Application and Requirements of AIoT-Enabled Industrial Control Units

Everton Hideo Nishimura(✉), Yuzo Iano, Gabriel Gomes de Oliveira, and Gabriel Caumo Vaz

School of Electrical and Computer Engineering, State University of Campinas, Campinas, Brazil
everton.nishimura@gmail.com, yuzo@unicamp.br, oliveiragomesgabriel@ieee.org

Abstract. Industrial Automation control systems are highly demanding devices regarding real-time control. They excel at this task and have been perfected since their introduction till today. Thanks to their own domain problems, they do not always receive innovations directly from their general-purpose computation counterparts. This paper will focus on the new generation of industrial automation controllers, which features advanced IoT capabilities, virtualization, and open-source software integration while creating the possibility to integrate machine learning models into the automation process. The AIoT-enabled industrial controllers will help close the borders between general programming and its techniques and frameworks, to the world of Automation, changing what is state of the art today.

Keywords: PLC · IoT · AIoT · Industry 4.0 · Industrial control · Machine learning · Data science

1 Introduction

While IoT (Internet of Things) grows very fast and is taking place in our everyday life, the Industrial Internet of Things has more constraints. Industrial control technology was developed towards reliability and real-time control, and the gains with IoT are not always worth the risk involved in networking unprepared devices. The Industry 4.0 movement started in Germany brought attention to this topic and highlighted the advantages of connected devices.

Next-generation control units [1] enabled for Industry 4.0 promise new possibilities such as connecting databases and AI (artificial intelligence) in servers with real-world actuators, bringing predictions on machine availability, smart scheduling of tasks, automated asset managing on the shop floor, and creation of virtual sensors. Manufacturing processes from all sorts of industries can benefit from AIoT-enabled (AI and IoT) control units, and this will probably be the difference between low and high productivity in the future [2].

Y. Iano et al. (Eds.): BTSym 2021, SIST 295, pp. 724–733, 2022.
https://doi.org/10.1007/978-3-031-08545-1_72

Even though there is almost no difference between the architecture of a modern PC-based PLC and a general-purpose computer, modern applications that fully utilize the potential of connected controls are difficult to create on legacy control software. Nonetheless, general-purpose computer operational software can offer a welcoming platform for AI, IoT, and industrial automation software.

This paper will focus on the way AIoT-enabled industrial controllers can meet the demands for Industry 4.0 using general-purpose operational systems as a basis while maintaining legacy programming as the main machine control. An example application will be presented as a simplified flow for deploying a machine classification model on a process machine.

2 Background

Industrial control technology has evolved to better compact systems, but always keeping distance from general-purpose computing, even when sharing most hardware components. This situation changed with the introduction of the German Industry 4.0 strategy [3]. Among the goals of the initiative, there was a bigger scope of integration between machines, transparency of information for all process stakeholders, artificial aided technical assistance, and autonomy powered by AI on the new smart systems.

While the initiative was successful in setting the goal of all modern development of industrial hardware and software, many efforts are required by the manufacturers of industrial control electronics, system integrators, and OEM manufacturers to create systems that can be considered connected.

2.1 Industrial Internet of Things

Most PLC today are not prepared to be connected to the internet, demanding the use of extra components in the industrial systems they will be used as gateways to create applications with the minimum risk of cyberattacks. IEC 62443 is proposed as a guide for all parts involved in the development of connected industrial systems [4]. Cybersecurity threats are not considered in most modern systems, however, this can be changed quickly with the advent of newer technologies, such as 5G. High-speed network capabilities on all devices, with minor hardware investments, can be a game-changer on the possibilities on IoT systems, and this trend will certainly reach industrial IoT systems such as PLCs.

The cyber-physical systems, as stated by modern definitions, need to be networked to create intelligent systems, promoting response to unforeseen events and evolution. The requirements to achieve such features are connectivity and data processing capabilities. Standard PLC technologies, as stated before, are hardly used as data storage or handler, normally delegating such operations to a general-purpose system consisting of a traditional computing server.

2.2 Industrial Machine Learning Development

Programming on traditional PLC has evolved to a very concise standard called IEC 61131-3, which defines several languages that focus on creating very easy-to-understand software for general automation. But, as stated before, the rising need of integrating AIoT on control units to leverage auto diagnosis and autonomy made it very impractical to continue working within industrial automation programming languages defined by IEC 61131-3. This can be explained by the fact that most AI and Data Science libraries are created in languages such as Python and R.

Even though it is clear that new capabilities should be introduced to industrial software, the behavior of the users of industrial systems tends to be geared towards being very conservative. This means that the complete disposal of what has been used or programmed in the last 20 years is inconceivable in most situations. This is a problem when there are so many requirements for new systems, but still the necessity for retro compatibility.

3 Methodology

The following subsections will present a discussion on the requirements for an Industry 4.0-enabled automation solution and how embedded Linux as a basis for PLC can be used for achieving them. Additionally, an application is proposed using a machine learning classification model on a device while maintaining legacy programming for machine operation. Within the scope of this work, this application will be run on real control hardware, but data consumed by the machine learning model will be acquired from a publicly available dataset from the Case Western Reserve University bearing data center.

The integration of general computing operational systems is an efficient solution to beat the challenge of AI and IoT. Within the scope of this work, we will focus on the advantages of using Linux-based devices, as they offer a highly stable environment and a generally suitable basis for small networked devices.

3.1 Small Size

Linux has been used for distributed intelligent devices not only for economic reasons. The lower processing and memory footprint fits well small devices. It is reasonable to think that with the promised omnipresence of 5G connectivity, much more devices will incorporate processing power to harness data from such devices. It is also possible to highlight the more flexible licensing and open-source capabilities of Linux systems.

3.2 Cybersecurity

IEC 62443 can serve as a guideline for reducing cyber threat risks on industrial automation systems [4]. This IEC uses the “defense in depth” approach, meaning that all components of the security chain should take as many countermeasures as possible. We are talking about technology provided by the supplier of automation components, as well as training for users so the security can be guaranteed.

Internet servers are notoriously powered by Linux systems. Therefore the technology of such servers evolved to keep cybersecurity threats at bay. Taking advantage of this notoriously advanced technology is a strategy used by many suppliers of automation systems. The Linux OS can take the role of a Gateway, separating the corporate network from the industrial network, or can be embedded in the protected device itself.

3.3 Connectivity

Communication between devices is not uncommon on automation systems. SCADA and distributed control systems both utilize socket communication to exchange information packages. These communications are usually structured on automation protocols like Modbus, and preferably OPC UA in novel systems. OPC UA is an improvement over OPC DA and promotes services and less coupling between server and master than previous protocols. On the IoT side, MQTT and other M2M should be considered.

Most communication protocols can be easily implemented on a Linux application. No special toolchain is needed, and most protocols have open-source distributions that can speed up the process of implementation. Visual tools like Node-RED can even help build IoT systems with the low code approach.

3.4 Data Storage

Most commitments to the return of investments on newer technologies come from harnessing the power of data science. To achieve such features, not only data processing capabilities are necessary, but also data storage. Most PLCs have limited data storing capabilities. Storage on devices in PLC software is generally done using files such as CSV or TXT, and this limitation is most likely linked with the lack of software to implement databases, rather than limitations on memory.

On the other hand, it is possible to manage and install databases on a Linux system with ease. While it is not advisable to have a permanent database on a field device, decentralized systems can benefit from local databases to gain autonomy on decisions and storage redundancy.

3.5 Data Science

The problem with data science comes in two parts: creating a model and running it. As we currently stand, modern PLC software cannot cope with any of these. Modern systems with integrated AI make use of a distributed architecture where all "data science" is managed on a server, and all the automation is done on the device. They communicate with each other using manufacturer-specific API or

OPC UA. Such server should be capable of handling the tensor operations, most likely using a GPU, depending on the requirements.

Running and creating a model on a small embedded Linux device is possible with some limitations. The model should preferably run on CPU, since industrial hardware tends to have lower power and size, making a standard computer GPU very difficult to incorporate on a field device. While AI accelerators like Google Coral exist, one solution should create the model on a powerful computer, and just run the model on a field device.

3.6 Modularity of Software Development

The constant need for onboarding new applications as well as keeping the legacy ones available can be related to the so-called "DevOps" discipline in IT systems. Modularization of applications is a trend and can be done with the help of software containers such as Docker, LXC, or Snap. They help create isolation environments that do not interfere with each other. To be able to manage containers is potentially a way to onboard legacy software on newer architectures [5].

Traditional PLC architecture is based on monolete design, which can be hard to maintain and expand. Meanwhile, Linux IAC can be potentially expanded by interfering with other applications using containerization techniques.

3.7 Open-Source

Linux is based on open-source, and it can obviously run much open-source software. Depending on the licensing for this open-source software, one big advantage is obtaining new functionalities without spending money. The return of investment with data science systems can be uncertain, and be determined years later acquiring significant data, and mitigating the starting costs with licensing, can be a kick-off. As manufacturers tend to be really protective with their software, besides some open-source libraries to work with IEC 61131, no efforts to support OSS are really made. Linux PLCs on the other hand can work with a whole scope of apps, languages, and frameworks.

3.8 Virtualization

Digital twins are representations of real-world processes and systems and can be used to simulate behaviors and analyses based on pre-existent data sets. A common approach to virtualization is creating a virtual instance of the targeted system.

Virtualization can be considered one step further modularization: this is where we create an OS instance on simulated hardware. By doing this, most programming features of such a system can be simulated, therefore combined with more simulation tools, the complete system can be virtualized to a good extent.

4 Example Application

Given all requirements discussed in the last section, this work will propose a concept application that can handle present and probably future needs for AIoT. We will focus on the development of a modern example that can integrate legacy automation needs such as Fieldbus and IEC 611131-3 programming while still benefiting from novel tools as a Machine Learning direct on the Edge and safe internet networking.

4.1 Platform Selection

As a starting point, we will benefit from having a commercial industrial platform as a basis for this application. A modern software engineering architecture such as software containers are alternatives to tackle the complexity of ever-evolving requirements on IoT solutions, creating a path for future upgrades in the system.

Many third-party open source and commercial apps are needed for the building of our solution, so containers are needed to guarantee that apps are running independently, having fewer dependencies as possible (Fig. 1).

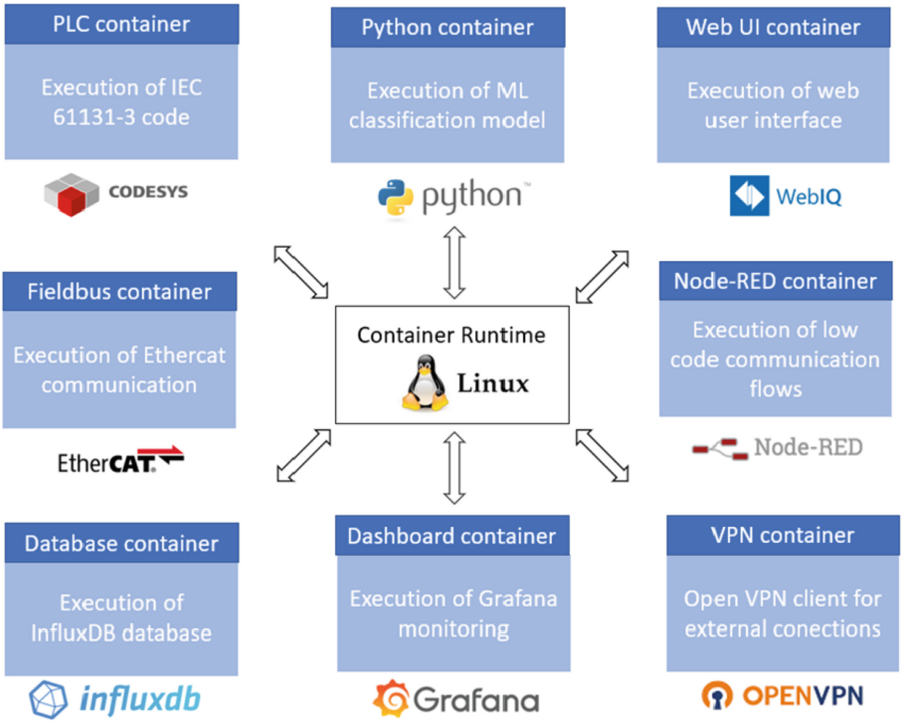
PLC container
Execution of IEC 61131-3 code
CODESYS
Python container
Execution of ML classification model
python
Web UI container
Execution of web user interface
WebIQ
Fieldbus container
Execution of Ethercat communication
EtherCAT
Container Runtime
Linux
Node-RED container
Execution of low code communication flows
Node-RED
Database container
Execution of InfluxDB database
influxdb
Dashboard container
Execution of Grafana monitoring
Grafana
VPN container
Open VPN client for external conections
OPENVPN

Fig. 1. Container structure on an industrial Linux host.

Industrial automation suppliers are aware of the complexity and extension of these requirements and among them, we selected Bosch Rexroth ctrlX Automation for the following features:

- Ready to use, modern embedded Linux onboard, with kernel modified to run real-time tasks;
- System architecture follows the CAS principle, using Snap containers to create future-ready modular solutions;
- Legacy IEC 61131-3 also runs in a container, granting retro compatibility;
- System-wide data transfer interface available, also in the form of a rest API;
- Python container available, shipped with integrated web IDE based on VS Code.

4.2 Application Description

Our example solution will run a classification model on the control unit. The concept is presented in the following figure where the control unit gets data from the field, and store it in a database. This data can be sent over to the Development station, where the developer can comfortably develop and evaluate a classifier model using common ML tools like scikit-learn. This model is later sent to the control unit where it can be used classification data from the field. Figure 2 shows a simplified flow intended.

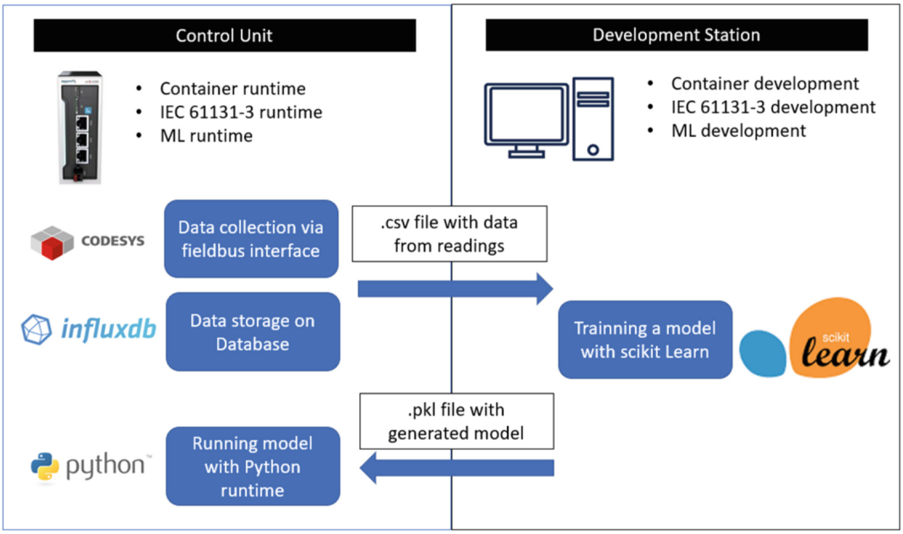

Fig. 2. Data science model creation and deployment.

Due to restrictions, we choose to create a model based on the Case Western Reserve University bearing data center [6]. This database is available and provides data from a real test station, with two accelerometers measuring vibrations from the AC motor near the spindle and on the opposite side. The goal here is to detect defects on different sets of bearings. This is a useful dataset because a similar model could be used to create an ML-enabled condition monitoring solution directly on the device controlling the motor itself.

Figure 3 shows the flow of information needed to create the proposed solution. The data exchange can be easily implemented using the ctrlX system-wide proprietary communication. The Data read from any fieldbus device such as an Ethercat slave can be sent to the python container, and classified as "normal operation" or "fault". Results can be archived on the database for quality control or maintenance. Given our topology, a container running Grafana can be used to show the results for machine operators and create alerts for maintenance. A container with OpenVPN can bring a safe internet connection for remote maintenance tasks.

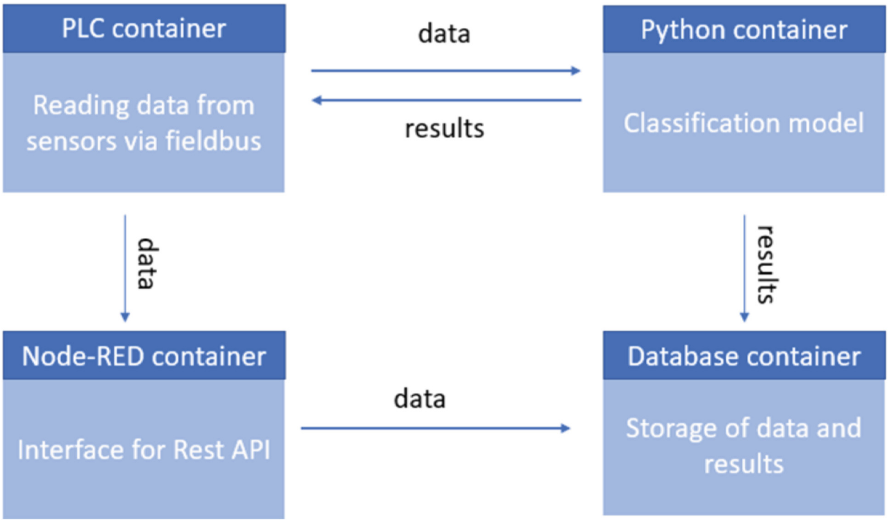

Fig. 3. Data flow between containers.

5 Analysis of the Proposed Application

Using the data from the bearing data center as an example, we choose to use a simple Random Forest algorithm to generate a classifier using scikit-learn [7]. The training and evaluation could be easily done on our development PC, reaching 85% accuracy, 89% precision, and 81% recall. The generated model could be converted and easily transferred to our control unit, using the integrated web IDE from our control unit. Using the integrated web IDE, we could create and test a python code that could access data from the PLC container in a cyclic behavior, needed for this kind of application.

The application can also be accessed remotely using a VPN server, which is fundamental for remote maintenance, both for evaluating the performance of the model on real-life data and the machine operation itself.

6 Conclusion

The state of the art in industrial automation solutions is leaping towards AIoT-enabled devices. In a similar manner that happened to our everyday devices, connectivity and artificial intelligence will be an integrated and integral part of any compatible solution. This trend will be consolidated in the next decade, but we already have tools to create effective future-proof automation solutions with Industry 4.0 requirements fulfilled. Legacy Industrial software existing today is a valuable asset for any machine builder, and it is possible to integrate it with newer developments in AIoT without much effort.

Our results in the application showcased were obtained from an already existing data set. Despite this, our proposal showed that embedding ML models on control units is possible and can be easily done on embedded Linux, and integrated with legacy machine control coding. This opens doors to integrate ML in real-world processes, with next to no delay on results since they are done on the same processing unit. This kind of application is possible because implementing python coding, databases, and monitoring apps and integrating them with IEC61131-3 software can be done on an AIoT ready control unit. While not mandatory, software containers can help us bring all this software together without worrying about dependencies and future upgrades.

Machine manufacturers interested in delivering Industry 4.0 solutions to their customers, should already be looking after the advantages of such systems in the fields of condition and process monitoring. In the past, creating such solutions would require extra hardware and make it not viable for economic reasons. With integrated machine learning on a device, no extra hardware or connectivity is necessary, making the deployment cost very low. The competitive advantage will most likely drive the machine-building markets to an interesting landscape where the return of investment for an AIoT enabled machine will be reduced by intelligent algorithms, remote maintenance capabilities, and general quality of life improvement brought by connectivity.

Within the next steps of this research, there are many topics to expand upon and improve. Studying solutions that can produce satisfactory results even without extra sensors is a challenge but can be used to shorten the RoI for low-cost machines. More computational heavy algorithms like deep learning may be necessary to process more complex data or improve results on classic ML algorithms. A practical workflow for ML deployment on AIoT enabled PLCs is also needed and can be created to shorten development times.

References

1. De Blasi, S., Engels, E.: Next generation control units simplifying industrial machine learning. In: 2020 IEEE 29th International Symposium on Industrial Electronics (ISIE), pp. 468–473 (2020). https://doi.org/10.1109/ISIE45063.2020.9152527
2. The AIoT Framework: AIoT Playbook v0.9.7 (2021). https://www.aiotplaybook.org. Accessed 1 Oct 2021
3. Vaidya, S., Ambad, P., Bhosle, S.: Industry 4.0 - a glimpse. Proc. Manuf. **20**, 233–238 (2018). https://doi.org/10.1016/j.promfg.2018.02.034
4. Maidl, M., Kröselberg, D., Christ, J., Beckers, K.: A comprehensive framework for security in engineering projects - based on IEC 62443. In: 2018 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), pp. 42–47 (2018). https://doi.org/10.1109/ISSREW.2018.00-33
5. Goldschmidt, T., Hauck-Stattelmann, S.: Software containers for industrial control. In: 2016 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), pp. 258–265 (2016). https://doi.org/10.1109/SEAA.2016.23
6. Case Western Reserve University: Bearing Data Center—Case School of Engineering—Case Western Reserve University (2021). https://engineering.case.edu/bearingdatacenter. Accessed 1 Oct 2021
7. Kamat, P., et al.: Bearing fault detection using comparative analysis of random forest, ANN, and autoencoder methods. In: Sharma, H., Gupta, M.K., Tomar, G.S., Lipo, W. (eds.) Communication and Intelligent Systems. LNNS, vol. 204, pp. 157–171. Springer, Singapore (2021). https://doi.org/10.1007/978-981-16-1089-9_14