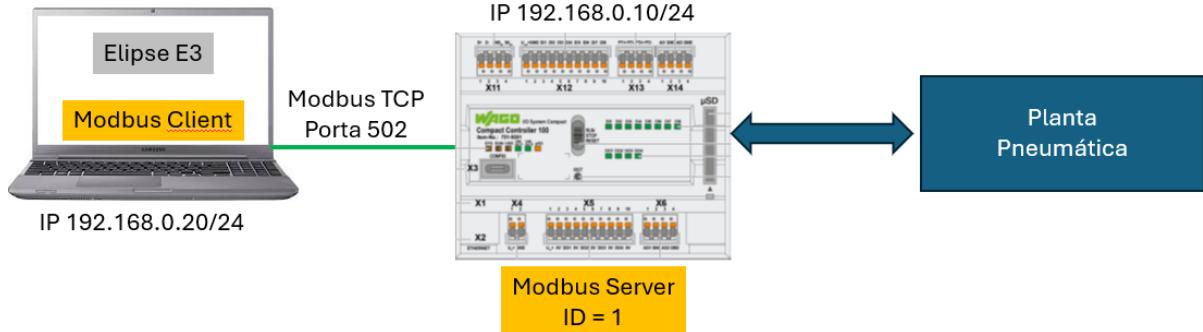
### 3.3 Sistema Supervisório

#### 3.3.1 Comunicação Modbus

A comunicação do sistema supervisório desenvolvido no ELIPSE E3 com o controlador foi feita utilizando o protocolo Modbus TCP, para isso, primeiramente foi necessário inserir um driver de comunicação Modbus para que o supervisório atue como cliente. Foi feita a configuração do driver definindo o protocolo como ModbusTCP e endereços dos dados iniciando de 1. Além disso, definiu-se a camada física como Ethernet direcionando a

conexão para o IP 192.168.0.10 na porta 502, que foram as configurações do CLP definidas anteriormente.

Figura 36 – Conexão Cliente Servidor do Supervisório com o CLP.



Fonte: Elaborado pelo autor.

Após a adição do driver, foi feita a configuração de variáveis para realizar a aquisição de dados e interação com o sistema. Com isso, os dados que serão coletados do processo e as variáveis que poderão ser manipuladas são representadas pelas “Tags”, definidas para satisfazer a necessidade do sistema, representadas na figura 37.

Figura 37 – Tags do Sistema Supervisório.

Nome	Disp...	Item	P1/N...	P2/N...	P3/N...	P4/N...	Tamanho/Índice	Varredura	Leitur...	Escrit...	Escala?	Mín. UE	Máx. ...	UE	Mín. E/S	Máx. E/S	
Modbus			0	0	0	0						0	10000		0	10000	
• MB_A11				1	8	0	1					1000	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	10000
• MB_AO1				1	8	0	2					1000	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	10000
• SET_POINT				1	1	0	1					1000	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	10000
• y_MANUAL				1	1	0	2					1000	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	10000
• kp				1	1	0	3					1000	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1000
• tn				1	1	0	4					1000	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1000
• tv				1	1	0	5					1000	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1000
• LIGA_DESL				1	6	0	1					1000	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1
• MANUAL				1	6	0	2					1000	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1
• RESET				1	6	0	3					1000	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1000

Fonte: Elaborado pelo autor.

Como pode ser observado na figura acima, para a configuração de tags no Eipse E3, é necessário definir 4 parâmetros, o primeiro, “P1” é o endereço do servidor no CLP e por padrão foi definido como 1 ou 255. O segundo, “P2” é o tipo de operador a ser trabalhado, cada uma dessas operações define uma função modbus de leitura e escrita e o tipo do dado. O “P3” não é utilizado e o “P4” é o endereço de cada variável.

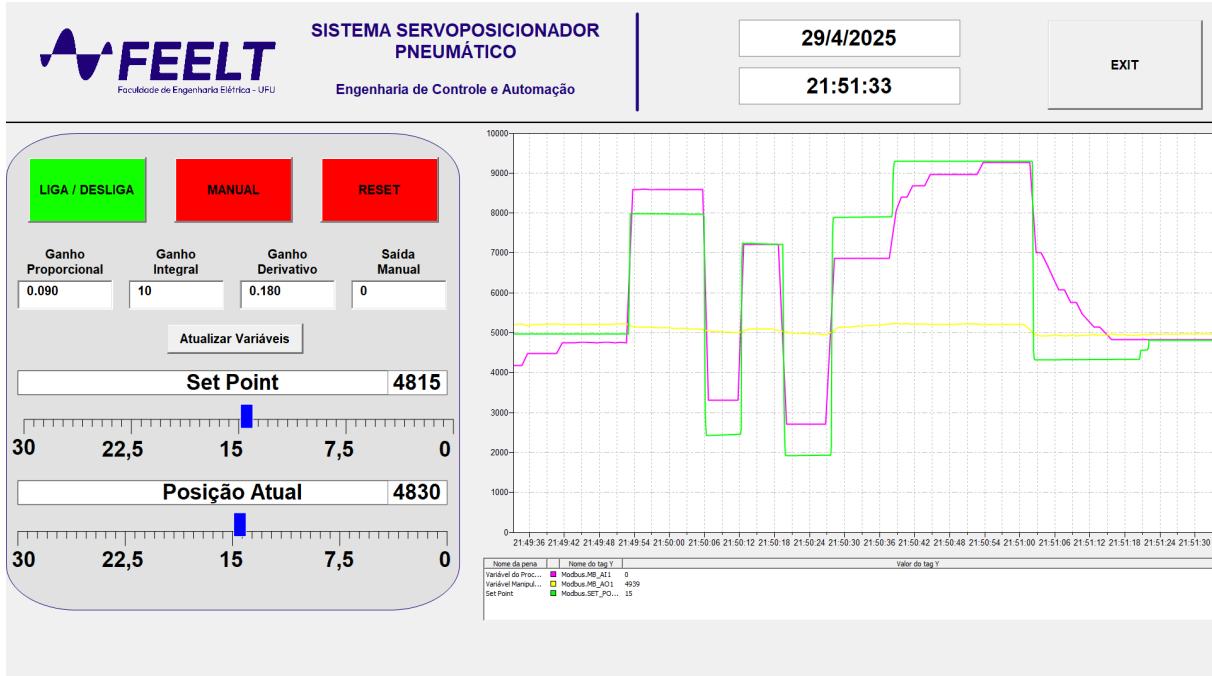
Para atender a necessidade do projeto foi criada uma função do parâmetro “P2” de número 8 que utiliza a função modbus 4 de leitura de bloco de registradores do tipo input no tipo de dado “word”. Essa função foi utilizada para realizar a leitura das saídas analógicas mapeadas no modbus. A função 1 utiliza a função modbus 3 de leitura de bloco de registradores do tipo holding e a função 16 de escrita de bloco de registradores do tipo holding no tipo de dado “word” e foi utilizada para realizar leitura e escrita das entradas analógicas mapeadas no modbus. Para as variáveis do tipo bobinas, utilizou-se a função 6

que engloba a função 1 de leitura de bloco de bits do tipo coil e a função 15 de escrita de bloco de bits do tipo coil no tipo de dado “bit”.

### 3.3.2 Tela do Sistema Supervisório

Após a configuração das variáveis que serão utilizadas, foi desenvolvida a tela do sistema supervisório para visualização dos dados de controle e atuação no sistema. A figura 38 representa o sistema desenvolvido.

Figura 38 – Tela do Sistema Supervisório.



Fonte: Elaborado pelo autor.

A tela conta com informações de data e hora e um botão “EXIT” para fechar a tela do sistema. Além disso, possui uma área de interação com o CLP, onde comandos são enviados e dados são recebidos através de botões: “LIGA/DESLIGA”, “MANUAL” e “RESET” que acionam os respectivos botões na lógica do controlador. Ademais, existem campos para alteração dos parâmetros do bloco PID e dois “sliders” de posição, sendo o primeiro para alterar o “Setpoint” do sistema e o outro que informa a posição atual do servoposicionador no sensor de posição.

Para visualização das variáveis do sistema, foi implementado um gráfico que a cada 20 ms atualiza as informações de “posição atual”, “Setpoint” e “comando da válvula”. Além do gráfico, foi criado um histórico que a cada 20 ms registra as informações de entrada e saída do sistema juntamente com um marcador de tempo num banco de dados Access com o intuito de guardar as informações obtidas e auxiliar na etapa futura de identificação do sistema.

# 4 Resultados e Discussões

Nesse capítulo são mostrados os resultados obtidos nesse trabalho. O capítulo inicia com testes realizados para analisar o funcionamento do processo de aquisição e envio de dados entre o CLP e a planta, em seguida, será feita a validação do controle local, da comunicação com o sistema supervisório e por último o processo de identificação do sistema e a sintonia do controlador PID.

## 4.1 Aquisição e Envio de Dados

Inicialmente, o projeto previa o uso do CLP TWDLCAA24DRF da Schneider para realizar o processo de aquisição de dados e envio para o CLP Siemens S7-1200 realizar o controle. Esse processo se mostrou problemático, uma vez que o software utilizado para controladores da Schneider não é compatível com sistemas operacionais mais atuais, tendo sido necessário a criação de uma máquinas virtual com Windows 7 para realizar os testes necessários. Com o software em funcionamento, os testes constataram que o módulo analógico do controlador não estava funcionando, sendo necessário reavaliarmos o projeto proposto.

Contudo, ao definir-se pela utilização do CLP WAGO Compact Controller 100, como explicado no tópico [3.1.2](#) o processo de aquisição e envio de dados não apresentou mais problemas, tendo sido necessário apenas conectar as entradas e saídas analógicas do controlador com as da planta e mapear essas variáveis no CODESYS.

### 4.1.1 Validação Comunicação CLP e Planta

Para validação da comunicação entre o CLP e a planta foram feitos testes de leitura e escrita de variáveis na saída do sensor de posição e na entrada da válvula. Para monitorar os valores registrados na planta foi utilizado um multímetro comparando os valores obtidos com o que era registrado pelo controlador. As figuras [39](#) e [40](#) mostram esta validação.

Figura 39 – Leitura da saída do sensor de posição pelo multímetro.



Fonte: Elaborado pelo autor.

De acordo com a figura 39, a medição da tensão de saída do sensor de posição foi de 4,755V, sendo que, como especificado anteriormente, devido à limitações físicas do trilho do posicionador, esse valor varia na faixa de 0,2 até 8,9V. Já a figura 40 mostra a leitura realizada pelo CLP correspondente a da figura 39 onde o valor lido foi de 15605 na variável “leituraSensor” que é o valor antes da linearização, variando de 63 a 29061, e 5352 na variável “posicaoAtual\_volts” que é o valor linearizado de 0 a 10000. Apesar das diferenças de escala, ao linearizar os valores obtidos na escala de 0 a 10000, a medição do multímetro equivale a 5235,63, a leitura da variável “leituraSensor” equivale a 5324,29 e a “posicaoAtual\_volts” é mantida em 5352. Conclui-se que apesar de uma diferença de 1,69% entre a medição física e a variável “leituraSensor” e de 2,22% entre a leitura do multímetro e a variável “posicaoAtual\_volts”, a diferença é consideravelmente pequena, validando a equivalência nas medições.

Figura 40 – Leitura da saída do sensor de posição pelo CLP.

Expressão:	Tipo	Valor	Valor p...	Endere...	Comen...
MB_posicaoDesejada_volts	WORD	5350		%IW5	
MB_y_manual	WORD	0		%IW6	
MB_kp	WORD	0		%IW7	
MB_tn	WORD	0		%IW8	
MB_tv	WORD	0		%IW9	
MB_posicaoAtual_volts	WORD	5352		%QW3	
MB_saidaPID_volts	WORD	5131		%QW4	
leituraSensor	WORD	15605		%IW3	
comandoValvula	WORD	16812		%QW1	
posicaoAtual_volts	REAL	5352.113			
saidaPID_volts	REAL	5130.7...			

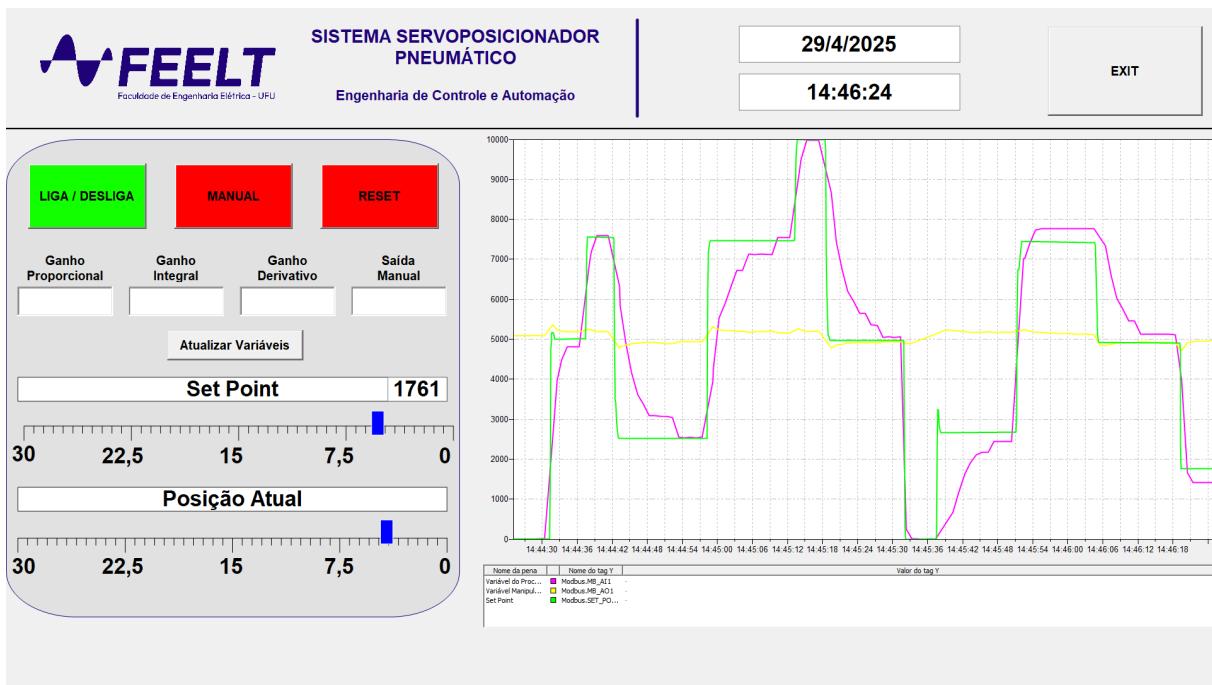
Fonte: Elaborado pelo autor.

## 4.2 Controle Local

Com o intuito de realizar testes do controle e do bloco PID, foram atribuídos valores empíricos aos parâmetros do bloco. Encontrar valores para os testes se mostrou um grande desafio, uma vez que o sistema é bastante instável nas extremidades, e apresenta uma resposta muito rápida para valores abaixo de 4,5V e acima de 5,5V aplicados na válvula.

A partir destas características e de testes com vários valores diferentes, chegou-se nos valor de  $K_p = 0,09$ , de modo a diminuir as variações bruscas de valores mesmo que a resposta seja mais lenta.  $T_i = 5$  para reduzir o erro em regime permanente e  $T_d = 0$ . Utilizando o gráfico do sistema supervisório, é possível validar o funcionamento do controle realizado pelo bloco PID utilizando os parâmetros definidos empiricamente.

Figura 41 – Gráfico do controle realizado com parâmetros empíricos.



Fonte: Elaborado pelo autor.

O gráfico indica que os parâmetros adotados foram suficientes para observar o funcionamento adequado do nosso controle PID e que o resultado se aproxima do desejado, apesar disso, ainda é visível que existe um erro de regime permanente e que a reposta do sistema é um pouco demorada. Desse modo, buscou-se outros métodos mais adequados para refinar nosso controle, acelerar o tempo de resposta e diminuir o erro presente no sistema.

## 4.3 Validação da Comunicação com o Supervisório

A comunicação do sistema supervisório (Eipse E3) e o controlador (WAGO Compact Controller 100), apresentou funcionamento adequado, podendo ser evidenciado,

primeiramente na análise do parâmetro “Qualidade” presente em cada uma das “Tags” utilizadas no processo de comunicação, onde valores entre 0 - 63 indicam qualidade ruim, ou seja, valor não útil, de 64 - 127 qualidade incerta e de 192 - 255 qualidade boa ([Elipse Software, 2019](#)). A figura 42 demonstra o indicador de qualidade quando é utilizado um endereço incorreto para uma variável comparado as outras variáveis corretas.

Figura 42 – Qualidade das variáveis utilizadas no supervisório.

Nome	Disp...	Item	P1/N1/...	P2/N...	P3/N...	P4/N...	Tamanho/Índice	Varredura	Valor	Qualidade	Estampa de tempo	Valor (sem escala)
Modbus			0	0	0	0						
• MB_AI1			1	8	0	7		1000	9	5349	20/04/2025 15:23:24,239	9 5349
• MB_AO1			1	8	0	2		1000	9	5133	20/04/2025 15:23:26,245	9 5134
• SET_POINT			1	1	0	1		1000	9	5350	192/04/2025 15:10:03,290	9 5350
• y_MANUAL			1	1	0	2		1000	9	0	192/04/2025 15:10:03,290	9 0
• kp			1	1	0	3		1000	9	0	192/04/2025 15:22:32,201	9 0
• tn			1	1	0	4		1000	9	0	192/04/2025 15:10:03,290	9 0
• tv			1	1	0	5		1000	9	0	192/04/2025 15:10:03,290	9 0
• LIGA_DESL			1	6	0	1		1000	9	1	192/04/2025 15:10:03,340	9 1
• MANUAL			1	6	0	2		1000	9	0	192/04/2025 15:10:03,340	9 0
• RESET			1	6	0	3		1000	9	0	192/04/2025 15:10:03,340	9 0

Fonte: Elaborado pelo autor.

Além disso, é possível realizar um teste semelhante a validação da comunicação entre CLP e planta para verificar a comunicação com o supervisório. Para isso, foi feita a leitura da tensão enviada para a válvula utilizando um multímetro e a comparou com o valor obtido pelo supervisório. A figura 43 mostra que a medição indicou que estão sendo enviados 5,127V para a válvula sendo que este valor varia entre 0 e 10V. Já a figura 44 indica que o valor lido pelo supervisório na tag “MB\_AO1” que recebe o valor de “MB\_saidaPID\_volts” do CLP foi de 5133, com este valor podendo variar de 0 a 10000, equivalente a 5,133V, com um erro percentual de 0,117%. Mais uma vez, validando a equivalência nas medições.

Figura 43 – Leitura da entrada da válvula pelo multímetro.



Fonte: Elaborado pelo autor.

Figura 44 – Leitura da entrada da válvula pelo supervisório.

Nome	Disp...	Item	P1/N1/...	P2/N...	P3/N...	P4/N...	Tamanho/Índice	Varredura	Valor	Qualidade	Estampa de tempo	Valor (sem escala)
Modbus			0	0	0	0						
• MB_AII			1	8	0	1		1000	9	5348	192 /04/2025 15:39:15,240	9 5348
• MB_AO1			1	8	0	2		1000	9	5133	192 /04/2025 15:39:16,238	9 5133
• SET_POINT			1	1	0	1		1000	9	5350	192 /04/2025 15:10:03,290	9 5350
• y_MANUAL			1	1	0	2		1000	9	0	192 /04/2025 15:10:03,290	9 0
• kp			1	1	0	3		1000	9	0	192 /04/2025 15:22:32,201	9 0
• tn			1	1	0	4		1000	9	0	192 /04/2025 15:10:03,290	9 0
• tv			1	1	0	5		1000	9	0	192 /04/2025 15:10:03,290	9 0
• LIGA_DESL			1	6	0	1		1000	9	1	192 /04/2025 15:10:03,340	9 1
• MANUAL			1	6	0	2		1000	9	0	192 /04/2025 15:10:03,340	9 0
• RESET			1	6	0	3		1000	9	0	192 /04/2025 15:10:03,340	9 0

Fonte: Elaborado pelo autor.

## 4.4 Identificação do Sistema e Sintonia do Controlador

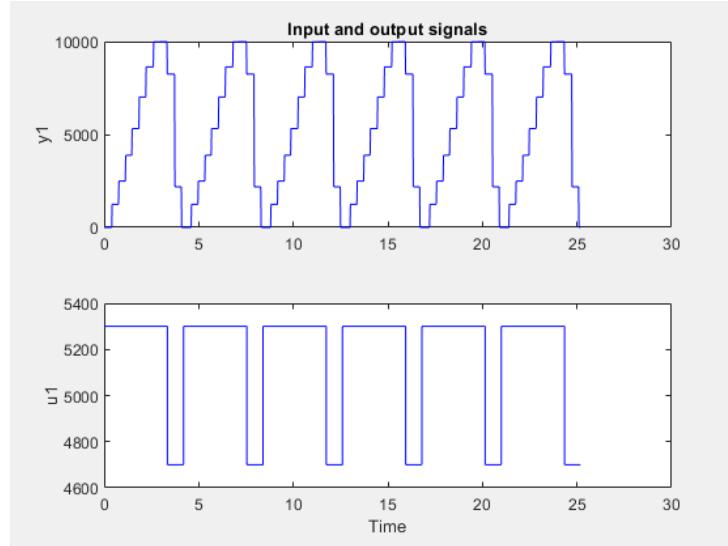
Utilizando as teorias dos tópicos 2.6 e 2.7 iniciou-se o processo de identificação do sistema. Para isso, foi utilizado o método da caixa preta, uma vez que não é conhecido a modelagem matemática teórica do sistema, foi necessário realizar a coleta de dados dos sinais de entrada e saída da planta a partir da aplicação de um degrau no sistema em malha aberta.

Para realizar esta coleta, foi ativado o modo “MANUAL” do bloco PID, de forma a definir o valor de saída “y\_manual” do controlador que varia de 0 a 10000. Uma vez que, como citado anteriormente, o sistema está estacionário quando a válvula é alimentada com 5 volts e que para valores de saída baixos ou elevados a resposta é muito rápida, foram realizadas coletas de dados utilizando 7 padrões diferentes de variação considerando os valores linearizados.

- Degrau positivo de 5200 e negativo de 4800.
- Degrau positivo de 5250 e negativo de 4750.
- Degrau positivo de 5300 e negativo de 4700.
- Degrau positivo de 5350 e negativo de 4650.
- Degrau positivo de 5400 e negativo de 4600.
- Degrau positivo de 5450 e negativo de 4550.
- Degrau positivo de 5500 e negativo de 4500.

A coleta foi realizada utilizando o histórico do sistema supervisório registrado no banco de dados Access. Uma vez finalizada, exportou-se as informações para um arquivo Excel, que em seguida, foi importado no Matlab, onde foi salvo a coluna com os valores do degrau aplicado em cada uma das coletas como  $u_n$  e os valores correspondentes de saída como  $y_n$ , com  $n$  de 1 a 7. A figura 45 mostra um exemplo do comportamento do sistema em malha aberta mediante a aplicação do degrau com os valores de 5300 e 4700.

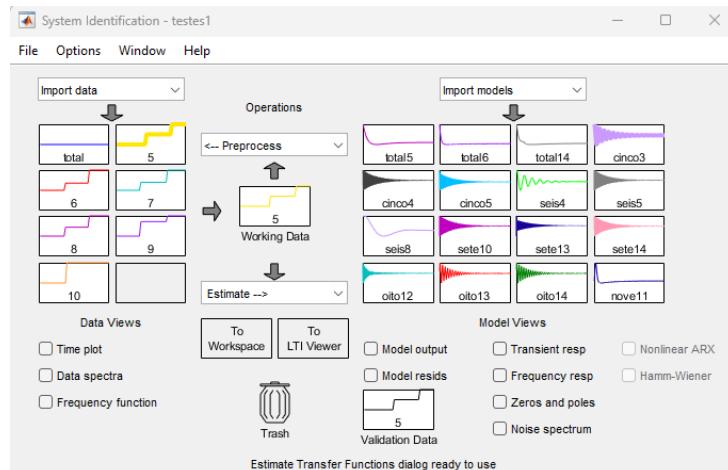
Figura 45 – Dados coletados para identificação do sistema com degrau 5300 e 4700.



Fonte: Elaborado pelo autor.

Com os dados importados no Matlab, foi utilizada a ferramenta de “System Identification” e para cada conjunto de dados realizou-se o processo de identificação de função de transferência variando o número de polos e zeros. A figura 46 demonstra alguns dos testes realizados.

Figura 46 – Ferramenta de Identificação de Sistemas.



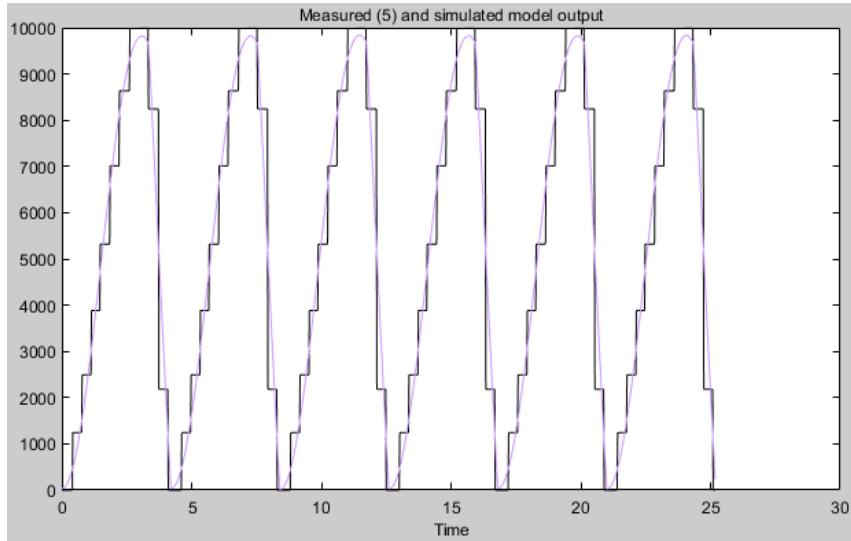
Fonte: Elaborado pelo autor.

Após várias estimativas, a que alcançou o maior valor de aproximação ao modelo foi utilizando o degrau com os valores de 5300 e 4700 e com a configuração de 2 zeros e 1 polo. A equação 4.1 foi a função de transferência obtida e que obteve uma aproximação de 78,33% de acordo com a ferramenta do Matlab.

$$G(s) = \frac{16,33s + 1,076}{s^2 + 0,01184s + 1,061} \quad (4.1)$$

A figura mostra a comparação do sistema estimado pela equação 4.1 e o sistema real a partir das amostras coletadas.

Figura 47 – Comparaçāo do modelo estimado com os dados reais.



Fonte: Elaborado pelo autor.

É possível observar que apesar do valor de aproximação de 78,33% não ter sido tão alto, visualmente a função estimada se assemelha com os dados reais. Comparando com os valores obtidos por (SANTOS, 2019), que obteve uma aproximação de 96.45% para uma função de um sistema do tipo integrador sem polos e zeros adicionais, é possível concluir que a diferença se deve pelo fato de que em seu trabalho, optou-se por realizar a identificação do sistema levando em consideração apenas os dados obtidos a partir da subida do degrau aplicado, não contemplando todo o processo. Desse modo, apesar do valor obtido ter sido mais baixo, avaliou-se que, por representar todo o processo, a função obtida foi satisfatória.

Ademais, analisando possíveis questões que poderiam aumentar o valor obtido, é possível perceber a partir do gráfico das amostras coletadas que a resposta do sistema na subida do degrau é diferente da descida, ou seja, seriam necessárias duas funções, uma para cada etapa, para representar o sistema mais fielmente. Por motivos de complexidade no momento da aplicação do PID necessário para atender as duas funções diferentes obtidas e do tempo limitado para a realização do projeto, optou-se por dar continuidade com apenas uma função mesmo que o valor da aproximação ter sido um pouco mais baixo.

Com a função de transferência obtida a partir da identificação do sistema, a próxima etapa foi a sintonia do sistema. Com o intuito de ponderar qual seria o método de sintonia mais adequado para o nosso sistema, foram feitas algumas análises a partir da função de transferência obtida. Com isso, identificou-se que os polos da equação são:  $s = -0,0592 \pm j 1,028$ , a parte real negativa indica que o sistema é estável e a parte imaginária significativa indica comportamento oscilatório no sistema. Além disso,

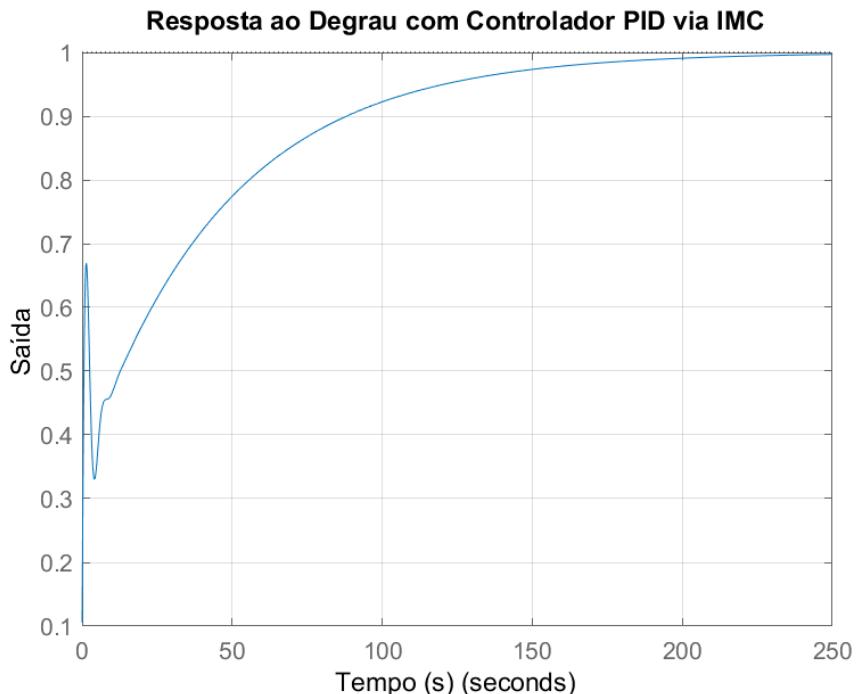
comparando com a forma padrão de segunda ordem, encontrou-se que  $\zeta \approx 0,0575$ , que é um valor muito baixo, resultado do sistema ser fortemente subamortecido.

Com essas informações, além do fato de buscar um sistema mais suave e robusto, optou-se por utilizar o Método do Modelo Interno. Sendo assim, para realizar os cálculos dos parâmetros do controlador e analisar a resposta ao degrau foi construído um script no Matlab.

Como as equações da tabela 7, não englobam as características da função de transferência obtida, foi necessário buscar alternativas, o que levou a teoria descrita no tópico 2.7.3.2. Os parâmetros do controlador PID serão definidos por meio da manipulação da constante  $\lambda$ , que é o fator de determinação de robustez e desempenho do controlador sendo que quanto maior, mais lento e mais robusto é o controlador.

Inciou-se por definir o  $\lambda = 1$  e os valores dos parâmetros do controlador PID obtidos foram:  $K_p = 0.065$ ,  $T_i = 2$  e  $T_d = 0.112$ . A figura a seguir demonstra o desempenho do controlador proposto.

Figura 48 – Resposta ao degrau do controlador com  $\lambda = 1$ .

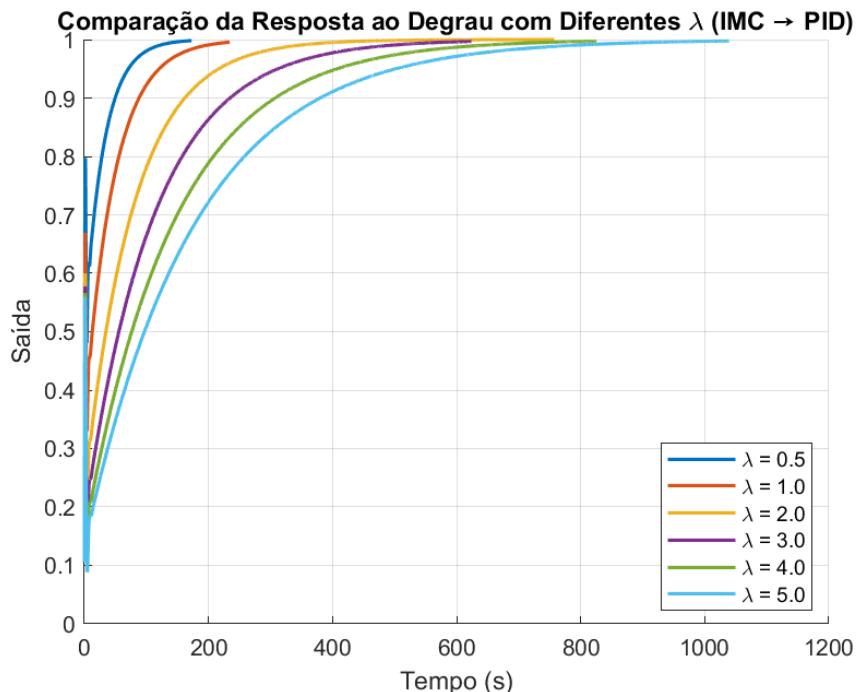


Fonte: Elaborado pelo autor.

Foram realizados ainda outros testes com diferentes valor de  $\lambda$  e uma vez que esse parâmetro está relacionado com a robustez do sistema, e quanto maior o valor, mais robusto e mais lento, foram feitas tentativas tanto com um valor menor que 1 quanto maiores para analisar o comportamento do sistema. A figura 49 mostra uma análise da resposta a degrau do sistema para diferentes valores de lambda: 0.5, 1, 2, 3, 4 e 5 e a seguir foram os respectivos parâmetros encontrados para o controlador PID:

- $\lambda = 0,5$ :  $Kp = 0.065$ ,  $Ti = 1$  e  $Td = 0.112$
- $\lambda = 2$ :  $Kp = 0.065$ ,  $Ti = 4$  e  $Td = 0.112$
- $\lambda = 3$ :  $Kp = 0.065$ ,  $Ti = 6$  e  $Td = 0.112$
- $\lambda = 4$ :  $Kp = 0.065$ ,  $Ti = 8$  e  $Td = 0.112$
- $\lambda = 5$ :  $Kp = 0.065$ ,  $Ti = 10$  e  $Td = 0.112$

Figura 49 – Resposta ao degrau do controlador com  $\lambda = 0,5; 1; 2; 3; 4; 5$ .



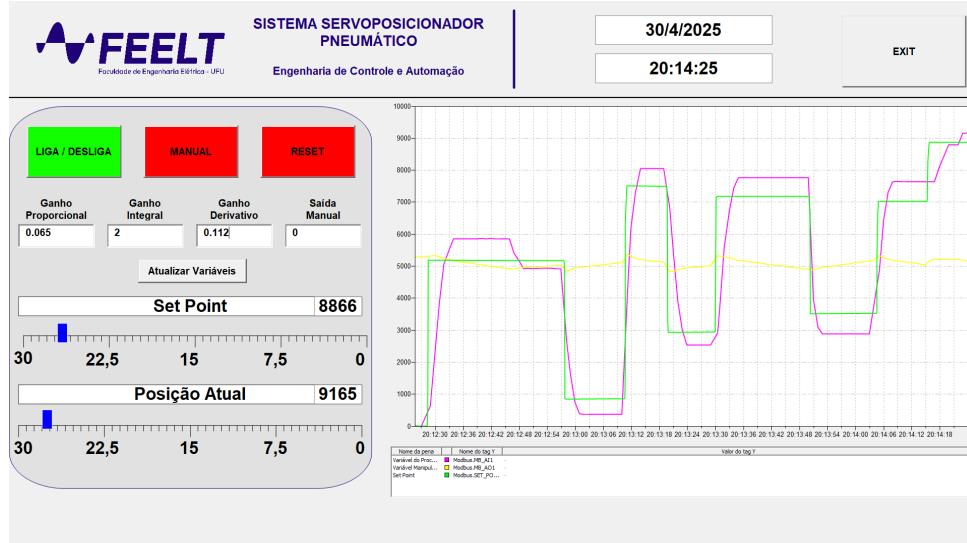
Fonte: Elaborado pelo autor.

## 4.5 Resultados da Sintonia no Controlador PID

Observando os valores obtidos através da identificação e sintonia do sistema, constata-se que os parâmetros definidos de maneira empírica se aproximaram consideravelmente. Como citado anteriormente, o controle do sistema anterior a identificação apresentava um funcionamento adequado, contudo, a presença de erro em regime permanente e a demora no tempo de resposta indicava que haveria espaço para melhoria, sendo assim, foram feitos testes com os novos parâmetros obtidos e comparou-se os resultados.

O gráfico da figura 50 representa o funcionamento do sistema utilizando os parâmetros obtidos da sintonia utilizando o  $\lambda = 1$ . Observa-se a alteração resultou em um tempo de resposta melhor, contudo, apresentou overshoot e um erro estacionário maior do que anteriormente.

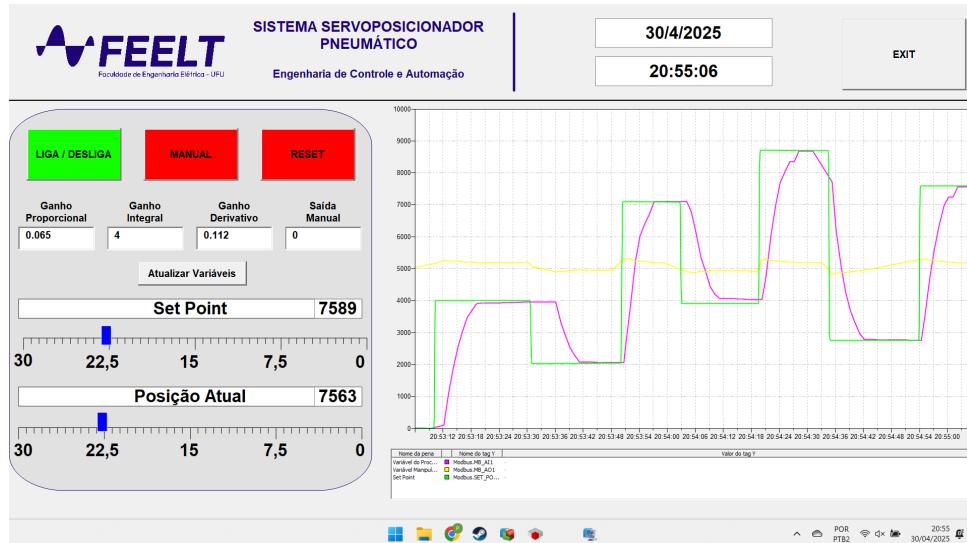
Figura 50 – Gráfico do controle realizado com parâmetro de sintonia onde  $\lambda = 1$ .



Fonte: Elaborado pelo autor.

Continuando os testes com os diferentes valores dos parâmetros obtidos com os diferentes  $\lambda$ , os valores que apresentaram o melhor resultado foram os obtidos utilizando  $\lambda = 2$ , como pode ser observado pelo gráfico da figura 51. A escolha se deve pelo fato de que para valor de lambda menores, o tempo de resposta foi bom, mas o erro em regime permanente foi considerável. Já para valores de  $Ti$  maiores, o tempo de resposta foi severamente impacto e o erro em regime permanente não diminuía suficiente para compensar.

Figura 51 – Gráfico do controle realizado com parâmetro de sintonia onde  $\lambda = 2$ .



Fonte: Elaborado pelo autor.

Dessa maneira, os valores escolhidos para os parâmetros PID foram:  $Kp = 0.065$ ,  $Ti = 4$  e  $Td = 0.112$ . O resultado foi um controlador com bom tempo de resposta e erro em regime estacionário pequeno o suficiente para os requisitos do projeto.

# 5 Conclusão

Este trabalho teve como objetivo realizar o controle e supervisão de uma planta pneumática, através da comunicação com CLP e seu monitoramento através de um sistema supervisório com o intuito de reabilitar a planta que não estava em condições de funcionamento e aplicação de sua função didática. Tendo em vista os resultados obtidos, considerou-se que os objetivos propostos foram alcançados.

Apesar de diferenças em relação ao que foi proposto inicialmente no aspecto da utilização de dois CLPs e a comunicação Modbus entre eles, foi possível realizar a aquisição de dados da planta e realizar o controle PID em malha fechada do servoposicionador. Ademais, concluiu-se que a alteração feita quanto à escolha do controlador e consequente eliminação da necessidade de se utilizar uma remota foi benéfica para o projeto, diminuindo dificuldades na comunicação e resultando em um menor tempo de resposta do processo.

Em relação a proposição de controle, devido a complexidade do sistema e a dificuldade em mensurar os todas as variáveis que influenciam no processo, além da restrição de tempo para o desenvolvimento do projeto, optou-se por não realizar uma modelagem matemática do processo. Contudo, os resultados obtidos utilizando o método da caixa preta com os dados coletados do funcionamento da planta para a identificação do sistema e posterior sintonia mostraram-se satisfatórios e resultaram em um controlador PID que contempla todo o processo possuindo robustez e tempo de resposta adequado.

Em comparação ao projeto de (SANTOS, 2019), o qual este trabalho tomou como ponto de partida, considerou-se que apesar de não suprir todos os pontos de melhoria indicados, os resultados obtidos apresentaram avanços no que diz respeito a função de transferência obtida ser mais condizente com o sistema como um todo, aproximando-se assim de uma representação mais completa do processo. Além disso, a eliminação da remota também foi considerada uma melhoria para o resultado final do trabalho.

Em um aspecto prático, concluiu-se que o importante objetivo de reabilitação da planta para cumprir sua função didática foi atingido. Tornando sua aplicação em aulas práticas possível e abrindo várias possibilidades de comprovação e aprofundamento das teorias aprendidas nas disciplinas dos cursos de Engenharia de Controle e Automação.

Vislumbrando a possibilidade de trabalhos futuros que buscarão outros caminhos ou melhorias a partir do que foi atingido no presente projeto, tem-se como sugestões: análise do sistema em duas etapas distintas, no avanço e no recuo, e consequentemente a obtenção de duas funções de transferência para uma representação mais precisa do processo. Por um outro caminho com foco maior na teoria, realizando o levantamento de todas as etapas e variáveis do processo e aplicação de conceitos matemáticos para obtenção do modelo.

# Referências Bibliográficas

- CAMPOS, M. C. M. M. de; TEIXEIRA, H. C. G. *Controles típicos de equipamentos e processos industriais*. [S.l.]: Editora Blucher, 2010. Citado 7 vezes nas páginas 30, 31, 33, 35, 36, 38 e 39.
- DOMINGUES, J. C. *Sintonia de controlador PID de sistema servopneumático de posicionamento*. Dissertação (B.S. thesis) — Universidade Tecnológica Federal do Paraná, 2019. Citado na página 14.
- ELIPSE. *Elipse E3*. 2025. Acesso em: 25 abr. 2025. Disponível em: <<https://www.elipse.com.br/produto/elipse-e3/>>. Citado 2 vezes nas páginas 23 e 24.
- Elipse Software. *KB-26658: Qualidade dos tags*. 2019. Acesso em: 26 abr. 2025. Disponível em: <<https://kb.elipse.com.br/kb26658-qualidade-dos-tags/>>. Citado na página 57.
- EURICH, A. M. *Validação experimental de um modelo matemático representativo de um sistema pneumático de posicionamento*. Dissertação (B.S. thesis) — Universidade Tecnológica Federal do Paraná, 2014. Citado 2 vezes nas páginas 14 e 17.
- FABIÁN, G. G. Desenvolvimento de bancada didática voltada ao estudo de sistemas pneumáticos. 2017. Citado na página 16.
- FESTO. *MLO-MME - Pneumatic actuators*. 2025. Acesso em: 25 abr. 2025. Disponível em: <[https://www.festo.com/cat/en-gb\\_gb/data/doc\\_ENGB/PDF/EN/MLO-MME\\_EN.PDF](https://www.festo.com/cat/en-gb_gb/data/doc_ENGB/PDF/EN/MLO-MME_EN.PDF)>. Citado na página 45.
- FESTO. *MPYE - Exhaust air flow control valves*. 2025. Acesso em: 25 abr. 2025. Disponível em: <[https://www.festo.com/cat/en-gb\\_gb/data/doc\\_ENUS/PDF/US\\_MPYE\\_ENUS.PDF](https://www.festo.com/cat/en-gb_gb/data/doc_ENUS/PDF/US_MPYE_ENUS.PDF)>. Citado 2 vezes nas páginas 44 e 45.
- FREITAS, C. M. Protocolo modbus: fundamentos e aplicações. *Eng. Elétrica. São Paulo*, p. 1–10, 2014. Citado 4 vezes nas páginas 26, 27, 28 e 29.
- JÚNIOR, R. F. F.; BRANDÃO, D. Identificação remota de plantas industriais utilizando tecnologias opc e cyberopc. 2009. Citado 5 vezes nas páginas 19, 31, 33, 34 e 37.
- MATLAB. *System Identification Toolbox*. 2025. Acesso em: 25 abr. 2025. Disponível em: <[https://www.mathworks.com/help/ident/index.html?s\\_tid=CRUX\\_lftnav](https://www.mathworks.com/help/ident/index.html?s_tid=CRUX_lftnav)>. Citado na página 35.
- MENDOZA, Y. E. A. et al. Desenvolvimento de um sistema servopneumático para regulação de velocidade de turbinas em pequenas centrais hidroelétricas. Florianópolis, SC, 2006. Citado 2 vezes nas páginas 14 e 18.
- NETWORKS, H. *Industrial Network Market Shares 2023*. 2023. Acesso em: 25 abr. 2025. Disponível em: <<https://www.hms-networks.com/news/news-details/21-05-2023-industrial-network-market-shares-2023>>. Citado na página 25.

- NISE, N. S. *Engenharia de Sistemas de Controle*. 8. ed. Rio de Janeiro: LTC, 2023. 108 p. E-book. Available at: <<https://integrada.minhabiblioteca.com.br/reader/books/9788521638285>>. Accessed on: May 2, 2025. ISBN 9788521638285. Citado na página 29.
- PEREIRA, U. V. *Uma nova fórmula de ajuste de controladores PID para funções de segunda ordem com tempo morto*. Dissertação (Trabalho de Conclusão de Curso de Graduação em Engenharia de Controle e Automação) — Universidade Federal do Rio Grande do Sul (UFRGS), 2011. Disponível em: <<https://lume.ufrgs.br/handle/10183/39903>>. Citado 2 vezes nas páginas 41 e 42.
- ROGGIA, L.; FUENTES, R. C. Automação industrial. *Santa Maria: E-tec Brasil*, p. 18, 2016. Citado 2 vezes nas páginas 19 e 21.
- SANTOS, A. R. Controle de um sistema servoposicionador pneumático utilizando controladores lógico programáveis e rede industrial modbus. Universidade Federal de Uberlândia, 2019. Citado 8 vezes nas páginas 14, 16, 33, 35, 43, 46, 60 e 64.
- SEN, S. K. *Fieldbus and networking in process automation*. [S.l.]: CRC Press, 2021. Citado 7 vezes nas páginas 18, 19, 23, 24, 25, 26 e 27.
- SILVA, E. A. D. *Introdução às linguagens de programação para CLP*. [S.l.]: Editora Blucher, 2021. Citado 4 vezes nas páginas 20, 21, 22 e 23.
- SKOGESTAD, S. Simple analytic rules for model reduction and pid controller tuning. *Journal of Process Control*, Elsevier, v. 13, n. 4, p. 291–309, 2003. Citado 2 vezes nas páginas 41 e 42.
- TAFFAREL, M. G. Projeto de uma linha de distribuição de ar-comprimido: estudo de caso do laboratório de ensino no curso de engenharia de produção. Universidade Federal do Pampa, 2015. Citado na página 17.
- WAGO. *WAGO Device Manager: Configuration and Parameterization of WAGO Products - Compact Controller 100*. [S.l.], 2024. Acesso em: 25 abr. 2025. Disponível em: <<https://www.wago.com/global/plcs-%E2%80%93-controllers/compact-controller-100/p/751-9301>>. Citado na página 47.
- WAGO. *Compact Controller 100 (751-9301)*. 2025. Acesso em: 25 abr. 2025. Disponível em: <<https://www.wago.com/global/plcs-%E2%80%93-controllers/compact-controller-100/p/751-9301>>. Citado na página 20.