

A Study on Optimizing Signal Path in Model Predictive Control and in Digital Twin of Three-Tank Pilot System Using Reference Architecture

Jukka Kortela* Kim Miikki**

* Aalto University School of Chemical Engineering, P.O. Box 16100, FI-00076 Aalto (e-mail: jukka.kortela@aalto.fi).

** Aalto University School of Chemical Engineering, P.O. Box 16100, FI-00076 Aalto (e-mail: kim.miikki@aalto.fi).

Abstract: This paper presents the model predictive control of a three-tank pilot system in a novel cloud-integrated industrial automation system. In addition, an accurate digital twin is implemented for a real system for a fault detection and diagnosis. The framework includes a state-of-the-art NodeJS-based gateway that communicates information between the cloud service and the automation system. The optimized signal path through the OPC DA is compared to the OPC UA Tunneler implementation through experiments on a real three-tank pilot system with an industrial ABB 800xA automation system. Furthermore, the results of the fault detection with the parity equations is presented. The results are presented and discussed.

Keywords: cyber physical system, industrial internet of things, rami, iira, iot, iiot, wireless sensors, opc ua, model predictive control, digital twin, three tank system, model, experiment, 5G, 6G

1. INTRODUCTION

Since the invention of the Programmable Logic Controller (PLC) in 1968, industrial automation has passed through milestones marked by advances in information technology. The first milestone is the linking of the PLC with the personal computer (PC) in 1986. The next main milestone was reached in 1992 with the introduction of Ethernet and Transmission Control Protocol and Internet Protocol (TCP/IP) connectivity for PLCs. The current trend in industries is the transformation from industrial Ethernet and wireless communications to advanced information technology (IT) solutions where traditional automation is merged with cyber-physical systems (CPS) combining communications, information and communication technology (ICT), data and physical elements and the ability to connect devices to one another. This transformation results in what is now known as Industrial Internet of Things (IIoT) or the 4th Industrial Revolution (Industry 4.0) where every step of a manufacturing process is interconnected. Cloud computing and data analytics are among the technologies driving the IIoT. (IEC, 2015)

According to IEEE, the term architecture in the context of information technology is "the fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution". (IEEE, 2022)

A reference architecture in the context of information technology documents such things as hardware, software, processes, specifications and configurations, logical modules and interrelationships. According to IBM Rational

Unified Process, a reference architecture "is, in essence, a predefined architectural pattern, or set of patterns, possible partially or completely instantiated, designed, and proven for use in particular business and technical contexts, together with supporting artifacts to enable their use. Often, these artifacts are harvested from previous projects". (Evensen, 2013)

A reference architecture for IIoT serves the purpose of providing common and consistent definitions for the IIoT, its subsystems and design patterns, and a common lexicon and taxonomy for discussing specification of implementations of IIoT.

Currently, there are several reference architectures that can be employed for deploying Factory of Future (FoF). Two of the most popular reference architectures, RAMI 4.0 and IIRA, and their interoperability will be discussed in the following paragraphs. RAMI 4.0 is the product of Industrie 4.0, which is a national project of the German government initiated in 2011 through German's Ministry of Education and Research (BMBF) and the Ministry for Economic Affairs and Energy (BMWi) (European Commission, 2017), which has since been joined by industries and academia to form a consortium called Plattform Industrie 4.0. Industrie 4.0 aims to increase value in manufacturing and decrease waste by transforming the way products are developed, produced, managed, and consumed. The project focuses on the industrial manufacturing sector and connects value chains by integrating things and processes to form cyber physical systems (CPS). The novelty of Industrie 4.0 results from the combination of already existing and new technologies such as embedded

computers, intelligent sensors, mobile broadband internet access, and Radio Frequency Identification (RFID) in the industrial environment into a uniform, integrated solution through standardized communication. (VDI Verein Deutscher Ingenieure e.V., 2015) The Industrie 4.0 concept is implemented through RAMI 4.0; a service-oriented architecture that has been designed for efficient sharing of data and information between all the shareholders taking part in the Industrie 4.0 ecosystem. RAMI 4.0 (registered DIN SPEC 91345 in Germany) ensures that all participants in Industrie 4.0 share a common perspective and build a common understanding. (DIN, 2016)

The Industrial Internet Consortium (IIC) first published IIRA in the form of a Technical Report in 2015. Founded by AT&T, Cisco, General Electric, IBM, and Intel, the mission of IIC is to reach industrial interoperability and consensus on IIoT platforms. The IIC is a part of the Object Management Group (OMG) and today has 19 working groups and over 250 members of industrial and academic background. In July 2019, the latest version of IIRA, IIRA v1.9, was published by the Industrial Internet Consortium Architecture Task Group, which is a subset of the IIC Technology Working group. Industrial Internet Consortium (2019) IIRA is a reference architecture of what IIC calls Industrial Internet Systems (IIS). These systems are defined as end-to-end application systems for industrial tasks. They include technical components as well as interactions with users. According to the IIC, IIRA is a "business-value-driven and concern-resolution-oriented" reference architecture for the IIoT. Industrial Internet Consortium (2019) IIRA itself is based on the Industrial Internet Architecture Framework (IIAF), which provides basic conventions, principles and definitions. The IIAF builds on the international standard ISO / IEC / IEEE 42010: 2011 and performs basic architectural description constructs, such as Concern, Stakeholders, and Viewpoint. The viewpoints are one of the key building blocks of IIS. There are four viewpoints: Business, Usage, Functional, and Implementation. (Industrial Internet Consortium, 2019)

This paper presents a novel, optimized architecture for model predictive control of a three-tank pilot system in the cloud. Section 2 presents the dynamic models of the three-tank system. Section 3 presents the MPC controller for the three-tank system. Section 4 presents the fault detection with parity equations. The architecture of the integrated system is presented in Section 5. Experimental results are presented in Section 6, followed by conclusions in Section 7.

2. MODELLING OF THE THREE TANK SYSTEM

The three-tank system consists of tanks T_1 , T_3 , and T_2 with the same cross-sectional area A_b , as shown in Fig. 1. These cylindrical tanks are connected in series by a cylindrical pipe with cross-sectional area A_c . Liquid is collected in a reservoir and is pumped back into tanks T_1 and T_2 using pumps 1 and 2 to maintain their levels. All tanks are equipped with a piezo-resistive pressure transducer that measures the liquid level in the tank.

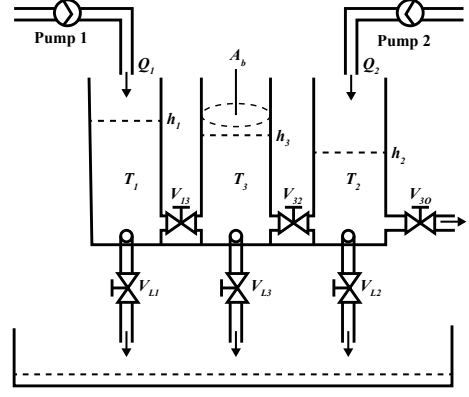


Fig. 1. Three tank system

The flow rates of pumps 1 and 2 are represented by Q_1 and Q_2 , respectively. The flow rate provided by a pump is proportional to the DC voltage applied to its motor.

The tanks are equipped with manually adjustable valves and outlets V_{13} , V_{32} , V_{3O} , V_{L1} , V_{L3} , and V_{L2} to simulate clogs and leaks. In the tested system, valves V_{13} , V_{32} , and V_{3O} were open and leakage valves V_{L1} , V_{L3} , and V_{L2} were closed.

The linearized state-space model parameters are given by Kortela (2022):

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}, B = \begin{bmatrix} \frac{1}{A_b} & 0 \\ 0 & \frac{1}{A_b} \\ 0 & 0 \end{bmatrix} \quad (1)$$

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, D = 0$$

$$\begin{aligned} a_{11} &= -\frac{\alpha_{13}A_c\sqrt{2g}}{2A_b\sqrt{h_{1s}-h_{3s}}}, a_{12} = a_{21} = 0, \\ a_{13} &= a_{31} = \frac{\alpha_{13}A_c\sqrt{2g}}{2A_b\sqrt{h_{1s}-h_{3s}}}, \\ a_{22} &= -\left(\frac{\alpha_{32}A_c\sqrt{2g}}{2A_b\sqrt{h_{3s}-h_{2s}}} + \frac{\alpha_{2O}A_c\sqrt{2g}}{2A_b\sqrt{h_{2s}}}\right), \\ a_{23} &= a_{32} = \frac{\alpha_{32}A_c\sqrt{2g}}{2A_b\sqrt{h_{3s}-h_{2s}}}, \\ a_{33} &= -\left(\frac{\alpha_{13}A_c\sqrt{2g}}{2A_b\sqrt{h_{1s}-h_{3s}}} + \frac{\alpha_{32}A_c\sqrt{2g}}{2A_b\sqrt{h_{3s}-h_{2s}}}\right) \end{aligned} \quad (2)$$

where A is the state matrix, B is the input matrix, C is the output matrix, D is the matrix that describes which inputs affect directly the outputs, $\alpha_{ij} \in [0, 1]$ denotes the outflow coefficient between tank i, j and out from the tank 2, g is the gravity constant, and h_{1s} , h_{2s} and h_{3s} are the operating points of the three levels, respectively.

3. MODEL PREDICTIVE CONTROL FOR THE THREE TANK PILOT SYSTEM

3.1 State-space model based MPC

As a detailed physical model of the three-tank system was available, it was a natural choice to use the linearized version of that model directly with MPC. The inputs to the MPC are the reference values for the two water levels (r)

and the measured process outputs for the levels (y). The outputs of the MPC are the manipulated variables, the speeds of the two water pumps (u). The linear state space system for the MPC is as follows (Maciejowski, 2002):

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + Ed(k) \\ z(k) &= Cx(k) \end{aligned} \quad (3)$$

where x are the states, E is the disturbance matrix and d are the disturbances.

3.2 Regulator

The process is described by the model

$$z(k) = CA^k x(0) + \sum_{j=0}^{k-1} H(k-j)u(j) \quad (4)$$

where $H(k-j)$ are the impulse response coefficients. Using the Equation (4), the regularized l_2 output tracking problem with the input, the input rate of movement, and the output constraints is formulated as

$$\begin{aligned} \min \phi &= \frac{1}{2} \sum_{k=1}^{N_p} \|z(k) - r(k)\|_{Q_z}^2 + \frac{1}{2} \sum_{k=1}^{N_p-1} \|\Delta u(k)\|_S^2 \\ s.t. \quad &x(k+1) = Ax(k) + Bu(k) + Ed(k), \\ &k = 0, 1, \dots, N_p - 1 \\ &z(k) = Cx(k), k = 0, 1, \dots, N_p \\ &u_{\min} \leq u(k) \leq u_{\max}, k = 0, 1, \dots, N_p - 1 \\ &\Delta u_{\min} \leq \Delta u(k) \leq \Delta u_{\max}, k = 0, 1, \dots, N_p - 1 \\ &z_{\min} \leq z(k) \leq z_{\max}, k = 1, 2, \dots, N_p \end{aligned} \quad (5)$$

where $\Delta u(k) = u(k) - u(k-1)$ and N_p is the prediction horizon, r is the future target vector, Q_z is the tracking error weight matrix, and S is the move suppression factor weight matrix. Sizes of the stacked matrices Z , R , U and D depend on the prediction horizon N_p . The predictions by the step response model (4) are expressed as

$$Z = \Phi x_o + \Gamma U + \Gamma_d D. \quad (6)$$

where Φ is the block Hankel matrix, Γ is the pulse response matrix, and Γ_d is the measured disturbance prediction matrix. Then the optimization problem (5) is expressed as

$$\begin{aligned} \psi &= \frac{1}{2} \sum_{k=1}^{N_p} \|z(k) - r(k)\|_{Q_z}^2 + \frac{1}{2} \sum_{k=1}^{N_p-1} \|\Delta u(k)\|_S^2 \\ &= \frac{1}{2} U' H U + g' U + \rho \end{aligned} \quad (7)$$

where

$$H = \Gamma' Q_z \Gamma + H_S \quad (8)$$

$$g = \Gamma' Q_z \Phi x_o - \Gamma' Q_z R + M_{u-1} u_{-1} + \Gamma' Q_z \Gamma_d D \quad (9)$$

$$\rho = (R - \Phi x_o - \Gamma_d D)' Q_z (R - \Phi x_o - \Gamma_d D) \quad (10)$$

H_S is the input weighting matrix, and M_{u-1} is the input weighting vector. The state-space based MPC regulator problem (5) is solved by the solution of the following convex quadratic program

$$\begin{aligned} \min \psi &= \frac{1}{2} U' H U + g' U \\ U_{\min} &\leq U \leq U_{\max} \\ \Delta U_{\min} &\leq \Delta U \leq \Delta U_{\max} \\ \bar{Z}_{\min} &\leq \Gamma U \leq \bar{Z}_{\max} \end{aligned} \quad (11)$$

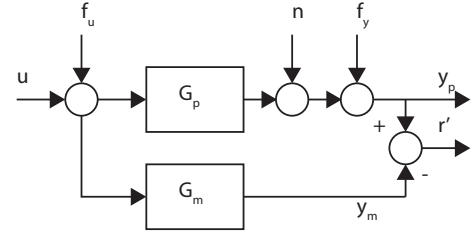


Fig. 2. Output errors

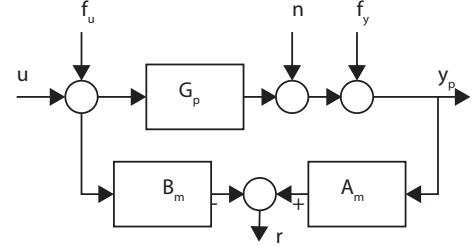


Fig. 3. Polynomial errors

where

$$\bar{Z}_{\min} = Z_{\min} - \Phi x_o - \Gamma_d D \quad (12)$$

$$\bar{Z}_{\max} = Z_{\max} - \Phi x_o - \Gamma_d D \quad (13)$$

4. FAULT DETECTION WITH PARITY EQUATIONS

A way to detect process faults is to compare the process behaviour with a process model describing the nominal non-faulty behaviour. The difference of signals between the process and the model are expressed as residuals. Therefore, the residuals describe discrepancies between the process and the model (Isermann, 2006).

4.1 Parity equations with transfer functions

Figures 2 and 3 illustrate the output and polynomial errors respectively, in the context of linear processes. The process under consideration is characterized by the following transfer function:

$$G_p(s) = \frac{y_p(s)}{u(s)} = \frac{B_p(s)}{A_p(s)} \quad (14)$$

and the process model is:

$$G_m(s) = \frac{y_m(s)}{u(s)} = \frac{B_m(s)}{A_m(s)} \quad (15)$$

This model is considered to be well-established and features fixed parameters known in advance:

$$G_p(s) = G_m(s) + \Delta G_m(s) \quad (16)$$

In this equation, $\Delta G_m(s)$ represents the model errors. For the output error, the residual is calculated as follows:

$$\begin{aligned} r'(s) &= y_p(s) - y_m(s) = y_p(s) - G_m(s)u(s) \\ &= G_p(s)[u(s) + f_u(s)] + n(s) + f_y(s) - G_m(s)u(s) \\ &= \Delta G_m(s)u(s) + G_p(s)f_u(s) + n(s) + f_y(s) \end{aligned} \quad (17)$$

Regarding polynomial error, it leads to:

$$\begin{aligned} r(s) &= A_m(s)y_p(s) - B_m(s)u(s) \\ &= A_m(s)[G_p(s)[u(s) + f_u(s)] + n(s) + f_y(s)] - B_m(s)u(s) \end{aligned} \quad (18)$$

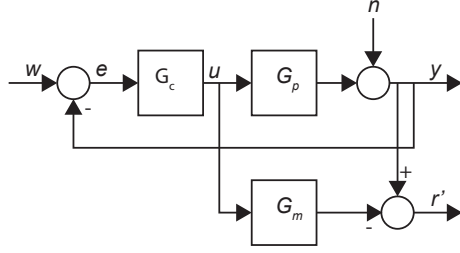


Fig. 4. Fault detection of closed loop with parity equations and output error r'

In an ideal scenario as in this paper where the process and the model are in perfect alignment, the residual simplifies to:

$$r(s) = A_m(s)[f_y(s) + n(s)] - B_m(s)f_u(s) \quad (19)$$

In this context, additive input faults F_u are modulated by the model polynomial $B_m(s)$, while additive output faults f_y are influenced by the polynomial $A_m(s)$. Both these polynomials may lead to the derivation of higher order terms.

Multi-output processes The output residual of a multi-input multi-output (MIMO) process with transfer function matrix $\mathbf{G}_p(s)$ is calculated by

$$\begin{aligned} \mathbf{r}'(s) &= \mathbf{y}_p(s) - \mathbf{y}_m(s) = \mathbf{y}_p(s) - \mathbf{G}_m(s)\mathbf{u}(s) \\ &= \mathbf{G}_p(s)[\mathbf{u}(s) + \mathbf{f}_u(s)] + \mathbf{n}(s) + \mathbf{f}_y(s) - \mathbf{G}_m(s)\mathbf{u}(s) \\ &= \Delta\mathbf{G}_m(s)\mathbf{u}(s) + \mathbf{G}_p(s)\mathbf{f}_u(s) + \mathbf{n}(s) + \mathbf{f}_y(s) \end{aligned} \quad (20)$$

If the process and the model are identical:

$$\mathbf{r}'(s) = \mathbf{G}_p(s)\mathbf{f}_u(s) + \mathbf{n}(s) + \mathbf{f}_y(s) + \mathbf{n}(s) \quad (21)$$

$$\mathbf{A}_p(s)\mathbf{y}_p(s) = \mathbf{B}_p(s)\mathbf{u}(s) \quad (22)$$

$$\mathbf{r}(s) = \mathbf{A}_m(s)[\mathbf{f}_y(s) + \mathbf{n}(s)] - \mathbf{B}_m(s)\mathbf{f}_u(s) \quad (23)$$

$$\mathbf{r}(s) = \mathbf{A}_m(s)[\mathbf{G}_p(s)[\mathbf{u}(s) + \mathbf{f}_u(s)] + \mathbf{n}(s) + \mathbf{f}_y(s)] - \mathbf{B}_m(s)\mathbf{u}(s) \quad (24)$$

and if the process and the model are identical

$$\mathbf{r}(s) = \mathbf{A}_m(s)[\mathbf{f}_y(s) + \mathbf{n}(s)] - \mathbf{B}_m(s)\mathbf{f}_u(s) \quad (25)$$

4.2 Model-based methods for closed-loop supervision

Application of parity equations in closed-loop is considered. As shown in Fig. 4 a residual r is generated by using a fixed

The calculation of the output error is as follows:

$$r'(s) = y_p(s) - y_m(s) = y_p(s) - G_m(s)u(s) \quad (26)$$

$$y_p(s) = G_p(s)u(s) + n(s) \quad (27)$$

$$u(s) = \frac{G_c(s)}{1 + G_c(s)G_p(s)}(w(s) - n(s)) \quad (28)$$

$$\begin{aligned} r'(s) &= (G_p(s) - G_m(s))u(s) + n(s) \\ &= (G_p(s) - G_m(s))\frac{G_c(s)}{1 + G_c(s)G_p(s)}(w(s) - n(s)) \\ &\quad + n(s) \\ &= \frac{G_c(s)(G_p(s) - G_m(s))}{1 + G_c(s)G_p(s)}w(s) \\ &\quad + \frac{1 + G_c(s)G_m(s)}{1 + G_c(s)G_p(s)}n(s) \end{aligned} \quad (29)$$

If the model agree with the real process, $G_p(s) = G(s)$, it holds

$$r'(s) = n(s) \quad (30)$$

Therefore, when there is perfect alignment between the process and the model, both the output and polynomial residuals are solely influenced by disturbances and process faults, similar to the scenario in an open-loop system, as comparison with Eq.(17) and (18) shows.

This means that that the same methodology employed for fault detection using parity equations based on transfer functions in open-loop systems can be effectively applied to linear closed-loop systems. Consequently, in scenarios with minimal disturbances n , it becomes feasible to detect various faults, including but not limited to sensor offsets, elevated Coulomb friction, or backlash in actuators.

5. THE ARCHITECTURE OF THE MODEL PREDICTIVE CONTROL FOR THE THREE TANK PILOT SYSTEM IN THE CLOUD

The experimental setup consists of a cloud with a Java Spring Application and MPC implemented in Java. The comparison of MPC with PID has been done in Kortela (2022). Apache Commons Math 3.6.1 API and oj! Algorithms (ojAlgo) were utilized to implement MPC, as shown in Fig. 5. The gateway PC reads the level measurements and writes the values of the pumps via the NodeJS OPC UA Server and alternatively through the UAGateway OPC UA tunneler/Matrikon OPC UA Tunneler software. It communicates with the cloud through the JSON protocol. The web browser receives the new pump and level values through the IO socket and draws them to the user interface using HTML 5 canvas. The connections of the ABB PM856A PLC and cards are defined in ABB Control Builder M Professional. In addition, the channels of the cards are connected to the related variables, which are defined in the application. The application is then uploaded to the PLC, making the variables available on the OPC server. A RealIO-type AO820 card is used to physically connect two pumps. Level measurements h_1 , h_3 , and h_2 are connected to an AI801 card with a RealIO data type. Twelve channels of a BoolIO-type DO801 card are reserved for six valves, with one channel reserved for each on and off mode. Wireless OPC UA Gateway (Wi-Fi, 5G, 6G) utilized in Digital Twin enables an even faster response time.

6. EXPERIMENTAL RESULTS

6.1 Experimental results of MPC

The MPC was tested on Heroku Cloud with ps:scale web=1, on a cloud server with an Intel Xeon Gold 6248

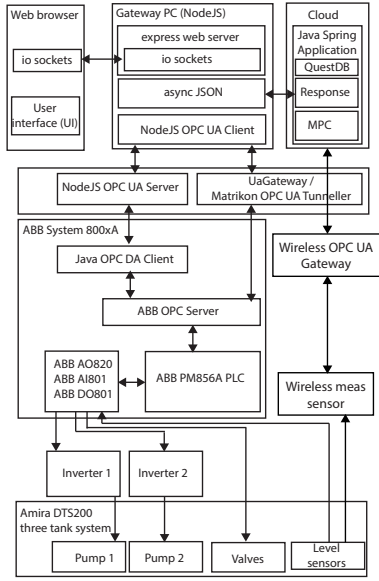


Fig. 5. The architecture of the model predictive control in the cloud

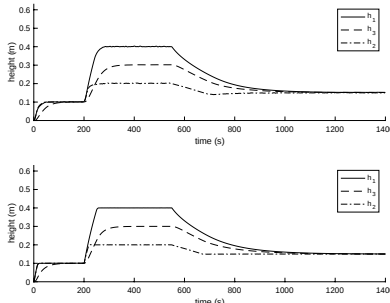


Fig. 6. Levels' response in experimental setup utilizing OPC UA Tunneller (above). Levels' response in experimental setup utilizing NodeJS OPC UA server and OPC DA client (below).

CPU and 16 GB of memory, as well as on a Macbook Pro with an Apple M1 processor and 16 GB of memory. The execution time of MPC on these systems was approximately 100 ms, which was the limiting factor for the speed of the wired system. The UAGateway and ABB OPC Server with their 3 level variables and 2 pump variables limited the control interval to 1 second. The parameters were substituted by The MPC was discretized with a sampling interval of 200 milliseconds when using the NodeJS OPC UA and OPC DA path, and a sampling interval of 1 second

Table 1. Three tank system parameters

Cross section area of the tank (A_b)	0.0154m ²	
Cross section area of the pipes (A_c)	$5 \cdot 10^{-5}$ m ²	
Valve opening position (α_{ij})	$\alpha_{ij} = 0.84$	
Maximum flow rate constraint (Q_{max})	$1.2 \cdot 10^{-4}$ m ³ /s	
Maximum level (h_{max})	0.63m	
Operating point	$Q_1 = 7 \cdot 10^{-5}$ m ³ /s	1
	$Q_2 = 4 \cdot 10^{-5}$ m ³ /s	2
	$h_1 = 0.45$ m	
	$h_2 = 0.25$ m	3
	$h_3 = 0.35$ m	

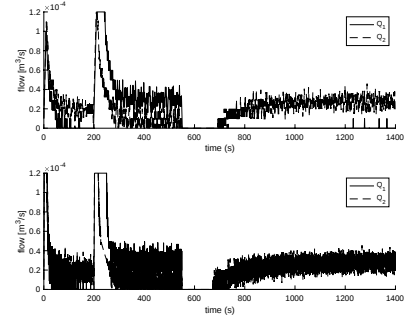


Fig. 7. Input flow rates in experimental setup utilizing NodeJS OPC UA server and OPC DA client (above). Input flow rates in experimental setup utilizing OPC UA Tunneller. (below).

when using the OPC UA Tunneller path. The model used the parameters defined in Table 2.

Fig. 6 and Fig. 7 show the response of the three tank levels and input flow rates using the NodeJS OPC UA and OPC DA path and the path with OPC UA Tunneller. Due to its smaller control interval of 200 milliseconds, the former controller provides a faster response.

6.2 Experimental results of Digital Twin

The mA data for the surface heights h_1 , h_3 and h_2 were calibrated so that they showed the correct millimeter value at the top and bottom of the tank with an accuracy of about a millimeter.

Pump 1 and Pump 2 were calibrated with 1% percent interval to obtain an exact calibrated equation for the pumps.

This was done for both pumps. In addition, with the help of the collected data, the α_{ij} parameters were accurately identified. The validation between digital twin and data is shown in Fig. 8.

Table 2. Accurate model of the faulty pumps

$U_{Signal1}$	0 – 100
$U_{Signal2}$	0 – 100
a_1	-32.41
b_1	1.475
a_2	-33.34
b_2	1.440
U_{Pump1}	$a_1 + b_1 * U_{Signal1}$
U_{Pump2}	$a_2 + b_2 * U_{Signal2}$
$flow1$	$1.0 * 10e - 6 * U_{Pump1} * 0.1$
$flow2$	$1.0 * 10e - 6 * U_{Pump2} * 0.1$

The optimized data model Fig.8, QuestDB database code below and Wireless connection (Wi-Fi, 5G, 6G) (Kortela et al., 2017) enable real-time implementation in the real ABB automation system, which is integrated in the cloud

```

ArrayList<Double> arrayList1 = new
    ArrayList<Double>();
final Connection connection =
    DriverManager.getConnection(
        "jdbc:postgresql://localhost
        :812/qdb", properties);

```

```

1 let dataObject = {
2   data: [
3     // 1 2 3 4 5
4     [0.1,0.2,0.3,0.3,0.3],
5     [0.1,0.2,0.3,0.3,0.3],
6     [0.4,0.45,0.6,0.3,0.3],
7     [0.7,0.8,0.9,0.3,0.3]
8   ],
9   labelsdouble: [
10    {key: "h1"}, //1
11    {key: "h3"}, //2
12    {key: "h2"}, //3
13    {key: "pump1"}, //4
14    {key: "pump2"}, //5
15  ]
16 };
17

```

Fig. 8. Optimized data model

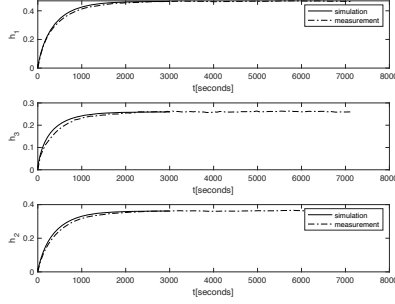


Fig. 9. Accurate calibration of three tank system

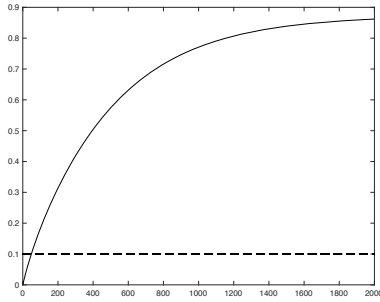


Fig. 10. Fault in closed loop

```

4 try (PreparedStatement preparedStatement
5     = connection.prepareStatement(
6       "SELECT * FROM aimeasurements1;"
7     ) {
8     try (ResultSet rs = preparedStatement
9         .executeQuery()) {
10        while (rs.next()) {
11            double tempvalue5 = new
12                Double(rs.getDouble(5));
13            arrayList1.add(tempvalue5);
14        }
15    }
16 }
17

```

For the test, the electrical error of the pumps that actually happened earlier was recreated for the test on a real three-tank system. Fig. 10 shows that digital twin with parity equations immediately recognizes stepwise and drift-wise changes of parametric faults.

7. CONCLUSIONS

The optimized architecture was presented for the model predictive control of a three-tank pilot system. The op-

timized signal path through the OPC DA was compared to the OPC UA Tunneler implementation by experiments on a real three-tank pilot system with an industrial ABB 800xA automation system. Due to the optimized signal path, the control interval is smaller, resulting in a faster response from the controller and detection of faults. In addition, the Digital Twin with parity equations immediately recognizes stepwise and drift-wise changes of parametric faults.

8. ACKNOWLEDGEMENT

This work has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No. 101057083 – Zero-SWARM. Pumps calibration was implemented by Otacilio Bezerra Leite Neto.

REFERENCES

- DIN (2016). Din spec 91345:2016-04, reference architecture model industrie 4.0 (rami4.0). Technical report.
- European Commission (2017). Digital transformation monitor. Technical report. URL https://ati.ec.europa.eu/sites/default/files/2020-06/DTM_Industrie%204.0_DE.pdf. Accessed 5 December 2022.
- Evensen, L. (2013). Why a reference architecture is important for you. Technical report. URL <https://www.ibm.com/blogs/cloud-computing/2013/07/11/why-a-reference-architecture-is-important-for-you/>. Accessed 5 December 2022.
- IEC (2015). Factory of the future. Technical report. URL <https://www.iec.ch/basecamp/factory-future>. Accessed 5 December 2022.
- IEEE (2022). Systems and software engineering – architecture description. Technical report. URL <http://www.iso-architecture.org/ieee-1471/defining-architecture.html>. Accessed 5 December 2022.
- Industrial Internet Consortium (2019). The industrial internet of things volume g1: Reference architecture. Technical report. URL <https://www.iiconsortium.org/pdf/IIRA-v1.9.pdf>.
- Isermann, R. (2006). *Fault-Diagnosis Systems, An Introduction from Fault Detection to Fault Tolerance*. Springer, Berlin.
- Kortela, J. (2022). Model-predictive control for the three-tank system utilizing an industrial automation system. *ACS Omega*, 7(22), 18605–18611. doi:10.1021/acsomega.2c01275.
- Kortela, J., Nasiri, B., Smirnov, A., Lahnamalampi, A., and Jämsä-Jounela, S.L. (2017). Educational setup for service oriented process automation with 5g testbed. *IFAC-PapersOnLine*, 50(2), 127–132.
- Maciejowski, J.M. (2002). *Predictive Control with Constraints, 1st ed.* Prentice Hal, Harlow.
- VDI Verein Deutscher Ingenieure e.V. (2015). Reference architecture model industrie 4.0 (rami4.0). Technical report. URL https://www.zvei.org/fileadmin/user_upload/Presse_und_Medien/Publikationen/2016/januar/GMA_Status_Report_Reference_Architecture_Model_Industrie_4.0_RAMI_4.0/GMA-Status-Report-RAMI-40-July-2015.pdf. Accessed 5 December 2022.