

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

**PAULO HENRIQUE CAMACHO**

**APLICAÇÃO DE UMA PLATAFORMA DE CLPs PARA DETECÇÃO DE  
FALHAS EM UM SISTEMA DE CONTROLE A EVENTOS DISCRETOS**

**DISSERTAÇÃO**

**CORNÉLIO PROCÓPIO**

**2022**

**PAULO HENRIQUE CAMACHO**

**APLICAÇÃO DE UMA PLATAFORMA DE CLPs PARA  
DETECÇÃO DE FALHAS EM UM SISTEMA DE CONTROLE A  
EVENTOS DISCRETOS**

**Application of a PLC Platform for Fault Detection in a Discrete Event  
Control System**

Dissertação apresentada como requisito para  
obtenção do grau de Mestre em Engenharia  
Elétrica da Universidade Tecnológica Fede-  
ral do Paraná (UTFPR).

Orientador: Prof. Dr. Wagner Endo  
Coorientador: Prof. Dr. Paulo Rogério Sca-  
lassara

**CORNÉLIO PROCÓPIO**

**2022**



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es).

Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.



PAULO HENRIQUE CAMACHO

**APLICAÇÃO DE UMA PLATAFORMA DE CLPS PARA DETECÇÃO DE FALHAS EM UM SISTEMA DE  
CONTROLE A EVENTOS DISCRETOS**

Trabalho de pesquisa de mestrado apresentado como requisito para obtenção do título de Mestre Em Engenharia Elétrica da Universidade Tecnológica Federal do Paraná (UTFPR). Área de concentração: Sistemas Eletrônicos Industriais.

Data de aprovação: 10 de Novembro de 2022

Dr. Paulo Rogério Scalassara, Doutorado - Universidade Tecnológica Federal do Paraná

Dr. Wagner Endo, Doutorado - Universidade Tecnológica Federal do Paraná

Dr. Cristiano Marcos Agulhari, Doutorado - Universidade Tecnológica Federal do Paraná

Dr. Joao Paulo Lima Silva De Almeida, Doutorado - Instituto Federal de Educação, Ciência e Tecnologia do Paraná (Ifpr)

Documento gerado pelo Sistema Acadêmico da UTFPR a partir dos dados da Ata de Defesa em 25/11/2022.

Dedico este trabalho primeiramente à Deus, que me confortou nos momentos mais difíceis e que indicou o caminho certo ao longo de todo o processo. Não foi fácil entrar no universo acadêmico. Agradeço eternamente aos meus pais, em especial minha mãe Selma, que confiou e sempre acreditou em minha evolução.

E à minha filha (princesa) Manuella, fontes inestimáveis e inesgotáveis de paixão e inspiração, sem os quais minha existência não teria sentido. Que não lhes falte saúde, paz e tranquilidade. Fica aqui minha gratidão eterna, pois sem vocês nada seria possível.

## **AGRADECIMENTOS**

Agradeço a Deus pela compaixão e a benção para concluir os anos de estudos e pesquisa, habilitando e preparando para desafios ainda maiores, me permitindo o meu crescimento.

Agradeço aos meus pais e familiares, pelo incentivo aos estudos ao longo de toda minha vida, e pela compreensão devido a minha ausência durante a elaboração desta dissertação. Sem vocês eu não seria nada.

Agradeço à minha mãe Selma que sempre acreditou em mim, me apoiando e jamais mediu esforços para fornecer as condições mais favoráveis possíveis para que eu pudesse me dedicar aos meus compromissos acadêmicos com afinco.

Ao meu orientador e ao meu coorientador, professores Wagner Endo e Paulo R. Scalasara, sou grato pela orientação e confiança depositados em minha pessoa.

Deixo registrado meu agradecimento a todos os amigos da pós que fiz durante estes 2 anos na UTFPR-CP, onde fui muito bem acolhido desde o começo.

À todos os professores e funcionários da Programa de Pós-Graduação em Engenharia Elétrica, pelo aprendizado e auxílio em todas as demandas recorrentes.

Enfim, minha gratidão à CAPES, responsável pelo financiamento.

“Pensar pequeno e pensar grande dá o mesmo trabalho. Mas pensar grande te liberta dos detalhes insignificantes”(LEMAN, Jorge Paulo).

## RESUMO

CAMACHO, Paulo Henrique. **Aplicação de uma Plataforma de CLPs para Detecção de Falhas em um Sistema de Controle a Eventos Discretos**. 2022. 56 f. Dissertação (Mestrado em Engenharia Elétrica) – Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2022.

Este trabalho consiste no desenvolvimento de um sistema centralizado de gerenciamento de falhas, que possibilita a designação de uma plataforma de controladores lógicos programáveis para o treinamento das redes neurais artificiais (RNA) como ferramenta de detecção de falhas em sistemas de controle a eventos discretos, permitindo diferentes estratégias de implementação de sistemas de controle distribuídos, arranjos de controladores, aplicação de técnicas de diagnóstico e tratamento de falhas em sistemas de automação, bem como referência para outros sistemas que possuam restrições de tempo. Quando um erro não é identificado e prontamente corrigido, todo o monitoramento de uma rede é comprometido, e os seus subsistemas se tornam vulneráveis a paradas repentinas durante a operação. Este trabalho tem a proposta do uso de uma RNA, a qual é responsável pela tarefa de modelagem de sistemas grandes e com diversas funcionalidades, sendo uma solução viável para analisar o tempo de operação de cada sistemas e a latência para tratamento de interrupções. A estratégia se aplica em modelar o sistema e identificar os processos de acordo com o tempo de operação de cada subsistemas pertencentes ao conjunto. O controlador lógico programável redundante irá atuar como um compensador, para que não haja paradas prematuras na operação, durante a manutenções no sistema principal com erro. Os resultados das simulações realizadas servem como base no estudo de caso. Também são validadas estratégias de detecção de falhas em uma planta industrial simulada através de *software*.

**Palavras-chave:** Detecção de Falhas. Eventos Discretos. Redes Neurais Artificiais. Controlador Lógico Programável. Sistema.

## ABSTRACT

CAMACHO, Paulo Henrique. **Application of a PLC Platform for Fault Detection in a Discrete Event Control System**. 2022. 56 p. Dissertation (Master's Degree in Electrical Engineering) – Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2022.

This work consists of the development of a centralized fault management system, which enables the assignment of a programmable logic controller platform for training artificial neural networks (ANN) as a fault detection tool in discrete event control systems, allowing different implementation strategies for distributed control systems, controller arrangements, application of fault diagnosis and treatment techniques in automation systems, as well as reference for other systems that have time constraints. When an error is not identified and promptly corrected, the entire monitoring of a network is compromised, and its subsystems become vulnerable to sudden stops during operation. This work proposes the use of an ANN, which is responsible for the task of modeling large systems with several functionalities, being a viable solution to analyze the operation time of each system and the latency to handle interruptions. The strategy applies to modeling the system and identifying the processes according to the operation time of each subsystem belonging to the set. The redundant programmable logic controller will act as a compensator, so that there are no premature stops in the operation, during maintenance in the main system with error. The results of the simulations performed serve as a basis in the case study. Fault detection strategies in a simulated industrial plant are also validated using *software*.

**Keywords:** Fault Detection. Discrete Events. Artificial Neural Networks. Programmable Logic Controller. System.



## LISTA DE ILUSTRAÇÕES

|   |    |
|---|----|
| Figura 1 – Representação de um neurônio artificial. . . . .   | 8  |
| Figura 2 – Trajetória de um sistema no espaço de estados, típica de um sistema a eventos discretos. . . . . | 9  |
| Figura 3 – Exemplo de Rede de Petri . . . . .   | 12 |
| Figura 4 – Diagrama Lógico de CLP. . . . .  | 15 |
| Figura 5 – Ciclo de varredura SCAN . . . . .  | 16 |
| Figura 6 – Modelo de 3 universos para descrição de falhas. . . . .  | 19 |
| Figura 7 – Estudo da plataforma de testes e simulações. . . . .   | 25 |
| Figura 8 – Ilustração da implementação da RNA. . . . .  | 27 |
| Figura 9 – Ilustração da implementação do arranjo de redundância dinâmica ( <i>hot-standby</i> ). . . . .   | 28 |
| Figura 10 – Representação gráfica na detecção com verificação de limites. . . . .                           | 30 |
| Figura 11 – Esquemático da divisão da planta em subsistemas. . . . .  | 31 |
| Figura 12 – Possíveis falhas que pode ocorrer na planta e em seus subsistemas. . . . .                      | 33 |
| Figura 13 – Sequência do processo ITS PLC “sistema de classificação”. . . . .                               | 34 |
| Figura 14 – Representação da divisão do treinamento da rede. . . . .  | 37 |
| Figura 15 – Performance da matriz de confusão. . . . .  | 39 |
| Figura 16 – Erro de entropia cruzada para RNA com 2 neurônios na camada escondida. . . . .                  | 40 |
| Figura 17 – Validação do treinamento com 2 neurônios na camada oculta. . . . .                              | 41 |
| Figura 18 – Erro de entropia cruzada para RNA com 6 neurônios na camada escondida. . . . .                  | 42 |
| Figura 19 – Validação do treinamento com 6 neurônios na camada oculta . . . . .                             | 43 |
| Figura 20 – Erro de entropia cruzada para RNA com 10 neurônios na camada escondida. . . . .                 | 43 |
| Figura 21 – Validação do treinamento com 10 neurônios na camada oculta. . . . .                             | 44 |
| Figura 22 – Erro de entropia cruzada para RNA com 15 neurônios na camada escondida. . . . .                 | 44 |
| Figura 23 – Validação do treinamento com 15 neurônios na camada oculta. . . . .                             | 45 |
| Figura 24 – Erro de entropia cruzada para RNA com 20 neurônios na camada escondida. . . . .                 | 45 |
| Figura 25 – Validação do treinamento com 20 neurônios na camada oculta . . . . .                            | 46 |
| Quadro 1 – Resumo das linguagens de programação definidas pela Norma IEC 1131-3. . . . .                    | 17 |
| Quadro 2 – Médias de tempos extraídos na execução. . . . .  | 36 |
| Quadro 3 – Dados da interação do treinamento. . . . .   | 38 |

## LISTA DE ABREVIATURAS, SIGLAS E ACRÔNIMOS

### SIGLAS

|     |  |
|-----|--|
| CLP | Controlador Lógico Programável               |
| IA  | Inteligência Artificial                      |
| IEC | International Electrotechnical Commission    |
| RdP | Rede de Petri                                |
| RNA | Rede Neural Artificial                       |
| ROC | Curva Característica de Operação do Receptor |
| SED | Sistema a Eventos Discretos                  |
| TFP | Taxa de Falso Positivo                       |
| TVP | Taxa de Verdeiro Positivo                    |

### ACRÔNIMOS

|         |  |
|---------|--|
| GRAFCET | Gráfico de Função de Comando, Transições de Etapas, do francês <i>Graphe Fonctionnel de Commande, Étapes Transitions</i> |
| OPC     | Comunicações de Plataforma Aberta, do inglês <i>Open Platform Communications</i>   |
| TCP     | Protocolo de Controle de Transmissão, do inglês <i>Transmission Control Protocol</i>                                     |

## SUMÁRIO

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>INTRODUÇÃO</b> . . . . .  | <b>1</b>  |
| 1.1      | MOTIVAÇÃO . . . . .  | 4         |
| 1.2      | OBJETIVOS . . . . .  | 5         |
| <b>2</b> | <b>FUNDAMENTAÇÃO TEÓRICA</b> . . . . .                                       | <b>7</b>  |
| 2.1      | INTELIGÊNCIA ARTIFICIAL E MODELAGEM COMPUTACIONAL                            | 7         |
| 2.2      | SISTEMAS DE CONTROLE A EVENTOS DISCRETOS . . . . .                           | 8         |
| 2.3      | MODELOS PARA CONTROLE E SIMULAÇÃO DE SISTEMAS AUTOMATIZADOS . . . . .        | 10        |
| 2.3.1    | Redes de Petri Aplicado a Sistema de Automação . . . . .                     | 10        |
| 2.3.2    | Grafcet . . . . .  | 13        |
| 2.3.3    | Mapeamento GRAFCET . . . . .   | 14        |
| 2.4      | CONTROLE LÓGICO PROGRAMÁVEL APLICADO A SEDS . . . . .                        | 14        |
| 2.4.1    | Programação Básica de Controladores . . . . .                                | 16        |
| 2.4.2    | Redes Industriais . . . . .  | 17        |
| 2.5      | FALHAS EM SISTEMAS AUTOMATIZADOS . . . . .                                   | 18        |
| 2.5.1    | Falha, Erro e Defeito . . . . .  | 19        |
| 2.5.2    | Detecção e Mascaramento de Falhas . . . . .                                  | 20        |
| 2.5.3    | Tolerância a Falhas e Redundância . . . . .                                  | 20        |
| 2.5.4    | Técnicas Adicionais de Detecção de Falhas em Sistemas de Automação . . . . . | 21        |
| 2.5.5    | Validação de Técnicas de Tolerância a Falhas . . . . .                       | 21        |
| <b>3</b> | <b>MATERIAIS E MÉTODOS</b> . . . . .   | <b>23</b> |
| 3.1      | PROPOSTA DA PLATAFORMA DE DETECÇÃO . . . . .                                 | 23        |
| 3.1.1    | Estratégias de Detecção de Falhas . . . . .                                  | 26        |
| 3.2      | PROPOSTA DE DETECÇÃO DE FALHAS E VERIFICAÇÃO DE LIMITES NO TEMPO . . . . .   | 28        |
| 3.3      | TESTE E VALIDAÇÃO DOS ALGORITMOS DE DETECÇÃO DE FALHAS . . . . .             | 30        |
| <b>4</b> | <b>RESULTADOS E DISCUSSÕES</b> . . . . .                                     | <b>35</b> |
| 4.1      | MODELAGEM E COLETA DE DADOS DO SISTEMA AUTOMATIZADO . . . . .                | 35        |
| 4.2      | RESULTADOS DA DETECÇÃO DOS TEMPOS . . . . .                                  | 36        |
| 4.3      | ANÁLISE TEMPORAL DO SISTEMA EM MALHA FECHADA DA ESTEIRA . . . . .            | 36        |
| 4.4      | TESTES DE VALIDAÇÃO DA METODOLOGIA PROPOSTA . . . . .                        | 37        |
| 4.5      | TREINAMENTO E ESTIMATIVA . . . . .   | 38        |
| <b>5</b> | <b>CONCLUSÕES</b> . . . . .  | <b>47</b> |
|          | <b>REFERÊNCIAS</b> . . . . .   | <b>49</b> |

|   |           |
|---|-----------|
| <b>ANEXO</b>  | <b>54</b> |
| <b>ANEXO A – REDE DE PETRI REPRESENTANDO O SISTEMA DE SELEÇÃO DE CAIXAS E OS ESTADOS DE ERRO IDENTIFICADOS PELA ESTRATÉGIA DE DETECÇÃO DE FALHAS. . . . .</b> | <b>55</b> |
| <b>ANEXO B – CARTA DE TEMPO REALIZADA PELO SOFTWARE FESTO © . . . . .</b>   | <b>56</b> |

## 1 INTRODUÇÃO

A abrangência da automação em cadeia produtiva eleva o nível de rendimento da indústria alcançando um patamar de constante evolução para acompanhar o crescimento exponencial do mercado consumidor. Com o avanço do Sistema de Automação Industrial (IAS), permitindo a integração de todos os níveis da produção para a elevação do desempenho e padronização das etapas por intermédio de ferramentas e decisões de gestores tornando as empresas competitivas (LUO; LI, 2014).

Os complexos industriais contêm autonomia na confecção e gerenciamento de seus processos conforme a demanda. Pesquisas apontam os impactos da automação no mercado de trabalho e realizam uma projeção transitória para que diversos postos de trabalhos sejam efetivamente substituídos por processos integralmente automatizados, cada vez menos necessitando da intervenção humana no ciclo de produção (CHADEEV; ARISTOVA, 2017).

Segundo a pesquisa realizada pelo Instituto de Pesquisa Econômica Aplicada (IPEA) a probabilidade de automação das ocupações no Brasil tem expandido de acordo com os avanços da tecnologia, existe a projeção de que boa parte dos postos de trabalho possam ser automatizadas nos próximos anos. Tal pesquisa também aponta que essas profissões com maior risco de automação apresentam uma tendência de crescimento ao longo do tempo, o que poderá resultar em um elevado nível de desemprego nos próximos anos caso os profissionais e o Estado não se preparem para esse cenário (ALBUQUERQUE *et al.*, 2019).

Conforme Chattal *et al.* (2019), as transformações da automação industrial proporcionaram uma série de vantagens sustentáveis e competitivas na indústria realizando a supervisão de processos, garantindo a integridade e saúde dos trabalhadores e executando múltiplas etapas controladas por Controlador Lógico Programável (CLP) e um sistema supervisor do tipo SCADA (Controle de Supervisão e Aquisição de Dados).

Para inserir a automação, o CLP, é o mais indicado, uma vez que, a sua arquitetura e robustez são de fácil manuseio. Este é responsável por gerenciar as etapas, entradas e saídas de sistemas automatizados. A utilização desta ferramenta é de grande relevância quando o foco é automatizar processos, facilitando a implementação de acessórios, como, sensores, atuadores, motores elétricos, interfaces, cabos e switches, permitindo a modelagem e controle dos eventos.

Recentemente, com a demanda de sistemas de automação complexos, exigindo um alto nível de controle de eventos discretos e métodos modulares de programação de CLPs, é de suma

importância realizar a investigação de métodos com inteligência artificial aplicada a processos industriais que está sujeito a falha (REFAAT *et al.*, 2021).

As aplicações industriais demandam de inúmeras variáveis para que o processo possa ocorrer com êxito. A partir de uma unidade de processamento dos dados, determina-se o rendimento da produção, influenciando conseqüentemente na otimização operacional industrial.

Em Lamb (2013), a maioria dos sistemas de controle de nível operacional para o chão de fábrica, usa I / O discreto (ou entrada / saída) nas variáveis de entrada e saída do processo. Otimizando o sistema e normalizando os dados com a norma internacional “International Electrotechnical Commission (IEC) 61131-3” (TIEGELKAMP; JOHN, 2010), oferecendo a versatilidade ao programador e no gerenciamento e leitura de todos os dados coletados do sistema, de forma empírica com a demanda produtiva e potencializando as soluções tecnológicas adotadas no controle dos sistemas de manufatura automatizado.

Os processos ocorrem para interligar variáveis contínuas e discretas em um mesmo sistema, que fazem parte da automação e controle de plantas automáticas, ou semiautomáticas, localizados na maioria das indústrias.

Júnior *et al.* (2003) define que as instalações da produção eram agrupadas por conjuntos automatizados separados, que dependiam de recursos humanos para gerenciar e integrar as partes dos processos. Com intuito de centralizar todo o processo de fabricação, foi realizada a reorganização do esquema para concentrar todo o processo por um único local, obtendo a agilidade para determinar algum tipo de falha, proporcionando um layout limpo e evitando danos aos equipamentos de alto custo.

O surgimento de falhas ocasiona a alteração do parâmetro do sistema, em decorrência de defeito apresentado no funcionamento do sistema. Define-se falha como a incapacidade da execução de uma função do respectivo sistema. Usando abordagens de controle tolerantes a falhas, o sistema pode continuar funcionando apesar das falhas. Falhas podem ocorrer em diferentes componentes do sistema, como atuadores, plantas, sensores (BLANKE *et al.*, 2006).

Neste trabalho são estudados sistemas automatizados, que têm como parâmetro essencial o tempo de intercorrência de um evento para o outro. Com intuito de aperfeiçoar métodos eficazes para detecção de falhas, ao discretizar eventos intercorrentes de variáveis do processo sem a perda de dados importantes para o funcionamento do sistema, pode ser difícil ou impossível estabelecer um modelo preciso devido à complexidade dos sistemas. Esta variação ocorre por não haver um modelo padrão ao projetar ou criar plantas industriais, que irá resultar em uma

demanda por processo a ser fabricado, diversificando alternativas na implementação de qualquer tipo de tecnologia.

A topologia empregada para solução de falhas em processos de manufatura consiste na integração de recursos de identificação de falhas através do treinamento das RNAs, ao localizar uma eminência de erro o processo transfere toda a execução do processo para o CLP de aplicação redundante para atuação imediata, no caso de uma intercorrência de falha no sistema monitorado pela rede.

Um dos pontos abrangentes da coleta de dados em sistemas com numerosos sinais de entrada e saída, se torna inviável em modelos reais, portanto, o uso de alocação destes dados em subsistemas que representa uma sequência lógica daquela parcela do processo serve para representar de forma resumida o ganho e desempenho para cada subsistema (DING *et al.*, 2019).

O gerenciamento dos dados consiste em coletar os parâmetros de um funcionamento normal de operação do sistema e guardar em uma memória, que serve como base para comparar variações durante o funcionamento, sendo possível analisar e apontar em quais subsistemas estão ocorrendo falhas. Esse processo provê a segurança necessária para analisar e localizar uma possível falha em sistemas distribuídos, tendo em vista que esta técnica irá gerar um residual que servirá como base para localização das falhas em um modelo de detecção centralizado. A geração deste conjunto de dados serve para mensurar erros ocorridos em subsistemas específicos, com os dados do parâmetro, analisar e apontar qual o teor do erro e em qual subsistema ele foi detectado. Estas falhas que ocorrem são reconhecidas como falhas distribuídas, pois não têm um padrão (DING *et al.*, 2019).

A quebra de componentes pertencentes ao processo de fabricação pode ser danosa, isso ocorre com certa frequência em sistemas automatizados por serem interligados fisicamente e digitalmente, ocasionando uma vulnerabilidade para todo o processo (ISO, 2010).

Com a fragmentação da sequência de produção, acarreta danos ao processo generalizado como parada de todo o sistema de fabricação, queda na produção, ociosidade em sistemas sem danos, atraso no tempo e geração de altos custos. Para interromper este ciclo são direcionados esforços em medidas contundentes para garantir alta disponibilidade e o uso de controladores e sistemas de comunicação redundantes. A implementação de um segundo componente instalado paralelamente e ligado à mesma malha do processo, garante que o sistema continue em funcionamento mesmo se houver uma falha no controlador principal, isto gera um custeio alto no quesito de hardware em ociosidade na planta (STREIT *et al.*, 2014).

A implementação de Redes Neurais Artificiais tem sido usada no decorrer dos anos, para tratar de problemas relacionados à classificação e, também, identificação de parâmetros, em casos que precisam encontrar um padrão na forma que ocorrem os eventos (YOO *et al.*, 2021; DAYOUB *et al.*, 2017). Ressalta-se que a utilização da síntese de processos para aplicação de aprendizagem constante torna-se uma forma viável para implementação de recursos na reorganização de faixas que estão fora da especificada pelo sistema (YOUSEFZADEH *et al.*, 2018).

## 1.1 MOTIVAÇÃO

Essa dissertação reúne a implementação de inteligência computacional para detectar falhas em ambientes automatizados, baseados em modelos reais, que irão fornecer bases de parâmetros na simulação de processamento lógico e com aplicações redundantes em sistemas de manufaturas e a demanda das plantas industriais. Supõe-se que as plantas estão suscetíveis a interferências do ambiente, a erros de programação e defeitos de cabos e sensores que estão ligados ao seu núcleo de processamento. Existem casos que necessitam trabalhar com linhas de produção flexível, que permite a reprogramação e a mudança do processo de acordo com o produto, fato que exige equipamentos de programação flexível e que seja rapidamente modificável, a fim de reduzir o tempo de inatividade e trocas de ferramentas durante substituição do projeto (CAPUTO *et al.*, 2018).

O monitoramento constante destas vias onde ocorrem o processo é fundamental, pois os possíveis erros estão localizados em avarias durante a execução, o nível de resposta a ser analisado é alta, pois as variáveis do processo ocorrem de forma analógica executando inúmeras repetições durante o funcionamento. A ideia é resolver intercorrências provenientes em malhas industriais, isto é, a partir de características extraídas dos tempos dos subsistemas, procurando-se identificar os parâmetros e padrões do modelo.

Para identificar e extrair os parâmetros de um modelo estocástico, a partir de uma planta industrial automatizada (simulada ou real), é preciso conhecer o processo de controle e de automatização, e estudar as falhas recorrentes. No caso descrito aqui, algumas características são analisadas para modelar os defeitos no sistema, desejando-se obter os parâmetros regulares de operação. Esse problema não é trivial, pois a ocorrência dos eventos não é estimada de forma previsível, portanto na maioria dos casos acontecem de forma aleatória desencadeando uma parada prematura de todo o conjunto. Por isso, são utilizadas neste trabalho as Redes Neurais



Artificiais.

Redes Neurais Artificiais são soluções viável para o tratamento de problemas de classificação e identificação de parâmetros, com o foco na versatilidade em trabalhar com uma alta demanda de dado, o processamento dos mesmos em conjunto com Aprendizado de Máquina que se agrupam como ferramenta de identificação viável em situações que não podem integrar novas informações com modelos treinados (CHUPONG; PLANGKLANG, 2019).

A aplicabilidade deste sistema pode ser ampla, pois incorpora ferramentas necessárias para análise e implementação de uma gama de processos, que tem como variável o tempo, para localização e tratamento de falhas. Este sistema atuaria no Controlador Lógico Programável, que tem como principal função o gerenciamento dos processos. Como atualmente as plantas industriais são projetadas para serem interconectadas entre os diversos ramos de um único processo, o sistema inteligente agrega informações que subsidiam para o remanejamentos das funções que eram executados no controlador principal para um secundário ligado a mesma rede, prevenindo a parada do processo.

## 1.2 OBJETIVOS

Propor e validar técnicas utilizando inteligência artificial para o gerenciamento de atuação do CLP principal para o redundante, a configuração do esquema será redesenhado para sobrepor o controlador com falha por uma unidade de processamento estável, possibilitando a continuidade da produção quando houver uma falha atuando no CLP principal. Permitindo a redistribuição das atribuições de um processo para outra unidade de processamento, como método de localização dos defeitos relacionados ao tempo de operação dos subsistemas, problemas provenientes de falhas recorrentes em sistemas semi-automatizados ou integralmente automatizados, gerando uma base de dados na identificação e classificação dos acertos e erros da rede.

Foi desenvolvida uma metodologia para avaliação e comparação dos resultados obtidos e algumas ferramentas, como as curvas de aprendizagem e matriz de confusão. A utilização das Redes Neurais Artificiais (RNA), irão nortear para realizar a análise (verificação e validação) do processo. Estes sistemas irão atuar em eventos discretos, operando diretamente na implementação dos testes e análises das metodologias de detecção de falhas.

A dissertação está estruturada da seguinte maneira: no Capítulo 2 é abordada a revisão bibliográfica dos assuntos referentes ao trabalho de pesquisa, com estudos sobre simulação de

sistemas e hardware aplicados para automatização em ambientes físicos, conceitos de automação de manufatura, apresentação do CLP, Sistemas a Eventos Discretos (SED), falhas em sistemas automatizados, modelagem e Inteligência Artificial (IA). Em seguida, no Capítulo 3, serão apresentados a metodologia utilizada na identificação de falhas a partir da parametrização do tempo discretizado usando as RNAs na extração de variáveis em decorrência do tempo, e a realização de testes e ensaios necessários para a validação da tecnologia, o estudo construtivo para abordar o sistema ideal que serão realizados os testes e simulações de falhas apresentados no capítulo seguinte. O Capítulo 4 trata dos resultados e discussões relacionado ao tema e apresenta uma modelo de rede eficiente para execução do projeto. As conclusões finais e as propostas para trabalhos futuros serão abordadas no Capítulo 5.

## 2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção serão abordados a definição de alguns conceitos relacionados à Inteligência Artificial e Sistemas de Controle a Eventos Discretos para o total entendimento do trabalho desenvolvido. Estes conceitos serão apresentados inicialmente.

### 2.1 INTELIGÊNCIA ARTIFICIAL E MODELAGEM COMPUTACIONAL

A tecnologia em torno da Inteligência Artificial vem evoluindo constantemente, graças a aquisições e tratamentos de dados oriundos de processos e aplicações que utilizam do processamento de informações, a abrangência e inovação só foi possível pelas notáveis evoluções de técnicas e componentes utilizados para sua implementação (LO *et al.*, 2021). Uma das áreas onde se torna mais abrangente o uso deste recurso é Aprendizagem de Máquina, pois utiliza técnicas de aprendizagem constante para melhorias em sistemas que já são implementados, entregando uma melhor performance. A proposta é utilizar as RNA, um tipo de Aprendizado de Máquina.

A aplicação de redes neurais artificiais é utilizada pelo seu princípio básico, inspirada na forma de processamento do cérebro humano, sistema inteligente capaz de aprender e se desenvolver através de experiências adquiridas em situações simuladas, com objetivo em reconhecer padrões nos dados apresentados em suas entradas, além de generalizar este conhecimento para a análise de novos dados (HAYKIN, 2001; MOHANASUNDARAM, 2020).

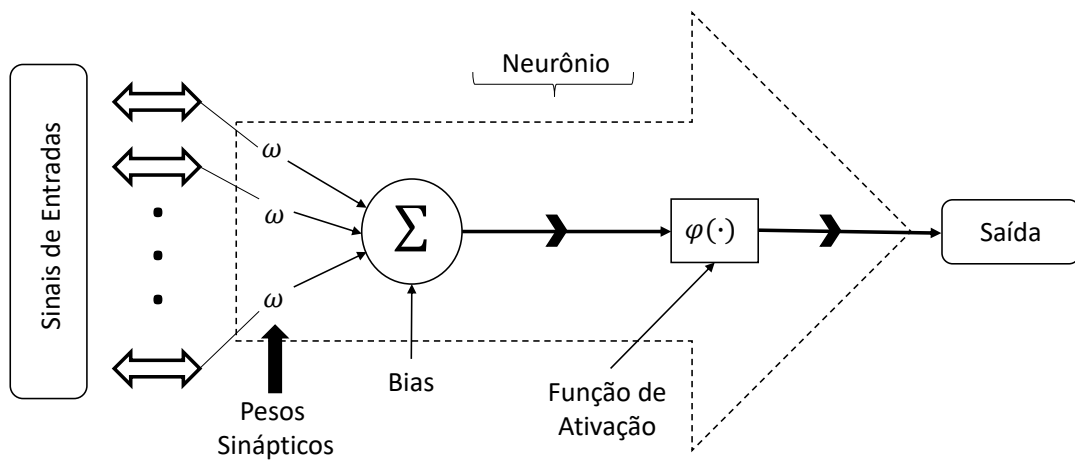
As Redes Neurais Artificiais foram apresentadas para comunidade acadêmica por volta da década de 50, que apresentou um modelo de máquina inspirada no funcionamento lógico do cérebro humano, mas com limitações da época tanto de hardware quanto de software. A ideia inicial não conseguiu sucesso na realização de nenhum tipo de operação, mas serviram de inspiração para trabalhos futuros que estruturaram modelos que os sucederam (FURTADO, 2019).

As RNAs são estruturas de processamento lógico-matemático que simulam atividades realizadas pelo cérebro humano. Trabalhando com uma ou, geralmente, várias entradas, a fim de gerar uma saída (FURTADO, 2019). Por tanto, quanto mais vezes que se utilizam as redes neurais em processos, obtendo experiência que serão guardadas para aplicações futuras, ou tratamento como base para identificar novas falhas.

O princípio de funcionamento das RNAs é aplicado em sistemas não lineares que são interligados por neurônios, que realizam a conexão entre as entradas ( $x$ ) e a função de ativação.

Os pesos sinápticos associados a estas ligações é responsável pela soma ponderada dos sinais recebidos, que transitam pela “*bias*”, atuando diretamente na elevação ou diminuição do fluxo de dados que serão restritos na entrada da função de ativação do neurônio, como apresentado na Figura 1. Destaca-se que a função de ativação aplicada neste esquema servirá como limitador, para converter as variáveis da entrada em um valor finito para sua saída, permitindo a padronização e a leitura final na identificação realizada pela rede (HAYKIN, 2001; SHEN, 2021).

**Figura 1 – Representação de um neurônio artificial.**



**Fonte: Adaptado de Furtado (2019).**

A classificação das redes neurais multicamadas ou de camada única depende do conjunto de sua arquitetura *feedforward*, como o número de unidades sensoriais (neurônios), número de nós por camada, tipo de conexão e realimentação, topologias estas usadas para agregar desempenho à rede. Os pesos sinápticos são utilizados para compor a velocidade ao conjunto, determinando o tempo de processamento das entradas do sistema, para uma solução real dos eventos, pertencendo à base comum de um neurônio artificial com pesos sinápticos proporcionais à entrada, influenciando no desempenho na RNA. Isso significa que os pesos decidem com que rapidez a função de ativação será ativada. O Bias é usado para ajustar a saída em conjunto com a soma ponderada das entradas para o neurônio, ou seja, é uma constante que ajuda o modelo a se adaptar aos dados fornecidos (FLECK *et al.*, 2016).

## 2.2 SISTEMAS DE CONTROLE A EVENTOS DISCRETOS

Os sistemas de automação modernos apresentam por característica a execução de tarefas através de um sequenciamento de operações, regidos pela recepção de estímulos e posterior

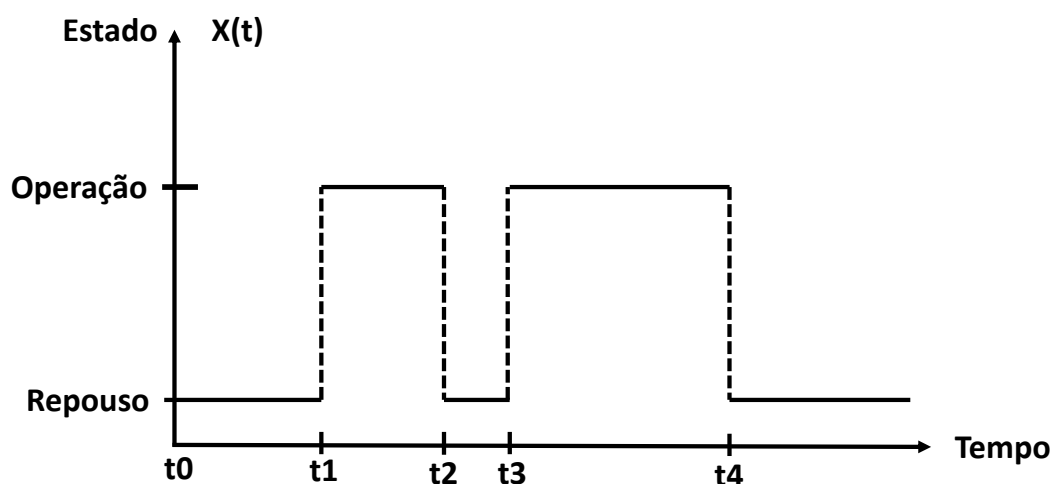
execução de ações que correspondem às próximas etapas do ciclo de execução. O uso de dispositivos digitais programáveis para realização do controle da planta pode causar considerável atraso no sinal de resposta devido ao tempo de latência do processo (CASSANDRAS; LAFORTUNE, 2008).

Em geral, os sistemas de manufatura podem ser classificados como Sistema a Eventos Discretos (SED), que são sistemas dinâmicos cuja mudança de estado ocorre em pontos discretos de tempo, em decorrência de eventos isolados, irregulares e desconhecidos (ARAÚJO *et al.*, 2009).

Estes são sistemas determinísticos quando submetido a uma excitação em um determinado estado, persiste até a mudança na variável de entrada do sistema, esta sequência de eventos apresenta sempre a mesma trajetória no espaço de estados, isto é, a sequência de estados visitados é sempre a mesma, para que estes valores pré-estabelecidos pertencentes a um conjunto discreto e o comportamento destas variáveis independe do tempo, sendo dirigido pela ocorrência de eventos assíncronos.

O comportamento de dois eventos consecutivos, pode ser representado como uma transição ou mudança de estado no sistema, de forma que sua evolução no tempo pode ser representada pela trajetória percorrida no seu espaço de estados. A sequência dos eventos pode ser ilustrada conforme a Figura 2.

Figura 2 – Trajetória de um sistema no espaço de estados, típica de um sistema a eventos discretos.



Fonte: Adaptado de Cury (2001).

No transcorrer do processo, observa-se o comportamento de um sistema que pode assumir dois estados: REPOUSO e OPERAÇÃO. A ocorrência do evento  $X(t) = 1$ , conduz o

sistema para o estado OPERAÇÃO, e a ocorrência do evento  $X(t) = 0$ , conduz o sistema para o estado REPOUSO. Nesta representação, o sistema está inicialmente no estado REPOUSO e se ocorre alguma excitação mudando a trajetória da análise, será visualmente demonstrado no gráfico de tempo gerado. Podendo assumir uma faixa de tolerância e variação pré-definidas, inclusive com a ocorrência de outros eventos ao longo do tempo ( $t_n$ ).

A simulação SED é uma ferramenta de apoio à decisão que é amplamente utilizada para resolver problemas logísticos e industriais, pode ser aplicada aos mais variados tipos de processos, em que o estado do sistema muda discretamente no tempo, não ocorrendo de forma contínua, invariante no tempo, respeitando o espaço de estados discreto e dirigido por eventos. O comportamento das operações não obedece a um padrão determinístico de entradas e saídas, mas sua transição aos eventos ocorre de forma contínua que contribui para aleatoriedade dos casos, geralmente caracterizado por uma distribuição probabilística que melhor representa o fenômeno real estudado (SAKURADA; MIYAKE, 2009).

## 2.3 MODELOS PARA CONTROLE E SIMULAÇÃO DE SISTEMAS AUTOMATIZADOS

### 2.3.1 Redes de Petri Aplicado a Sistema de Automação

As Redes de Petri são representações gráficas e lógicas da evolução dos estados de um sistema, ela atua como controladora sequencial pré-estabelecida de atuação de cada fase. A evolução de todos os eventos só ocorre se todas as etapas anteriores forem executadas com êxito. A cada evento que ocorre no sistema, é associada a uma passada de transição no modelo de Rede de Petri, de modo que a ocorrência de um evento no sistema que realiza transição do estado atual ao próximo estado é representado pelo disparo que ocorre na passagem das etapas (CARDOSO; VALETTE, 1997).

Estes processos podem ainda evoluir em sequência, de forma repetida. Essas diferentes interações entre os processos, num sistema a eventos discretos, serão modeladas, utilizando a Rede de Petri.

Conforme Vieira *et al.* (2007), a abordagem recente de estudos, revelam que os sistemas permitem considerar simultaneamente as características contínuas e discretas de seus componentes e de suas inter-relações. Segundo esta abordagem os sistemas podem ser classificados como “Sistemas Híbridos”. Duas características são apresentadas na referida obra: Autômatos (MENEZES, 2009) e Redes de Petri (CASSANDRAS; LAFORTUNE, 2008). Estes formalismos

apresentam em comum a capacidade de representar linguagens através de estruturas de transição de estados. Portanto, possuem operações formais que permitem que o modelo utilizado para representar cada componente do sistema seja utilizado para compor o modelo do sistema como um todo. Permitem, ainda, a análise do comportamento, a verificação de propriedades e a síntese do controlador.

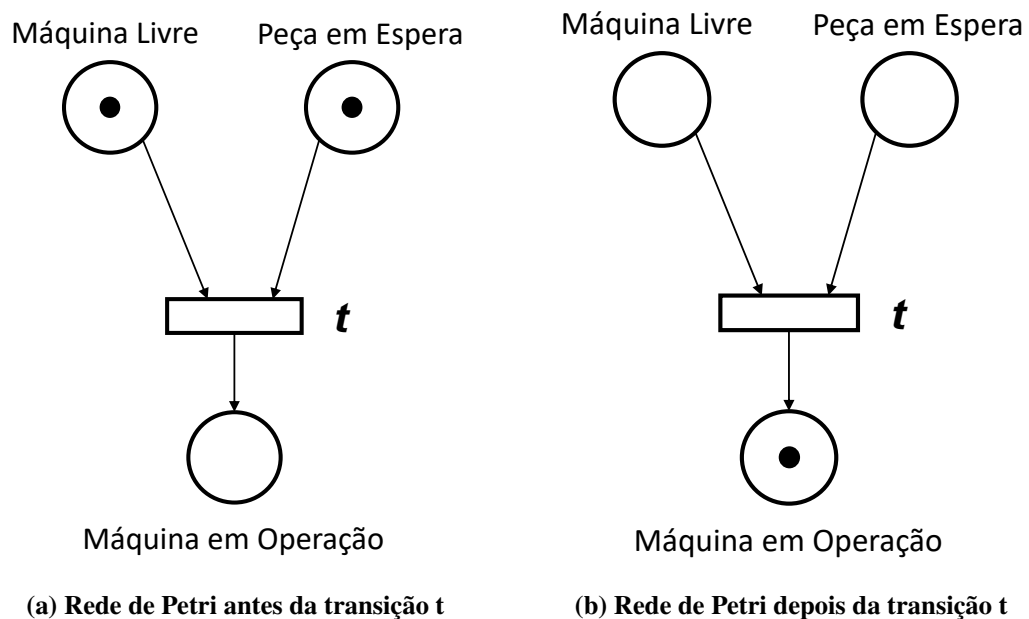
Os elementos básicos que definem um grafo de uma Rede de Petri (RdP), de acordo com (CARDOSO; VALETTE, 1997) são:

- **Estados:** representado por um círculo, pode ser interpretado como uma condição. Por exemplo, “máquina livre”, “peça em espera”.
- **Transição:** representada por uma barra ou retângulo, é associada a um evento que ocorre no sistema e altera seu estado atual, como, por exemplo, “iniciar uma máquina”.
- **Ficha:** representado por um ponto num lugar, indica que a condição associada ao lugar é verificada. Por exemplo, uma ficha no lugar “máquina livre”, indica que a máquina está livre, a condição associada ao lugar é verdadeira. O conjunto de fichas em um sistema define o seu estado corrente.

Além desses elementos, há setas indicando quais lugares se conectam com quais transições. Estas setas sempre apontam de uma transição para um lugar ou vice-versa, isso indica o fluxo das fichas quando uma determinada transição é disparada. Para a passagem de uma transição a ser disparada é preciso que seu evento ocorra, e que ela esteja sensibilizada, ou seja, seus lugares de entrada contenham fichas suficientes. Após o disparo de uma transição, os lugares de saída recebem as suas devidas fichas.

Os elementos da RdP são mostrados na Figura 3, assim como uma visualização da evolução da RdP, caso a transição definida por  $t$  seja disparada.

Figura 3 – Exemplo de Rede de Petri



Fonte: Adaptado de Cardoso e Valette (1997)

Para se modelar um SED é necessário, num primeiro momento, elaborar uma o grafo de uma RdP. Juntamente com essa descrição é necessário definir um conjunto finito dos estados que o sistema pode alcançar, o qual deve ser suficiente para descrever o comportamento, as variáveis que se deseja estudar, e um conjunto de eventos, que descrevam todas as transições entre os estados deste sistema, podendo ser definida com uma quádrupla (CARDOSO; VALETTE, 1997):

$$R = \langle P, T, Pre, Post \rangle; \quad (1)$$

Onde:

- $P$  é um conjunto finito de lugares de dimensão  $n$ ;
- $T$  é um conjunto finito de transições de dimensão  $m$ ;
- $Pre : P \times T \rightarrow \mathbb{N}$  é a aplicação de entrada (lugares precedentes ou incidência anterior), com  $\mathbb{N}$  sendo o conjunto de números naturais;
- $Post : P \times T \rightarrow \mathbb{N}$  é a aplicação de saída (lugares seguintes ou incidência posterior).

$$N = \langle R, M \rangle; \quad (2)$$



Onde:

- $N$  é o conjunto de números naturais;
- $R$  é uma RdP, definida pela quádrupla anterior;
- $M$  é a marcação inicial dada pela aplicação.

$$M : P \rightarrow \mathbb{N} \quad (3)$$

$M(p)$  é o número de fichas contidas no lugar  $p$ .

Para ilustrar a notação apresentada, considere a RdP da Figura 3. A quádrupla de tal rede é dada pela seguintes nomenclatura,  $P =$  Máquina livre, Peça em espera, Máquina em operação,  $T = t$  e os valores das aplicações de entrada e saída são dados por:  $Pre$  (Máquina livre,  $t$ ) = 0  $Pre$  (Peça em espera,  $t$ )= 1 e  $Post$  (Máquina em operação,  $t$ )= 1. A marcação inicial  $M$  é:  $M' = [1 \ 1 \ 0]$ , seguindo a ordem de  $P$ . Após o disparo de  $t$  a marcação se torna  $M' = [0 \ 0 \ 1]$ .

### 2.3.2 Grafcet

Segundo Bittar (1993), Gráfico de Função de Comando, Transições de Etapas, do francês *Graphe Fonctionnel de Commande, Étapes Transitions* (GRAFCET) surgiu da necessidade de uma normalização na automação de sistemas discretos de grande complexidade, pois ele nada mais é, do que uma ferramenta como linguagem de entrada para controle sequencial booleano. Seu uso na industrial vem se ampliando, bem como o número de pesquisadores que estudam o uso teórico desse modelo. É em particular, uma ferramenta bastante útil ao projetista na especificação de projetos de automatização.

El-Mahdy *et al.* (2021) definem o GRAFCET, como todo sistema cujas evoluções podem ser expressas sequencialmente, para análise e decomposição em etapas, permitindo visualizar o diagrama de estado, sendo uma ferramenta útil na especificação de problemas de automação. Pois quando se trabalha com sistemas integrados a complexidade das variáveis se torna inviável para modelagem e extração de dado referente ao processo.

### 2.3.3 Mapeamento GRAFCET

Para realizar o design do sequenciamento lógico do processo, são necessário regras no GRAFCET:

**Regras de sintaxe:** A alternância “etapa-transição” e “transição-etapa” deve ser respeitada qualquer que seja a sequência percorrida:

- duas etapas não devem ser ligadas diretamente.
- duas transições não devem ser ligadas diretamente.

**Início do Grafcet:** A situação inicial do Grafcet corresponde geralmente a posição “máquina parada”. Esta situação pode ser definida por uma ou mais “etapas iniciais”.

**Passagem de uma transição:** A evolução do Grafcet é feita com a passagem de uma transição, que só acontece se houver as duas condições abaixo:

- quando a etapa ou etapas imediatamente precedentes estão ativa.
- quando o resultado esperado (transição) é obtido.

**Evolução das etapas:** A passagem de uma transição provoca simultaneamente a ativação de todas as etapas seguintes e a desativação de todas as etapas anteriores. Para facilitar a descrição do GRAFCET, cada transição pode ser identificada, de preferência à esquerda da barra.

## 2.4 CONTROLE LÓGICO PROGRAMÁVEL APLICADO A SEDS

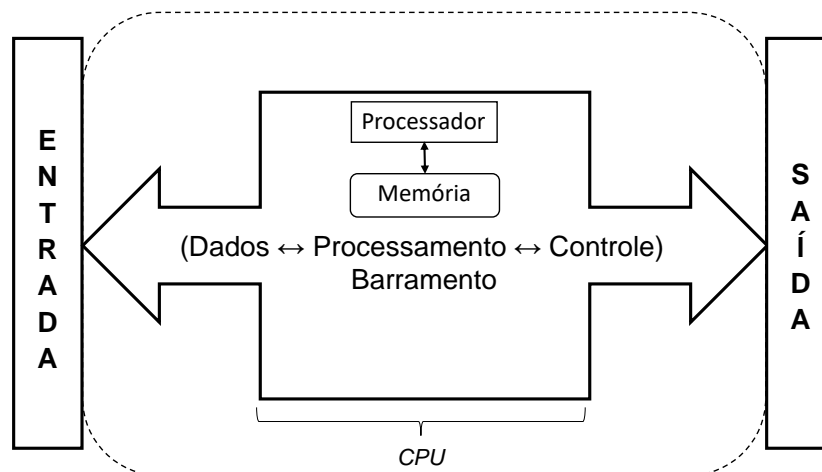
Conforme Petruzella (2014), os controladores lógicos programáveis são hoje a tecnologia de controle de processos industriais mais amplamente utilizada. Esses controladores reduziram muito a fiação associada aos circuitos de controle convencional a relé, além de apresentar outros benefícios, como a facilidade de programação e instalação, controle de alta velocidade, compatibilidade de rede, verificação de defeitos e conveniência de teste e alta confiabilidade.

A maioria dos processos de automação são gerenciados por controlador lógico programável, possibilitando a flexibilidade de operação, a fim de reduzir a dificuldade e as perdas de tempo na reprogramação da lógica de controle dos processos industriais. O modo de operação de um CLP é basicamente efetuar a varredura das entradas/saídas (E/S), que são conectadas

fisicamente com elementos do processo e atualizar as saídas conforme a lógica implementada. Essas informações de campo podem ser coletados no formato discreto ou analógico, ou ainda de entrada, ou de saída, como, por exemplo: chaves de fim de curso, transdutores de pressão, botões de impulso, partidas de motores, solenoides e etc (FRANCHI; CAMARGO, 2008).

O esquemático básico de um CLP é descrito na Figura 4. O processamento das entradas de acordo com a lógica de controle armazenada na memória faz com que haja mudanças de estado na interface de saída. O controlador também é dotado de uma interface de comunicação, que permite a interligação de diferentes CLPs em redes industriais, fato que será explorado adiante. A alimentação do sistema é realizada por meio de fontes de tensão, que podem ser tanto contínuas quanto alternadas, conforme as necessidades específicas do controlador.

**Figura 4 – Diagrama Lógico de CLP.**



**Fonte: Adaptado de Petruzella (2014).**

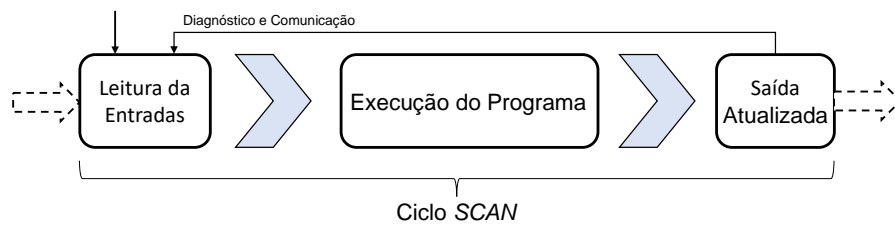
A compatibilidade do CLP com os inúmeros processos de fabricação, ocorre pela sua versatilidade no recebimento dos dados, processamento e resposta. O dispositivo que permite construir lógicas de automação flexíveis de instalação rápida ocupando pequeno espaço físico, sujeito a severos níveis de ruídos gerados pelo ambiente. É um equipamento para tarefas de automação, tem seu funcionamento baseado num sistema de microcomputador onde se tem uma estrutura de software que realiza continuamente ciclos de varredura (GHEORGHIU *et al.*, 2021).

CLP é um dispositivo de operação contínua, onde a cada ciclo de operação (chamado *scan*) o processador lê todos os dados do barramento de entradas, executa a lógica de programação definida e atualiza os estados das saídas. Com ciclos curtos, da ordem de nanossegundos, o CLP sincroniza de forma artificial os eventos de natureza assíncrona dos SEDs. A Figura 5 descreve tal ciclo. Entretanto, uma vez que há perda de tempo ao verificar cada entrada e atualizar cada

saída, o tempo de *scan* tende a aumentar conforme aumenta o número de portas de entrada e saída.

Para agilizar a execução dos programas, o CLP utiliza uma área específica da memória RAM como um buffer entre o processador e os barramentos de entrada e saída (E/S). A cada início de ciclo de programa, o processador lê todas as entradas e armazena seus estados na memória RAM. À medida que o programa é executado, os dados são lidos da memória e as operações lógicas são executadas. Os sinais de saída resultantes são armazenados no espaço reservado na memória RAM para que, ao fim do ciclo de programa, todas as saídas sejam transferidas da RAM para os canais de saída, que mantêm o estado até a próxima atualização (BOLTON, 2015).

**Figura 5 – Ciclo de varredura SCAN .**



**Fonte: Adaptado de Petruzella (2014).**

#### 2.4.1 Programação Básica de Controladores

Para realizar o controle de processos através de sistemas microcontrolados requer a programação dos controladores numa “linguagem de máquina”, ou seja, uma sequência de códigos binários que representem as instruções do programa. Entretanto, o uso da linguagem de baixo nível exige que o profissional tenha um conhecimento maior em programação. Uma vez que CLPs são ferramentas projetadas para profissionais sem grande conhecimento nesta área, a maioria dos fabricantes adotou a linguagem gráfica *Ladder Diagram*, de baixa abstração e que pode ser convertida para linguagem de máquina através de software para uso no CLP (BOLTON, 2015).

Todavia, a norma IEC 1131 – 3 estabelece, além do *Ladder*, outras cinco linguagens possíveis: *Tabelas de decisão*, *Functions*, *Sequential Function Charts*, *Instruction List* e *Structured Text*, sendo as duas primeiras linguagens gráficas e as duas últimas textuais.

O padrão de *Instruction Lists* é uma abordagem similar às linguagens de assemblers. Consiste em uma boa alternativa para tratamento de algoritmos matemáticos ou processamento

intenso de dados. Já a linguagem *Structured Text* se assemelha à linguagem Pascal, aplicável especialmente à solução de problemas matemáticos e algoritmos. As linguagens de *Sequential Function Charts* e *Functions* são ambas utilizadas para estruturar o programa principal, sendo esta última definida como uma compilação lógica de elementos de linguagens de programação (MOLINA *et al.*, 2007).

As linguagens de programação de CLP definidas pela norma IEC 1131-3 estão sintetizadas no Quadro 1.

**Quadro 1 – Resumo das linguagens de programação definidas pela Norma IEC 1131-3.**

| Classes   | Linguagens de Programação                         |
|-----------|---|
| Tabulares | Tabela de decisão                                 |
| Gráficas  | Ladder<br>Functions<br>Sequential Function Charts |
| Textuais  | Instruction Lists<br>Structured Text              |

**Fonte: Adaptado de Moraes e Castrucci (2001).**

#### 2.4.2 Redes Industriais

Para realizar o gerenciamento e comunicação dos dados gerados no controle e supervisão na manufatura integrada, exige de um controle de ramificação confiável e de resposta rápida. Desta forma, se faz necessário interligar os controladores industriais através de redes de comunicação de forma a facilitar a gestão e aumentar a eficiência dos processos (VELAGIC *et al.*, 2011).

O controle centralizado de processos utilizados no advento dos CLPs logo se mostrou ineficaz, pois demandava altos custos de instalação e manutenção particularmente por conta da distância entre a unidade de controle e os dispositivos de E/S (sensores e atuadores) (KHANDELWAL *et al.*, 2019).

Assim, com o crescimento das demandas por eficiência e produtividade, desenvolveram-se os Sistemas de Controle Distribuído, que se caracterizam pela presença de unidades de controle (CLPs) espalhados ao longo do barramento, flexibilizando a manutenção e redução dos custos de cabeamento. Estes sistemas tipicamente apresentam equipamentos de monitoramento, controle e possibilitam o uso de elementos redundantes.

De acordo com Ribeiro (1999), as variáveis de especificação de uma rede de automação

são:

- Taxa de transmissão de dados: quantidade de dados transmitidos num período de tempo;
- Topologia física da rede: arranjo físico dos equipamentos no sistema de automação (por exemplo: ponto a ponto, anel, estrela e linha);
- Meio físico de transmissão: tipo de cabeamento utilizado na rede;
- Tecnologia de comunicação: forma de gerenciamento entre os pontos de comunicação da rede (tipicamente Mestre/Escravo ou Produto/Consumidor);
- Algoritmo de acesso ao barramento: algoritmo utilizados pelos dispositivos para acessar ou fornecer informações para a rede.

A característica mais importante de uma rede de automação é o protocolo de comunicação utilizado. Estes protocolos definem o padrão operacional da rede e são um fator determinante na substituição ou inserção de um novo controlador na rede, uma vez que este deve estar apto a comunicar-se utilizando o protocolo em operação na planta. Algumas tecnologias podem operar em outros tipos de redes através do uso de gateways de ligação. Alguns dos principais protocolos utilizados atualmente são: AS-Interface, Hart, CAN, PROFIBUS, INTERBUS, FIELDBUS, Ethernet, entre outros.

## 2.5 FALHAS EM SISTEMAS AUTOMATIZADOS

Sistemas computacionais estão constantemente sujeitos a falhas, que podem ser provocadas por diversos fatores. Neste contexto, estão os sistemas de automação, que têm se tornado cada vez mais dinâmicos e complexos devido às crescentes demandas de qualidade, flexibilidade de produção, tempo de entrega, redução de custos, entre outros fatores (JÚNIOR, 2011). Reitera-se aqui o fato de o CLP, por se tratar de um conjunto de dispositivos eletrônicos capaz de processar informações de acordo com um programa, é um sistema computacional.

Uma característica particular dos sistemas de automação é o compartilhamento de informações entre os instrumentos que compõem a planta. Desta forma, o comportamento global do sistema é a resultante da composição de diversas máquinas de estado que modelam o sistema, o que gera uma inevitável explosão combinatória do número de estados, o que inviabiliza a

previsão de todos os possíveis estados de falha que o sistema possa apresentar (CARDOSO; VALETTE, 1997).

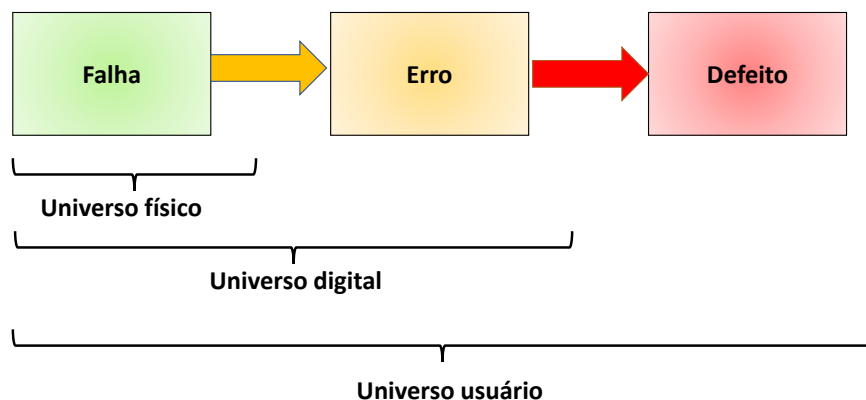
Considerando a possibilidade de falhas tanto humanas quanto computacionais e não sendo possível prever todos os estados de falha, nota-se que não é possível descartar a existência de risco de falhas neste tipo de sistema. Assim, evidencia-se a necessidade de desenvolver sistemas que reconheçam a existência de falhas e incorporem estratégias de tolerância às mesmas que resultem num nível aceitável de operação (KOREN; KRISHNA, 2020).

### 2.5.1 Falha, Erro e Defeito

Tendo-se como objetivo o atendimento das especificações de projeto de um dado sistema, são classificados como defeito o desvio da especificação que não pode ser tolerado e deve ser evitado. O sistema pode ser definido em estado de erro se o processamento posterior a partir desse estado pode levar a um defeito. Por fim, define-se falha como a causa do erro, seja física ou algorítmica (CZAJKOWSKI; PATAN, 2016; KOREN; KRISHNA, 2020).

Uma forma mais simples de abstrair este conceito é o Modelo de três Universos sugerido por Pradhan (1996), que associa falhas ao universo físico, erros ao universo da informação e defeitos ao universo do usuário. A Figura 6 apresenta o esquema deste modelo de relacionamento dos conceitos.

**Figura 6 – Modelo de 3 universos para descrição de falhas.**



Fonte: Adaptado de Pradhan (1996).

Em geral, as falhas são classificadas em:

- Falha física: falhas de funcionamento de componentes;

- Falha humana: falhas de projeto e de interação;

As principais causas de falhas são problemas de especificação, implementação, defeitos em componentes, imperfeições de manufatura, fadiga de componentes físicos ou distúrbios externos como radiação, interferência eletromagnética, variações ambientais ou também problemas de operação (CZAJKOWSKI; PATAN, 2016).

Uma falha pode ainda ser classificada conforme:

- Natureza: falha de hardware, software, de projeto, de operação;
- Duração: persistente, temporária ou intermitente;
- Extensão: local a um módulo, global;
- Valor: determinado ou indeterminado no tempo.

### 2.5.2 Detecção e Mascaramento de Falhas

A abordagem do problema das falhas em sistemas automatizados passa por dois conceitos essenciais: a detecção e o mascaramento. A detecção é a identificação de um elemento inoperante ou com comportamento inadequado no sistema, para que seja possível tal identificação, são aplicados mecanismos tais como: duplicação, comparação, testes de limite de tempo, watchdogs, testes estruturais, entre outros (CHENG *et al.*, 2021).

O mascaramento, por sua vez, é um procedimento adotado no ato em que uma falha é identificada. O objetivo deste procedimento é minimizar os danos provocados pela falha, tentando fazer com que o sistema permaneça em funcionamento de tal forma que um usuário externo não perceba a ocorrência da falha, ou ao menos o sistema seja mantido num estado seguro. Normalmente o mascaramento de falhas está diretamente ligado à redundância.

### 2.5.3 Tolerância a Falhas e Redundância

Em sistemas que exigem alta confiabilidade e disponibilidade, não é suficiente prever e remover falhas. É necessário também garantir a correta operação do sistema ou ao menos garantir que o sistema permaneça em condições seguras mesmo quando da ocorrência de falhas (CHENG *et al.*, 2021).



Discorrendo sobre o assunto, Koren e Krishna (2020) afirmam que a tolerância à falhas é um exercício de exploração e administração de redundância, que é a propriedade de possuir mais de um recurso minimamente necessário para executar o trabalho desejado. Na incidência de uma falha, a redundância pode ser explorada tanto para detectá-la como para mascará-la, mantendo assim os níveis de funcionalidade desejados. Aqui deve-se enfatizar a distinção entre detecção e mascaramento, sendo:

- Detecção de falhas: identificação de um elemento inoperante ou com comportamento inadequado no sistema;
- Mascaramento de falhas: uso de técnicas para que o sistema permaneça em funcionamento mesmo na ocorrência de uma falha.

#### 2.5.4 Técnicas Adicionais de Detecção de Falhas em Sistemas de Automação

Em plantas controladas por CLPs, falhas estão mais vezes relacionadas aos sensores, atuadores e fiação do que com o próprio CLP. Das falhas que ocorrem no CLP, as mais frequentes se dão nos canais de entrada e saída ou na fonte de energia, enquanto falhas no processador são mais raras. Exemplos típicos de falhas em sistemas de automação são os relés, quando estão com defeito os seus contatos tendem a permanecerem unidos sempre fechando o circuito, rompimentos de cabos, curto-circuitos, entre outros. Muitos CLPs possuem procedimentos de análise de falhas integrados, que realizam autotestes e exibem códigos de falhas a partir dos quais pode-se ter uma ideia da origem da falha e possíveis métodos de recuperação (BOLTON, 2015).

#### 2.5.5 Validação de Técnicas de Tolerância a Falhas

Sistemas de automação em geral são sistemas de tempo real, em alguns casos são também críticos, ou seja, sistemas onde falhas não tratadas podem causar sérios danos e perdas. Dado que, na maior parte destes sistemas, as taxas de falhas de componentes são baixas e as falhas acontecem aleatoriamente, uma forma de avaliar a eficácia das técnicas de tolerância a falhas é a injeção de falhas (WU *et al.*, 2019).

A técnica caracteriza-se pela realização de experimentos onde o próprio sistema ou um protótipo de simulação é submetido a falhas forçadas intencionalmente. O comportamento do sistema mediante as falhas injetadas é então observado e classificado, permitindo que a

confiabilidade do sistema seja avaliada e validada (KOREN; KRISHNA, 2020). Exemplos possíveis de injeção de falhas são: variação da tensão elétrica fornecida para os equipamentos, rompimento de cabos, interferências eletromagnéticas, entre outros.

A aplicação de filtro robusto realizado pelo Riera *et al.* (2011), permitiu que a comunicação entre a unidade de processamento e suas ramificações se torne mais seguras, aplicando o filtro para análise da comunicação do CLP com os seus periféricos, realizando o diagnóstico a partir de modelos existentes, garantindo o funcionamento pleno do processo.

### 3 MATERIAIS E MÉTODOS

Nesta seção são apresentadas as características do modelo do simulador para o gerenciamento de falhas proposto. São expostos os contornos do problema, ou seja, quais as tarefas a serem realizadas pelo sistema de gerenciamento de falhas e quais os tipos de erros tratados.

#### 3.1 PROPOSTA DA PLATAFORMA DE DETECÇÃO

A partir do exposto nos capítulos anteriores, tem-se que o projeto da plataforma de testes e ensaios deve considerar os aspectos de implementação de lógicas de controle aplicáveis a SEDs, utilizando controladores programáveis e viabilizando estratégias de detecção e mascaramento de falhas. Adiciona-se também o requisito de acessibilidade remota do sistema, que permite o uso de aplicações de monitoramento e gestão de informação do processo em tempo real. A arquitetura proposta será discutida neste capítulo, enquanto a próxima seção discute a montagem da plataforma e implementação dos métodos de detecção e mascaramento de falhas.

Sintetizando-se o exposto nos capítulos anteriores, a plataforma deve abranger os seguintes requisitos de projeto:

1. Controle de SEDs: Os equipamentos da plataforma devem estar aptos a receber lógicas de sequenciamento de operações e controle de processos, priorizando-se as linguagens de programação definidas pela norma IEC 1131 – 3;
2. Acessibilidade remota, supervisão e gestão de informação do processo: os equipamentos da plataforma devem dispor de infraestrutura de comunicação que permita o fluxo de dados com sistemas de monitoramento remoto da operação da planta e bancos de dados que armazenem e gerenciem os dados do processo;
3. Implementação de técnicas de detecção de falhas: os equipamentos da plataforma devem estar arrançados para permitir o uso de metodologias de detecção de falhas. No contexto das plantas industriais, priorizam-se as técnicas de comparação (processamento paralelo) e watchdogs;
4. Implementação de técnicas de mascaramento de falhas: Os equipamentos da plataforma devem dispor de infraestrutura de comunicação que permita o mascaramento de falhas.

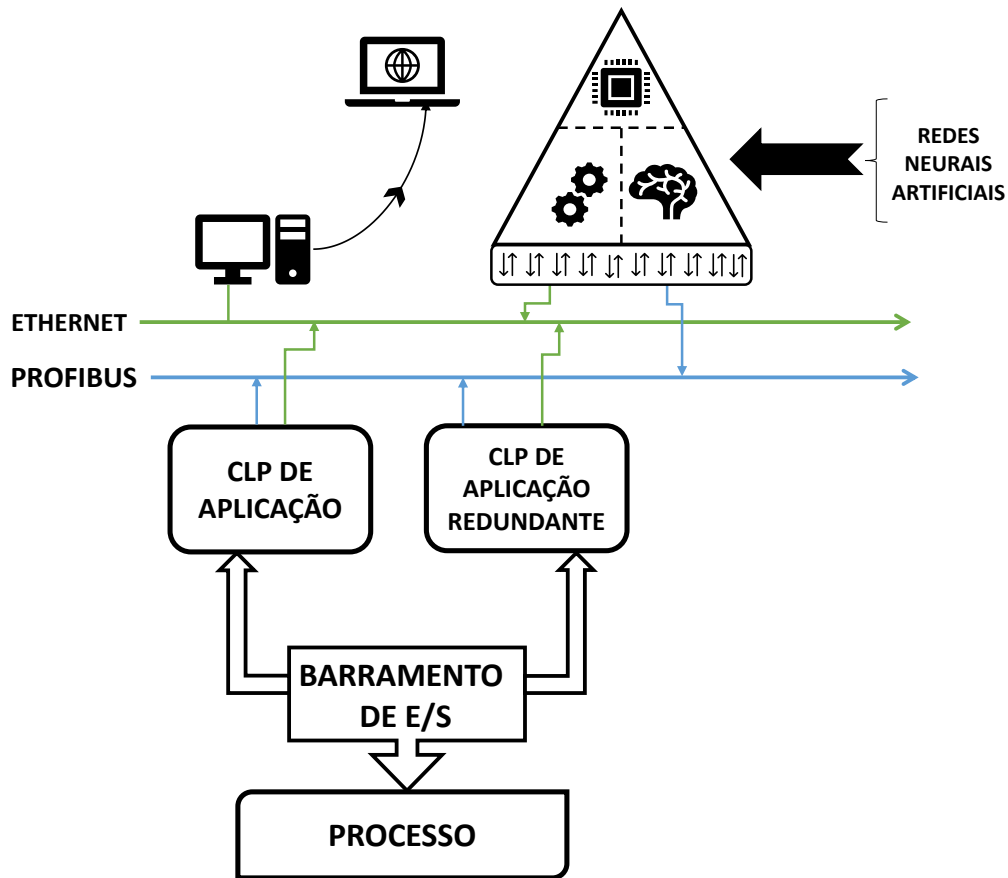
A fim de se obter os resultados desejados, é necessário um arranjo adequado dos equipamentos de forma a flexibilizar a utilização e maximizar as possibilidades de exploração das diferentes técnicas de controle, detecção e mascaramento de falhas.

Os principais componentes da plataforma são os CLPs, equipamentos típicos de aplicações industriais, cujas características construtivas adequam-se às necessidades dos ambientes industriais. A característica modular dos CLPs também é um aspecto importante, uma vez que componentes podem ser adicionados ou removidos conforme as necessidades específicas da planta e da infraestrutura da rede. Outro fator de destaque para a escolha dos CLPs é que a maioria dos fabricantes disponibiliza estruturas de temporização e comparação predefinidas, agilizando a implementação tanto das lógicas de controle como das técnicas de detecção de falhas.

Observando o exposto, o esquemático da arquitetura proposta é apresentado na Figura 7. São utilizados três CLPs, redes industriais e computadores para possibilitar o acesso remoto. Os computadores de supervisão do processo e log de falhas são elementos opcionais que podem operar no sistema se utilizando da rede Ethernet para a comunicação entre os CLPs pertencente a rede. O computador de monitoramento do sistema de identificação de falhas na RdP será utilizado apenas durante a etapa de validação das estratégias de detecção e mascaramento.

A utilização da Rede de Petri no Anexo A para modelar o sistema de distribuição de caixas, descrevendo as etapas sequenciais da operação responsável por comandar a planta piloto, tem por objetivo demonstrar de maneira didática a operação das diversas malhas de controle utilizando os mesmos equipamentos e ferramentas de configuração desenvolvidos para aplicação em controle industrial.

Figura 7 – Estudo da plataforma de testes e simulações.



Fonte: Autoria própria.

O CLP de supervisão recebe os dados dos módulos de E/S através da rede industrial e/ou do barramento de entradas e saídas do processo para executar os procedimentos de detecção de falhas através de estratégias como watchdogs e outras.

O processo em questão é formado por um sistema interligado por CLPs na mesma rede de controle, representado pela Figura 7. O sistema original consiste no CLP de Supervisão, designado para monitorar a Rede e reconhecer as falhas e defeitos da mesma, em seguida os CLP de Aplicação e CLP de Aplicação Redundante são conectados em paralelo para a mobilização imediata caso o principal venha apresentar falhas.

O intuito deste estudo é a aplicação de um método de reconhecimento de falhas, para eliminar o CLP de Supervisão, baseando na discretização dos eventos e reconhecimento de aprendizagem constante do sistema.

A rede Ethernet permite a comunicação dos CLPs com os computadores dos níveis de supervisão e gerenciamento da planta automatizada, bem como o acesso remoto ao sistema.

Esta configuração permite a configuração do Protocolo de Controle de Transmissão, do inglês *Transmission Control Protocol* (TCP), um dos protocolos sobre os quais se estabelecem as comunicações na Internet. Permite também o uso dos protocolos específicos como, Comunicações de Plataforma Aberta, do inglês *Open Platform Communications* (OPC), responsável pela comunicação industrial, através do qual é possível criar bancos de dados, históricos, gráficos de tendência, gerenciar alarmes, entre outras funcionalidades utilizando-se programas específicos para fins de automação como os softwares SCADA.

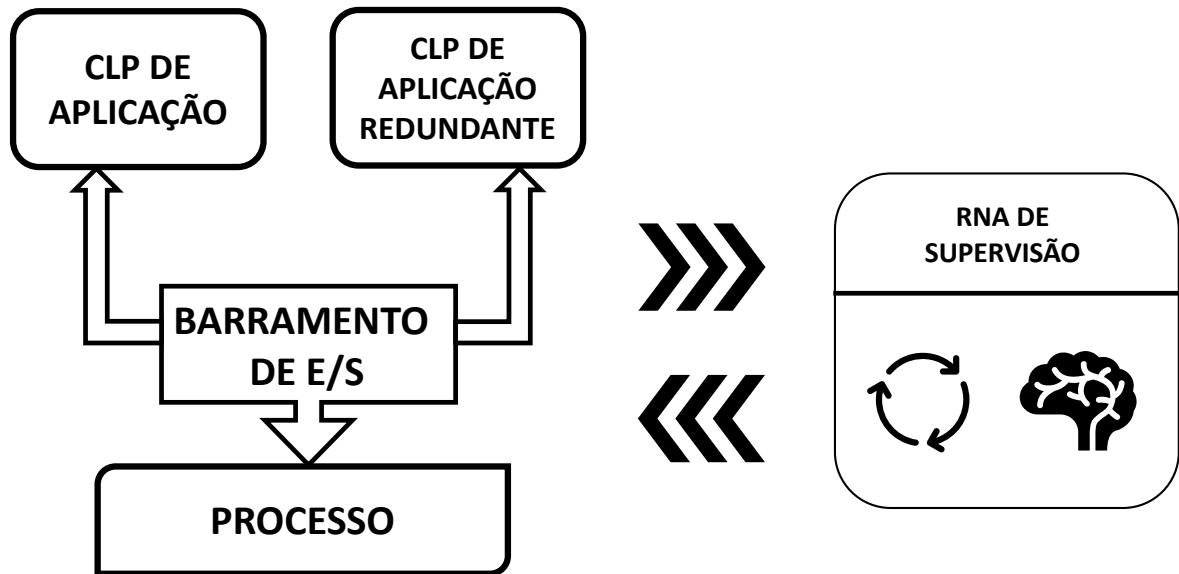
### 3.1.1 Estratégias de Detecção de Falhas

Diante das técnicas abordadas para o tratamento de falhas em sistemas de manufaturas apresentadas ao longo deste documento, nesta plataforma será implementada uma estratégia de detecção e outra de correção das falhas. Por se tratar de sistemas que dependem de variáveis externas ao seu sistema de controle o desempenho se torna relativo a erros e travamentos no processo, pois se a falha persistir a produção se torna suscetível a paradas obrigatórias para reparo, isso discorre de tempo no processo. O procedimento da detecção está focado em parametrizar e monitorar pelo tempo de atuação dos subsistemas que são relativos aos componentes da planta industrial (sensores e atuadores).

Uma vez que não é possível “consertar” um componente de forma automática, a planta terá seu funcionamento interrompido no caso de uma falha desta natureza, por outro lado, o foco do procedimento de correção de falhas é aplicar o CLP redundante, que está interligado em paralelo ao sistema para atuar como controlador principal no gerenciamento do processo. Caso haja uma falha ou um conjunto sistêmico de falhas no CLP de aplicação principal, o redundante será habilitado e controlará a planta de modo que o funcionamento não seja interrompido.

A configuração para analisar e apontar as falhas no sistema será rearranjada de forma que possa realizar o monitoramento constante do ciclo de varredura do CLP principal, para que cada sequência executada seja analisada e classifica de acordo com os tempos de cada execução. Este conjunto será implementado no treinamento da RNA de supervisão, podendo ser embarcada e implementada na configuração do processo. Entretanto, para garantir que a estratégia de correção não interfira na estratégia de detecção, optou-se por este arranjo, descrito na Figura 8.

Figura 8 – Ilustração da implementação da RNA.

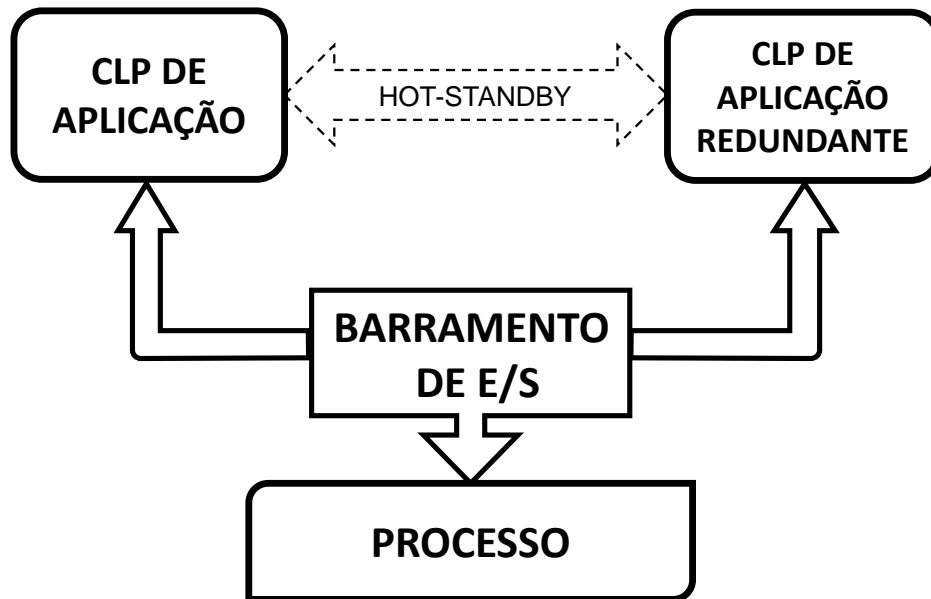


Fonte: Autoria própria.

Para certificar que a rede de supervisão não falhe na hora de identificar e apresentar que o sistema esteja em falha, as configurações dos controladores devem obedecer uma hierarquia, onde será utilizado o esquema de redundância dinâmica em regime de *hot-standby*. Este tipo de arranjo é comum em aplicações industriais, para garantir ao sistema a continuidade de operação mesmo que um dos CLPs apresente problemas.

Este conjunto será implementado no treinamento da Rede Neural Artificial (RNA) de supervisão, podendo ser embarcada e implementada na configuração do processo. Entretanto, para garantir que a estratégia de correção não interfira na estratégia de detecção, optou-se por este arranjo, descrito na Figura 9.

Figura 9 – Ilustração da implementação do arranjo de redundância dinâmica (*hot-standby*).



Fonte: Autoria própria.

A troca de um comando para o próximo, deve-se certificar de que um CLP não sobreponha o outro no controle da planta, do contrário, podem ocorrer problemas de evolução do sistema, por exemplo, o acionamento indevido de componentes causado por um CLP em falha.

### 3.2 PROPOSTA DE DETECÇÃO DE FALHAS E VERIFICAÇÃO DE LIMITES NO TEMPO

Um importante aspecto a ser abordado no contexto dos métodos mais simples e intuitivos refere-se à detecção de falhas com verificação de limites, baseando-se na medição direta de uma determinada variável e a comparação de seu valor absoluto, ou de sua tendência com as fronteiras previamente especificadas. A verificação dos valores absolutos norteia a conversão para dois valores úteis para o trabalho, que são zero e um, quando o sistema estiver ocorrendo em falha a RNA irá identificar e transferir toda a operação para o CLP redundante. A análise que servirá para discriminar se o sistema está com falha, ou não, é dado pelo tempo que discorre entre a subida do gráfico indicando que o sistema foi excitado e o instante de descida e a permanência do mesmo em zero, delimitado “*t<sub>min</sub>*” e “*t<sub>max</sub>*”, que seriam as taxas aceitáveis que o processo pode transcorrer. O funcionamento normal do sistema consiste em verificar se a variável **t** está ou não contida no intervalo:

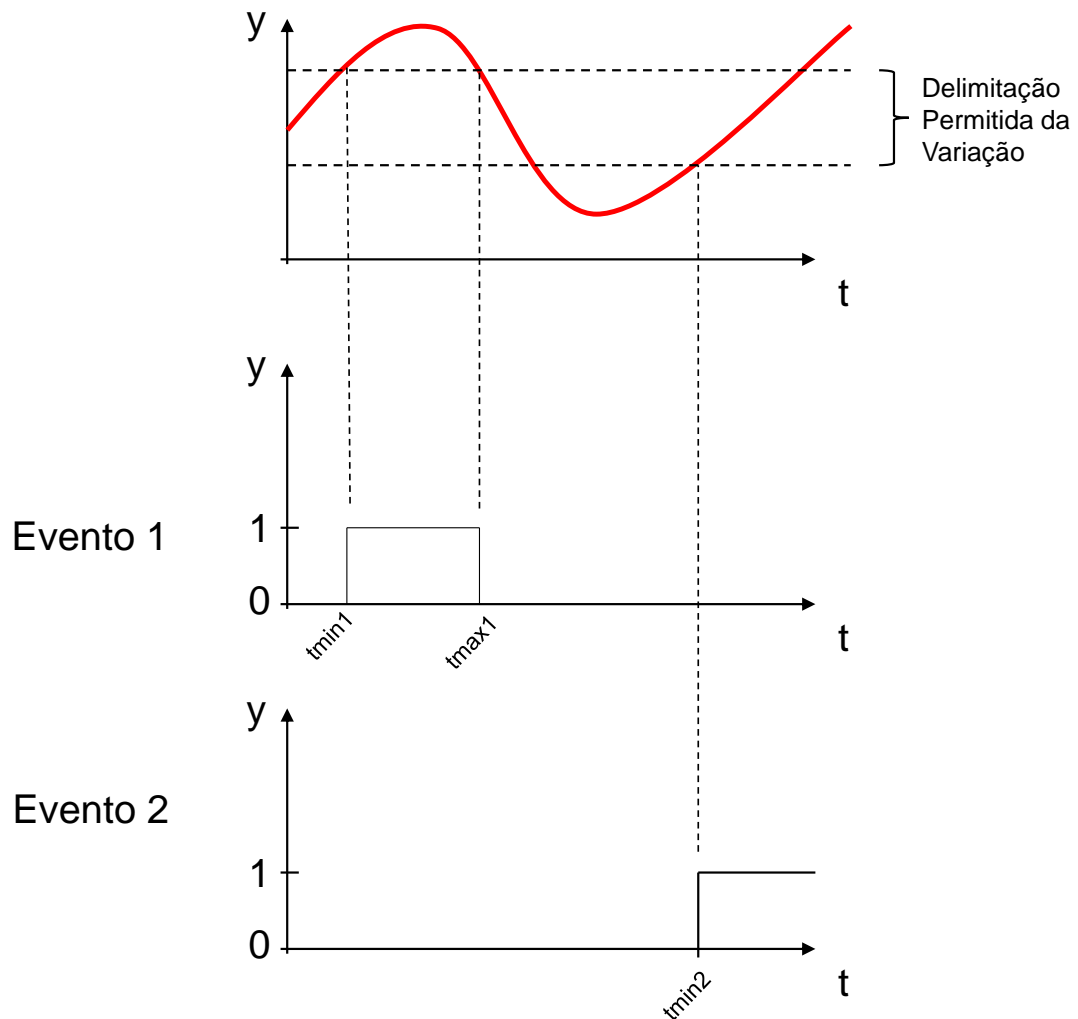
$$t_{min} < \mathbf{t} < t_{max} \quad (4)$$



Para Rebouças (2011), este tipo de abordagem considera que o processo está funcionando normalmente quando a variável monitorada encontra-se dentro dos limites de ocorrência dos eventos. No momento que a variável operacional identifica alterações nos limites de atuação de funcionamento estabelecidos, deduz-se que haverá uma falha em algum ponto do processo, ele ativa a resposta associada ao evento em questão. Esta resposta é enviada para o controlador de supervisão que sinaliza a ocorrência de falha, e atua diretamente para localizar o ponto que o sistema se encontra embargado e direciona informações para que o CLP reserve opere de forma simultânea ao erro. Esta resposta, por mais simples que pareça, envia dados que são processados e evitam a parada do processo.

A ideia inicial do estudo é selecionar a melhor rede de identificação e as melhores redes de detecção, para compor o sistema final, com o intuito de avaliar o desempenho desse sistema foram realizados testes de simulações, variando cada um dos parâmetros das possíveis falhas individualmente. Podendo ser observado na Figura 10, o janelamento entre as delimitações ocorrem para truncar a capacidade de leitura da RNA, permitindo que a rede só analise valores úteis para reconhecer o erro no processo. Isso ocorre, pois quando o evento está em função da variável tempo, pode haver infinitas variações irrelevantes ao processo e sobrecarregar os dados de leitura na identificação de falhas, recorrente de desvios dos limites pré-estabelecido ao evento.

Figura 10 – Representação gráfica na detecção com verificação de limites.



Fonte: Adaptado de Rebouças (2011).

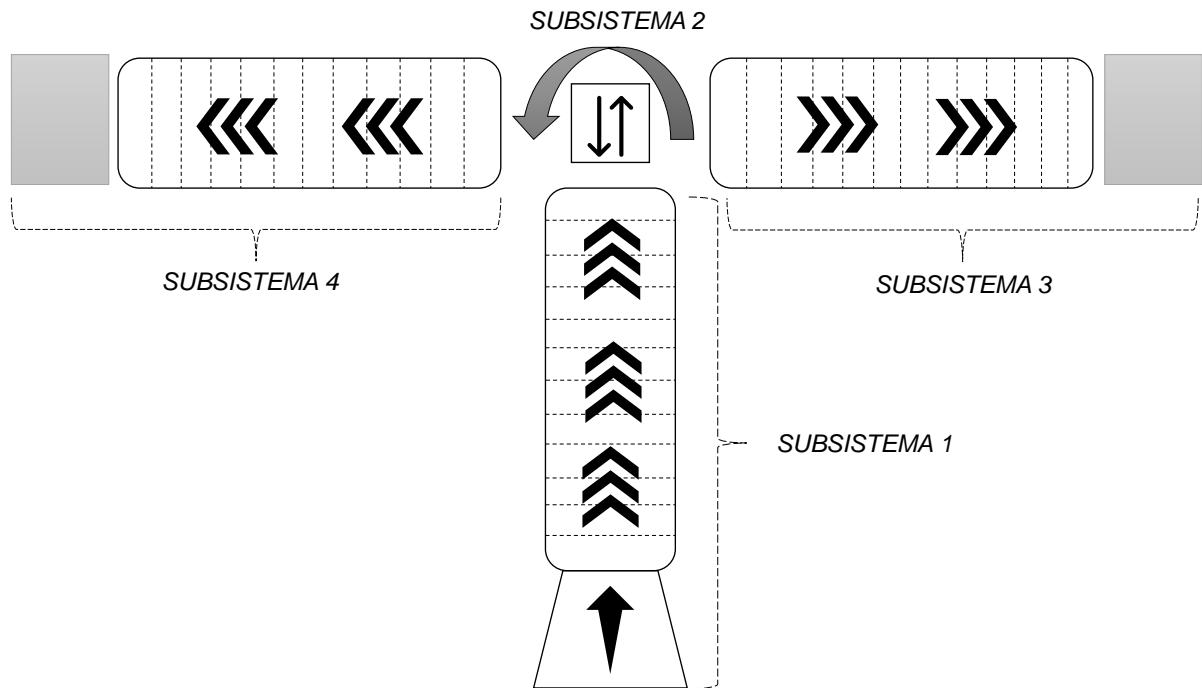
A maior dificuldade de se utilizar as representações no formato de carta de tempo Anexo B é perda de dados na delimitação do sistema nas séries temporais. Isto limita os algoritmos, as estruturas de dados e as técnicas de Aprendizagem de Máquinas que podem ser úteis ao processo, desconsiderando variáveis importantes.

### 3.3 TESTE E VALIDAÇÃO DOS ALGORITMOS DE DETECÇÃO DE FALHAS

Para realizar os diversos métodos consolidados na detecção e isolamento de falhas, a utilização de estratégias baseadas em redundância física de componentes de hardware, como sensores, atuadores e controladores, se torna inviável pelo seu alto custo orçamentário, e adquirindo complexidade para o sistema. A dedicação será voltada para as estruturas neurais de identificação do modelo e de detecção das falhas, mostrando ao final como serão realizadas as simulações.

A seqüência estudada para desenvolver as aplicações e teste em uma configuração de controle automatizado, foram esboçadas no formato de planta baixa para identificar quais componentes pertencentes e aplicados, e quantos recurso é mobilizado para desenvolver testes, aplicações e treinamento da ferramenta desenvolvida neste trabalho. Conforme a Figura 11:

**Figura 11 – Esquemático da divisão da planta em subsistemas.**



**Fonte: Adaptado de Riera *et al.* (2011).**

Para descrever a seqüência será utilizado o diagrama dos tempos extraídos da operação que controla quatro subsistemas na distribuição das caixas, a partir das dimensões de altura será designado o deslocamento da caixa para um sentido da esteira, realizando a separação de acordo com o tamanho, excitando o sensor no começo do processo. O modo de operação desta esteira é bem simples, funcionando de forma linear e respeitando a seqüência das etapas. O processo irá depender de variáveis externas que realiza a leitura das seguintes operações:

- Subsistema 01: Transporta a caixa montada para o distribuidor central circular.
- Subsistema 02: Base giratória.
- Subsistema 03: Seleciona a caixa de tamanho pequeno *P*.
- Subsistema 04: Seleciona a caixa de tamanho grande *G*.

Os sensores e máquinas elétricas atuando diretamente no sistema são os seguintes:

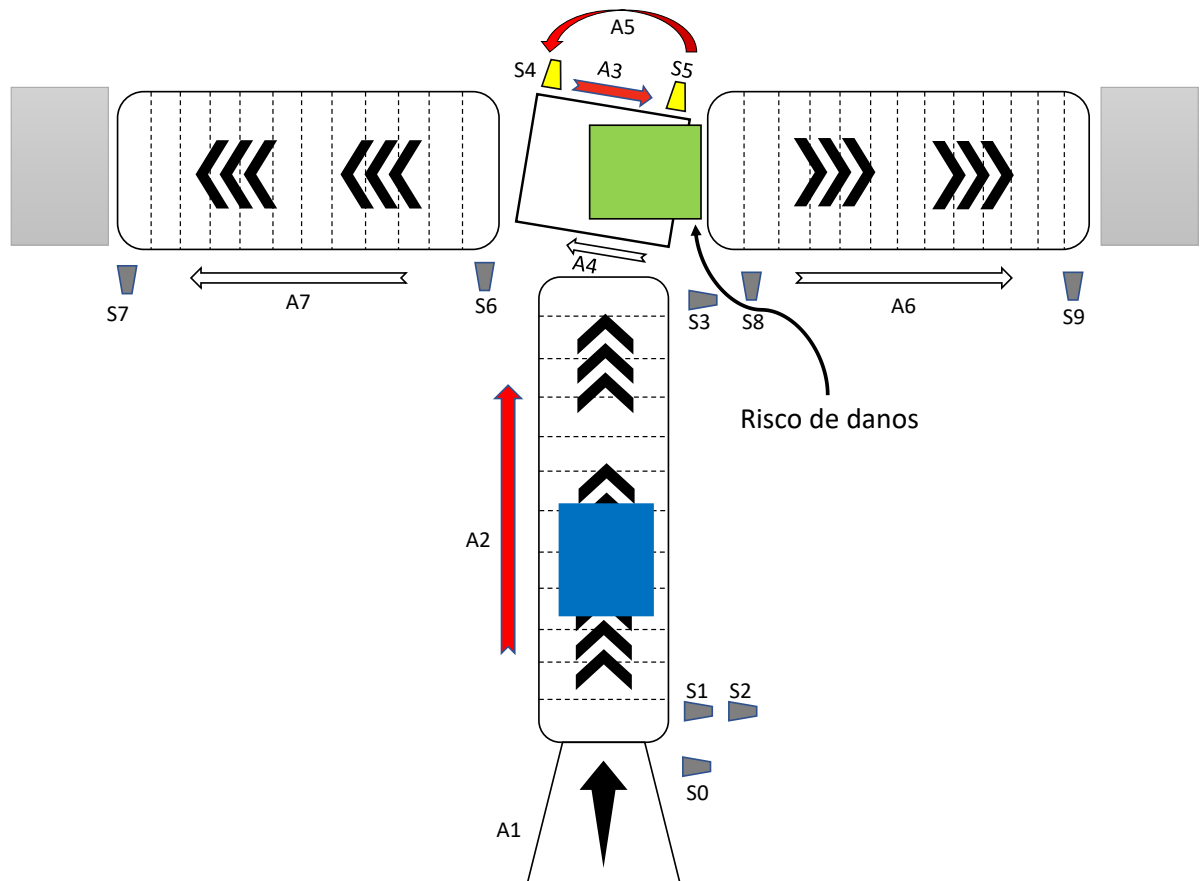
- ATUADOR 1,2,3,4,5,6,7: Responsável pela movimentação da esteira, realizando a comutação dos motores para a transição das etapas;
- Base Giratória: Empregado para distribuir entre os subsistemas, seletora de caixas;
- AT-, AT+: Indica se o atuador está em ação simples para dá avanço a esteira, ou recolhido;
- SENSOR 0: Sensor de presença da caixa na entrada da esteira;
- SENSOR 1, 2: Sensor responsável por selecionar o tipo de caixa que está na entrada da esteira principal;
- SENSOR 3: Sensor de presença para esperar a Base Giratória retornar à configuração inicial e receber a próxima caixa;
- SENSOR 4, 5: Sensor de presença localizado na Base Giratória, com o sensoramento de deslocamento de sentido;
- SENSOR 6, 7: Sensor de presença utilizado no deslocando do pallet de caixa *P*;
- SENSOR 8, 9: Sensor de presença utilizado no deslocando do pallet de caixa *G*;

Nesta configuração abordada no diagrama anterior, explicando o funcionamento normal de um processo, sem a ocorrência de erros, podemos observar que a distribuição é bem linear, seguindo uma série de etapas para que as caixas sejam separadas corretamente de acordo com as configurações do CLP que controla o sistema.

Para deixar a experiência ainda mais lúdica, demonstrando um de muitos erros que possa vir ocorrer em um processo simples como este que estamos abordando, a Figura 12 contextualiza uma sequência com erros, que possa vir causar danos à caixa e interromper o processo se houver um gargalo naquela situação da imagem. Como podemos definir a caixa da cor verde que está em sentido ao elevador das caixas *P* pertencente ao *SUBSISTEMA 3*, por uma falha no CLP que avançou etapas naquele instante acionando o retorno da base giratória para posição de pegar a próxima caixa azul, que representa *G*, está vindo em sequência na esteira principal. A princípio o sistema continua o processo como se nada tivesse ocorrido na linha de distribuição, gerando uma falha naquele ponto ou carregando para todas as próximas etapas seguintes. Normalmente o processo é supervisionado por um operador que deve interromper o processo e solicitar manutenção para aquele equipamento.

São situações reais vividas diariamente em processos industriais, que ocasiona danos aos produtos, atraso na produção, desvio na separação, quebras de equipamento na linha e mau aproveitamento de recursos humanos.

Figura 12 – Possíveis falhas que pode ocorrer na planta e em seus subsistemas.

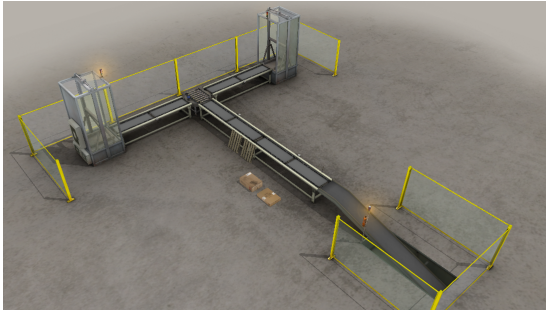


Fonte: Adaptado de Riera *et al.* (2011).

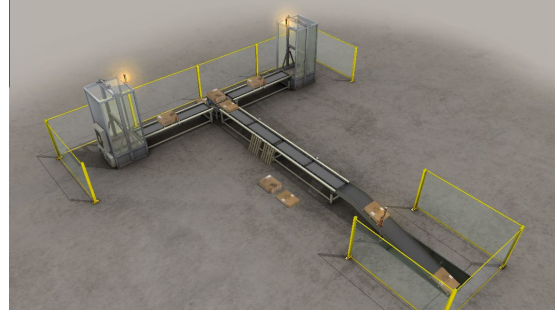
Ao realizar os testes na planta de simulação no modo automático, o CLP externo é responsável em controlar a simulação. O *software* apresenta uma característica principal que é a sua interatividade em tempo real na execução. A interatividade permite aos usuários causar disfunções que ocasiona travamentos na planta virtual.

Isto é feito através da introdução de falhas de circuito aberto ou curto-circuito nos sensores e atuadores virtuais, através da adição, remoção ou entramento de objetos durante a simulação. A Figura 13 mostra um exemplo de interatividade entre os subsistemas para distribuição de caixas. A Figura 13(e) apresenta o resultado de um gargalo na base giratória de simulação causando todo o comprometimento da sequência lógica.

**Figura 13 – Sequência do processo ITS PLC “sistema de classificação”.**



**(a) O sistema de estudo de caso embutido no ITS PLC Professional Edition**



**(b) ITS PLC “sorting system”**



**(c) Visão ampla mostrando os dois elevadores**



**(d) Visão superior da bifurcação do sistema**



**(e) Gargalo na mesa giratória**



**(f) Vista lateral indicando o elevador da caixa grande G**

## 4 RESULTADOS E DISCUSSÕES

Gil *et al.* (2002) define que a pesquisa pode integrar diversos grupos, tendo como parâmetros os objetivos que se deseja atingir com aquele estudo e os procedimentos adotados para concluí-lo, ou abrir novos ramos para contribuições futuras. Assim a pesquisa proposta neste projeto pode ser classificada como exploratória, pois tem por objetivo aprimorar a aplicação de técnicas de gerenciamento de falhas. Quanto aos procedimentos a serem adotados, esta dissertação tem caráter experimental, uma vez que é proposto, utilizar os recursos de Redes Neurais Artificiais para localização de erros em processos automatizados.

Uma descrição de como é conduzido o estudo em implementar técnicas reais na utilização de um módulo externo ao processo, para analisar e conduzir aos efeitos retrativos quanto ao surgimento de anomalias em sistemas automatizados são tratados nesta seção. São necessários experimentos que visam verificar a aplicabilidade real do modelo proposto e compará-lo com os trabalhos presentes na literatura. Realizou-se cinco teste diferentes para analisar quais obtiveram melhor desempenho para reconhecer falhas.

### 4.1 MODELAGEM E COLETA DE DADOS DO SISTEMA AUTOMATIZADO

A coleta de dados deve ser realizada de forma criteriosa, pois contém as informações necessárias para o processo, analisando e gerando os resultados a partir dos dados coletados. O sistema utilizado na implementação deste trabalho é emulado virtualmente em *software*. Trata-se de uma planta de seleção baseada em esteiras e sensores de presença, que recebe caixas de dois tamanhos, são processadas pelo CLP que distribui em duas vias de saída, realizando a separação a partir de suas características. O ambiente de monitoramento proposto contém o controlador, que gerencia as esteiras de forma que as caixas grandes sejam direcionadas para o elevador, e caixas pequenas para o outro. Para realizar a modelagem das ações previstas no sistema, foi elaborado o Quadro 2 que estão contidos os tempos, de modo, especificar a localização de possíveis falhas, a planta foi subdividida em quatro subsistemas, apresentados na Figura 11.

**Quadro 2 – Médias de tempos extraídos na execução.**

|           | <b>Caixa Baixa</b> | <b>Caixa Alta</b> |
|-----------|--------------------|-------------------|
| <b>E1</b> | 13,52 s            | 14,7 s            |
| <b>E2</b> | 12,9 s             | 12,68 s           |
| <b>E3</b> | 1,63 s             | 1,96 s            |
| <b>E4</b> | 3,45 s             | 3,4 s             |
| <b>E5</b> | 8,12 s             | 7,31 s            |
| $\mu$     | 7,92 s             | 8,01 s            |
| $\sigma$  | 4,81 s             | 5,00 s            |

O Quadro 2 traz a sequência das etapas “ $E_N$ ” em relação ao tempo de operação de cada subsistemas para contabilizar um ciclo na simulação de um processo sem ocorrência de erro, coletados de um emulador de planta automatizada com sensores e atuadores. ITS PLC Professional Edition, software que reproduz as funções e lógica de uma esteira seletora de caixa, a fim de permitir a execução de outros softwares, para retirada de dados de um processo em ambiente virtual.

#### 4.2 RESULTADOS DA DETECÇÃO DOS TEMPOS

Com base nos dados extraídos, e adquirindo elementos necessários para realizar o treinamento das RNAs, na identificação e solução, de possíveis erros que venha ocorrer no processo. No primeiro caso, buscou-se desenvolver uma rede neural que fosse capaz de detectar e identificar cada uma das falhas listadas na Seção 2.5 (Falhas em Sistemas Automatizados). A rede é responsável em realizar o treinamento a partir dos erros comumente apresentados em plataforma industrial, devendo atuar de forma instantânea para alocação do CLP de aplicação redundante, como solução ideal para não parada compulsória.

#### 4.3 ANÁLISE TEMPORAL DO SISTEMA EM MALHA FECHADA DA ESTEIRA

A automatização presente nas fábricas requer que os controladores programáveis devem ser muito rápidos, de fácil adaptação e que os seus erros possam ser analisados precisamente durante a execução do processo e detectados durante operações. Como o elemento de estudo depende de variáveis lógicas que variam no tempo, optar por selecionar as falhas de acordo com



sua ocorrência de eventos existentes no processo, coletando e analisando o tempo de operação de cada subsistema, é possível intervir no processo com o CLP redundante, que está apto e alocado em paralelo ao sistema para assumir o controle a qualquer ordem direta identificada pela rede neural.

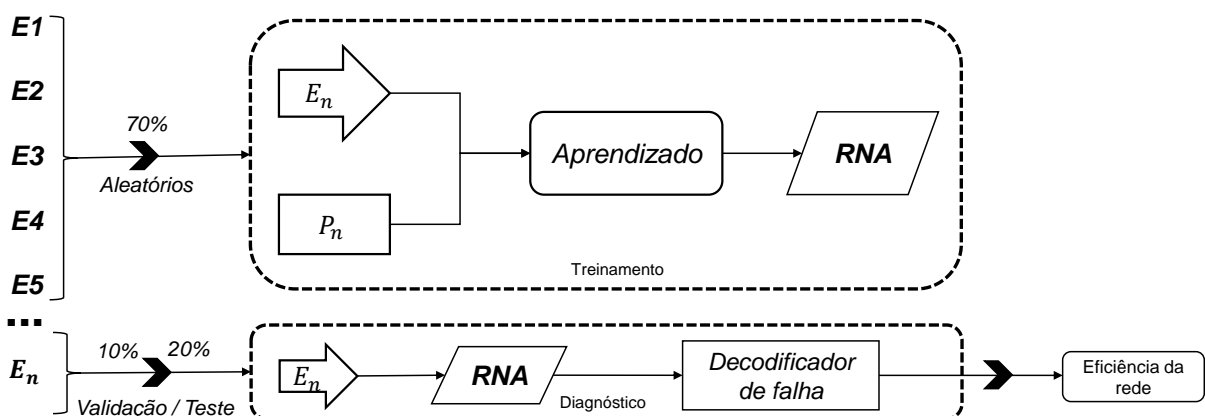
Quanto mais simples for configuração da rede melhor será o seu desempenho, por se tratar de uma configuração simplificada o modelo de identificação de parâmetros e a convergência das variáveis da entrada irá respondendo com menor níveis de ruídos.

Tendo em vista a coleta de dados foram extraídos tempos e alocados em uma matriz, responsável em guardar estas informações temporais e a variação de cada etapa do processo, base esta é utilizada para realizar a parametrização de 250 amostras sem falhas, analisada de forma aleatória a cada etapa concluída do processo. Servindo de conjunto base para inserir erros qualitativos em vetores, simulando defeitos e avarias no processo investigado.

#### 4.4 TESTES DE VALIDAÇÃO DA METODOLOGIA PROPOSTA

A rede neural artificial perceptron de multicamadas pode ser categorizada em dois aspectos, previsão e classificação através de múltiplos ciclos treinamento/diagnóstico, a divisão para realizar o treinamento da rede estão esquematizada na Figura 14. Nos trechos seguintes serão descritos a técnica empregada detalhadamente.

Figura 14 – Representação da divisão do treinamento da rede.



Fonte: Autoria própria.

Para realização do processo de treinamento 70% das amostras foram disponibilizadas de forma aleatória e o " $P_n$ " é a ponderação realizada pela rede neural para tratar e alimentar o próximo estágio. Este conjunto contém amostras com e sem erros na delimitação permitida, cabe

a rede neural selecionar estes possíveis casos com falhas e indicar em qual setor está ocorrendo.

Após realizar o treinamento da rede neural associada, será necessário reservar 30% da rede para validação e reconhecimento dos parâmetros com falhas nas demais amostras analisadas, garantindo um segurança maior na efetividade nos acertos que convergiram. Com isso concluindo a totalidade necessária para obter um grau de confiança ao sistema.

#### 4.5 TREINAMENTO E ESTIMATIVA

Para a etapa de simulação da Rede Neural, foi criada uma base de dados com as características descritas no Quadro 3. As amostras utilizadas para treinamentos e testes na entrada da rede neural artificial, são usadas para alimentar a rede de identificação de falhas, e os erros foram alocados durante o processo real da simulação. Reservando e distribuindo os percentuais da base de dados destinados ao Treinamento, Validação e Teste da RNA, foram arbitrados em suas amostras as taxas de 10% e 20% para teste e validação da rede e 70% como casos aleatórios para rede confirmar que foram aprendidos identifica as variáveis com erros de acordo com a Figura 14.

No Quadro 3 indica a convergência da rede para um modelo satisfatório em suas configurações ideais, para obtenção de uma resposta rápida no reconhecimento de falhas durante o processo.

**Quadro 3 – Dados da interação do treinamento.**

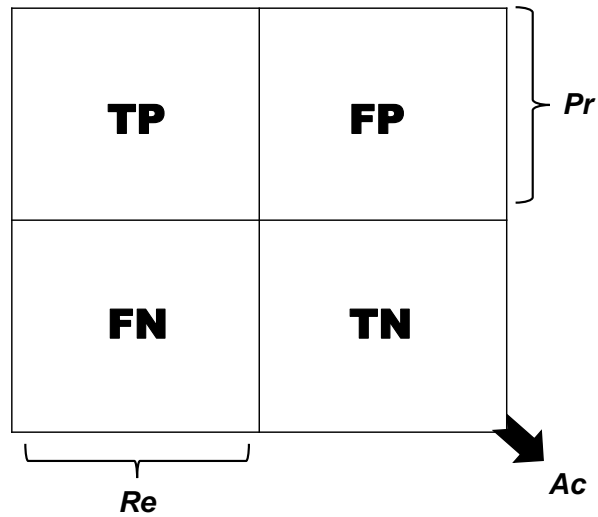
| <b>PROGRESSO</b>   |                       |
|--------------------|-----------------------|
| <i>Épocas</i>      | 8 - Interação         |
| <i>Performance</i> | $1,76 \times 10^{-9}$ |
| <i>Gradiente</i>   | $1,66 \times 10^{-8}$ |
| <i>Pesos</i>       | $1 \times 10^{-11}$   |

Com os dados coletados de um processo sem ausência de erros, e outros com falhas pontualmente adicionadas ao sistema. Podemos obter e avaliar 5 RNAs contendo números diferentes de neurônios na camada oculta (2, 6, 10, 15 e 20). A finalidade desta etapa foi identificar qual é a configuração de Rede Neural que apresenta o melhor desempenho no processo de classificação.

Para analisar os dados de desempenho do problema de classificação da aprendizagem do processo onde a saída pode ser duas ou mais classes. O uso da matriz de confusão é bem útil

nestes casos, pois irá direcionar os diferentes tipos de combinações para um valor previsto no sistema.

**Figura 15 – Performance da matriz de confusão.**



Fonte: Autoria própria.

A Figura 15 representa de forma simples como deve ser efetuados os cálculos para mensurar os acertos da rede e quais obtiveram maior taxa de acertabilidade durante o treinamento e teste maciço de treinamento.

- TP → Taxa de Verdadeiro Positivo;
- TN → Taxa de Verdadeiro Negativo;
- FP → Falso Positivo;
- FN → Falso Negativo;

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

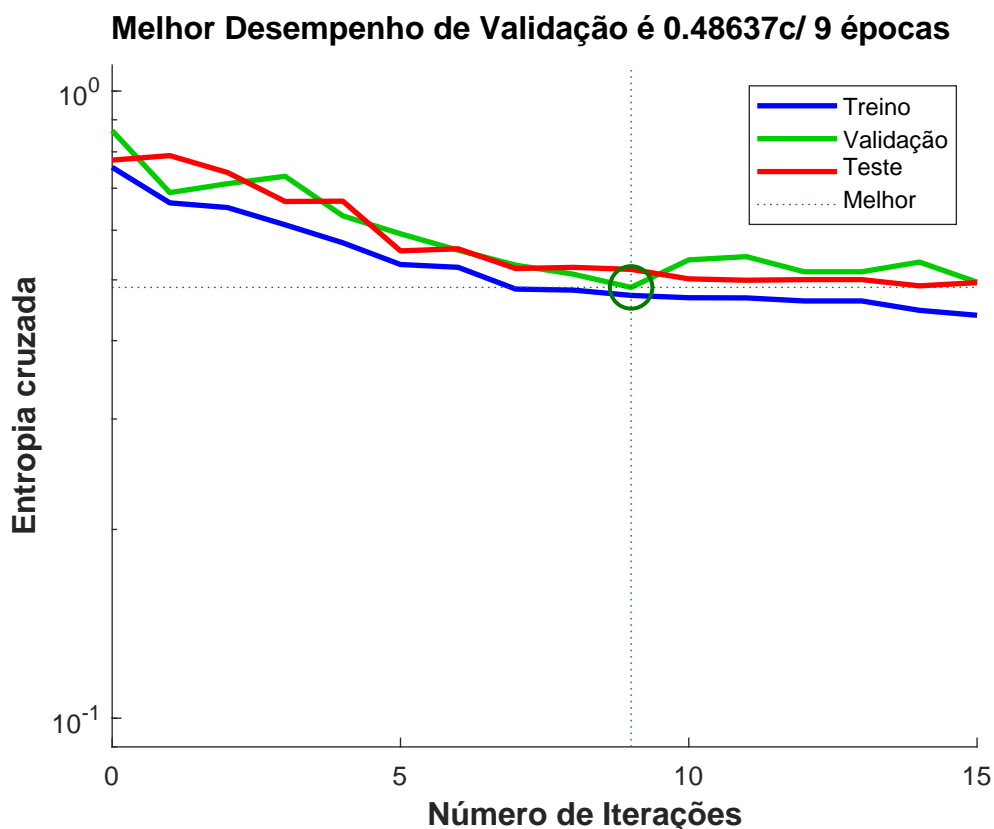
$$Precision = \frac{TP}{TP + FP} \quad (6)$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (7)$$

O *Recall* é a medida que apresenta a qualidade da RNA para prever os positivos, ou seja, indica os acertos da rede para identificar falhas reais no processo. Já o *Precision* verifica qual modelo apresentou melhor performance para atingir o objetivo. E por fim a *Accurary* realiza a análise no contexto geral para indicar a acertabilidade real da rede, apontando previsões corretas para o treinamento, dando em porcentagem o desempenho da RNA. Todas elas, quanto mais altas forem as taxas melhor será a experiência do uso da rede.

O gráfico da entropia cruzada, representada na Figura 16, apresenta o erro entre os resultados obtidos pelo teste de validação e pelos resultados esperados. O objetivo é ajustar iterativamente os pesos da RNA para que se obtenha uma função de transferência otimizada. Pode-se observar na Figura 16 que o mínimo de iteração necessária para atingir a validação da rede é de 9 épocas, com erro de entropia cruzada de 0,48637.

Figura 16 – Erro de entropia cruzada para RNA com 2 neurônios na camada escondida.

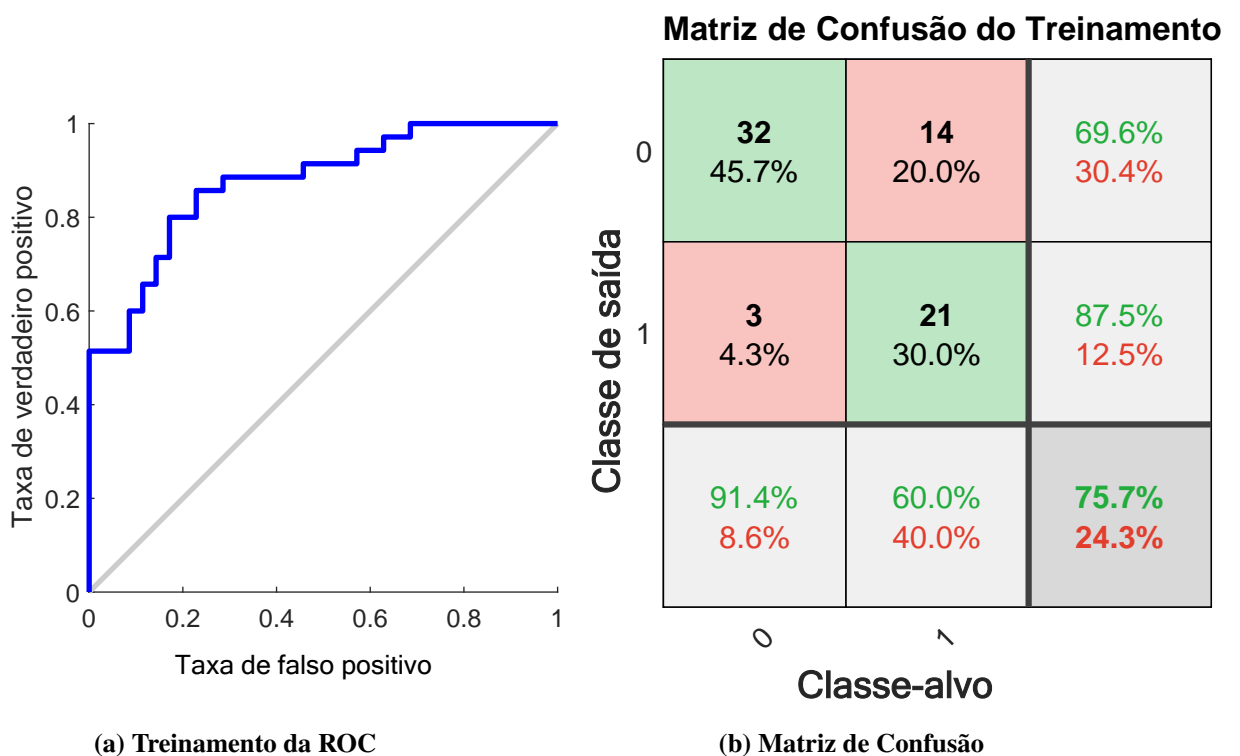


Conceitualmente conseguimos observar que o treinamento da rede irá depender de quantas vezes essa rede será treinada e quantas vezes ela acertará o objetivo, convergindo com “Melhor” desempenho de acertos.

A Curva Característica de Operação do Receptor (ROC) é um gráfico bidimensional em que a Taxa de Verdadeiro Positivo (TVP) é plotada no eixo Y e a Taxa de Falso Positivo (TFP) é plotada no eixo X. O gráfico da curva ROC descreve a relação entre os benefícios (verdadeiros positivos), que contabiliza para rede como acertos, e os erros gerados por falhas de (falsos positivos). O desempenho é mensurado com o tangenciamento da curva próximo ao eixo Y, que demonstra que a RNA está acertando um maior número de TVP. Já se o gráfico tendenciar a seguir a reta em diagonal há indicativos que ele está errando mais que acertando os dados.

A Figura 17(a) demonstra a variação para os valores da curva ROC da RNA com 2 neurônios na camada oculta, para a classe alvo (0 e 1). Bom neste caso o classificador pode só responder apenas dois níveis lógicos, o primeiro caso é quando o sistema identificar falha a RNA irá gerar um nível lógico 1 em sua saída apontando que aquele caso está com erro, já o segundo nível lógico 0, indica o funcionamento normal da rede sem erro.

Figura 17 – Validação do treinamento com 2 neurônios na camada oculta.



Para facilitar a análise dos resultados, foram geradas matrizes de confusão para todas as RNAs trabalhadas. A matriz de confusão 17(b) pode ser entendida como um tipo de quadro que permite a visualização do desempenho de classificação em aprendizado de máquina. As colunas da matriz representam a instância da classe atual, enquanto as linhas representam os casos de uma classe predita e a diagonal a efetividade dos acertos das RNAs. Nesta configuração,

com apenas 2 neurônios na camada escondida, a rede conseguiu obter uma classificação geral de 75,5% de acertos, ou seja, quando executado a verificação dos casos com a RNA, o classificador conseguiu convergir apenas 189 a 190 amostras de um total de 250 amostras injetados na entrada, carregando uma taxa de 24,3% com erros de falsos positivos em sua análise.

O gráfico da Figura 18, conseguimos observar que a rede com 6 neurônios na camada escondida precisou de menos épocas para atingir o melhor desempenho, que se estabilizou com uma entropia cruzada de 0,58484 com 3 épocas de interação, depois se manteve igualmente estável durante os testes de treinamento e validação da rede. A curva ROC da Figura 19(a) foi a pior dentre as propostas para este trabalho, pois sua taxa de falso positivo foi bem mais suscetivo a sinalização de acertos em uma rede que com vários erros.

**Figura 18 – Erro de entropia cruzada para RNA com 6 neurônios na camada escondida.**

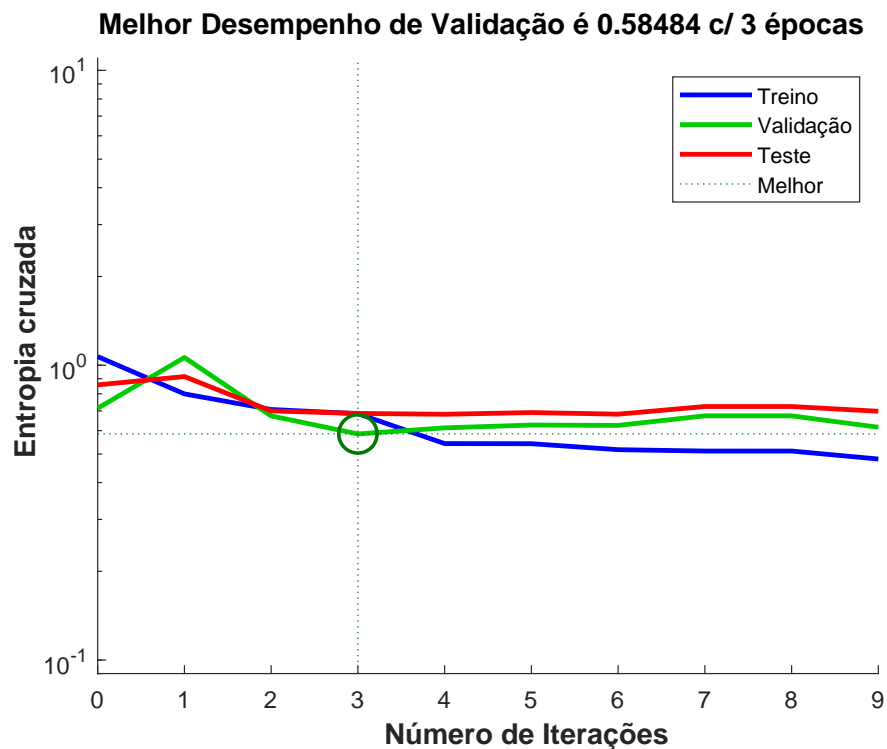


Figura 19 – Validação do treinamento com 6 neurônios na camada oculta

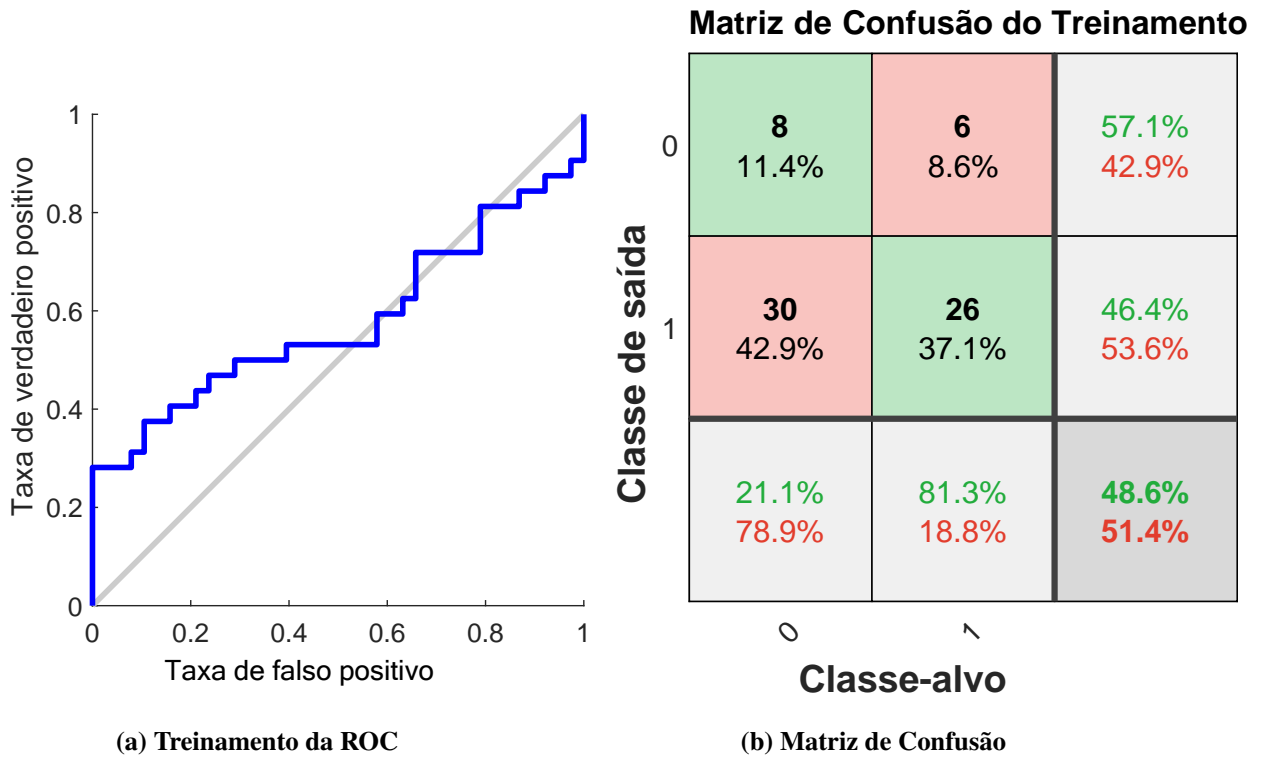


Figura 20 – Erro de entropia cruzada para RNA com 10 neurônios na camada escondida.

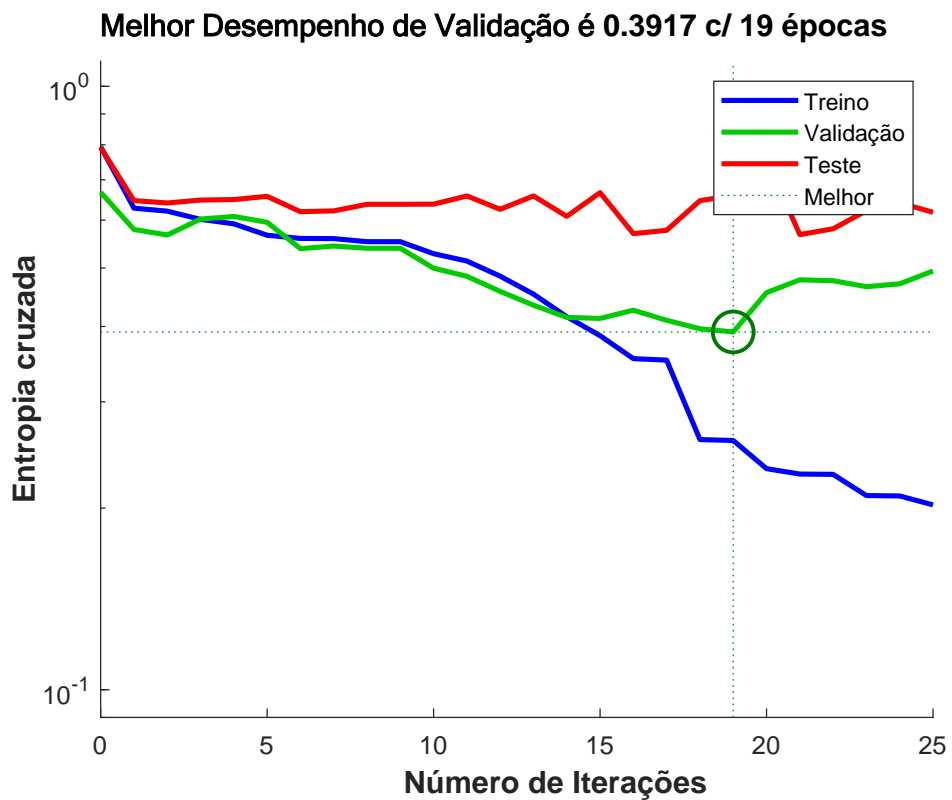


Figura 21 – Validação do treinamento com 10 neurônios na camada oculta.

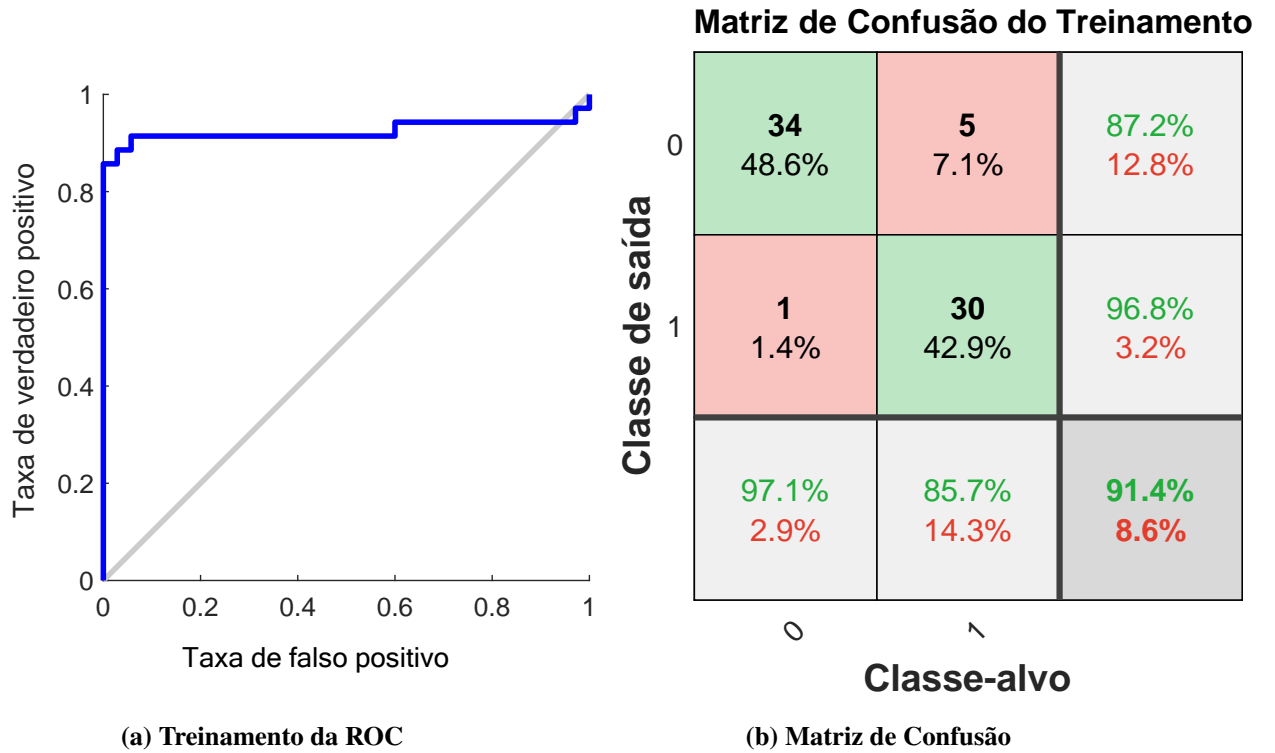


Figura 22 – Erro de entropia cruzada para RNA com 15 neurônios na camada escondida.

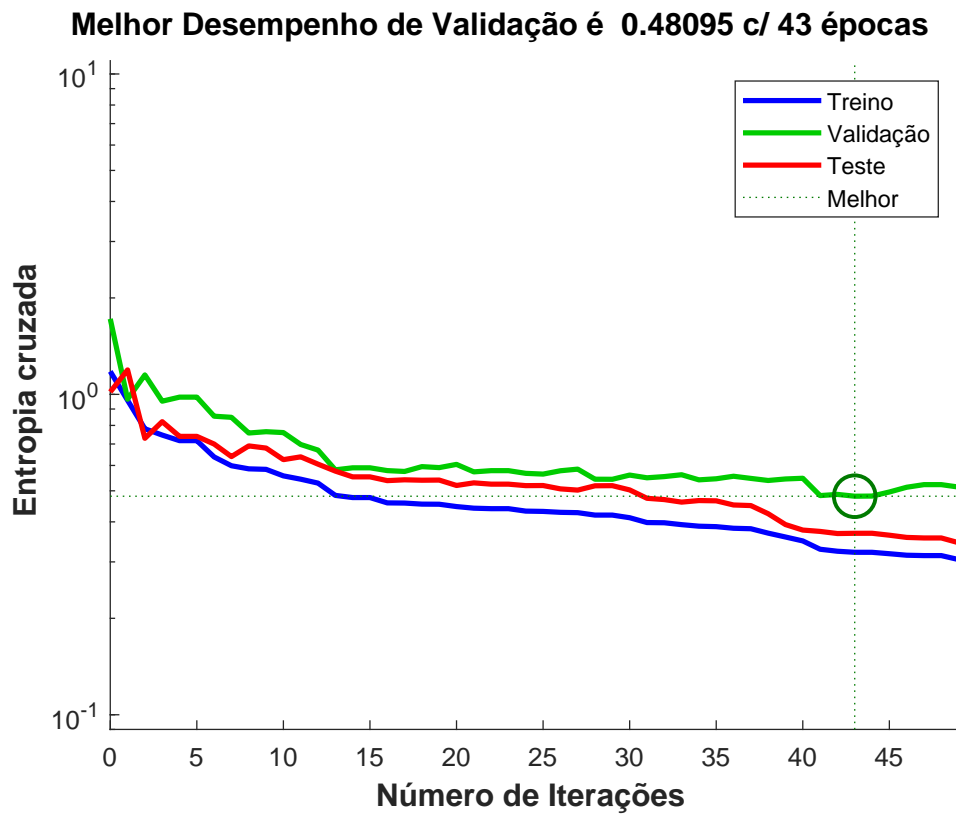




Figura 23 – Validação do treinamento com 15 neurônios na camada oculta.

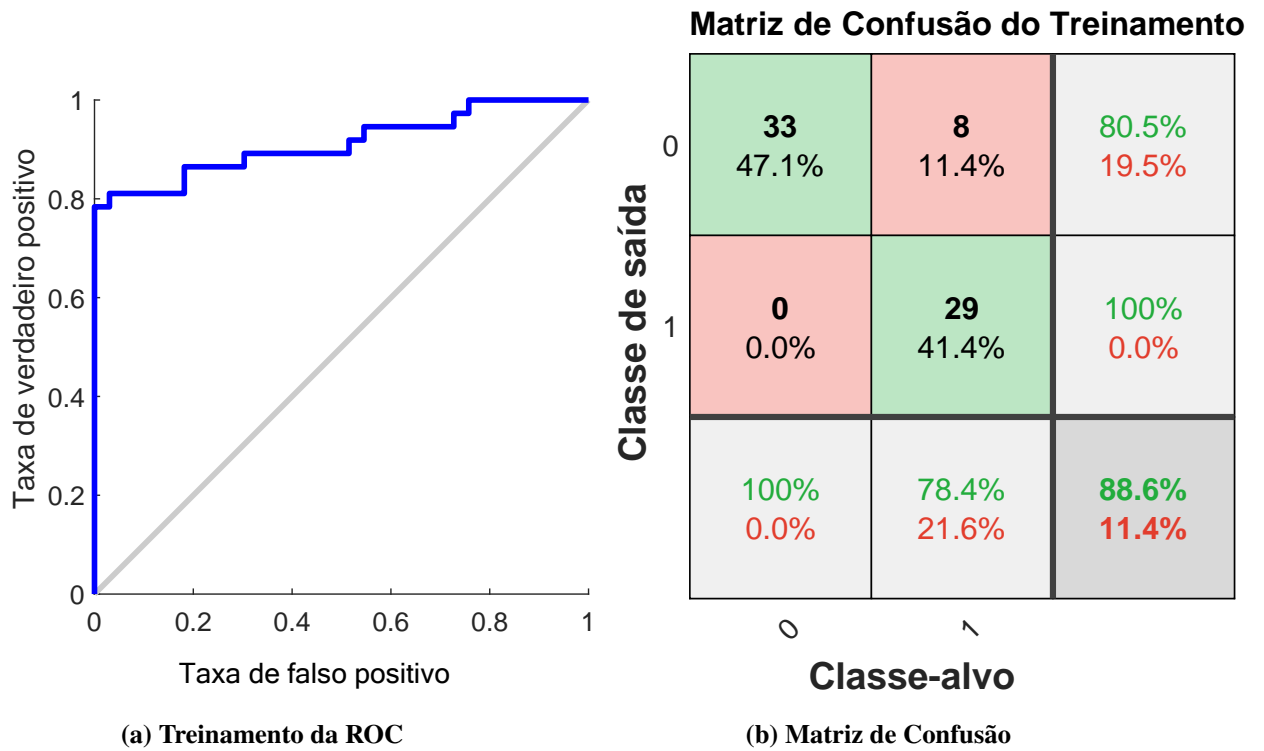


Figura 24 – Erro de entropia cruzada para RNA com 20 neurônios na camada escondida.

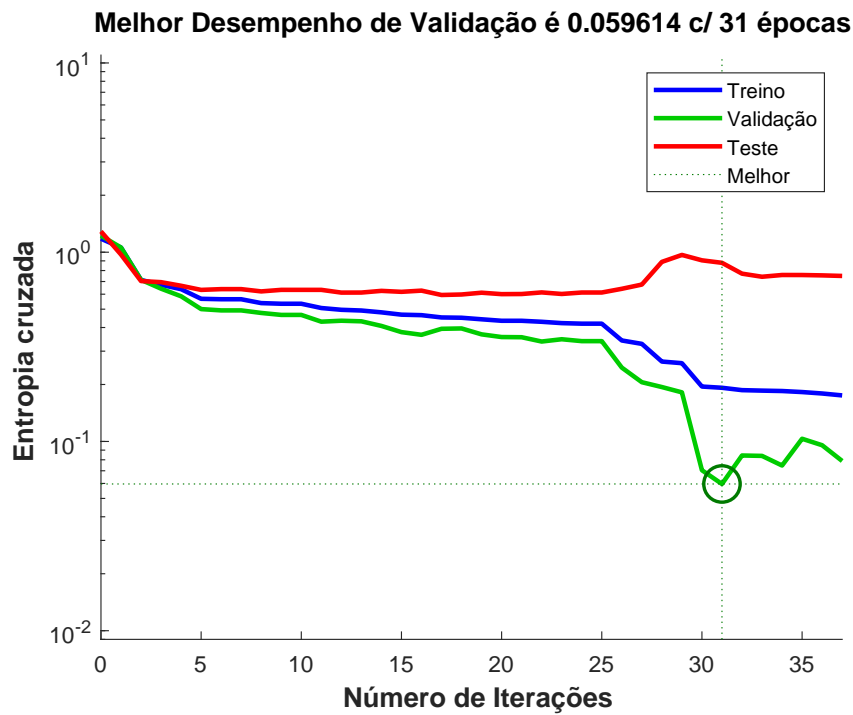
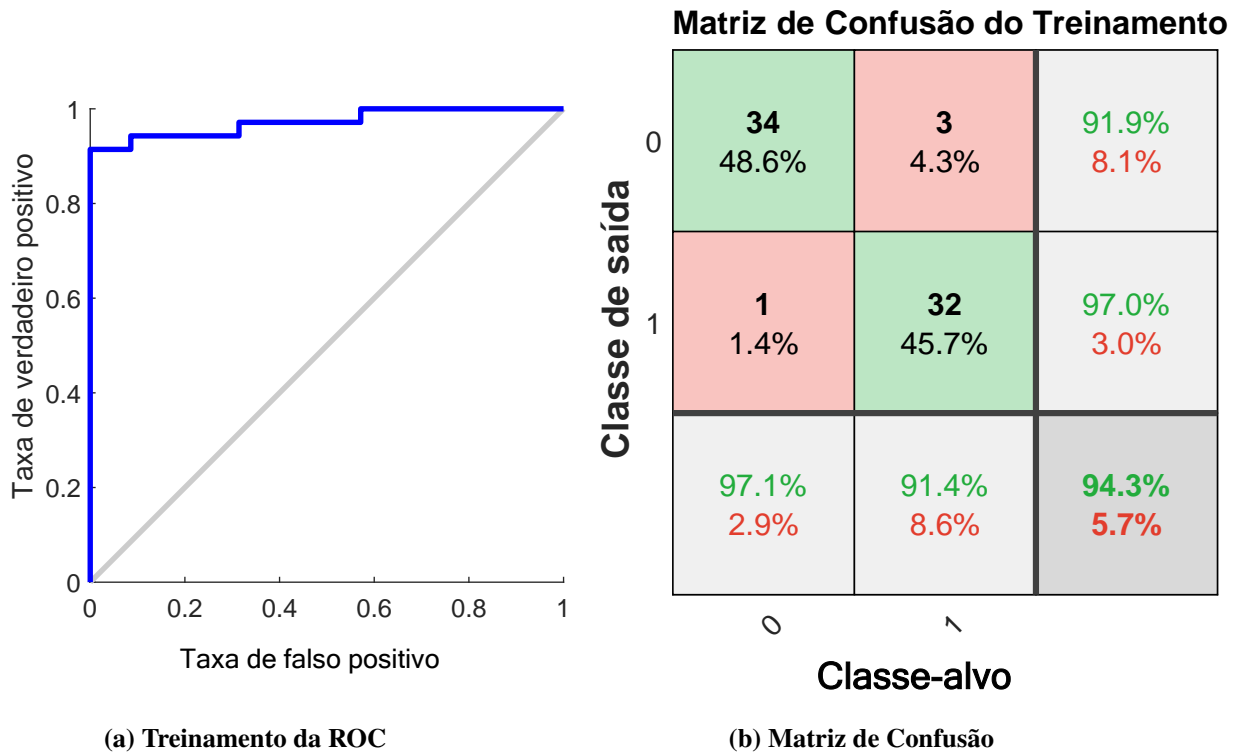


Figura 25 – Validação do treinamento com 20 neurônios na camada oculta



Dentro de todas as variações de treinamento e teste da rede, para localizar erro no processo de simulação, a RNA com 20 neurônios em sua camada escondida foi a que mais se adequou com os dados de treinamento, como observamos na Figura 24 e confirmando sua efetividade na curva ROC 25(a) e matriz de confusão 25(b). O referido número de neurônios demonstrou o melhor desempenho para as RNAs usadas, obtendo precisão geral de 94% de acerto na identificação.

Outra análise que podemos extrair destes dados está localizado na Figura 24, que demonstra que sua curva de validação foi umas das menores durante o processo, pois neste caso a curva gerada pelo treinamento da rede tende a sobrepôr as outras convergindo ainda mais rápido a rede, seria o melhor momento de se parar a análise. E muitas vezes não estaria extraindo um excelente desempenho daquela rede, por tanto ao realizar um estudo minucioso para alterar parâmetros da RNA com o foco em melhorar o desempenho.

Quando adicionado mais neurônios na camada oculta da rede, seu comportamento é um tanto incomum, pois ela não responde para uma boa convergência, apresentando divergências, e gerando desempenho e gráficos muito abaixo dos testes anteriores.

## 5 CONCLUSÕES

O presente trabalho foi desenvolvido com o intuito de fornecer uma solução de detecção e temporização dos eventos para identificar falhas em processos automatizados. Para isso, o sistema recorreu a redes neurais que melhor se adaptou as condições de produtividade, a partir dos erros comumente apresentados neste processo, indicando ao usuário os momentos em que as falhas estavam ocorrendo, para reposição a partir do CLP de aplicação redundante interligado a rede industrial.

A Inteligência Artificial foi utilizada para eliminar um dos controladores na malha industrial para o monitoramento de falhas, tornando-se uma condição viável para aplicações industriais em grandes escalas, pois são dispositivos de baixo custo de aquisição, sendo estruturas eficientes na identificação de modelos, detecção e o diagnóstico de falhas.

Os CLPs podem ser utilizados em aplicações paralelas e redundantes, interligados com redes neurais treinadas e condicionadas como elementos de supervisão. A rede industrial PROFIBUS amplia as possibilidades de comunicação entre os controladores, conferindo potencial de expansão do número de entradas e saídas através da inserção de barramentos de E/S remotos. As mensagens de diagnóstico dos escravos são fontes de dados para o gerenciamento do sistema de automação e facilitam a correção de eventuais problemas.

O estudo de caso desenvolvido neste trabalho ilustrou a utilização da plataforma. Foi implementada uma arquitetura mista, com o uso de dois CLPs voltados para o controle da planta industrial e a rede neural na supervisão da evolução dos estados da planta. A técnica que utiliza a IA, como forma de supervisão e na implementação do método de detecção de falhas através da análise dos dados de entradas simultaneamente discretizados, para reconhecimento de padrões no processo, localizando e verificando se os estados da planta evoluem conforme o esperado. Os CLPs destinados ao controle foram ligados num arranjo de redundância *hot-standby*, atuando um como contingência do outro, ou seja, se o controlador principal falhar, o controlador de redundância entra em ação imediatamente, sem perda da continuidade ao processo. O CLP principal é responsável por comandar a planta, enquanto o redundante deve assumir o comando do barramento apenas quando o principal entrar em modo de falha.

Os testes e implementação das estratégias foram realizados por meio de experimentos com injeção de falhas no sistema, tanto nos instrumentos da planta como nos CLPs. A modelagem através de Rede de Petri serviu como base para a validação da estratégia de detecção de falhas,

constatando a correção parcial dos subsistemas em falha realizadas pela Inteligência Artificial de supervisão.

Outro ponto de interesse neste trabalho, relacionado aos métodos aplicados para validar a estratégia de correção de falhas e detecção de um problema no CLP principal e a troca de comando ocorreram em tempo curto o bastante para não provocar uma descontinuidade. Isto indica eficiência da estratégia de coordenação dos CLPs e demonstra que as estratégias podem ser utilizadas em conjunto, conferindo ao sistema de identificação que no futuro pode ser um sistema tolerante a falhas de instrumentos como a falhas de controladores.

Em contraponto, vale salientar quando realizados os testes em plataforma prática para avaliar o desempenho das redes de detecção e correção de falhas atuando de maneira simultânea ou paralelamente, está suscetível em gerar falsos positivos, uma vez que as variáveis de entrada da rede são idênticas. Outro aspecto que reforça essa teoria são pesquisas correlacionadas ao tema do trabalho, que indicam que as Inteligências Artificiais são instáveis, podendo convergir ou não para elementos reais em aplicações diversas.

Em futuras pesquisas, objetiva-se a ampliação de outras falhas relacionadas aos mais diversos tipos de processos e o isolamento do erro de forma simultânea a substituição do CLP de aplicação. Ainda, pretende-se buscar algoritmos mais robustos de segmentação dos componentes, especialmente para localizar o erro e apresentar qual componente pertencente ao processo se encontra com falha e qual o grau desta falha, indicando um parâmetro para o estudo da causalidade do erro com o processo. Outro ponto que deve ser trabalhado nesta pesquisa e a avaliação da performance do atual classificador do sequenciamento do processo, reiniciando de forma automática.

## REFERÊNCIAS

ALBUQUERQUE, Pedro H; SAAVEDRA, Cayano Azeiteiro Portela Bárcena; MORAIS, Rafael Lima de; ALVES, Patrick Franco; PENG, Yaohao. Na era das máquinas, o emprego é de quem? estimação da probabilidade de automação de ocupações no Brasil. *In: Texto para Discussão*. Rio de Janeiro: Instituto de Pesquisa Econômica Aplicada – IPEA, 2019. p. 15–29.

ARAÚJO, CELSO DE; JÚNIOR, SALOMÃO CHOUERI; SANTOS, WINDERSON EUGENIO DOS; SILVEIRA, PAULO ROGÉRIO DA; CRUZ, EDUARDO CESAR ALVES. **Automação e Controle Discreto**. São José dos Campos, SP: Saraiva Educação SA, 2009.

BITTAR, Rita de Cassia da Silveira Marconcini. **A utilização do GRAFCET como ferramenta na automação industrial**. 1993. 31 p. Dissertação (Mestrado) — Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, 1993.

BLANKE, Mogens; KINNAERT, Michel; LUNZE, Jan; STAROSWIECKI, Marcel; SCHRÖDER, Jochen. **Diagnosis and fault-tolerant control**. California-eua: Springer, 2006. v. 2.

BOLTON, William. **Programmable Logic Controllers**. Oxford: Newnes, 2015. 137 p.

CAPUTO, Antonio C; PELAGAGGE, Pacifico M; SALINI, Paolo. Economic comparison of manual and automation-assisted kitting systems. **IFAC-PapersOnLine**, Elsevier, v. 51, n. 11, p. 1482–1487, 2018.

CARDOSO, Janette; VALETTE, Robert. **Redes de petri**. Florianópolis, SC: Editora da UFSC Florianópolis, 1997.

CASSANDRAS, Christos G; LAFORTUNE, Stéphane. **Introduction to discrete event systems**. California, EUA: Springer Cham, 2008.

CHADEEV, VM; ARISTOVA, NI. Control of industrial automation. *In: IEEE. 2017 Tenth International Conference Management of Large-Scale System Development (MLSD)*. Moscow – Russian Federation, 2017. p. 1–5.

CHATTAL, Muhammad; BHAN, Veer; MADIHA, Hina; SHAIKH, Shoaib Ahmed. Industrial automation & control through PLC and LabVIEW. *In: IEEE. 2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (ICoMET)*. Sukkur, Pakistan, 2019. p. 1–5.

CHENG, Yukun; ZHENG, Mengce; HUANG, Fan; ZHANG, Jiajia; HU, Honggang; YU, Nenghai. A fast-detection and fault-correction algorithm against persistent fault attack. *In: IEEE. 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. Shenyang, China, 2021. p. 557–568.

CHUPONG, Charnon; PLANGKLANG, Boonyang. Comparison study on artificial neural network and online sequential extreme learning machine in regression problem. *In: IEEE. 2019 7th International Electrical Engineering Congress (iEECON)*. Hua Hin, Thailand, 2019. p. 1–4.

CURY, José Eduardo Ribeiro. Teoria de controle supervisorio de sistemas a eventos discretos. **V Simpósio Brasileiro de Automação Inteligente (Minicurso)**, Canela-RS, p. 8, 2001.

CZAJKOWSKI, Andrzej; PATAN, Krzysztof. Design of predictive fault tolerant control by the means of state space neural networks. *In: IEEE. 2016 24th Mediterranean Conference on Control and Automation (MED)*. Athens, Greece, 2016. p. 201–206.

DAYOUB, Feras; SUNDERHAUF, Niko; CORKE, Peter I. Episode-based active learning with bayesian neural networks. *In: IEEE. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. Honolulu, HI, USA, 2017. p. 26–28.

DING, Xianxian; LI, Qing; CUI, Jiarui. Subspace method aided fault detection for distributed identical systems. *In: IEEE. 2019 Chinese Automation Congress (CAC)*. Hangzhou, China, 2019. p. 2054–2058.

EL-MAHDY, Mostafa H; MAGED, Shady A; AWAD, Mohammed I. Active fault tolerant control of discrete event system subjected to sensors fault. *In: IEEE. 2021 17th International Computer Engineering Conference (ICENCO)*. Cairo, Egypt, 2021. p. 24–29.

FLECK, Leandro; TAVARES, Maria Hermínia Ferreira; EYNG, Eduardo; HELMANN, Andrieli Cristina; ANDRADE, MA de M. Redes neurais artificiais: Princípios básicos. **Revista Eletrônica Científica Inovação e Tecnologia**, Curitiba - PR, v. 1, n. 13, p. 47–57, 2016.

FRANCHI, Claiton Moro; CAMARGO, Valter Luís Arlindo de. **Controladores lógicos programáveis: sistemas discretos**. São José dos Campos, SP: Saraiva Educação SA, 2008.

FURTADO, Maria Inês Vasconcellos. Redes neurais artificiais: uma abordagem para sala de aula. **Atena Editora**, Ponta Grossa, PR, p. 36–43, 2019.

GHEORGHIU, Corina-Ioana; DEACONU, Ioan-Dragoş; CHIRILĂ, Aurel-Ionuţ; NĂVRĂPESCU, Valentin. Educational platform for an underground smart parking using programmable logic controllers. *In: IEEE. 2021 International Conference on Applied and Theoretical Electricity (ICATE)*. Craiova, Romania, 2021. p. 1–6.

GIL, Antonio Carlos *et al.* **Como elaborar projetos de pesquisa**. São Paulo - SP: Atlas, 2002. v. 4.

HAYKIN, Simon. **Redes neurais: princípios e prática**. PORTO ALEGRE, Brasil: Bookman Editora, 2001.

ISO, ISO. 12100–1: safety of machinery–general principles for design. **International Organization for Standardization. Technical Report**, 2010.

JÚNIOR, Antônio Pereira de Araújo; CHAGAS, Christiano Vasconcelos das; FERNANDES, Raphaela Galhardo. Uma rápida análise sobre automação industrial. 2003.

JÚNIOR, Reinaldo Squillante. **Diagnóstico e tratamento de falhas críticas em sistemas instrumentados de segurança**. 2011. Tese (Doutorado) — Universidade de São Paulo, 2011.

KHANDELWAL, Shubham; ALDHANDI, Suresh; DETROJA, Ketan P. Centralized control with decoupling approach for large scale multivariable processes. *In: IEEE. TENCON 2019-2019 IEEE Region 10 Conference (TENCON)*. Kochi, India, 2019. p. 2633–2638.

KOREN, Israel; KRISHNA, C Mani. **Fault-tolerant systems**. Cambridge - EUA: Morgan Kaufmann, 2020. v. 5.

LAMB, Frank. **Industrial automation: Hands-on**. Nashville Tennessee - EUA: McGraw-Hill Education, 2013. v. 1.

LO, Fasain; SU, Fangfang; CHEN, Shaoze; QIU, Jinfeng; DU, Jiong. Artificial intelligence aided innovation education based on multiple intelligence. *In: IEEE. 2021 IEEE International Conference on Artificial Intelligence, Robotics, and Communication (ICAIRC)*. Fuzhou, Fujian P.R, China, 2021. p. 12–15.

LUO, Hao; LI, Qing. Evaluation indicators architecture for industrial automation system integration. *In: IEEE. 2014 IEEE International Conference on Information and Automation (ICIA)*. Hailar, China, 2014. p. 1082–1085.

MENEZES, Paulo Blauth. **Linguagens Formais e Autômatos: Volume 3 da Série Livros Didáticos Informática UFRGS**. Porto Alegre, RS: Bookman Editora, 2009.

MOHANASUNDARAM, N. Non linear predictive modelling for ic engine using artificial neural network. *In: IEEE. 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*. Palladam, India, 2020. p. 801–807.

MOLINA, FJ; BARBANCHO, JLCM; LEON, C; MOLINA, A; GOMEZ, A. Using industrial standards on plc programming learning. *In: IEEE. 2007 Mediterranean Conference on Control & Automation*. Athens, Greece, 2007. p. 1–6.

MORAES, Cícero Couto de; CASTRUCCI, Plínio de Lauro. Engenharia de automação industrial. **Rio de Janeiro: LTC**, p. p295, 2001.

PETRUZELLA, Frank D. **Controladores lógicos programáveis**. Porto Alegre, RS: AMGH Editora, 2014.

PRADHAN, Dhiraj K. **Fault-tolerant computer system design**. Hoboken, New Jersey, U.S: Prentice-Hall, Inc., 1996.

REBOUÇAS, Diogo Leite. **Utilização de redes neurais artificiais para detecção e diagnóstico de falhas**. 2011. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Norte, 2011.

REFAAT, Tarek K; AMER, Hassanein H; ALKADY, Gehad I; DAOUD, Ramez M; ELSAYED, Hany M. Machine operating speed, fault security and fault tolerance for performability analysis in industrial automation. *In: IEEE. 2021 IEEE 30th International Symposium on Industrial Electronics (ISIE)*. Kyoto, Japan, 2021. p. 1–6.

RIBEIRO, Marco Antônio. Automação industrial. **Tek Treinamento & Consultoria Ltda**, Salvador, 1999.

RIERA, Bernard; BENLORHFAR, Rachid; ANNEBICQUE, D; GELLOT, F; VIGARIO, B. Robust control filter for manufacturing systems: application to plc training. **IFAC Proceedings Volumes**, Elsevier, v. 44, n. 1, p. 14265–14270, 2011.

SAKURADA, Nelson; MIYAKE, Dario Ikuo. Aplicação de simuladores de eventos discretos no processo de modelagem de sistemas de operações de serviços. **Gestão & Produção**, SciELO Brasil, v. 16, p. 25–43, 2009.

SHEN, Yang. The application of artificial intelligence in computer network technology in the era of big data. *In: IEEE. 2021 international conference on computer technology and media convergence design (CTMCD)*. Kunming, China, 2021. p. 173–177.

STREIT, Alberto; RÖSCH, Susanne; VOGEL-HEUSER, Birgit. Redeployment of control software during runtime for modular automation systems taking real-time and distributed i/o into consideration. *In: IEEE. Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*. Barcelona, Spain, 2014. p. 1–4.

TIEGELKAMP, Michael; JOHN, Karl-Heinz. **IEC 61131-3: Programming industrial automation systems**. California, EUA: Springer, 2010. v. 166.



VELAGIC, Jasmin; KAKNJO, Admir; OSMIC, Nedim; DZANANOVIC, Tarik. Networked based control and supervision of induction motor using opc server and plc. *In: IEEE. Proceedings ELMAR-2011*. Zadar, Croatia, 2011. p. 251–255.

VIEIRA, Agnelo Denis *et al.* Método de implementação do controle de sistemas e eventos discretos com aplicação da teoria de controle supervisório. Florianópolis, SC, 2007.

WU, Yinan; PENG, Gongzhuang; ZHANG, Heming. A heuristic pathfinding algorithm for dynamic fault tolerance in manufacturing networks. *In: IEEE. 2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*. Vancouver, BC, Canada, 2019. p. 1580–1585.

YOO, Jihoon; MIN, Byeongjun; KIM, Sangsoo; SHIN, Dongil; SHIN, Dongkyoo. Study on network intrusion detection method using discrete pre-processing method and convolution neural network. *IEEE Access*, IEEE, IEEE Electronic, v. 9, p. 142348–142361, 2021.

YOUSEFZADEH, Amirreza; ORCHARD, Garrick; STROMATIAS, Evangelos; SERRANO-GOTARREDONA, Teresa; LINARES-BARRANCO, Bernabe. Hybrid neural network, an efficient low-power digital hardware implementation of event-based artificial neural network. *In: IEEE. 2018 IEEE International Symposium on Circuits and Systems (ISCAS)*. Florence, Italy, 2018. p. 1–5.

## **ANEXO**

**ANEXO A – REDE DE PETRI REPRESENTANDO O SISTEMA DE SELEÇÃO DE CAIXAS E OS ESTADOS DE ERRO IDENTIFICADOS PELA ESTRATÉGIA DE DETECÇÃO DE FALHAS.**

