Fogification of Industrial Robotic Systems: Research Challenges

Shaik Mohammed Salman
shaik.salman@se.abb.com
ABB AB, Västerås, Sweden
Mälardalen University, Västerås, Sweden

Vaclav Struhar
vaclav.struhar@mdh.se
Mälardalen University, Västerås

Alessandro V. Papadopoulos
alessandro.papadopoulos@mdh.se
Mälardalen University, Västerås

Moris Behnam
moris.behnam@mdh.se
Mälardalen University, Västerås

Thomas Nolte
thomas.nolte@mdh.se
Mälardalen University, Västerås

ABSTRACT

To meet the demands of future automation systems, the architecture of traditional control systems such as the industrial robotic systems needs to evolve and new architectural paradigms need to be investigated. While cloud-based platforms provide services such as computational resources on demand, they do not address the requirements of real-time performance expected by control applications. Fog computing is a promising new architectural paradigm that complements the cloud-based platform by addressing its limitations. In this paper, we analyse the existing robot system architecture and propose a fog-based solution for industrial robotic systems that addresses the needs of future automation systems. We also propose the use of Time-Sensitive Networking (TSN) services for real-time communication and OPC-UA for information modelling within this architecture. Additionally, we discuss the main research challenges associated with the proposed architecture.

CCS CONCEPTS

• **Computer systems organization → Cloud computing**; **Robotics**; **Robotic components**.

ACM Reference Format:
Shaik Mohammed Salman, Vaclav Struhar, Alessandro V. Papadopoulos, Moris Behnam, and Thomas Nolte. 2019. Fogification of Industrial Robotic Systems: Research Challenges. In *Workshop on Fog Computing and the IoT (IoT-Fog '19), April 15–18, 2019, Montreal, QC, Canada.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3313150.3313225

1 INTRODUCTION

Industrial robots have become an integral part of the industrial automation environment with traditional application areas such as spot welding, spray painting and machining [7]. Currently, each robot comes with a dedicated controller that provides motion control, programming interfaces and physical interfaces for integrating field devices such as sensors and actuators via industrial networks [4]. The controller is designed to meet the real-time constraints demanded by motion control algorithms as well as real-time requirements of industrial networks. These controllers, however, have fairly limited computational resources restricting the integration of complex functionality such as image processing, multi-robot motion control and other complex applications [24]. Although multi-robot motion control within a single controller is possible with solutions such as ABBs multimove functionality, the number of robots that can be controlled is still limited. Additionally, flexible production requirements of future automation systems impose demands such as firmware updates and hardware maintenance without any production downtime [6]. Meeting such requirements within the existing architecture is non trivial. Supporting the required infrastructure for augmented reality based immersive human-machine interaction concepts as shown by Paelke et al. [17] and Guhl et al. [8] will also require significant computational capacity and communication bandwidth. While increasing hardware capabilities within the controller can be a presented as a solution, this only addresses some of the concerns, validating the need to investigate cloud and fog-based architectures.

While cloud computing offers significant computational resources on demand, it does not guarantee real-time performance as required by traditional control applications [3, 5, 9]. Fog computing [26], is a new paradigm that allows utilization of computational resources near the edge of the network close to the source of the data. It introduces an intermediate layer between the cloud and the end devices that consists of a number of devices, called fog nodes, that are interconnected to form a network and these devices offer their computational resources (e.g., CPU, storage), for use by applications within this network. While the well established cloud computing paradigm provides services ranging from collection of historical data to big data analysis, fog computing complements the cloud functionality by providing local data processing. This capability, along with real-time communication mechanisms such as TSN [20], enables the fog-based architecture to provide predictable communication times.

Authors of [11, 15] have discussed fog-based solutions for general robotic systems and highlighted the advantages of using fog-based architecture for such applications. While Hao et al. [10] provided a generic software architecture for fog computing, Faragardi et al. [3] provided a time predictable framework for a smart factory integrating the fog and cloud layers. Skarin et al. [22] developed a test bed to study the feasibility of a fog-based approach for control applications, while Pallasch et al. [18] and Mubeen et al. [16] showed

IoT-Fog '19, April 15–18, 2019, Montreal, QC, Canada

ACM ISBN 978-1-4503-6698-4/19/04…$15.00
https://doi.org/10.1145/3313150.3313225

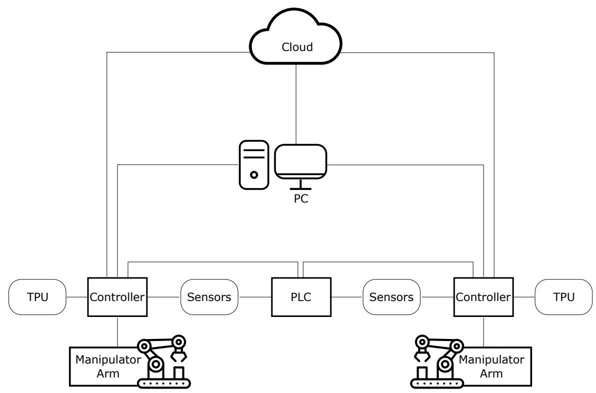

Figure 1: Existing Industrial Robotic System Architecture.

the feasibility of using an edge based solution for combining cloud and field devices. Vick et al. [1] presented the concept of "Using OPC-UA for Distributed Industrial Robot Control".

Based on the evidences provided by the above studies, in this paper, we propose the "Fogification" of industrial robotic system architecture to enhance the capabilities of industrial robotic systems by taking advantage of the fog computing paradigm. We use the term "Fogification" to define the integration of fog computing platform within industrial systems such that they benefit from the low-latency, distributed fog based resources within the local networks as well as from the high performance cloud computing environment. In order to highlight the advantages of using the fog-based architecture for industrial robotic systems, we analyse the existing architecture and identify its main limitations. Based on these limitations, we introduce a fog-based architecture for industrial robots and we identify the main research challenges associated with the proposed architecture.

2 EXISTING SYSTEM ARCHITECTURE

A single industrial robotic system typically consists of a mechanical unit called the manipulator, a controller, and a graphical controller interface device (see Fig. 1). The controller is the main processing unit that executes the control algorithms for robot motion. It also provides mechanisms to program the robot motion and to configure any additional behaviour required within the robot environment. The controller also provides multiple communication interfaces for interaction with fieldbus networks and the enterprise network including cloud based services.

In a multi-robot system, a number of robots are programmed together to accomplish a process application such as welding and painting. Here, the individual controllers are connected to each other to form a local network. To ensure synchronisation between different controllers, a fieldbus network and a PLC is utilised.

2.1 System Components

In this section, we give a brief overview of the different components of the current robotic system as shown in Fig. 1.

2.1.1 Manipulator. The manipulator is the mechanical arm with varying degrees of freedom.

2.1.2 Controller. The controller is responsible for controlling the manipulator motion and providing required interfaces for interaction with the robot environment such as other robot controllers, devices such as PLCs, conveyors and other sensors and actuators.

2.1.3 Teach Pendant Unit (TPU). The TPU acts as the human machine interface device for the controller. It is physically connected to the robot controller and can be used to manually move the manipulator, to configure different parameters of the system and to visualise the current state of the system via the device display.

2.1.4 Programmable Logic Controllers (PLC). The PLCs are used for controlling the synchronisation and coordination between different devices within the robot environment.

2.1.5 Sensing Devices. Sensing devices are used to provide information about the robot environment. These are connected to the controller via industrial networks or PLCs.

2.1.6 PC Software. The PC software provides visualisation, simulation, configuration and editing tools for robot programming and monitoring.

2.1.7 Cloud. The controllers are capable of connecting to the cloud services where data from the controller is stored and used for analysis. Currently, the cloud services are mainly utilised for collecting system data and not for control.

2.2 Classification of Controller Functions

The functionality of the existing controller firmware can be classified under three main categories, i.e, control, configuration and communication.

2.2.1 Control. The control functionality is responsible for path planning, trajectory generation and low level control [14] and it requires real-time capabilities from the system. Currently, the controller system provides real-time guarantees via a real-time operating system.

2.2.2 Configuration. The configuration functionality allows the users to configure the system behaviour such as defining the maximum speed of the robots and providing information about the robot environment in terms of available sensors and the connected networks. The configuration functionality does not require real-time guarantees and is usually carried out offline while the system configuration is updated when the manipulators are not in motion.

2.2.3 Communication. The communication functionality refers to user interaction features such as the robot programming language, the interface for the teach pendant unit, communication with different field devices via fieldbus networks and connectivity to enterprise networks and the cloud services. These communication functions impose both real-time requirements as well as non real-time requirements on the system. Connectivity to fieldbus interfaces, for example, requires real-time guarantees while the connectivity to enterprise networks can be non real-time.

2.3 Limitations

The existing architecture relies heavily on the controller component to achieve the functional behaviour of the system. In addition to limited resources on the controller, the existing architecture has certain limitations in terms of supporting flexible production requirements of future automation systems [6]. Some of the key limitations are discussed below.

L1: *Computational Resources* - The available computational resources within the existing controllers are not sufficient for the implementation of complex functionality. For example, computationally demanding tasks such as image processing when using vision based sensors cannot be carried out within the controller due to the limited resources.

L2: *Fleet Management* - In the existing setup, introducing new functionality or improving performance of the existing system via over the air software updates is limited since doing so requires production downtime. Also, controllers have limited connectivity to networks outside the factory environment, restricting the robot vendors from updating the controller firmware.

L3: *Environment Interaction* - Currently, the data from advanced sensors such as cameras is not directly shared with all the controllers which are part of the same environment. Normally, another computer is necessary to do the pre-processing. This can introduce latencies and affect the ability to have comprehensive information for better path planning and control.

L4: *Hardware Dependency* - The controller firmware is designed to make optimal use of the available controller hardware. Replacing the hardware with different hardware without significant changes to the controller firmware is non-trivial.

L5: *Connectivity* - In order to support communication with various industrial networks along with enterprise and cloud connectivity, the controllers need to provide multiple communication interfaces. Maintaining multiple interfaces increases the total system cost. Additionally, the current solutions do not allow for a seamless communication of the mobile platform, and they are typically based on wired, rather than wireless connection.

3 FOG BASED SYSTEM ARCHITECTURE

In the fog-based system architecture, we propose the utilisation of the fog layer as the computational platform and TSN [20] as the communication mechanism for both fog-to-fog communication as well as fog-to-field communication. To support a standard mechanism for information exchange, we propose the use of OPC-UA data modelling standards [1]. We describe the key components of the architecture in Fig. 2.

3.1 Computational Platform

In the fog-based system architecture, the controller functionality is provided by a fog-based computational platform. This platform will provide the required computing capacity to execute the system functions. Fig. 2 shows the fog-based system architecture. Here, the controller component is replaced by the robot drive system and the entire controller functionality is moved to the fog layer. Replacing the individual controllers with a fog-based platform provides multiple advantages, such as compute capacity on demand and additional storage. This addresses limitation L1. The robot application manager, as shown in Fig. 2, acts as an orchestrator that distributes the controller functionality within the fog layer and manages the firmware updates by interacting with the cloud layer. It stores the updates in one of the fog nodes and applies these changes to the system when it is idle, addressing the limitation L2.

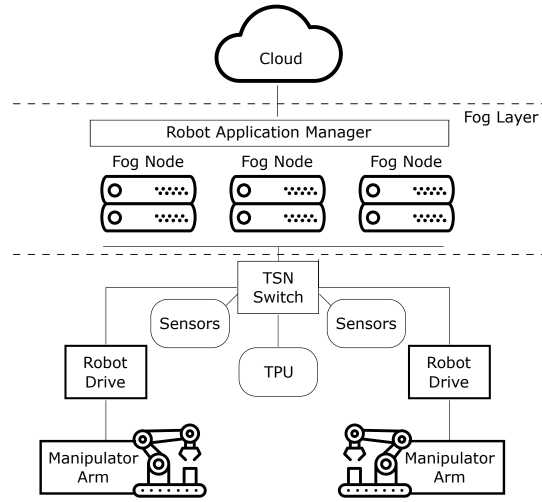

Figure 2: Proposed fog based architecture for Industrial Robotic System.

3.2 Communication Interfaces

System components such as the manipulator and the TPU, which are normally physically connected to individual controllers, will form a local network with the fog platform via TSN [20]. Also, field level sensors that are normally interfaced via fieldbus networks and connected to individual controllers will now be connected via the TSN network to the fog platform. It can also be possible to integrate the fieldbus network to the fog platform via gateways. The information from such sensors can now be shared by different applications running on the platform for improved system performance, addressing the limitation L3. Using OPC-UA standards for data modelling will provide a standard mechanism for information exchange, replacing the multiple communication protocols currently used, addressing a part of the limitation L5.

3.3 Software Deployment

To execute the controller functions in the fog layer, the application needs to be designed such that it is hardware agnostic, addressing the limitation L4. In the new architecture, we propose the use of a service oriented approach to meet the functional requirements [1]. For example, the robot program interpreter can be deployed as an independent service that can execute on one of the fog nodes. The trajectory generator, waiting for the input from the robot program interpreter, executes on another node within the network, while the low level control component is executing on yet another node. The real-time communication interface, responsible for data exchange with the sensors and actuators, can run as a background service, which can be subscribed to by other services for data manipulation and control activities. An important assumption we make here is that the real-time requirements of all the above components can be met.

4 RESEARCH CHALLENGES

Introduction of fog computing in the proposed industrial system architecture brings a myriad of research challenges that needs to be addressed. The need to meet the extra-functional properties of the system, such as safety, security [21], availability and reliability [20], makes the deployment of fog computing complex and needs many considerations.

In this section, we identify some of the research challenges that must be addressed in order to build the proposed industrial robot system.

4.1 Orchestration and Inner Fog Architecture

The fog layer is a distributed environment consisting of many heterogeneous fog nodes offering their computational capacity. For an efficient use of the resources, a fog orchestrator needs to be defined. Fog orchestration [25] is a key component that partitions the workload between fog nodes, keeps track of available resources, cooperates with the cloud and manages unexpected situations that occur in the fog layer. Choosing the right orchestration techniques is crucial as it affects the behavior of the entire system. Here, the challenge is to find an appropriate architecture, in a holistic way, that will meet all the functional/extra-functional requirements of the presented industrial system.

Skarlat et al. [23] proposed a hierarchical approach where fog orchestrator manages a fog colony (logical unit of fog devices) and divides work among this fog colony, neighbouring colonies or cloud. Whereas in [12], the fog orchestrator works only as a workload balancer that couples a fog node with a device requiring computational resources. Nodal collaboration [13] defines how fog nodes communicate to each other. It provides two basic models: peer-to-peer, where each node can directly communicate to each other, and the client-server model, where there is a hierarchy of servers providing services and client nodes consuming them.

4.2 Real-Time Guarantees

The proposed system must meet strict timing constraints and these constraints are valid for both the computation time and data transmission time in the network. Therefore, we need to find appropriate scheduling and analysis mechanisms for the conjunction of both the computational and the transmission part. The data transmission time can be bounded by use of TSN. Pop et al. [20] proposes the use of TSN in fog Computing, whereas in the paper [19], optimization strategies for TSN are shown.

Additionally, the problem of variable timing constraints during the operational process needs to be tackled. For example, the robots may require control instructions at changeable rates, i.e., complex small-grained movements or movements at high speed demand instructions at high rates, while slow speed motions need lower pace of instructions from the Trajectory Generator.

4.3 Resource Isolation and Virtualization

In the fog layer, the concurrently running applications may influence each other, especially during high utilization. It is given by the fact that the fog nodes are realised using common hardware, utilizing common resources as system buses, CPUs and memory. A proper mechanism that ensures that a number of concurrently running application on a single fog node do not interfere with each other in an unpredictable manner must be designed. Additionally, the applications must finish the computation within a given time and improper isolation of resources may introduce unpredictable delays and affect the timing requirements. Moreover, failure of an application must not lead to failure of other applications running on a single node.

4.4 Resource Estimation and Workload Optimization

To ensure real-time performance, a proper analysis of the system must be done to estimate the resources necessary for an application. Also, the system workload can vary significantly depending on the applications running at any given time. Appropriate strategies to deal with these situations must be developed. One of the solutions can be dynamical provisioning of fog nodes [2], where, if the workload is high (or is predicted to be high), additional fog nodes are started up.

4.5 Monitoring and Optimization

The architectural transition from a dedicated controller to the whole distributed layer of fog nodes introduces additional complexities to the robot control. There may be errors in the system due to faulty resource allocation, faults in communication between the robots and the fog, faults in communication between fog nodes and in virtualization, etc. Thus, the whole fog layer needs thorough monitoring to enable traceability of functional/non-functional properties of the system to address the errors due to these faults. Additionally, monitoring is a key component to enable optimization and dynamic reconfiguration of the fog layer.

4.6 Safety

The system should identify and recover from unexpected states caused by failure of fog nodes (hardware or software) or communication links. Employing a single orchestrating node that manages the whole fog network introduces a single point of failure that may paralyze the whole fog layer and affect the safety of the system. Techniques to address such scenarios need to be investigated.

4.7 Security

The use of fog computing allows centralized updating and deploying applications in the fog environment. An attacker can remotely take over the manufacturing process and make the system unsafe. This may be a potential security risk that must be taken into account. Therefore, a proper access control and intrusion detection mechanism must be implemented at different layers of the architecture.

5 CONCLUSION AND FUTURE WORK

The traditional robot system architectures need to evolve to a new architectural paradigm to meet the demands of flexible production environments. The proposed fog-based system architecture addresses such demands while introducing new research challenges that need to be addressed. In our future work, we intend to address the research challenges associated with the fog-based architecture.

ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 764785, FORA—Fog Computing for Robotics and Industrial Automation.

REFERENCES

[1] A. Vick and J. Krueger. 2018. Using OPC UA for Distributed Industrial Robot Control. In *ISR 2018; 50th International Symposium on Robotics*. 1–6.

[2] Said El Kafhali and Khaled Salah. 2017. Efficient and dynamic scaling of fog nodes for IoT devices. *The Journal of Supercomputing* 73, 12 (2017), 5261–5284.

[3] Hamid Reza Faragardi, Saeid Dehnavi, Mehdi Kargahi, Alessandro V. Papadopoulos, and Thomas Nolte. 2018. A time-predictable fog-integrated cloud framework: One step forward in the deployment of a smart factory. In *2018 Real-Time and Embedded Systems and Technologies (RTEST)*. IEEE, 54–62.

[4] Brendan Galloway and Gerhard P. Hancke. 2013. Introduction to Industrial Control Networks. *IEEE Communications Surveys & Tutorials* 15, 2 (2013), 860–880.

[5] Marisol García-Valls, Tommaso Cucinotta, and Chenyang Lu. 2014. Challenges in real-time virtualization and predictable cloud computing. *Journal of Systems Architecture* 60, 9 (2014), 726–740.

[6] Thomas Goldschmidt, Stefan Hauck-Stattelmann, Somayeh Malakuti, and Sten Grüner. 2018. Container-based architecture for flexible industrial control applications. *Journal of Systems Architecture* 84 (2018), 28–36.

[7] Antoni Grau, Marina Indri, Lucia Lo Bello, and Thilo Sauter. 2017. Industrial robotics in factory automation: From the early stage to the Internet of Things. In *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 6159–6164.

[8] Jan Guhl, Son Tung, and Jorg Kruger. 2017. Concept and architecture for programming industrial robots using augmented reality with mobile devices like microsoft HoloLens. In *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 1–4.

[9] Daniel Hallmans, Kristian Sandström, Thomas Nolte, and Stig Larsson. 2015. Challenges and opportunities when introducing cloud computing into embedded systems. In *2015 IEEE 13th International Conference on Industrial Informatics (INDIN)*. IEEE, 454–459.

[10] Zijiang Hao, Ed Novak, Shanhe Yi, and Qun Li. 2017. Challenges and Software Architecture for Fog Computing. *IEEE Internet Computing* 21, 2 (2017), 44–53.

[11] Siva Leela Krishna Chand Gudi, Suman Ojha, Benjamin Johnston, Jesse Clark, and Mary-Anne Williams. 2018. Fog Robotics for Efficient, Fluent and Robust Human-Robot Interaction. In *2018 IEEE 17th International Symposium on Network Computing and Applications (NCA)*. IEEE, 1–5.

[12] Y. Lin and H. Shen. 2015. Cloud Fog: Towards High Quality of Experience in Cloud Gaming. In *2015 44th International Conference on Parallel Processing*. 500–509.

[13] Redowan Mahmud, Ramamohanarao Kotagiri, and Rajkumar Buyya. 2018. Fog Computing: A Taxonomy, Survey and Future Directions. In *Internet of Everything*, Beniamino Di Martino, Kuan-Ching Li, Laurence T. Yang, and Antonio Esposito (Eds.). Springer Singapore, Singapore, 103–130.

[14] Michal Matuga. 2018. Control and positioning of robotic arm on CNC cutting machines and their applications in industry. In *2018 Cybernetics & Informatics (K&I)*. IEEE, 1–6.

[15] Nader Mohamed, Jameela Al-Jaroodi, and Imad Jawhar. 2018. Fog-Enabled Multi-Robot Systems. In *2018 IEEE 2nd International Conference on Fog and Edge Computing (ICFEC)*. IEEE, 1–10.

[16] S. Mubeen, P. Nikolaidis, A. Didic, H. Pei-Breivold, K. Sandström, and M. Behnam. 2017. Delay Mitigation in Offloaded Cloud Controllers in Industrial IoT. *IEEE Access* (2017), 4418–4430.

[17] Volker Paelke. 2014. Augmented reality in the smart factory: Supporting workers in an industry 4.0. environment. In *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*. IEEE, 1–4.

[18] Christoph Pallasch, Stephan Wein, Nicolai Hoffmann, Markus Obdenbusch, Tilman Buchner, Josef Waltl, and Christian Brecher. 2018. Edge Powered Industrial Control: Concept for Combining Cloud and Automation Technologies. In *2018 IEEE International Conference on Edge Computing (EDGE)*. IEEE, 130–134.

[19] P. Pop, M. L. Raagaard, S. S. Craciunas, and W. Steiner. 2016. Design optimisation of cyber-physical distributed systems using IEEE time-sensitive networks. *IET Cyber-Physical Systems: Theory Applications* 1, 1 (2016), 86–94.

[20] Paul Pop, Michael Lander Raagaard, Marina Gutierrez, and Wilfried Steiner. 2018. Enabling Fog Computing for Industrial Automation Through Time-Sensitive Networking (TSN). *IEEE Communications Standards Magazine* 2, 2 (2018), 55–61.

[21] D. Quarta, M. Pogliani, M. Polino, F. Maggi, A. M. Zanchettin, and S. Zanero. 2017. An Experimental Security Analysis of an Industrial Robot Controller. In *2017 IEEE Symposium on Security and Privacy (SP)*. 268–286.

[22] Per Skarin, William Tarneberg, Karl-Erik Å rzén, and Maria Kihl. 2018. Towards Mission-Critical Control at the Edge and Over 5G. In *2018 IEEE International Conference on Edge Computing (EDGE)*. IEEE, 50–57.

[23] Olena Skarlat, Matteo Nardelli, Stefan Schulte, and Schahram Dustdar. 2017. Towards QoS-Aware Fog Service Placement. In *2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC)*. IEEE, 89–96.

[24] Vaclav Struhar, Alessandro V. Papadopoulos, and Moris Behnam. 2018. Fog Computing for Adaptive Human-Robot Collaboration. In *International Conference on Embedded Software 2018*.

[25] Karima Velasquez, David Perez Abreu, Marcio R. M. Assis, Carlos Senna, Diego F. Aranha, Luiz F. Bittencourt, Nuno Laranjeiro, Marilia Curado, Marco Vieira, Edmundo Monteiro, and Edmundo Madeira. 2018. Fog orchestration for the Internet of Everything: state-of-the-art and research challenges. *Journal of Internet Services and Applications* (2018).

[26] Ashkan Yousefpour, Caleb Fung, Tam Nguyen, Krishna Kadiyala, Fatemeh Jalali, Amirreza Niakanlahiji, Jian Kong, and Jason P. Jue. 2018. All One Needs to Know about Fog Computing and Related Edge Computing Paradigms: A Complete Survey. (2018).