

# Industrial Automation as a Service: A New Application to Industry 4.0

Michel M. Fernandes, Jeferson A. Bigheti, Ricardo P. Pontarolli and Eduardo P. Godoy

**Abstract** — The fourth industrial revolution aims to achieve greater productivity, also seeking to improve the quality and efficiency of production processes. For this, the use of information and automation technologies together becomes indispensable. Collaborative automation through the sharing and use of services has been a recent paradigm in the quest to obtain a distributed, flexible and integrated on network architecture. In this article, the development of control and automation applications as a service using a Microservices Oriented Architecture is investigated. The architecture is elaborated using the Molecular framework, which simplifies the development of microservices and applications that make up the Microservice Oriented Architecture used in this work. The development and evaluation of several services are presented, such as Data Acquisition (DAQ), Programmable Logic Controller (PLC) and PID Controller (PIDPlus). The experimental results in different scenarios demonstrated the feasibility of implementing control and automation applications as a service. A relevant advantage that this type of architecture provides is interoperability through automatic network communication between applications and microservices, which is required for Industry 4.0.

**Index Terms**— OpenPLC, Service Oriented Architecture, Molecular Framework, Industrial Internet of Things.

## I. INTRODUÇÃO

A Quarta Revolução Industrial, conhecida no Brasil como Manufatura Avançada, nos Estados Unidos da América como *Smart Factory* e na Alemanha como Indústria 4.0 (I4.0), caracteriza o avanço dos sistemas de produção atuais fundamentado na convergência entre as tecnologias operacionais, automação e Tecnologia da Informação (TI) [1]. O desafio é promover a integração das tecnologias necessárias para a implementação desse conceito, visando obter uma nova realidade produtiva, totalmente integrada para proporcionar melhores decisões em relação aos sistemas produtivos, fazendo o uso de informações disponibilizadas em tempo real [1].

O desenvolvimento de sistemas industriais por meio da Internet das Coisas Industrial (IIoT - *Industrial Internet of Things*) e I4.0 envolve a integração entre tecnologias, equipamentos e sistemas de automação e controle alocados em diferentes níveis hierárquicos, proporcionando a descentralização, modularidade e independência dos sistemas. Nesse sentido, um paradigma recente é a automação colaborativa com o compartilhamento de serviços, obtendo uma arquitetura flexível, escalável e integrada à rede [2].

This study was financed in part by the São Paulo Research Foundation (FAPESP) grant 2018/19984-4.

M.M. Fernandes, J.A. Bigheti, R.P. Pontarolli, E. P. Godoy are with São Paulo State University (Unesp), Sorocaba, SP 18087-180, Brazil (eduardo.godoy@unesp.br).

A Arquitetura Orientada a Serviços (SOA - *Service Oriented Architecture*) permite o fornecimento de serviços em nuvem alocados em diferentes dispositivos, facilitando e padronizando as interações entre eles, podendo assim ser considerada um caminho promissor para os desafios da implantação da I4.0 [3].

A SOA é uma arquitetura de sistemas de TI na qual se busca fragmentar as aplicações convencionais em “Serviços” [4]. Na prática, trata-se de uma arquitetura modular, na qual os elementos devem ser compostos de acordo com as necessidades do negócio. Neste tipo de sistema, cada serviço deve ser passível de descoberta e independente quanto à plataforma e à linguagem [5], com o objetivo de obter baixo acoplamento de um serviço em relação ao outro, escalabilidade e a capacidade de manter uma arquitetura distribuída. Dentro deste contexto surge a ideia de microsserviços, ou seja, serviços com poucas responsabilidades, pequenos e autônomos que podem trabalhar de forma independente ou em conjunto com outros serviços.

Com o uso de arquiteturas de serviços ou microsserviços, informações oriundas de sistemas que possuem características heterogêneas diversas podem ser adquiridas de modo transparente ao usuário ou à algum tipo de aplicação. Dessa forma, acontece a migração da tradicional arquitetura hierárquica em camadas estabelecida pela ISA-95 (*International Society of Automation*) para uma arquitetura voltada a serviços em nuvem. Nesse contexto, a ideia de virtualizar equipamentos e dispositivos industriais evolui para um conceito de disponibilização de seus recursos e funcionalidade com um serviço [6].

No sentido de desenvolver novas alternativas para a implementação da I4.0, este artigo apresenta uma nova proposta de desenvolvimento de aplicações de automação industrial com um serviço. Essas aplicações de automação com um serviço contemplam funcionalidades de aquisição de dados, programação no padrão IEC 61131-3, controle contínuo e discreto e supervisão de dados industriais implementados neste artigo como microsserviços.

Este artigo está apresentado da seguinte forma. A seção II apresenta uma revisão da literatura sobre aplicações industriais usando serviços. A descrição da arquitetura orientada a microsserviços utilizada, bem como o desenvolvimento dos serviços e aplicações é apresentada na seção III. A seção IV discute e compara os resultados de diversas aplicações de automação e controle como um serviço. A seção V apresenta uma discussão sobre o artigo. As conclusões e contribuições deste artigo são elencadas na seção VI.

## II. APLICAÇÕES INDUSTRIAIS ORIENTADAS A SERVIÇOS

A evolução das arquiteturas de automação e controle industriais evidencia a tendência para a convergência do uso das

tecnologias de automação e da informação. A primeira arquitetura era centralizada e com comunicação ponto a ponto entre equipamentos. A segunda e a terceira incorporaram o conceito de redes de comunicação, permitindo a distribuição dos equipamentos, redução de custos e facilidade de manutenção. Adicionalmente, a terceira arquitetura incorporou as chamadas arquiteturas de controle via redes (*NCS – Networked Control Systems*), permitindo a descentralização do controle, modularização e flexibilização [21]. O uso das redes sem fio em NCS, promovendo interoperabilidade entre redes com fio existentes e novas redes sem fio nortearam o surgimento dos sistemas de controle via redes sem fio (*WNCS – Wireless Networked Control Systems*).

A quarta e atual arquitetura é a baseada nos Sistemas Ciber Físicos (*CPS - Cyber Physical Systems*) alavancados pela integração total demandada pela IIoT e I4.0. Nessa nova arquitetura, um sistema de armazenamento de informações em nuvem é compartilhado pelos equipamentos e sistemas, tornando possível o uso de serviços padronizados para comunicação entre os mesmos. Os serviços podem ser acessados por aplicações, sistemas e outros serviços independentemente de onde eles estão alocados, propiciando uma arquitetura colaborativa [21].

Na SOA, as camadas dos distintos níveis hierárquicos da arquitetura ISA-95 inexistem, portanto, a comunicação entre os elementos não é somente entre camadas adjacentes, sendo assim, cada elemento pode fornecer dados para os demais que compõem a arquitetura. O conceito de serviços concede à esses elementos a possibilidade de serem clientes ou servidores, dependendo da necessidade da aplicação ou do processo, envolvendo a interação com base na troca de mensagens, entre os elementos que compõem os níveis hierárquicos de um processo industrial [7]. Além do mais, a SOA possibilita que qualquer tipo de aplicação industrial, em um contexto de CPS, IIoT e I4.0, possa ser composta por meio da combinação e seleção de novos serviços disponibilizados em nuvem.

Uma diversidade de pesquisas produziram SOA e analisaram a sua aplicação em sistemas industriais inseridos no contexto de IIoT e I4.0 [8], [9], [10] e [11]. Em [8] é afirmado que SOA é uma arquitetura padronizada para comunicação distribuída entre sistemas e dispositivos, independentemente dos protocolos. A SOA oferece mecanismos para a descoberta, autonomia, composição e abstração de serviços baseados em padrões, proporcionando uma importante contribuição para as aplicações industriais [10]. A SOA facilita a criação e reuso de serviços, promovendo escalabilidade, modularidade e disponibilidade para as aplicações industriais [11]. Uma discussão sobre os desafios da migração de sistemas de controle de processo tradicionais (legados) para SOA é apresentado em [12], exemplificando através de um caso de estudo de migração de uma planta de grande porte com diferentes equipamentos e redes de comunicação.

Considerando aplicações no contexto de automação e controle usando SOA, uma revisão detalhada sobre alguns projetos Europeus é apresentada em [10], principalmente os projetos SOCRADES [22] e IMC-AESOP [23]. Estes projetos demonstram a utilização de SOA em todas as camadas

hierárquicas da ISA-95, desde o nível de Dispositivos, percorrendo pelo nível de Controle e Supervisão, até nível de Sistemas Corporativos. Os projetos ARROWHEAD [24] e o PRODUCTIVE 4.0 [25] derivam dos projetos anteriormente citados e apresentam a atual relevância da temática de SOA para atingir os novos objetivos da Indústria 4.0.

Analisando esses projetos, dois pontos importantes podem ser citados. Primeiro que eles desenvolveram toda uma infraestrutura computacional de SOA para a aplicação industrial. E segundo que diante disso, utilizaram diferentes meios de comunicação entre serviços para interconexão entre dispositivos e aplicações como o DPWS (*Device Profile for Web Services*), REST (*Representational State Transfer*) e OPC UA (*Open Platform Communications Unified Architecture*) [13]. Em contraste com esse pensamento, essa pesquisa se concentrou no uso de um *framework* de serviços que fornece um serviço para comunicação entre os dispositivos e aplicações, simplificando o desenvolvimento e investigando seu uso para aplicações industriais.

Na área de computação, a difusão e evolução da SOA originou o conceito de Arquiteturas Orientadas a Microserviços (MOA). Ainda que várias pesquisas ligadas a aplicações industriais de SOA possam ser encontrados na literatura, o mesmo não ocorre quando se foca em MOA [17]. Em [14] é apresentada uma revisão a respeito da utilização industrial da MOA, concluindo que o uso dessa arquitetura ainda é bastante incipiente [14], [15].

Dois projetos europeus focam na implementação da MOA para aplicações inseridas no contexto de I4.0, CPS e IIoT. Em [16] é apresentada uma proposta de arquitetura orientada a microserviços para a implementação do conceito de fábrica digital e se diferencia pela criação de gêmeos digitais. Cinco grupos principais de serviços são propostos nessa arquitetura, na qual a comunicação é realizada via padrão REST com HTTP (*Hypertext Transfer Protocol*) e WebSocket via TCP/IP (*Transmission Control Protocol / Internet Protocol*). Os microserviços Orquestrador (*Orchestrator*) e Agendador (*Scheduler*) podem ser destacados, os quais organizam e coordenam os demais serviços para viabilizar a criação e composição de serviços de alto nível e aplicações de processo. Uma proposta de MOA em nuvem para monitoramento, otimização e planejamento das cadeias de fornecimento de manufatura baseada em IIoT é apresentada [17].

Seguindo essa tendência e demanda por aplicações industriais usando MOA, essa pesquisa apresenta as seguintes contribuições: apresentar uma nova proposta de desenvolvimento de aplicações de automação industrial usando microserviços (MOA), avaliar o uso do *framework Molecular* de microserviços para aplicações industriais e demonstrar a viabilidade do desenvolvimento de aplicações industriais de automação e controle usando uma arquitetura de microserviços.

### III. ARQUITETURA ORIENTADA A MICROSERVIÇOS (MOA)

#### A. Descrição da Arquitetura

O *framework Molecular* é uma estrutura de desenvolvimento que utiliza a linguagem *JavaScript* para

aplicações com arquiteturas orientadas a microsserviços (MOA) [23]. O *Molecular* é executado sobre a plataforma *Node.js* e possui código aberto (*open source*). Na Fig. 1 pode ser vista a arquitetura do *Molecular*, onde os serviços são executados em estruturas individuais que se comunicam por meio de um serviço de comunicação (*transporter*). Uma das vantagens do *Molecular* é que todos os microsserviços possuem um recurso de registro e descoberta automático. Outro recurso que se destaca é o balanceamento de carga, que possui a função de distribuir a carga da comunicação entre os microsserviços de maneira dinâmica e uniforme.

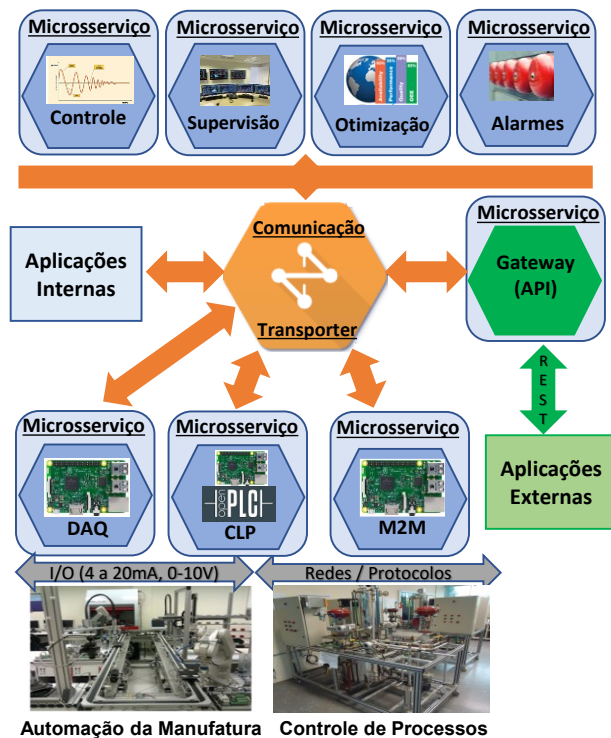


Fig. 1. Automação Industrial como um Serviço: Arquitetura de Microsserviços usando o *Molecular*.

Na arquitetura da Fig. 1, os nós de gerenciamento do *Molecular* podem conter um ou mais serviços. Cada serviço contido na arquitetura é representado por um hexágono azul. Esses serviços podem executar e oferecer tarefas distintas que são denominadas de ações. O microsserviço de gateway é representado pelo hexágono verde. Sua função viabilizar a comunicação das aplicações externas com os serviços internos (hexágonos azuis), sendo basicamente uma interface de conexão. A comunicação entre os microsserviços é realizada por meio de um serviço de comunicação (*Transporter*), representado pelo hexágono laranja. Esse fato é um grande diferencial para o uso da arquitetura orientada a serviços, pois evita a necessidade de programação relacionado à comunicação em rede. As Aplicações, que utilizam vários microsserviços, são apresentadas por retângulos (verde e azul).

Na Fig. 1 apresenta a MOA para aplicações de controle de processos e automação focadas na 4.0 [3] usada neste artigo. A MOA é composta por microsserviços divididos em dois níveis hierárquicos, sendo eles, os Microsserviços de Infraestrutura e

os Microsserviços de Negócios/Processos. Os microsserviços que compõem a infraestrutura como Gateway, Comunicação (*Transporter*), Aquisição de Dados (DAQ), Controlador Lógico Programável (CLP) e Comunicação Máquina a Máquina (M2M) são encarregados de executar as funções básicas necessárias para a composição dos serviços de nível superior nas aplicações. Os microsserviços de Negócios/Processos executam as funções de alto nível como Supervisão, Controle, Alarmes e Otimização, e necessitam operar em composição com os demais serviços.

O desenvolvimento e implementação de mecanismos de segurança para a MOA desenvolvida não foi foco deste artigo. No entanto, é importante enfatizar que essa arquitetura é compatível e suporta diferentes mecanismos diferentes. Para a arquitetura desenvolvida, propomos a diferentes mecanismos de segurança que serão implementados futuramente. Entre ele é possível citar o controle de acesso do desenvolvedor aos microsserviços via chave criptografada, uso de requisições seguras no gateway via HTTPs, autenticação de usuários com token, opções de conexão segura com o serviço de comunicação (*Transporter*), desenvolvimento de um serviço de guarda para controle de acesso entre microsserviços usando *JSON Web Token*. A implementação desses mecanismos proporcionará para a MOA os níveis de segurança requeridos em aplicações industriais.

#### B. Desenvolvimento dos Serviços e Aplicações

Nas arquiteturas a microsserviços, uma aplicação é desenvolvida por meio da composição dos microsserviços desenvolvidos. Essa composição define quais microsserviços serão usados e a sequência de execução desses microsserviços para obtenção da funcionalidade requerida. A composição dos microsserviços pode ser via Orquestração ou Coreografia. Nesta pesquisa foi utilizado a composição por orquestração.

Neste artigo foram usados os microsserviços Controle, DAQ e CLP mostrados na Fig. 1. Todos esses serviços foram implementados em sistema embarcado (Raspberry Pi 3B+ com distribuição Raspberry Pi OS). O Microsserviço de Controle foi desenvolvido utilizando uma versão modificada do algoritmo *PIDPlus* [18], desenvolvido para aplicações de controle via rede. Este serviço é responsável pelo controle de malhas de processo. Ele também recebe alguns parâmetros de entrada do controlador, como os ganhos do controlador, os quais também podem ser sintonizados através da comunicação entre os serviços.

O Microsserviço de Aquisição de dados (DAQ) é responsável pela aquisição de dados de variáveis usando módulos de hardware alocados no processo. O microsserviço DAQ possui uma ação de leitura de entradas (sensores) e uma ação para atualizar as saídas (atuadores). O Microsserviço CLP incorpora as funções de um CLP na arquitetura, através da integração do *OpenPLC* [19] como um microsserviço na estrutura do *Molecular*. O *OpenPLC* reúne as funcionalidades de um CLP com as vantagens de uma arquitetura aberta de *software* e hardware. O Microsserviço CLP é capaz de executar lógicas e sequenciamento de tarefas que são executadas em CLPs convencionais, bem como a aquisição e o monitoramento

de dados de entradas e saídas do controlador. No entanto, o diferencial desse artigo é que ao ser disponibilizado como um serviço, o microserviço CLP pode ser replicado e implantado em múltiplas plataformas como computador, sistema embarcado ou nuvem.

As aplicações desenvolvidas na arquitetura da Fig. 1 podem ser do tipo Internas ou Externas. As Aplicações Internas são inteiramente implementadas nas plataformas suportadas pelo *Molecular* (JavaScript, Node.js, Java, .NET e Go) [20]. Nesse tipo de aplicação, a comunicação entre os microserviços é direta por meio do microserviço *Transporter*. As Aplicações Externas são as implementadas em qualquer outra plataforma de desenvolvimento distinta das suportadas pelo *Molecular*. Para essas aplicações, a comunicação entre os microserviços necessita ser realizada usando o padrão REST via microserviço Gateway. A aplicação externa fornece compatibilidade de uso da arquitetura da Fig. 1 em aplicações industriais, onde as plataformas de *software* para desenvolvimento de aplicações são consolidadas e tradicionais.

Neste artigo, duas aplicações foram desenvolvidas: Aplicação de Monitoramento e Aplicação de Controle. A Aplicação de Monitoramento é encarregada da aquisição e monitoramento de dados de malhas de processo usando a MOA. A aquisição de dados é realizada através da requisição do serviço DAQ. O objetivo dessa aplicação é o desenvolvimento de uma interface para monitoramento das variáveis de processo e também para a análise do comportamento dessas variáveis e/ou desempenho do controle ao longo do tempo. A Aplicação de Controle é responsável pela efetivação do controle em malha fechada de variáveis de interesse, por meio da composição dos microserviços DAQ e Controle PIDPlus.

Na Fig. 2 pode ser observado um exemplo de comparação entre a orquestração de microserviços de uma Aplicação Interna e de uma Aplicação Externa de Controle em malha fechada usando a MOA da Fig. 1. Nos diagramas das aplicações é possível verificar a sequência de execução dos microserviços definida em cada aplicação (de 1 a 12 no caso da Aplicação Interna e de 1 a 18 no caso da Aplicação Externa). Conforme citado, constata-se que a diferença entre as sequências de execução dos microserviços está na necessidade pela Aplicação Externa de uso do serviço Gateway (API) para acessar os serviços DAQ e Controle (PID+) necessários para implementação da malha de controle fechada. A Aplicação Interna é capaz de se comunicar diretamente via serviço e comunicação (T).

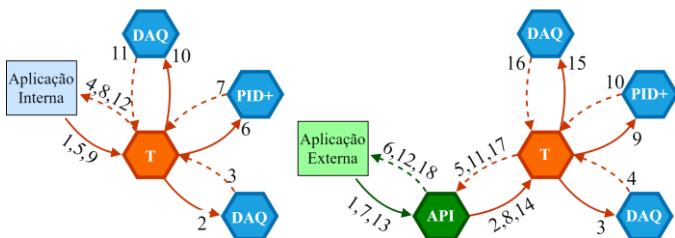


Fig. 2. Aplicações (Interna e Externa) no Molecular por Orquestração de Serviços.

IV. RESULTADOS E DISCUSSÕES

Buscando avaliar e comprovar o uso da arquitetura a microserviços da Fig. 1 para aplicações de controle de processos e automação, experimentos em cenários de aplicações industriais foram realizados. O serviço de comunicação (*Transporter*) padrão do *Molecular* utilizado para realizar a comunicação foi o NATS (sistema de mensagens de alto desempenho baseado em filas distribuídas para aplicativos). O serviço API Gateway (Nó 1) efetua a comunicação (REST) entre as aplicações externas de automação industrial com os microserviços desenvolvidos no *Molecular*.

A. Experimentos com Orquestração de Microserviços

Nos primeiros experimentos realizados, foi utilizada uma planta didática como malha de controle de velocidade de motor CC. A estrutura geral do experimento realizado envolvendo a MOA é demonstrada na Fig. 3. As Aplicações Externas de controle de processos por orquestração implementam a sequência de execução dos microserviços mostrada na Fig. 2. Nesse experimento, toda comunicação entre os serviços (*Transporter*) e via Gateway foi realizado via rede sem fio Wi-Fi.

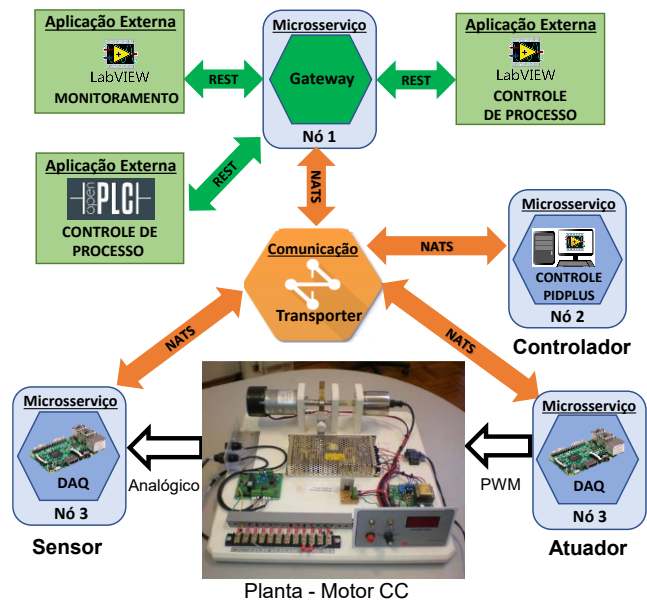


Fig. 3. Estrutura do Experimento de Controle de Processo da Planta Didática com Orquestração dos Serviços DAQ e PIDPlus.

Aplicação Industrial via LabVIEW

No primeiro experimento da Fig. 3, duas aplicações externas foram desenvolvidas a partir da orquestração dos microserviços Gateway, *Transporter*, DAQ e Controle PIDPlus, sendo elas, a aplicação de Monitoramento e a de Controle de Processos. Essas duas aplicações foram desenvolvidas por meio do *software* LabVIEW. Os serviços API Gateway e *Transporter*, bem como as aplicações externas Controle e Monitoramento são executados em um computador.

O serviço DAQ (Nó 3) coleta os dados da variável de processo (PV) e atualiza a variável manipulada (MV) na planta.



A variável de processo (PV) é obtida pelo sensor que utiliza um sinal analógico. O atuador recebe os sinais disponibilizados pelo serviço de Controle (Nó 2) e atua na planta através de um canal PWM. O serviço de Controle PIDPlus (Nó 2) efetua o cálculo do controle de processo, a partir dos dados coletados pelo sensor (velocidade do motor – PV) e atua sobre o motor (tensão aplicada – MV).

A Fig. 4 apresenta a saída (resposta) do controle de velocidade do motor, armazenadas pela aplicação de Supervisão. As curvas de saída (Fig. 4) mostram a resposta de velocidade do motor em relação à referência estabelecida, mostrando que o controle do motor é estável com rastreamento de referência definido.

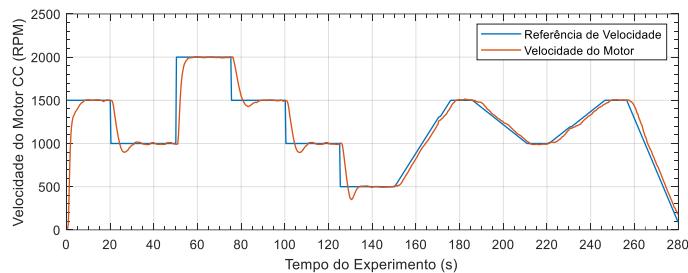


Fig. 4. Aplicação de Controle via LabVIEW usando Serviços da Fig. 3.

Dados estatísticos relacionados à medição do tempo de orquestração (ou sequência) da execução dos serviços mostraram que o tempo de resposta usando a arquitetura de microsserviços é adequada e compatível com aplicações de automação e controle de processos. O valor médio do tempo total de orquestração foi de 105,5 ms e sua variabilidade (desvio padrão) foi de 43,9 ms.

### Aplicação via IEC 61131-3

Em aplicações de automação industrial, geralmente são usados *software* de desenvolvimento de aplicações compatíveis com as linguagens de programação de controladores programáveis definidas pela IEC 61131-3: LD (Diagrama Ladder), FBD (Diagrama de Blocos), SFC (Diagrama de Funções Sequenciais), IL (Lista de Instruções), ST (Texto Estruturado). Para permitir essa funcionalidade de desenvolvimento de aplicações com a arquitetura a microsserviços (MOA), foi utilizado o ambiente de edição de programas do *OpenPLC* [19].

Este editor (*OpenPLC Editor*) possui compatibilidade com as cinco linguagens de programação definidas pela IEC 61131-3. Para a integração com a MOA, blocos lógicos customizados contendo códigos (funções em Linguagem C) de requisições via API REST do serviço Gateway da arquitetura foram desenvolvidos para o *OpenPLC Editor*. Os blocos desenvolvidos foram o “Serviço DAQ Ação de leitura”, “Serviço PIDPlus” e “Serviço DAQ Ação de saída”, que foram incluídos na biblioteca do *OpenPLC Editor* e podem ser usados em qualquer programa, como o desenvolvido para este experimento mostrado na Fig. 5.

Dessa forma, compatibilizou-se a comunicação entre os microsserviços e os recursos de programação do editor de programas do *OpenPLC*. Este programa é compilado e

executado no servidor do *OpenPLC*. Dessa maneira, o *OpenPLC* pode criar uma Aplicação Externa para composição dos microsserviços disponíveis na arquitetura da Fig. 1.

No segundo experimento da Fig. 3, uma Aplicação Externa de “Controle de Processos” foi desenvolvida a partir da orquestração dos microsserviços Gateway, *Transporter*, DAQ e Controle PIDPlus. A diferença deste experimento, em relação ao anterior, é que ele usa o *software* do *OpenPLC*. A Fig. 5 apresenta o programa do *OpenPLC* executando a sequência de orquestração de serviços para Controle de Processo. Nessa orquestração, da mesma forma como explicado no experimento anterior, a cada ciclo da malha de controle, a Aplicação Externa (*OpenPLC*) implementa a sequência de execução dos microsserviços mostrada na Fig. 2.

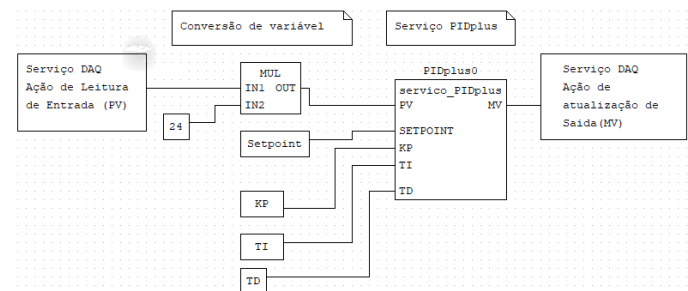


Fig. 5 .Aplicação Externa no padrão IEC-61131-3 no *OpenPLC* via Orquestração de Serviços.

O mesmo perfil de referência do experimento anterior foi usado neste experimento com a orquestração via *OpenPLC*. A Fig. 6 apresenta os resultados da curva de resposta em relação à referência (*setpoint*) estabelecida. A aplicação da MOA com orquestração de microsserviços via padrão IEC 61131-3 comprova a compatibilidade e flexibilidade de uso da arquitetura em aplicações industriais. É importante destacar que o reuso dos serviços proporciona escalabilidade e modularidade para as aplicações, facilitando a implantação de novas malhas de controle e padronizando o acesso às informações da planta através da linguagem padronizada na IEC 61131-3.

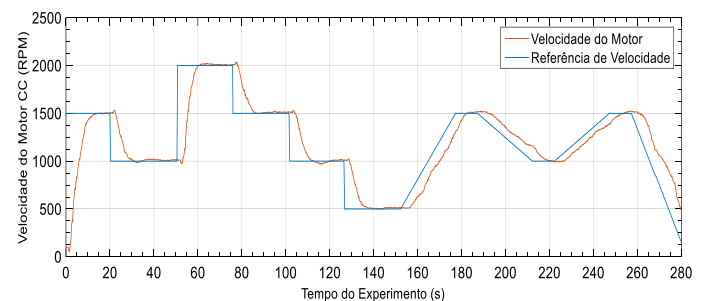


Fig. 6. Aplicação de Controle via *OpenPLC* usando Serviços da Fig. 3.

### B. Experimentos com Microsserviço CLP

Na segunda etapa de experimentos, a funcionalidade do microsserviço CLP é apresentada e avaliada. Nesses experimentos, foram utilizadas a planta didática de controle contínuo de velocidade de motor CC e uma planta virtual de automação discreta de separação de peças desenvolvida

utilizando o simulador *Factory I/O*. A estrutura geral desse experimento envolvendo a MOA é demonstrada na Fig. 7. Nesse experimento, toda comunicação entre o serviço CLP e a aplicação foi realizada via rede sem fio Wi-Fi.

A implementação do microserviço CLP utiliza um sistema embarcado composto por um Raspberry Pi 3B+ com uma placa de expansão de IO modelo UniPi 1.1, com o *software OpenPLC* rodando localmente no Raspbian Linux. Nesse experimento, a planta virtual no simulador *Factory I/O*, o serviço de comunicação (NATS) e a aplicação interna (Node-RED) rodam num computador.

Uma Aplicação Interna de “Monitoramento” foi desenvolvida para demonstrar graficamente os dados das entradas e saídas do microserviço CLP, bem como os sinais do controlador PID nativo do *OpenPLC*. Essa aplicação se comunica com os microserviços diretamente através do *Transporter* (NATS).

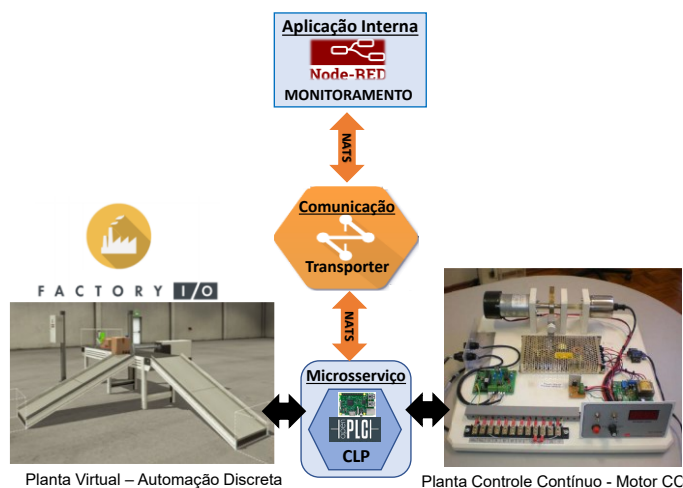


Fig. 7. Estrutura do Experimento de Automação e Controle de Processo usando Serviço CLP.

Na aplicação da Fig. 7, o *Factory I/O*, *software* que simula a planta industrial, se comunica pelo protocolo Modbus TCP diretamente com o *OpenPLC*, que é responsável por controlar o funcionamento da planta industrial. No caso da planta didática de motor, o *OpenPLC* aquisita as informações do processo (velocidade e comando do motor) através das entradas e saídas da placa UniPi. Em ambas as aplicações (Controle Contínuo da planta real e Controle discreto da planta virtual), a lógica de controle do CLP foi desenvolvida em linguagem Ladder por meio do *OpenPLC Editor*.

A aplicação de Monitoramento permite monitorar o sistema de automação da planta virtual e controlar a planta real, enviando parâmetros para o controlador PID e monitorando o comportamento da resposta da malha fechada. Na Fig. 8 pode ser observado a interface de monitoramento no Node-RED. Do lado esquerdo (Fig. 8) são apresentados os estados lógicos das entradas e saídas digitais do microserviço CLP, que controlam a planta virtual elaborada no *Factory I/O*. Na parte central está o controle de velocidade do motor em RPM. No lado superior direito estão os campos destinados ao envio dos parâmetros do

controlador PID (KP – Ganho Proporcional, TI – Tempo Integrativo e TD – Tempo Derivativo). Na parte inferior direita existem dois gráficos que mostram o comportamento da saída de tensão elétrica (V) aplicada ao motor (variável manipulada), e a velocidade do motor (variável de processo) em relação ao *setpoint* (RPM) respectivamente. O rastreamento da velocidade de referência estabelecida para o motor, como mostra a Fig. 8, mostra a efetividade do uso da arquitetura com microserviços para a automação e o controle do processo industriais.

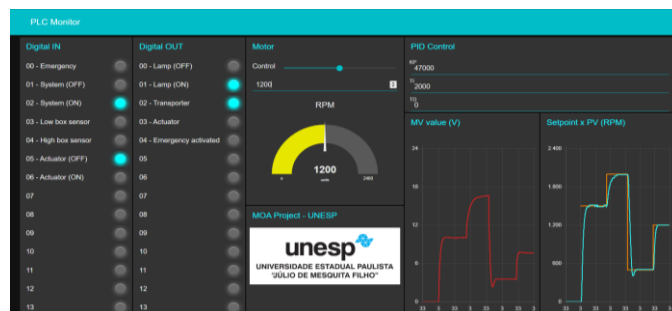


Fig. 8. Aplicação Interna de Monitoramento no Node-RED usando Serviço CLP da Fig. 7.

Finalmente, os resultados obtidos foram satisfatórios em ambos os casos de estudo apresentados (experimentos de orquestração de microserviços na Fig. 3 e experimentos com microserviço CLP de automação discreta e de controle contínuo de processo na Fig.7). Ambos os casos apresentaram confiabilidade e desempenho compatíveis com o requerido para uma aplicação tradicional de automação e controle, demonstrando a viabilidade de uso da arquitetura orientada a microserviços (MOA) no lugar de uma arquitetura tradicional de automação e controle. Adicionalmente, ambos os casos não apresentaram, durante os testes realizados, erros operacionais de hardware ou software (como por exemplo algum travamento de hardware executando os microserviços) ou falhas de comunicação em rede entre as aplicações e microserviços que levassem o sistema à uma condição de parada ou de instabilidade do controle executado.

A confiabilidade da comunicação em rede entre aplicações e microserviços foi usada como medida de confiabilidade da solução, considerando que esta comunicação é a base de operação da arquitetura orientada a microserviços desenvolvida. Essa confiabilidade é dada pela porcentagem de todas as mensagens transmitidas e recebidas corretamente durante a execução dos experimentos de automação como um serviço. Um valor médio de 96% foi obtido nos casos de estudo apresentados, sendo que eventuais problemas (4%) não impactaram nos experimentos realizados.

Os tempos de resposta das malhas de controle de ambos os casos de estudo são compatíveis com aplicações de automação e controle. No caso de estudo da aplicação de controle com orquestração de microserviços, o valor médio do tempo de orquestração foi de 105,5 ms e sua variabilidade (desvio padrão) foi de 43,9 ms. Para evitar sobreposição de requisições na malha de controle, um tempo de resposta mínimo de 300 ms foi definido. No caso da aplicação com o microserviço CLP, o

controle é executado localmente na planta (não há uso do microsserviço DAQ). O tempo de resposta da malha de controle está relacionado com o ciclo de execução do programa no OpenPLC, que por padrão é 50 ms. Considere, por exemplo, que os valores de especificação técnica para projetos de automação da Petrobras definem o tempo de resposta máximo de 500 ms para variáveis de processo do tipo vazão, pressão e pressão diferencial, e de 1000 ms para variáveis de temperatura, nível e analítica. Portanto, os tempos de resposta obtidos com as aplicações de automação como um serviço são menores ou cumprem os requisitos das aplicações tradicionais.

## V. DISCUSSÕES E CONSIDERAÇÕES FINAIS

A integração das tecnologias de informática e automação industrial em uma arquitetura orientada a microsserviços (MOA) permitiu a utilização e compartilhamento de microsserviços para a concepção de uma arquitetura industrial escalável, flexível, interoperável e distribuída em rede. A implementação da Automação Industrial como um Serviço apresentada nesta pesquisa contribui com um novo modelo de interações entre os diferentes sistemas industriais, equipamentos, usuários e aplicações, a qual vem de encontro com as mudanças e requisitos das aplicações de Indústria 4.0.

Apesar de haver desvantagens, as vantagens de adoção e uso das arquiteturas de microsserviços em aplicações de automação industrial são maiores e mais relevantes. O principal benefício dessas arquiteturas é a integração vertical entre camadas e a interoperabilidade de todo o sistema. Esse benefício é alcançado através da comunicação transparente e padronizada entre serviços e aplicações realizada pelo serviço de comunicação (*Transporter*). Embora o benefício dessa integração não tenha sido significativo nos experimentos deste artigo por conta da simplicidade, a facilidade de monitoramento ficou clara devido à interoperabilidade e acesso padronizado às informações dos serviços e aplicações.

A arquitetura de microsserviços é um paradigma diferente das arquiteturas monolíticas industriais tradicionais em termos de desenvolvimento, manutenção e comissionamento/implantação. As desvantagens ou dificuldades estão mais relacionadas com o aprendizado de novos conceitos, ferramentas e ambientes de desenvolvimento e de suas adoções na indústria do que com a complexidade do uso de microsserviços. O desenvolvimento dos microsserviços deve ser baseado em estruturas padronizadas (*containers*) e com suporte a múltiplas linguagens, como no caso do *framework Moleculer*. A manutenção da aplicação e o comissionamento dos microsserviços necessita de uma estrutura de repositório para melhor gerenciamento de versões e do estado dos microsserviços operacionais, o que não é comum em aplicações industriais. Um outro ponto a ser destacado é a complexidade de gerenciamento em grandes aplicações contendo muitos serviços. Devido à característica distribuída em rede dos microsserviços, a manutenção de falhas ou verificação de erros é mais complexa. Por outro lado, essas atividades podem ser feitas totalmente *online* e remotamente através da rede numa arquitetura de microsserviços.

O desenvolvimento de cada microsserviço em estruturas

individuais oferece facilidade na replicação e reuso dos serviços em outros tipos de aplicações. A redundância de serviços (por meio da replicação/cópia do mesmo serviço), também é uma outra importante característica da arquitetura de microsserviços para aplicações de controle e automação, ainda que não explorada nos experimentos realizados. Nesse caso, o serviço de comunicação do *Moleculer* verifica que existem duas ou mais instâncias (redundantes) do mesmo serviço, e quando houver a requisição desse serviço, ele automaticamente escolhe a melhor delas para a execução naquele momento.

## VI. CONCLUSÃO

O uso de arquiteturas orientadas a serviços em aplicações industriais relacionadas à Indústria 4.0 é uma realidade necessária para a promoção da interoperabilidade e interação vertical de dispositivos e sistemas, além da habilitação de uma arquitetura distribuída e modular.

O desenvolvimento dos microsserviços e aplicações de automação e controle com um serviço apresentadas nesta pesquisa foi baseado no uso do *framework Moleculer*. Essa é uma contribuição importante deste artigo, ao invés de criar toda a infraestrutura computacional de SOA como pode ser verificado em outros artigos na literatura, porque reduziu o tempo de desenvolvimento da arquitetura e permitiu o foco total no desenvolvimento dos serviços e aplicações direcionadas à automação e controle.

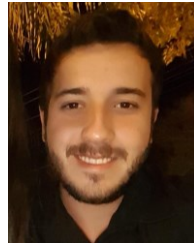
O diferencial deste artigo foi o desenvolvimento e uso dos microsserviços DAQ, CLP e Controle PIDPlus e de aplicações de Controle e Monitoramento via orquestração dos serviços. Considerando o microsserviço CLP, a possibilidade de se ter um controlador programável como um serviço em nuvem é indicada para aplicações em que não se exige determinismo e é vantajosa por não ser necessária a implantação de controladores físicos na planta. O microsserviço DAQ se assemelha à operação de uma remota de I/O em rede, com a vantagem de ser programável e poder incorporar funcionalidades customizadas pra cada aplicação. O microsserviço de Controle flexibiliza o desenvolvimento e aplicação de algoritmos de controle conforme a necessidade da aplicação. Nesse artigo, o uso do algoritmo de controle PIDPlus, comumente usado em controle via rede, se mostrou adequado para controlar as malhas de processo que se comunicam em rede através do microsserviço de comunicação (*Transporter*).

Diversos experimentos em cenários de automação e controle foram apresentados para comprovação da efetividade da aplicação e validação do desenvolvimento da automação industrial como um serviço. Os resultados apresentaram a viabilidade e eficiência da arquitetura de microsserviços usando o *framework Moleculer*, onde foi possível controlar todos os cenários testados. Adicionalmente às aplicações de controle, verificou-se que o desenvolvimento das aplicações de supervisão e monitoramento são simplificadas e facilitadas.

Na continuação dessa pesquisa os esforços estarão concentrados na implementação dos mecanismos de segurança descritos para a arquitetura e na análise do impacto desses mecanismos em seu desempenho operacional.

## REFERÊNCIAS

- [1] G. Arbib, M.S. Salerno, E. Zancul, G. Amaral And L.M. Lins. O Brasil e a Nova Onda de Manufatura Avançada: O que aprender com Alemanha, China e Estados Unidos. *Novos estudos CEBRAP*, 36(3), 29-49, 2017.
- [2] D. Jerker, "Local Cloud Internet," *IEEE Ind. Electron. Mag.*, no. DECEMBER, pp. 8–21, 2017.
- [3] J. A. Bigheti, M. M. Fernandes, and E. P. Godoy, "Control as a Service: A Microservice Approach to Industry 4.0," *2019 IEEE Int. Work. Metrol. Ind. 4.0 IoT, MetroInd 4.0 IoT 2019 - Proc.*, pp. 438–443, 2019.
- [4] Z. Xiao, I. Wijegunaratne, and X. Qiang, "Reflections on SOA and Microservices," *Proc. - 4th Int. Conf. Enterp. Syst. Adv. Enterp. Syst. ES 2016*, pp. 60–67, 2017.
- [5] A. Theorin, J. Hagsund, and C. Johnsson, "Service orchestration with OPC UA in a graphical control language," *19th IEEE Int. Conf. Emerg. Technol. Fact. Autom. ETFA 2014*, pp. 1–6, 2014.
- [6] O. Givehchi, J. Imtiaz, H. Trsek, and J. Jasperneite, "Control-as-a-service from the cloud: A case study for using virtualized PLCs," *IEEE Int. Work. Fact. Commun. Syst. - Proceedings, WFCS*, pp. 1–4, 2014.
- [7] E. C. Moraes, "Desenvolvimento de Interfaces Baseadas em Serviço para Integração de Sistemas Heterogêneos na Manufatura," p. 241, 2017.
- [8] F. Jammes *et al.*, "Promising technologies for SOA-based Industrial Automation Systems," *Ind. Cloud-Based Cyber-Physical Syst. IMC-AESOP Approach*, vol. 9783319056, pp. 89–109, 2014.
- [9] F. Blomstedt *et al.*, "The arrowhead approach for SOA application development and documentation," *IECON Proc. (Industrial Electron. Conf.)*, pp. 2631–2637, 2014.
- [10] P. Leitão, A. W. Colombo, and S. Karnouskos, "Industrial automation based on cyber-physical systems technologies: Prototype implementations and challenges," *Comput. Ind.*, vol. 81, pp. 11–25, 2016.
- [11] J. V. Espí-Beltrán, V. Gilart-Iglesias, and D. Ruiz-Fernández, "Enabling distributed manufacturing resources through SOA: The REST approach," *Robot. Comput. Integr. Manuf.*, vol. 46, no. November 2016, pp. 156–165, 2017.
- [12] O. Carlsson, J. Delsing, F. Arrigucci, A. W. Colombo, T. Bangemann, and P. Nappay, "Migration of industrial process control systems to service-oriented architectures," *Int. J. Comput. Integr. Manuf.*, vol. 31, no. 2, pp. 175–198, 2018.
- [13] D. Henneke, M. Elattar, and J. Jasperneite, "Communication patterns for Cyber-Physical Systems," *IEEE Int. Conf. Emerg. Technol. Fact. Autom. ETFA*, vol. 2015-October, pp. 8–11, 2015.
- [14] P. Di Francesco, P. Lago, and I. Malavolta, "Migrating Towards Microservice Architectures: An Industrial Survey," *Proc. - 2018 IEEE 15th Int. Conf. Softw. Archit. ICSA 2018*, pp. 29–38, 2018.
- [15] S. Sarkar, G. Vashi, and P. P. Abdulla, "Towards Transforming an Industrial Automation System from Monolithic to Microservices," *IEEE Int. Conf. Emerg. Technol. Fact. Autom. ETFA*, vol. 2018-Sept, pp. 1256–1259, 2018.
- [16] M. Ciavotta, M. Alge, S. Menato, D. Rovere, and P. Pedrazzoli, "A Microservice-based Middleware for the Digital Factory," *Procedia Manuf.*, vol. 11, no. June, pp. 931–938, 2017.
- [17] J. Innerbichler, S. Gonul, V. Damjanovic-Behrendt, and B. Mandler, "NIMBLE Collaborative Platform: Microservice Architectural Approach to Federated IoT Johannes," vol. 53, no. 9, pp. 1689–1699, 2019.
- [18] J. Song, A. K. Mok, D. Chen, M. Nixon, T. Blevins, and W. Wojsznis, "Improving PID control with unreliable communications," *Isa Expo 2006*, vol. 2006, no. October 2006, pp. 105–116, 2006.
- [19] "The OpenPLC Project." [Online]. Available: <https://www.openplcproject.com/>. [Accessed: 24-Aug-2020].
- [20] "Moleculer - Progressive microservices framework for Node.js." [Online]. Available: <https://moleculer.services/>. [Accessed: 23-Aug-2020].
- [21] "Bigheti, J. Arquitetura de automação e controle orientada a microsserviços para a Indústria 4.0. Tese de Doutorado. Engenharia Elétrica. Universidade Estadual Paulista. Available: <https://repositorio.unesp.br/handle/11449/191993>. [Accessed: 28-Jan-2021].
- [22] "SOCRADES Project". [Online]. Available: <http://www.socrades.net>. [Accessed: 28-Jan-2021].
- [23] "IMC-AESOP Project", [Online]. Available: <https://www.imc-aesop.org/>. [Accessed: 28-Jan-2021].
- [24] "ARROWHEAD Framework", [Online]. Available: <https://www.arrowhead.eu/>. [Accessed: 28-Jan-2021].
- [25] "PRODUCTIVE 4.0 Project". [Online]. Available: <https://productive40.eu/>. [Accessed: 28-Jan-2021].



**Michel de Mattos Fernandes** received the B.Eng. degree in Electrical Engineering from the University of Marília - UNIMAR (SP) (2015) and the M.S. in Electrical Engineering at São Paulo State University (Unesp) - Bauru (SP). He is currently an Industrial Electrician Designer with research interests in automation and supervision of industrial systems, Internet

of Things, embedded electronics and Industry 4.0. <http://lattes.cnpq.br/8481523580238627>.



**Jeferson André Bigheti** received the B.Tech. degree in Electrical Systems in 1990, the Academic specialization in Mechatronic Engineering in 2007 and the M.Sc. and Ph.D. in Electrical Engineering in 2011 and 2020, all from the São Paulo State University (Unesp) at Bauru (SP-Brazil). Currently he is a professor of the National Service for Industrial Learning

(SENAI) at Lençóis Paulista (SP-Brazil). His research interests include automation, supervision of industrial systems, Internet of Things and Industry 4.0. <http://lattes.cnpq.br/6243932250481498>.



**Ricardo Pasquati Pontarolli** received the B.Tech. degree in Industrial Automation from Technology College, São Paulo State, Brazil, in 2011. He is currently studying M.S. in Electrical Engineering at São Paulo State University (Unesp) and laboratory technician in industrial automation at the Federal Institute of São Paulo (IFSP) in Boituva (SP-Brazil). His

research interests include industrial automation, supervision of industrial systems, Internet of things, embedded electronics, Industry 4.0, and Security mechanisms. <http://lattes.cnpq.br/6917700616010402>.



**Eduardo Paciencia 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>.