CLOUD-BASED CONTROL OF INDUSTRIAL CYBER-PHYSICAL SYSTEMS

Peter Papcun[1*] , Erik Kajáti[1], Chao Liu[2], Ray Y. Zhong[2] and Iveta Zolotová[1]
[1]Department of Cybernetics and Artificial Intelligence, Faculty of Electrical Engineering and Informatics, Technical University of Košice, Letná 9, 042 00 Košice, Slovak
peter.papcun@tuke.sk

[2]Department of Mechanical Engineering, University of Auckland, Auckland 1010, New Zealand
cliu810@aucklanduni.ac.nz

ABSTRACT

This paper presents an implementation of a control algorithm to a cloud system. The motivation is that cloud implementations of low-level systems in the production industry are gradually becoming more common. Microsoft Azure platform is utilized for the cloud-based control and the case is tested using a customized laboratory model, which can be presented as an agent in a typical production system. The model offers the regulation of a ball on an inclined surface and uses two asynchronous motors connected to frequency converters to control the position of the ball. These frequency converters are controlled by a Programmable Logic Controller (PLC). Windows Communication Foundation (WCF) services and Azure IoT Hub were selected to be used with the cloud-based control system. Experimental results have shown our solution can control the system with sampling period equal or higher than 100ms. The latency of WCF service is at around 100ms and latency of Azure IoT Hub is at around 1000ms, so the prediction algorithms could be implemented in the cloud for the latter. This research also shows the feasibility of migrating machine learning algorithms that demand high computing power to the cloud to reduce the computing burden on the local control units.

Keywords: Cloud, Industry 4.0, Internet of Things (IoT), Windows Communication Foundation (WCF), Microsoft Azure, Cloud-based control.

---

[*] Corresponding Author

1 INTRODUCTION

Industry 4.0 facilitates the vision and execution of a “Smart Factory”. Within the modular structured Smart Factories, Cyber-Physical Systems (CPS) monitor physical processes, create a virtual copy of the physical world and make decentralized decisions. Over the Internet of Things (IoT), CPS communicate and cooperate with each other and with humans in real time, and via the Internet of Services, both internal and cross-organizational services are offered and utilized by participants of the value chain.

There are four design principles in Industry 4.0 that support companies in identifying and implementing Industry 4.0 scenarios [1]:

- Interoperability: The ability of machines, devices, sensors, and people to connect and communicate with each other via the IoT or the Internet of People.
- Information transparency: The ability of information systems to create a virtual copy of the physical world by augmenting digital plant models with sensor data. This requires aggregating the raw sensor data to higher-value context information.
- Technical assistance: First, the ability of assistance systems to support humans by aggregating and visualizing information comprehensible for making informed decisions and solving urgent problems on short notice. Second, the ability of cyber-physical systems to physically support humans by conducting a range of tasks that are unpleasant, exhausting, or unsafe for their human co-workers.
- Decentralized decisions: The ability of cyber-physical systems to make decisions on their own and to perform their tasks as autonomously as possible. Only in a case of exceptions, interferences, or conflicting goals, tasks are delegated to a higher level.

Cloud computing, also known as on-demand computing, is a kind of Internet-based computing that provides shared processing resources and data to computers and other devices on demand. It is a model for enabling ubiquitous, on-demand access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services), which can be rapidly provisioned and released with minimal management effort. Cloud computing and storage solutions provide users and enterprises with various capabilities to store and process their data in third-party data centers [2].

The Industrial Internet of Things (IIoT) has been heralded primarily as a way to improve operational efficiency. But in today’s environment, companies can also benefit greatly from seeing it as a tool for finding growth in unexpected opportunities. In the future, successful companies will use the IIoT to capture new growth through three approaches: boost revenues by increasing production and creating new hybrid business models, exploit intelligent technologies to fuel innovation, and transform their workforce.

The rest of this paper is organized as follows. Section 2 presents the literature review. Section 3 talks about the proposed system architecture. Section 4 demonstrates the synthesis of PID controller. System implementation is discussed in Section 5. Section 6 gives the discussions based on the experimental results. Conclusions are drawn in Section 7.

2 LITERATURE REVIEW

During recent years, the Industry 4.0 concept brings new demands and trends in different areas. Machine tools are changing in this industrial revolution and have also gone through different stages of technological advancements [3]. Cyber-Physical Machine Tools (CPMT) provides a promising solution for Machine Tool 4.0 – a new generation of machine tools. Liu et al. [4] proposed a generic system architecture to provide guidelines for advancing existing Computer Numerical Control (CNC) machine tools to CPMT. The processes require more and more computational power. The end devices can offer some of the computational power, but these devices often do not meet the requirements for the computationally demanding processes. These processes may include ones that require algorithms of the computer vision, machine learning, or data analytics [5]. With the rapid development of computer networks and Internet connections anywhere in the world, there is an opportunity to offload some of the computationally demanding algorithms to the cloud from the end devices.

Givehchi et al. [7] presented a cloud-based solution that aims at offering control-as-a-service for an industrial automation case (Figure 1). According to the authors, Programmable Logic Controller (PLC) could not only remain in the shop floor as a physical device, but also be implemented as a virtual entity and delivered to the field as a service from a CPS via the network.

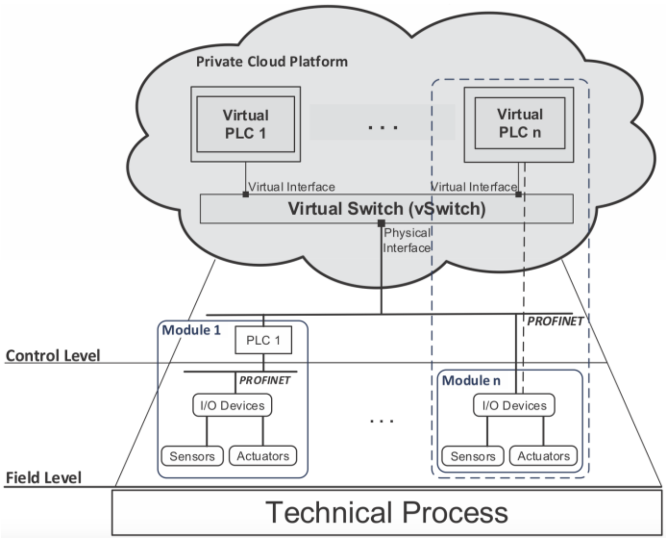

Figure 1: A generic cloud-based control approach [7]

Their results showed that there is some reduction of performance for cloud-based scenario compared with a hardware PLC. This reduction in performance is mostly in systems with higher sampling rates (32ms and faster). In processes that require sampling rate 64ms and slower, the difference between a cloud-based scenario and the hardware PLC is lower and mostly negligible, but it also depends on the type of the application. Hence, their solution is promising for soft real-time applications.

Schlechtendahl et al. [8] conducted two use cases of cloud-based control system using two milling machines. The data transferred between the control system and the machine tools have been analyzed. Then the data was used to analyze if a control system as a service (CSaaS) is possible. A communication test setup was developed as shown in Figure 2.

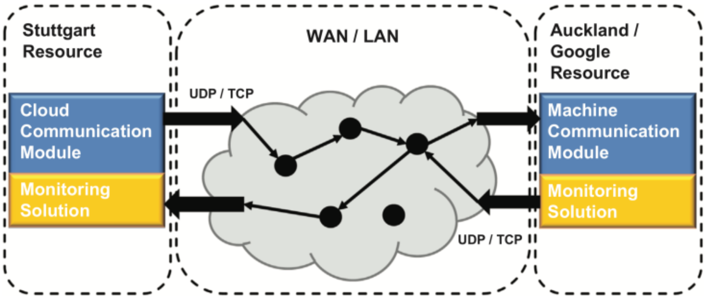

Figure 2: Test setup communication [8]

Cloud communication module – located in Stuttgart, Germany, is creating the data that is transferred from the cloud to the machine. The machine communication module, which receives the data from Stuttgart, is located either in Auckland, New Zealand or in Google cloud center located in Europe. The machine communication module receives and log the data, and as a second step creates and transmits the data from the machine to the cloud system. Communication channels can be configured for different connection protocols - User

Data Protocol (UDP), Transmission Control Protocol (TCP) and WebSocket Protocol. The authors conclude that CSaaS between New Zealand and Germany is not possible since the network challenges are serious. The control system should be located closer to the machine. CSaaS between the Google Cloud Centre in Europe and Germany is possible for processes that require slow cycle times [8].

3 THE PROPOSED CPS-ENABLED SYSTEM ARCHITECTURE

Based on the review of the literature we have chosen a laboratory model to test our proposed architecture (Figure 3). The laboratory model works with the sampling time 100ms, therefore it does not require hard real-time control. The system is controlled with the PLC which is connected to the local computer in the laboratory through OPC-DA connection. The model can be controlled locally with PLC controller or from cloud with the use of Microsoft Azure Cloud services – IoT Hub and Windows Communication Foundation (WCF) Framework.

The laboratory model Traverse (Figure 4) is located in a laboratory at the Department of Cybernetics and Artificial Intelligence, Faculty of Electrical Engineering and Informatics at Technical University of Košice.

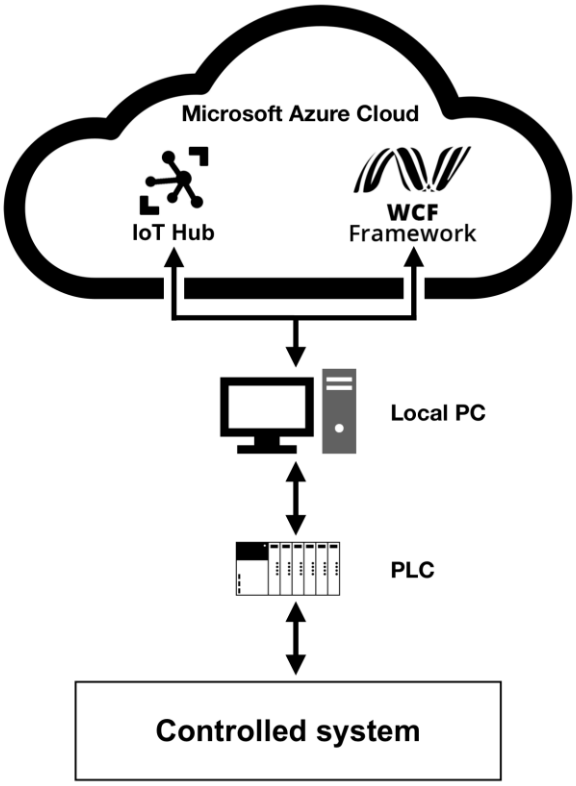

Figure 3: Proposed architecture for cloud control

The controlled agent offers a regulation of a ball on an inclined surface. The inclined surface consists of two cylinders that are connected to the wooden construction of the bridge (Figure 5). The length of the bridge is 67cm and the diameter of the ball is 10cm. Thus, the total length of the path that the ball can travel is 57cm from end to end. This bridge is suspended on two steel cables connected to the axles of two asynchronous motors with an electro-mechanic brake and incremental rotary encoders. Axle rotation of the left or the right motor is transformed to the height change of the left or right end of the bridge. The motors are controlled with two frequency converters. These frequency converters are controlled by an analog signal from the PLC.

Position sensing of the ball is provided by the two cylinders that support ball. One of the cylinders is made from brass, and the other is wrapped in copper wire. These cylinders use the principles of rheostat to transform the physical position of the ball to the voltage signal. This voltage signal is connected to the analog voltage input of the PLC.

Figure 4: Laboratory model Traverse

Figure 5: The bridge

Despite its specific functions, this model can represent various kinds of agents in an industrial system. One example is the industrial control systems. If there is more than one agent in the system, it becomes a multi-agent system where agents can collaborate or cooperate with each other.

This solution is an example of using cloud systems in the technological level of control. Our solution tests the control algorithms implemented on a cloud and proposes an architecture for the communication with the cloud. This architecture can be used to migrate parts of the algorithms, that demand high computing power, to the cloud. For example, processes of control that use image recognition, neural networks and machine learning. To examine control with the cloud technology, we applied a proportional-integral-derivative (PID) algorithm. This case study compares local PID control and cloud-based PID control on the Microsoft Azure cloud platform. In future research and development, control algorithm with image recognition will be migrated to the cloud.

4 SYNTHESIS OF PID CONTROLLER

At first, the model Traverse has to be identified for the synthesis of PID controller. Then the controller is designed and implemented on the PLC for the local control of the model. Furthermore, these algorithms are implemented on the cloud.

4.1 Identification of the system

The equation (1) represents the kinematic model of the ball motion.

$$s = \frac{1}{2} g t^2 I(\sin(\alpha) - f_d \cos(\alpha)) \qquad \textbf{(1)}$$

Where s is the length of the path travelled by the ball in time t, g is gravity acceleration, I represent torque of the ball, α is the inclination of the bridge measured in degrees. Parameter f_d is the coefficient of rolling friction of the ball that is quite high, due to the fact, that ball is tightly fitted between the two cylinders. The coefficient of rolling friction was measured experimentally.

The transfer function of the system in Laplace form ($F_s(s)$) is:

$$F_s(s) = \frac{1.422s^2 + 6.547s + 24.674}{s^3} \qquad \textbf{(2)}$$

The accuracy of the equation was tested by comparing the responses of the Matlab/Simulink simulated system (Figure 6) and the real system to the step signal (Figure 7).

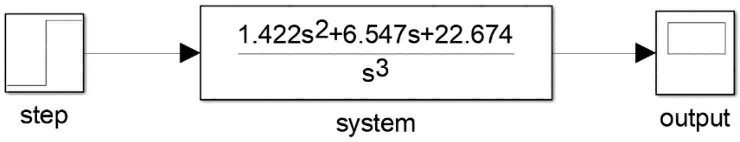

Figure 6: Simulation scheme in Matlab/Simulink

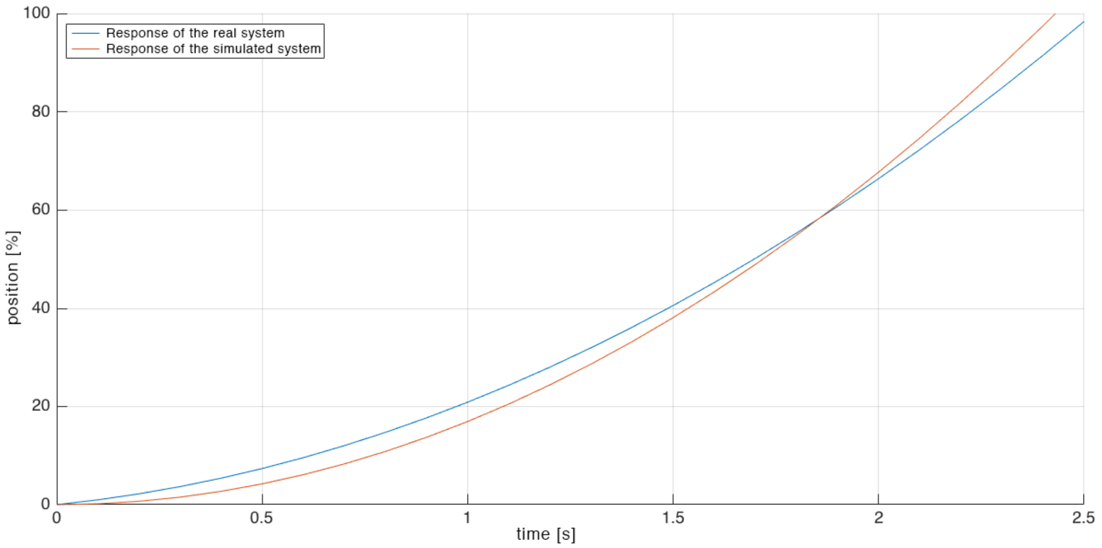

Figure 7: Response to the step signal

4.2 Design of the controller

A PID controller was designed with Naslin method [11]. Transfer function of the PID controller (*$F_c(s)$*) is described by the equation (3) where *K* is proportional coefficient, *T_i* is integral coefficient, and *T_d* is derivative coefficient of the PID controller.

$$F_c(s) = K + \frac{1}{T_i s} + T_d s \tag{3}$$

$F_C(s)$ and *$F_S(s)$* functions were applied to closed-loop transfer function *$F_{FBC}(s)$* (4):

$$F_{FBC}(s) = \frac{F_c(s)F_s(s)}{1 + F_c(s)F_s(s)} \tag{4}$$

The denominator of the transfer function of the feedback control represents the characteristic polynomial (5):

$$F_{CHP}(s) = a_i s^i + a_{i-1} s^{i-1} + \cdots + a_1 s + a_0 \tag{5}$$

The coefficients (a_i) are used to compute coefficients for the controller by Naslin method with equation (6):

$$a_i^2 \geq 2. a_{i-1}. a_{i+1} \tag{6}$$

PID controller was applied in classical closed-loop feedback control (Figure 8), where *w(t)* is desired value, *e(t)* is regulation error, *u(t)* is system input, and *y(t)* system output. The transfer function (4) was derived from the scheme shown in Figure 8.

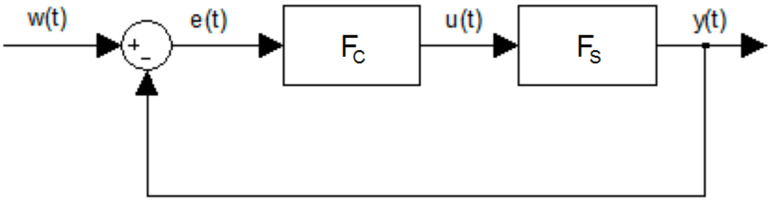

Figure 8: Feedback control scheme

These coefficients were then tested in the closed-loop system within software Matlab/Simulink with positive results (Figure 9) for multiple desired output values.

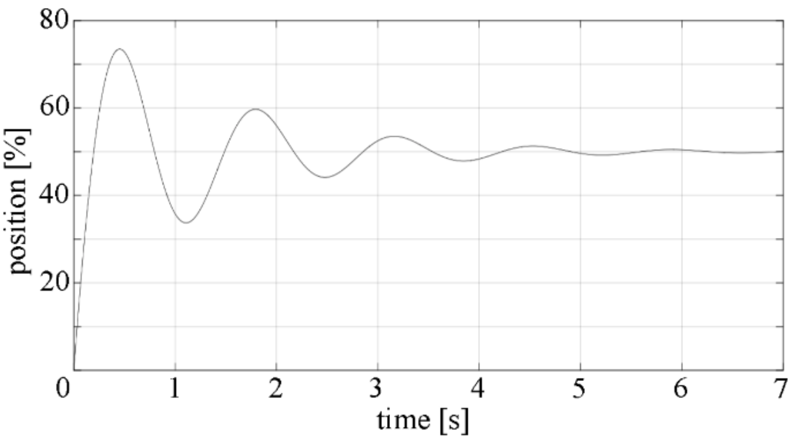

Figure 9: Simulation of system with PID controller. The setpoint is equal to 50%.

5 SYSTEM IMPLEMENTATION

5.1 Local control

The local control was implemented with the use of the PLC Allen-Bradley ControlLogix 5573. The ladder diagram program was created with implementation of discrete local control algorithm in Studio5000 development environment (7):

$$u(k) = Pe(k) + I\sum_{i=0}^{k} e(i) + D(e(k) - e(k-1)) \tag{7}$$

Where *u(k)* is control signal and *e(k)* is an error signal. The error signal in step *k*, *e(k)* is computed as a difference between the desired value *w(k)* and the actual output of the system *y(k)*.

The local algorithm works with 100ms sampling period. Anytime, when the error signal is outside of the <-1; 1> interval and the difference of the current output signal and the output signal in the previous step is outside of the <-0.3; 0.3> interval the controller algorithm's output is computed and applied. These two conditions ensure that controller does not compute when the desired value is the same as the actual output of the system so the computational resources are saved. PLC also limits the inclination of the bridge within the <-6; 6> degrees interval.

5.2 Cloud-based control using WCF service

Cloud-based control was applied with the use of Microsoft Azure cloud platform. Windows Communication Foundation (WCF) is a framework for building service-oriented applications. WCF can send data as asynchronous messages from one service endpoint to another. The service endpoint can be a part of a continuously available service hosted by IIS, or it can be a service hosted in an application. The endpoint can be a client of a service that requests data from a service endpoint. The messages can be as simple as a single character or a word sent as XML, or as complex as a stream of binary data [9].

As a consequence of using Windows Service (WS) standards, WCF enables creating service-oriented applications. Service-oriented architecture (SOA) relies on Web services to send and receive data. The services have the general advantage of being loosely-coupled instead of hard-coded from one application to another. A loosely-coupled relationship implies that any client created on any platform can connect to any service as long as the essential contracts are met [9].

After testing the latency of the WCF service. The average latency time of 100ms was computed. This time represents the delay between sending values from the local unit and reception of the computed answer from the cloud. This relatively low latency time ensures that for control the model with WCF service, the same algorithm can be used as we use for local control.

For cloud-based control with WCF service, a program with a graphical interface was created. The program was developed with the use of Microsoft Visual Studio 2015 development environment. The program works with 100ms sampling period. So, in every step, data are downloaded from PLC. Data consist of the actual position of the ball and desired input value. This data is sent to the cloud, where control algorithm is implemented. Then the computed control values are sent back from the cloud. After the program receives the control value from the cloud, the value is sent to PLC. The control value is then saturated to ensure that inclination of the bridge is within <-6; 6> degrees bounds, and the bridge is inclined to desired and saturated value.

5.3 Cloud-based control using IoT Hub

Cloud-based control was applied with the use of Microsoft Azure cloud platform. Azure IoT Hub is a fully managed service that enables reliable and secure bidirectional communications between millions of IoT devices and a solution back end. Azure IoT Hub has the following features [10]:

- Provides reliable device-to-cloud and cloud-to-device messaging at scale.
- Enables secure communications using per-device security credentials and access control.
- Provides extensive monitoring for device connectivity and device identity management events.
- Includes device libraries for the most popular languages and platforms.

Azure IoT Hub implements the service-assisted communication pattern to mediate the interactions between devices and the solution back end. The goal of service-assisted communication is to establish trustworthy, bidirectional communication paths between a control system, such as IoT Hub, and special-purpose devices that are deployed in untrusted physical space [10].

The latency of the IoT Hub has been tested by sending the actual time from the end device, IoT Hub service then immediately reply by sending the message back to the end device, after receiving message we subtracted time from message from actual time and we get time difference which is the latency. The average latency time was computed as 1000ms. This latency represents the duration between sending values from local unit and receiving the computed answer from the cloud. It is recognized that the latency is quite high for this system, hence the prediction of the position of the ball was applied to the control algorithm. Calculation of the following positions is achieved by applying the kinematic model of ball motion on the inclined surface. In equation (1), time t is represented by the average latency; a is represented by the actual angle of inclination of the bridge; value s is represented by the actual position calculated from the received packet. This sum is then used as the input to the control algorithm (regulation) for latency compensation.

To implement cloud-based control with IoT Hub service, a control program with WCF service is utilized. In every step, the data is downloaded from the PLC. The data consists of the ball's actual position, the desired input value and the actual inclination of the bridge. This data is sent to the cloud, where control algorithm is used to compute the predicted ball position as explained above. Then the control value is sent back from the cloud. After the program receives the control value from the cloud, the value is sent to PLC. The control value is then saturated to ensure that inclination of the bridge is within <-1.5; 1.5> degrees bounds, and the bridge is inclined to desired and saturated value. Saturation values with IoT Hub are different from <-6; 6> bounds that are used with WCF and local control. This is because of the relatively short bridge length and high latency of the IoT Hub service, so the ball will move slower.

6 EXPERIMENTAL RESULTS AND DISCUSSION

We have tested the response of the system for many different desired values. These values represent movement from 10% to 100% of the total length of the bridge. The ball was always positioned at the left side of the bridge (Figure 5). The left side of the bridge represents the position 0%. In each test, the desired position was set after the first second.

All results from the experiments were compared. Comparisons of the local algorithm and cloud-based algorithms are in the following figures. Here we have chosen the position differences of 20% and 60% as two representatives. Since our experiments have shown that the results of position differences between 10% and 35% were similar; likewise, the results between 40% and 100% were similar. The position difference refers to the distance from the starting point to the desired position. For example, the position differences from 0% to 20% and from 50% to 70% are both 20%. Figure 10 shows the response of the system when the desired value for the position of the ball is 20%.

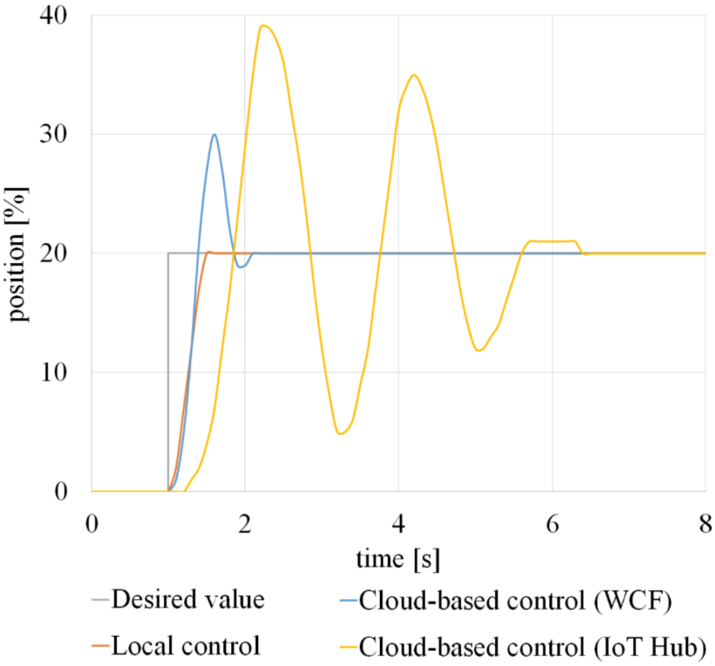

Figure 10: Comparison of local and cloud algorithms, w=20%

The local algorithm (the red line) performed the best, bringing the ball to the desired position without overshooting in less than one second. The blue line represents the cloud-based control using the WCF service. The regulation was relatively fast too, with one overshoot to 30%. The yellow line represents the cloud-based control using the IoT Hub. This line is oscillating because we have used the proportional controller only at the IoT Hub and the latency was relatively high. After almost six seconds the ball was brought to the desired position and the predicted positions were calculated. The system was unstable without the predicted values being used and with the latency of approximate 1000ms.

Figure 11 shows the response of the system when the desired value for the position of the ball is 60%. It is noted that a paradox occurs since the cloud-based control using the WCF service is faster than the local control. We have observed this phenomenon in some specific measurements. It is caused by higher latency that delays the effect of the derivative coefficient of the PID controller. But it is also noted that the cloud-based control using the WCF service is oscillating more than the local algorithm. The local control overshoots once and the ball stops for a moment. The integration part of the controller ensures that the ball is rolled back to the desired position. These two algorithms control the ball to the desired position in about three seconds. Cloud-based control using the IoT Hub again shows oscillating progress and finishes in nearly seven seconds. Again, this oscillation is caused by using only the proportional controller on the IoT Hub. The ball can be controlled to the desired position only when the predicted positions of the ball are used.

Table 1 shows the comparison of minimum, maximum and average regulation time of each algorithm when the desired position was 20%. The measurements were repeated ten times. Only small differences of regulation time are observed in the local control. The regulation time for the cloud-based control using the WCF service is approximately double compared to the local control. Larger deviation in the regulation time of the control using the WCF is present because of slightly variable latencies. The slowest control is the cloud-based control using the IoT Hub and it also has the largest deviations in the values. The time needed to control the ball with the IoT Hub is more than ten times of the time achieved by the local control.

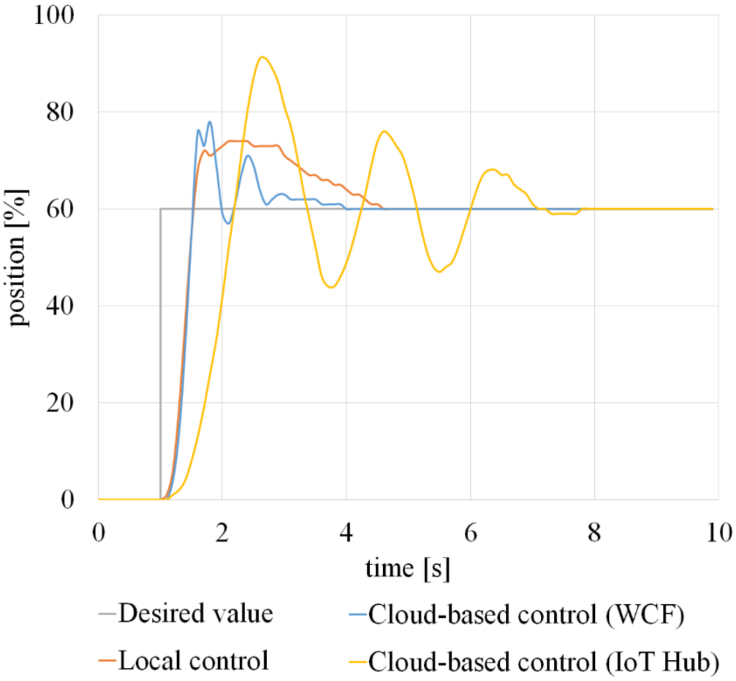

Figure 11: Comparison of local and cloud algorithms, w=60%

Table 2 compares the average overshoot of each algorithm when the desired position was 20%. The measurements were repeated ten times. The local control does not overshoot. Cloud-based control using the WCF service overshoots by 50% at average. Cloud-based control using the IoT Hub overshoots by 90% at average because only the proportional gain is used.

Table 1: Regulation times to desired position 20%

Algorithm used	Minimum	Maximum	Average
Local control	0.42 s	0.55 s	0.47 s
Cloud (WCF)	0.98 s	1.36 s	1.12 s
Cloud (IoT Hub)	5.02 s	6.05 s	5.42 s

Table 2: Overshoot, the desired position is 20%

Algorithm used	Average overshoot
Local control	0 %
Cloud (WCF)	50 %
Cloud (IoT Hub)	90 %

Table 3 compares the minimum, maximum and average regulation time of each algorithm when the desired position was 60%. The measurements were repeated ten times. There are only small differences of the regulation time in the local control. The regulation time of the cloud-based control using WCF service are paradoxically lower by approximate half a second than that of the local control. The slowest control is the cloud-based control using the IoT Hub and it also has the largest deviations in the values.

Table 3: Regulation times to desired position 60%

Algorithm used	Minimum	Maximum	Average
Local control	3.36 s	3.79 s	3.55 s
Cloud (WCF)	2.92 s	3.56 s	3.05 s
Cloud (IoT Hub)	6.23 s	7.12 s	6.78 s

Table 4 compares the average overshoot of each algorithm when the desired position was 60%. The measurements were again repeated ten times. Cloud-based control with WCF service was slightly faster than the local control but overshooting and oscillating more. The average overshoot was less dispersed when the desired position was 60% than when it was 20%. It is because our PID controller works better for shorter distances. This is caused by the choice of the operating region during the synthesis of the PID controller and also there are nonlinearities in the system and PID controller was computed for linear system. The computed coefficients for the PID controller are not the best for all situations but regulation was sufficient to control our model.

Table 4: Overshoot, the desired position is 60%

Algorithm used	Average overshoot
Local control	23 %
Cloud (WCF)	28 %
Cloud (IoT Hub)	35 %

The control using the WCF service has the latency of approximate 100ms. This is low enough to implement the same algorithms for the cloud-based and the local control. There was only a small difference between the local control and cloud-based control quality. The control using the IoT Hub service had the latency of approximate 1000ms. This required calculating the future ball position as the input to the control algorithm. Without it the system was unstable because of high latency and relatively short bridge.

Our experimental tests of cloud-based control used a simple PID control algorithm. This algorithm can be replaced with a more complex algorithm. The developed architecture can be used to migrate parts of the algorithms requiring high computational resources to the cloud, e.g. image recognition, artificial neural networks or machine learning algorithms.

7 CONCLUSION

In this paper, a cloud-based PID control on Microsoft Azure cloud platform is implemented and compared to the local control method. We consider the research contribution of our work to be the definition of the methodology reflected in our architecture (Figure 3). The industries taking advantages of IIoT benefits start with migration from OPC-DA to OPC-UA. This transfer is expensive and will take a long time. We have proposed an architecture which can be used for OPC-DA systems to connect to the cloud using IoT technology to fill the gap during the process of migration. This methodology was tested on the Traverse laboratory model. One of our objectives was the implementation of the control algorithm to the cloud. This objective was accomplished. The results of PID control were sufficient to control our model. The feasibility of implementing different types of control algorithms in the proposed architecture has been validated through the experiments. The opportunities provided by the cloud technology are numerous.

Currently, the control units used in production systems such as PLC and programmable automation controller (PAC) have very limited computing power. Progress in the improvement of these units is not fast enough due to the high requirement of safety and security. Cloud systems allow for computing redundancy to ensure safety. Intelligent sensors and actuators can also be connected directly to the Internet by switches or routers and controlled from the cloud systems. This type of cloud-based control may be defined as CaaS (Control as a Service, [7] and [14]). CaaS is a software service specialized for control. CaaS is not defined as a basic group of cloud services yet. The future control algorithms can be built on these services. We have verified that it is possible to control the selected physical system with a sampling period of 100ms. Based on that, we proved that our solution can control systems with sampling period equal or higher than 100ms that do not require a hard real-time control. We have also proved that this system can be successfully controlled with cloud service that has latency at around 1000ms with the use of prediction algorithms, although it requires some restrictions to the control algorithm.

In the future we plan to migrate control algorithms requiring image recognition to the cloud. The implementation of the control of the Traverse model may omit the use of PLC later. Communication with frequency converters through the analog signal from PLC might be replaced with communication over a serial link using e.g. the Arduino platform. Arduino can have the Ethernet shield connected and can communicate with the cloud directly, thus eliminating the need to use a PC. The cost of hardware components needed to control the model from a cloud will be significantly reduced. Also, we are working on the solution where local computer will not be needed but instead we will use and compare eWon industrial router with Raspberry Pi for data acquisition and transmitting to the cloud. We are also exploring the area of edge-enabled architecture as described in [15] and [16]. Moreover, cloud-based SCADA systems [17] as described in [18] is another future research direction. In the future, a combination of these architectures into one functional system will be conducted.

ACKNOWLEDGEMENTS

This publication was supported by the grant VEGA - 1/0663/17, 2017-2020.

8 REFERENCES

[1] M. Hermann, T. Pentek, and B. Otto, “Design Principles for Industrie 4.0 Scenarios”, 2016, ISSN 1530-1605

[2] Q. F. Hassan, “Demystifying Cloud Computing”, The Journal of Defense Software Engineering, Jan/Feb, 2011, p.16-21

[3] XU, Xun. Machine Tool 4.0 for the new era of manufacturing. The International Journal of Advanced Manufacturing Technology, 2017, 92.5-8: 1893-1900

[4] LIU, Chao, et al. A systematic development method for cyber-physical machine tools. Journal of Manufacturing Systems, 2018.

[5] ZHONG, Ray Y., et al. Big data analytics for physical internet-based intelligent manufacturing shop floors. International journal of production research, 2017, 55.9: 2610-2621

[6] J. Lee, B. Bagheri, and H. A. Kao, “A cyber-physical systems architecture for industry 4.0-based manufacturing systems”, Manuf Lett, 2015, p.18 - 23

[7] Givehchi O., Imtiaz J., Trsek H., Jasperneite J., “Control-as-a-service from the cloud: A case study for using virtualized PLCs”, In: Workshop on Factory Communication

Systems (WFCS 2014), Toulouse, France, p. 1 - 4, 2014, DOI: 10.1109/WFCS.2014.6837587

[8] A SCHLECHTENDAHL, Jan, et al. Extended study of network capability for cloud based control systems. Robotics and Computer-Integrated Manufacturing, 2017, 43: 89-95.

[9] Microsoft Documentation, “What is Windows Communication Foundation”, http://www.msdn. microsoft.com/en-us/library/ms731082. Aspx

[10] Microsoft Documentation, “What is IoT Hub”, http://www.azure.microsoft.com/documentation/articles/iot-hub-what-is-iot-hub/

[11] MADARÁSZ, L., et al. Základy automatického riadenia. Elfa, sro, TU Košice, 2010. ISBN 978-80-8086-162-9.

[12] Liguš, I. Zolotová, P. Karch, and J. Ligušová, “Information and control system of traverse and its integration into cybernetic centre”, In: Electronics and electrical engineering, 2010, No. 6 (2010), p. 147 – 152, ISSN 1392-1215

[13] H. Hibino, “Method to Control Manufacturing Cell by Driving Simulation Model”, In: International Journal of Automation Technology, 2014, Vol. 8, No. 4, p. 539 – 549, ISSN 1883-8022

[14] H. Esen, “Control as a Service (CaaS): Cloudbased Software Architecture for Automotive Control Applications”, Talk or presentation, 2015

[15] T. Lojka, M Bundzel, and I. Zolotová, “Industrial Gateway for Data Acquisition and Remote Control”, Acta Electrotechnica et Informatica. Vol. 15/2 (2015), s. 43-48. - ISSN 1335-8243, 2015.

[16] Zolotová, M. Bundzel, T. Lojka, “Industry IoT gateway for cloud connectivity”, IFIP Advances in Information and Communication Technology, Vol. 460, 2015, p.59-66, ISBN 978-331922758-0

[17] T. Lojka, I. Zolotová, “Improvement of Human-Plant Interactivity via Industrial Cloud-Based Supervisory Control and Data Acquisition System”, Advances in Production Management Systems: APMS 2014: International Conference on Advances in Production Management Systems, Vol. 440, no. Part 3, 2014, p. 83 – 90, ISSN 1868-4238

[18] Sajid, H. Abbas, and K. Saleem, “Cloud-Assisted IoT-Based SCADA Systems Security: A Review of the State of the Art and Future Challenges”, Special section on the plethora of research in Internet of Things, 2016