

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO RIO GRANDE - FURG
PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM COMPUTACIONAL

SISTEMA DE AUTO-CONFIGURAÇÃO RESILIENTE BASEADO NA
ARQUITETURA CPS-5C

Manuel Leonardo Simosa Sotillo

Dissertação para obtenção do Título de
Mestre em Modelagem Computacional
Orientador: Prof^a. Dr^a. Sílvia Silva Da Costa Botelho

Rio Grande, Março 2019

Ficha catalográfica

S611s Simosa Sotillo, Manuel Leonardo.
Sistema de auto-configuração resiliente baseado na arquitetura
CPS-5C / Manuel Leonardo Simosa Sotillo. – 2019.
57 f.

Dissertação (mestrado) – Universidade Federal do Rio Grande –
FURG, Programa de Pós-Graduação em Modelagem Computacional,
Rio Grande/RS, 2019.
Orientadora: Dra. Sílvia Silva da Costa Botelho.

1. Indústria 4.0 2. Sistemas Ciberfísicos 3. Sistemas Auto
Adaptativos 4. Sistemas de Controle Resiliente 5. OPC-UA I. Botelho,
Sílvia Silva da Costa II. Título.

CDU 004



ATA DE DEFESA DE DISSERTAÇÃO

ATA N° 03/2019

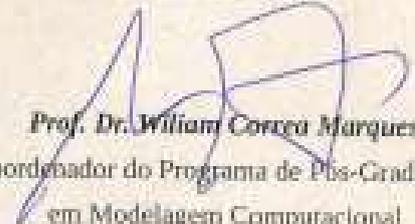
Ao décimo quinto dia do mês de março, na sala de reuniões do C3 da Universidade Federal do Rio Grande - FURG, realizou-se a terceira sessão de defesa de dissertação do ano de dois mil e dezenove do Programa de Pós-Graduação em Modelagem Computacional. O mestrando **Manuel Leonardo Simosa Sotillo** apresentou e defendeu a dissertação "Sistema de auto-configuração resiliente baseado na arquitetura CPS-5C" como requisito parcial para a obtenção do grau de Mestre em Modelagem Computacional. A sessão foi aberta às quatorze horas e trinta minutos pela Prof. Dr. Silvia Silva da Costa Botelho, orientadora da dissertação e presidente da Comissão de Avaliação. A referida Comissão esteve integrada, além da presidente e orientadora, pelo Prof. Dr. Emanuel da Silva Diaz Estrada, da FURG, pelo Prof. Dr. Eder Mateus Nunes Gonçalves, da FURG, e pelo Prof. Dr. Renato Ventura Bayan Henriques, da UFRGS. Após cada integrante da Banca ter arguido o mestrando, e este ter tido a oportunidade de responder a cada um, a Comissão reuniu-se para deliberar sobre o conceito a ser atribuído ao trabalho. A Comissão de Avaliação considerou a candidata neste requisito parcial para a obtenção do grau de Mestre em Modelagem Computacional. Por fim, o presidente publicou e encerrou a sessão da qual eu, William Correa Marques, na qualidade de Coordenador do Programa de Pós-Graduação em Modelagem Computacional, lavrei a presente ata que vai assinada por todos os membros da Comissão de Avaliação. Rio Grande, quinze de março de 2019.


Prof. Dr. Silvia Silva da Costa Botelho
Orientadora


Prof. Dr. Emanuel da Silva Diaz Estrada
Membro Interno


Prof. Dr. Eder Mateus Nunes Gonçalves
Membro Interno


Prof. Dr. Renato Ventura Bayan Henriques
Membro Externo


Prof. Dr. William Correa Marques
Coordenador do Programa de Pós-Graduação
em Modelagem Computacional

DEDICATÓRIA

*Dedico este trabalho a Deus, o autor da vida,
em especial a minha família, meus pais
Elayne y Lorenzo e a minha irmã Ana Karina,
pelo incansável e essencial apoio, Sem vocês
nada seria possível! Amo vocês!*

AGRADECIMENTOS

Agradecimento especial a Deus: Senhor, pela proteção e cuidado incondicional. Te amo pelo que és e pelo que faz por mim.

Aos meus pais e irmãos que me deram o apoio moral, emocional e espiritual para que eu pudesse concretizar um sonho, além de entender a minha ausência. Amo do tamanho do céu!

Um reconhecimento às professoras Silvia Botelho como orientadora e Gionara Tauchen como, por sua paciência e compreensão com a minha situação, como estrangeiro.

Aos professores, funcionários, colegas e grupo de pesquisa NAUTEC-3DCS, que de alguma forma, contribuíram para o meu desenvolvimento acadêmico.

Aos amigos da FURG e da OEA pelo incentivo e os momentos gratos nesta caminhada.

A CAPES, à Universidade Federal do Rio Grande (FURG) e a OEA pela oportunidade e apoio financeiro.

Fácil não foi, entretanto, tudo valeu a pena!

RESUMO

Os Sistemas Ciber-Físicos (CPS) são comumente usados para o desenvolvimento da Indústria, especificamente o paradigma da Indústria 4.0, permitindo a conexão do mundo virtual com o mundo físico e o aproveitamento das tecnologias da informação. A arquitetura CPS de Jay Lee 5C, permite a aquisição de dados, administração do sistema e respostas de controle, para a adoção de diferentes reconfigurações. Este trabalho propõe uma arquitetura para o desenvolvimento da reconfiguração do sistema, levando em consideração a interoperabilidade entre componentes físicos e a capacidade de manter as condições normais de operação após a presença de eventos, em uma arquitetura 5C. Um caso de uso é feito em uma planta SMAR-PD3 que valida a proposta, obtendo comunicação através do protocolo OPC-UA, a reconfiguração baseada em um modelo MAPE-K e a redução dos tempos de recuperação. Os resultados obtidos permitem conhecer as reconfigurações baseadas em simulações proativas para a redução dos tempos de recuperação.

Palavras-chaves: *Indústria 4.0, Sistemas Ciberfísicos, Sistemas Auto Adaptativos, Sistemas de Controle Resiliente, OPC-UA.*

ABSTRACT

Cyber-Physical Systems (CPS) are commonly used for the development of Industry, specifically the Industry 4.0 paradigm, allowing the connection of the virtual world with the physical world and the use of information technologies. The CPS architecture of Jay Lee 5C allows data acquisition, system administration, and control responses for the adoption of different reconfigurations. This work proposes an architecture for the development of the reconfiguration of the system, taking into account the interoperability between physical components and the ability to maintain normal operating conditions after the presence of events in a 5C architecture. A use case is made in a SMAR-PD3 plant that validates the proposal, obtaining communication through the protocol OPC-UA, the reconfiguration based on a MAPE-K model and the reduction of recovery times. The results obtained allow to know the reconfigurations based on proactive simulations for the reduction of recovery times.

Keywords: *Industry 4.0, Cyber-Physical Systems, Self-Adaptive Systems, Resilient Control Systems, OPC-UA.*

ÍNDICE

DEDICATÓRIA	4
AGRADECIMENTOS.....	5
RESUMO	6
ABSTRACT	7
ÍNDICE	8
LISTA DE FIGURAS	10
LISTA DE QUADROS	11
LISTA DE ABREVIATURAS	12
1 INTRODUÇÃO.....	13
1.1 Contextualização e Justificativa.....	13
1.2 Objetivos.....	14
1.2.1 Objetivo Geral	14
1.2.2 Objetivos Específicos.....	14
1.3 Organização do Trabalho	15
2 FUNDAMENTAÇÃO TEÓRICA	16
2.1 Indústria 4.0	16
2.2 Sistemas Ciberfísicos	18
2.2.1 Arquitetura 5C de Sistemas Ciber-Físicos.....	19
2.3 Auto Adaptação	21
2.3.1 Taxonomia de Auto Adaptação.....	22
2.3.2 MAPE-K.....	23
2.4 Interoperabilidade.....	25
2.4.1 Protocolos industriais para CPS	25
2.4.2 Arquitetura Unificada OPC	26
2.5 Resiliência.....	29

2.5.1	Tempos de Resiliência	30
2.6	Sistema de Controle Resiliente.....	31
2.7	Trabalhos relacionados	33
2.8	Considerações Finais	35
3	ARQUITETURA PROPOSTA	38
3.1	Interoperabilidade.....	39
3.1.1	Monitoreamento	40
3.1.2	Execução.....	43
3.2	Resiliência.....	43
3.2.1	Conhecimento	44
3.2.2	Análise	44
3.3	Auto Adaptação	44
3.3.1	Planejamento	45
4	ESTUDO DE CASO.....	46
4.1	Funcionamento da planta SMAR PD3	46
4.2	Implementação.....	47
4.3	Validação do caso de uso.....	52
5	CONCLUSÕES E TRABALHOS FUTUROS	54
6	REFERÊNCIAS	55

LISTA DE FIGURAS

Figura 2.1	Perspectiva histórica da Revolução Industrial.....	16
Figura 2.2	Decomposição da hierarquia de automação com serviços distribuídos	17
Figura 2.3	Arquitetura 5C	19
Figura 2.4	Estrutura de Gerenciador Auto Adaptativo	24
Figura 2.5	Tecnologias de Comunicação OPC-UA	27
Figura 2.6	Estrutura de Especificações OPC-UA	28
Figura 2.7	Tempos de Resiliência.....	30
Figura 2.8	Arquitetura Conceitual de CPS de resiliência	32
Figura 2.9	Taxonomia da Auto Adaptação	35
Figura 2.10	Modos de simulação de operação reativa/proativa.....	36
Figura 2.11	Descrição básica dos tipos de dados usados na estrutura 5C	37
Figura 3.1	Arquitetura geral da proposta	38
Figura 3.2	Arquitetura Proposta.....	39
Figura 3.3	Objeto Tipo de Sistema Resiliente	41
Figura 3.4	Gráfico de Máquina de Estados.....	42
Figura 3.5	Modelo de Informação da máquina de estados	43
Figura 3.6	Eventos de Degradação e recuperação nas simulações	44
Figura 3.7	Rede de planejamento.....	45
Figura 4.1	Planta Smar-PD3	47
Figura 4.2	Topologia do estudo de caso	48
Figura 4.3	Esquerda: Objetos do servidor; Direita: Objetos da SMARPD3.....	49
Figura 4.4	Variável de Processo com Controlador PID	49
Figura 4.5	Variável de Processo com Controlador Fuzzy	50
Figura 4.6	Desempenhos dos controladores	50
Figura 4.7	Plano de Reconfiguração	51
Figura 4.8	Desempenho com <i>Auto Configuração</i>	51
Figura 4.9	Variável de Processo com <i>Auto Configuração</i>	51
Figura 4.10	Tempos de Recuperação com tolerância de 0,1	52
Figura 4.11	Tempos de Recuperação com tolerância de 0,2	53

LISTA DE QUADROS

Quadro 2.1 Tempos de Resiliência	31
Quadro 2.2 Resumo de Trabalhos Relacionados	34
Quadro 2.3 Relação das tarefas MAPE-K e as dimensões da taxonomia de Auto Adaptação.....	37
Quadro 3.1 Estados e Transições da Máquina de Estados.....	42

LISTA DE ABREVIATURAS

ABSA	Auto Adaptação Baseada em Arquitetura
CPS	Sistemas Ciberfísicos – Cyber Physical System
DPWS	Device Profile for Web Services – Serviços Web para Perfis de Dispositivos
DoS	Denial of Service – Denegação de Serviço
ERP	Enterprise Resource Planning - Planejamento dos Recursos da Empresa
FSM	Finite State Machine – Máquina de Estados Finitos
ICS	Industrial Control System – Sistema de Control Industrial
IoT	Internet of Things – Internet das Coisas
MAS	Multi-Agent System – Sistema Multi Agente
MES	Manufacturing Execution System - Sistema de Execução de Manufatura
MQTT	Message Queue Telemetry Transport
MTtF	Mean Time to Failure – Tempo Meio de Falha
OPC-UA	OLE for Process Control Unified Architecture
PHM	Prognostic and health Management – Gerenciamento o de Saúde e Prognóstico
RICS	Resilience Industrial Control System – Sistemas de Control Industrial de Resiliência
SAS	Self-Adaptive System – Sistemas Auto Adaptativos
SCADA	Supervisory Control and Data Adquisition
SCM	Supply Chain Management - Gestão de Cadeia de Suprimentos
SOA	Service Oriented Architecture – Arquitetura Orientada a Serviços
SOAP	Simple Object Access Protocol– Protocolo de Acesso a Objetos Simples
SVM	Support Vector Machine – Máquina de Vectors de Suporte
TIC	Tecnologias da Informação e Comunicação
WS	Serviços Web – Web Services
WSDL	Web Service Description Language – Linguagem de Descrição de Serviços Web
XML	Extensible Mark Up Language

1 INTRODUÇÃO

1.1 Contextualização e Justificativa

A Indústria 4.0, procura usar tecnologias avançadas (V. GARCÍA; IRISARRI; PÉREZ, 2017), desenvolvendo sistemas que podem adaptar-se autonomamente e atingindo os seus objetivos de alto nível (WEYNS, 2017). Os desafios do CPS na Indústria 4.0 consideram vários pontos, como maior integração entre máquinas e ferramentas e desenvolvimento de sistemas autônomos com capacidade de adaptação.

Os sistemas auto adaptativos são usados comumente em máquinas e processos inteligentes, onde a maioria das decisões críticas e otimizações são realizadas sem a intervenção do operador, apresentando propriedades de autoconhecimento, autodiagnostico e autoconfiguração. (DUMITRACHE; CARAMIHAI; STANESCU, 2013).

A auto-adaptação tem desafios como a quantificação da confiabilidade para saber que está em uma distribuição espaço, a garantia de consistência semântica, a manutenção de desempenho em tempo real, a tomada de decisão com dados incompletos, eventos incoerentes, imprecisos e desordenados; e finalmente, a eliminação de erros que perturbam a decisão do sistema de controle (ZHOU et al., 2017a). A auto-adaptação pode ser garantida com Sistemas Ciber-Físicos, gerando arquiteturas especializadas que permitem atacar a crescente complexidade, a incerteza e dinamismo nos sistemas (GARCÍA-VALLS et al., 2014), além da integração dos mundos digitais com o mundo físico (FLEISCHMANN; KOHL; FRANKE, 2016).

O desenvolvimento de CPS e seus processos precisa definir uma arquitetura de base, capaz de descrever com detalhes a sua estrutura. Uma arquitetura de CPS, que facilita a auto adaptação através de camadas é a arquitetura 5C, proposta por (LEE; BAGHERI; KAO, 2015). A arquitetura 5C agrega os componentes de um CPS em 5 camadas, onde a primeira camada "*Conexão Inteligente*" é formada por sensores que capturam os dados brutos do ambiente e agrupa os atuadores responsáveis pelas ações. A segunda camada de "*Conversão*", passa os dados brutos a dados rotulados, apresentando um sentido semântico para descrever a estrutura do sistema. A camada "*Cibernética*" cria um modelo virtual do sistema, usando algoritmos para analisar e controlar sua operação, baseados em dados históricos. A camada "*Cognição*" analisa todo o sistema, tendo como base um modelo virtual do sistema, que contem seus padrões para automatização de tarefas. A última camada "*Configuração*" obtém informações da pirâmide toda, para configurar e parametrizar o controle da planta, especificamente fazendo um feedback de controle do espaço cibernético até o espaço físico. A camada, atua como um sistema de controle de resiliência RCS (*Resilience Control System*).

A camada de *Configuração* também pretende garantir interoperabilidade entre os componentes do sistema, podendo ser garantida pelo uso de topologias descritivas, padrões e

protocolos de comunicação industrial, para uma descentralização orquestração. Além disso, o sistema deve permanecer protegidos contra a ocorrência de falhas e ameaças que podem ser imprevisíveis no momento do design ou surgem durante o tempo de execução (RATASICH et al., 2018). A interoperabilidade pode ser alcançada, através de um modelo de aplicação geral que cubra a funcionalidade dos sistemas. (FERNBACH; GRANZER; KASTNER, 2011).

Além da autoconfiguração e a interoperabilidade a camada de *Configuração* propõe um sistema de controle com resiliência. A resiliência pode ser conceituada como o conjunto de políticas a serem aplicadas para projetar os sistemas de computação que podem se adaptar dinamicamente de maneira previsível a eventos inesperados (GARCIA; LIN; MEERKOV, 2012).

MAPE-K é um modelo conhecido de desenvolvimento de sistemas auto adaptativos, o qual contém um loop de controle inteligente para supervisão de sistemas. Primeiramente a proposta propõe um método automatizado para coleta de dados e requisitos necessários do sistema, analisando estas informações e determinando se algo precisa ser modificado. Posteriormente cria planos ou sequência de ações que especificam as mudanças necessárias, executando-as finalmente (IBM, 2006).

Especificamente, o trabalho visa propor uma arquitetura interoperabilidade para a autoconfiguração resiliente de Sistemas Ciberfísicos. Posteriormente, a aplicação em um estudo de caso e na avaliação dos tempos de recuperação em uma planta didática.

1.2 Objetivos

1.2.1 Objetivo Geral

Avaliar os conceitos de resiliência e suas métricas, focadas na auto-configuração de Sistemas Ciberfísicos, através de uma metodologia para o desenvolvimento de arquiteturas baseadas em OPC-UA, que permita implementação da camada de Configuração da arquitetura 5C.

1.2.2 Objetivos Específicos

- Definir os conceitos da Indústria 4.0, mais especificamente os Sistemas Ciberfísicos e suas arquiteturas CPS, visando a Auto Adaptação e Resiliência.
- Identificar os protocolos industriais para o desenvolvimento de aplicativos de controle de CPS, baseados na camada de Configuração da arquitetura 5C.
- Propor uma arquitetura que permita a implementação da camada Configuração em uma estrutura 5C, levando em conta os conceitos de resiliência e autoconfiguração.
- Aplicar a metodologia em um estudo de caso na planta didática de automação SMAR PD3, locada no Centro de Ciências Computacionais.
- Avaliar os resultados obtidos com o funcionamento da metodologia, validando a interoperabilidade e autoconfiguração resiliente.

1.3 Organização do Trabalho

O presente trabalho está organizado da seguinte maneira. O *Capítulo 1*, contém uma visão geral do tema, mostrando a justificativa, seus objetivos e a sua organização. O *Capítulo 2*, descreve a fundamentação dos sistemas Ciberfísicos e a arquitetura 5C, além dos conceitos de interoperabilidade, resiliência e auto adaptação. O *Capítulo 3*, descreve a arquitetura proposta levando em conta o modelo *MAPE-K* para a implementação na camada de *Configuração* da arquitetura 5C. O *Capítulo 5*, mostra os resultados do estudo numa planta didática. O *Capítulo 6*, apresenta as conclusões baseadas na resiliência, autoconfiguração e interoperabilidade, focados na recuperação de Sistemas Ciberfísicos ante eventos indesejados mediante reconfigurações proativas.

2 FUNDAMENTAÇÃO TEÓRICA

A análise da literatura pesquisada, que definiu a base conceitual da pesquisa, considera conceitos como *Indústria 4.0*, *Sistemas Ciber-físicos*, *Arquitetura 5C*, *Resiliência*, *Auto Adaptação*, *MAPE-K* e o *Padrão OPC-UA*.

2.1 Indústria 4.0

O processo industrial transforma matéria-prima em produtos comerciáveis, lidando ao longo da história com algumas revoluções tecnológicas. A indústria cada dia mais mecanizada e automatizada, visa a sistemas de produção modulares, customizáveis e individuais, mantendo a eficiência e os custos da produção em massa.

Desde século XVIII a primeira revolução industrial foi a fundamentada na introdução da produção mecânica intensificando-se ao longo de todo o século XIX. Na década de 1870, a eletrificação e a divisão do trabalho, induziram à segunda revolução industrial. A terceira revolução industrial, também chamada de “revolução digital”, surgiu na década de 1970, quando a eletrônica avançada e a tecnologia da informação desenvolveram ainda mais a automação dos processos de produção. Agora, paradigma chamado de “*Indústria 4.0*”, promoveu a ideia de virtualização com autonomia e auto comportamento das máquinas, como uma abordagem para reforçar o poder competitivo da indústria. A Figura 2.1 apresenta o progresso industrial na perspectiva histórica (OZTEMEL; GURSEV, 2018).

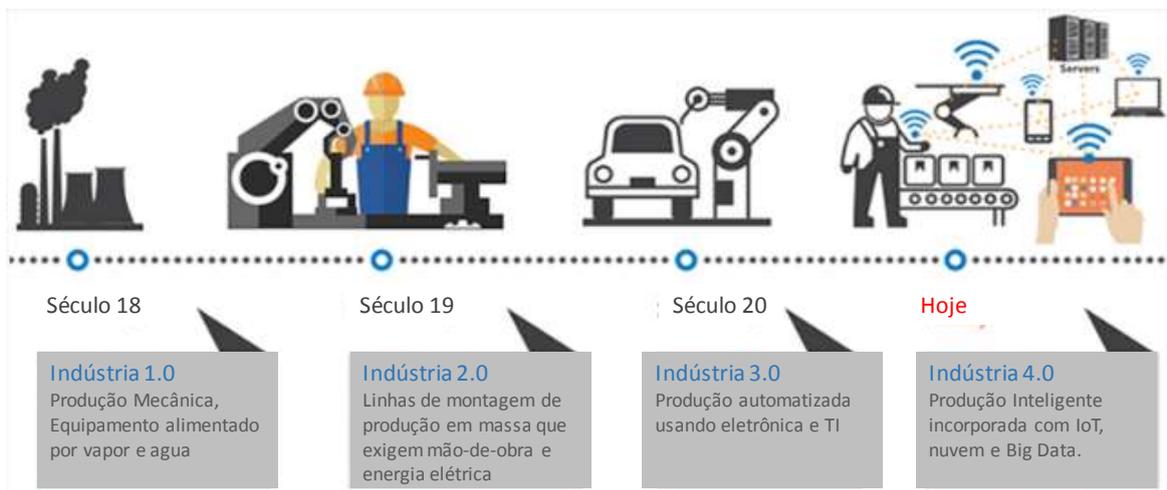


Figura 2.1 Perspectiva histórica da Revolução Industrial
 Fonte: (OZTEMEL; GURSEV, 2018) (Tradução do autor)

Os termos “*Indústria 4.0*”; “*Smart-Factory*”; “*Factory of the future*” são termos que descrevem uma visão das fábricas no futuro, considerando fábricas muito mais inteligentes, flexíveis, dinâmicas e ágeis (COELHO, 2016). As manufaturas serão equipadas com mais e mais capacidades de auto avaliação através do tempo. A principal ideia por trás da Indústria 4.0 é o estudo de aplicativos para o desenvolvimento de fábricas inteligentes, integradas e totalmente autônomas (OZTEMEL; GURSEV, 2018). Os processos na Indústria 4.0, propõem uma nova maneira de organização e controle nos sistemas de automação, ver Figura 2.2, onde o objetivo é a produção em massa, contando com uma estrutura de componentes distribuídos no ambiente de produção.

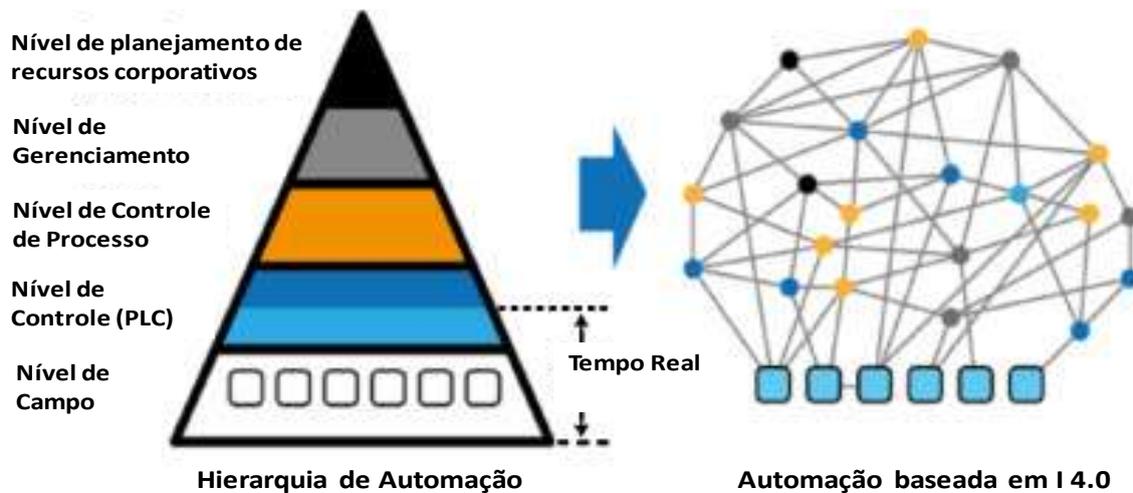


Figura 2.2 Decomposição da hierarquia de automação com serviços distribuídos
 Fonte: (LEE; BAGHERI; KAO, 2015) (Tradução do autor)

A Indústria 4.0 define uma perspectiva onde a máquina domina o processo de manufatura digital. Portanto para alcançar uma transformação bem-sucedida deve-se possuir um roteiro claro de implementação (OZTEMEL; GURSEV, 2018). Enquanto alguns dos pesquisadores focam sua atenção na digitalização, outros consideram o aspecto da comunicação como a característica mais importante na estrutura de produção. Assim outros também apostam pela inteligência e autonomia.

A convergência das tecnologias da informação com a sensorial e a robótica estão transformando a internet tradicional na Internet das Coisas, tendo novo cenário de oportunidades baseadas no uso de tecnologias da informação (OZTEMEL; GURSEV, 2018). A Indústria 4.0 é guiada pelo uso de CPS (*Sistemas Ciber-físicos*). Estes sistemas envolvem monitoramento e interação entre homem e máquina (LEE; BAGHERI; KAO, 2015).

2.2 Sistemas Ciberfísicos

Os Sistemas Ciber-físicos são sistemas que integram capacidades computacionais e físicas usando redes de comunicação entre sensores e atuadores (BORDEL et al., 2018). Devido a simplicidade do conceito, são consideradas outras definições.

- Os CPS podem ser considerados como sistemas inteligentes os quais usam tecnologias cibernéticas e interagem com componentes físicos, apresentando uma forte combinação de elementos físicos e computacionais, integrando computação, comunicação e controle sobre um sistema de informação (LEITÃO; COLOMBO; KARNOUSKOS, 2016).
- Um CPS é uma integração de computação com elementos físicos nos processos físicos, geralmente com ciclos de retroalimentação, os quais afetam o seu controle (BORDEL et al., 2017).
- Os Sistemas Ciberfísicos, representam uma integração da computação com sistemas físicos, com o objetivo de agrupar objetos físicos num sistema de computação e sistemas de redes, a ser usados como objetos virtuais (DUMITRACHE; CARAMIHAI; STANESCU, 2013).

O desenvolvimento de CPS industriais é garantido comumente por várias arquiteturas, as dois mais comuns, Sistemas Multi-Agente, MAS (*Multi-Agent Systems*) e Arquitetura Orientada a Serviços SOA (*Service Oriented Architecture*) (ZHOU et al., 2017a).

A Arquitetura Multi-Agente é baseada em uma sociedade de agentes que representam os objetos de um sistema, capazes de interagir para obter seus objetivos individuais quando eles não contam com conhecimento e/ou habilidades suficientes para alcançar os objetivos gerais. Um agente é um componente autônomo, que representa objetos físicos ou lógicos no sistema, capaz de atuar para obter seus objetivos além de interagir com outros agentes (LEITÃO; COLOMBO; KARNOUSKOS, 2016). As MAS são também definidos como um paradigma de comunicação introduzido no campo de inteligência artificial distribuída, caracterizado pela descentralização e execução paralela de atividades baseadas em agentes autônomos (VOGEL-HEUSER; LEE; LEITÃO, 2015).

Por outro lado as arquiteturas orientadas a serviços permitem que as aplicações distribuídas obtendo interoperabilidade e reutilização entre software através de serviços (VIZCARRONDO et al., 2017). Os princípios da SOA promovem aplicações onde todas as funções são implementadas e expostas como serviços a serem descobertos e usadas por outros elementos da rede (CÂNDIDO et al., 2010). Uma metodologia geral de engenharia para desenvolver e implementar uma SOA para CPS, começará com a especificação de serviços dedicados ao dispositivo e especificações de hardware do sistema. Depois será completada com a especificação e implementação de métodos e ferramentas que permitam a composição e a orquestração desses serviços (LEITÃO; COLOMBO; KARNOUSKOS, 2016).

O desenvolvimento de CPS usa arquiteturas para entender seu processos e fluxos internos descrevendo a sua estrutura interna. Muitas arquiteturas foram propostas para a implementação de

Sistemas Ciberfísicos, por exemplo, as arquiteturas NIST (GRIFFOR et al., 2017), OSMOSE (GUSMEROLI et al., [s.d.]), RAMI 4.0 (KOSCHNICK; HANKEL; REXROTH, 2015), incluindo a arquitetura 5C (LEE; BAGHERI; KAO, 2015). O desenvolvimento de CPS e seus processos precisam definir uma arquitetura de base, capaz de descrever com detalhes a sua estrutura, portanto apresenta-se logo a camada 5C de Jay Lee, como a base da proposta.

2.2.1 Arquitetura 5C de Sistemas Ciber-Físicos

A arquitetura 5C consiste em uma descrição de funcionalidades distribuídas através de 5 níveis com alguma indicação do fluxo de informação, para ser usado no âmbito da manufatura. A Figura 2.3 apresenta a visão geral da arquitetura 5C a seguir, descreve-se em detalhes cada camada.

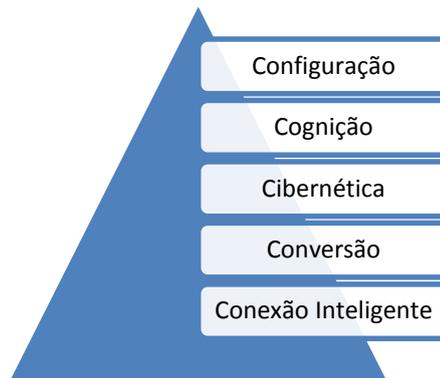


Figura 2.3Arquitetura 5C

Fonte: (LEE; BAGHERI; KAO, 2015)(Tradução do autor)

Camada de Conexão Inteligente

A Aquisição de dados confiáveis de máquinas e seus componentes, representa o primeiro passo no desenvolvimento de um CPS-5C, medindo diretamente as condições do ambiente através de sensores ou obtendo dados de outro tipo de fonte como sistemas de controle ou corporativos. Dois fatores importantes nesse nível a ser considerados são, primeiramente que para trabalhar com vários tipos de dados é necessário um método para gerenciar a aquisição de dados; por outro lado, a seleção de sensores apropriados com tipo e especificação para tornar os sistemas mais inteligentes.

Camada de Conversão

A conversão de dados para informação é responsável por converter os dados vindos das máquinas ou componentes em informações relevantes, representando a condição dos sistemas de máquinas monitorados. Os dados devem ser convertidos em informações significativas para uma aplicação no mundo real, incluindo avaliação de integridade e diagnóstico de falhas, usando ferramentas e metodologias disponíveis para a transformação de dados em informação.

Camada Cibernética

O nível cibernético atua como núcleo central de informações nesta arquitetura, integrando os dados enviados de cada máquina conectada e formando uma rede de informações no sistema inteiro. Os dados coletados em massa são analisados, garantindo a extração de informações adicionais, que permitem ter uma visão dos status das máquinas individualmente como o seu desempenho no grupo. As análises além disso fornecem às máquinas propriedades de auto-comparação, onde o desempenho de uma única máquina pode ser comparado com as outras.

Um elemento importante desta camada é o "*Digital Twin*", entendido como uma representação virtual do sistema físico capaz de atuar como base de conhecimento para decisões, simulações e estudo de comportamento no tempo (LEE; BAGHERI; KAO, 2015).

Camada de Cognição

O nível de cognição, aproveita a informação gerada nas camadas anteriores, diagnosticando seu potencial fracasso e conhecendo a sua degradação de forma antecipada, além de fornecer informações de raciocínio para mitigar os efeitos das degradações em diferentes componentes do sistema. Baseado no aprendizado adaptativo e avaliação histórica de saúde, o sistema pode utilizar alguns algoritmos de previsão que permitam identificar falhas potenciais e o as medidas de mitigação.

A apresentação adequada do conhecimento adquirido aos usuários experientes apoia a decisão correta a ser tomada. As informações comparativas entre máquinas, apoiam a decisão sobre a prioridade das tarefas para otimizar. Para este nível, os gráficos de informações são necessários para transferir completamente o conhecimento adquirido aos usuários.

Camada de Configuração

O nível de configuração faz a retroalimentação do espaço cibernético para o espaço físico e atua como um controle, para tornar as máquinas autoconfiguráveis e auto adaptáveis. A camada, atua como um sistema de controle de resiliência RCS (*Resilience Control System*) para aplicar decisões corretivas e preventivas, que foram calculadas na camada cognitiva do sistema.

Propõe-se que o sistema monitore sua condição de integridade e identifique as falhas antecipadas, enviando informações de monitoramento e reconfigurações para o nível de operação e de gerenciamento. Os operadores e gerentes de fábrica podem tomar a decisão correta com base nas informações de manutenção e ao mesmo tempo, as máquinas podem ajustar sua programação de fabricação, a fim de reduzir perdas por mau funcionamento da máquina, alcançando um sistema resiliente.

Segundo (LEE; BAGHERI; KAO, 2015), a camada propõe três características principais, que são autoconfiguração, interoperabilidade e resiliência.

A autoconfiguração é uma característica de um sistema auto adaptativo que pode mudar seus atributos internos como uma resposta automática devido à troca de contextos ou eventos. A interoperabilidade permite que dispositivos físicos de diferentes fabricantes e tecnologias funcionem sem restrições de comunicação. A resiliência permite que o sistema mantenha seu estado operacional normal especificamente focado nos tempos de recuperação após um evento indesejado. Estes três conceitos são apresentados com maior detalhe nas seções seguinte.

Até agora, a camada de *Configuração* não tem uma metodologia proposta que aborde claramente a integração destes três termos, portanto, a base deste trabalho é a discussão deles, como desafios para a definição de uma arquitetura.

2.3 Auto Adaptação

A autoconfiguração é definida como uma propriedade de auto-adaptação, abordando problemas relacionados à quantificação da confiabilidade, garantias de coerência semântica, manutenção de desempenho em tempo de execução, decisões inconsistentes, bem como eventos desordenados e a eliminação de erros que perturbam o controle do sistema (FERNBACH; GRANZER; KASTNER, 2011).

Auto-adaptação refere-se a um modo de operação autônoma, alcançar a adaptação do sistema para cumprir seus objetivos. Sistemas auto adaptativos podem ajustar seu comportamento em resposta a sua percepção do ambiente e variáveis internas (WEYNS, 2017). As estratégias de auto-adaptação utilizadas para o melhoramento de CPS, podem torna-lo altamente flexível em função da estrutura topológica. Mas também dificulta a análise e a manutenção da confiabilidade (ZHOU et al., 2017b). Portanto é preciso definir concretamente onde e quando é feita a autoadaptação.

Os Sistemas auto adaptativos têm um amplo espectro de implementações, devido à natureza abstrata do seu conceito, portanto as taxonomias são necessárias para localizar o escopo do sistema de acordo com suas características e requisitos.

2.3.1 Taxonomia de Auto Adaptação

A presente seção apresenta uma taxonomia abrangente para a auto-adaptação, mostrando uma visão geral da proposta feita por (KRUPITZER et al., 2015). As características mais importantes de autoadaptação, contém dimensões de tempo, causa, nível, técnica e controle.

Causa

A causa é o motivo pela qual é gerada a adaptação, identificando-se três tipos principais. Primeiramente as mudanças em *Recursos Técnicos*, por exemplo, um defeito de um componente de *hardware*, falha de software ou a disponibilidade de uma conexão na rede alternativa. A segunda, uma alteração no *Contexto*, por exemplo, o estado de uma variável. Por último as *Alterações do Usuário*, por exemplo, uma configuração ou preferências. A inclusão explícita do usuário como motivo de adaptação não é mencionada em outras taxonomias por não se considerar como “auto” adaptações.

Nível

O nível refere-se à camada de automação onde vai ser implementada a mudança. O nível mais baixo é nos Recursos Técnicos, podendo alterar os componentes físicos e virtuais. O nível de Comunicação é visto em duas perspectivas, por um lado a infraestrutura de rede garantindo a conexão da rede física e por outro lado os padrões de comunicação garantindo a conexão lógica. O nível Software de sistema, baseada em trocas a nível de sistemas operativos ou middleware. As Aplicações Adaptativas, no topo dos níveis, adaptam o sistema pelo gerenciamento de funções.

Tempo

O Tempo permite saber o momento quando o sistema vai se adaptar, com respeito ao tempo do evento. As configurações tradicionais são depois de que os eventos acontecem e são consideradas reativas. Por outro lado, quando os eventos têm sido identificados antecipadamente por ferramentas de simulação, a reconfiguração é considerada proativa.

A engenharia resiliente, promove uma abordagem de reconfiguração proativa, procurando maneiras de aumentar a capacidade do sistema de rastrear antecipadamente os riscos. Para enfrentar esse desafio, os métodos de segurança convencionais, como a análise de risco, devem ser usados. No entanto, a resiliência deve ser implementada para garantir a rápida recuperação e adaptação a um evento adverso (BASTAN; BENESL; FIEDLER, 2018). A engenharia de resiliência propõe uma mudança de estratégia de métodos reativos a métodos proativos, avaliando possíveis ameaças (WEI; JI, 2010).

Técnica

A técnica define como vai mudar o sistema, ressaltando as três técnicas mais importantes. Mudança de *Parâmetros*, mudança de *Estrutura* topológica que permite a troca de componentes do sistema em tempo de execução e mudança de *Contexto*, baseada na alteração do ambiente através de atuadores.

Controle de Adaptação

O controle contém duas abordagens, sendo elas interna e externa. A interna unifica os componentes lógicos de adaptação com os recursos físicos do sistema, por exemplo um dispositivo inteligente ou um agente. A abordagem externa divide o sistema em lógica de adaptação e recursos gerenciados, o que aumenta a capacidade de manutenção por meio da modularização, por exemplo o uso de arquiteturas orientadas a serviços.

A unidade de controle precisa de métricas baseadas em restrições para medir sua efetividade, usando modelos, regras, políticas, objetivos e funções de utilidade.

O grau de descentralização, pode ser centralizado ou descentralizado. Um controle de adaptação centralizado pode ser uma solução para sistemas com uma pequena quantidade de recursos para gerenciar. Quando se trata de grandes sistemas com muitos componentes para gerenciar, uma abordagem descentralizada para uma divisão das responsabilidades pode melhorar o desempenho do sistema.

2.3.2 MAPE-K

A auto-adaptação tem sido reconhecida como uma abordagem eficaz para lidar com sistemas complexos, que têm incerteza e alto dinamismo. Uma engenharia bem reconhecida de abordagem é realizar a auto-adaptação através de um gerente autônomo (loop de controle com retroalimentação) chamado MAPE-K (ARCAINI; RICCOBENE; SCANDURRA, 2015). MAPE-K é uma solução que implementa um método automatizado de coletar os detalhes necessários do sistema; analisar aqueles detalhes para determinar se algo precisa mudar; criar um plano ou sequência de ações que especificam as mudanças necessárias e realizar essas ações (IBM, 2006).

O desenvolvimento de um sistema auto adaptativo, considera o uso do modelo MAPE-K proposto por (IBM, 2006). O modelo permite realizar circuitos de controle num sistema, conforme à Figura 2.4, utilizando cinco módulos principais, os quais são, *Monitoramento*, *Conhecimento*, *Análise*, *Planejamento* e *Execução*.

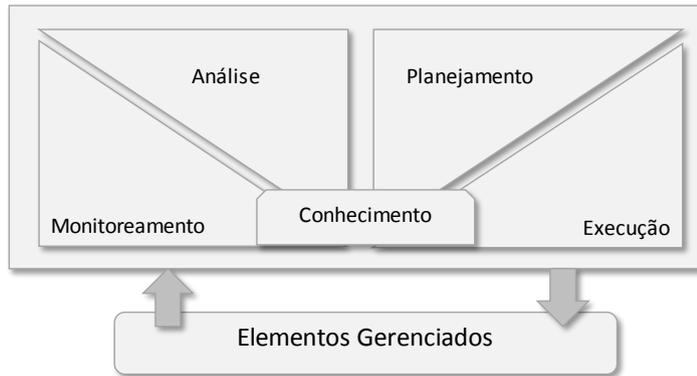


Figura 2.4 Estrutura de Gerenciador Auto Adaptativo
 Fonte: (IBM, 2006) (Tradução do autor)

O módulo de *Monitoramento* fornece os mecanismos que coletam, agregam, filtram e relatam detalhes (como métricas e topologias) coletados de um recurso gerenciado (IBM, 2006). Os parâmetros das máquinas podem refletir o status de produção, além de todas as características dos componentes dos CPS (FLEISCHMANN; KOHL; FRANKE, 2016). A detecção de comportamento anômalo em sistemas industriais pode ser baseada em conhecimento, sinais ou modelos.

A função de *Análise* fornece os mecanismos que correlacionam e modelam situações complexas (por exemplo, modelos de previsão e enfileiramento de séries temporais). Esses mecanismos permitem que o gerente autônomo aprenda sobre o ambiente de TI e ajude a prever situações futuras (IBM, 2006). O objetivo da fase de análise é descobrir as características das operações e seus estados, dividindo séries temporais em um conjunto de estados de destino (KISSKALT et al., 2018).

A função de *Planejamento* fornece os mecanismos que constroem as ações necessárias para atingir metas e objetivos. O mecanismo de planejamento usa informações sobre políticas para orientar seu trabalho (IBM, 2006). Um sistema pode ser adaptado alterando seus parâmetros ou sua estrutura. A adaptação de parâmetros requer conhecimento sobre o algoritmo subjacente do componente errôneo e, portanto, é tipicamente executada pelo próprio componente ou dentro de um subsistema (RATASICH et al., 2018).

Execução fornece os mecanismos que controlam o cumprimento de um plano com considerações para atualizações dinâmicas (IBM, 2006).

Uma fonte de *Conhecimento* é uma implementação de um registro, dicionário, banco de dados ou outro repositório que fornece acesso ao conhecimento de acordo com as interfaces prescritas pela arquitetura (IBM, 2006). Os eventos relevantes podem ser armazenados numa base de dados para referências futuras (ZHAI et al., 2004).

2.4 Interoperabilidade

A interoperabilidade pode ser alcançada através de protocolos de comunicação industrial, abstraindo as tecnologias de rede, criando uma visão uniforme do processo dados e permitindo a comunicação entre dispositivos com diferentes tecnologias (FERNBACH; GRANZER; KASTNER, 2011). Os padrões de comunicação são comumente aplicados para descrição e troca de informação em aplicações de engenharia, permitindo a comunicação on-line de sinais de processo.

2.4.1 Protocolos industriais para CPS

Devido à quantidade de protocolos de comunicação, fabricantes e infraestruturas de TI, é requerido um protocolo que permita a unificação e integração de componentes de um sistema. Nesse contexto algumas iniciativas desenvolvidas no intuito de oferecer uma linguagem universal para descrição, comunicação e integração de elementos no âmbito industrial. Dois destes protocolos são o Perfil de Dispositivos para Serviços da Web DPWS (*Device Profile for Web Services*) e a Arquitetura Unificada OPC-UA (*OPC Unified Architecture*). Na literatura encontra-se um conjunto de trabalhos comparam as principais características dessas duas especificações, a fim de identificar seus pontos fortes e fracos, decidindo qual é mais adequada para a construção de SOA em nível de dispositivo (CÂNDIDO et al., 2010).

O OPC-UA constitui-se de um padrão cliente-servidor, em que o servidor é usado para expor informações no modelo de informação e ser acessadas através de clientes. OPC-UA contém pilhas de comunicação usadas pelo servidor e cliente. As pilhas de comunicação estão contidas, exceto nos casos em que os servidores são concatenados em uma arquitetura em camadas, sendo o servidor parte de uma camada superior e o cliente na camada inferior. O DPWS, em contraste, depende principalmente de um padrão par-a-par no nível do dispositivo, e padrão cliente-servidor para fornecer acesso a partir de camadas de gerenciamento de alto nível.

A especificação dos recursos técnicos no domínio de DPWS, precisa de um gerenciamento dedicado baseada em WS. Por outro lado, OPC-UA está focada no gerenciamento e supervisão de recursos definindo uma arquitetura geral e extensível para interoperabilidade com modelos padrão de informação. Os dois contêm especificações de *Segurança* e *Canal Seguro*.

O DPWS não inclui nenhum formato de troca binária, mas a especificação é aberta para permitir essa extensão. Por outro lado, OPC-UA define uma codificação binária portátil para otimizar o tamanho da mensagem e o tempo de processamento ao trocar grandes conjuntos de dados.

Neste contexto, a Arquitetura Unificada do OPC (OPC-UA) é um dos padrões mais importantes que suportam o WS (FERNBACH; GRANZER; KASTNER, 2011), e é o protocolo adotado nesta proposta.

2.4.2 Arquitetura Unificada OPC

O OPC clássico, foi desenvolvido em 1996 para solucionar problemas de interoperabilidade em sistemas de automação industrial, integrando dados entre máquinas, equipamentos e dispositivos diretamente no chão de fábrica. Essa primeira versão estava associada a tecnologias proprietárias tais como OLE, COM e DCOM. Baseado no OPC clássico foi proposta uma arquitetura unificada (*Unified Architecture-UA*), tornando o OPC independente de tecnologias proprietárias para se converter um padrão. Assim, em 2006 foi feito seu desenvolvimento em Java, .NET e C baseado no paradigma orientado a serviços (*SOA, Service Oriented Architecture*), sendo acrescentadas novas tecnologias como XML (*Extensible Mark Up Language*) (OPC FOUNDATION, 2018).

O OPC-UA virou independente de plataforma, permitindo a troca de mensagens entre vários tipos de sistemas e dispositivos ou entre Clientes e Servidores, em muitos tipos de redes. O protocolo suporta comunicação robusta e segura através de serviços, os quais asseguram a identidade dos clientes e servidores, resistindo a possíveis ataques. A informação é transmitida utilizando os modelos de informação padrão de OPC-UA e modelos de informação de proprietários (OPC FOUNDATION, 2018).

O servidor integra dados, alarmes, eventos, histórico em seu modelo de informação, além disso fornece acesso a eles usando um conjunto integrado de serviços. Os serviços incluem um modelo de informação de segurança, usado para conexões. Cada dispositivo ou elemento físico da planta é representado como um objeto contido em um nodo, tendo propriedades e métodos. Os nodos possuem atributos e referências como conexões lógicas que ampliam sua definição e outras propriedades.

OPC-UA usa uma pilha de comunicação que consiste em dois mapeamentos diferentes (XML Web Services e UA Native) atendendo a grande variedade de necessidades de integração, presentes no ambiente moderno de plantas industriais. A pilha de comunicação, oculta esses mapeamentos por trás de uma API bem definida (*Application Programming Interface*) que torna transparente o uso de protocolos de comunicação disponíveis. Isso significa que os servidores OPC-UA suportam os dois protocolos simultaneamente e os clientes podem escolher qual se adapta melhor à aplicação atual (HANNELIUS; SALMENPERÄ; KUIKKA, 2008). Como mostrado na Figura 2.5, o OPC-UA pode ser mapeado usando padrões da Web, incluindo XML, WSDL, SOAP e outras especificações WS, juntamente com outras definidas por OPC-UA (MENDES et al., 2010).

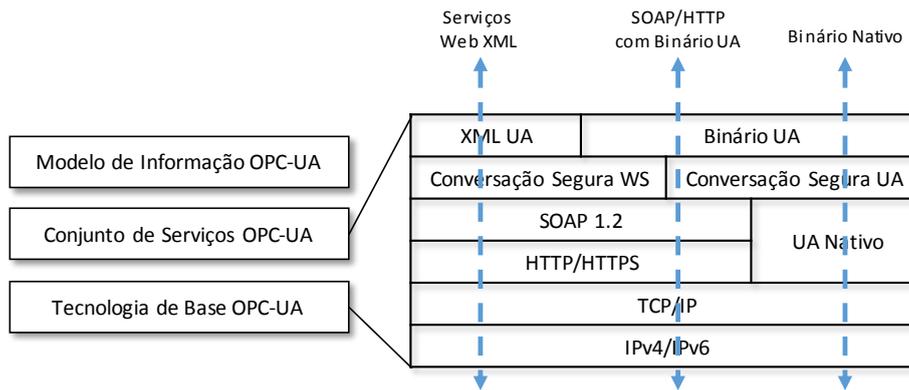


Figura 2.5 Tecnologias de Comunicação OPC-UA
Fonte: (CÂNDIDO et al., 2010) (Tradução Autor)

A melhora do desempenho em baixo nível, é garantida usando protocolos adicionais, tais como, *Codificação Binária*, *UA Nativo* e *TCP*. A nível de comunicação intermediário os protocolos de transporte e codificações de informações dependendo dos requisitos do aplicativo, por exemplo mensagens XML e Codificação Binária transportadas por mensagens *SOAP*. Finalmente para o nível de aplicações a *Codificação Binária* é incorporada em *XML* para se comunicar através de serviços (MENDES et al., 2010).

O *OPC-UA* trabalha como middleware de comunicação, padronizando o transporte de informação para sistemas de automação e conecta tarefas de engenharia off-line e a comunicação em tempo de execução. O protocolo define como as informações são trocadas entre os sistemas envolvidos, lidando com o gerenciamento de dados e comunicações (MONOSTORI et al., 2016).

O conjunto de Objetos e informações que o Servidor disponibiliza para os clientes é conhecido como Espaço de Endereço, representado como um conjunto de nós conectados por referências. A interoperabilidade entre Clientes e Servidores é garantida estruturando hierarquicamente, como uma rede inter-relacionada de nós mas podem existir referências não hierárquicas (OPC FOUNDATION, 2018).

Conjunto de Especificações de OPC-UA

As definições dos principais recursos de OPC-UA são ordenadas por especificações, (ver Figura 2.6), as quais são especificações de *núcleo*, *tipos de acesso* e *utilidades*.

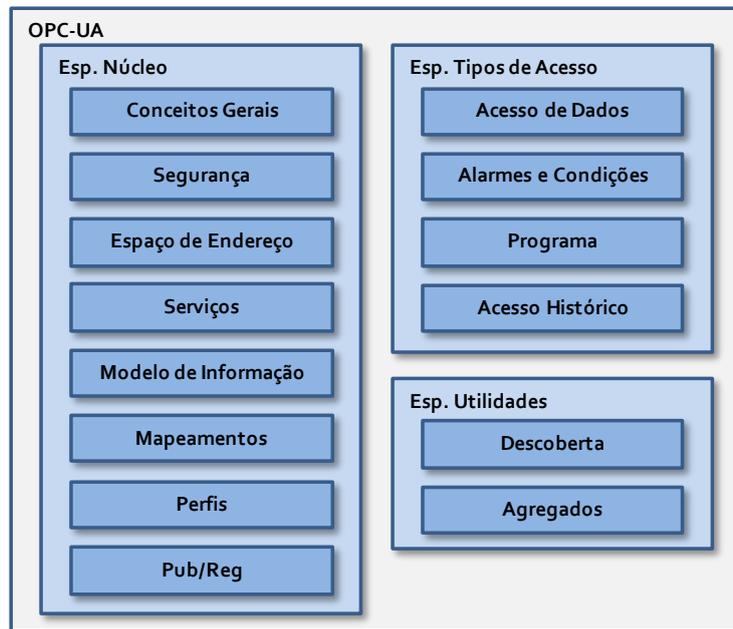


Figura 2.6 Estrutura de Especificações OPC-UA
 Fonte: (OPC FOUNDATION, 2018) Tradução do autor

As especificações de *Núcleo*, contém *Conceitos Gerais*, para definir a ordem de leitura e regras fundamentais para usar o protocolo. A especificação de *Segurança*, apresenta as estratégias a ameaças advindas dos ambientes físicos, do hardware e do software. O *Espaço de Endereços* descreve a padronização de objetos, referências e métodos. Os *Serviços* contêm umas capacidades que podem ser aproveitadas para a comunicação de Servidores e Clientes. O *Modelo de Informação* descreve um modelo geral a ser usado em aplicações básicas do protocolo. O *Mapeamento* faz a relação entre modelos de informação de segurança, serviços e estruturas de dados. Os *Perfis* fornecem funcionalidades a serem usadas por usuários específicos. *Publicação/Subscrição* é utilizada para a publicação de informações e podem ser acessadas por usuários registrados.

O grupo de *Especificações de Tipos de Acesso* agrupa *Acesso de Dados*, definindo o modelo de informação com os tipos de variáveis, descrições complementares de nodos e atributos necessários. Os *Alarmes e Condições*, representam o modelo de informações tarefas para geração de notificações baseadas em monitoramentos de variáveis. Os *Programas* descrevem o modelo de informação para soluções de engenharia que permitem obter um resultado específico e por último *Histórico* descreve seu modelo de informação para acesso a informação histórica.

O grupo de Especificações de *Utilidades*, envolve primeiramente a especificação de *Descoberta*, o qual especifica como os Clientes e Servidores interagem nas primeiras trocas de informação e processos de reconhecimento. A especificação de *Agregados* define o modelo de informação de resultados a funcionalidades e utilidades matemáticas que fornece o protocolo. Para mais detalhes, verificar as especificações da Fundação OPC-UA.

Serviços de OPC-UA

Os serviços de OPC-UA permitem ao cliente/servidor ter comunicação e realizar trocas de informação, usando os seguintes serviços. O serviço de *Descoberta* é utilizado para localizar servidores que estão disponíveis na rede e leitura de configurações de segurança. *Canal Seguro*, garante confidencialidade e integridade das mensagens no processo de comunicação. O serviço de *Sessão*, é usado para conectar usuários específicos na camada de aplicação. *Vistas* é a subdivisão do espaço de endereçamento para gerar espaços menores, considerando que o espaço de endereçamento geral é uma vista padrão que abrange todos os nós. *Atributos* faz a leitura e edição de valores nos atributos dos objetos. *Métodos* tem funções internas para geração de eventos e lançamento de funcionalidades. *Monitoramento de Item*, facilita o monitoramento de valores e atributos, gerando notificações quando os valores estão condições específicas. *Registro* é usado pelos clientes para criar e manter registros, de notificações de mensagens. *Gestão de Nodos* pode criar, deletar e modificar nodos no espaço de endereçamento, através de aplicações que tem interação direta com este serviço. *Query* permite ao usuário acessar ao endereço de nodos sem o uso de navegadores, facilitando a filtragem por critérios. Para mais detalhes, verificar as especificações da Fundação OPC-UA.

2.5 Resiliência

A resiliência ajuda a recuperar estados normais do sistema após a ocorrência de incidentes, em vez de evitar a ocorrência deles. A prevenção de incidentes é objeto de estudo em outras áreas de segurança do processo (por exemplo, avaliação de risco). No entanto, é muito difícil prever e evitar todas as ameaças, portanto, a resiliência é necessária como uma medida de segurança adicional. Deve ser especialmente reconhecido como uma característica importante da indústria de processo (DINH et al., 2012). Uma falha é um evento que ocorre quando um sistema se desvia do comportamento pretendido, devido a erros e variações de desempenho causada por trocas de um ou mais componentes do sistema (PRADHAN et al., 2016).

O princípio da flexibilidade para a resiliência conta com processos que possam operar sob vários distúrbios, correspondendo a estados estacionários, enquanto a controlabilidade se refere ao estado dinâmico e a capacidade de atingir pontos-alvo em um determinado tempo. Um desempenho de controle resiliente aceitável é aquele onde as saídas controladas e entradas manipuladas são

mantidas dentro dos limites especificados a partir de seus Setpoint sob quaisquer incertezas (DINH et al., 2012).

A robustez, adaptabilidade e tolerância a falhas, não cuidam os tempos com que o sistema de controle se recupera para operação normal após o incidente indesejável, portanto todas essas propriedades são características da resiliência. A resiliência é um conjunto de todas as propriedades acima no conceito proposto (BASTAN; BENESL; FIEDLER, 2018).

As pesquisas sobre resiliência concentram-se em duas áreas principais, organização e tecnologias da informação. A resiliência organizacional avalia a capacidade de uma organização de sobreviver a ameaças, incluindo a prevenção ou a mitigação de condições perigosas ou prejudiciais que ameaçam a normalidade do sistema (BASTAN; BENESL; FIEDLER, 2018). Cabe destacar aqui, a resiliência é usada como uma política para manter o desempenho de um processo industrial.

2.5.1 Tempos de Resiliência

Os cálculos e identificação de tempos de resiliência em um evento indesejado, são definidos por (WEI; JI, 2010), como métricas para levar em conta o bom desempenho de um sistema de controle. A Figura 2.7 apresenta o comportamento da variável de desempenho de um processo no tempo, onde um incidente no momento (t_i^0) inicia o comportamento a ser estudado.

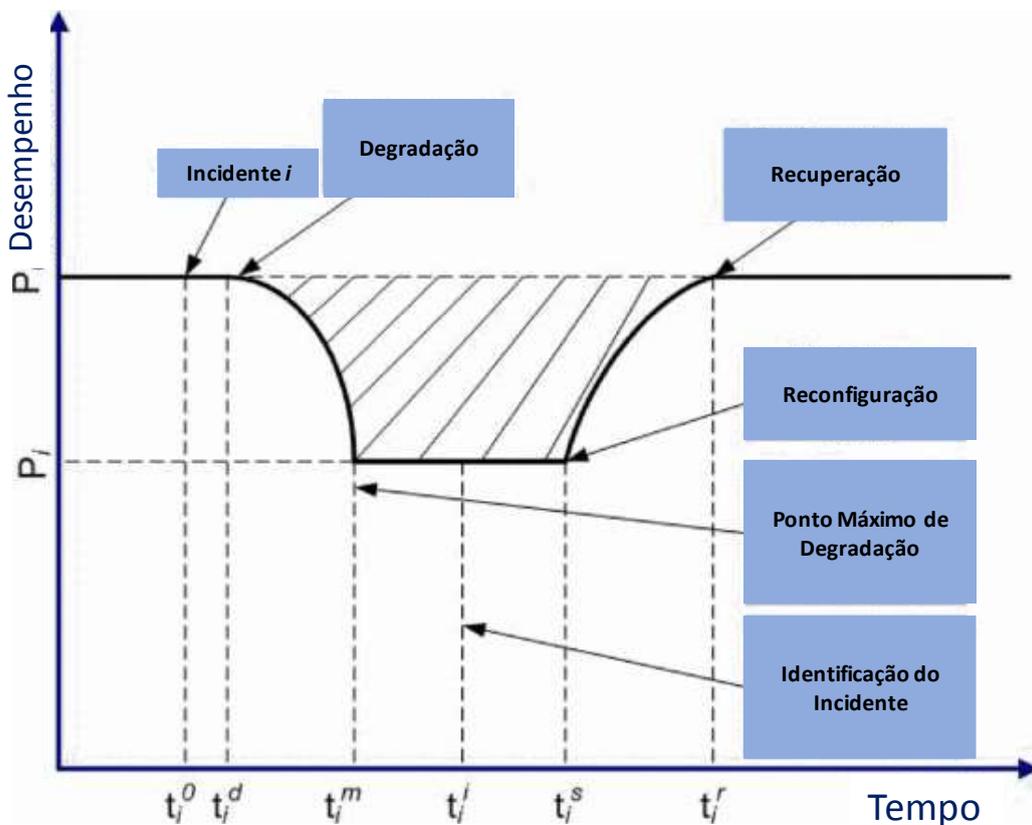


Figura 2.7 Tempos de Resiliência
 Fonte:(WEI; JI, 2010) (Tradução do autor)

O início da degradação, compreende o momento onde a variável sai da normalidade (t_i^d), degradando-se até chegar a pontos máximos (t_i^m). A identificação do evento (t_i^i) pode acontecer antes ou depois de alcançar o ponto máximo de degradação. A recuperação inicia no momento em que algum controle é feito sobre o sistema, assim o momento (t_i^s) pode ser considerado como reconfiguração ou início de recuperação. O retorno da variável a condições normais é o momento de recuperação (t_i^r). Os tempos a considerar para estudar a resiliência como política de reconfiguração são apresentados na Quadro 2.1.

Quadro 2.1 Tempos de Resiliência

Tempo	Cálculo
Proteção	$t_i^d - t_i^0$
Degradação	$t_i^m - t_i^0$
Identificação	$t_i^i - t_i^0$
Recuperação	$t_i^r - t_i^s$

Fonte: (WEI; JI, 2010) (Tradução do autor)

O momento de identificação pode ocorrer antes ou depois de momento de degradação máximo e além disso o conceito de resiliência concentrasse no retorno da variável de desempenho e a sua normalidade. A proposta em termos de resiliência, concentra-se na diminuição de tempos de recuperação baseados em momentos de identificação antes da máxima degradação, incluso antes do evento acontecer. Uma alternativa a esta demanda é o uso de Sistemas de Controle Resiliente.

2.6 Sistema de Controle Resiliente

Um sistema de controle resiliente é um sistema projetado e operado de tal forma que as consequências de eventos indesejáveis podem ser minimizadas e o sistema pode ser restaurado (WEI; JI, 2010). Um conceito mais preciso define que é um sistema que mantém o monitoramento de estados e um nível aceitável de normalidade operacional em resposta a distúrbios, incluindo ameaças de natureza não esperada e maliciosa (BASTAN; BENESL; FIEDLER, 2018).

Levando em conta os RCS e os CPS, uma visão que mistura os dois conceitos, é proposta por (HEHENBERGER et al., 2016), um sistema construído com arquiteturas resilientes que permitem a reconfiguração topológica em qualquer nível, por exemplo que a rede lógica de sensores, será virtualmente definida e separada da rede física. Se houver alguma falha detectada na rede do sensor ou se os propósitos de detecção forem alterados, os sensores serão logicamente reconfigurados por meio do raciocínio na estrutura topológica, ver Figura 2.8. Da mesma forma, falhas de atuadores e controladores podem ser detectados e acomodados através de reconfiguração funcional. Esta perspectiva é proposta também por (WANG; NIU; CHENG, 2015), assegurando que os métodos para a construção de sistemas resilientes envolvem, resposta flexível intercambiando elementos da estrutura topológica, tomada de decisão descentralizada, modularidade, redundância e independência de interações entre componentes.

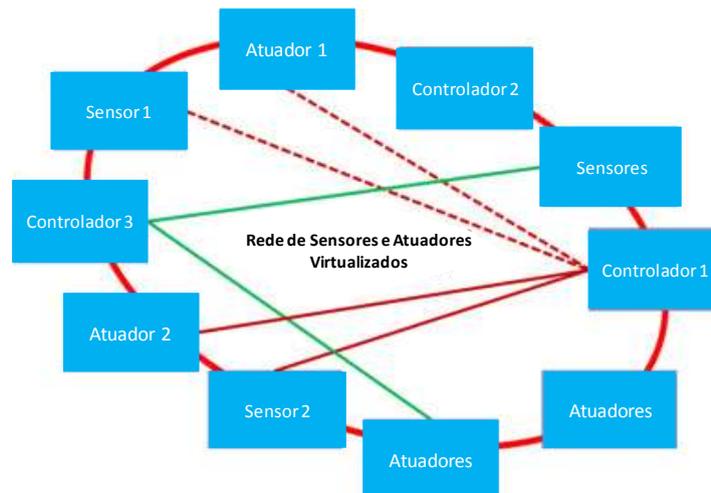


Figura 2.8 Arquitetura Conceitual de CPS de resiliência
 Fonte: (HEHENBERGER et al., 2016) (Tradução do autor)

Os aspectos a considerar para o desenvolvimento de um sistema de controle resiliente são, monitoramento de características e desempenho, resposta a eventos e condições, antecipação de eventos futuros e finalmente a reorganização do sistema (BASTAN; BENESL; FIEDLER, 2018).

A Figura 2.8 ilustra um exemplo de implementação de tal arquitetura CPS resiliente que permite a reconfiguração lógica dos elementos, tais como sensores, atuadores e controladores. Assim, quando as condições externas ou internas mudem, também mudaria a estrutura topológica. Por exemplo, o *Controlador 1* usa o *Sensor 1* para controlar o *Atuador 1*, se supnhamos que o *Atuador 1* esteja com defeito, pode ser adotado o *Atuador 2* sempre e quando tenha uma funcionalidade semelhante.

A reconfiguração dinâmica e a resiliência dos atuadores virtuais são uma chave para alcançar o comportamento resiliente do sistema (HEHENBERGER et al., 2016). A autonomia da resiliência é importante também porque esses sistemas podem ser muito complexos para a reconfiguração manual (PRADHAN et al., 2016).

2.7 Trabalhos relacionados

O objetivo da pesquisa foi compreender o funcionamento dos Sistemas Ciberfísicos e auto adaptativos, analisando o enfoque que eles tem de acordo à Indústria 4.0, especialmente baseados na proposta de arquitetura 5C de (LEE; BAGHERI; KAO, 2015) como solução particular a processos que precisam manter propriedades resilientes. Portanto, foram pesquisados os termos: "*Industry 4.0*", "*CPS*", "*5C Architecture*", "*Self-Adaptive*", "*Resilience*", "*Self-Configuration*", "*MAPE-K*"; "*Autonomic Control*".

A pesquisa está focada na auto-adaptabilidade do CPS considerando que o escopo da pesquisa é baseado na proposta da camada de Configuração de Jay Lee. A camada de configuração é implementada por (PISCHING et al., 2017), abordando uma perspectiva de enviar informações para um usuário, fazendo decisões de manutenção e controle. Embora esta perspectiva seja válida para aprender a lógica das decisões do operador, o ciclo cognitivo do CPS incluindo uma pessoa não satisfaz plenamente o conceito de auto-adaptação. Outras implementações das camadas são apresentadas por (MA; WANG; ZHAO, 2017), baseadas em estratégias Jidoka e (VOGEL-HEUSER; LEE; LEITÃO, 2015) usa políticas de produção. Por outro lado (ALAM; EL SADDIK, 2017) projeta um controlador inteligente na nuvem, como implementação da camada de configuração, usando um controlador de redes bayesianas e Fuzzy. A proposta é baseada em contextos de reconfiguração e de componentes físicos e virtuais para fins de monitoramento. (BRUTON et al., 2016) implementa a camada de configuração usando configurações de camada de cognição, no entanto, é reativa e não leva em conta a resiliência.

As implementações baseadas em arquiteturas Multi-Agente como no caso de (GOUYON; PÉTIN; MOREL, 2007) e (CUPEK et al., 2016), apresentam um sistema de reconfiguração orientada pelo produto. (LEPUSCHITZ et al., 2011) reconfigura componentes físicos usando blocos de função da IEC 61499. As reconfigurações destas implementações são eficazes, mas são feitas no nível do hardware, especificamente na configuração de parâmetros e eles não levam em conta a resiliência.

Satisfazendo o retorno da informação de reconfiguração para a camada física do CPS, (WANG; NIU; CHENG, 2015) apresenta uma composição de serviço sensível ao contexto. (HEAVEN et al., 2009) propõe a seleção de serviços através da execução de planos reativos. (GARCIA VALLS; LOPEZ; VILLAR, 2013) expõe gráficos de reconfiguração e sua devida redução de complexidade para obter um caminho de configuração ideal. Com uma perspectiva de resiliência (WEI; JI, 2010), propõe a antecipação incidentes, tendo em conta a incompatibilidade do modelo topológico, a mudança de parâmetros e falhas nos componentes físicos. (PRADHAN et al., 2018) apresenta uma nova arquitetura que pode ser reconfigurado em tempo de execução (CHARIOT), organizando e gerenciando autonomamente os sistemas IoT. Uma maneira de fazer as

reconfigurações proativamente é apresentado por (ARSHAD; HEIMBIGNER; WOLF, 2007), que estabelece um sistema com configurações, gerenciado por um planejador de tempo. Os trabalhos apresentados apesar de não lidar diretamente com a implementação da camada de configuração satisfazendo os conceitos propostos por Lee, contribuem consideravelmente ao âmbito da auto-adaptação, a interpretação do conceito de resiliência, limitações ao nível da interoperabilidade na indústria e criação de rotas de configuração para CPS.

O Quadro 2.2 apresenta um resumo dos trabalhos relacionados, ressaltando a suas características mais relevantes.

Quadro 2.2 Resumo de Trabalhos Relacionados

Referência	Arquitetura	Causa	Efeito	Resiliencia
(GOUYON; PÉTIN; MOREL, 2007)	MAS	Produto	Componentes Industriais	Não
(CUPEK et al., 2016)	MAS	Contexto	Linhas de Produção	Não
(LEPUSCHITZ et al., 2011)	MAS	Contexto	Serviços de Controle LLC	Não
(VOGEL-HEUSER; LEE; LEITÃO, 2015)	MAS	Contexto	Ajuste de parâmetros	Não
(HOANG et al., 2012)	SOA	Contexto	Serviços de Software	Não
(GARCIA; LIN; MEERKOV, 2012)	Software Embarcado	Desempenho Sensores	Seleção de Sensores	Sim
(KISSKALT et al., 2018)	SOA	Desempenho Sensores	Fusão de Sensores	Não
(MÖNKS et al., 2014)	SOA	Desempenho Sensores	Fusão de Sensores	Não
(WANG; NIU; CHENG, 2015)	SOA	Contexto	Seleção de Serviços	Não
(FLEISCHMANN; KOHL; FRANKE, 2016)	SOA	Desempenho Sensores	Fusão de Informação	Sim
(HEAVEN et al., 2009)	SOA	Contexto	Seleção de Serviços	Não
GARCÍA-VALLS et al., 2014)	SOA	Contexto	Seleção de Serviços	Não
(WEI; JI, 2010)	MAS	Não Auto adaptativo	Não Auto adaptativo	Sim
(PRADHAN et al., 2016)	SOA/IoT	Contexto	Seleção de Serviços	Sim
(PRADHAN et al., 2018)	SOA/IoT	Contexto	Seleção de Serviços	Sim
(VACHTSEVANOS et al., 2018)	Software Embarcado	Sensores de Veículos	Control Veiculo	Sim
(ZHOU et al., 2017b)	MAS	Contexto	Seleção de Serviços	Sim
(RIEGER; MOORE; BALDWIN, 2013)	MAS	Políticas	Serviços e ativos	Sim

2.8 Considerações Finais

A taxonomia de auto adaptação apresentada por (PFEIFFER et al., 2016) , ajuda a sinalizar ao escopo da proposta apresentando os tópicos levados em conta. Cada uma das características amostrasse na Figura 2.9.

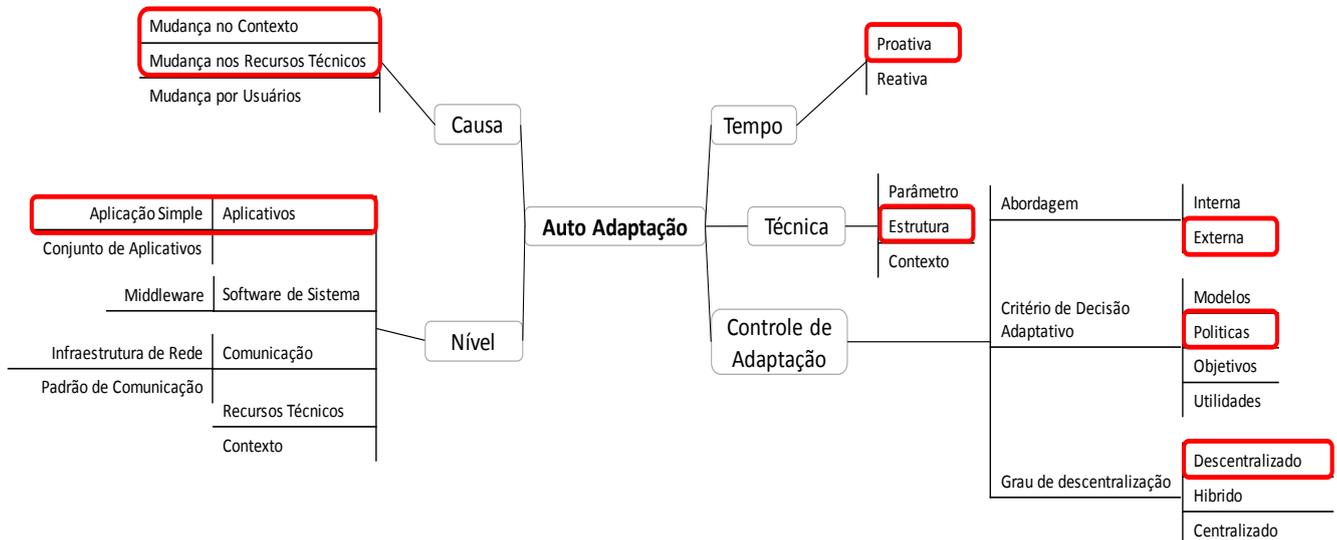


Figura 2.9 Taxonomia da Auto Adaptação
 Fonte: (KRUPITZER et al., 2015)(Tradução do autor)

O CPS recebe informação baseada no monitoramento de *Recursos Técnicos* e *Contexto*, os quais serão consideradas como os *Causas* de adaptações para a proposta. Pretende-se prover de autoadaptação e orquestrar os componentes físicos da planta, portanto define-se o sistema em um *Nível de Aplicação* que garantirá os serviços de reconfiguração ao chão de fábrica.

Com respeito ao *Tempo*, pode-se considerar uma planta em um sistema produtivo, a qual geralmente é controlada por um sistema execução de manufatura (representada no quadro amarelo da Figura 2.10) ou pelas camadas de controle de um CPS, podendo salvar dados de produção e compartilhar estes dados com módulos estatísticos e de simulação (quadros azuis) (PFEIFFER et al., 2016). As simulações apresentadas são de três tipos, garantindo o reconhecimento de falhas no sistema.

As simulações podem ser off-line ou on-line. As simulações off-line (quadro azul “a”), é sustentado por aprendizado estatístico e permite analisar de maneira prospectiva, para posteriormente avaliar as soluções. As simulações on-line mantêm um monitoramento usando um horizonte de simulação dinâmico e são divididas em proativas e reativas. As simulações proativas (quadro azul “b”) apoiam as análises prospectivas para a avaliação de alternativas antes de que os eventos aconteçam e as simulações reativas (quadro azul “c”) simplesmente avaliam as alternativas em tempo de execução (PFEIFFER et al., 2016).

Quadro 2.3 Relação das tarefas MAPE-K e as dimensões da taxonomia de Auto Adaptação

	Tempo	Causa	Nível	Técnica	Controle de Adaptação
Monitoramento	Monitoramento em tempo de execução (Reativo)	Estados das variáveis do sistema real	Topologia	Protocolos de comunicação baseadas em topologias (Conexão Inteligente/Conversão)	Reconfigurações externas, que perseguem um objetivo resiliente de uma maneira descentralizada
Conhecimento	Geração de Simulações para ser usadas antes dos eventos (Proativo)	Disponibilidade de histórico para geração de simulações	Aplicação	Uso de Software de Simulação (Cibernético)	
Análise	Monitoramento de variáveis simuladas (Proativo)	Disponibilidade de horizonte de simulação dependendo da dinâmica do sistema	Aplicação	Monitoramento e avaliação das simulações (Cognição)	
Planejamento	Seleção das melhores alternativas de simulação (Proativo)	Disponibilidade de resultados de simulação	Aplicação	Algoritmos para geração de planos de autoconfiguração (Configuração)	
Execução	Envio de planos de autoconfiguração cada vez que este disponível uma atualização (Proativo)	Disponibilidade de Planos de autoconfiguração	Topologia	Protocolos de comunicação baseadas em topologias (Conexão Inteligente/Conversão)	

Uma vez identificada a taxonomia auto adaptativa do sistema, é preciso definir também estruturas topológicas que descrevam os elementos a fim de conhecer suas características e recursos técnicos, podendo controla-los e detectar os seus eventos e comportamentos. Os elementos físicos a serem considerados são sensores e atuadores, bem como variáveis de controle que descrevam o processo, representando eles em um espaço virtual ou modelo de informação.

O desenvolvimento da proposta requer um protocolo que permita a comunicação independente em cada camada da arquitetura 5C, garantindo interoperabilidade e descentralização. Para estes fins foi selecionada a *Arquitetura Unificada OPC*, analisando as suas características e levando em conta para o desenvolvimento do presente trabalho. A Figura 2.11 mostra a estrutura básica da pirâmide 5C, destacando os dados a considerar em cada camada.

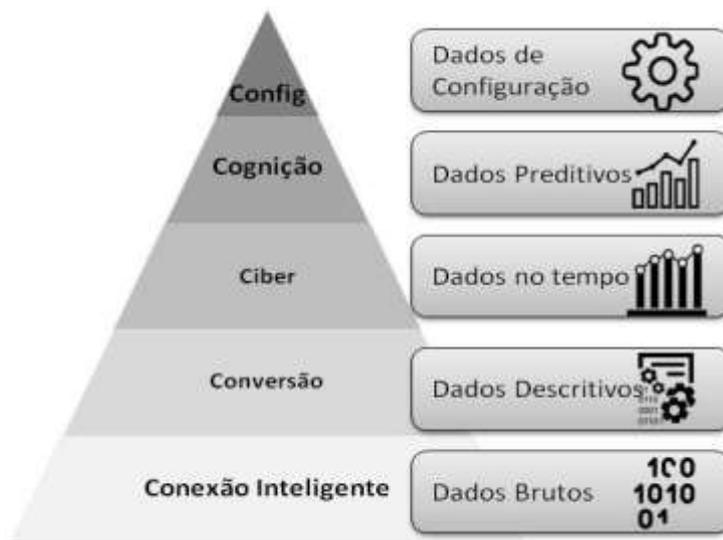


Figura 2.11 Descrição básica dos tipos de dados usados na estrutura 5C
 Fonte: (LEE; BAGHERI; KAO, 2015) Tradução Autor

3 ARQUITETURA PROPOSTA

A arquitetura proposta é orientada a serviços *SOA* (*Service Oriented Architecture*), requerendo aplicativos, que são expostos usando Serviços Web. Além da disponibilidade de dados desde o chão de fábrica, e aplicativos ERP (*Planejamento dos Recursos da Empresa*), MES (*Sistema de Execução de Manufatura*) ou SCADA (*Sistema de Supervisão e Aquisição de Dados*).

A proposta considera os clientes como o ponto de conexão entre camadas, fazendo requisições de serviço ao servidor em uma camada diferente. A Figura 3.1 apresenta uma arquitetura geral onde as camadas da arquitetura 5C são representadas como servidores OPC-UA interagindo com o servidor de *Configuração* que contém um cliente. O servidor de *Configuração* a partir de agora é nomeado como servidor de *Autoconfiguração*.

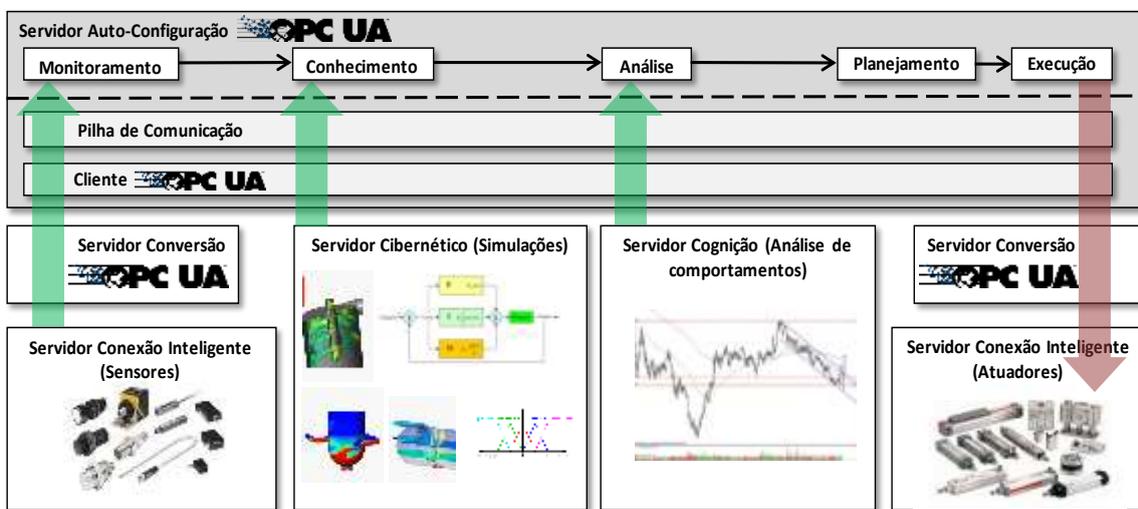


Figura 3.1 Arquitetura geral da proposta

A proposta é baseada em um modelo de auto-adaptação MAPE-K atuando como um *Sistema de Controle Resiliente*, ver Figura 3.1, garantindo interoperabilidade através do protocolo OPC-UA.

O servidor de *Conexão Inteligente* obtém dados brutos, lidos diretamente dos instrumentos físicos e sensores, permitindo-lhes para ser compartilhado com o servidor *Conversão*, para processá-los e convertê-los em informações úteis. A troca de dados brutos na informação depende das abstrações e o processamento de dados que permite reconhecer informação estatística, bem como a capacidade de estruturar o sistema topologicamente. Uma fusão de sensores pode ser um exemplo desta camada, onde dados não processados de diferentes sensores podem ser tratados como um, considerando-se como uma fonte de dados.

A informação representada como uma versão digital do sistema em um servidor *Cibernético*, permitindo reconhecer padrões. O servidor de *Cognição* processa os dados do servidor *Cibernético*, o qual gera previsões dos comportamentos e possíveis estratégias para o seu controle.

O servidor de *Auto-Configuração* cria planos de reconfiguração, compartilhando-os através do protocolo OPC-UA. As informações de configuração retornam ao servidor de *Conexão Inteligente*, permitindo controlar diretamente o processo e os atuadores.

A proposta consiste num ciclo *MAPE-K* mostrado na Figura 3.2, contendo os módulos de: *Monitoramento*, o qual capta os valores do sistema, gera informação relevante para o controle resiliente e salva as informações em históricos para futuras revisões. *Conhecimento*, lê os históricos e gera simulações. A *Análise* gera um buffer de eventos discretos para avaliar os desempenhos nas simulações e identificação de momentos de degradação e recuperação. *Planejamento* cria rotinas de reconfiguração partindo dos eventos e desempenhos e por último o módulo de *Execução* usa a pilha de comunicação OPC-UA para conexão direta com o servidor de *Conexão Inteligente*.

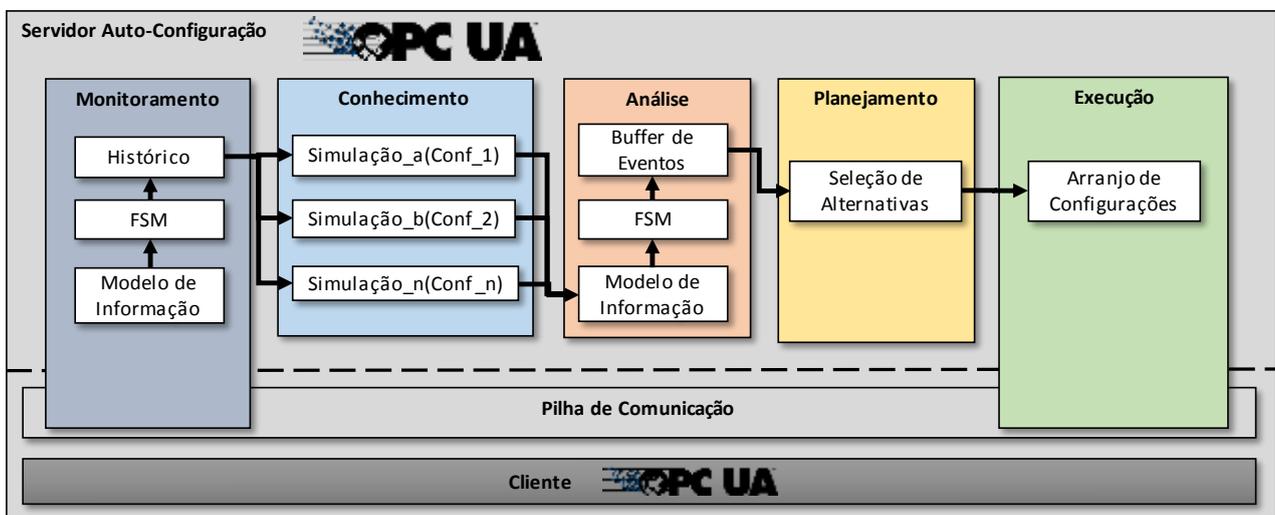


Figura 3.2 Arquitetura Proposta

3.1 Interoperabilidade

A interoperabilidade é garantida pelo uso do protocolo OPC-UA, criando um modelo de informação baseado em circuitos de controle com feedback. Os objetos *Sistema Resiliente* e *Máquina de Estados Finitos Resilientes* foram criados. O modelo de informação *Sistema Resiliente* estende o modelo básico fornecido pelo Protocolo OPC-UA, que contém atributos e propriedades. A *Máquina de Estados Finitos Resilientes*, também estende a base modelo que contém estados e transições. Os estados da máquina definem as condições do servidor juntamente com as variáveis de desempenho. A informação mais completa sobre o mundo está no servidor de conversão, porque apenas as variáveis de resiliência são consideradas no servidor de *Auto-configuração*.

Os modelos de informação são usados no módulo de *Monitoramento e Execução*, do modelo MAPE-K apresentado na Figura 3.2. Estes módulos têm comunicação direta com os servidores de *Conversão e Conexão Inteligente*. O *Monitoramento* captura valores do sistema gerando informações relevantes sobre eventos discretos, gerando históricos. No módulo de *Execução*, as mensagens XML são enviadas para a camada *Conexão Inteligente* cada vez que um ciclo MAPE-K é atualizado. Os tempos dos ciclos do modelo MAPE-K dependem das capacidades de computação e interações dos servidores *Cibernética e Cognição*. As reconfigurações são constantemente atualizadas e lidas proativamente para controlar aos atuadores.

3.1.1 Monitoreamento

O monitoramento é garantido também pelo uso de uma máquina de estados finitos, usando à especificação de "Programa OPC-UA", onde cada variável monitorada é definida dentro de um valor máximo e mínimo, os quais definem um estado sistema. Uma máquina de estados finitos FSM (*Finite State Machine*) atua como um tipo particular de condições no modelo de informação OPC-UA que abstrai um modelo matemático em um número finito de estados (DOROFEEV, 2018).

São definidos dois tipos de objetos, *Sistema Resiliente* para o loop fechado de controle e *Máquina de Estados Finitos Resilientes* para uso como condições de monitoramento.

Sistema Resiliente OPC-UA

Os objetos do sistema resilientes representam as variáveis de um loop de controle fechado. O modelo de informação estende o modelo base proporcionado pelo protocolo OPC-UA, contendo atributos e propriedades mostradas na Figura 3.3

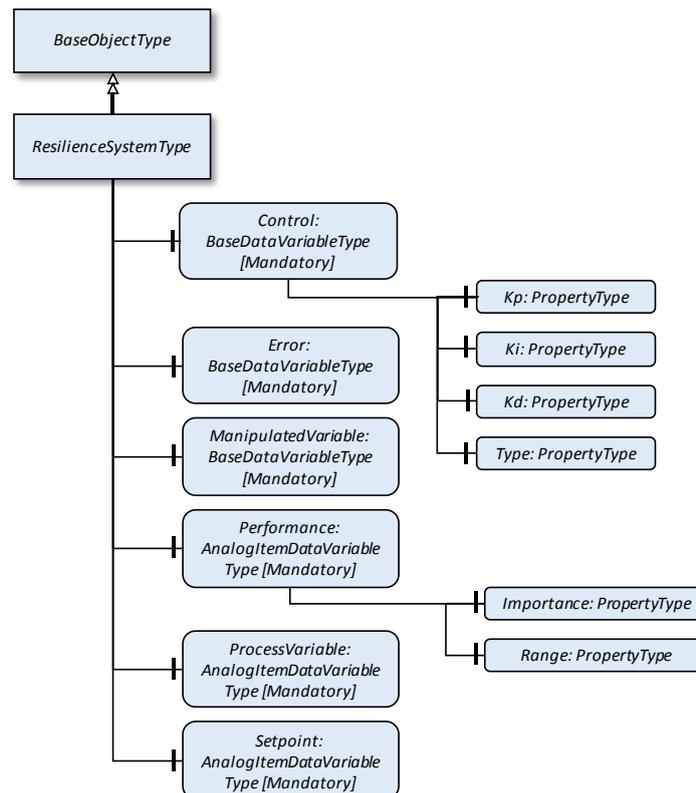


Figura 3.3 Objeto Tipo de Sistema Resiliente

O objeto de *Sistema Resiliente* contém variáveis e referências que descrevem o conteúdo semântico. As variáveis são, controle (*Control*), variável manipulada (*Manipulated Variable*), variável de processo (*Process Variable*), ponto de ajuste (*Setpoint*) e desempenho (*Performance*). O desempenho facilita o cálculo de reconfigurações, contém atributos de normalidade (*Range*) e a importância (*Importance*) para avaliação de alternativas de reconfiguração no ciclo MAPE-K. O Monitoramento da variável de desempenho será o parâmetro de entrada da máquina de estados finitos que atua como condições, disparando transições para mudanças e reconhecimento de estados.

Máquina de Estados Finitos Resilientes

A *Máquina de Estados Finitos Resilientes* tem sua representação por estados e transições. Os estados definem a situação do servidor de *Auto-Configuração* e da cada variável de desempenho pertencente às instâncias de loop de controle e as transições representam as interações entre estados conforme a Figura 3.4.

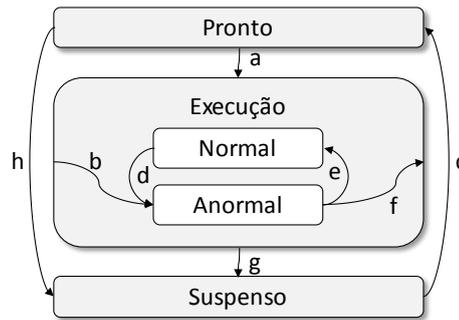


Figura 3.4 Gráfico de Máquina de Estados

O Quadro 3.1 apresenta o modelo de informação para a *Máquina de Estados Finitos de Resiliência*. O estado Pronto é alcançável quando os serviços de *Canal Seguro*, *Sessão*, *Monitoramento de Item* e *Registro* estão disponíveis, caso contrário, algum deles esteja com erro, o estado será *Suspenso*. O estado de *Execução* indica que a variável de desempenho está sendo monitorada e identificando seu estado *Normal* e *Anormal*.

As transições *Pronto-Execução*, *Execução-anormal* e *Suspenso-Pronto* são usadas quando o cliente lança o método Start e estão disponíveis os serviços do estado *Pronto*.

As transições *Normal-Anormal* e *Anormal-Normal*, representam os momentos de degradação e recuperação respectivamente. Por último as transições *Anormal-Execução*, *Normal-Execução* e *Execução-Suspenso*, são geradas quando o cliente faz a chamada do método de *Suspenção*.

Quadro 3.1 Estados e Transições da Máquina de Estados

Nome		Causa
Pronto		<input checked="" type="checkbox"/> Serviço de Canal Seguro OPC-UA <input checked="" type="checkbox"/> Serviço de Sessão OPC-UA <input checked="" type="checkbox"/> Serviço de Monitoramento de Item OPC-UA <input checked="" type="checkbox"/> Serviço de Registro OPC-UA
Execução	Normal	<input checked="" type="checkbox"/> Serviço de Atributos OPC-UA: Leitura de dados de processo e avaliação da normalidade
	Anormal	
Suspenso		<input checked="" type="checkbox"/> Serviço de Canal Seguro OPC-UA <input checked="" type="checkbox"/> Serviço de Sessão OPC-UA <input checked="" type="checkbox"/> Serviço de Monitoramento de Item OPC-UA <input checked="" type="checkbox"/> Serviço de Registro OPC-UA
a	Pronto- Execução	<input checked="" type="checkbox"/> Serviço de Método OPC-UA: Start
b	Execução-Anormal	
c	Suspenso-Pronto	<input checked="" type="checkbox"/> Serviço de Método OPC-UA: Suspenção
d	Normal-Anormal	
e	Anormal-Normal	
f	Anormal- Execução	<input checked="" type="checkbox"/> Serviço de Método OPC-UA: Suspenção
g	Execução -Suspenso	
h	Pronto-Suspenso	

Os estados não pretendem representar completamente a variável e o sistema de controle, embora informações mais completas sobre o mundo possam estar disponíveis nas camadas inferiores da arquitetura 5C. O modelo de informação para a máquina de estados é mostrada na Figura 3.5, apresentando estados, transições, métodos e variáveis padrão do protocolo.

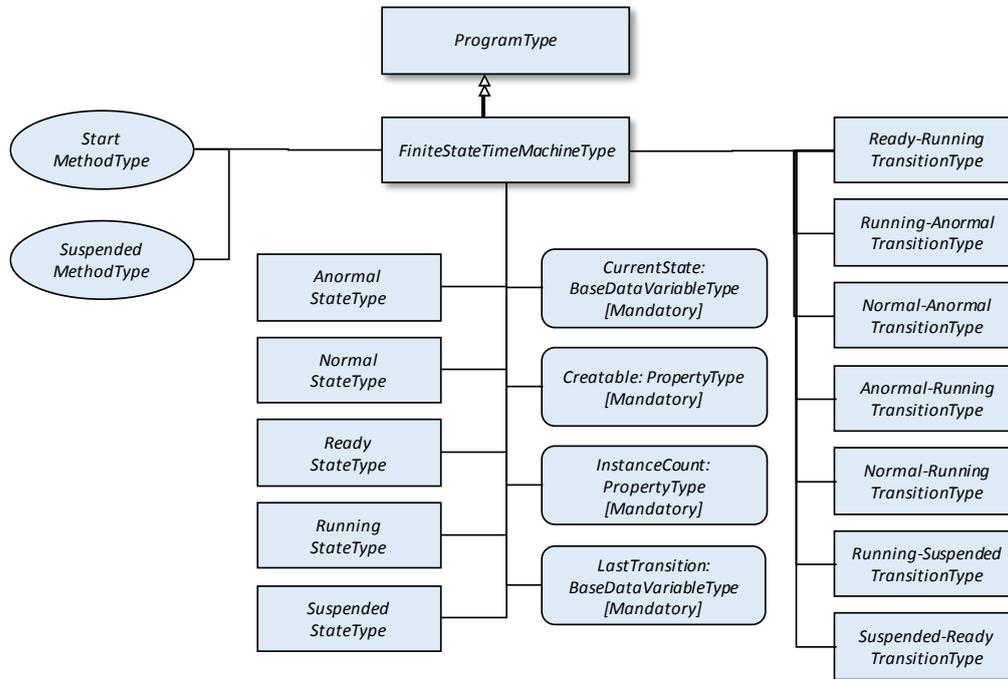


Figura 3.5 Modelo de Informação da máquina de estados

3.1.2 Execução

No módulo de Execução, são enviadas mensagens XML desde o servidor de *Auto-Configuração* até a servidor de Conexão Inteligente, cada vez que é atualizado o ciclo MAPE-K. Os tempos de atualização dependem das capacidades de computação e a dinâmica do sistema que define o horizonte de simulação. As reconfigurações são lidas de uma forma proativa pelo servidor de *Conexão Inteligente*, especificamente pelos atuadores que influem no ciclo de controle. A constante atualização das reconfigurações assegura que a qualidade dos resultados não seja debilitada por configurações posteriores.

3.2 Resiliência

A proposta visa diretamente a redução dos tempos de recuperação, identificando momentos de degradação e recuperação em tempos de simulação. Esse comportamento é garantido nos módulos de *Conhecimento e Análise*, ver Figura 3.2. O módulo de conhecimento gera simulações. O módulo de análise avalia as simulações criando alternativas de reconfiguração, destacando os momentos de degradação e recuperação. Estes dois módulos também usam o modelo de informação OPC-UA proposto com variáveis simuladas.

3.2.1 Conhecimento

O módulo permite a criação de simulações, as quais usam os mesmos objetos de *Sistemas Resilientes* do módulo de monitoramento, gerando diferentes linhas temporais ou versões simuladas do sistema com diferentes configurações. A *Máquina de Estado Finitos Resilientes* permite identificar os eventos e tempos de controle resiliente, além de compartilhar o seu histórico.

3.2.2 Análise

O módulo de *Análise* atua como monitoramento das variáveis simuladas, criando buffers de eventos discretos. O buffer de análise apresenta quanto tempo os sistemas simulados encontram-se em normalidade. A identificação é feita num tempo paralelo de simulação ao tempo de execução. Uma representação gráfica desta dinâmica é apresentada na Figura 3.6, onde primeiras três linhas horizontais representam as variáveis de desempenho em três diferentes simulações com seus estados normais (linhas azuis). Os eventos ressaltados no gráfico, da simulação 2, identificam o momento de degradação (t_i^d) e o momento de recuperação (t_i^r).

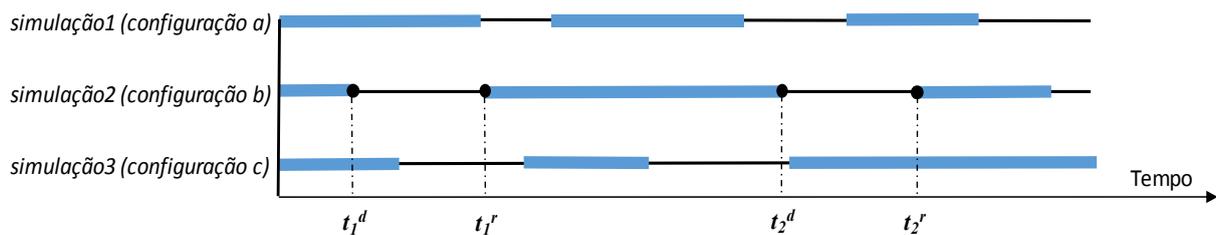


Figura 3.6 Eventos de Degradação e recuperação nas simulações

3.3 Auto Adaptação

O servidor de *Auto-Configuração* recebe informações do servidor de *Conversão e Conexão Inteligente*, especificamente o monitoramento de *Recursos Técnicos* e o *Contexto*. As variáveis representam o estado do mundo, portanto é considerada a *Causa* do sistema adaptações. A *Técnica* de reorganização topológica de elementos no modelo de informação, por exemplo, nós de controle na camada *Conexão inteligente*. O *Nível* é de *Aplicativo*, usando um Modelo MAPE-K com variáveis de resiliência, especificamente um Programa OPCUA, o qual oferece serviços de reconfiguração à *Conexão Inteligente*.

O tempo de reconfiguração é *Proativo* porque as reconfigurações são identificadas em tempos de simulação, pelos módulos de *Conhecimento* e *Análise*, ver Figura 3.2, e portanto, são usados antes do possível evento. O controle da reconfiguração pode ser descentralizado, assumindo controle em sistemas externos. A *Política* geral que orientam as reconfigurações podem ser de naturezas diferentes, levando em conta a *Resiliência*.

3.3.1 Planejamento

A autoconfiguração como característica da auto-adaptação é proposta usando o módulo planejamento, fazendo reconfiguração dos planos, tendo em conta os desempenhos entre cada um dos momentos de recuperação e degradação identificados na Análise, ver Figura 3.2. O módulo de planejamento gera planos de reconfigurações para obter a maior soma possível de desempenho entre cada momento de recuperação e degradação (t_i^r e t_i^d), considerando todas as simulações a priori.

A Figura 3.7 apresenta nodos gerados por cada momento de recuperação ou degradação, assim os nodos t_1^a , t_1^b , t_1^c existem devido a pelo menos um momento de degradação ou recuperação. Os nodos estão interconectados com setas, as quais representam igual quantidade de tempo até o próximo nodo, mas um acumulado de desempenho diferente como em R_1^a , R_1^b , R_1^c . As rotas de reconfiguração garantem a decisão pelo maior desempenho em cada momento de avaliação. As reconfigurações representaram uma mudança topológica proativa que garante a resiliência.

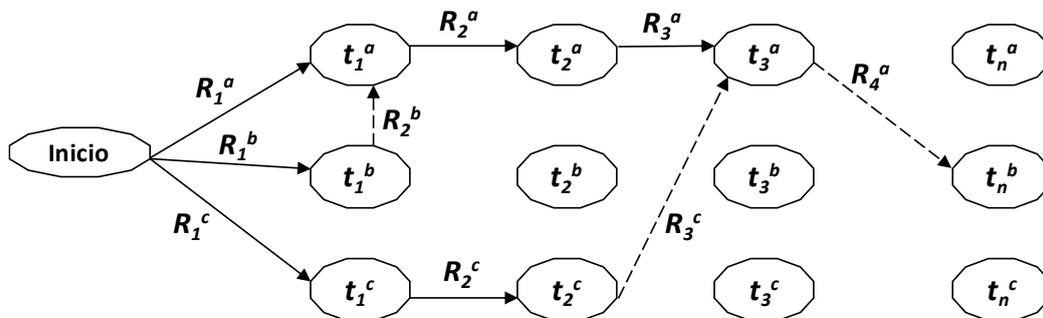


Figura 3.7 Rede de planejamento

4 ESTUDO DE CASO

Este estudo de caso foi desenvolvido nos laboratórios do Centro de Ciências da Computação da Universidade Federal do Rio Grande - FURG. A proposta cria planos de reconfiguração na camada de *Auto-configuração*, os quais são enviados e usados na camada de *Conexão Inteligente* em tempo de execução. O monitoramento e simulações do sistema são a base para a criação de planos de reconfiguração em um circuito de controle fechado, no sistema de controle de nível do tanque da planta SMAR PD-3, ver Figura 4.1.

Uma vez implementada a proposta, a planta representa a camada física do CPS, capaz de operar com diversas malhas de controle, utilizando equipamentos e ferramentas para controle industrial. A planta permite experimentar com as principais variáveis de processo como temperatura, pressão, vazão e nível.

A validação e a prova da proposta são realizadas através de um estudo de caso, na implementação de uma arquitetura 5C, em uma planta de ensino SMAR PD-3. Os resultados obtidos permitem conhecer os efeitos da proatividade nas reconfigurações para a redução dos tempos de recuperação.

4.1 Funcionamento da planta SMAR PD3

O sistema tem um tanque de mistura (3) que recebe duas vazões de água, ambas provenientes inicialmente do tanque reservatório (1). A primeira vazão é bombeada pela bomba (4) para o tanque de aquecimento (2), podendo ser regulada pela válvula (6) e observada pelo rotâmetro (8). No tanque de aquecimento (2), a água é aquecida pelas resistências (10) e segue para o tanque de mistura (3). A segunda vazão é bombeada diretamente para o tanque de mistura (3) pela bomba (5) podendo ser regulada pela válvula (7) e observada pelo rotâmetro (9), similar ao de número (8). Além disso, o tanque de mistura (3) possui capacidade total de 24 litros, não possui controlador de nível e a saída do fluxo ocorre por três válvulas manuais (11) posicionadas em diferentes alturas no tanque.



Figura 4.1 Planta Smar-PD3
 Fonte: (SMAR, 2015)

4.2 Implementação

Foram usados um controle PID e um controle Fuzzy, como nodos de serviços para o controle do sistema, ver (VINICIUS B, 2016) para maiores detalhes. Um evento de troca de Setpoint foi feito de 50% até 70% de altura no tanque, ressaltando que a altura do tanque é de um metro e assumindo que o comportamento é indesejado. A estrutura utilizada foi implementada em 3 servidores, que permitem a comunicação através do protocolo OPC-UA, dois como implementação da camada de *Conexão Inteligente* e um como implementação da camada de *Auto-Configuração*.

O sensor (transmissor de nível LD302) e atuador (posicionador de válvula FY302), foram conectados diretamente com dois Raspberry Pi como interfaces de comunicação. Os Raspberry Pi leram e escreveram sinais 4-20 mA e 0-24 V (MARQUES JUNIOR, 2018). A camada de Auto-Configuração, foi desenvolvida numa workstation Dell Precision T7610 (Linux). Ver Figura 4.2.

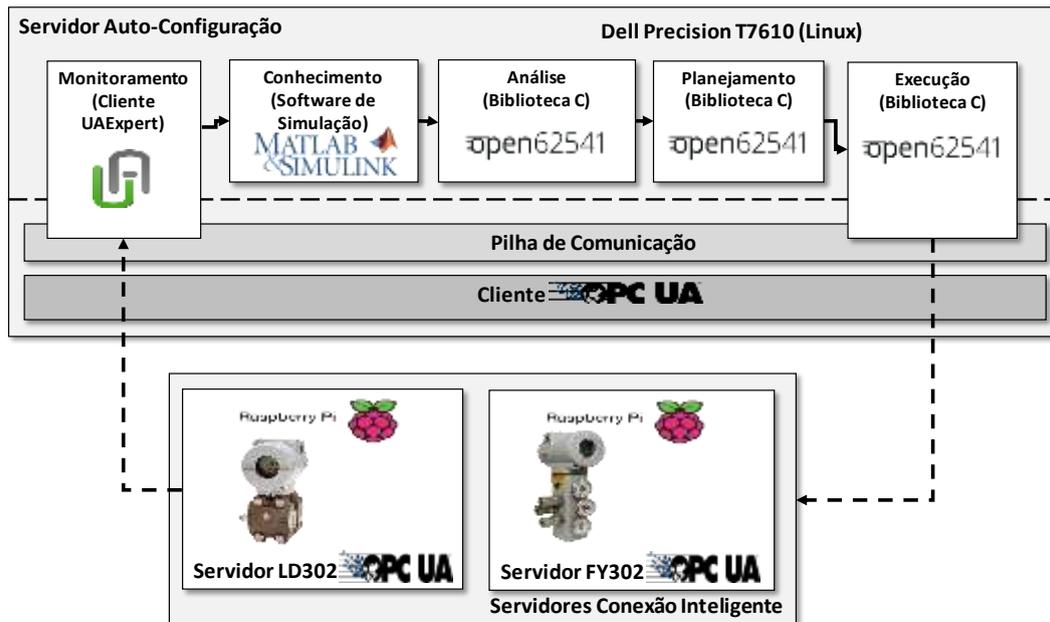


Figura 4.2 Topologia do estudo de caso

As bibliotecas OPC-UA de código aberto usada foi *open62541* levando em conta o levantamento feito por (KONZGEN, 2019), onde assegura que todas as bibliotecas podem ser compiladas nos sistemas *Linux* e *Windows*, mas a biblioteca selecionada tem mais recursos implementados. Para efeitos de compreensão apresenta-se um ciclo MAPE-K com os resultados de cada módulo, considerando que o sistema está em constante atualização.

O módulo de *Monitoramento* foi desenvolvido usando a biblioteca *open62541* e o software modelador *UAModeler*. Os objetos à esquerda da Figura 4.3 são os padrões da fundação OPC-UA e à direita estão as instâncias da proposta, ressaltando os nodos das variáveis da camada de *Conexão Inteligente*. O sensor LD302 está como *variável de processo* e o atuador FY302 como *variável manipulada*. Ao módulo de monitoramento pode-se conectar algum outro cliente dedicado a supervisão de processos ou HMI, como o cliente *UAExpert*.

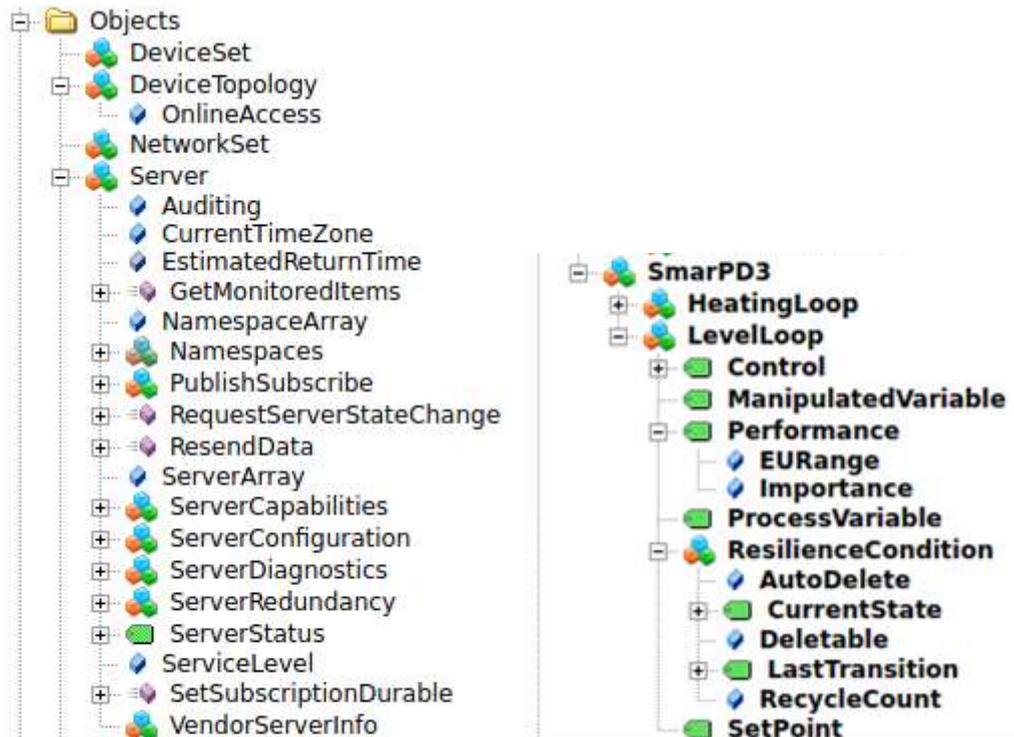


Figura 4.3 Esquerda: Objetos do servidor; Direita: Objetos da SMARPD3

O módulo de *Conhecimento*, usa a modelagem da planta, simulando o seu comportamento quando são usados os controladores PID e Fuzzy no software de simulação de Matlab/Simulink. As Figura 4.4 e Figura 4.5 apresentam os controladores PID e Fuzzy respectivamente, apresentando as respostas da variável de processo para um ciclo de MAPE-K.

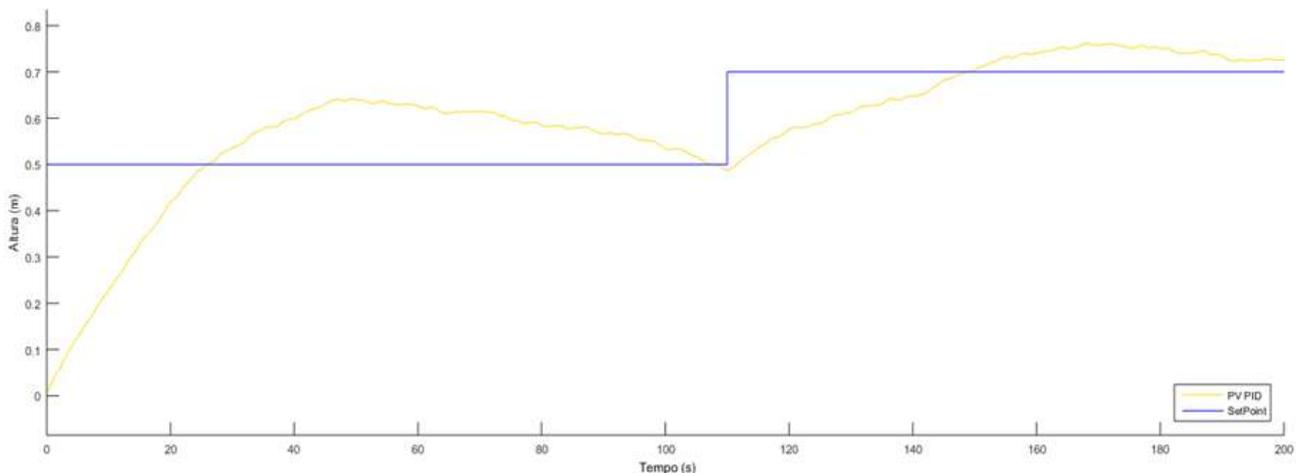


Figura 4.4 Variável de Processo com Controlador PID

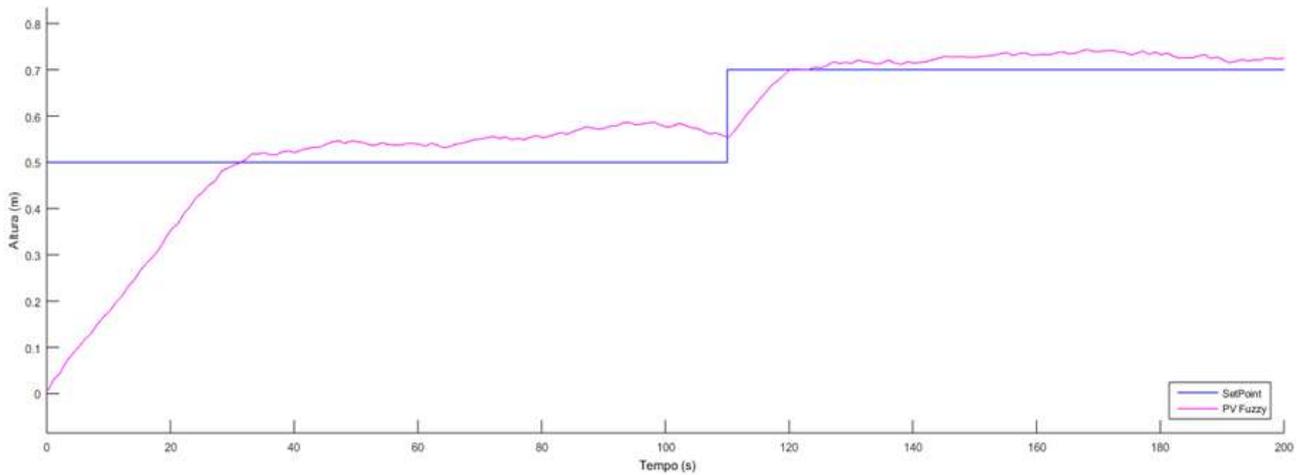


Figura 4.5 Variável de Processo com Controlador Fuzzy

A *Análise*, transforma as saídas e entradas da variável de desempenho a normal, identificando os *momentos de recuperação* (t_i^r) e *degradação* (t_i^d), nesses momentos são feitas as avaliações para a geração dos planos de reconfiguração. Na Figura 4.6, a linha amarela e fúcsia representam os desempenhos dos controladores PID e Fuzzy respectivamente. As linhas pretas tracejadas são os limites da normalidade. Os eventos ressaltados no gráfico identificam os momentos de recuperação de cada uma das simulações depois do evento acontecer. Por um lado, o controlador PID tem um *momento de recuperação* no tempo 24,8 segundos e o controlador Fuzzy 4,9 segundos, considerando uma faixa normal baseada numa tolerância de 0,1 e o evento no tempo 110 segundos.

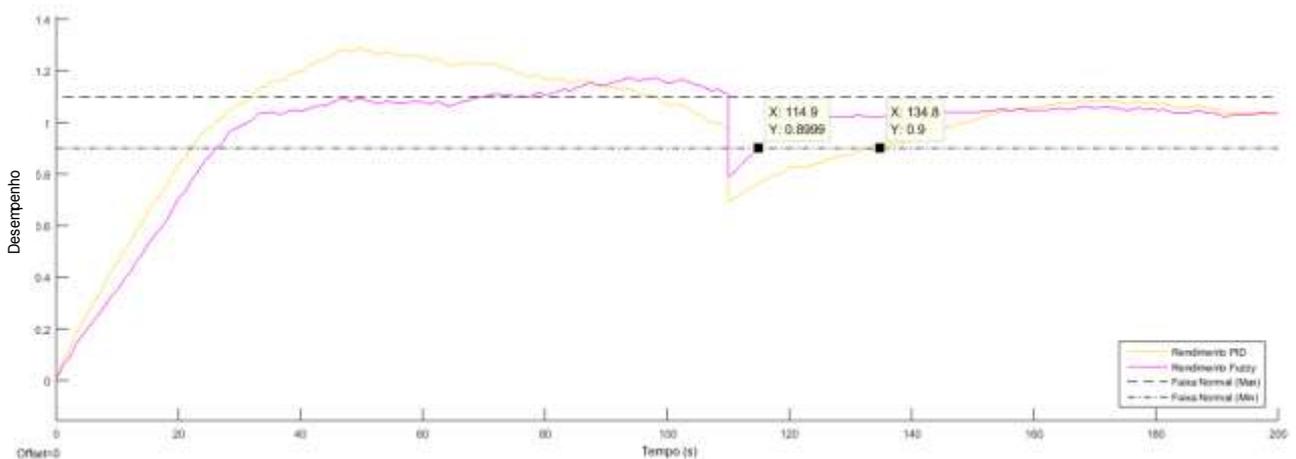


Figura 4.6 Desempenhos dos controladores

O *Planejamento*, tem como resultado uma sequência de *momentos de reconfiguração* (t_i^s) considerado também como trocas de nodos na estrutura topológica do sistema, avaliando os desempenhos entre cada momento de identificação. As trocas são representadas de uma forma meramente referencial na linha preta da Figura 4.7 e os saltos representam cada um dos *momentos de reconfiguração*.

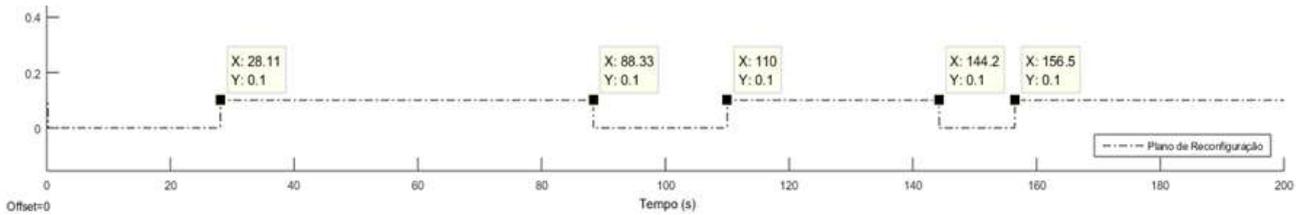


Figura 4.7 Plano de Reconfiguração

O módulo *Execução* é implementado através do envio e atualização dos planos de reconfiguração através da pilha de comunicação de OPC-UA. Os planos são enviados para o servidor do FY302, como serviços de atualização de um arquivo XML que tem as configurações e os momentos de reconfiguração. O desempenho e a variável de processos estão apresentadas na Figura 4.8 e Figura 4.9 respectivamente.

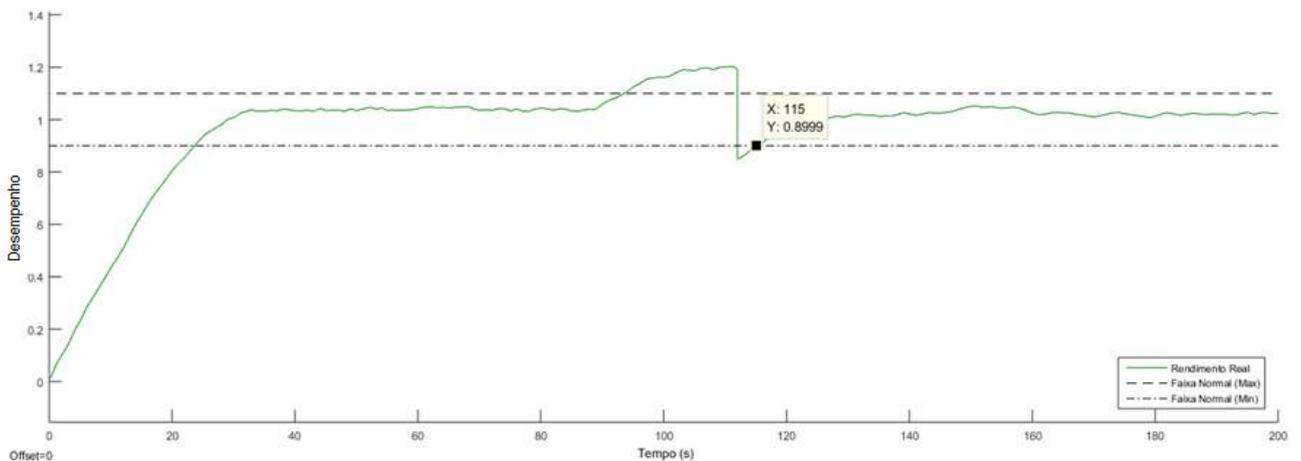


Figura 4.8 Desempenho com Auto Configuração

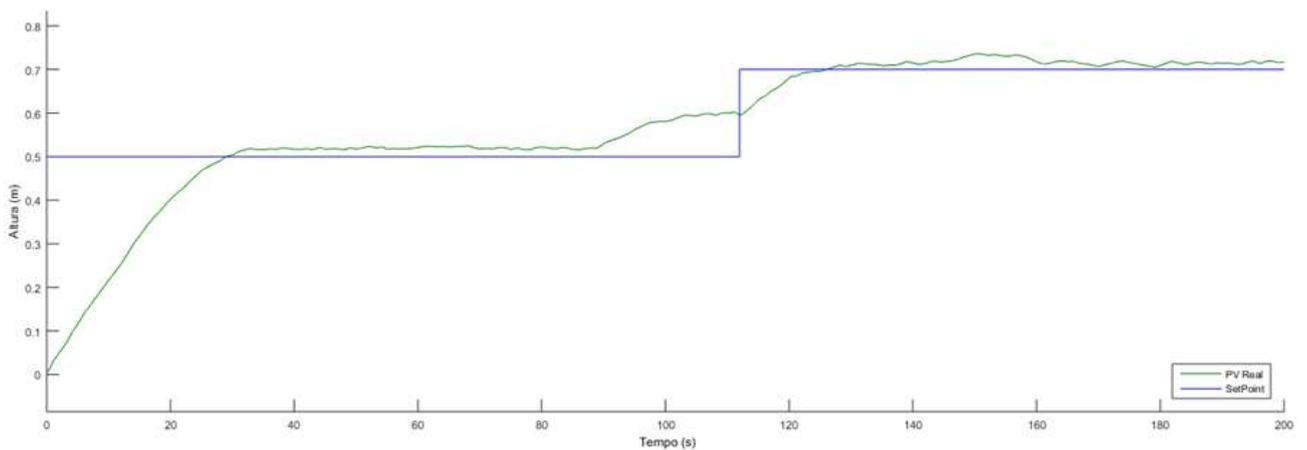


Figura 4.9 Variável de Processo com Auto Configuração

4.3 Validação do caso de uso

Os planos de reconfiguração podem ser enviados proativamente do módulo de *Execução* para o servidor *Conexão Inteligente*, mas os tempos de simulação não estão relacionados aos tempos de execução da planta. A garantia de que as reconfigurações são realizadas de forma proativa é deslizando o horizonte de tempo de simulação em comparação com o horizonte de tempo de execução. Os planos usados com zero segundos de deslocamento são considerados reativos com base em uma simulação proativa.

Os deslocamentos de tempo, entre o tempo de simulação e o tempo de execução foram definidos em -6, -4, -2, 0 e 2 segundos, obtendo diferentes resultados de tempos de recuperação do desempenho do sistema depois que acontece o evento. 100 testes foram feitos em cada intervalo diferente, para obter a recuperação média e analisar a diferença entre eles. Os tempos de recuperação na Figura 4.10 considerando o tempo diferente mudanças e uma tolerância de 10% acima do desempenho de 100%. As linhas cinza, amarela e azul representam os tempos de recuperação o controlador Fuzzy, PID e auto-configuração, respectivamente. O controlador Fuzzy manteve um média de 18,98 segundos, o controlador PID 17,28 segundos e a proposta mostra uma redução de 11,87 s.

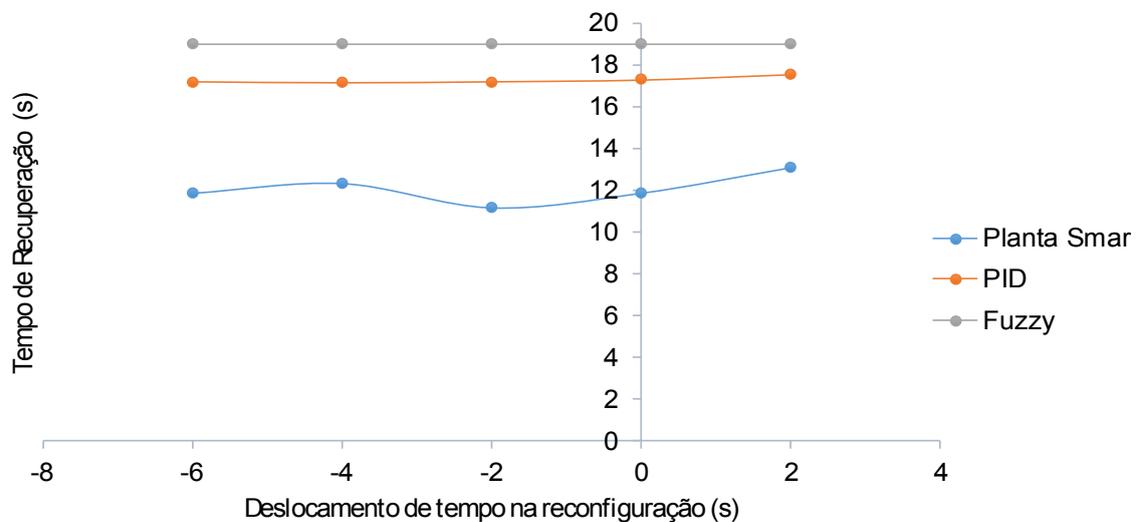


Figura 4.10 Tempos de Recuperação com tolerância de 0,1

Os tempos de recuperação na Figura 4.11 considera uma tolerância de 20% para definir o estado normal no desempenho da planta. As linhas cinza, amarela e azul representam tempos de recuperação usando o controlador Fuzzy, PID e a autoconfiguração, respectivamente. Os melhores tempos de recuperação foram no atraso de -2 segundos (proativo). O controlador Fuzzy manteve uma média de 8,24 segundos, o controlador PID 4,5 segundos e a proposta mostra uma redução de 2,74 segundos.

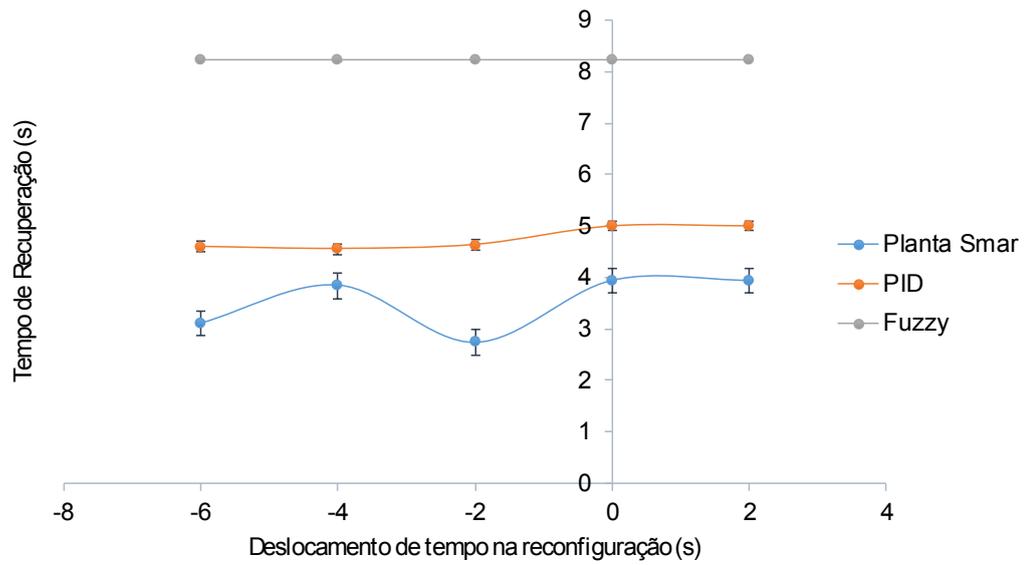


Figura 4.11 Tempos de Recuperação com tolerância de 0,2

A validação da proposta permite a diminuição dos tempos de recuperação, considerando que a reatividade e proatividade influenciam os resultados. O deslocamento de tempo proativo de 2 segundos é o melhor tempo de proatividade para este estudo de caso.

5 CONCLUSÕES E TRABALHOS FUTUROS

A proposta garante interoperabilidade, auto-adaptação e resiliência, usando tecnologias da informação para obter reconfigurações e monitoramento de recursos técnicos a través de topologias.

As reconfigurações proativas mudaram os componentes de topologia, usando o modelo de computação auto adaptável MAPE-K.

O protocolo OPC-UA garantiu o desenvolvimento da configuração camada da arquitetura CPS-5C, mais especificamente usando arquiteturas orientadas a serviços.

A proposta permite a construção de planos proativos de reconfiguração com reduções dos tempos de recuperação após evento indesejado.

O sistema de nível de tanques da estação de treinamento SMAR-PD3 foi controlado através da proposta, reduzindo os tempos de recuperação de eventos.

O plano de reconfiguração para o estudo de caso, mostrou o menor tempo de recuperação, com uma proatividade de 2 segundos.

As reconfigurações proativas não têm os mesmos efeitos em todos os tempos proativos, podendo sobrepor as reconfigurações de eventos anteriores quando for considerado deslocamentos maiores a 2 segundos, isto é demonstrado na Figura 4.10 e Figura 4.11.

Trabalhos futuros visam desenvolver as camadas faltantes da arquitetura 5C, as quais são *Cibernético e Cognição*.

6 REFERÊNCIAS

- ALAM, K. M.; EL SADDIK, A. C2PS: A digital twin architecture reference model for the cloud-based cyber-physical systems. **IEEE Access**, v. 5, p. 2050–2062, 2017.
- ARCAINI, P.; RICCOBENE, E.; SCANDURRA, P. Modeling and Analyzing MAPE-K Feedback Loops for Self-Adaptation. **Proceedings - 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, SEAMS 2015**, p. 13–23, 2015.
- ARSHAD, N.; HEIMBIGNER, D.; WOLF, A. L. Deployment and dynamic reconfiguration planning for distributed software systems. **Software Quality Journal**, v. 15, n. 3, p. 265–281, 2007.
- BASTAN, O.; BENESL, T.; FIEDLER, P. Resiliency, the Path to Safety II. **IFAC-PapersOnLine**, v. 51, n. 6, p. 468–472, 2018.
- BORDEL, B. et al. Cyber-physical systems: Extending pervasive sensing from control theory to the Internet of Things. **Pervasive and Mobile Computing**, v. 40, p. 156–184, 2017.
- BORDEL, B. et al. Process execution in Cyber-Physical Systems using cloud and Cyber-Physical Internet services. **The Journal of Supercomputing**, v. 74, n. 8, p. 4127–4169, 2018.
- BRUTON, K. et al. Enabling Effective Operational Decision Making on a Combined Heat and Power System Using the 5C Architecture. **Procedia CIRP**, v. 55, p. 296–301, 2016.
- CÂNDIDO, G. et al. SOA at device level in the industrial domain: Assessment of OPC UA and DPWS specifications. **IEEE International Conference on Industrial Informatics (INDIN)**, p. 598–603, 2010.
- COELHO, P. M. N. Rumo à Indústria 4.0. **Dissertação de mestrado**, p. 65, 2016.
- CUPEK, R. et al. Agent-based manufacturing execution systems for short-series production scheduling. **Computers in Industry**, v. 82, p. 245–258, 2016.
- DINH, L. T. T. et al. Resilience engineering of industrial processes: Principles and contributing factors. **Journal of Loss Prevention in the Process Industries**, v. 25, n. 2, p. 233–241, 2012.
- DOROFEEV, K. Skill-based Engineering Approach using OPC UA Programs. n. June, 2018.
- DUMITRACHE, I.; CARAMIHAI, S. I.; STANESCU, A. From mass production to intelligent cyber-enterprise. **Proceedings - 19th International Conference on Control Systems and Computer Science, CSCS 2013**, p. 399–404, 2013.
- FERNBACH, A.; GRANZER, W.; KASTNER, W. Interoperability at the management level of building automation systems: A case study for BACnet and OPC UA. **IEEE International Conference on Emerging Technologies and Factory Automation, ETFA**, p. 1–8, 2011.
- FLEISCHMANN, H.; KOHL, J.; FRANKE, J. A Modular Architecture for the Design of Condition Monitoring Processes. **Procedia CIRP**, v. 57, p. 410–415, 2016.
- GARCÍA-VALLS, M. et al. Low complexity reconfiguration for real-time data-intensive service-

- oriented applications. **Future Generation Computer Systems**, v. 37, p. 191–200, 2014.
- GARCIA, H. E.; LIN, W. C.; MEERKOV, S. M. A resilient condition assessment monitoring system. **Proceedings - 2012 5th International Symposium on Resilient Control Systems, ISRCS 2012**, p. 98–105, 2012.
- GARCIA VALLS, M.; LOPEZ, I. R.; VILLAR, L. F. ILAND: An enhanced middleware for real-time reconfiguration of service oriented distributed real-time systems. **IEEE Transactions on Industrial Informatics**, v. 9, n. 1, p. 228–236, 2013.
- GOUYON, D.; PÉTIN, J.-F.; MOREL, G. A product-driven reconfigurable control for shop floor systems. **Studies in Informatics and Control**, v. 16, n. 1, p. ??-??, 2007.
- GRIFFOR, E. R. et al. Framework for cyber-physical systems: volume 1, overview. v. 1, 2017.
- GUSMEROLI, S. et al. OSMOSE : A Paradigm for the Liquid-Sensing Enterprise. [s.d.].
- HANNELIUS, T.; SALMENPERÄ, M.; KUIKKA, S. Roadmap to adopting OPC UA. **IEEE International Conference on Industrial Informatics (INDIN)**, p. 756–761, 2008.
- HEAVEN, W. et al. A case study in goal-driven architectural adaptation. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 5525 LNCS, p. 109–127, 2009.
- HEHENBERGER, P. et al. Design, modelling, simulation and integration of cyber physical systems: Methods and applications. **Computers in Industry**, v. 82, p. 273–289, 2016.
- IBM. IBM: An Architectural Blueprint for Autonomic Computing, 4th Ed. (2006). n. June, 2006.
- KISSKALT, D. et al. A novel approach for data-driven process and condition monitoring systems on the example of mill-turn centers. **Production Engineering**, v. 12, n. 3–4, p. 525–533, 2018.
- KONZGEN. Classificação de Single Board Computer para Embarque de Sistemas Ciberfísicos Baseado na Arquitetura 5C. 2019.
- KOSCHNICK, G.; HANKEL, M.; REXROTH, B. RAMI 4.0-Structure The Reference Architectural Model Industrie 4.0 (RAMI 4.0) Contact: Reference Architectural Model Industrie 4.0. v. 0, n. April, 2015.
- KRUPITZER, C. et al. A survey on engineering approaches for self-adaptive systems. **Pervasive and Mobile Computing**, v. 17, n. PB, p. 184–206, 2015.
- LEE, J.; BAGHERI, B.; KAO, H. A. A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems. **Manufacturing Letters**, v. 3, p. 18–23, 2015.
- LEITÃO, P.; COLOMBO, A. W.; KARNOUSKOS, S. Industrial automation based on cyber-physical systems technologies: Prototype implementations and challenges. **Computers in Industry**, v. 81, p. 11–25, 2016.
- LEPUSCHITZ, W. et al. Toward self-reconfiguration of manufacturing systems using automation

- agents. **IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews**, v. 41, n. 1, p. 52–69, 2011.
- MA, J.; WANG, Q.; ZHAO, Z. SLAE–CPS: Smart lean automation engine enabled by cyber-physical systems technologies. **Sensors (Switzerland)**, v. 17, n. 7, 2017.
- MARQUES JUNIOR, M. R. N. Um dispositivo para aquisição de dados no contexto da Indústria 4.0. **Projeto de Graduação em Engenharia de Automação, Universidade Federal do Rio Grande, Rio Grande, Brasil.**, 2018.
- MENDES, J. M. et al. Process optimization of service-oriented automation devices based on Petri nets. **IEEE International Conference on Industrial Informatics (INDIN)**, p. 274–279, 2010.
- MÖNKES, U. et al. Assisting the Design of Sensor and Information Fusion Systems. **Procedia Technology**, v. 15, p. 35–45, 2014.
- MONOSTORI, L. et al. Cyber-physical systems in manufacturing. **CIRP Annals**, v. 65, n. 2, p. 621–641, 2016.
- OPC FOUNDATION. **OPC-UA Specifications**. Disponível em: <<https://opcfoundation.org>>.
- OZTEMEL, E.; GURSEV, S. Literature review of Industry 4 . 0 and related technologies. **Journal of Intelligent Manufacturing**, n. January, 2018.
- PFEIFFER, A. et al. Manufacturing Lead Time Estimation with the Combination of Simulation and Statistical Learning Methods. **Procedia CIRP**, v. 41, p. 75–80, 2016.
- PISCHING, M. et al. Arquitetura Para Desenvolvimento De Sistemas Ciber-Físicos Aplicados Na Indústria 4.0. **XIII Simpósio Brasileiro de Automação Inteligente Inteligente**, p. 326–331, 2017.
- PRADHAN, S. et al. Achieving resilience in distributed software systems via self-reconfiguration. **Journal of Systems and Software**, v. 122, p. 344–363, 2016.
- PRADHAN, S. et al. CHARIOT: Goal-Driven Orchestration Middleware for Resilient IoT Systems. **ACM Trans. Cyber-Phys. Syst.**, v. 2, n. 3, p. 16:1--16:37, 2018.
- RATASICH, D. et al. A Roadmap Towards Resilient Internet of Things for Cyber-Physical Systems. n. October, 2018.
- RIEGER, C. G.; MOORE, K. L.; BALDWIN, T. L. Resilient control systems: A multi-agent dynamic systems perspective. **IEEE International Conference on Electro Information Technology**, p. 1–16, 2013.
- SCHLEIPEN, M. OPC UA supporting the automated engineering of production monitoring and control systems. **IEEE International Conference on Emerging Technologies and Factory Automation, ETFA**, p. 640–647, 2008.

- SMAR. Plantas didáticas. **Site Oficial Smar**, 2015.
- V. GARCÍA, M.; IRISARRI, E.; PÉREZ, F. Integración Vertical en plantas industriales utilizando OPC UA e IEC-61499. **Enfoque UTE**, v. 8, n. 1, p. 287, 2017.
- VACHTSEVANOS, G. et al. Resilient Design and Operation of Cyber Physical Systems with Emphasis on Unmanned Autonomous Systems. **Journal of Intelligent and Robotic Systems: Theory and Applications**, v. 91, n. 1, p. 59–83, 2018.
- VINICIUS B, O. Comparação Da Utilização Do Controlador Fuzzy E Pid. 2016.
- VIZCARRONDO, J. et al. MAPE-K as a service-oriented architecture. **IEEE Latin America Transactions**, v. 15, n. 6, p. 1163–1175, 2017.
- VOGEL-HEUSER, B.; LEE, J.; LEITÃO, P. Agents enabling cyber-physical production systems. **At-Automatisierungstechnik**, v. 63, n. 10, p. 777–789, 2015.
- WANG, T.; NIU, C.; CHENG, L. A two-phase context-sensitive service composition method with a workflow model in cyber-physical systems. **Proceedings - 17th IEEE International Conference on Computational Science and Engineering, CSE 2014, Jointly with 13th IEEE International Conference on Ubiquitous Computing and Communications, IUCC 2014, 13th International Symposium on Pervasive Systems**, , p. 1475–1482, 2015.
- WEI, D.; JI, K. Resilient Industrial Control System (RICS): p. 1–8, 2010.
- WEYNS, D. Software Engineering of Self-Adaptive Systems: An Organised Tour and Future Challenges. **Handbook of Software Engineering**, p. 1–41, 2017.
- ZHAI, Z. GUO et al. **Comparison of unfractionated heparin and low molecular weight heparin in pulmonary thromboembolism: meta-analysis**. [s.l: s.n.]. v. 26
- ZHOU, P. et al. A decentralized compositional framework for dependable decision process in self-managed cyber physical systems. **Sensors (Switzerland)**, v. 17, n. 11, 2017a.
- ZHOU, P. et al. A Comprehensive Technological Survey on Dependable Self-Managing CPS : The Decade of Researches on Correctness and Dependability. n. July, p. 1–57, 2017b.