



UNIVERSIDADE FEDERAL DE UBERLÂNDIA  
FACULDADE DE ENGENHARIA ELÉTRICA

**DESENVOLVIMENTO DE FERRAMENTA  
COMPUTACIONAL EMBARCADA PARA SIMULAÇÃO  
DE PROCESSOS INDUSTRIAIS**

**ALINE MARA OLIVEIRA**

**2019**

Ficha Catalográfica Online do Sistema de Bibliotecas da UFU  
com dados informados pelo(a) próprio(a) autor(a).

O48  
2019

Oliveira, Aline Mara, 1986-  
Desenvolvimento de ferramenta computacional  
embarcada para simulação de processos industriais [recurso  
eletrônico] / Aline Mara Oliveira. - 2019.

Orientador: Aniel Silva de Moraes.  
Coorientador: Márcio José da Cunha.  
Dissertação (Mestrado) - Universidade Federal de  
Uberlândia, Pós-graduação em Engenharia Elétrica.  
Modo de acesso: Internet.  
Disponível em: [http://dx.doi.org/10.14393/ufu.di.  
2019.2041](http://dx.doi.org/10.14393/ufu.di.2019.2041)  
Inclui bibliografia.

1. Engenharia elétrica. I. Silva de Moraes, Aniel , 1979-,  
(Orient.). II. José da Cunha, Márcio, 1978-, (Coorient.). III.  
Universidade Federal de Uberlândia. Pós-graduação em  
Engenharia Elétrica. IV. Título.

CDU: 621.3

Bibliotecários responsáveis pela estrutura de acordo com o AACR2:  
Gizele Cristine Nunes do Couto - CRB6/2091  
Nelson Marcos Ferreira - CRB6/3074

**ALINE MARA OLIVEIRA**

**DESENVOLVIMENTO DE FERRAMENTA COMPUTACIONAL  
EMBARCADA PARA SIMULAÇÃO DE PROCESSOS INDUSTRIAIS**

**Dissertação** apresentada ao Programa de Pós-graduação em Engenharia Elétrica da Universidade Federal de Uberlândia, como parte dos requisitos para a obtenção do título de **MESTRE EM CIÊNCIAS**.

**Banca Examinadora:**

Aniel Silva de Moraes, Dr. (Orientador) - UFU  
Márcio José da Cunha, Dr. (Co Orientador) - UFU  
Renato Ferreira Fernandes Junior, Dr. – UFU  
Henrique José Avelar, Dr. - CEFET

---

Aniel Silva de Moraes, Dr.  
Orientador

---

José Roberto Camacho, Dr.  
Coordenador do Curso de Pós-graduação

**UBERLÂNDIA – MG**

**2019**

*Dedico este trabalho ao meu marido Alex, à minha mãe Sônia, à minha filha Ângela e toda a minha família. Sou grata pelo apoio, carinho, pela companhia constante e compreensão recebida por todos eles, os quais foram fundamentais para o desenvolvimento desta dissertação.*

## AGRADECIMENTOS

Agradeço a **Deus** por permitir chegar até aqui.

Agradeço aos professores **Márcio José da Cunha e Aniel Silva de Moraes** pela orientação e disposição em auxiliar, apoio fundamental para a conclusão desta dissertação.

Estendo os meus agradecimentos aos professores **Renato Ferreira Fernandes Júnior e Luís Claudio de Oliveira Lopes** pelo incentivo e disponibilidade.

Agradeço à minha família, em especial minha mãe **Sônia** pelo apoio, motivação e compressão da minha ausência. Agradeço ao meu marido **Alex** pelo apoio, carinho e companhia constante que me incentivou a buscar sempre fazer o melhor.

Agradeço também à **Pós-graduação em Engenharia Elétrica** da Universidade Federal de Uberlândia pela oportunidade, em especial as secretárias **Patrícia e Cinara** pela atenção e prontidão nos procedimentos juntos à secretaria do programa e à **CAPES** (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) pelo apoio financeiro.

“Sonho é uma simulação do futuro. Quando pressionado presto atenção nele e tenho boas respostas”  
[Sidarta Ribeiro]

## RESUMO

O uso de simulação teve nos últimos anos uma demanda crescente nas aplicações industriais. A globalização trouxe um aumento da competitividade e fez com que as empresas se preocupassem mais em melhorar a qualidade de seus produtos, reduzir tempo de implementação dos projetos, otimizar recursos, dentre outros. Nesse contexto, existe a demanda de ferramentas que auxiliem na melhoria dos processos e possibilitem simular alternativas de controle de forma simples e confiável.

Contudo, para as ferramentas disponíveis no mercado, há uma grande dificuldade em mensurar os impactos causados por problemas intrínsecos ao meio de transmissão, como atrasos de comunicação variantes no tempo, atualização lenta e não periódica, dentre outros tipos de interferências, que causam grande dificuldade para o funcionamento ótimo de controladores industriais em controle de processos.

Também deve-se destacar que a grande maioria dos laboratórios das Universidades não dispões de grande diversidade de dispositivos de campo e plantas industriais para o desenvolvimento de aulas práticas. Dessa forma, a aplicação de métodos computacionais para uso em simuladores dinâmicos proporciona ao estudante uma experiência mais próxima com equipamentos industriais e permite ao aluno fazer qualquer tipo de experimento sem restrições de segurança.

Assim, o que se pretende neste trabalho é desenvolver um bloco funcional embarcado independente de tecnologia, que permita simular processos industriais de diferentes características considerando os atrasos de rede de comunicação envolvidos. Com o intuito de validar esse objetivo, foram apresentados resultados experimentais com uso do protocolo Modbus embarcado e uma plataforma de prototipação.

*Palavras – Chave: Controle PID, Modbus, Processos industriais, Simulação, Sistemas embarcados.*

## ABSTRACT

The use of simulation has in recent years a growing demand in industrial applications. Globalization has brought increased competitiveness and has made companies more concerned about improving the quality of their products, reducing project implementation time, optimizing resources, among others. In this context, there is a demand for tools that help in the improvement of processes and make it possible to simulate control alternatives in a simple and reliable way.

However, for the tools available in the market, there is a great difficulty in measuring the impacts caused by problems intrinsic to the transmission medium such as time-varying communication delays, slow and non-periodic updating, among other types of interferences that cause great difficulty to the performance of industrial controllers in process control.

We should also observe that most university laboratories do not have a large diversity of field devices and industrial plants to use in practical classes. Thus, the application of computational methods in dynamic simulators provides to student a closer experience with industrial equipment and allows student to do any type of experiment without security restrictions.

This way, what was intended in this work is to develop an embedded functional block independent of technology, which allows simulating industrial processes of different characteristics considering the network communication delays involved. In order to validate this objective, we presented experimental results with the embedded Modbus protocol and a prototyping platform.

*Keywords: PID Control, Modbus, Industrial process, Simulation, Embedded systems.*

---

## LISTA DE FIGURAS

|                      |  |    |
|----------------------|--|----|
| <b>Figura 1.1</b> -  | Sistema de controle via rede .....   | 1  |
| <b>Figura 2.1</b> -  | Relação entre entrada e saída de um sistema.....   | 6  |
| <b>Figura 2.2</b> -  | Ilustração das especificações de regime transitório.....   | 9  |
| <b>Figura 2.3</b> -  | Resposta ao degrau de um sistema de primeira ordem com atraso de transporte.....   | 11 |
| <b>Figura 2.4</b> -  | Comparação das respostas ao degrau de sistemas de primeira ordem com tempo morto com aproximações de Padé de primeira e segunda ordem..... | 14 |
| <b>Figura 2.5</b> -  | Etapas do processo de identificação.....   | 15 |
| <b>Figura 2.6</b> -  | Componentes de um sistema digital em uma malha de controle.....  | 18 |
| <b>Figura 2.7</b> -  | Subsistemas em um sistema de controle em malha fechada.....  | 26 |
| <b>Figura 2.8</b> -  | Oscilação sustentada para aplicação da sintonia pelo método de Z&N ...   | 29 |
| <b>Figura 2.9</b> -  | Diagrama em blocos controle em cascata.....  | 30 |
| <b>Figura 2.10</b> - | Arquitetura geral de um NCS .....  | 24 |
| <b>Figura 2.11</b> - | Exemplos de dispositivos conectados em Modbus .....  | 33 |
| <b>Figura 2.12</b> - | Modelo de Mensagem Modbus.....   | 33 |
| <b>Figura 2.13</b> - | Cabeçalho MBAP Modbus .....  | 34 |
| <b>Figura 2.14</b> - | Comunicação entre Mestre e Escravo Modbus sem erros.....   | 34 |
| <b>Figura 2.15</b> - | Comunicação entre Mestre e Escravo Modbus sem erros .....  | 35 |
| <b>Figura 2.16</b> - | Modelo requisição-resposta para o Protocolo Modbus.....  | 36 |
| <b>Figura 3.1</b> -  | Arquitetura de rede do sistema de testes.....  | 43 |
| <b>Figura 3.2</b> -  | Construção de um sistema equivalente discreto .....  | 45 |
| <b>Figura 3.3</b> -  | Tela de configuração software Syscon da Smar .....   | 46 |
| <b>Figura 3.4</b> -  | Tela de configuração blocos Modbus Smar.....   | 47 |
| <b>Figura 3.5</b> -  | Estratégia de controle para processos SISO em rede Modbus.....   | 48 |
| <b>Figura 3.6</b> -  | Estratégia de controle para processos SISO cascata em rede Modbus .....  | 48 |
| <b>Figura 3.7</b> -  | Estratégia de controle para processos SISO no CLP.....   | 49 |
| <b>Figura 3.8</b> -  | Estratégia de controle cascata para processos SISO no CLP.....   | 49 |

|                     |   |    |
|---------------------|---|----|
| <b>Figura 4.1</b> - | Respostas dos modelos da tabela 3.2 em Malha aberta ..... | 53 |
| <b>Figura 4.2</b> - | Gráfico de resposta em malha fechada.....                 | 54 |
| <b>Figura 4.3</b> - | Gráfico de resposta ao degrau em malha fechada.....       | 55 |

## LISTA DE TABELAS

|  |    |
|--|----|
| <b>Tabela 2.1</b> - Regra de sintonia de Ziegler-Nichols baseados no ganho crítico $K_{cr}$ e no período crítico $P_{cr}$ .....            | 29 |
| <b>Tabela 4.1</b> - Representação dos modelos Simulados.....   | 50 |
| <b>Tabela 4.2</b> - Parâmetros para os controladores PID – Ajustes finais.....   | 54 |
| <b>Tabela 4.3</b> - Critérios de desempenho dos modelos simulados.....   | 56 |
| <b>Tabela 4.4</b> - Valores calculados de FIT (%) para modelos simulados em Malha aberta em comparação com o software Matlab.....          | 58 |
| <b>Tabela 4.5</b> - Valores calculados de FIT (%) para modelos simulados em Malha Fechada comparação entre controle com BBB e com CLP..... | 58 |

## LISTA DE QUADROS

|                   |   |    |
|-------------------|---|----|
| <b>Quadro 2.1</b> | Características das tabelas Modbus..... | 37 |
| <b>Quadro 2.2</b> | Endereços Modbus .....                  | 38 |

## LISTA DE SÍMBOLOS

|            |                                     |
|------------|-------------------------------------|
| $\tau$     | Constante de tempo do Sistema       |
| $K$        | Ganho do processo                   |
| $\omega_n$ | Frequência natural                  |
| $\zeta$    | Fator de amortecimento              |
| $\tau_d$   | Atraso puro de tempo                |
| $t_r$      | Tempo de subida                     |
| $t_p$      | Instante de pico                    |
| $M_p$      | Sobressinal ( <i>overshoot</i> )    |
| $t_s$      | Tempo de acomodação                 |
| $\theta$   | Atraso de transporte ou tempo morto |
| $T_s$      | Período de amostragem               |

## LISTA DE ABREVIATURAS

|      |   |
|------|---|
| BBB  | <i>BeagleBone Black</i>   |
| CLP  | Controlador Lógico Programável  |
| FIT  | Índice de percentagem da variação da saída                                    |
| FPE  | <i>Final Prediction Error</i> (Índice de preditor de erro final)              |
| FT   | Função de Transferência   |
| HIL  | <i>Hardware in the Loop</i> (Simulação de hardware em malha de controle)      |
| MBCF | MODBUS <i>Configuration</i> (bloco de configuração MODBUS)                    |
| MBCM | MODBUS <i>Control Master</i> (bloco mestre de controle MODBUS)                |
| MIMO | <i>Multiple-input, Multiple-output</i> (Múltiplas-entradas, múltiplas-saídas) |
| MPZ  | <i>Matched Pole-Zero</i> (polo-zero combinado)                                |
| MV   | <i>Manipulate Variable</i> (Variável manipulada)                              |
| NCS  | <i>Network Control System</i> (Sistema de controle via redes)                 |
| PID  | Proporcional Integral Derivativo  |
| PV   | <i>Process Variable</i> (Variável de processo)                                |
| SISO | <i>Single-input, Single-output</i> (Uma-entrada, Uma-saída)                   |
| ZN   | Ziegler & Nichols   |

## SUMÁRIO

|  |           |
|--|-----------|
| <b>1. INTRODUÇÃO.....</b>  | <b>1</b>  |
| <b>2. REVISÃO BIBLIOGRÁFICA.....</b>                                     | <b>5</b>  |
| <b>2.1. Modelagem Matemática de Sistemas Dinâmicos.....</b>              | <b>5</b>  |
| 2.1.1. <i>Representação do modelo em Função de Transferência.....</i>    | 6         |
| 2.1.2. <i>Sistema de Primeira Ordem.....</i>                             | 7         |
| 2.1.3. <i>Sistema de Segunda Ordem.....</i>                              | 8         |
| 2.1.4. <i>Funções de transferência com atraso de transporte.....</i>     | 10        |
| 2.1.5. <i>Aproximações de Padé.....</i>                                  | 11        |
| 2.1.6. <i>Identificação de sistemas.....</i>                             | 14        |
| 2.1.7. <i>Validação do Modelo.....</i>                                   | 17        |
| <b>2.2. Sistemas Discretos.....</b>                                      | <b>17</b> |
| 2.2.1. <i>A Transformada-z.....</i>                                      | 18        |
| 2.2.2. <i>Função de transferência discreta.....</i>                      | 19        |
| 2.2.3. <i>Projeto de Controladores Digitais Utilizando Emulação.....</i> | 20        |
| <b>2.3. Sistema de Controle.....</b>                                     | <b>24</b> |
| 2.3.1. <i>Ações Básicas de Controle.....</i>                             | 25        |
| 2.3.2. <i>Controle de processos em cascata.....</i>                      | 29        |
| 2.3.3. <i>Sistema de controle via rede de comunicação.....</i>           | 23        |
| <b>2.4. Protocolo de Comunicação Modbus TCP/IP.....</b>                  | <b>33</b> |
| 2.4.1. <i>Modelo das Mensagens.....</i>                                  | 35        |
| 2.4.2. <i>Modelos requisição-resposta.....</i>                           | 36        |
| 2.4.3. <i>Modelos de dados.....</i>                                      | 37        |

|   |           |
|---|-----------|
| 2.4.4. Regras de Endereçamento.....                 | 38        |
| 2.5. Estado da Arte.....                            | 39        |
| <b>3. DESCRIÇÃO DOS TESTES.....</b>                 | <b>43</b> |
| 3.1. Arquitetura e descrição do Sistema.....        | 43        |
| 3.2. Metodologia dos testes.....                    | 44        |
| <b>4. RESULTADOS.....</b>                           | <b>50</b> |
| 4.1. Simulações dos processos em malha aberta.....  | 50        |
| 4.2. Simulações dos processos em malha fechada..... | 53        |
| 4.3. Análises de desempenho.....                    | 56        |
| 4.4. Validação e análise final dos resultados.....  | 57        |
| <b>4. CONCLUSÕES.....</b>                           | <b>60</b> |
| <b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>              | <b>62</b> |

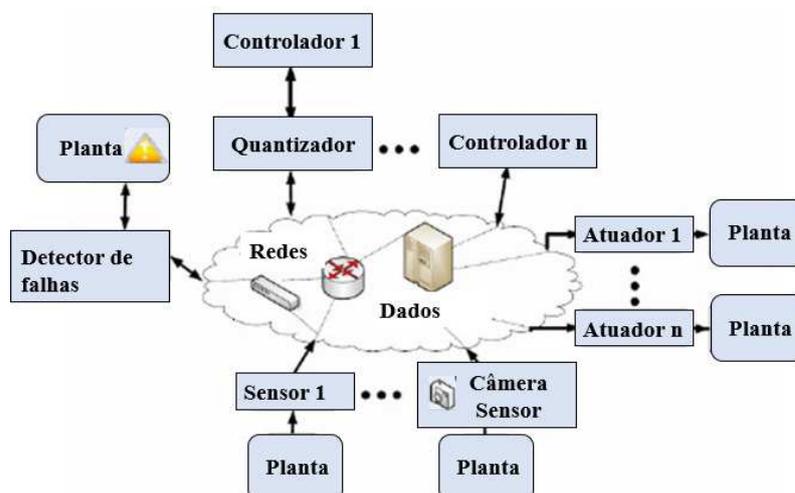
# CAPÍTULO 1

## Introdução

A evolução das redes de comunicação produziu muitas mudanças notáveis no controle tradicional em indústrias modernas. Sistemas de controle inteligentes baseados em campo, que combinam tecnologia de comunicação e controle descentralizado, estão executando um papel cada vez mais ativo na moderna automação industrial com funções e desempenho melhorados, mas com custo muito reduzido, oferecendo abordagens mais práticas e flexíveis. O conceito de redes industriais fornece muitos benefícios, dentre eles, maior flexibilidade e modularidade de instalações, facilidade de configuração do sistema, comissionamento e manutenção (GALLOWAY & HANCKE, 2013).

Aplicações recentes de sistemas de controle distribuído demonstram o surgimento de uma nova abordagem para a utilização de redes industriais. Nessa abordagem, o controlador e a planta ficam fisicamente separados e são conectados por uma rede de comunicação. Este tipo de implementação em sistemas onde as malhas de controle são fechadas sob uma rede de comunicação industrial tem sido denominado de Sistema de Controle via Redes ou *Network Control System* (NCS) (GUPTA & CHOW, 2010). A figura 1.1 exemplifica a aplicação típica de um sistema de controle via rede de comunicação.

Fig 1.1 Sistema de controle via rede típico



Fonte: Adaptado de Mahmoud (2016)

O uso de redes industriais digitais em malhas de controle trouxe para os processos industriais muitos benefícios, como redução de fiação e capacidades aprimoradas de comunicação, mas também gerou algumas consequências, como: atrasos, possível perda de dados e quantização de erros (CALDIÉRI, 2016). Portanto, muitos estudos se tornaram necessários para entender os efeitos das redes em sistemas de controle baseados em rede e suas limitações.

As redes usadas na integração de equipamentos em processos industriais podem ser classificadas quanto à sua disponibilidade ao usuário como redes “fechadas” ou “abertas”. As redes fechadas, também conhecidas como proprietárias, são constituídas de produtos, protocolos e interfaces de propriedade de determinados fabricantes, não sendo permitida a sua utilização sem a permissão ou aquisição de pacotes de hardware e software. Por outro lado, as redes abertas são de livre uso pois são de tecnologias baseadas em padrões de domínio público, dessa forma, podem ser utilizadas em quaisquer aplicações e independe de fabricante. Atualmente o uso de redes abertas e padronizadas ganham cada vez mais evidência devido à facilidade para adaptar e integrar sistemas (FERNANDES, 2009).

Existe uma grande diversidade de redes industriais, de uso livre ou não, disponíveis atualmente no mercado. Dentre os protocolos abertos, o Modbus, criado pela empresa Modicon em 1979, é um dos mais simples, abrangentes e de aplicação difundida podendo ser usado por uma grande variedade de produtos existentes em diversas empresas. (GUARESE et al., 2012). Devido a essas características, a rede de comunicação Modbus foi o protocolo selecionado para a realização dos testes deste trabalho.

No contexto dos sistemas de controle, existem diversos tipos de controladores, que variam em complexidade e performance. Os mais complexos utilizam-se de técnicas controle adaptativo, controle preditivo, controle ótimo, controle robusto, dentre outros. Entretanto, atualmente, mesmo com os desenvolvimentos tecnológicos aplicados aos processos industriais, ainda se observa que os controles Proporcional Integral Derivativo (PID) ainda são os controladores mais comuns e mais amplamente utilizados para tarefas regulatórias na maioria dos processos industriais (PADHAN & MAJHI, 2013). Isso ocorre, principalmente, porque esses controladores possuem estruturas de controle simples, preço acessível, fácil implementação e eficácia para sistemas lineares, além de uma vasta literatura acadêmica.

Apesar dessa versatilidade, é importante observar que a eficiência da aplicação de controladores PID está relacionada com a correta sintonia das malhas de controle, que será determinante na qualidade do produto final e na otimização dos recursos e custo de produção.

Visto que, na maioria dos processos industriais, a produção é de forma contínua e sua interrupção gera perdas financeiras consideráveis, há uma grande dificuldade para que os operadores façam ajustes de sintonia diretamente nos dispositivos locais das plantas. Devido à essa característica operacional, foi identificado que a maioria das malhas de controle em operação automática apresenta problemas de sintonia ou sintonia deficitária (FERNANDES, 2009). Visando solucionar esses problemas, foram criadas ferramentas de simulação que permitem os ajustes e análise operacionais de forma *offline* e sem a necessidade de descontinuação da produção.

## MOTIVAÇÃO

Um simulador de processos é uma ferramenta valiosa, principalmente quando se trata de validação de projetos, ampliação de plantas industriais e redução de custos. Devido a estes e outros fatores, o interesse em técnicas e pacotes computacionais para a modelagem, simulação e otimização de processos industriais tem aumentado muito nos últimos anos. Juntamente com esse interesse cresce também a qualidade das ferramentas disponíveis no mercado, mas em muitas situações estas não atendem grande parte das expectativas dos usuários e a uma grande variedade de aplicações (SOARES, 2003). Outro fator importante está relacionado com a obtenção de licenças, pois simuladores, principalmente os de processos e controladores avançados, são em geral de custo elevado.

Transpondo do ambiente industrial para o acadêmico, a simulação de processos industriais é de grande importância para o ensino e a capacitação de profissionais. Tem a capacidade de expor a realidade de um ambiente industrial, sem a necessidade de aquisição de grande quantidade de equipamentos e sem os riscos associados a situações críticas da planta real, mas podendo também simular tais situações críticas de forma segura, modificar parâmetros, visualizar e analisar variáveis do processo (FELIPE, 2014).

## OBJETIVO

O presente trabalho tem por objetivo desenvolver uma ferramenta computacional, embarcada em uma plataforma de prototipação, de tecnologia não proprietária e que permita simular processos industriais em redes distribuídas. A validação da ferramenta apresentada é feita por meio da comparação de testes experimentais com diferentes plataformas de

simulação, possibilitando avaliar a influência dos atrasos na transmissão de dados e tempo de processamento dos dispositivos utilizados. Para atingir os objetivos expostos, o presente trabalho apresenta uma revisão teórica nas áreas de teoria de controle, modelos matemáticos, identificação de sistemas e de protocolos de comunicação, a metodologia adotada para a realização dos testes e os resultados obtidos.

## LISTA DE PUBLICAÇÕES

Ao longo do desenvolvimento deste projeto foram publicados os seguintes trabalhos:

1. Oliveira, A. M.; Cunha, M. J.; Ricci, M. M.; Fernandes, R. F. Estratégias de controle em diferentes ambientes de simulação. **XII CEEL - Conferência de Estudos em Engenharia Elétrica**, Universidade Federal de Uberlândia, 2018.
2. Oliveira, A. M.; Cunha, M. J.; Fernandes, R. F.; Almeida, M. B. Development of embedded computational tool for simulation of industrial processes. **13th IEEE/IAS International Conference on Industry Applications (Induscon)**, São Paulo, 2018.  
<https://doi.org/10.1109/INDUSCON.2018.8627316>

# CAPÍTULO 2

---

## Revisão Bibliográfica

Este capítulo trata de alguns temas importantes para a compreensão deste trabalho e para a posterior discussão dos resultados, tais como: modelagem matemática, sistemas discretos, sistemas de controle e protocolos de comunicação.

### 2.1. Modelagem Matemática de Sistemas Dinâmicos

Modelagem matemática é a área do conhecimento que estuda técnicas para desenvolver e implementar modelos matemáticos de sistemas reais (AGUIRRE, 2007). A modelagem é um procedimento usado para conhecer a dinâmica dos processos e, a partir dessas informações, desenvolver projetos de controle apropriados para cada situação particular (COELHO & COELHO, 2004).

Um modelo matemático é análogo matemático de um sistema real e representa algumas das características observadas em tal sistema. Infelizmente não existe um modelo perfeito de um sistema real, mas sim uma família de modelos com desempenhos variados e com algumas características do processo estudado (AGUIRRE, 2007). Como o modelo é uma representação aproximada, a validade dos resultados obtidos depende totalmente da qualidade do modelo matemático utilizado e a da precisão requerida para a finalidade que ele se dispõe (COELHO & COELHO, 2004).

Os modelos matemáticos podem ser elaborados de duas formas, por meio de análise físico-matemática ou por análise experimental. Na análise físico-matemática, também conhecida como modelagem caixa branca, a obtenção dos modelos baseia-se nas leis da física que descrevem os fenômenos envolvidos no processo, como as leis de conservação de massa, energia e momento. A dificuldade de aplicar essa técnica está relacionada com a necessidade de um profundo conhecimento das características dos sistemas a serem modelados, o que na maioria das vezes é inviável devido ao grande esforço e tempo necessário para desenvolvimento do modelo. Com a técnica de análise experimental, também conhecida por modelagem caixa

preta, por outro lado, a determinação do modelo é baseada nas medidas ou observações do sistema, não exigindo conhecimento prévio da natureza do sistema (AGUIRRE, 2007).

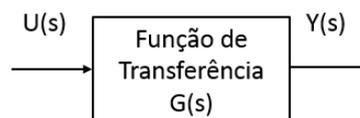
O modelo de um sistema é representado por uma equação matemática. Essa equação pode ser utilizada para diferentes propósitos: fazer previsões sobre o comportamento de um sistema sob determinadas condições operacionais sem a necessidade de realizar experimentos na planta real, analisar e projetar sistemas de controle por meio de simulações *offline*, avaliar características operacionais da planta para detectar erros e fazer diagnósticos, otimizar produção e minimizar custos operacionais por meio de informações que facilitam a tomada de decisões gerenciais (COELHO & COELHO, 2004).

A utilização do modelo para simular processos é um procedimento mais seguro, de baixo custo e pode ser usado para atender a diferentes propósitos. A representação de sistemas pode ser realizada basicamente de três maneiras: pela função de transferência, resposta impulsiva ou por meio de equações de estados (COELHO & COELHO, 2004). Nesse trabalho será aplicada a abordagem direcionada à representação por função de transferência, pois é a representação mais difundida para uso em controladores comerciais e permite executar os experimentos práticos de forma simples e direta.

### 2.1.1 Representação do modelo em Função de Transferência

Funções de transferência (FT) são funções que modelam o comportamento dinâmico de um par entrada-saída de um sistema (OGATA, 2010). Uma estimativa da função de transferência de um sistema linear e invariante no tempo pode ser obtida dividindo-se a transformada de Laplace da saída pela da entrada. Essa relação determina o quanto da variável de entrada será transferida à variável de saída (AGUIRRE, 2007). A Figura 2.1 ilustra a relação entre entrada e saída em um sistema genérico.

Figura 2.1 – Relação entre entrada e saída de um sistema.



Fonte: Adaptado de COELHO & COELHO (2004)

A Função de transferência  $G(s)$  é dada por:

$$G(s) = \frac{Y(s)}{U(s)} \quad (2.1)$$

Na equação 2.1,  $U(s)$  representa a transformada de Laplace da entrada  $u(t)$  e  $Y(s)$  a transformada de Laplace da saída  $y(t)$ . Dessa forma, a relação entre a entrada e saída é

normalmente representadas como a razão de dois polinômios em  $s$  (AGUIRRE, 2007), conforme apresentado na equação 2.2:

$$G(s) = \frac{B(s)}{A(s)} \quad (2.2)$$

Onde

$$A(s) = \sum_{i=0}^n a_i s^i \quad (2.3)$$

$$B(s) = \sum_{j=0}^m b_j s^j \quad (2.4)$$

Dois elementos fundamentais devem ser observados em uma função de transferência, pois caracterizam a resposta temporal e em frequência. Tais elementos são os zeros e pólos da função (AGUIRRE, 2007).

Os zeros de  $G(s)$  são os valores para os quais  $G(s)$  é nula, ou seja, são os zeros do polinômio  $A(s)$ . De forma semelhante, os polos de  $G(s)$  são os zeros do polinômio  $B(s)$ . A ordem da FT é obtida observando o número de pólos. Nessa representação, demonstrada pela equação 2.2, a ordem da função é igual a  $n$ . Para FT de sistemas estáveis, a função tem que ser própria, ou seja, tem-se que mandatoriamente  $n \geq m$  (AGUIRRE, 2007).

Os sinais podem ser representados utilizando-se variáveis contínuas ou discretas, por meio das transformadas de Laplace e transformada Z, respectivamente (OGATA, 2007). Vários processos industriais são bem representados por funções de transferência de primeira ou segunda ordem, nos tópicos a seguir serão apresentadas as estruturas básicas dessas duas configurações.

### 2.1.2 Sistema de Primeira Ordem

Os sistemas de primeira ordem são representados por modelos de equações diferenciais que apresentam somente a primeira derivada da sua saída, ou seja, equações diferenciais de primeira ordem (OGATA, 2007).

Se o sistema  $S$  for de primeira ordem, pode ser representado pela equação diferencial 2.5.

$$\tau \frac{dy(t)}{dt} + y(t) = Ku(t) \quad (2.5)$$

Na equação 2.5, o elemento  $K$  é o ganho do processo e  $\tau$  é a constante de tempo do sistema. Aplicando-se a transformada de Laplace, obtém-se a dinâmica do sistema representada por uma função de transferência de primeira ordem 2.6.

$$\frac{Y(s)}{U(s)} = \frac{K}{\tau s + 1} \quad (2.6)$$

De acordo com Aguirre (2007), muitos processos industriais podem ser descritos por FTs desse modelo. A resposta ao degrau pode ser obtida tomando-se a transformada inversa de Laplace de  $G(s)/s$ , uma vez que  $1/s$  é a transformada de Laplace do degrau unitário.

Dessa maneira, decompondo em frações parciais, podemos reescrever  $Y(s)$  como:

$$Y(s) = \frac{K}{s} - \frac{K\tau}{\tau s + 1} \quad (2.7)$$

Após aplicar a transformada inversa de Laplace em 2.7, obtemos:

$$y(t) = L^{-1}[Y(s)] \quad (2.8)$$

$$y(t) = K \left( 1 - e^{-\frac{t}{\tau}} \right), t \geq 0 \quad (2.9)$$

Na equação 2.9, se igualarmos  $t = \tau$ , podemos observar que  $\tau$  é o tempo a partir do início da perturbação no sistema na qual a resposta atinge o valor de 63,2% da variação total até o regime permanente. Analisando essa equação, nota-se que a constante de tempo  $\tau$  determina a velocidade de resposta do sistema, ou seja, quanto maior for a constante de tempo, mais lenta será a resposta dinâmica a uma perturbação na entrada (AGUIRRE, 2007).

Em alguns casos, a função de primeira ordem pode não apresentar a aproximação desejada com o processo real para determinadas aplicações. Nesses casos, pode-se partir para a representação da função de transferência baseada no modelo de segunda ordem, que pode trazer um pouco mais de complexidade ao modelo, mas pode ser mais satisfatória em processos que tem requisitos mais exigentes.

### 2.1.3 Sistema de Segunda Ordem

As funções de transferência de segunda ordem podem ser representadas pelo pela equação 2.10.

$$G(s) = K \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (2.10)$$

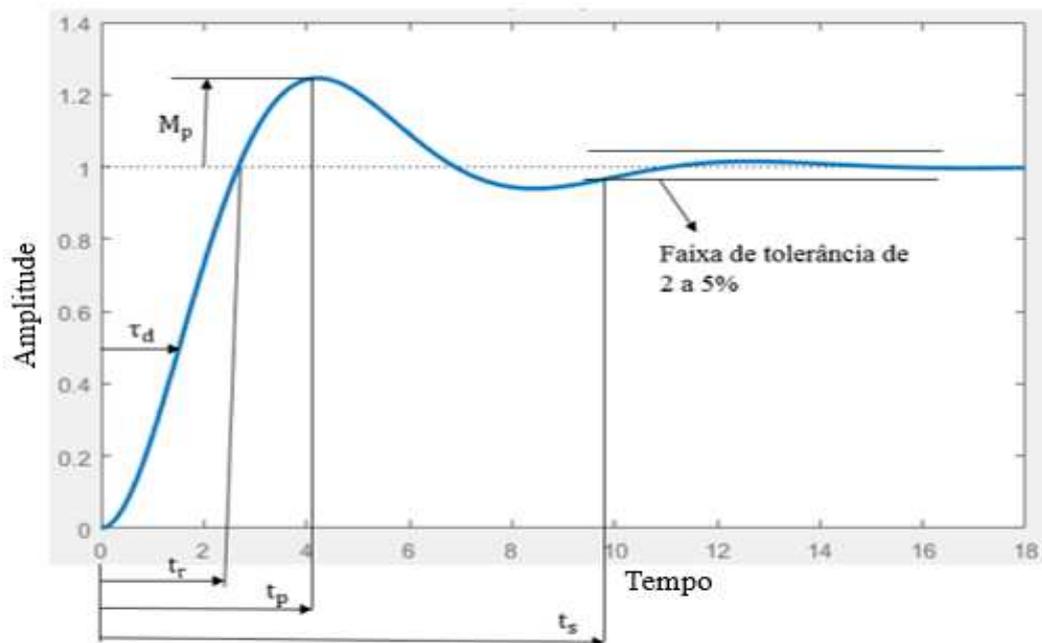
Onde a frequência natural  $\omega_n$  e o fator de amortecimento  $\zeta$  são os parâmetros que determinam as características dinâmicas do sistema. O fator de amortecimento  $\zeta$  é o termo responsável pelo grau de oscilação do sistema. Quando  $\zeta < 1$  o sistema será oscilatório e resultará em polos complexos e conjugados. Todavia, se  $\zeta > 1$  teremos um sistema de segunda ordem com

polos reais e estes podem ser analisados como dois sistemas de primeira ordem em série, utilizando-se o princípio da superposição. Dependendo do valor de  $\zeta$ , a resposta da saída do processo pode ser classificada como: subamortecida ( $0 < \zeta < 1$ ), criticamente amortecida ( $\zeta = 1$ ) ou sobreamortecida ( $\zeta > 1$ ) (AGUIRRE, 2007).

Algumas características importantes podem ser analisadas por meio da observação da do sinal de saída, após a aplicação de variação no sinal de entrada de um sistema. A curva de reação da resposta pode ser dividida em dois regimes, o regime transitório e o regime permanente.

A resposta transitória de um sistema de segunda ordem pode ser caracterizada pelas seguintes grandezas: tempo  $\tau_d$ , tempo de subida  $t_r$ , instante de pico  $t_p$ , sobre-sinal (overshoot)  $M_p$  e tempo de acomodação  $t_s$  (OGATA, 2007). A figura 2.2 apresenta as referidas grandezas.

Figura 2.2 – Ilustração das especificações de regime transitório.



Fonte: Adaptado de Ogata (2007)

Podemos definir estas especificações como:

- (i) Tempo  $\tau_d$ : é o tempo necessário para que a resposta alcance, pela primeira vez; a metade do valor final;
- (ii) Tempo de subida  $t_r$ : é o tempo necessário para que resposta passe de 10 % a 90 % ou de 0% a 100% de seu valor final. Para sistemas de segunda ordem sub amortecidos normalmente se usa o tempo de 0% a 100% e para sistemas sobre amortecidos normalmente de 10% a 90 %;

- (iii) Instante de pico  $t_p$ : é o tempo necessário para que a resposta alcance o primeiro pico;
- (iv) Sobre-sinal  $M_p$ : é o máximo valor de pico da curva de resposta subtraído do valor no estado estacionário e tudo dividido pelo valor no estado estacionário, sendo comum o seu uso em valor percentual;
- (v) Tempo de acomodação  $t_s$ : é o tempo necessário para que a curva de resposta alcance valores dentro de uma faixa de 2% a 5% e aí permaneça (OGATA, 2007).

É importante observar que todas estas especificações estão no domínio do tempo.

Apesar dos modelos de primeira e segunda ordem representarem grande parte dos processos industriais muitos sistemas dinâmicos reais apresentam um atraso puro de tempo. Esses atrasos muitas vezes são causados por atrasos no tempo de respostas de sensores e atuadores, atrasos de comunicação e ciclos de processamento dos controladores. Portanto, para a maioria das simulações de processos é importante considerar esses atrasos na elaboração do modelo sempre que o sistema original apresente tal característica.

#### 2.1.4 Funções de transferência com atraso de transporte

Se o sistema possuir atraso puro no tempo, também conhecido com tempo morto, o desenvolvimento é análogo aos modelos de primeira e segunda ordem, com a diferença que o tempo em que sistema começa a responder a perturbação será diferente de zero.

Considerando que o sistema apresente um atraso  $\theta$ , a nova saída do processo será representada por:

$$y_s(t) = y(t - \theta) \quad (2.11)$$

Aplicando-se a transformada de Laplace, temos:

$$Y_s(s) = L[y(t - \theta)] = \int_0^{\infty} y(t - \theta)e^{-st} dt \quad (2.12)$$

$$Y_s(s) = Y(s)e^{-\theta s} \quad (2.13)$$

Assim, obtém-se a função atraso como:

$$\frac{Y_s(s)}{Y(s)} = e^{-\theta s} \quad (2.14)$$

Substituindo  $Y(s)$  obtido anteriormente em 2.7, tem-se a função de transferência de primeira ordem com atraso:

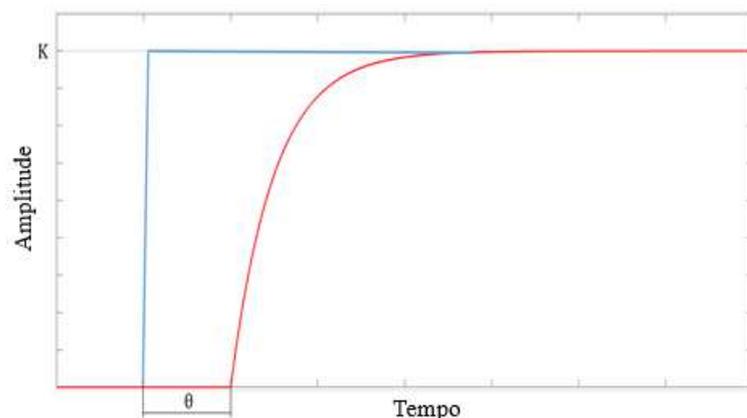
$$\frac{Y(s)}{U(s)} = \frac{K}{\tau s + 1} e^{-\theta s} \quad (2.15)$$

Onde  $s$  é o operador Laplace ( $s = d/dt$ ),  $K$  é o ganho,  $\theta$  é o tempo morto,  $\tau$  é a constante de tempo, e  $Y(s)$  e  $U(s)$  são as transformadas de Laplace de saída e da entrada, respectivamente.

Na equação 2.15, o atraso da função temporal de  $\theta$  unidades de tempo, no domínio de Laplace é representado multiplicando-se a transformada da função por  $e^{-\theta s}$ .

O efeito dessa multiplicação causado em uma função de transferência é o deslocamento da resposta em  $\theta$  unidades de tempo na função temporal. Esse atraso de transporte, ou tempo morto  $\theta$ , é o tempo a partir do início da aplicação de um sinal, degrau na entrada do sistema até que seja notada uma mudança na saída do mesmo. A figura 2.3 mostra a representação gráfica de um modelo com atraso de transporte.

Figura 2.3 – Resposta ao degrau de um sistema de primeira ordem com atraso de transporte



Fonte: Adaptado de Ogata (2007)

Uma dificuldade encontrada com a aplicação da representação dada pela equação 2.15 para o atraso puro de tempo é que  $e^{-\theta s}$  não pode ser expresso como uma razão de polinômios em  $s$  com grau finito. Assim, não é possível representar exatamente sistemas com atrasos puros de tempo por meio de funções de transferência. Uma forma de contornar esse problema e obter o modelo no formato mais conveniente, é representar  $e^{-\theta s}$  de forma aproximada como uma razão de polinômios em  $s$ , usando aproximações de Padé.

### 2.1.5 Aproximações de Padé

As técnicas de aproximação de Padé permitem obter um formato reduzido dos modelos de primeira ou segunda ordem com atraso de transporte em razão de polinômios em  $s$ .

No método clássico de Padé, que foi proposto por Padé em 1892, leva-se em consideração os pólos e zeros do sistema de ordem aumentada a fim de reduzi-los, o que é muito útil em diversas aplicações. O equacionamento que segue é baseado em Aguirre (2007).

Considerando a função de transferência de ordem aumentada no domínio de Laplace

$$G(s) = \frac{g_0 + g_1s + \dots + g_qs^q}{h_0 + h_1s + \dots + hs^q} \quad (2.16)$$

Define-se os coeficientes de Padé como sendo:

$$c_0 = \frac{g_0}{h_0} \quad (2.17)$$

$$c_k = \frac{g_k - \sum_{j=1}^k h_j c_{k-j}}{h_0}, k > 0 \quad (2.18)$$

Onde  $g_i = h_i = 0$  para  $i > q$  e  $c_i$  são os coeficientes da expansão de  $G(s)$  em séries de Taylor em torno do ponto  $s = 0$ , e são chamados de coeficientes de Padé.

Portanto, deseja-se obter um modelo de ordem reduzida que esteja no formato apresentado na equação 2.19.

$$R(s) = \frac{a_0 + a_1s + \dots + a_qs^q}{b_0 + b_1s + \dots + bs^q} \quad (2.19)$$

Entretanto, nem sempre o modelo reduzido é estável. Quando a redução da ordem resulta em um modelo instável, outras técnicas devem ser utilizadas. Uma alternativa é a utilização do Método Clássico de Padé com retenção de polos. O objetivo deste método é utilizar os polos dominantes e conhecidos do sistema de ordem aumentada para realizar a redução do modelo. Para se obter os valores dos demais polos e zeros do sistema, deve-se considerar a Equação 2.19 no sistema representado pela equação 2.20.

$$0 = b_0 + b_1s_z + \dots + s_z^m, z = 1, 2, \dots, n_{pr} \quad (2.20)$$

Sendo  $n_{pr}$  o número de pólos retidos do modelo de ordem aumentada.

Já na aproximação de Padé para tempo morto, a função de transferência de um processo com tempo de atraso inclui o fator matemático  $e^{-\theta s}$ , em que  $\theta$  é o tempo morto. Uma aproximação do tempo morto pode ser obtida escrevendo  $e^{-\theta s}$  como  $\frac{1}{e^{\theta s}}$  expressando o denominador como uma série de Taylor, o resultado é expresso pela equação 2.21.

$$e^{-\theta s} = \frac{1}{e^{\theta s}} = \frac{1}{1 + \theta s + \frac{\theta^2 s^2}{2!} + \frac{\theta^3 s^3}{3!} + \dots} \quad (2.21)$$

Mantendo somente os dois primeiros termos da série no denominador, que são os mais significativos quantitativamente, tem-se:

$$e^{-\theta s} \cong \frac{1}{1 + \theta s} \quad (2.22)$$

Essa aproximação, que é simplesmente um atraso de primeira ordem, é uma aproximação ruim de tempo morto, não fornecendo boa representação para a maioria dos processos. Uma maneira de melhorar esse comportamento pode ser desenvolvida expressando-se o tempo morto como:

$$e^{-\theta s} = \frac{e^{-\theta s/2}}{e^{\theta s/2}} \quad (2.23)$$

Expandindo o numerador e o denominador em uma série de Taylor e mantendo somente os termos de primeira ordem, resulta na equação 2.24.

$$e^{-\theta s} \cong \frac{1 - \frac{\theta}{2}s}{1 + \frac{\theta}{2}s} \quad (2.24)$$

Essa aproximação é também conhecida como aproximação de Padé de primeira ordem. Uma outra aproximação bastante conhecida para tempo morto é a aproximação de Padé de segunda ordem, representada pela equação 2.25.

$$e^{-\theta s} \cong \frac{\theta^2 s^2 - 6\theta s + 12}{\theta^2 s^2 + 6\theta s + 12} \quad (2.25)$$

Para melhor compreensão dos modelos resultantes da aproximação de Padé, um exemplo de aplicação é apresentado a seguir. O objetivo desse exemplo é mostrar as características das aproximações de Padé para diferentes ordens.

Considere a seguinte função de transferência:

$$G_p(s) = \frac{1}{s + 1} e^{-s} \quad (2.26)$$

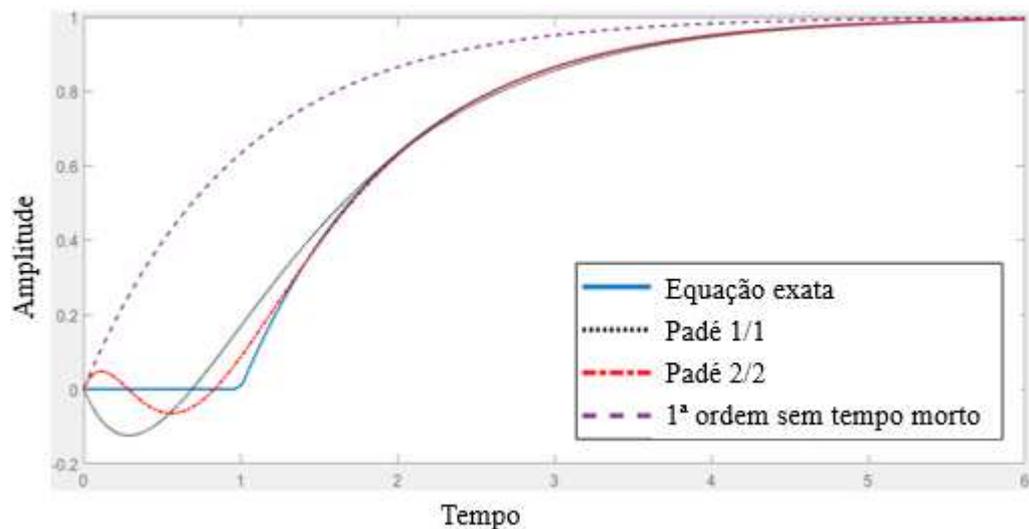
Aplicando-se as aproximações de Padé de 1ª e 2ª ordem, obtemos as equações 2.27 e 2.28 respectivamente:

$$G_{FTMA(1)} = \frac{-s + 2}{s^2 + 3s + 2} \quad (2.27)$$

$$G_{FTMA(2)} = \frac{s^2 - 6s + 12}{s^3 + 7s^2 + 18s + 12} \quad (2.28)$$

As respostas ao degrau unitário dessas três aproximações do tempo morto, juntamente com a função original, são mostradas na figura 2.4.

Figura 2.4 Comparação das respostas ao degrau de sistemas de primeira ordem com e sem tempo morto com aproximações de Padé de primeira e segunda ordem.



Embora nenhuma das aproximações do tempo morto  $e^{\theta s}$  seja muito precisa, é mais útil quando está multiplicado por várias funções transferência de primeira ordem ou segunda ordem. Nesse caso, a aproximação do tempo morto combinado com essas funções de transferência apresenta ser satisfatório, pois essas funções de transferência filtram o conteúdo de altas frequências dos sinais que atravessam o tempo morto.

Observa-se que com a aproximação de Padé 1/1, representada pela equação 2.27, o tempo morto é substituído por um polo à esquerda e um zero à direita, e que a resposta no tempo apresenta um *undershoot* no início (resposta inversa), ou seja, o sinal passa primeiro por uma pequena queda de valor antes de iniciar o movimento de subida.

Na aproximação de segunda ordem (Padé 2/2), representada pela equação 2.28, encontram-se dois polos à esquerda e dois zeros à direita representando o atraso. Ocorre uma pequena subida no início da resposta no tempo, seguida por uma queda e novamente subida.

Com a observação gráfica apresentada na figura 2.4, nota-se que cada aproximação de Padé de ordem maior corresponde a um cruzamento no eixo horizontal no início da resposta no tempo. Considerado uma aproximação de infinitos termos, tendendo ao valor real, ocorreria uma infinita sucessão de cruzamentos no eixo, resultando em um tempo inicial praticamente de resposta nula.

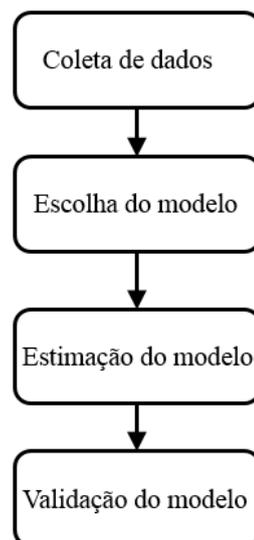
### 2.1.6 Identificação de sistemas

Os modelos matemáticos podem ser elaborados basicamente de duas formas, por meio de análise de características físicas do processo (caixa branca) ou por análise experimental

(caixa preta) e podem ser representadas por diferentes formatos de equações matemáticas. Visando simplificar o processo de modelagem, diferentes técnicas experimentais surgiram como alternativa à modelagem caixa branca. A área de modelagem que estuda essas técnicas é denominada Identificação de Sistemas. Uma das características principais dessa técnica é a não dependência de conhecimento prévio do sistema para a obtenção de modelos que possam representá-lo. O que se pretende descrever com tais modelos são as relações de causa e efeito entre as variáveis de entrada e saída (LJUNG, 1999).

A identificação do sistema tem como finalidade estudar técnicas alternativas de modelagem matemática e construir modelos matemáticos de sistemas dinâmicos baseados em dados observados do sistema (AGUIRRE, 2007). A figura 2.5 apresenta as etapas do processo de identificação de forma resumida.

Figura 2.5 – Etapas do processo de identificação.



Adaptado de Coelho & Coelho (2004)

De acordo com Aguirre (2007), a identificação de sistemas realizada de forma experimental consiste em:

- 1) Coletar os dados dos sinais de entrada  $u(k)$  e de saída real  $y(k)$  do sistema a ser identificado.
- 2) Escolha dos modelos matemáticos que melhor representam o processo, baseado nos dados coletados.
- 3) Fazer os ajustes necessários, baseados em critérios pré-estabelecidos e estimar o modelo.

4) Validar o modelo estimado, verificando se incorpora ou não as características do sistema original.

Caso não se obtenha um resultado esperado, é feita nova estimação até que o modelo seja validado.

Em relação à coleta e armazenamento de dados, a modelagem e identificação de sistemas possui duas classificações: identificação *online* e identificação *offline*. Na identificação *offline* a planta é excitada e os dados de entrada e saída do processo são coletados e armazenados em um banco de dados. Posteriormente, as medidas obtidas são processadas para estimar os parâmetros do modelo com algoritmos de estimação não recursivos. Nessa técnica de estimação são utilizados usualmente modelos discretos, pelo fato de os algoritmos de identificação trabalharem com sinais amostrados (AGUIRRE, 2007). Dentre as técnicas *offline* mais aplicadas estão a resposta em frequência, resposta temporal e identificação paramétrica.

Na identificação *online* o procedimento é iterativo e o cálculo dos parâmetros é feito simultaneamente ao ensaio de identificação. Aplica-se métodos recursivos, necessitando de pouca memória de armazenamento sendo capaz de atualizar a estimação dos parâmetros do modelo a cada período de amostragem. O método online mais popular é a identificação paramétrica usando mínimos quadrados recursivo. A aplicação dessa técnica é muito interessante em processos que tenham parâmetros variantes no tempo e na aplicação de controle preditivo e redes neurais (COELHO & COELHO, 2004).

A escolha do modelo é um processo que envolve conflitantes objetivos, principalmente relacionados à complexidade do modelo. A definição de um bom modelo é de caráter subjetivo e envolve muitas vezes procedimentos de tentativa e erro. O novo modelo só é aceito se todas as restrições do processo forem atendidas. No caso de o modelo não ser aceito, todos os procedimentos de identificação realizados anteriormente devem ser reavaliados até que um modelo apropriado seja encontrado (COELHO & COELHO, 2004).

O processo de estimação de parâmetros determina os valores dos parâmetros do modelo e pode ser formulado com um problema de otimização. O melhor modelo é selecionado baseado em critérios de desempenho preestabelecidos. Uma vez obtidos os parâmetros, as técnicas de validação de modelos são aplicadas (AGUIRRE, 2007).

Na etapa de validação do modelo pode-se utilizar a comparação entre as respostas obtidas pelo modelo estimado com o sistema real. Se o erro cometido no ajuste está em valores preestabelecidos e a resposta do modelo estimado reflete corretamente a resposta do sistema, o

modelo é considerado adequado (COELHO & COELHO, 2004). No tópico seguinte serão apresentados os métodos de validação com maiores detalhes.

### 2.1.7 Validação do Modelo

A validação do modelo é um procedimento para avaliar se o modelo obtido é suficientemente adequado para a finalidade a que se propõe. Normalmente é executada utilizando uma amostra de dados do processo real diferente dos coletados para o processo de identificação, e, então, comparando como o modelo se comporta em relação a esses dados e avaliando os erros de predição encontrados nesse processo (AGUIRRE, 2007). O critério de validação muitas vezes tem caráter subjetivo e dependem especialmente da aplicação que será dada ao modelo.

Dentre os parâmetros de validação usuais estão os critérios de informação de Akaike (AIC), *Final Prediction Error* (FPE) e percentagem da variação da saída (FIT). Os critérios de Akaike são utilizados para determinação da qualidade do modelo, através de simulações com diferentes conjuntos de dados e diferentes modelos. De acordo com a teoria de Akaike, o melhor modelo é aquele que tem menor FPE (FERNANDES, 2009).

No método de validação por percentagem da variação da saída (FIT), a qualidade do modelo é representada em variação de 0 a 100% e o melhor modelo é o que apresenta maior valor de FIT (Fernandes, 2009). A equação 2.29 apresenta como é calculado o FIT.

$$FIT = \left( 1 - \frac{\sum_{k=1}^N Norm(y(k) - \hat{y}(k))}{\sum_{k=1}^N Norm(y(k) - \bar{y}(k))} \right) 100 \quad (2.29)$$

$$Norm(V) = \sum_{k=1}^N \sqrt{ABS(V)^2} \quad (2.30)$$

Onde,  $y(k)$  é a saída real do processo no instante  $k$ ,  $\hat{y}(k)$  é a saída estimada, obtida pela aplicação dos parâmetros obtidos, e  $\bar{y}(k)$  é a média das  $N$  amostras de  $y(k)$  ao longo da identificação.

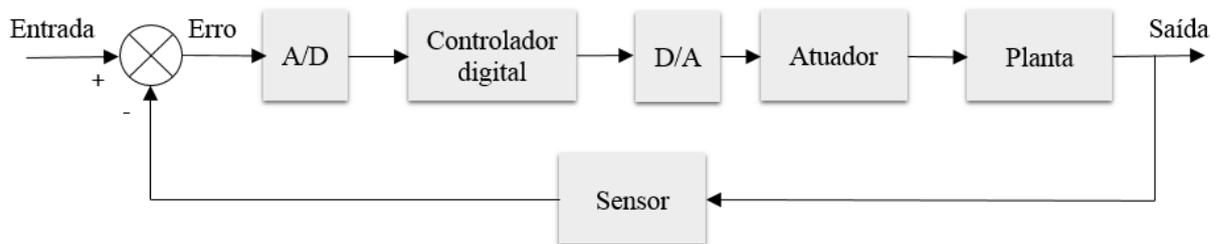
## 2.2. Sistemas Discretos

Apesar de a grande maioria dos processos industriais serem contínuos, boa parte do controle aplicado atualmente é feito por meio de controladores digitais. Uma vez que são utilizadas técnicas baseadas em modelagem matemática, se faz necessário encontrar modelos discretos que atendam às aplicações de projeto e possam ser usadas pelos sistemas para calcular

a saída do sinal de controle (COELHO & COELHO, 2004). Uma das características dos sistemas discretos é que o sinal de controle é amostrado com uma taxa  $T_s$ , denominada período de amostragem.

O diagrama da figura 2.6 ilustra os componentes de uma malha de controle de um sistema digital e como estes estão relacionados dentro do processo.

Figura 2.6 - Componentes de um sistema digital em uma malha de controle.



Fonte: Adaptado de Ogata (2010)

No início do processo, um valor de entrada é estabelecido. Em seguida, esse sinal é recebido por um conversor Analógico/Digital que é responsável por fazer a amostragem e quantização, de forma que possa ser processado pelo controlador digital. No controlador o sinal é recebido, processado e transformado de acordo com a programação de controle preestabelecida. O sinal sai do controlador e passa novamente por um processo de conversão, nesse momento de digital para analógico. O atuador da planta recebe esse sinal e gera a saída do processo. Essa saída é percebida por um sensor e o sinal de erro é alterado por meio do comparador, fechando a malha de controle do processo.

Devido à grande maioria dos processos industriais serem compostos por essa estrutura, faz-se necessário a aplicação de métodos de discretização tanto nos processos reais quanto nos simulados.

Para calcular a função de transferência discreta a partir de funções contínuas de primeira e segunda ordem, podem ser aplicadas diversas técnicas. Nos tópicos seguintes serão apresentados como são estruturados os modelos discretos e como são modelados os projetos de controladores digitais utilizando emulação.

### 2.2.1 A Transformada-z

A transformada-z é uma ferramenta matemática muito útil na transformação de sistemas contínuos lineares para sistemas discretos. Assim como a Transformada de Laplace transforma uma função contínua no tempo  $t$  para o domínio da frequência complexa,  $s$ , a Transformada-z, transforma uma função discreta no tempo para o domínio da frequência complexa.

A transformada-z de um sinal  $f[n]$  é definida como:

$$F(z) = \mathcal{Z}\{f[k]\} = \sum_{k=0}^{\infty} f[k]z^{-k} \quad (2.31)$$

Onde  $f[k]$  é o sinal amostrado de  $f(t)$  e  $k=1,2,3,\dots$  se refere às amostras discretas coletadas nos tempos  $t_1, t_2, t_3,\dots$ . Da mesma forma como a função de transferência de um sistema contínuo é a transformada de Laplace do sinal de saída sobre o sinal entrada, a função de transferência de um sistema discreto é a transformada-z do sinal de saída sobre o sinal de entrada (OGATA, 2010).

Essa relação permite que, dada uma equação de diferenças do sistema, seja encontrada facilmente a função de transferência do sistema discreto.

### 2.2.2 Função de transferência discreta

A função de transferência discreta é a relação entre a transformada z da saída,  $Y(z)$ , pela transformada z da entrada,  $U(z)$ , isto é,

$$G(z) = \frac{Y(z)}{U(z)} \quad (2.32)$$

A relação  $G(z)$  é uma razão de dois polinômios em  $z$  e está representada por

$$G(z) = \frac{B^d(z)}{A^d(z)} \quad (2.33)$$

Onde

$$B^d(z) = \sum_{j=0}^m b_j^d z^j \quad (2.34)$$

$$A^d(z) = \sum_{i=0}^n a_i^d z^i \quad (2.35)$$

Sendo  $n \geq m$ ,  $a_n^d = 1$  e  $n$  é a ordem do sistema. Os elementos  $\{b_j^d, a_i^d, n\}$  com  $j \in [0, m]$  e  $i \in [0, n]$  são desconhecidos e devem ser determinados através de uma modelagem matemática do sistema ou através da identificação. Algumas vezes assume-se que o valor de  $n$  é conhecido a priori.

Na avaliação de modelos matemáticos discretos, deve-se selecionar um período de amostragem,  $T_s$ , para cada aplicação particular, de acordo com uma das seguintes relações apresentadas em 2.36.

$$\frac{T_{95}}{T_s} = 5 \dots 15 ; T_s = \tau/10 \quad (2.36)$$

Onde  $T_{95}$  é o tempo que a resposta do sistema leva para alcançar 95% do valor final e  $\tau$  é a constante de tempo dominante do sistema.

O modelo de entrada/saída para o processo, utilizando a transformada  $z$ , pode ser obtido como:

$$Y(z) = G(z)U(z) \quad (2.37)$$

Onde  $G(z)$  é a função de transferência do processo, incluindo o segurador de ordem zero, e pode ser representada por uma razão de dois polinômios em  $z^{-1}$ , isto é,

$$G(z) = z^{-d} \frac{B(z^{-1})}{A(z^{-1})} = \frac{Y(z)}{U(z)} \quad (2.38)$$

ou

$$A(z^{-1})Y(z) = z^{-d}B(z^{-1})U(z) \quad (2.39)$$

A equação 2.39 é uma representação usual na literatura de identificação e controle digital. Os polinômios  $A(z^{-1})$  e  $B(z^{-1})$ , relacionados aos pólos e zeros de malha aberta do processo, respectivamente, são da forma:

$$A(z^{-1}) = 1 + a_1z^{-1} + a_2z^{-2} + \dots + a_{na}z^{-na} \quad (2.40)$$

$$B(z^{-1}) = b_0 + b_1z^{-1} + b_2z^{-2} + \dots + b_{nb}z^{-nb} \quad (2.41)$$

A correspondente equação a diferenças da planta, considerando a transformação exposta na equação 2.31, é dada por:

$$y(k) = -a_1y(k-1) - a_2y(k-2) + b_0u(k) + b_1u(k-1) + b_2u(k-2) \quad (2.42)$$

Onde  $y(k)$  e  $u(k)$  são os sinais de saída e de controle, respectivamente, disponíveis nos instantes de amostragem  $t = nT_s$ , onde  $n \in Z^+$ .

### 2.2.3 Projeto de Controladores Digitais utilizando Emulação

Projeto por emulação consiste em projetar um controlador contínuo, no domínio de  $s$ , utilizando as técnicas clássicas de controle e, na sequência, utilizando o mapeamento de polos e zeros do plano  $s$  para o plano  $z$ , pela transformação  $z = e^{sT}$ , encontra-se o controlador discreto equivalente ao contínuo. A grande vantagem desse método é que as técnicas de controle utilizadas já são conhecidas, ou seja, não é necessário se fazer um estudo aprofundado sobre projeto de controladores digitais. Por outro lado, a desvantagem é que esse método ignora totalmente o fato de que os conversores A/D, D/A e o computador digital serão utilizados. Dessa forma, não considera os erros de quantização típicos da conversão de sinais analógicos para digitais.

Isso impõe a necessidade de que o período de amostragem seja o menor possível, fazendo assim com que o controlador fique muito semelhante ao controlador analógico (PHILLIPS & NAGLE, 1994).

Após a determinação do controlador digital, é necessário colocá-lo de uma forma que um computador digital possa interpretá-lo. Essa forma, em geral, é apresentada como a equação de diferenças do controlador.

Seja a função de transferência do controlador  $G_c(z)$  representada pela equação 2.43.

$$G_c(z) = \frac{b_m z^m + b_{m-1} z^{m-1} + \dots + b_1 + b_0}{z^n + a_{n-1} z^{n-1} + \dots + a_1 z + a_0} \quad (2.43)$$

Definindo o sinal de entrada de  $G_c(z^{-1})$  como  $E_c(z^{-1})$ , e o sinal de saída como  $U_c(z^{-1})$ , tem-se que:

$$G_c(z^{-1}) = \frac{U(z^{-1})}{E(z^{-1})} \quad (2.44)$$

Logo,

$$\begin{aligned} & U(z^{-1})(1 + a_{n-1}z^{-1} + \dots + a_1z^{1-n} + a_0z^{-n}) \\ &= E(z^{-1})(b_m z^{m-n} + b_{m-1} z^{m-1-n} + \dots + b_1 z^{1-n} + b_0 z^{-n}) \end{aligned} \quad (2.45)$$

Aplicando a transformada z inversa para o domínio de k e colocando u(k) em evidência, tem-se,

$$\begin{aligned} u(k) &= -a_{n-1}u(k-1) - \dots - a_1u(k+1-n) - a_0u(k-n) \\ &+ b_m e(k+m-n) + b_{m-1}e(k+m-1-n) + \dots \\ &+ b_1 e(k+1-n) + b_0 e(k-n) \end{aligned} \quad (2.46)$$

Através da Equação 2.46 é possível calcular o valor da ação de controle (saída do controlador) utilizando os valores passados do erro e da própria saída do controlador. Essa equação é conhecida como lei de controle do controlador digital e sua aplicação é dada no algoritmo de implementação do controlador (PHILLIPS & NAGLE, 1994).

De acordo com Franklin (2010), é importante lembrar que os métodos de digitalização usando equivalentes discretos são aproximações, não há solução exata para todas as possibilidades de entrada porque G(s) responde pelo histórico de tempo completo de e(t), enquanto que G(z) só tem acesso ao sinal amostrado e(kT<sub>s</sub>). De certo modo, as várias técnicas de digitalização simplesmente fazem suposições diferentes sobre o que acontece com e(t) entre os pontos de amostragem (FRANKLIN et al.,2010).

### 2.2.3.1 Método de discretização por mapeamento de polos e zeros

Nesse método de digitalização, também conhecido como polo-zero combinado,  $G(z)$  é encontrado extrapolando a relação entre os polos  $s$  e  $z$ . Se tomarmos a transformada  $z$  de uma função amostrada  $x(k)$ , então os polos de  $X(z)$  estão relacionados aos polos de  $X(s)$  de acordo com a relação  $z = e^{sT}$ . A técnica *Matched Pole-Zero* (MPZ) aplica a relação  $z = e^{sT}$  aos polos e zeros de uma função de transferência, embora, estritamente falando, essa relação não se aplica nem à transferência de funções, nem mesmo aos zeros de uma sequência de tempo. Como todos os métodos de digitalização da função de transferência, o método MPZ é uma aproximação. A aplicação dessa aproximação é motivada, em parte, pelo fato de que  $z = e^{sT}$  ser a transformação correta de  $s$  para  $z$  e, parcialmente, pela quantidade mínima de álgebra requerida para determinar a função de transferência digitalizada à mão, assim como para facilitar a verificação dos cálculos do computador (FRANKLIN et al.,2010).

Deve-se observar também que, como os sistemas físicos geralmente têm mais polos que zeros, é útil adicionar arbitrariamente zeros em  $z = -1$ , de modo a igualar o número de zeros e polos do controlador discreto, resultando em um termo  $1 + z^{-1}$  em  $G(z)$  (FRANKLIN et al.,2010).

Para exemplificar melhor o método, um exemplo de aplicação é apresentado a seguir. Seja a equação de transferência 2.47 de um processo contínuo.

$$G(s) = K_c \frac{s + a}{s + b} \quad (2.47)$$

A equação discretizada de  $G(s)$  pela técnica de MPZ é dada por:

$$G(z) = K_d \frac{z - e^{-aT}}{z - e^{-bT}} \quad (2.48)$$

Onde:

$$K_c \frac{a}{b} = K_d \frac{1 - e^{-aT}}{1 - e^{-bT}} \quad (2.49)$$

e

$$K_d = K_c \frac{a}{b} \left( \frac{1 - e^{-bT}}{1 - e^{-aT}} \right) \quad (2.50)$$

Para um  $G(s)$  com um denominador de ordem superior, deve-se adicionar o termo  $(z + 1)$ . Para exemplificar, considere o processo contínuo representado pela equação 2.51.

$$G(s) = K_c \frac{s + a}{s(s + b)} \quad (2.51)$$

Dessa forma, discretizando  $G(s)$ , obtemos  $G(z)$  pela equação 2.52.

$$G(z) = K_d \frac{(z+1)(z-e^{-aT})}{(z-1)(z-e^{-bT})} \quad (2.52)$$

Onde:

$$K_d = K_c \frac{a}{2b} \left( \frac{1-e^{-bT}}{1-e^{-aT}} \right) \quad (2.53)$$

Nos métodos de digitalização descritos até agora, o mesmo poder de  $z$  aparece no numerador e no denominador de  $G(z)$ . Isto implica que a diferença de equação no tempo  $k$  exigirá uma amostra da entrada no momento  $k$ . Por exemplo, o  $G(z)$  na equação 2.48 pode ser escrito como:

$$G(z) = K_d \frac{1-\alpha z^{-1}}{1-\beta z^{-1}} \quad (2.54)$$

Onde:

$$\alpha = e^{-aT} \quad (2.55)$$

e

$$\beta = e^{-bT} \quad (2.56)$$

Por inspeção, verifica-se que a 2.54 resulta na equação da diferença 2.57

$$u(k) = \beta u(k-1) + k_d [e(k) - \alpha e(k-1)] \quad (2.57)$$

### 2.2.3.2 Método de mapeamento de polos e zeros modificado

No método MPZ, observa-se pela equação 2.57, que  $u(k)$  é dependente de  $e(k)$ , no mesmo tempo de amostragem  $k$ . Se a estrutura do hardware do computador utilizado não permitir essa relação, ou se as composições forem particularmente longas, pode ser desejável derivar  $G(z)$  para que tenha uma menor potência de  $z$  no numerador do que no denominador; portanto, a saída do computador  $u(k)$  exigiria apenas a entrada do tempo anterior, ou seja,  $e(k-1)$ . Para fazer isso, deve-se basicamente, modificar a sequência de polo e zero correspondente, de modo que o numerador seja de ordem menor que o denominador por 1 (FRANKLIN et al., 2010).

Seja a equação de transferência 2.58 de um processo contínuo.

$$G(s) = K_c \frac{s+a}{s(s+b)} \quad (2.58)$$

Discretizando  $G(s)$ , obtemos  $G(z)$  pela equação 2.59.

$$G(z) = K_d \frac{z - e^{-aT}}{(z - 1)(z - e^{-bT})} \quad (2.59)$$

Onde:

$$K_d = K_c \frac{a}{b} \frac{1 - e^{-bT}}{(1 - e^{-aT})} \quad (2.60)$$

Para encontrar a equação da diferença, multiplicamos a parte superior e inferior da equação 2.60 por  $z^{-2}$  para obter a equação 2.61.

$$G(z) = K_d \frac{z^{-1}(1 - e^{-aT}z^{-1})}{1 - z^{-1}(1 + e^{-bT}) + z^{-2}e^{-bT}} \quad (2.61)$$

Por inspeção, verifica-se que a 2.61 resulta na equação da diferença 2.62.

$$u(k) = (1 + e^{-bT})u(k - 1) - e^{-bT}u(k - 2) + k_d[e(k - 1) - e^{-aT}e(k - 2)] \quad (2.62)$$

Na equação 2.62, um intervalo de amostragem completo está disponível para realizar o cálculo da saída  $u(k)$ , porque depende apenas de  $e(k-1)$ . Uma análise discreta desse controlador, portanto, expõe mais precisamente o comportamento do sistema real. No entanto, como esse controlador está usando dados de um ciclo anterior, ele normalmente não funciona tão bem quanto o controlador MPZ em termos de desvios do sistema desejado, na presença de distúrbios aleatórios (FRANKLIN et al.,2010).

### 2.3. Sistemas de Controle

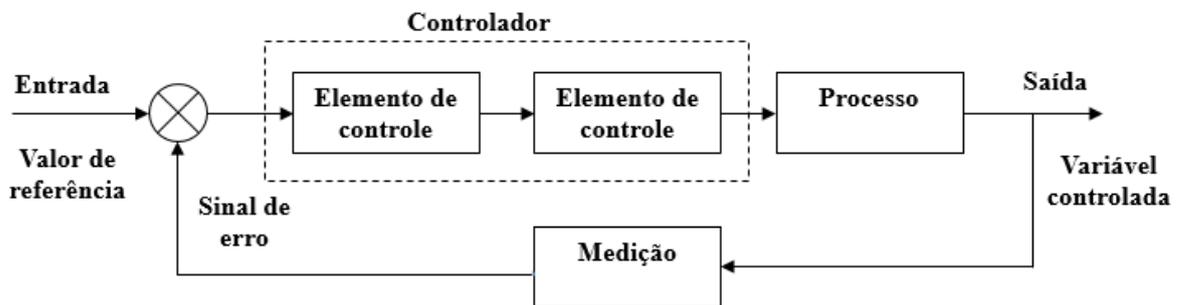
O controle de processo industrial é um campo do controle em que várias tecnologias de detecção e automação são usadas para supervisionar as atividades de produção em uma fábrica. Nas últimas décadas, o uso de controle de processo tem crescido rapidamente em conjunto com vários sistemas de gerenciamento de fábrica, como gerenciamento de estoque, verificação de qualidade de produto, monitoramento de segurança de fábrica e controle ambiental, otimizando toda a operação da planta. No controle do processo industrial, as estratégias de controle geralmente são construídas em torno da organização do sensoriamento e da atuação para atender a tarefas de controle abrangentes e complicadas. Os dados de interesse no controle de processo consistem principalmente nas variáveis de processo (PVs) e variáveis manipuladas (MVs), sendo os PVs adquiridos dos sensores de campo e as MVs comandos que são aplicados aos dispositivos de atuação. O objetivo do controle envolve atingir e manter os valores de referência

para essas variáveis de importância. Para alcançar esses objetivos, os dados de controle precisam ser comunicados de forma confiável por meio de links de comunicação seguros e essa comunicação não deve ter tempo de atraso demasiadamente elevado, para não comprometer o tempo de resposta necessário e não causar prejuízos à estabilidade do processo.

### 2.3.1 Ações Básicas de Controle

Para que os requisitos de controle sejam atendidos e os sistemas de controle funcionem conforme as expectativas projetadas, o controlador deve receber os dados provenientes dos sensores e calcular como deve ser a ação dos atuadores do processo para compensar o sinal de erro, baseado em parâmetros de referência fornecidos pelos usuários, de forma a atingir e manter o estado objetivo. O sinal de saída do controlador, portanto, é o sinal de entrada para o elemento corretivo e o responsável pela efetividade do controle, conforme ilustrado na figura 2.7.

Figura 2.7 – Subsistemas em um sistema de controle em malha fechada



Fonte: Adaptado de Ogata (2007)

Para que o controlador realize os cálculos e forneça a ação apropriada, podem ser empregadas diversas técnicas de controle, como um que liga e desliga, com o atuador baseado na diferença entre a leitura do sensor e o *setpoint* (controle ON/OFF), o controle Proporcional Integral e Derivativo (PID), lógica Fuzzy, redes neurais artificiais e algoritmos evolutivos. O controle Proporcional, Integral e Derivativo (PID) é a estratégia de controle em malha fechada mais utilizada na indústria, estimando-se que mais de 90% dos controles em malha fechada utilizem alguma variante do controle PID (ATZINGEN, 2017).

#### 2.3.1.1 Ação Proporcional

A ação proporcional é a mais usual dentre as ações de controle em processos industriais. O efeito da ação proporcional dá-se de acordo com a equação 2.50, onde o fator  $K_p$  é o ganho

do controlador. A saída do controlador é diretamente proporcional ao sinal de erro  $e(t)$  que é uma função do tempo.

$$u_p(t) = K_p e(t) \quad (2.63)$$

O objetivo principal da ação proporcional é estabilizar a variável de processo. O controlador atua corrigindo o erro existente no processo até que uma nova posição de equilíbrio seja estabelecida. Todavia, quando aplicado de forma pura e em processos sem pólos na origem, apresenta um erro final, também conhecido por *offset* ou erro de regime permanente. Esse erro torna-se um limitante para aplicações exclusivas de controle puramente proporcionais. Deve-se levar em conta ainda, que um aumento no valor do ganho proporcional tem impacto direto na velocidade de resposta do sistema e no valor do erro em regime permanente.

### 2.3.1.2 Ação Integral

Para solucionar os problemas de erro de regime permanente, uma das formas mais usuais aplicadas no controle de processo é a inclusão de um elemento de ação integral. O efeito da ação integral é representado pela equação 2.64, onde a saída  $u_i(t)$  deste controlador é proporcional à integral do erro  $e(t)$  com o tempo.

$$u_i(t) = K \frac{1}{T_i} \int_0^t e(t) dt \quad (2.64)$$

A ação integral elimina erro em regime permanente para entradas com ordem menores que o novo sistema de controle porque a ação integral adiciona um polo na origem, ou seja, adiciona um “s” no denominador da função transferência. Por esse motivo, podemos dizer que a ação integral é complementar à ação proporcional e seu objetivo é remover qualquer erro de desvio permanente.

Entretanto a ação integral apresenta alguns problemas. A introdução de pólos sem a adição de zeros gera uma desigualdade de uma unidade entre pólos e zeros, o que implica em levar o gráfico do lugar das raízes para mais próximo do semi-plano direito do plano s, ou seja, reduz a estabilidade relativa. Outro fator importante a ser observado é que, quanto maior for a razão  $K/T_i$ , mais o sistema tenderá a apresentar um comportamento oscilatório, além de um sobrepasso elevado. Pode-se verificar também que, a ação integral elimina o desvio permanente, mas não consegue eliminar o pico do erro (*overshoot*), uma vez que o mesmo ocorre antes que ela comece a atuar, pois está atrasada em relação à ação de controle proporcional.

### 2.3.1.3 Ação Derivativa

A ação derivativa tem como alvo melhorar o comportamento transitório do sistema de malha fechada, tomando ações que apressam a ação corretiva do controlador. A ação do controlador derivativo é dada pela equação 2.65.

$$u_d(t) = K \left[ T_d \frac{de(t)}{dt} \right] \quad (2.65)$$

Onde a saída  $u_d(t)$  é proporcional à derivada do sinal de erro. Portanto, qualquer erro de variação lenta, mesmo que seja grande, se aumentar de forma constante, não resultará em ação corretiva por parte do controlador derivativo, tornando-se uma desvantagem deste tipo de controlador e não viabilizando o uso desta ação de forma pura, apenas combinado com outras formas de controle.

### 2.3.1.4 O Controlador PID

Combinando-se as ações proporcional, integral e derivativa apresentadas anteriormente, obtemos o algoritmo de controle PID básico encontrado em grande parte da literatura, representado pela equação matemática 2.66.

$$u(t) = K_p e(t) + K \frac{1}{T_i} \int_0^t e(t) dt + K \left[ T_d \frac{de(t)}{dt} \right] \quad (2.66)$$

O sinal de controle é representado por  $u(t)$ , e têm a função de corrigir o erro de controle, e  $e(t)$  é resultado da diferença entre o *setpoint* e o valor medido na saída do processo. Assim, o sinal de controle é a soma de três termos: o proporcional, o integral e o derivativo. Os parâmetros do controlador são o ganho  $K$ , o tempo integral  $T_i$  e o tempo derivativo  $T_d$ . Passando para o plano  $s$ , três diferentes estruturas do algoritmo de controlador PID são mais comumente encontradas na literatura: a forma padrão ou ISA (*The Instrumentation Systems and Automation Society*), a forma paralela e a forma série. As equações a seguir representam cada forma respectivamente (FERNANDES, 2009).

$$G_c = \frac{U(s)}{E(s)} = K_c \left( 1 + \frac{1}{T_i s} + T_d s \right) \quad (2.67)$$

$$G_c = \frac{U(s)}{E(s)} = K_c + \frac{1}{T_i s} + T_d s \quad (2.68)$$

$$G_c = \frac{U(s)}{E(s)} = K_c \left( \frac{T_i s + 1}{T_i s} \right) + (T_d s + 1) \quad (2.69)$$

Considerando que, a maioria dos processos industriais usam microprocessadores para realizar os cálculos de controle, sejam estes centralizados ou distribuídos, os equipamentos digitais recebem os sinais, amostram as variáveis de controle e calculam as saídas do controlador de forma discreta. Portanto, o controle é realizado com intervalos de amostragem  $T_s$  regulares (FERNANDES, 2009).

Devido às diferenças apresentadas para os parâmetros dos controladores, e das diferentes formas que podem ser calculados, é necessário saber explicitamente em qual algoritmo é baseado o controlador utilizado para aplicar a sintonia correta (Fernandes, 2009).

O PID do controlador do PLC usado nesse trabalho, da fabricante SMAR, utiliza de bloco de controle baseado no modelo ISA, como mostrado na equação 2.67. Para a configuração deve-se manipular os parâmetros GAIN, RESET e RATE, que são as constantes de sintonia para os termos Proporcional, Integral e Derivativo, respectivamente. Outra característica que deve ser observada ao utilizar esse bloco de controle é a possibilidade de escolher entre ação de controle direta ou reversa. Na ação direta o erro será obtido subtraindo o *setpoint* (SP) da variável do processo (PV), enquanto na ação reversa o erro é calculado subtraindo a PV do SP.

Para que o controle PID funcione de forma eficiente, é importante que os parâmetros  $K_c$ ,  $T_i$  e  $T_d$  sejam os melhores possíveis de acordo com as especificações desejadas. Para obter esses parâmetros diversas técnicas de sintonia podem ser aplicadas, desde sintonia manual, que é a mais simples, até a implementação de sistemas automatizados com uso de aprendizado de máquina (ATZINGEN, 2017).

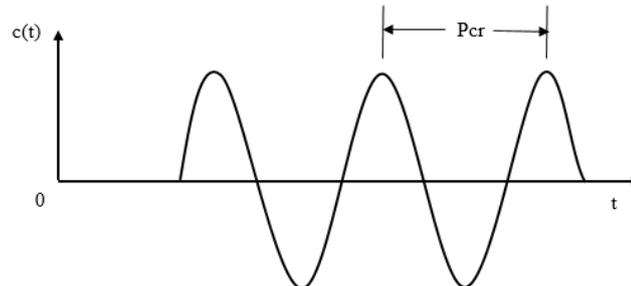
#### 2.3.1.5 Sintonia de malhas de controle PID

A sintonia de malhas de controle é um procedimento fundamental no desempenho dos processos industriais. É a sintonia que irá garantir a estabilidade do sistema de controle, como será a qualidade do produto final e o controle de insumos do processo de forma eficiente. A sintonia de um controlador PID consiste na determinação dos parâmetros Proporcional, Integral e Derivativo que proporcionem melhor robustez e que atenda aos critérios de desempenho desejados para o processo (Fernandes, 2009).

Uma das técnicas mais difundidas e frequentemente empregadas para a sintonia de controladores foi proposta por Ziegler e Nichols em 1942. Essa técnica é baseada no valor de  $K_p$ , que resulta em estabilidade marginal, e é útil quando os modelos matemáticos das plantas são desconhecidos, sugerindo valores de  $K_p$ ,  $T_i$  e  $T_d$  que resultarão em uma operação estável do sistema. O método de estabilidade marginal de Ziegler e Nichols consiste em primeiramente

definir  $T_i = \infty$  e  $T_d = 0$ . Na sequência aumentar o valor de  $K_p$  gradativamente de zero ao valor crítico  $K_{pr}$ , em que a saída mostra uma oscilação sustentada pela primeira vez, conforme exemplificado pela figura 2.8.

Figura 2.8 – Oscilação sustentada para aplicação da sintonia pelo método de Z&N



Fonte: Adaptado de Ogata (2010)

Após esse procedimento, com os valores de  $K_{cr}$  e  $P_{cr}$  obtidos experimentalmente, Ziegler-Nichols sugeriram a aplicação das relações matemáticas apresentadas na tabela 2.1 para calcular os valores de  $K_p$ ,  $T_i$  e  $T_d$ .

Tabela 2.1 – Regra de sintonia de Ziegler-Nichols baseados no ganho crítico  $K_{cr}$  e no período crítico  $P_{cr}$  - Fonte Ogata, 2007, p.560

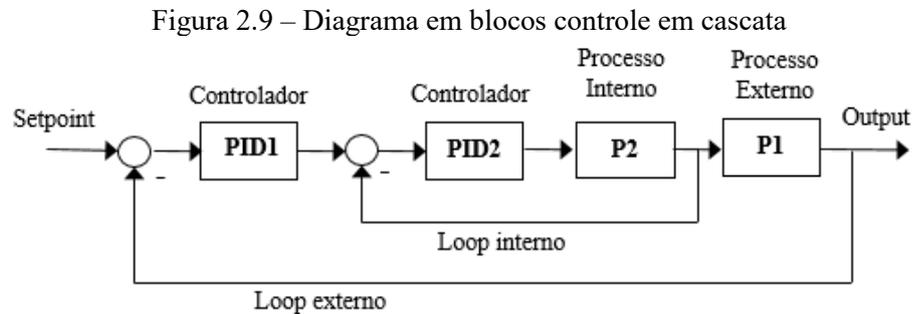
| Tipo de Controle | $K_P$         | $T_i$          | $T_d$          |
|------------------|---------------|----------------|----------------|
| <b>P</b>         | 0,5 $K_{cr}$  | $\infty$       | 0              |
| <b>PI</b>        | 0,45 $K_{cr}$ | 1/1,2 $P_{cr}$ | 0              |
| <b>PID</b>       | 0,6 $K_{cr}$  | 0,6 $P_{cr}$   | 0,125 $P_{cr}$ |

Apesar da técnica de sintonia apresentada ser de simples aplicação e atender à processos que não tem equações matemáticas conhecidas, de acordo com Ogata (2007), possui o inconveniente de não fornecem os valores definitivos dos parâmetros logo na primeira tentativa, e sim estimativas desses valores, proporcionam o ponto de partida na sintonia fina. Dessa forma, deve-se analisar o sistema resultante e se for preciso fazer uma série de sintonias finais até que um resultado aceitado seja obtido.

### 2.3.2 Controle de processos em cascata

Em alguns processos, que apresentam grande inércia e uma contínua perturbação na variável regulante, a utilização da malha de controle simples não atende satisfatoriamente aos requisitos do projeto. O controle cascata ajuda nesse tipo de aplicação por obter rejeição rápida de perturbação antes que essa se propague para as outras partes da planta. Nessa técnica, um controlador primário atua na regulação de um controlador secundário, melhorando a velocidade de resposta e reduzindo as perturbações. Normalmente encontra-se duas variáveis de processo,

dois controladores e um elemento final de controle (WADE, 2004). A Figura 2.9 apresenta um diagrama em blocos da estrutura de controle em cascata.



Fonte: Adaptado de Ogata (2007)

Como ilustrado, a malha de controle é composta por dois controladores com realimentação negativa. A saída do controlador primário (mestre) estabelece o ponto de ajuste (SP) do controle secundário (escravo) e a saída do escravo atua no elemento final de controle. Essa estrutura forma, então, duas malhas fechadas de controle, uma interna e outra externa.

Durante o projeto da estratégia de controle em cascata deve-se observar se a malha interna (variável secundária) responde mais rapidamente às variações no distúrbio e na variável manipulada do que a malha externa (variável primária). O desdobramento de uma malha comum em um sistema em cascata só é útil e aplicável quando houver a disponibilidade de uma variável intermediária de controle mais rápida que a variável primária, e quanto maior essa diferença melhor (SMITH & CORRIPIO, 2008).

De acordo com Wade (2004) a constante de tempo da malha interna deve ser pelo menos três vezes menor que a constante de tempo da malha externa e idealmente cinco vezes menor. Ele ainda afirma que, caso a velocidade de resposta da malha externa esteja com valores próximos à malha interna pode ocorrer uma interação adversa entre os dois controladores da malha, com características de “disputa”. Além disso, afirma também que caso a malha externa responda rapidamente às variações de entradas e ruídos, o uso de controle cascata não se faz necessário, uma vez que um controlador de realimentação simples já seria suficiente para lidar com ambos os processos, interno e externo.

Alguns procedimentos devem ser adotados ao considerar a configuração de malhas de controle em cascata, são eles:

- 1) O controlador na malha externa e interna devem ser colocados em modo manual, nessa ordem.
- 2) A malha interna deve ser identificada e seu controlador projetado para atuar com menor tempo de acomodação possível.

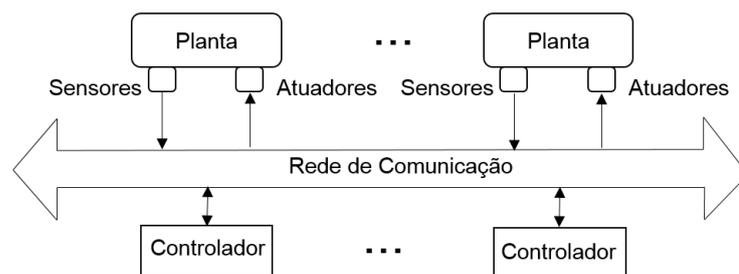
- 3) Após obter os parâmetros do controlador da malha interna, colocá-la em modo automático, mantendo a malha externa em modo manual.
- 4) Identificar a malha externa e regular o tempo de acomodação pelo menos três vezes maior que o ajustado na malha interna.
- 5) E, finalmente, colocar a malha externa em modo de operação automático.

### 2.3.3 Sistema de controle via rede de comunicação

Os Sistemas de Controle de Rede (NCSs) consistem em sistemas distribuídos cuja operação entre sensores, atuadores e controladores é coordenada por meio de uma rede de comunicação digital de banda limitada compartilhada (ATZINGEN, 2017). A característica universal dos NCSs é que os equipamentos são distribuídos espacialmente, podem operar de maneira assíncrona, mas têm sua operação regulada para atingir a um objetivo geral (BAILLIEUL & ANTSAKLIS, 2007).

A proliferação desses sistemas levantou questões importantes em comunicações, processamento de informações e controle, lidando com a relação entre as operações da rede e a qualidade da operação geral do sistema (BAILLIEUL & ANTSAKLIS, 2007). A figura 2.10, ilustra a conexão dos sensores e atuadores aos controladores via redes de comunicação, como Fieldbus, Profibus, Modbus, etc.

Figura 2.10 – Arquitetura geral de um NCS



Fonte: Adaptado de Hespanha et al (2007)

Um sensor fornece medições e status de uma propriedade física, por exemplo, a temperatura de um líquido associado ao processo. Com base na medição dos sensores, o controlador determina qualquer ajuste nos atuadores que seja necessário para manter o processo em um valor alvo, ou seja, o ponto de ajuste para que o processo tenha o comportamento desejado. O loop de controle é executado periodicamente a uma velocidade rápida o suficiente para corrigir quaisquer desvios indesejados no processo (HESPANHA et al., 2007).

A aplicação de redes industriais na interconexão entre os elementos de controle tem como benefício proporcionar uma arquitetura mais flexível e menores custos de instalação e manutenção dos processos (ATZINGEN, 2017). Todavia os NCSs apresentam como desvantagem alguns inconvenientes que não existiam em sistemas analógicos e de difícil detecção como: atrasos, instabilidade, limitações de largura de banda e perda de pacotes. O principal desafio na NCS é compreender como esses fatores influenciam no desempenho e na estabilidade do sistema e estudar a melhor maneira de minimizar esses impactos levando em consideração as aplicações para as quais cada rede é designada e o tipo de rede NCS implementado (CALDIÉRI, 2016).

Além do uso da rede em aplicações de controle, também é possível trafegar no mesmo meio outros dados do processo, como informações de diagnóstico e segurança. As informações de diagnóstico são geralmente utilizadas para monitorar o estado de equipamentos em uma planta industrial, já as informações de segurança são usadas para implementar funções críticas, tais como o fechamento seguro de um equipamento e a operação de circuitos de proteção. Em ambas as aplicações, assim como no controle de processo, o requisito tempo real é muito importante (LIANG, 2015).

Nesse contexto, dois conceitos são essenciais quando se trata de sincronizar as redes de comunicação para aplicação em tempo real, esses conceitos são determinismo e latência. Uma rede é considerada determinística quando as informações trocadas entre os seus dispositivos têm tempo de resposta entre a detecção de um item de dados e sua entrega ao dispositivo de controle regular e conhecido. Já o tempo real de transmissão, definido como suficientemente rápido para alcançar os objetivos da aplicação, é avaliado como tempo de latência. Habitualmente, o determinismo é alcançado evitando colisões de mensagens e limitando o tamanho máximo da mensagem. A baixa latência é obtida usando mídia de alta velocidade e minimizando o número de vezes que um sinal deve ser retransmitido, como em uma rede de malha.

De modo geral, para transmitir um sinal de tempo contínuo através de uma rede, o sinal deve ser amostrado, codificado em um formato digital, transmitido pela rede e os dados devem ser decodificados no lado do receptor. O atraso total entre a amostragem e a eventual decodificação no receptor pode ser altamente variável porque os tempos de acesso à rede e de transmissão dependem de condições de rede altamente variáveis, como congestionamento e qualidade do canal. Outro fator relevante está relacionado à possibilidade de perda de pacotes durante o trânsito na rede. Normalmente, as desistências de pacotes resultam de erros de

transmissão em links de rede física ou de estouro de buffer devido a congestionamento. Atrasos de transmissão longos às vezes resultam em reordenamento de pacotes, o que essencialmente equivale a um descarte de pacotes que chegam desatualizados. Alguns protocolos, considerados mais confiáveis, garantem a entrega final dos pacotes de dados, no entanto, não solucionam a totalidade dos problemas, uma vez que a retransmissão de dados antigos geralmente não é muito útil nas aplicações de NCSs (HESPANHA et al.,2007).

#### **2.4. Protocolo de Comunicação Modbus TCP/IP**

O MODBUS é um dos protocolos de mensagens mais utilizados em automação industrial e controle de processos, principalmente em razão de ser simples e de fácil implementação. Foi desenvolvido pela empresa *Modicon Industrial Automation Systems* na década de 70, hoje *Schneider Electric*. Em 2004 os direitos de uso do protocolo foram transferidos para a *Modbus Organization*, uma organização sem fins lucrativos, e a sua utilização é isenta de taxas de licenciamento. Dessa forma, a rede MODBUS é extensamente aplicada na integração de dispositivos, principalmente quando há necessidade de integração de múltiplos sistemas, devido à sua ampla utilização na indústria e custo reduzido (NASCIMENTO & LUCENA, 2003).

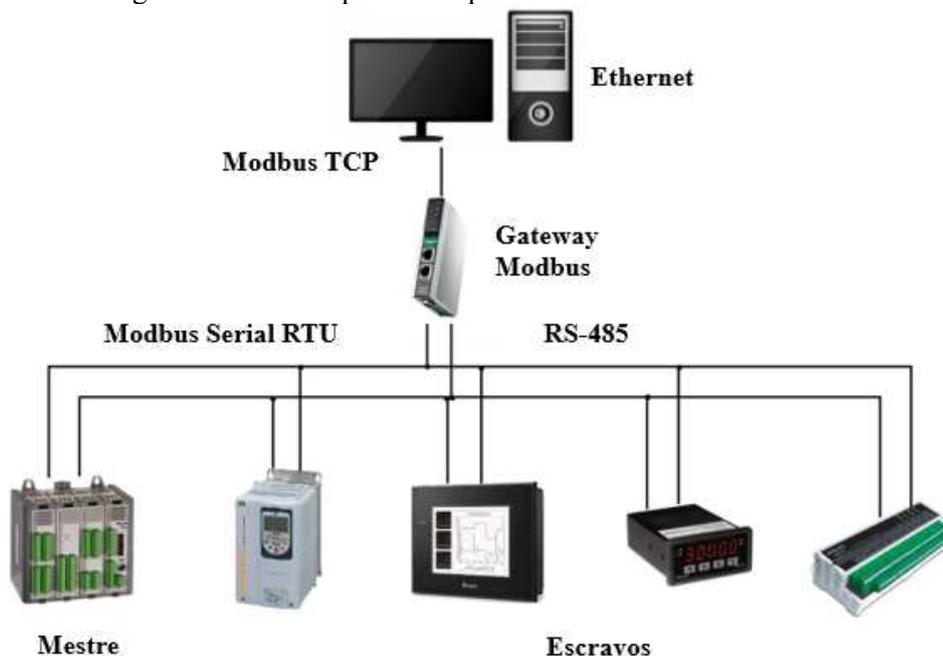
O Modbus é um protocolo de rede popular no ambiente de produção industrial e possui padrões de códigos de diversas aplicações disponíveis de forma livre. Uma das formas de implementação é usando o FreeMODBUS, que é uma implementação gratuita do protocolo MODBUS especialmente direcionado para sistemas embarcados. Desde a versão 0.7 FreeMODBUS suporta Modbus TCP definido no *Modbus Messaging* do Guia de Implementação do TCP / IP v1.0a. O suporte do sistema é fornecido pelo sistema operacional FREERTOS de código aberto e em tempo real. Ele é executado em uma plataforma mínima de hardware construída em torno do microcontrolador Atmel ATmega1284P de 8 bits, com 16 KB de RAM interna e 128 KB de memória Flash interna. O recebimento e transmissão de quadros de mensagens é implementado como uma máquina de estados que é acionada por retornos de chamada da camada de abstração de hardware, isso facilita a portabilidade para novas plataformas e possibilita adicionar futuramente novas funções Modbus.

O Modbus utiliza a arquitetura Mestre-Escravo, ou Cliente-Servidor, e permite a comunicação em diferentes tipos de barramentos e redes, como linhas seriais e redes TCP/IP. O protocolo serial opera basicamente de duas formas, ASCII (*American Standard Code for*

*Information Interchange*) e RTU (*Remote Terminal Unit*). Outras versões, em que o protocolo também é encontrado, são as derivações do Modbus como o Modbus Plus e o Modbus TCP/IP (STEMMER, 2010).

O modo de operação ASCII envia cada byte da mensagem como dois caracteres ASCII, o que torna a transmissão assíncrona. Já no modo RTU, cada byte da mensagem é enviado como dois caracteres hexadecimais de 4 bits, esta alta densidade de caracteres aumenta o fluxo de dados quando comparado com o modo ASCII, exigindo uma transmissão síncrona de dados. (STEMMER, 2010). A figura 2.11 mostra um exemplo de rede com protocolo Modbus.

Figura 2.11 – Exemplos de dispositivos conectados em Modbus



Fonte: Adaptado do site [www.embarcados.com](http://www.embarcados.com)

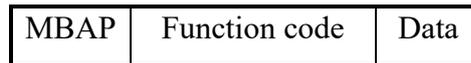
Uma das versões disponíveis de comunicação de rede Modbus, o Modbus Plus, conta com diversos recursos adicionais de roteamento, diagnóstico, endereçamento e consistência dos dados, que possibilitam aumentar a velocidade dos dispositivos em rede. Entretanto esta versão é mantida sob o domínio da empresa Schneider Electric. Para a aplicação Modbus TCP, que é livre, os dados são encapsulados em formato binário e utiliza-se o protocolo TCP/IP e Ethernet para transporte os dados.

Nos tópicos seguintes serão exploradas, especificamente, as características do protocolo Modbus TCP, utilizado ativamente na realização deste trabalho.

### 2.4.1 Modelo das Mensagens

O protocolo Modbus TCP utiliza a pilha TCP/IP para comunicação e adiciona ao quadro Modbus um cabeçalho específico chamado MBAP (MODBUS Application Protocol). O modelo de mensagem Modbus TCP/IP é mostrado na Figura 2.12.

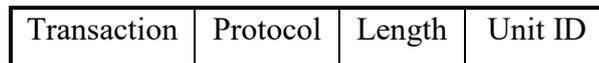
Figura 2.12 – Modelo de Mensagem Modbus



Fonte: Adaptado de Modbus Organization (2012)

O cabeçalho MBAP é de uso geral, depende de uma camada de rede confiável, tem tamanho de 7 bytes, e é composto pelos campos apresentados na Figura 2.13.

Figura 2.13 – Cabeçalho MBAP Modbus

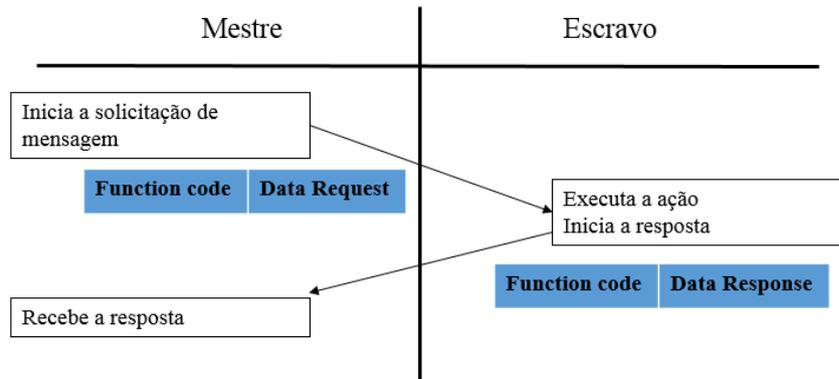


Fonte: Adaptado de Modbus Organization (2012)

- Transaction identifier: É um identificador de transações, usado para identificação da resposta para a transação nas requisições enviadas entre mestre e escravos (2 bytes);
- Protocol identifier: Normalmente é 0 (zero) para Modbus (2 bytes), mas pode ser usado como recurso para expandir o comportamento do protocolo;
- Length: contagem de todos os próximos bytes (2 bytes). É usado para projetar o comprimento do restante do pacote.
- Unit Identifier: O campo Unit ID normalmente não é usado para dispositivos TCP/IP. Entretanto pode ser usado para aplicações de integração de protocolo Modbus de TCP/IP para serial. Basicamente, é utilizado para identificar o escravo remoto em uma rede Modbus RTU (1 byte).

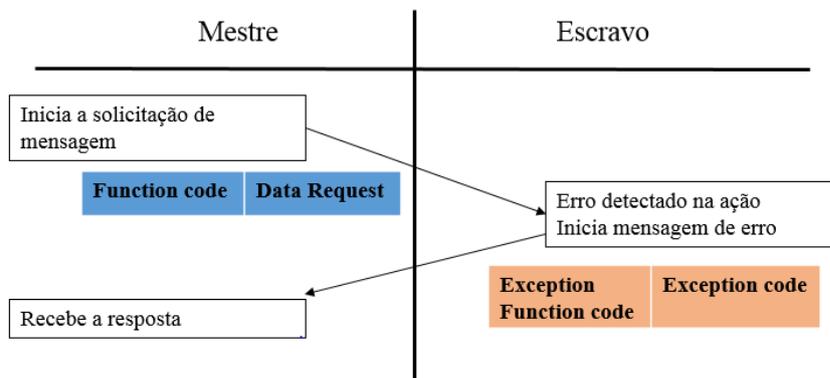
O segundo campo (*Function Field*), de 1 byte, especifica a função a ser executada pelo dispositivo que recebe a mensagem. As funções válidas estão na faixa entre 1 a 255, entretanto nem todas estão implementadas. Quando o dispositivo receptor for capaz de executar a dada função, será enviada uma mensagem de resposta com o resultado da operação. O dispositivo mestre deve verificar o campo de função das respostas dos escravos, pois este campo também tem a finalidade de informar se houve problemas com a função solicitada. Um exemplo de como essa verificação é feita é apresentado nas Figuras 2.14 e 2.15. As funções mais comuns são as de leitura e escrita de dados, mas existem também funções de diagnóstico e de acesso de arquivos.

Figura 2.14 – Comunicação entre Mestre e Escravo Modbus sem erros



Adaptado de Modbus Organization (2012)

Figura 2.15 – Comunicação entre Mestre e Escravo Modbus com erros



Adaptado de Modbus Organization (2012)

Dessa forma, o escravo, na sua resposta ao mestre, usa o campo de código da função para indicar se a resposta enviada é normal ou se algum erro ocorreu. Para uma resposta normal, o escravo apenas copia para a resposta o código da função original. Entretanto, na ocorrência de um erro, é enviada uma resposta de exceção (*exception response*), o escravo retorna um código de função equivalente ao original, mas com o bit mais significativo em 1.

O terceiro campo (*Data*) pode variar de 00H à FFH e contém informações relacionadas com o código da função no campo de funções, como por exemplo, o número de variáveis discretas a serem lidas ou ativadas.

#### 2.4.2 Modelos requisição-resposta

O Protocolo Modbus é baseado no modelo de comunicação mestre-escravo, no qual apenas um único dispositivo mestre pode inicializar a comunicação, esse pedido também é conhecido como query. Os demais dispositivos escravos, respondem enviando os dados solicitados pelo mestre, ou realizam alguma ação solicitada por ele.

Os dados transmitidos pela rede podem ser discretos ou numéricos, podendo ser usados tanto por variáveis analógicas quanto digitais. O dispositivo mestre pode endereçar cada dispositivo escravo da rede individualmente ou acessar a todos da rede através de mensagens em *broadcast*. Quando a mensagem é enviada especificamente para um escravo, apenas o dispositivo acessado retorna uma mensagem de resposta a esta solicitação. Entretanto para as mensagens enviadas a todos os dispositivos de rede (*broadcast*) nunca são geradas respostas de retorno a essa solicitação. O modelo de requisição-resposta para o protocolo Modbus é apresentado na Figura 2.16.

Figura 2.16 – Modelo requisição-resposta para o Protocolo Modbus



Fonte: Adaptado de Nascimento & Lucena (2003)

Na existência de algum erro de comunicação, ou se o escravo não estiver apto para atender a função requisitada, o dispositivo escravo monta uma mensagem denominada *exception*, a qual justifica o não atendimento da função. Veja uma representação do modelo das mensagens na Figura 2.16.

#### 2.4.3 Modelos de dados

Os dados que podem ser acessados pelo MODBUS são armazenados, de forma geral, em um dos quatro bancos de dados, ou faixas de endereços: *Discretes Inputs*, *Coils*, *Input Registers* e *Holding Registers*. O comportamento de cada bloco é descrito no Quadro 2.1.

Quadro 2.1 – Características das tabelas MODBUS

| Tabela                  | Código | Tipo de Objeto | Acesso ao Mestre  | Comentários                                    |
|-------------------------|--------|----------------|-------------------|--|
| <i>Discretes Inputs</i> | 01     | 1 Bit          | Somente Leitura   | Pode ser gerado por um dispositivo de I/O      |
| <i>Coils</i>            | 02     | 1 Bit          | Leitura e escrita | Pode ser alterado por um programa de aplicação |

|                          |    |                  |                   |  |
|--------------------------|----|------------------|-------------------|--|
| <b>Input Registers</b>   | 03 | 1 Word (16 bits) | Somente Leitura   | Pode ser gerado por um dispositivo de I/O      |
| <b>Holding Registers</b> | 04 | 1 Word (16 bits) | Leitura e escrita | Pode ser alterado por um programa de aplicação |

Fonte: Adaptado de Modbus Organization (2012)

Os dados *Discretes Inputs* e *Coils* são de leitura de entradas e saídas discretas. Já os *Input Registers* e *Holding Registers* são de leitura dos registradores, sendo que, o primeiro é leitura dos valores das entradas dos registros e o segundo leitura de registradores do dispositivo escravo.

Nos bancos de dados apresentados, cada tipo tem os seus respectivos critérios de acesso. Os instrumentos escravos, por exemplo, têm acesso direto aos dados, pois estão localmente hospedados, enquanto que o dispositivo Mestre precisa solicitar o acesso por meio dos códigos de função. Os dados, que podem ser acessados pelo Modbus, são de forma geral um subconjunto da memória principal do dispositivo. Outra vantagem de os dados serem divididos nesse sistema de blocos é a de simplificar a obtenção a diferentes tipos de informações na camada de aplicação e a capacidade de restringir ou permitir acesso a esses dados.

#### 2.4.4 Regras de Endereçamento

O espaço de endereços MODBUS compreende 256 diferentes endereços, como é mostrado na Quadro 2.2.

Quadro 2.2 – Endereços Modbus

|                    |                                    |                    |
|--------------------|------------------------------------|--------------------|
| <b>0</b>           | <b>1 ... 247</b>                   | <b>248 ... 255</b> |
| Endereço broadcast | Endereços individuais dos escravos | Reservado          |

Fonte Adaptado de Modbus Organization (2012)

O endereço 0 (zero) é o endereço de Broadcast. Quando um mestre envia uma mensagem para esse endereço reservado, todos os escravos a recebem simultaneamente. Na faixa de 1 a 247, são os endereços disponíveis para cada escravo. Em um barramento Modbus, cada escravo deve ter um endereço único e exclusivo. O mestre endereça um escravo colocando seu respectivo endereço no campo de endereço da ADU, como visto na seção de Modelo das Mensagens. Quando este escravo retorna a resposta, ele coloca seu próprio endereço no campo de endereço da resposta para indentificá-lo. O mestre de uma rede MODBUS não possui endereço, apenas seus escravos. Os endereços de 248 a 255 são reservados e não devem ser usados (Modbus Organization, 2012).

## 2.5. Estado da Arte

Os sistemas de controle em rede (NCS), são sistemas em que os controles em circuito fechado são implementados no controlador de nível superior enquanto a aquisição de dados e a atuação do dispositivo são distribuídos no campo. Em muitos casos estudados, o não determinismo e o atraso no tempo na transmissão da comunicação causam degradação nos efeitos de controle. Muitos pesquisadores exploraram e investigaram métodos de melhoria em perspectiva de ciência de controle, como estudos sobre transmissão máxima permitida para controle, tempo máximo de atraso e apresentaram vários algoritmos alternativos para ajuste de parâmetros do controlador (Liang, 2015).

De acordo com Fadaei & Salahshoor (2008), o aumento no atraso de transmissão da rede degrada o desempenho de controle na abordagem de implementação do sistema de controle em rede. Para comprovar essa afirmação, demonstraram o projeto prático e a implementação de um sistema de controle em cascata baseado em rede, que conecta um controlador Smar (DFI-302) e um controlador lógico programável da Siemens (S7- 315-2DP) através de uma ethernet industrial. A interoperabilidade da configuração baseada em rede proposta foi garantida usando o navegador Smar OPC Tag e as instalações do servidor Siemens WinCC OPC.

Fadaei & Salahshoor (2008) também examinaram a performance de um controlador PID implementado em duas configurações diferentes entre blocos de funções (FB) Fieldbus Foundation (FF) e um sistema de controle em rede (NCS) por meio de um CLP Siemens remoto. Na implementação do sistema de controle FB, o ponto de ajuste desejado foi fornecido pelo software Interface Homem-Máquina (IHM) através de um link Ethernet Modbus. As malhas em cascata foram realizadas na estação CLP da Siemens remota e o ponto final do elemento enviado para a estação FF via Ethernet na implementação do NCS (Fadaei & Salahshoor, 2008). Embora Fadaei tenha apontado as causas do atraso de transmissão nas abordagens Fieldbus e NCS e tenha apresentado resultados experimentais dos efeitos de atraso em ambas as abordagens, a solução que ele propôs foi uma nova estratégia de controle, usando PID fuzzy.

Da mesma forma, Li (2009) também investigou os atrasos associados ao uso de redes Fieldbus Foundation (FF) H1 em malhas de controle do ponto de vista prático. Foram usados um sistema de controle distribuído (DCS) e uma estação de trabalho FF H1 para implementar teste em malha fechada com a arquitetura de controle no campo. Avaliações analíticas e experimentais foram realizadas com uma malha de teste usando canais analógicos conectados como um benchmark. Li (2009) identificou três segmentos de atrasos induzidos pela rede FF

H1, sendo estes: atrasos de comunicação, atrasos associados à filtragem/conversão e atrasos por execução de blocos de função. Li (2009) indicou que os atrasos induzidos pela rede FF-H1 podem ser significativos e sugeriram várias maneiras de reduzi-los. Sugeriu reduzir os atrasos no aspecto das propriedades específicas do Fieldbus, tais como no projeto do macro-ciclo, na configuração do intervalo de aquisição do sinal e com a eliminação do tempo ocioso entre alguma execução específica do FB e a comunicação. Além disso, Li (2009) verificou que um atraso adicional inesperado de um macrociclo pode ser introduzido e concluiu que atrasos significativos podem ser introduzidos se os canais analógicos tradicionais de um DCS forem substituídos por uma rede FF H1. Li (2009) definiu os atrasos induzidos pela rede como o tempo adicional necessário quando os canais analógicos usados pelos sistemas de controle distribuídos (DCSs) são substituídos por redes digitais. Li (2009) ressalta que sua contribuição pode ser usada por engenheiros de controle para estimar atrasos induzidos pelas redes FF H1 em seus modelos analíticos e possibilitar a análise prévia de seus impactos na estabilidade e no desempenho do sistema de controle.

Liang (2015), investigou por uma perspectiva diferente os algoritmos de controle. Para Liang é possível obter melhorias do desempenho de controle de um sistema de controle em rede(NCS), reduzindo o atraso de transmissão de dados (DTD). Foram apresentadas duas arquiteturas de rede diferentes para implementação do sistema e expostas melhorias no DTD para o sistema de controle experimental. Os efeitos da configuração do esquema de controle em DTD na forma de FB foram investigados e foram propostas melhorias correspondentes por realocação de FB e rearranjo de tabela de planejamento. Efeitos de controle sob as metodologias propostas também foram apresentados.

Liang (2015) comprovou experimentalmente que, esquema de configuração na atribuição de bloco de função para diferentes instrumentos de campo e disposição da tabela de programação pode afetar notavelmente o desempenho do controle. Com a alocação de blocos de função aprimorada e, conseqüentemente, o esquema de configuração proposto no artigo de Liang, um ciclo de controle mais curto e, assim, um controle mais eficiente pode ser alcançado. Constatou também que o desempenho de controle pode ser melhorado selecionando apropriadamente como combinar o ciclo de amostragem e o ciclo de controle.

Ainda em relação aos estudos do impacto do atraso de tempo das comunicações nos sistemas de controle, a pesquisa de Pang & Nishitani (2006) apontou que o tempo de execução dos FBs foi significativo sobre os atrasos de comunicação no Fieldbus FF. Para Pang & Nishitani (2006), otimizar a configuração dos parâmetros de rede, reduzindo o número de links

externos, pode contribuir para aumentar a frequência de controle e melhorar o seu desempenho. Pang & Nishitani (2006) também mostraram a variedade e a diversidade de abordagens para atingir essa meta.

Ainda no trabalho de Pang & Nishitani (2006), um PID ideal e um PID fuzzy foram ajustados, minimizando a integral de tempo multiplicada pelo Erro Absoluto (ITAE) e a saída do controlador ao quadrado, para um sistema de controle em rede (NCS). O ajuste é elaborado para um processo de ordem mais alta e um sistema de atraso de tempo usando o Algoritmo Genético (GA) e duas variantes da Otimização de Enxame de Partículas (PSO) e os desempenhos de malha fechada são comparados. O artigo mostra que a variação aleatória no retardo de rede pode ser tratada eficientemente com controladores PID baseados em lógica fuzzy sobre os controladores PID convencionais.

De acordo com Tran et al (2013) para projetar um controlador PI / PID em sistemas de controle em rede (NCSs) com atrasos, exige-se não apenas simplicidade e flexibilidade, mas também alta estabilidade especificada e robustez do sistema. Por abordagem, de ganho e de fase, Tran et al (2013) propõe um novo controlador PI / PID normalizado para NCSs, baseado na análise da estabilidade e robustez do sistema sob o efeito de atrasos induzidos pela rede. Especificamente, os autores levam em conta os atrasos totais medidos na rede para formular as margens de ganho e fase do sistema de malha fechada, na forma de um conjunto de equações. Este conjunto de equações é então resolvido para calcular as formas fechadas de parâmetros de controle que nos permitem propor o controlador PI / PID normalizado satisfazendo simultaneamente dois requisitos buscados: a simplicidade, sem resolver o problema de otimização para um novo processo, e a alta flexibilidade para lidar com grande escala de atrasos aleatórios e lidar com muitos processos diferentes em diferentes condições de rede.

No trabalho de Caldiéri (2016), foi analisada uma implementação em hardware dedicado (Arduino) do protocolo industrial Modbus para a comunicação via TCP/IP e Wi-Fi. Foi demonstrado a confiabilidade na comunicação via Modbus implementada, obtendo baixo “jitter” na transmissão de mensagens e baixo índice de perda de mensagens transmitidas. Caldiéri propôs que, para o controle de processos via rede que utilizam transmissores com e sem fio, principalmente nos casos de perda de dados transmitidos, presença de atrasos de comunicação variantes no tempo e alteração o período de amostragem, o controlador PIDPlus mostrando-se com melhor robustez e desempenho superior ao PID tradicional. Para testes de casos de falhas, tanto sequenciais quanto aleatórias, e nos casos de perdas de dados transmitidos, Caldiéri demonstrou que o PIDPlus foi capaz de manter desempenho equiparável

ao caso ideal (sem falhas), por esse motivo recomendou, como substituição ao PID, no controle de processos sujeitos a atrasos de rede de comunicação. Dessa forma, Caldiéri (2016) demonstrou que é possível adaptar instrumentos analógicos em dispositivos Modbus compatíveis e transmitir sinais de controle e realimentação.

Em relação ao uso de simulações, para Mossin (2007), é possível destacar duas abordagens distintas para laboratórios remotos desenvolvidos com foco na área de automação. A primeira é baseada em laboratórios com acesso remoto a instrumentos reais e a segunda em acesso remoto a simuladores dinâmicos de malhas de controle e, eventualmente, de dispositivos de campo. De acordo com Mossin (2007), a segunda abordagem tem a vantagem de permitir ao aluno fazer qualquer tipo de experimento, sem restrições de segurança, entretanto, o uso de dispositivos de campo e instrumentos reais tem a desvantagem de limitar o estudante a executar poucos tipos de experimentos para preservar sua segurança pessoal, a segurança dos dispositivos e até mesmo do laboratório. Para Mossin (2007), uma das vantagens do contato direto a dispositivos reais em relação ao acesso remoto a dispositivos simulados é proporcionar ao estudante uma experiência mais realista com equipamentos industriais. Entretanto, também deixa claro que nos casos em que o simulador for realista o suficiente, esta vantagem desaparece, tornando mais viável e mais interessante o sistema totalmente simulado.

# CAPÍTULO 3

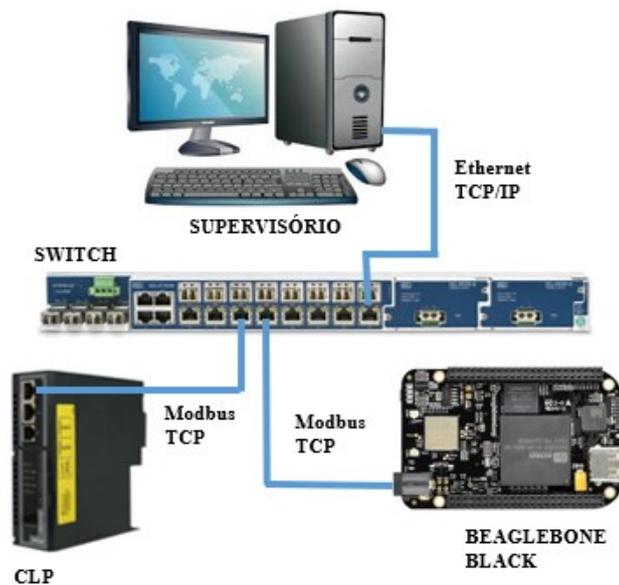
## Descrição dos Testes

Nesta seção serão apresentados a arquitetura de rede, as metodologias e os procedimentos utilizados para a realização dos testes de simulação realizados para validar a ferramenta computacional desenvolvida durante este trabalho.

### 3.1 Arquitetura e descrição do sistema

A estrutura do sistema de controle proposta deve fornecer a capacidade para implementar as estratégias de controle desejadas no banco de testes por meio de rede de comunicação remota. Para atingir a esse fim, a arquitetura é formada pela interconexão de componentes: uma plataforma de prototipação, um Controlador Lógico Programável (CLP), um computador e um switch Ethernet. A Figura 3.1 ilustra a estrutura da arquitetura do sistema de controle projetada para esse propósito de aplicação de controle prático.

Figura 3.1 – Arquitetura de rede do sistema de testes



A plataforma de prototipação Beaglebone Black (BBB) é baseada no processador AM3359, um ARM Cortex-A8 da Texas Instruments, que opera em até 720MHz, e possui 32K de cache L1, 256K de cache L2, 176K de ROM e 64K de RAM interna. Essa

plataforma foi utilizada neste trabalho para simular os dados de instrumentos e atuadores de uma planta industrial, por meio de modelos matemáticos. Além de gerar os dados, a BBB também tem a função de ler e escrever as informações em endereços dos registros da rede de comunicação para troca de mensagens com outros dispositivos. O CLP foi usado para recebendo as informações da planta simulada da BBB e efetuar o controle do processo.

Por meio de pesquisas, dentre os protocolos de rede disponíveis, o Modbus apresentou maior facilidade de implementação, por ser um protocolo aberto, possuir diversas aplicações de forma livre, dispor de implementações gratuitas e especialmente direcionadas para sistemas embarcados, além da facilidade para fazer a portabilidade para novas plataformas. Por essas razões, foi utilizado como uma rede de dados, comum entre o Beaglebone Black e o Controlador da marca SMAR modelo DF63.

O computador foi usado para supervisão e coleta de dados, para futura plotagem de gráficos e análise dos resultados, e o switch para conexão entre os equipamentos em rede.

O sistema de supervisão utilizou a comunicação Ethernet TCP/IP. O CLP representa o papel de mestre e a BBB atua na rede de comunicação como escravo. Dessa forma, o CLP tem acesso aos dados de processo gerados pela BBB por meio da comunicação Modbus mestre-escravo.

Usando esse conjunto de equipamentos, as aplicações de controle abrem a possibilidade para serem definidas em diferentes opções como modo manual, controle de loop único (PID), modo automático, controle em cascata, controle *override*, controle de razão, etc. Nos tópicos seguintes serão apresentados os procedimentos utilizados durante a realização deste trabalho e como foram estimados os parâmetros para o sistema de controle.

### **3.2 Metodologia dos testes**

Para verificar o desempenho da ferramenta computacional desenvolvida foram feitos testes com modelos SISO, lineares, de primeira e segunda ordem. Os modelos simulados possuem essa limitação devido ao processo de discretização implementado, que suporta polinômios de até segundo grau no numerador e denominador da função de transferência.

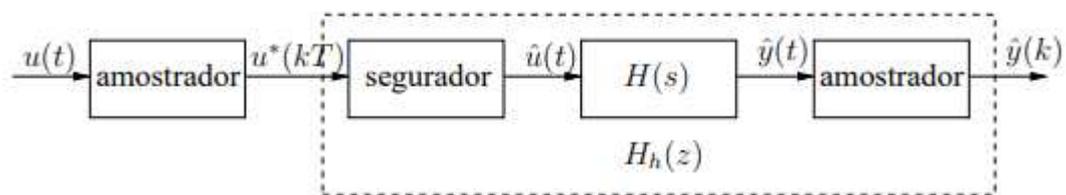
Inicialmente, os testes foram realizados com o software MATLAB®, para verificar a estabilidade e controlabilidade dos modelos e dessa forma, selecionar os processos a serem usados nos testes da ferramenta. Após a seleção dos modelos, que atenderam aos critérios preestabelecidos, de estabilidade e linearidade, os resultados obtidos foram armazenados para

serem utilizados como referência para contrastar com os resultados obtidos nas demais plataformas de testes. Na sequência, todos os modelos foram configurados em malha aberta, sendo que os resultados obtidos foram comparados com os fornecidos pelo software MATLAB®.

A partir dos testes de malha aberta, os métodos de ajuste de malha de controle, já exemplificados na introdução teórica deste trabalho, foram usados para encontrar os valores de  $K_p$ ,  $K_i$  e  $K_d$  para o controlador PID. Para melhores resultados, por meio de simulações, foi feito um ajuste fino nos valores calculados.

Com os modelos e os ajustes do controlador definidos, foram iniciados os procedimentos para os testes de simulação da função de transferência dentro da plataforma de prototipação. O programa interno gravado na BBB foi desenvolvido por meio do software Visual Studio, em linguagem C. O programa gravado contém os modelos contínuos dos processos, a função que transforma o modelo contínuo em discreto e o código de implementação do protocolo de comunicação Modbus. A construção de um sistema equivalente discreto do processo contínuo, ocorre conforme a figura 3.2.

Figura 3.2 – Construção de um sistema equivalente discreto



Fonte: Adaptado de Franklin (2010)

Conforme demonstrado na figura 3.2, o simulador recebe os valores da função de transferência em variáveis contínuas no tempo e, por meio de processo de discretização, gera a função discreta.

O acesso à BBB para execução de aplicativos, transferência de arquivos, instalação de aplicativos etc., pode se dar de diferentes maneiras. Na Beaglebone Black as formas mais comuns são por meio de porta serial, micro-USB e porta Ethernet. Ao se conectar a BBB pela micro-USB é gerado uma conexão Ethernet, dessa forma a BBB pode ser acessada pelo seu IP (192.168.7.2) ou com um servidor HTTP colocando o endereço <http://192.168.7.2> em qualquer browser.

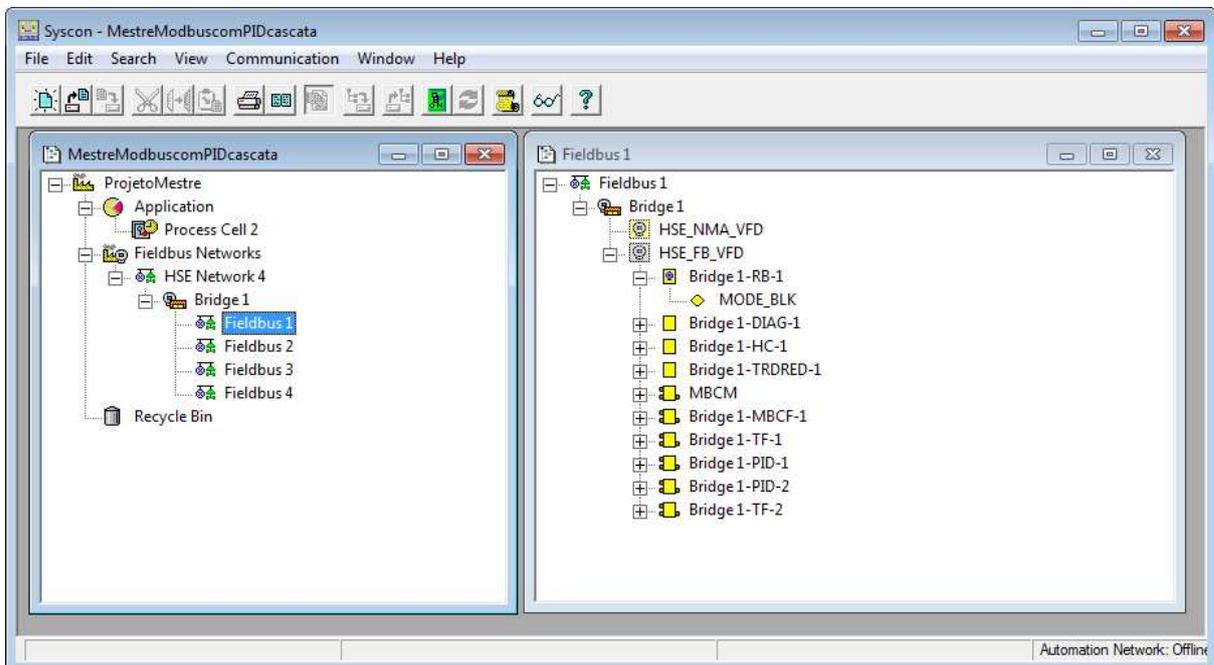
Na sequência, foi necessário fazer as configurações de rede. Nesta etapa foram preparados os cabos, switch e outros dispositivos de rede para conectar o CLP e a BBB na rede Modbus. Durante a configuração do software de rede, cada dispositivo foi configurado de forma

a ser reconhecido na rede pelos demais e desempenhar o papel que foi designado. A BBB foi configurada para atuar como escravo na rede Modbus e o CLP como mestre. A conexão por porta Ethernet possibilita aplicar protocolos que usam o meio físico derivados de protocolos TCP/IP, como o Modbus TCP, o que aumenta a possibilidade de interface com diversas redes e equipamentos, uma vez que permite a conexão com hubs e switches.

O próximo passo foi configurar o hardware e software do CLP. A configuração de hardware consiste em configurar o software do CLP para reconhecer os módulos do CLP instalados fisicamente. A configuração de software, consiste em programar o CLP para efetuar as funções de controle desejadas.

Para a configuração do mestre da rede, um Controlador Lógico Programável (CLP) marca SMAR modelo DF63, foi utilizado o software Syscon da fabricante SMAR. Por meio desse software, a programação é muito facilitada com recursos que permitem a configuração dos equipamentos e dos laços de controle na rede adicionar blocos que representam diferentes funções (controlador PID, entrada analógica, saída analógica etc.), e fazer as ligações apropriadas entre os blocos adicionados. Um exemplo é mostrado na Figura 3.3.

Figura 3.3 – Tela de configuração software Syscon da Smar

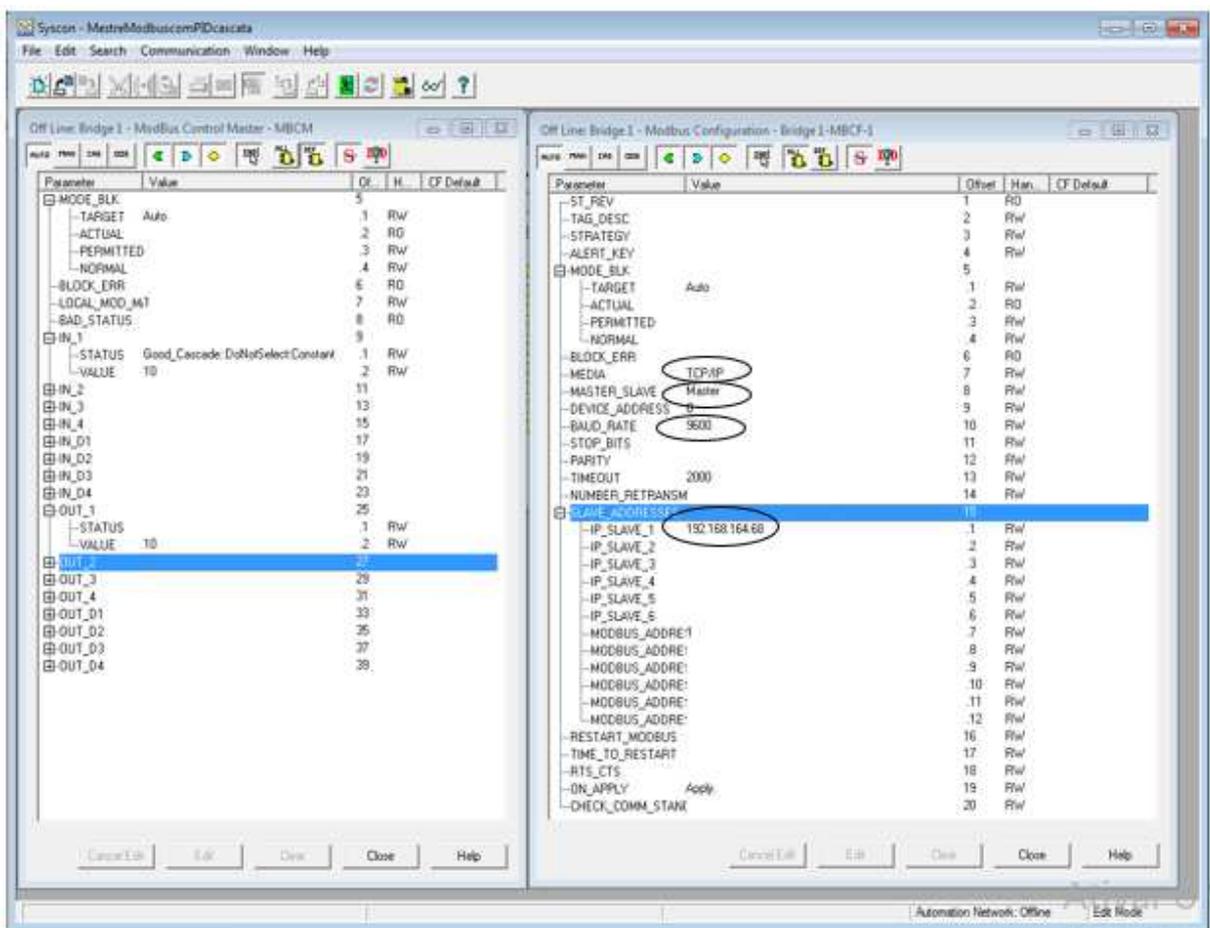


Um dos blocos disponíveis, e que foi usado no decorrer desse trabalho para validar os testes em malha fechada, é o bloco de funções de transferência. O bloco *Transfer Function* (TF), consiste em uma saída e uma entrada, e tem como finalidade representar sistemas de até 2ª ordem, através da configuração dos coeficientes A, B, C, D, E e F de uma função de transferência do tipo exemplificado na equação 3.1.

$$G(s) = \frac{As^2 + Bs + C}{Ds^2 + Es + F} \quad (3.1)$$

Para a configuração do CLP como mestre na rede Modbus, foram utilizados os blocos MODBUS *Configuration* (MBCF), que configura como serão transmitidos os dados através da rede, e MODBUS *Control Master* (MBCM), que é o bloco responsável por ler e escrever dados através da rede MODBUS. A figura 3.4 exemplifica a configuração os blocos MODBUS na tela do software Syscon.

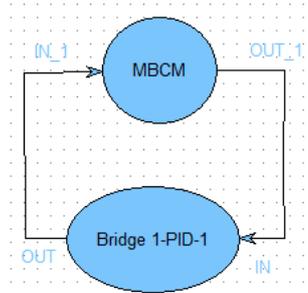
Figura 3.4 – Tela de configuração blocos Modbus Smar



Com as configurações adequadas dos parâmetros de rede MODBUS, já foi possível, a partir desse ponto, estabelecer comunicação com a BBB e fazer leitura e escrita em sua memória, por meio da rede de comunicação. O próximo passo foi fechar a malha de controle e controlar os modelos em malha fechada. Todos os processos foram ajustados com os mesmos valores de PID, mesmo em diferentes cenários, para fins de comparação de similaridade de modelagem.

Em um dos testes realizados, exemplificado na figura 3.5, o BBB operou como uma planta de controle contínua, disponibilizando valores de modelos matemáticos que simulam processo SISO para serem controlados via rede Modbus pelo CLP.

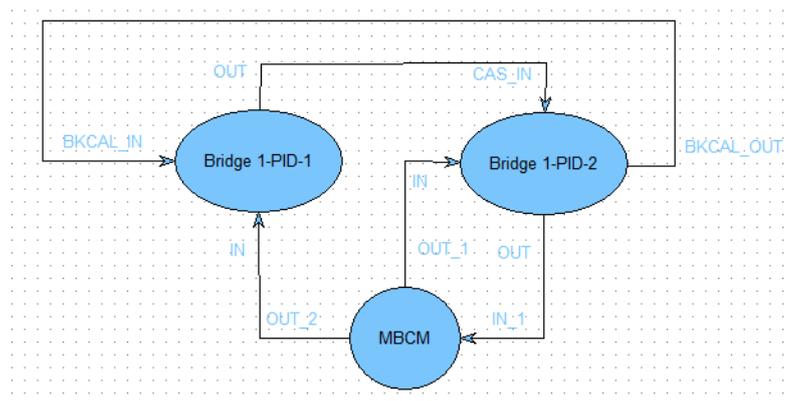
Figura 3.5 – Estratégia de controle para processos SISO em rede Modbus



O bloco MBCM lê os dados nos endereços de rede Modbus que são fornecidos pelo BBB. Nessa simulação foram usados os endereços de leitura dos registradores  *Holding Registers*, que permitem leitura e escrita de valores, tanto pelo mestre da rede quanto pelo escravo.

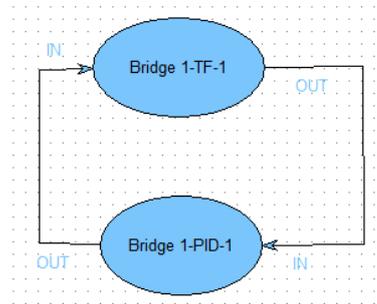
Foi executada também uma simulação de controle de processo em cascata, como mostra a figura 3.6. A diferença principal é que, na simulação de controle cascata, foram usados três registros Modbus, dois para as saídas, uma de cada malha de controle, e outra para a entrada e dois controladores PID.

Figura 3.6 – Estratégia de controle para processos SISO cascata em rede Modbus



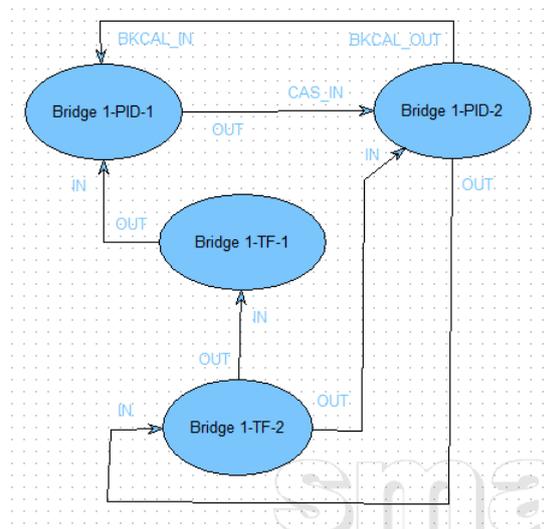
De forma similar, para título de comparação entre as ferramentas que simulam processo via hardware, foi feito também um teste no qual o próprio CLP, que dispõe de bloco funcional TF, internamente, gera os valores simulados da planta contínua e faz o controle em malha fechada do sistema. A figura 3.7 exemplifica a estratégia de controle aplicada para malha fechada de um processo de malha única e SISO.

Figura 3.7 – Estratégia de controle para processos SISO no CLP



Para o controle cascata, da mesma forma que foi simulado com o BBB e CLP em rede, também foi feita uma simulação com os blocos funcionais internos do CLP.

Figura 3.8 – Estratégia de controle cascata para processos SISO no CLP



Na secção seguinte serão apresentados os processos simulados, para as estratégias apresentadas, resultados obtidos para cada um deles e os critérios usados para a validação dos testes.

# CAPÍTULO 4

## Resultados

Para validação da arquitetura proposta, foram feitos testes de identificação e sintonia em diferentes cenários, de forma *online* e *offline* utilizando um software matemático, uma plataforma de prototipação e um controlador industrial. Os testes foram realizados tanto em malha aberta quanto em malha fechada.

Nesta seção são apresentados os testes realizados, foram feitas simulações de sistemas SISO, de primeira e segunda ordem, com diferentes atrasos de transporte e constantes de tempo. O propósito principal é a validação da arquitetura proposta e a sintonia de modelos de sistemas industriais nos ambientes escolhidos.

### 4.1. Simulações dos processos em malha aberta

Nessa etapa do trabalho, foram usados modelos com diferentes características a fim de verificar o comportamento da ferramenta desenvolvida. A tabela 4.1 apresenta os modelos dos processos escolhidos com suas respectivas aproximações de Padé.

Tabela 4.1 – Representação dos modelos Simulados

| Nº | Processo  | Aproximação de Padé  |
|----|---|--|
| 1  | $G(s) = \frac{1,8e^{-4s}}{s + 1}$   | $G_{FTMA}(s) = \frac{-3,6s + 1,8}{2s^2 + 3s + 1}$  |
| 2  | $G_1(s) = \frac{1,2e^{-22,5s}}{14s + 1}$<br>$G_2(s) = \frac{1,8e^{-4s}}{s + 1}$ | $G_{FTMA1}(s) = \frac{-1,2s + 0,1067}{14s^2 + 2,244s + 0,0889}$<br>$G_{FTMA2}(s) = \frac{-1,8s + 0,9}{s^2 + 1,5s + 0,5}$ |
| 3  | $G(s) = \frac{5}{15s + 1} e^{-2s}$  | $G_{FTMA}(s) = \frac{-5s + 5}{15s^2 + 16s + 1}$  |
| 4  | $G(s) = \frac{3}{4s + 1} e^{-0.5s}$   | $G_{FTMA}(s) = \frac{-0,75s + 3}{s^2 + 4,25s + 1}$   |
| 5  | $G(s) = \frac{2}{s^2 + 3s + 2}$   | -  |

|   |                    |   |
|---|--------------------|---|
| 6 | $\frac{4}{5s + 1}$ | - |
|---|--------------------|---|

Os processos 1, 3 e 4 são de primeira ordem, de malha única, SISO, com tempo morto. O processo 5 é de segunda ordem, SISO, sem tempo morto. O processo 6, de primeira ordem, SISO, sem tempo morto. O processo 2 é um processo de primeira ordem, SISO, com tempo morto, composto por duas equações, que são controladas em cascata.

Inicialmente, é necessário adequar as equações que não estão no formato de polinômios para inseri-las no software de simulação. Uma das formas de se fazer essa adequação é com o uso da aproximação de Padé, que basicamente aproxima atrasos de tempo por modelos racionais. Tais aproximações são úteis para modelar efeitos de atraso de tempo tais como atrasos de transporte e computação dentro do contexto de sistemas de tempo contínuo. O modelo matemático gerado pela aproximação de Padé permite atender aos padrões de entrada de dados do bloco função de transferência (TF) do controlador CLP e do programa desenvolvido no BBB, que deve estar no formato de divisão de polinômios, conforme a equação 4.1.

$$G(s) = \frac{As^2 + Bs + C}{Ds^2 + Es + F} \quad (4.1)$$

Como exemplo, considere o processo  $G(s)$  contínuo, representado pela equação 4.2

$$G(s) = \frac{1,8e^{-4s}}{s + 1} \quad (4.2)$$

Usando a equação de aproximação de Padé 2.24, apresentada no capítulo 2, e substituindo o valor de tempo morto,  $\theta$ , temos que:

$$e^{-4s} \cong \frac{1 - \frac{4}{2}s}{1 + \frac{4}{2}s} \quad (4.3)$$

Dessa forma, resolvendo matematicamente, temos:

$$e^{-4s} \cong \frac{1 - 2s}{1 + 2s} \quad (4.4)$$

Substituindo  $e^{-4s}$  na equação 4.1, temos:

$$G_{FTMA}(s) = \frac{1,8\left(\frac{1 - 2s}{1 + 2s}\right)}{s + 1} \quad (4.5)$$

Rearranjando a equação, obtemos:

$$G_{FTMA}(s) = \frac{-3,6s + 1,8}{2s^2 + 3s + 1} \quad (4.6)$$

No simulador, são inseridos os valores  $A=0$ ,  $B=-3,6$ ,  $C=1,8$ ,  $D=2$ ,  $E=3$ ,  $F=1$ .

Nesse caso, como o termo  $D$  é diferente de 1, o programa divide todos os termos por  $D$ . Dessa forma, os novos valores de  $A$  à  $F$  serão  $A=0$ ,  $B=-1,8$ ,  $C=0,9$ ,  $D=1$ ,  $E=1,5$ ,  $F=0,5$ .

Em seguida, calculando as raízes ( $R_1$  e  $R_2$ ) do denominador da função 4.6.

Aplicando o método de mapeamento de polos e zeros modificado, apresentado no capítulo 2 obtemos as equações 4.7, 4.8 e 4.9.

$$\alpha = e^{-R_1 T_s} \quad (4.7)$$

$$\beta = e^{-R_2 T_s} \quad (4.8)$$

$$\delta = \frac{\left(B - \frac{CE}{F}\right) - \left(A - \frac{DC}{F}\right) R_2}{R_1 R_2} \quad (4.9)$$

Calculando os coeficientes da equação à diferença, obtemos:

$$a_1 = \alpha + \beta \quad (4.10)$$

$$a_2 = -\alpha\beta \quad (4.11)$$

$$b_0 = \frac{A - C(D - 1)}{F} \quad (4.12)$$

$$b_1 = -\left(\frac{C}{F}(\alpha + \beta) + \left(A - \frac{DC}{F} - \delta\right)(1 + \beta) + \delta(1 + \alpha)\right) \quad (4.13)$$

$$b_2 = \frac{C}{F}\alpha\beta + \left(A - \frac{DC}{F} - \delta\right)\beta + \delta\alpha \quad (4.14)$$

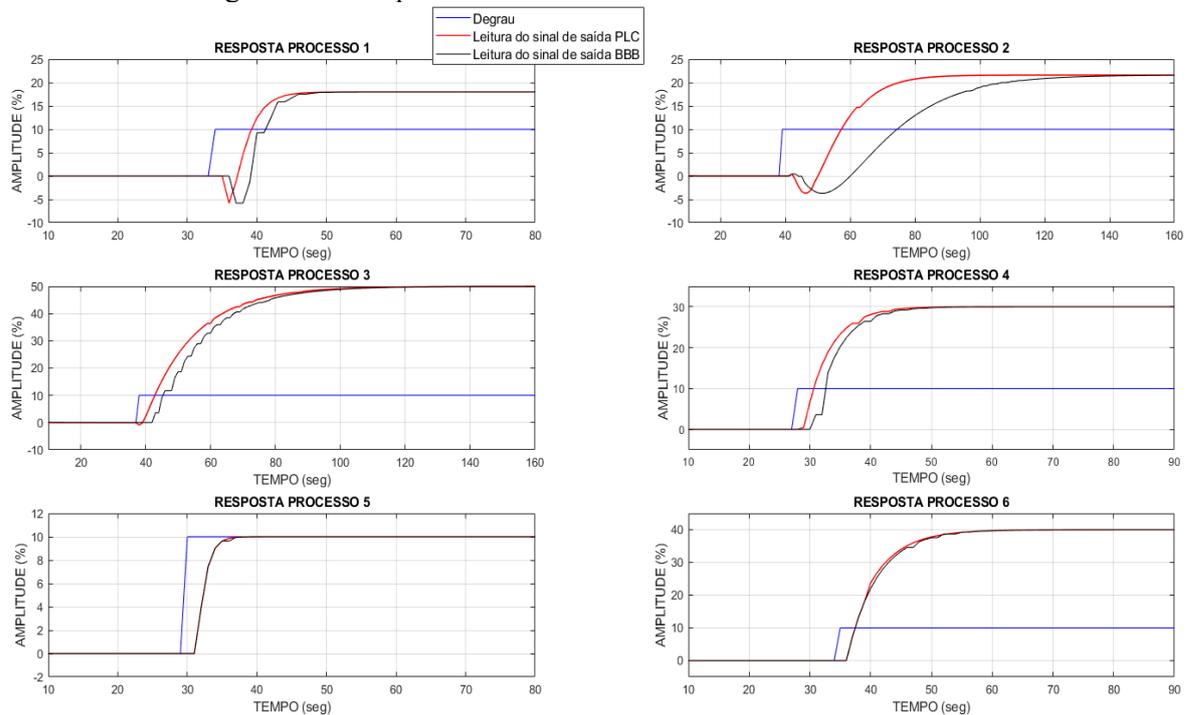
Tendo como base a equação a diferenças da planta discreta, o processo contínuo representado pela equação 4.6, pode ser reescrito de forma discreta como:

$$y(k) = 0.9744y(k-1) + 0.2231y(k-2) - 0e(k) - 0.5805e(k-1) + 1.0282e(k-2) \quad (4.15)$$

Como exemplo, o desenvolvimento matemático foi executado para o processo representado pela equação 4.2, entretanto, o mesmo método pode ser aplicado para as outras equações.

Os testes em malha aberta, obtidos com os processos da tabela 4.1, estão representados na figura 4.1. Foi aplicado um valor de 10 para a entrada de todos os processos.

Figura 4.1 – Respostas dos modelos da tabela 4.1 em Malha aberta



Analisando a figura 4.1, os modelos simulados com os dados gerados pelo BBB e pelo CLP em malha aberta, verificou-se o comportamento dos processos sem a ação de controle. Para os processos 5 e 6 o resultados foram tão semelhantes que as duas curvas ficaram praticamente sobrepostas. Nos processos 1, 3 e 4, notou-se que houve um tempo de reação aproximadamente 1 segundo maior no processo simulado no BBB, em relação ao simulado no bloco funcional TF do CLP. Para o processo 2, que foi executado em cascata, houve um resultado de tempo de reação mais divergente.

#### 4.2. Simulações dos processos em malha fechada

Para cada simulação em malha fechada, da mesma forma que em malha aberta, também foram executados dois testes. Um teste utilizou o recurso de função de transferência simulada em bloco de funções no CLP, e o outro, simulou a função de transferência dentro da Beaglebone Black, por meio de rede de comunicação Modbus TCP. Ambos os processos foram controlados pelo controlador PID do CLP.

Os parâmetros do controlador, utilizados para a elaboração dos testes dos modelos em malha fechada estão representados na tabela 4.1.

TABELA 4.2 – Parâmetros para os controladores PID

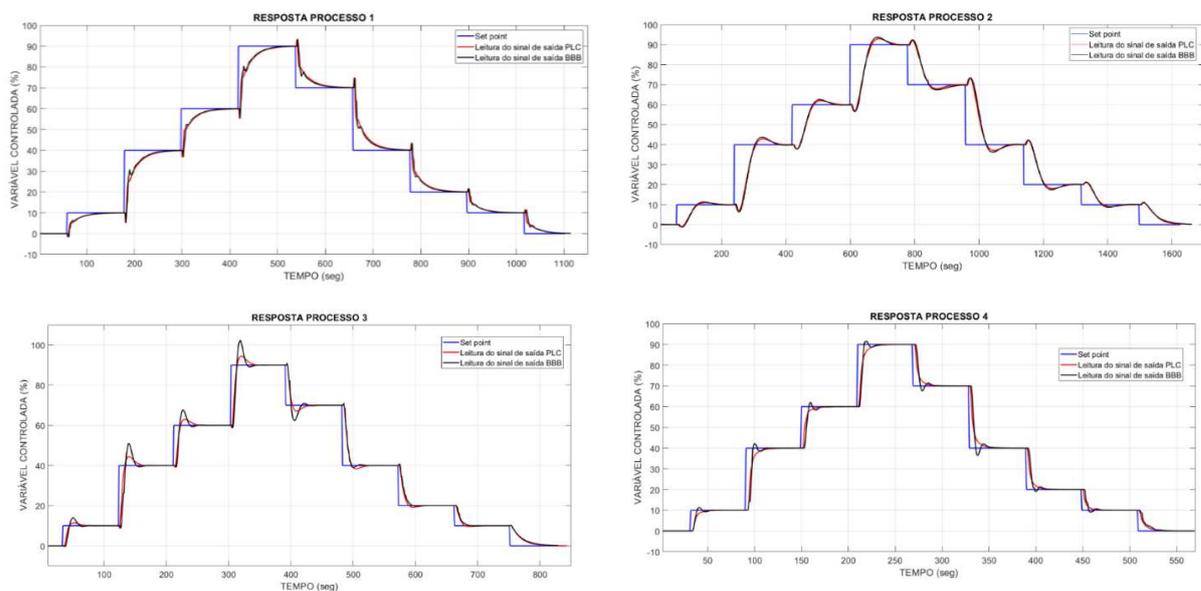
| Nº  | Kp   | Ti   | Td     |
|-----|------|------|--------|
| 1   | 0,25 | 8,33 | 0,6    |
| 2.1 | 0,5  | 0,8  | 0,1875 |
| 2.2 | 0,25 | 8,33 | 0,6    |
| 3   | 0,5  | 9    | 0,8    |
| 4   | 0,35 | 5    | 0,16   |
| 5   | 0,4  | 3    | 1      |
| 6   | 0,3  | 10   | 0,5    |

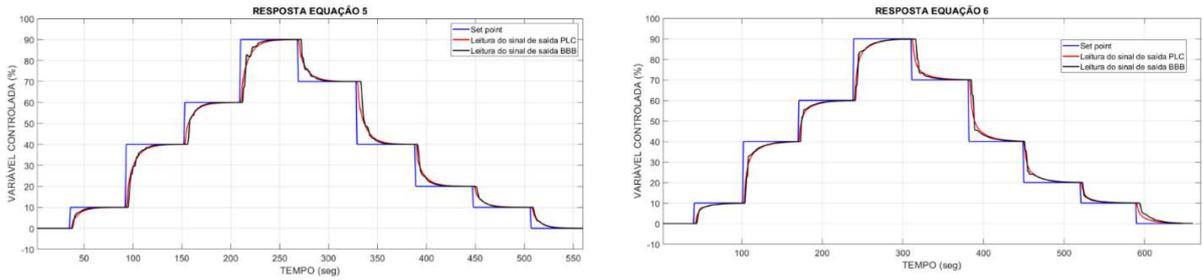
Os números da primeira coluna da tabela 4.2 correspondem aos processos listados na tabela 4.1. Os valores dos parâmetros  $K_p$ ,  $T_i$  e  $T_d$  foram obtidos por meio de técnicas de sintonia propostas por Z&N, apresentada no capítulo introdutório deste trabalho, e por meio de ajustes finos para obtenção de melhores respostas de controle. A equação PID usada para a simulação no MATLAB® foi a PID padrão (ISA), apresentado na equação 2.51 do capítulo de revisão bibliográfica.

Todos processos foram sintonizados com os mesmos valores de PID, mesmo que em diferentes cenários. Após a realização de cada teste, os arquivos de texto foram salvos e posteriormente usados para plotagem dos gráficos e validação dos resultados.

Os testes em malha fechada foram executados em todos os modelos apresentados na tabela 4.1, variando o valor do *setpoint* dos controladores de 0 a 90%. A figura 4.2 apresenta os gráficos obtidos nesse processo de simulação, na ordem de modelos apresentada na tabela 4.1.

Figura 4.2 – Gráfico de resposta em malha fechada

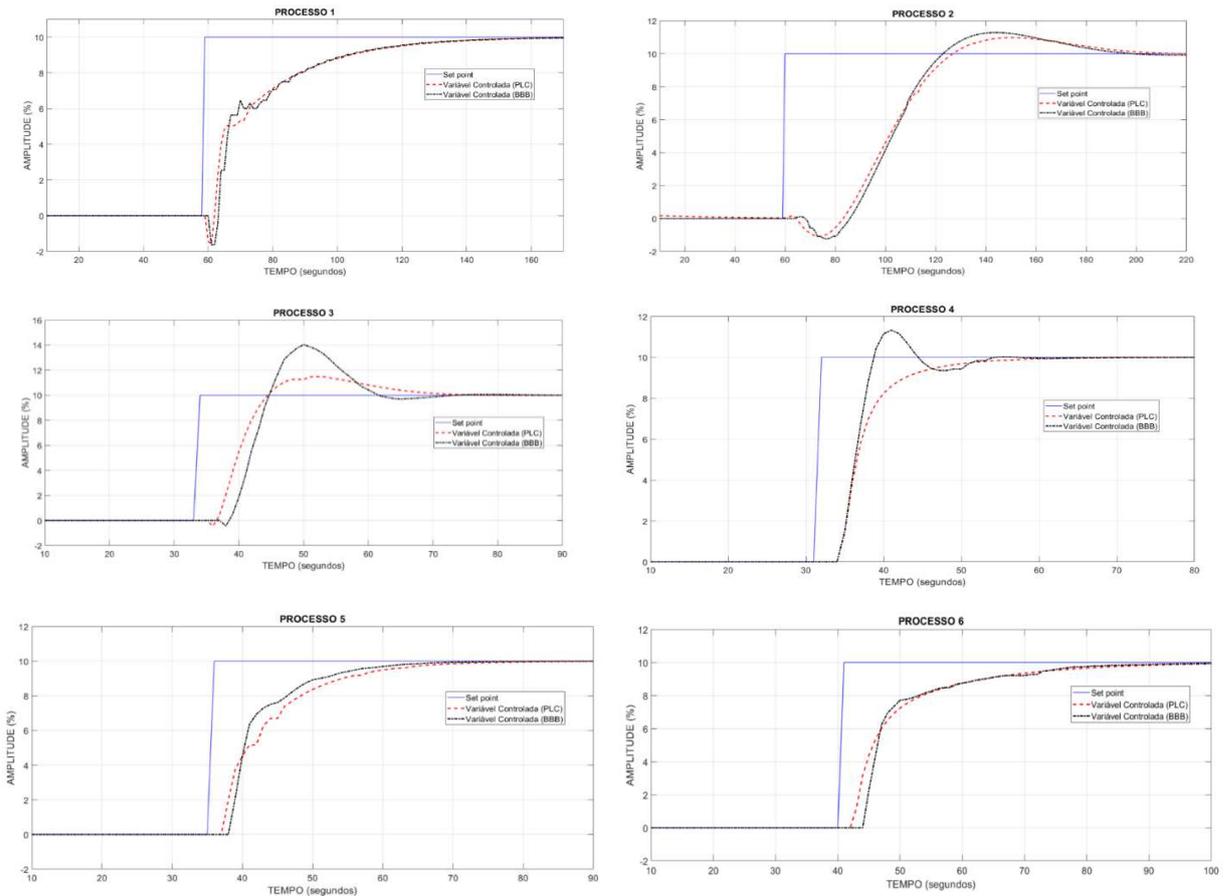




A linha azul representa o Set Point (SP), e a linha vermelha a variável de controle (CV) obtida pelos dados gerados no bloco TF do CLP. A linha preta representa o CV que obteve os dados gerados pela Beaglebone Black.

Para melhor visualização e análise dos resultados, as curvas de simulação geradas pela variação do *setpoint* nas plataformas BBB e CLP foram aumentadas na etapa de 0 a 10% da amplitude e estão representadas na figura 4.3.

Figura 4.3 – Gráfico de resposta ao degrau em malha fechada



O valor do SP foi representado na linha azul contínua. A variável controlada com dados gerados internamente no CLP é apresentada em vermelho pontilhado e aqueles gerados pelo BBB na linha preta traço-ponto.

Nos processos 1, 2 e 3, observa-se que no instante em que o degrau é aplicado a variável CV do CLP responde imediatamente, apresentando inicialmente uma pequena parte negativa, para então iniciar sua fase de subida. A resposta é esperada e tem essa característica devido ao método de Padé aplicado, que previne o aparecimento do tempo de atraso constante do processo na resposta funcional e modifica a formatação do modelo de processo para ajustar o padrão de função de transferência usado no CLP e no BBB. Da mesma forma, no processo simulado por meio do BBB, a resposta também apresenta inicialmente componente negativo antes de iniciar o passo de elevar a variável.

Nos processos 3 e 4, podemos observar que o sinal coletado do controle do modelo da BBB apresentar maior sobre sinal (*overshoot*) comparado com o coletado no CLP, apesar de os dois ambientes terem sido ajustados com os mesmos valores de PID.

Os processos 1,5 e 6 foram os que apresentaram comportamentos em resposta ao degrau mais aproximados.

A principal diferença observada entre as duas plataformas é que o sinal amostrado da BBB apresenta maiores oscilações em relação ao CLP. Embora os gráficos sejam ligeiramente diferentes, uma análise baseada em critérios de desempenho pode ser mais relevante para determinar a adequação dos resultados obtidos.

### 4.3 Análises de desempenho

Para a análise das respostas obtidas no processo de simulação, foram avaliados os critérios de desempenho: Estabilidade, Tempo de atraso, Sobressinal (*Overshoot*), Tempo de subida (*Rise Time*) e Tempo de ajuste (*Settling Time*). A variação do set point aplicada e os ajustes do controlador foram os mesmos em ambas as plataformas, a fim de simplificar a aplicação dos critérios de comparação. A tabela 4.3 apresenta os Critérios de Desempenho obtidos em cada um dos processos. A numeração de cada processo segue a sequência apresentada na tabela 4.1.

Tabela 4.3 – Critérios de desempenho dos modelos Simulados

| Processo/<br>Teste | Critérios de Desempenho |                     |    |                      |                      |                          |
|--------------------|-------------------------|---------------------|----|----------------------|----------------------|--------------------------|
|                    | Estabilidade            | Tempo de atraso [s] | de | <i>Overshoot</i> [%] | <i>Rise Time</i> [s] | <i>Settling Time</i> [s] |
| 1.1 (CLP)          | Sim                     | 8                   |    | -                    | 40                   | 62                       |
| 1.2 (BBB)          | Sim                     | 8,5                 |    | -                    | 40                   | 62                       |
| 2.1 (CLP)          | Sim                     | 43                  |    | 9,8                  | 32                   | 119                      |
| 2.2 (BBB)          | Sim                     | 44                  |    | 12,8                 | 27                   | 116                      |
| 3.1 (CLP)          | Sim                     | 7                   |    | 14,8                 | 5,5                  | 31                       |

|                  |     |     |      |      |      |
|------------------|-----|-----|------|------|------|
| <b>3.2 (BBB)</b> | Sim | 9   | 40,4 | 4,5  | 27   |
| <b>4.1 (CLP)</b> | Sim | 5,6 | -    | 8    | 16   |
| <b>4.2 (BBB)</b> | Sim | 5,3 | 13,3 | 3,5  | 19   |
| <b>5.1 (CLP)</b> | Sim | 6   | -    | 17,5 | 28   |
| <b>5.2 (BBB)</b> | Sim | 5,5 | -    | 12,5 | 21   |
| <b>6.1 (CLP)</b> | Sim | 5,5 | -    | 21   | 34   |
| <b>6.2 (BBB)</b> | Sim | 6,5 | -    | 19,5 | 33,5 |

Em ambas as plataformas, os tempos de atraso, que é o tempo obtido para a variação de amplitude de 0 a 50% da etapa, foram muito próximos. O tempo de subida e o tempo de acomodação também apresentaram valores muito semelhantes. Todas as plataformas geraram respostas estáveis, porém os processos 2, 3 e 4 apresentaram valores de *Overshoot* consideráveis.

Na simulação do processo 2, que foi um processo em cascata composto de duas funções de primeira ordem com tempo de atraso, as respostas obtidas tanto pelo CLP quanto pelo BBB também apresentaram uma parte negativa antes de iniciar o processo de ascensão. O tempo de resposta negativa foi maior que o apresentado no processo 1 e adequado ao esperado para as equações aplicadas no segundo teste. Como no primeiro teste, o processo 2 apresentou oscilações, mas em menor grau. Essa menor oscilação dos valores de CV obtidos pode estar associada à menor influência dos fatores de atraso de leitura apresentados pelo BBB nesse cenário, uma vez que o processo 2 tem como características valores muito maiores de atraso e constante de tempo.

Mesmo utilizando funções de transferência iguais em ambos os experimentos, os valores obtidos pela BBB apresentaram oscilações que não foram demonstradas na simulação pela TF gerado no CLP. Este fenômeno pode ser explicado por aspectos relacionados às condições de amostragem e tempo de atraso na comunicação de transmissão, típico no sistema de controle de rede. Assim, a diferença entre os dois ambientes está amplamente relacionada à característica de cada plataforma, uma vez que o tempo de leitura e o determinismo afetam diretamente o comportamento da variável controlada.

#### 4.4 Validação e análise final dos resultados

A validação de modelos deve, preferencialmente, levar em conta o uso pretendido para o modelo. A presente seção considera especificamente o caso de modelos que serão utilizados para aplicações em malha fechada, tais como projetos de sintonia de controladores e a

simulação de malhas de controle. Em alguns casos, o modelo pode ter um bom desempenho em malha aberta, mas ser inaceitável em malha fechada (AGUIRRE, 2003).

O índice de validação a ser utilizado é o índice de percentagem da variação da saída (FIT), que representa a qualidade do modelo de 0 a 100%. O melhor modelo é aquele que tem o maior FIT. Foram avaliados os valores das simulações efetuadas em malha aberta e malha fechada em dois ambientes de simulação, pela BBB e por bloco funcional de FT dentro do CLP, ambos em comparação com o modelo calculado pelo software Matlab.

Como os dados do CLP foram coletados via software supervisorio, a comparação entre as curvas foi feita retirando o tempo de atraso de 1 segundo de leitura, característico para a leitura dos dados pelo OPC da Smar. Os valores obtidos estão representados na tabela 4.4.

Tabela 4.4 – Valores calculados de FIT (%) para modelos simulados em Malha aberta em comparação com o software Matlab

| <b>Processo</b> | <b>BBB (%)</b> | <b>CLP (%)</b> |
|-----------------|----------------|----------------|
| <b>1</b>        | 99,0514        | 99,9999        |
| <b>2</b>        | 99,9651        | 78,0183        |
| <b>3</b>        | 99,1383        | 98,0207        |
| <b>4</b>        | 99,4381        | 99,6094        |
| <b>5</b>        | 99,9099        | 99,9999        |
| <b>6</b>        | 99,9797        | 99,8592        |

Após a avaliação da adequação dos modelos em malha aberta, foram comparados os valores de FIT para os modelos em malha fechada nos dois cenários estudados, com função de transferência na BBB e bloco de função do CLP. Os resultados dessa avaliação estão representados na tabela 4.5.

Tabela 4.5 – Valores calculados de FIT (%) para modelos simulados em Malha Fechada comparação entre controle com BBB e com CLP (Degrau de 0 a 10%)

| <b>Processo</b> | <b>FIT entre BBB e CLP (%)</b> |
|-----------------|--------------------------------|
| <b>1</b>        | 94,3246                        |
| <b>2</b>        | 82,7549                        |
| <b>3</b>        | 93,1525                        |
| <b>4</b>        | 93,4650                        |
| <b>5</b>        | 95,1490                        |
| <b>6</b>        | 94,0568                        |

Somando todos os valores em percentual obtidos e dividido pelo número de processos, obtivemos em média FIT de 92,1505%. Os dados obtidos para valores de FIT em malha fechada evidenciam que, durante a execução dos testes obteve-se valores, em média, baixos de desvio

em comparação das duas plataformas testadas, o que comprova a validade do algoritmo também para identificação em malha fechada.

# CAPÍTULO 5

---

## Conclusões

A grande vantagem de usar softwares de simulação é a possibilidade de explorar diferentes cenários, técnicas de controle e analisar influências causadas pelos tempos de amostragem e nos atrasos de transporte de dados com grande fidelidade, mas sem a necessidade de atuar em processos físicos. Por ser uma plataforma não proprietária tem também o benefício de poder ser usada em aplicações educacionais, permitindo aos estudantes projetar e sintonizar processos simulados de plantas industriais. A possibilidade de substituir o CLP por um controle emulado em computador ou em outra plataforma de prototipação reduz ainda mais os custos e aumenta viabilidade desse projeto em instituições de ensino.

A simulação do sistema de controle em rede pôde ser validada por meio de resultados experimentais com o protocolo Modbus embarcado e um controlador PID em simulações dinâmicas. Foram abordados dois cenários experimentais, no primeiro, tanto a função interna que simula a geração de dados da planta quanto o controle PID do processo são realizados internamente pelo CLP. Já no segundo cenário, o controle continua sendo feito pelo CLP, no entanto os dados do processo são gerados em outro dispositivo externo, com transferência de dados via rede de comunicação Modbus embarcado.

As principais diferenças entre os ambientes de simulação estão relacionadas em grande parte às características próprias de cada plataforma, sendo que, o principal fator envolvido foi o atraso em controle de sistemas em rede. Como esses atrasos estão relacionados a diferentes fatores, tais como, o não determinismo, tempo de execução dos blocos funcionais e do programa da plataforma de prototipação, condições de amostragem e tempo na transmissão da comunicação, não foi possível mensurá-los individualmente. Apesar das diferenças, as plataformas de simulação propostas nesse trabalho apresentaram respostas próximas e critérios

de desempenho bem semelhantes, o que comprova a eficácia e confiabilidade da ferramenta desenvolvida.

Como sugestões para trabalhos futuros, pode-se realizar a simulação de processos que envolvam mais saídas e entradas do controlador (MIMO), sistemas não lineares, outros tipos de controle, sinais ruidosos e outros protocolos de rede. Além disso, pode-se incluir a geração de uma interface gráfica, que permita a mudança de parâmetros do processo de forma mais simplificada.

# REFERÊNCIAS

---

AGUIRRE, L. A. (2007). **Introdução à Identificação de Sistemas, Técnicas Lineares e Não Lineares aplicadas a sistemas reais**. Terceira Edição, Belo Horizonte, Editora UFMG, 2007.

ATZINGEN, G. V. **Simulação, controle e automação de um forno tipo túnel utilizando tecnologia**. 2017. 139 f. Tese (Doutorado). Universidade de São Paulo, Pirassununga, 2017.

BAILLIEUL, J.; ANTSAKLIS, P. J. Control and Communication Challenges in Networked Real Time Systems. **Proceedings of IEEE Technology of Networked Control Systems**, v. 95, p. 09-28, 2007.

<https://doi.org/10.1109/JPROC.2006.887290>

CALDIÉRI, M.R. **Implementação do Modbus para aplicação em Sistema de Controle via rede sem fio**. 2016. 66 f. Dissertação (Mestrado) - Programa de Pós-Graduação em Engenharia Elétrica, Universidade Estadual Paulista, Bauru, 2016.

COELHO, A. A. R., COELHO, L. S. (2004). **Identificação de Sistemas Dinâmicos Lineares**. Florianópolis, Editora da UFSC, 2004.

FADAEI, A., SALAHSHOOR, K. Design and implementation of a new fuzzy PID controller for networked control systems. **ISA Transactions**, 2008.

<http://dx.doi.org/10.1016/j.isatra.2008.07.003>

FELIPE., T. A. **Desenvolvimento de uma ferramenta computacional aplicada na simulação de processos industriais**. 2014. 62 f. Dissertação (Mestrado) - Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Uberlândia, Uberlândia, 2014.

FERNADES J., R.F. **Identificação Remota de Plantas Industriais utilizando Tecnologias OPC e CYBEROPC**. 2009. 127 f. Dissertação (Mestrado) - Programa de Pós-Graduação em Engenharia Elétrica, Universidade de São Paulo, São Carlos, 2009.

FRANKLIN, GENE F.; POWELL, J. DA , EMAMI-NAEINI ,ABBAS (2010). **Feedback Control of Dynamic Systems**, Prentice Hall Press, Upper Saddle River, NJ, 2010.

GALLOWAY, B.; HANCKE, G. P. Introduction to Industrial Control Networks. **IEEE Communications Surveys & Tutorials**. 2013, v.15, p. 860-880.

<https://doi.org/10.1109/SURV.2012.071812.00124>

GUARESE, G. B. M.; SIEBEN, F. G.; WEBBER, T.; DILLENBURG, M. R.; MARCON, C. Exploiting Modbus Protocol in Wired and Wireless Multilevel Communication Architecture. In: **Brazilian symposium on Computing System Engineering**, 2., 2012, Natal. Anais... 2012, p. 13-18.

<https://doi.org/10.1109/SBESC.2012.12>

GUPTA, R. A.; CHOW, M. Y. Networked Control System: Overview and Research Trends. **IEEE Transactions on Industrial Electronics**, 2010, v. 57, n. 7, p. 2527-2535.

<https://doi.org/10.1109/TIE.2009.2035462>

J. HESPANHA, P. NAGHSHTABRIZI, AND Y. XU, A survey of recent results in networked control systems, **Proc. IEEE**, vol. 95, no. 1, pp. 138–162, Jan. 2007.

<https://doi.org/10.1109/JPROC.2006.887288>

LI, Q., RANKIN, D. J., JIANG, J. Evaluation of delays induced by foundation fieldbus H1 networks. **IEEE Transactions on Instrumentation and Measurement**, 2009.

<http://dx.doi.org/10.1109/TIM.2009.2019312>

LIANG, G. Control and communication co-design: Analysis and practice on performance improvement in distributed measurement and control system based on fieldbus and Ethernet. *ISA Transactions*, 54, 169–192, 2015.

<https://doi.org/10.1016/j.isatra.2014.08.012>

LJUNG, L. (1999) **System Identification, Theory for the User**, 2nd edition, Prentice Hall, Upper Saddle River, NJ, 1999.

Modbus Organization. MODBUS APPLICATION PROTOCOL SPECIFICATION V1.1b3. [S.l.], 2012. 1–50 p.

Modbus Organization. MODBUS MESSAGING ON TCP/IP IMPLEMENTATION GUIDE V1.0b. [S.l.], 2006. 1–46 p.

MOSSIN, E.A. **Laboratório remoto para ensino a distância de sistemas de controle distribuído**. 2007. 114 f. Dissertação (Mestrado) - Programa de Pós-Graduação em Engenharia Mecânica, Universidade de São Paulo, São Carlos, 2007.

NASCIMENTO, J.M.A., LUCENA, P.B. Protocolo Modbus. Redes para Automação Industrial, UFRN. 2003.

OGATA, K. (2010). **Engenharia de Controle Moderno**. Quinta Edição. São Paulo, Editora Pearson Prentice Hall, 2010

PADHAN, D. G., & MAJHI, S. (2013). Enhanced cascade control for a class of integrating processes with time delay. **ISA Transactions**, 52(1), 45–55.

<http://dx.doi.org/10.1016/j.isatra.2012.08.004>

PANG Y, NISHITANI H. Analysis of control interval for foundation fieldbus-based control systems. **ISA Transactions**, 2006.

[https://doi.org/10.1016/S0019-0578\(07\)60224-9](https://doi.org/10.1016/S0019-0578(07)60224-9)

PHILLIPS, C.L., H.T. NAGLE, Jr. (1994). **Digital Control System Analysis and Design**; 3rd Edition, Upper Saddle River, NJ; Pearson Education, 1994.

SMAR. PROCESSVIEW - Plataforma de Supervisão de processos. Site da SMAR, 2013.

Disponível em: < <http://www.smar.com/en/software> > Acesso em: 15 Fevereiro 2018.

SMAR. SYSCON – Software de programação de CLP. Site da SMAR, 2013.

Disponível em: < <http://www.smar.com/en/software> > Acesso em: 15 Fevereiro 2018.

SMAR. **MANUAL DE INSTRUÇÕES DOS BLOCOS FUNCIONAIS LIBRARY B**.

SMAR. Sertãozinho, p. 346. 2013.

SMAR. **MANUAL DO USUÁRIO**. SMAR. Sertãozinho, p. 646. 2013.

SMITH, C. A.; CORRIPIO, A. (2008). **Princípios e Prática do Controle Automático de Processo**. 3ª. ed. Rio de Janeiro: LTC - Livros Técnicos e Científicos Editora S.A., 2008.

SOARES, R.P. **Desenvolvimento de um Simulador Genérico de Processos Dinâmicos.** 2003. 101 f. Dissertação (Mestrado) - Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2003.

STEMMER, M. R. **Redes locais industriais: a integração da produção através das redes de comunicação.** Editora UFSC, 2010.

TRAN, H. D., GUAN, Z. H., DANG, X. K., CHENG, X. M., & YUAN, F. S. A normalized PID controller in networked control systems with varying time delays. **ISA Transactions**, 2013.

<http://dx.doi.org/10.1016/j.isatra.2013.05.005>

WADE, H. L. **Basic and Advanced Regulatory Control: System Design and Application.** 2nd edition. New York: ISA - The Instrumentation, System and Automation Society, 2004.