

# Architecture for Digital Twin implementation Focusing on Industry 4.0

R. Rolle, V. Martucci, and E. Godoy, *Member, IEEE*

**Abstract**—Industry 4.0 is a new technological wave that dramatically transformed the industrial environment. It consists in taking to the shop floor solutions and concepts from Internet of Things, such as high connectivity, integration and cooperation between distinct elements. The development of smart devices, the continuous evolution of software tools and the implementation of new communication protocols make it possible to implement sophisticated techniques for control, supervision and optimization of industrial processes. Digital Twins are an approach for integrating physical systems to virtual counterparts that replicate in the virtual space the operation of real instruments and machines. Its main objective is to improve the real system performance by using information generated by the virtual counterpart, for example, by executing performance analysis, bottleneck identification, failure prediction, detection of deviations and others. Some of the biggest challenges on this context are creating trustworthy models with acceptable computational costs, assuring real-time communication and developing deep analysis methods. This paper presents the development of a modular architecture for implementing Digital Twins. The architecture is based on an open-source tool for process control, lightweight protocols and flexible tools for modeling and 3D visualization. Tools were chosen in order to allow diverse machines and productive systems to be modeled and represented on the Digital Twin architecture.

**Index Terms**—Digital twins, Industry 4.0, virtualization.

## I. INTRODUÇÃO

A Indústria 4.0 é uma revolução no ambiente industrial cuja proposta é transformar as plantas produtivas tradicionais em fábricas inteligentes [1], de forma a atender as demandas por produção mais rápida, eficiente e flexível. Esta revolução é marcada pela forte integração do chão-de-fábrica às camadas superiores de gerenciamento da manufatura através da Internet, permitindo a implementação de novos modelos de negócios e conceitos de operação que focam nas necessidades individuais dos clientes, em oposição ao modelo de linhas de produção em massa [2]. Para se manter competitivas no mercado, as empresas devem lidar com fatores tais como o curto ciclo de vida dos produtos, flutuações de demanda, expectativas em relação ao tempo de entrega e qualidade [3].

A digitalização da manufatura vem sendo incorporada à prática industrial através de diversas técnicas, tais como os Sistemas Cyber-Físicos, engenharia de sistemas baseada em modelos, uso de *big data* nos sistemas industriais entre outros [4], uma vez que os grandes avanços da Tecnologia

da Informação têm possibilitado o desenvolvimento de ferramentas computacionais mais complexas e poderosas para simulação, emulação e processamento de dados. De forma geral, busca-se integrar os sistemas físicos de produção ao aparato de *software* que serve como suporte para a tomada de decisão, avaliações diversas e implementação de estratégias de otimização.

Neste contexto, os Gêmeos Digitais (GDs) desempenham um importante papel na Indústria 4.0. A ideia de construir um GD consiste na criação de uma réplica virtual (gêmeo) de um dado sistema mecatrônico em ambiente de simulação. O sistema virtual deve refletir o comportamento do sistema real e prover informação útil para diversas finalidades, tais como previsão de falhas, métricas de desempenho e otimização [5].

Embora a pesquisa e aplicação de GDs tenha crescido nos últimos anos, a pesquisa sistemática sobre o tópico é rara e muitas questões ainda estão em aberto. O próprio conceito de GD ainda carece de uma definição unificada, uma vez que se encontram na literatura definições que apresentam os GDs como objetos, mas também outras definições que os consideram como um uma tecnologia ou meio [6]. Outra questão ainda em aberto é a definição de um *framework* completo de aplicação e uma solução de referência [6], [4].

De acordo com estudos recentes, são observados alguns desafios típicos quanto à implementação de GDs. Na parte real, tem-se um grande número de dispositivos sem inteligência, ou seja, que não podem ser integrados por si mesmos. Sendo assim, são necessárias técnicas apropriadas para se extrair dados do chão-de-fábrica de forma inteligente sem que seja necessário substituir todos os equipamentos antigos por dispositivos inteligentes (*smart devices*), o que geraria elevados custos de implementação [1]. Na parte virtual, a modelagem de sistemas e máquinas complexas não é uma tarefa trivial e requer mão-de-obra apropriada [6]. Outro desafio é obter um balanço adequado entre a complexidade do modelo e o seu custo computacional. Quanto à transmissão de dados, manter o sincronismo entre a planta real e sua contraparte virtual é uma tarefa crítica, que requer métodos, interfaces e protocolos apropriados [7].

Encontram-se na literatura diversas propostas de *frameworks* e arquiteturas conceituais de GDs [8], ou seja, sem o desenvolvimento do aparato para a implementação prática. Considerando as arquiteturas que contemplam a implementação prática de GDs, a maioria faz uso de soluções fechadas e proprietárias [9]. Estas soluções não fornecem flexibilidade em relação ao uso e integração de ferramentas para realização das atividades necessárias. Neste contexto, o objetivo deste trabalho é desenvolver uma arquitetura distribuída, modular e

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001 and by grant 2018/19984-4, São Paulo Research Foundation (FAPESP).

R.P. Rolle, V. de O. Martucci and E. P. Godoy are with São Paulo State University (Unesp), Sorocaba, SP 18087-180, Brazil (e-mails: rodrigo.rolle@unesp.br, vinicius.martucci@unesp.br, eduardo.godoy@unesp.br).

flexível para implementação de GDs, baseada na ferramenta de controle de processos *OpenPLC* [10], um *software* de código aberto para controle de processos.

A arquitetura desenvolvida independe de soluções industriais proprietárias e possibilita a criação de GDs de processos automatizados com o uso de *hardware* remoto de Entrada e Saída (E/S), sem a necessidade de inserir novos instrumentos ou *smart devices*. A flexibilidade da arquitetura permite que os módulos possam ser desenvolvidos em ferramentas diferentes (modelagem e visualização) e proporciona escalabilidade, para que novos módulos com diferentes funcionalidades possam ser incorporados à arquitetura. Ao contrário das arquiteturas na literatura, a proposta deste trabalho possibilita que a parte virtual do GD seja composta por aplicações separadas, promovendo melhor gestão do custo computacional das ferramentas de *software*. A disposição modular reduz a interdependência entre os módulos e aumenta a capacidade de generalização, para que variados tipos de sistemas possam ser descritos e representados na arquitetura proposta. Adicionalmente, a comunicação em rede da arquitetura padroniza a troca de informações entre as partes real e virtual do GD.

O artigo está estruturado da seguinte forma. Na seção II é exposta uma revisão bibliográfica sobre GDs, seus conceitos e aplicações no atual contexto da pesquisa acadêmica. A seção III apresenta a descrição da arquitetura proposta, com os detalhes de implementação de cada módulo e da comunicação. Na seção IV é mostrado um cenário experimental de aplicação da arquitetura. A seção V apresenta os resultados do experimento e discussões. A seção VI elenca as conclusões e perspectiva futura deste trabalho.

## II. REVISÃO BIBLIOGRÁFICA

### A. Conceito de Gêmeos Digitais

Os GDs (tradução para a nomenclatura inglesa “*Digital Twins*”) são um conceito apresentado inicialmente por Grieves em 2003 [11], entretanto atualmente existem diversas explicações e definições propostas para o termo [12]. Uma definição típica apresenta o GD como um objeto, que inclui um conjunto de informações, modelos e dados no ambiente virtual. Por outro lado, alguns autores elencam os GDs como tecnologias ou métodos, tal como um método de simulação [6]. De forma geral, pode-se definir um GD como um sistema integrado capaz de simular, monitorar, calcular, regular e controlar os estados e processos de um sistema. O GD é desenvolvido por meio de aquisição de dados, uso de tecnologia de manufatura virtual e é baseado em unidades de controle, comunicação e computação [6]. Uma definição geral mais utilizada até o momento diz que os GDs são “uma simulação probabilística integrada de um produto complexo, utilizando os melhores modelos físicos disponíveis para espelhar a vida de seu gêmeo correspondente” [13]. Uma definição mais estrita apresenta o GD como um conjunto de informação virtual que descreve completamente uma produção física real ou potencial desde o nível micro atômico até o nível macro geométrico, apresentando as características de virtualização, dinamismo, multi-escala, integração, computação e ultra-realismo [6]. Em poucas palavras, pode-se dizer que o GD é uma solução

para o problema de “como simular produtos físicos de forma realística”.

### B. Elementos Básicos dos Gêmeos Digitais

O modelo conceitual de um gêmeo digital é composto de três partes, sendo (1) produtos físicos no Espaço Real, (2) produtos virtuais no Espaço Virtual e (3) conexões de dados e informações que interligam os dois espaços [11]. O Espaço Real é um ambiente complexo, dinâmico e diversificado, no qual interagem pessoas, máquinas e materiais. Seus integrantes apresentam diferentes características de desempenho e comportamento, que podem variar ao longo do tempo por conta do desgaste e envelhecimento. Deve-se preparar o espaço físico para a integração com o GD, ou seja, utilizar dispositivos inteligentes para que seja possível coletar dados de campo para uso na contraparte virtual [6], [14].

O Espaço Virtual, por sua vez, é onde se realizam as operações de replicação do funcionamento do Espaço Real. A representação dos produtos físicos é realizada especialmente por tecnologias de desenho assistido por computador (CAD - acrônimo inglês para *Computer-Aided Design*) e modelagem 3D [6], sendo que ambos são comumente utilizados na etapa de projeto de produtos industriais. A modelagem dos produtos virtuais inclui três aspectos, que são os elementos, comportamentos e regras. Os elementos são características físicas e geométricas dos produtos, os comportamentos se referem tanto ao comportamento do produto em si quanto à interação do produto com o usuário. Quanto às regras, em geral incluem a evolução, otimização e modelos de previsão que são estabelecidos de acordo com a regra de operação do produto físico [14]. Quanto à complexidade e custo computacional da implementação virtual, é possível criar versões leves do modelo virtual, de tal forma que se representem apenas os aspectos mais relevantes do produto, o que reduz a complexidade e aumenta a velocidade de processamento. O objetivo de tal simplificação é possibilitar a simulação de sistemas complexos e sistemas-de-sistemas em tempo real e com custo computacional aceitável, reduzindo também o tempo e o custo da comunicação da informação gerada [11].

O elemento de integração entre os espaços Real e Virtual é a camada de comunicação, que possibilita mapeamento bidirecional e interoperação entre eles [6]. A conexão pode ser estabelecida através de diversas tecnologias, tais como redes de comunicação, computação em nuvem entre outros. A Figura 1 ilustra um esquemático conceitual de GDs.

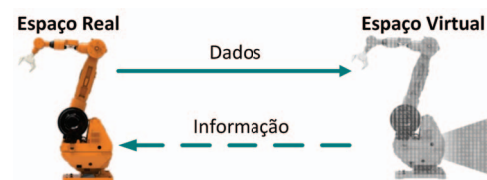


Fig. 1. Esquemático conceitual da implementação de GDs. Elaborado pelos autores com base na imagem disponível em [11].

### C. Contexto e Aplicações

Desde a primeira apresentação, o conceito de GDs esteve relacionado ao contexto do gerenciamento de ciclo de vida de produto, ou PLM (acrônimo inglês para *Product Lifecycle Management*). Tanto a indústria quanto a academia têm se interessado no estudo dos GDs, uma vez que através deles é possível obter um panorama completo dos produtos concebidos, desde a etapa de projeto e desenvolvimento até o fim do ciclo de vida do produto [14].

Em [13] é apresentado um panorama do uso de GDs para o desenvolvimento de veículos para a NASA (acrônimo inglês para *National Aeronautics and Space Administration*) e a Força Aérea dos Estados Unidos da América, considerando os elevados requisitos dos veículos aeroespaciais, a dificuldade de realização de experimentos com equipamento real e destacando as vantagens do uso de simulação de alta fidelidade integrada às plataformas de dados de saúde e desempenho dos equipamentos reais.

Em [15] é proposto um *framework* conceitual de integração de GDs a sistemas de tomada de decisão baseados em ferramentas de gestão da produção. Os GDs não são capazes de dar suporte à gestão de produção por si próprios, por isso são integrados a sistemas existentes de gestão, tais como MES (acrônimo inglês para *Manufacturing Execution Systems*), ERP (acrônimo inglês para *Enterprise Resource Planning*) e outros, que dão suporte a diferentes etapas do processo produtivo. Dada a relevância destes sistemas no contexto industrial atual, os autores propõem que eles sejam integrados num sistema único de suporte à tomada de decisão que não os substitui, mas agrega suas informações de forma a fornecer uma forma mais efetiva de tomada de decisão, abrangendo todo o processo de produção e utilizando também a informação gerada pelos GDs.

Em [14], os autores propõem um *framework* de *design* de produtos baseado em GDs, considerando a relevância do *Design Orientado a Dados* no contexto do desenvolvimento de produtos e enfatizando o fato de que até aqui os GDs têm sido implementados com mais frequência nas etapas de manufatura e pós-produção. São constatadas limitações nos métodos tradicionais de projeto quanto ao uso de grandes volumes de dados oriundos de diversas formas, o que motiva a proposta um *framework* que integra os GDs a métodos de *design* e gestão de ciclo de vida de dados (DLM - acrônimo inglês para *Data Lifecycle Management*), de forma que se possa explorar o potencial dos GDs na etapa de *design*.

Em [12] é discutida a colaboração que os GDs podem trazer para as ferramentas de PLM. Dentro do contexto da gestão de manufatura, são discutidas as limitações dos métodos atuais e as possíveis melhorias que podem ser alcançadas pelo uso de GDs e *big data*. Os autores apresentam aplicações potenciais de GDs nas três fases do ciclo de vida do produto (projeto, manufatura e serviço), sendo que para cada fase é apresentado um estudo de caso.

Em [16] são discutidos possíveis usos dos GDs no contexto da moldagem por injeção de plástico, um dos principais métodos de manufatura de peças plásticas. Os autores discutem o uso de GDs nas etapas de projeto e manufatura, elencando

requisitos como otimização, estabelecimento de projeções de qualidade e manutenção.

Em [6] é apresentado um *framework* de aplicação de GDs ilustrado por um caso de estudo. É desenvolvido um modelo conceitual para as três camadas do GD (física, virtual e de comunicação), bem como é discutida a gestão e utilização dos dados gerados. Ao final o *framework* é implementado numa planta automatizada de soldagem, para a qual os autores desenvolvem a contraparte virtual e implementam as técnicas de aquisição de dados e operação síncrona.

Outro estudo de caso é apresentado em [9], onde os autores implementam um GD numa estação didática de manufatura utilizando uma solução industrial proprietária. Este estudo utiliza-se do GD para verificação do processo em tempo de execução e também para otimização, com uso de algoritmos genéticos e validação de novos parâmetros em ambiente virtual antes da implementação na planta física.

Em [7] é proposta uma arquitetura de monitoramento de processos com a capacidade de manter o sistema virtual em sincronia com a planta física em tempo de execução. São implementados métodos de comissionamento virtual e validação com alguns ajustes, de forma a permitir que o sistema virtual seja adaptado e atualizado automaticamente com base na identificação de mudanças observadas no sistema físico, mudanças estas que podem ser reconfigurações da planta física ou oscilações de desempenho de componentes ao longo do tempo causados por desgaste, oscilações de pressurização, lubrificação, entre outros.

Em [17] é apresentado um *framework* para gestão e controle de produção baseado em GDs para implementação de produção inteligente (*smart production*) de produtos complexos no chão-de-fábrica. O conceito integra soluções orientadas a dados (*big data*) e ferramentas de planejamento e gestão da produção. Um caso de estudo é implementado numa planta de construção de satélites, na qual são inseridas soluções de IoT para aquisição de dados em tempo real. É desenvolvido o modelo virtual do chão-de-fábrica e implementado todo o aparato de gestão e otimização da produção.

## III. ARQUITETURA PROPOSTA

Baseado na análise da literatura, é proposta uma arquitetura para implementação de GDs com foco nas técnicas e ferramentas típicas da Indústria 4.0, apresentada na Figura 2.

### A. Controle do Processo de Manufatura

O controle de processo de manufatura é realizado pelo *OpenPLC* (versão 3.0) [10], que é uma plataforma de código aberto que incorpora as funcionalidades de um CLP e permite o uso de diversos tipos de *hardware* como interfaces E/S em processos de automação. Através do editor *OpenPLC Editor*, é possível desenvolver lógicas de programação nas cinco linguagens definidas pela norma IEC 61131-3. Um aplicativo servidor é executado no Windows, Linux ou Raspberry Pi e realiza a comunicação com o *hardware* de interface. Através do uso de *software* auxiliar, é possível comunicar-se com o servidor através do protocolo de comunicação UDP (acrônimo inglês para *User Datagram Protocol*), que é um padrão leve



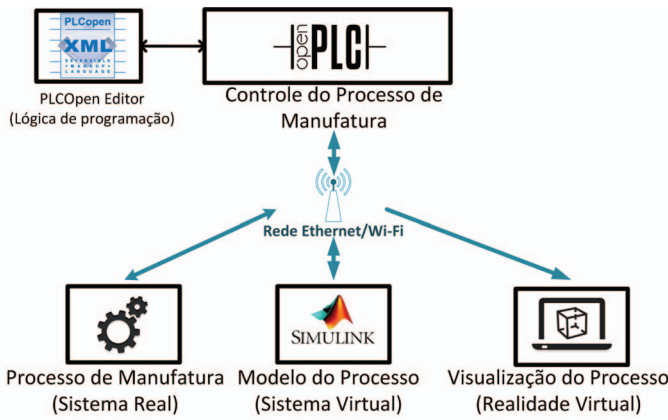


Fig. 2. Arquitetura modular para implementação de GDs proposta.

de comunicação comumente utilizado em aplicações de tempo real.

O *OpenPLC* é responsável por realizar o controle do sistema de manufatura de acordo com a lógica de programação. É necessário que os sinais de sensores e atuadores reais sejam refletidos para o espaço de E/S virtual para que então sejam transmitidos aos módulos de simulação. Esta operação é realizada através de uma lógica simples de replicação expressa no diagrama *Ladder*.

Para evitar confusão entre os endereços de E/S utilizados na aplicação real e nas aplicações virtuais, divide-se o espaço de endereçamento do *OpenPLC* em duas partes, sendo uma destinada à parte real e outra à virtual. A Figura 3 ilustra esta divisão.

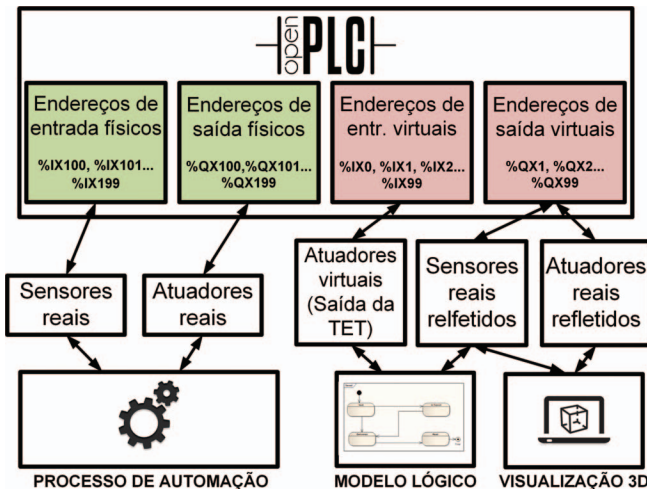


Fig. 3. Divisão dos espaços de endereçamento de E/S no *OpenPLC*.

De acordo com o esquemático, tem-se que:

- O processo de automação real é regido pela lógica *Ladder* definida no CLP;
- O modelo lógico recebe os sinais dos sensores reais refletidos no espaço de E/S virtual e retorna para o CLP os estados dos atuadores virtuais nele computados;
- O aplicativo de visualização 3D recebe os sinais dos sensores e atuadores reais refletidos no espaço de E/S

virtual para gerar a animação 3D em tempo de execução.

### B. Processo de Manufatura

O espaço físico é um ambiente de produção normalmente complexo, diverso e dinâmico, no qual estão inseridos por diversos tipos de equipamentos. As máquinas e materiais são arranjados no espaço de trabalho de forma a permitir o sequenciamento e execução das tarefas do processo produtivo. Todos estes elementos finais (sensores e atuadores) devem ser conectados ao sistema de controle de processos, de forma a permitir a coleta de dados para processamento no CLP e nas demais ferramentas de virtualização. O grupo de equipamentos deve ser conectado ao *OpenPLC* através de *hardware* de E/S apropriado, de tal forma que o servidor possa reger a operação da planta. Embora o chão-de-fábrica normalmente seja composto por processos discretos e contínuos, este trabalho tem foco em aplicações que envolvem variáveis discretas.

### C. Modelo do Processo

Para viabilizar a criação do GD do sistema de automação, é necessário replicá-lo em ambiente virtual. Neste contexto, são utilizadas aplicações de simulação com o objetivo de criar um modelo lógico que descreva o comportamento do sistema, de tal forma que um *software* possa imitar em ambiente virtual as características de operação de um processo real. Com base num modelo fidedigno, é possível ainda realizar operações de avaliação, estudo e otimização sobre o sistema por meio de análise e reconfiguração. Resumidamente, a modelagem de sistemas de automação pode ser realizada através de três etapas: (1) obter um conjunto de variáveis mensuráveis de entrada, (2) obter um conjunto de variáveis de saída que variem de acordo com as variáveis de entrada e (3) definir funções matemáticas que relacionem as entradas e saídas mensuradas, de forma a obter um modelo do funcionamento do sistema [18]. A modelagem de Sistemas a Eventos Discretos (SEDs) pode ser realizada através de abordagens específicas, tais como máquinas de estados, autômatos finitos, redes de Petri e outros.

A ferramenta de modelagem utilizada neste trabalho é a Tabela de Estados e Transições (TET - tradução livre do inglês *State-Transition Table*), que é parte da biblioteca de blocos do aplicativo de simulação *Simulink*, integrante do *software* de engenharia *MATLAB* (versão R2016b). Este recurso permite a descrição do comportamento lógico de um sistema através de relações condicionais (*if-else*), permitindo que se definam os estados das variáveis de saída, as condições de transição e a implementação de temporização. Este bloco possui compatibilidade com as interfaces de comunicação UDP disponíveis no *Simulink*, que por sua vez permitem a troca de informação com o servidor do *OpenPLC*. Desta forma, pode-se efetuar a alimentação da TET continuamente com dados do CLP.

O primeiro passo para criar uma TET é observar o sistema de automação, analisando quais são seus estados e quais fenômenos provocam a mudança de estado. Em geral os estados dizem respeito às condições dos atuadores, enquanto as transições são mais frequentemente associadas à leitura de elementos sensores.

Após a identificação dos estados e transições, deve-se realizar a medição dos tempos de transição quando o sistema possui transições temporizadas. A ferramenta TET possui um comando próprio para inserir atrasos de temporização (*delay*) na evolução do modelo lógico.

Por fim é necessário transcrever os estados e condições de transição na tabela da ferramenta. Nesta tabela, cada estado recebe um nome, é definida a condição (ligado ou desligado) de cada atuador enquanto o estado estiver ativo, é definida a condição de transição que fará o sistema sair deste estado e finalmente é definido o estado subsequente, uma vez que ocorra a condição de transição.

Para fins de demonstração, foi criado um processo-exemplo composto por uma entrada (sensor) e uma saída (atuador). Uma vez que se aplica um sinal de entrada, a saída é habilitada por cinco segundos e então é novamente desligada, aguardando um novo sinal de entrada. O diagrama resultante da montagem da TET é apresentado na Figura 4.

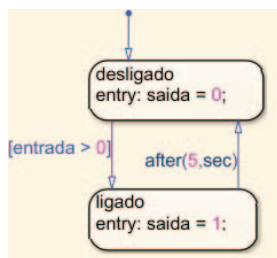


Fig. 4. TET do processo-exemplo com duas variáveis e transição temporizada.

#### D. Visualização do Processo

Uma vez que o modelo em TET é voltado para o comportamento lógico do processo, a replicação do processo físico em ambiente virtual é complementado por um módulo de realidade virtual, que permite a visualização do processo em interface 3D. Este *software* é apto a trocar dados com o CLP em tempo de execução e assim mostrar os estados da planta com base nos dados de E/S obtidos dos sensores e atuadores do processo de manufatura. O programa executa modelos CAD 3D animados utilizando dados de CLPs, seguindo a ideia proposta por [19], que utiliza a ferramenta *Visual Studio* em conjunto com o motor de jogos *Microsoft XNA*. Esta ferramenta tem grande capacidade de generalização, uma vez que é possível desenvolver tais modelos CAD para diversos tipos de máquinas e processos de produção.

O aplicativo desenvolvido para esta arquitetura é integrado ao *OpenPLC* através de mensagens UDP. Foi implementado um cliente que recebe e trata as mensagens provenientes do servidor, de forma que o ambiente de realidade virtual seja atualizado periodicamente com os dados do CLP e faça a replicação da operação da planta real. De acordo com os dados dos sensores reais, são gerados os objetos de processo com os atributos específicos. A movimentação é gerenciada pelo modelo CAD com base nas interações dos objetos, sendo que os atuadores são acionados ou desligados de acordo com os sinais provenientes do CLP.

#### E. Integração e Comunicação

Uma vez que a ideia de Indústria 4.0 pode ser vista como um desdobramento da Internet das Coisas, os padrões de comunicação na Indústria 4.0 seguem uma linha similar, exceto quanto a algumas necessidades específicas das aplicações industriais. Neste trabalho o padrão de comunicação implementado foi o UDP sobre redes Ethernet (sem fio ou cabeadas). Este protocolo permite a comunicação de pequenos pacotes de dados entre computadores baseada no endereçamento IP. Por outro lado, o UDP não possui garantia de entrega nem proteção contra duplicação dos dados. A única exceção é a comunicação do *OpenPLC* com o *hardware* de E/S, que é realizada através do protocolo Modbus TCP.

O *OpenPLC* não possui módulo de comunicação UDP integrado, por isso é utilizado o recurso auxiliar de *software* chamado *SimLink*, uma ferramenta que acessa o *buffer* de E/S, lê os endereços de saída do espaço virtual e comunica cada *bit* periodicamente como um pacote UDP através de uma porta específica definida pelo usuário. Este *software* também é capaz de fazer a escrita de dados de entrada no espaço virtual à medida que são enviados pacotes UDP dos aplicativos de simulação para o CLP através das portas estabelecidas. O intervalo de envio e recebimento de pacotes UDP pode ser definido em milissegundos através do código-fonte.

## IV. CENÁRIO EXPERIMENTAL

A arquitetura proposta foi implementada num laboratório de automação a fim de ter sua funcionalidade verificada. Os módulos funcionais foram implementados e integrados. Esta seção apresenta os detalhes de implementação do GD.

#### A. Processo de Manufatura

O processo de manufatura utilizado no arranjo experimental é a estação didática de automação *Festo MPS Sorting*. Esta estação recebe peças de três cores diferentes e realiza a separação das peças por cor, através da leitura de sensores. Ela possui quatro sensores de presença e quatro atuadores, sendo uma esteira de deslocamento, dois desviadores e um bloqueador de passagem associado à esteira. O controle foi realizado através do *OpenPLC*. Para que o controle seja executado, é necessário um microcontrolador compatível com o *OpenPLC*, que funciona como interface remota de E/S. Este microcontrolador foi integrado ao processo industrial através de uma placa de condicionamento de sinais. Em seguida foi desenvolvido o diagrama *Ladder* para controle do processo. A Figura 5 apresenta a estação de trabalho.

#### B. Modelo do Processo

Uma vez definido o processo de automação real, a etapa seguinte é a criação de sua contraparte virtual. A partir do mapeamento dos estados e variáveis de transição do sistema, é possível gerar a TET que corresponde ao comportamento real da estação. Os estados identificados foram:

- Espera: a esteira permanece ligada para que uma nova peça possa ser processada e os desviadores (atuadores que empurram as peças para as bandejas de saída) são



Fig. 5. Estação de trabalho *Sorting*.

desligados. Quando o sensor de presença é excitado ocorre a transição para o próximo estado;

- Atraso de leitura: a trava permanece habilitada por dois segundos para que a peça seja acomodada junto aos sensores, evitando erros de leitura e a passagem de mais de uma peça por vez. Após um segundo ocorre a transição para o próximo estado;
- Leitura: são recebidos os sinais dos sensores para identificar o material da peça. Conforme a leitura dos sensores é definido o próximo estado;
- Estágio 1: a trava é liberada para que a peça siga para as bandejas e o atuador correspondente (desviador) é acionado. Após 20 milissegundos ocorre a transição para o próximo estado ;
- Estágio 2: a trava é fechada novamente e o desviador correspondente à cor da peça permanece habilitado. A planta volta para o estado “Espera” após o tempo típico de processamento de cada peça (prata: 1 segundo; preta: 1,5 segundos; vermelha: 2 segundos).

Os estados identificados são mostrados esquematicamente na Figura 6.

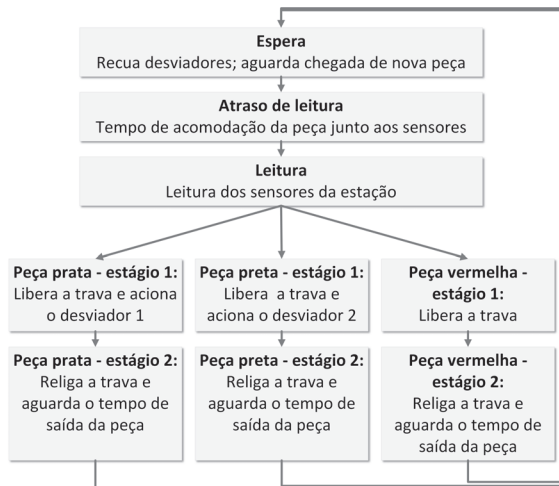


Fig. 6. Estados identificados na estação *Sorting*.

De posse desses dados, foi possível elaborar a TET no aplicativo *Simulink*. Inicialmente o bloco foi testado e validado *off-line*, para que em seguida fossem implementados os blocos de comunicação UDP. O bloco recebe os dados dos quatro sensores reais e gera os estados dos atuadores virtuais (sim-

ulados), de forma que devem ser gerados os mesmos estados obtidos na planta real.

O *Simulink* não executa suas rotinas de simulação na base de tempo real (*wall clock*), mas numa base de tempo acelerada, de acordo com a complexidade dos modelos em execução e a capacidade de processamento do computador utilizado. Por isso foi necessário utilizar um bloco adicional à simulação, para que se obtivesse uma base de tempo próxima ao *wall clock*, como mostrado na Figura 7. Embora não seja possível sincronizar as bases de tempo do *OpenPLC* e do aplicativo de simulação, com o uso do bloco foi obtida aproximação suficiente para o contexto deste trabalho, de forma que não foram notadas distorções quanto à temporização do sistema físico em relação à TET.

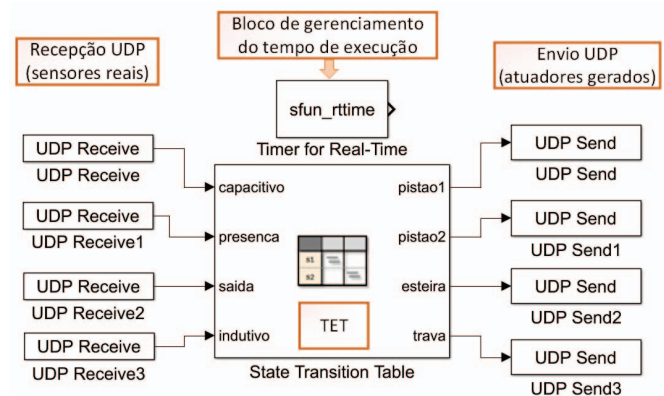


Fig. 7. Aplicação completa da TET e blocos adjuntos no *Simulink*.

Com base no preenchimento da tabela, o aplicativo gera ainda um diagrama esquemático da máquina de estados resultante. Os estados dos atuadores simulados são enviados para o *OpenPLC* para que se possam implementar lógicas de comparação entre os processos real e virtual. A Figura 8 mostra a TET resultante.

### C. Visualização do Processo

Foi elaborado um modelo CAD 3D animado da estação didática, bem como das peças que são processadas. Este modelo apresenta na tela os estados dos sensores e atuadores do sistema físico. Os dados são obtidos por meio da comunicação UDP estabelecida entre o CLP e o aplicativo de realidade virtual. A cada vez que uma nova peça é identificada pelos sensores da planta de automação física, é gerada uma peça no ambiente virtual, inicialmente não colorida. Uma vez que o CLP recebe os dados dos sensores responsáveis pela leitura da cor, estas informações são encaminhadas para o ambiente virtual e a peça recebe coloração de acordo com os dados obtidos. A Figura 9 mostra o aplicativo de visualização 3D em execução.

Outra solução que pode ser utilizada para a visualização 3D é o aplicativo *Factory I/O* [20] (versão 2.3.6), uma ferramenta que permite a rápida construção de processos de fabricação virtuais utilizando peças industriais comuns, tais como esteiras, sensores, elevadores, manipuladores e outros. O acionamento dos equipamentos de campo é realizado por meio



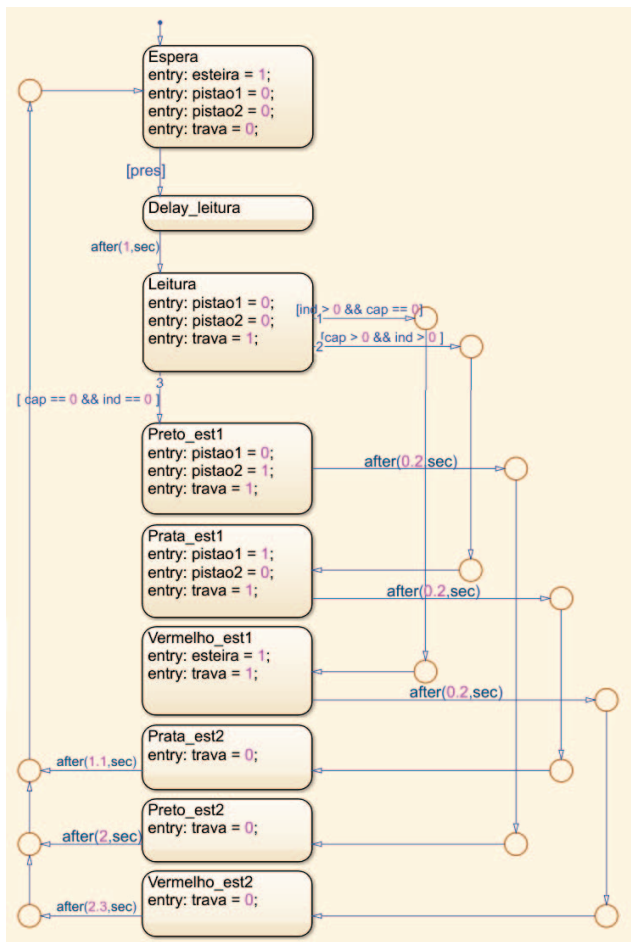


Fig. 8. Esquemático da TET da planta de automação.

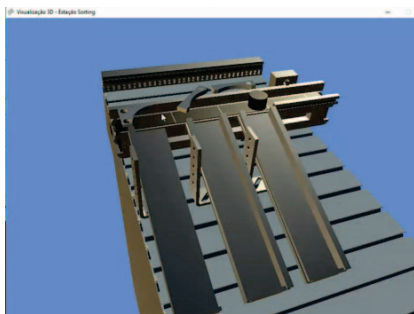


Fig. 9. Ambiente de realidade virtual integrado ao OpenPLC para exibição do status da planta de automação.

de sinais externos, que podem ser obtidos por meio de placas de aquisição de dados, comunicação com CLPs compatíveis ou por meio de protocolos de comunicação. No contexto desta arquitetura, é utilizado o protocolo Modbus TCP/IP. A Figura 10 mostra a visualização da estação *Sorting* no ambiente do *Factory I/O*.

**D. Temporização do GD**

Com todos os módulos implementados, foram realizados testes de velocidade de comunicação de forma a verificar se o desempenho é suficiente para a manutenção do sincronismo

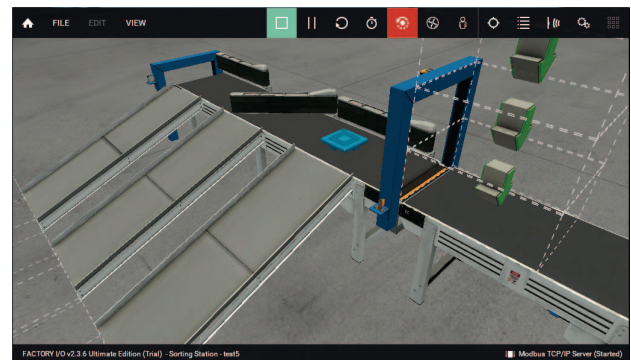


Fig. 10. Aplicativo *Factory I/O* integrado ao *OpenPLC* para exibição do status da planta de automação.

do GD. Para isso, foram realizadas medições dos tempos de comunicação entre os módulos do GD através do aplicativo de análise de tráfego em redes *Wireshark*. Reitera-se que são utilizados dois protocolos de comunicação distintos: o UDP/IP (para a comunicação entre o servidor do *OpenPLC* e os módulos Modelo do Processo e Visualização do Processo) e o Modbus TCP/IP (para a comunicação entre o servidor do *OpenPLC* e o módulo Processo de Manufatura).

Na operação do *OpenPLC*, a sequência de comunicação (ciclo padrão de 500ms) é dada por:

- 1) Envio das mensagens de comando para o módulo Processo de Manufatura;
- 2) Recebimento das mensagens de comando para o módulo Processo de Manufatura;
- 3) Envio das mensagens de comando para os módulos Modelo do Processo e Visualização do Processo;
- 4) Recebimento das mensagens de resposta do módulo Modelo do Processo.

A Figura 11 apresenta as medições de latência realizadas.

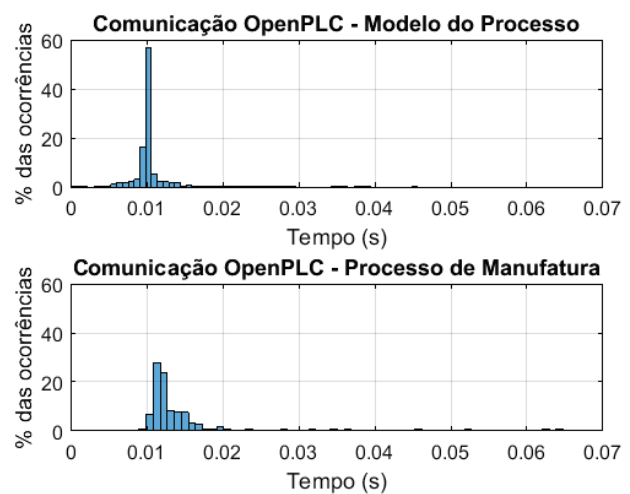


Fig. 11. Histogramas das latências de comunicação entre o *OpenPLC* e o modelo do processo (protocolo UDP/IP) e entre o *OpenPLC* e o hardware de E/S do processo de manufatura (protocolo Modbus TCP/IP) na arquitetura de GD.

Nota-se que a comunicação UDP/IP apresenta baixa variabilidade, enquanto na comunicação Modbus TCP/IP observa-

se maior variabilidade, mas de pequena amplitude, decorrente da comunicação sem fio entre o servidor do *OpenPLC* e o *hardware* de controle. Ambas são suficientemente rápidas para a execução das tarefas de controle do processo automatizado e sincronização com o espaço virtual.

## V. RESULTADOS E DISCUSSÕES

O grupo composto pela estação física, o modelo de simulação a eventos discretos (TET) e o módulo de visualização 3D é o GD implementado, cujas partes físicas e virtuais são integradas através de infraestrutura de rede. No cenário experimental, foi possível visualizar a replicação dos estados da planta física no lado virtual, bem como a convergência e interação entre os lados real e virtual. Assim, a arquitetura preenche os requisitos de um GD conforme descrito na literatura.

Para demonstração, um vídeo do sistema em operação está disponível no YouTube (<https://youtu.be/Gm6bLytgsxE>), no qual é possível visualizar a operação da planta física e a replicação dos estados no ambiente virtual.

Durante a fase de implementação do projeto, foi possível verificar os principais aspectos funcionais da arquitetura. O *OpenPLC* foi capaz de prover todas as funcionalidades de um CLP industrial, incluindo as cinco linguagens de programação da norma IEC 61131-3, os protocolos de comunicação industrial e as diversas opções de *hardware* de E/S para estabelecer o controle dos processos físicos. O *OpenPLC* também possibilitou a implementação da comunicação UDP entre os diversos módulos funcionais.

A TET proporcionou modelagem adequada para o sistema a eventos discretos utilizado no cenário experimental. Ela é baseada numa abordagem de máquinas de estado, permitindo programação na forma de condicionais *if-else* e condições de transição definidas pelo usuário, que podem ser baseadas em sinais de entrada ou em temporização. Uma vez que os blocos de TETs são parte do ambiente de simulação *Simulink*, as tabelas podem ser utilizadas em conjunto com os blocos de comunicação UDP nativos, disponíveis na biblioteca de blocos. Assim, a TET que descreve o comportamento lógico da planta de manufatura pôde ser alimentada em tempo de execução com dados do CLP, replicando assim os estados do sistema físico no ambiente virtual para finalidades de comparação e diagnóstico. Por padrão, a interface UDP *Sim-Link* dispara pacotes UDP a cada 50ms, resultando numa frequência de atualização de 20Hz da TET, bem como do aplicativo de visualização 3D. Este valor foi adequado para a execução sincronizada do gêmeo digital e serviria também para aplicações de variáveis contínuas. Em [21], por exemplo, utiliza-se uma taxa de atualização de 100ms para a simulação de movimentos de robôs.

A ferramenta de visualização 3D foi adaptada para suportar o padrão de mensagens UDP encaminhadas através do *Open-PLC* e *software* auxiliar de interface. Utilizando um modelo CAD animado, foi possível prover uma experiência de realidade virtual com custo computacional aceitável. Considerando o uso do sistema em rede local, o atraso entre a ocorrência de um evento na planta real e a correspondente exibição no ambiente 3D é imperceptível por observação. Quanto à criação de

modelos 3D animados, verificou-se que trata-se de uma tarefa complexa e trabalhosa, apesar de fornecer maior semelhança visual com o processo real. As atividades de modelagem 3D do processo e animações dos objetos (parametrização de propriedades, interação, vinculação e sequenciamento) não são simples e demandam elevado tempo para realização.

O uso do aplicativo *Factory I/O* oferece uma alternativa para desenvolvimento do módulo de visualização do processo, com tempo de desenvolvimento substancialmente menor em relação à construção de modelos CAD animados. Entretanto, resulta em menor semelhança visual com a estação real, uma vez que os modelos são construídos apenas com peças disponíveis na biblioteca do aplicativo, que não pode ser personalizada. Neste caso, a conectividade é feita via protocolo Modbus TCP/IP, que é aceito pelo *OpenPLC*. Quanto à sincronização da velocidade de execução, o *software* dispõe de um parâmetro ajustável de configuração da velocidade de execução do modelo, a qual foi suficiente para sincronizar a operação real com o modelo virtual. A criação de ambos os modelos virtuais (TET e visualização 3D) não está limitada às ferramentas de *software* apresentadas. Outros programas apropriados podem ser utilizados na arquitetura, desde que estejam aptos a se comunicar com o *OpenPLC* através dos protocolos UDP ou Modbus TCP/IP.

Na arquitetura de GD implementada, a comunicação entre o *OpenPLC* e o Processo de Manufatura é realizada através do protocolo Modbus TCP/IP (rede sem fio Wi-Fi) e a comunicação entre o *OpenPLC* e os módulos de Modelo do Processo e Visualização 3D são através do protocolo Ethernet UDP/IP. Considerando a capacidade de processamento computacional do servidor do *OpenPLC* (e dos módulos da arquitetura) e que a comunicação entre o *OpenPLC* e os módulos é muito rápida (baixo número de mensagens), é possível obter o sincronismo na execução do GD. A latência da comunicação em rede entre os módulos existe, porém possui uma variabilidade muito pequena e que poderia ser desprezada, pois não tem impacto significativo na operação do GD quando comparado com a dinâmica de execução do processo de manufatura. Para fins de comparação, em [6] e [9] os autores constataam latência inferior a 1s entre a execução real e o modelo virtual. Na arquitetura de GD deste trabalho, a latência está diretamente relacionada à comunicação UDP, o que permite latência inferior aos exemplos da literatura.

Em conformidade com os conceitos da Indústria 4.0, a comunicação sem fio foi priorizada nesta arquitetura. Todos os dispositivos foram integrados por meio de uma rede sem fio implementada num roteador local. Cada módulo virtual pode ser implementado numa máquina separada, desde que se faça o controle do endereçamento IP para evitar erros de comunicação. Como os testes foram executados numa rede restrita com pouco tráfego, não houve problemas quanto a perda de pacotes ou atrasos anormais de comunicação.

## VI. CONCLUSÕES E PERSPECTIVA FUTURA

Este trabalho apresentou o desenvolvimento de uma arquitetura modular e flexível para construção de GDs com foco nos conceitos e ferramentas do contexto da Indústria 4.0. Foi



desenvolvida uma arquitetura de implementação centrada na ferramenta de controle de processos *OpenPLC*, que permite a comunicação de dados via redes *Ethernet* por meio de uma interface de comunicação UDP. Embora a comunicação via UDP não seja nativa é possível, com recurso auxiliar, acessar os *buffers* do *OpenPLC* e realizar a transmissão dos *bits* de E/S através de portas UDP definidas pelo usuário. A construção da arquitetura priorizou o uso de ferramentas *open-source*, em contraposição a diversos trabalhos encontrados na literatura que se utilizam de soluções de automação industrial proprietárias disponibilizadas a alguns pesquisadores por meio de acordos entre empresas e universidades, mas não disponíveis para a comunidade acadêmica em geral. Outro fundamento foi a manutenção da modularidade da plataforma, de modo a permitir a expansão, inclusão ou remoção de módulos de forma a impactar o mínimo possível na operação dos demais.

O cenário experimental apresentou uma implementação prática da arquitetura. O GD foi construído a partir da integração entre o processo de automação real, o módulo de visualização 3D e o aplicativo de simulação rodando paralelamente em tempo de execução. O ambiente virtual converge com o real, produzindo dados fidedignos para a validação em tempo de execução e permitindo que o sistema de controle implemente ações na planta real com base nos dados obtidos por meio de simulação. Sendo a TET um modelo de comportamento lógico do sistema, foi implementado um módulo de visualização 3D que complementa o ambiente virtual do sistema. Assim, o espaço real é replicado tanto em seu comportamento quanto em seus aspectos visuais em ambiente virtual. O modelo 3D implementado no aplicativo *Factory I/O* demonstrou a flexibilidade e a expansibilidade da arquitetura, mostrando que é possível agregar diversas aplicações ao conjunto, desde que disponham de um protocolo de comunicação que seja suportado pelo *OpenPLC*.

Uma limitação encontrada no desenvolvimento deste trabalho foi a indisponibilidade de uma ferramenta que permitisse a implementação de formalismos de modelagem de SEDs (tais como autômatos finitos, Redes de Petri e outros) de forma interativa, ou seja, permitindo que as transições ocorram à medida que sinais externos são fornecidos ao simulador. As TETs foram a ferramenta que se mostrou mais adequada ao projeto por conta da disponibilidade de blocos de comunicação UDP, que permitiram a interatividade desejada. Por outro lado, as TETs não podem ser definidas como um método formal de modelagem.

Futuramente espera-se incluir um módulo de aquisição e análise de dados do processo produtivo, que permitirá a implementação de métricas de avaliação do processo industrial para que seja viável sua análise e otimização em nível gerencial. Este módulo deve proporcionar uma análise mais profunda dos dados gerados nos espaços real e virtual, implementando técnicas típicas do contexto de gerenciamento industrial (tais como MES e outros) para produzir alarmes, previsões e auxiliar na detecção de gargalos. Espera-se ainda estudar a distribuição dos módulos entre diversos computadores, de forma a avaliar o impacto das oscilações de comunicação em rede no funcionamento da arquitetura. Outro tópico a ser analisado é a expansão da arquitetura para uso em redes de maior

tráfego e, em última instância, na Internet. Deve-se avaliar o efeito da inserção de tráfego na rede, uma vez que todos os experimentos foram realizados localmente. A integração da arquitetura à Internet agregaria conectividade, mas apresentaria também novos desafios quanto à implementação de estratégias de segurança. Por se tratar de processos simultâneos que requerem comunicação em tempo de execução, é necessário evitar o excessivo aumento de tamanho dos pacotes de dados a fim de manter a velocidade e evitar a sobrecarga da rede.

## AGRADECIMENTOS

Os autores agradecem a cooperação do professor Fabrício Junqueira no desenvolvimento do módulo de visualização 3D.

## REFERÊNCIAS

- [1] M. A. Pessoa, M. A. Pisching, L. Yao, F. Junqueira, P. E. Miyagi, and B. Benatallah, "Industry 4.0, how to integrate legacy devices: A cloud iot approach," in *IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2018, pp. 2902–2907.
- [2] N. Jazdi, "Cyber physical systems in the context of industry 4.0," in *Automation, Quality and Testing, Robotics, 2014 IEEE International Conference on*. IEEE, 2014, pp. 1–4.
- [3] R. A. Diogo, C. A. Vicari, E. de FR Loures, M. A. Busetti, and E. A. Santos, "An implementation environment for automated manufacturing systems," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 10552–10557, 2008.
- [4] B. Schleich, N. Anwer, L. Mathieu, and S. Wartzack, "Shaping the digital twin for design and production engineering," *CIRP Annals*, vol. 66, no. 1, pp. 141–144, 2017.
- [5] F. Tao and M. Zhang, "Digital twin shop-floor: a new shop-floor paradigm towards smart manufacturing," *Ieee Access*, vol. 5, pp. 20418–20427, 2017.
- [6] Y. Zheng, S. Yang, and H. Cheng, "An application framework of digital twin and its case study," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–13, 2018.
- [7] H. Zipper, F. Auris, A. Strahilov, and M. Paul, "Keeping the digital twin up-to-date — process monitoring to identify changes in a plant," in *2018 IEEE International Conference on Industrial Technology (ICIT)*, Feb 2018, pp. 1592–1597.
- [8] W. Kritzinger, M. Karner, G. Traar, J. Henjes, and W. Sihn, "Digital twin in manufacturing: A categorical literature review and classification," *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 1016–1022, 2018.
- [9] J. Vachálek, L. Bartalský, O. Rovný, D. Šišmišová, M. Morháč, and M. Lokšík, "The digital twin of an industrial production line within the industry 4.0 concept," in *Process Control (PC), 2017 21st International Conference on*. IEEE, 2017, pp. 258–262.
- [10] T. R. Alves, M. Buratto, F. M. de Souza, and T. V. Rodrigues, "Openplc: An open source alternative to automation," in *Global Humanitarian Technology Conference (GHTC), 2014 IEEE*. IEEE, 2014, pp. 585–589.
- [11] M. Grieves, "Digital twin: manufacturing excellence through virtual factory replication," *White paper*, 2014.
- [12] F. Tao, J. Cheng, Q. Qi, M. Zhang, H. Zhang, and F. Sui, "Digital twin-driven product design, manufacturing and service with big data," *The International Journal of Advanced Manufacturing Technology*, vol. 94, no. 9–12, pp. 3563–3576, 2018.
- [13] E. Glaessgen and D. Stargel, "The digital twin paradigm for future nasa and us air force vehicles," in *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA*, 2012, p. 1818.
- [14] F. Tao, F. Sui, A. Liu, Q. Qi, M. Zhang, B. Song, Z. Guo, S. C.-Y. Lu, and A. Nee, "Digital twin-driven product design framework," *International Journal of Production Research*, pp. 1–19, 2018.
- [15] M. Kunath and H. Winkler, "Integrating the digital twin of the manufacturing system into a decision support system for improving the order management process," *Procedia CIRP*, vol. 72, no. 1, pp. 225–231, 2018.
- [16] Y. Liao, H. Lee, and K. Ryu, "Digital twin concept for smart injection molding," in *IOP Conference Series: Materials Science and Engineering*, vol. 324, no. 1. IOP Publishing, 2018, p. 012077.

- [17] C. Zhuang, J. Liu, and H. Xiong, "Digital twin-based smart production management and control framework for the complex product assembly shop-floor," *The International Journal of Advanced Manufacturing Technology*, vol. 96, no. 1-4, pp. 1149–1163, 2018.
- [18] C. G. Cassandras and S. Lafortune, *Introduction to discrete event systems*. Springer Science & Business Media, 2009.
- [19] G. J. Santos and H. de Carvalho, "Projeto e desenvolvimento de interface tridimensional para sistema supervisor e de treinamento," Escola Politécnica da Universidade de São Paulo, 2016.
- [20] R. Games, "Factory i/o (documentation)," jan 2019, disponível em: <https://factoryio.com/docs/>. Acesso em: 07 jun. 2019.
- [21] M. Saez, F. P. Maturana, K. Barton, and D. M. Tilbury, "Real-time manufacturing machine and system performance monitoring using internet of things," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 4, pp. 1735–1748, 2018.



**Rodrigo Pita Rolle** received the B. Eng. degree in Control and Automation Engineering at Universidade Tecnológica Federal do Paraná (UTFPR) (PR - Brazil) in 2016. Currently he is a M.Sc student in Electrical Engineering at São Paulo State University (Unesp) (SP - Brazil). His research interests include automation, remote control and supervision, failure detection in industrial plants, digital twins and Industry 4.0. <http://lattes.cnpq.br/7657925458356954>



**Vinicius de Oliveira Martucci** is a B. Eng. student in Control and Automation Engineering at São Paulo State University (Unesp) (SP - Brazil). His research interests include automation, remote control and supervision, failure detection in industrial plants, digital twins, Internet of Things, Industry 4.0. <http://lattes.cnpq.br/0863261000402374>



**Eduardo Paciência Godoy** received the B.Eng. degree in Control and Automation Engineering at Itajubá Federal University (MG-Brazil) in 2003 and the M.Sc. and Ph.D. degrees in Mechanical Engineering at University of São Paulo at São Carlos (SP-Brazil) in 2007 and 2011, respectively. Currently he is an Associate Professor of the São Paulo State University (Unesp) at Sorocaba (SP-Brazil). His research interests include industrial networks and automation, networked control systems (NCS), wireless networks and telemetry, embedded electronics, Internet of Things and Industry 4.0. <http://lattes.cnpq.br/0072632067545698>.