



## REAL-TIME MODEL PREDICTIVE CONTROL OF AN INDUSTRIAL INTERNET PLATFORM

**Al-Shaimaa A. Younis**

Faculty of Engineering  
Minia University

[elshaimaa.younis@mu.edu.eg](mailto:elshaimaa.younis@mu.edu.eg)

**Ahmed M. Moustafa**

Faculty of Engineering  
Minia University

[ahmed.mahmoud@mu.edu.eg](mailto:ahmed.mahmoud@mu.edu.eg)

**Mohammed Moness**

Faculty of Engineering  
Minia University

[m.moness@mu.edu.eg](mailto:m.moness@mu.edu.eg)

### Abstract

Industrial Internet of Things (I<sup>2</sup>oT) gives more accessibility and reliability of the industrial systems. Thence, most industrial processes have been tended to be controlled and monitored through the Internet. The control design is an important requirement for industrial processes. For a long time, there are two main types of controllers that have shown their effectiveness in the industrial control applications, the Proportional-Integral-Derivative (PID) controller and Model Predictive Control (MPC). PID and MPC are introduced to be designed for the control of the Double-tank System (DTS). Genetic Algorithm (GA) is used for tuning the PID controller with an Integral of the Absolute value of the Error (IAE) performance index. MPC is designed using the state-space model of the system to get more information about the system during the optimization. Simulation results of the controllers are presented to realize the performances of the designed controllers. Node.js frameworks are developed to implement the PID and MPC controllers. Node.js is a runtime environment built on Chrome's V8 engine that executes JavaScript code. Real-time implementation is displayed using the Node.js platform to demonstrate the best performance of the controllers in the real-time control system.

**Keywords**— *Double-tank System (DTS), Industrial Internet of Things, Model Predictive Control (MPC), Receding Horizon, Control Horizon, PID Control, Genetic Algorithm (GA), Nodejs.*

### 1. INTRODUCTION

The concept of the Internet of Things (IoT) depends on converting traditional devices or things to devices connected to the internet. In the industrial field, the factories have exploited the IoT to improve the execution of industrial applications and smart manufacturing after the fourth industrial revolution (Industry 4.0). Industrial Internet of Things (I<sup>2</sup>oT) or that known as industry 4.0 merges the concept of IoT to the Cyber-Physical Systems (CPS). I<sup>2</sup>oT includes machine-to-machine (M2M) communication and different industrial communication technologies with automated applications that called the industrial internet[1]. The industrial internet application needs to be controlled inside the network that has operated in. The industrial internet is made up of sensors,

controllers, actuators that interconnected with each other by local area networks to an edge gateway. These edge gateways are connected to the internet to get the advantages of the IoT[2]. The controller design is the main objective of the industrial internet application. In the industrial systems, the Proportional-Integral-Derivative (PID) controller and Model Predictive Control (MPC) have distinguished in designing more efficient controllers. PID is considered the most popular controller in industrial applications because of its simplicity and capability[3]. MPC is a considerable diffusion technology in the industrial field and many other fields, for control design of complicated multivariable processes. MPC design takes into consideration the system dynamics and limitations [4], this gives the importance of designing MPC in different control systems.

Double-tank System (DTS) is one of the industrial production processes with two-degree-of-freedom (2-DOF). The DTS is addressed by (Astrom and Ostberg) [5] as an educational laboratory. The main objective of this laboratory was an emphasis on the connection between control theory and its application in the real world. This is to understand what it is teaching in automatic control courses and how it could be applied in real applications. The importance of the DTS is not limited in educational experiments but it also has many applications in processing and production industries, such as food processing, beverage, dairy filtration, effluent treatment, pharmaceutical industry, water purification system, industrial chemical processing and Heating, Ventilating and Air Conditioning (HVAC) systems, etc. [6-8]. Also, a chemical reactor is a standard example that controls a liquid level at a specific height to get a constant flow rate of the tank output [9]. From the previous works of controlling the DTS, the PID controller has proved its efficiency in liquid level control. The parameters of the PID controller are adjusted with different tuning methods [8, 10] such as Ziegler-Nichols (Z-N), Cohen-Coon (C-C), Ciancone correlation (CCR), and Pole Placement (PP). The PID also has been tuned by different optimization techniques such as Genetic Algorithm (GA) [11], Harmony Search (HS) optimization algorithm [12], Whale Optimization Algorithm (WOA) [13], Particle Swarm Optimization (PSO) algorithm [14] and Modified Flower Pollination Algorithm (MFPA) [15] to enhance control performance. In [16], fuzzy controllers are used with the PID controller to improve performance. The fuzzy control eliminates greater errors, while PID deals with smaller errors. This concept of dealing with different values of errors using different types of controllers gives the application of hybrid controllers. The hybrid controller of the DTS is addressed in [17] by using mixed controllers that switch between different types of controllers. The PID and different optimal controllers are applied for the implementation of the hybrid controller technique. The Time Minimal Control (TMC) is utilized with the PID controller to generate the hybrid controller in [18]. The model-

based backstepping controller is one of the nonlinear control designs that developed for the state-coupled, two-tank to control the level of the lower tank [6]. Different modern control techniques can be applied such as Model-Free Adaptive Control (MFAC) [15], Model Reference Adaptive Control (MRAC) [19], Artificial Neural Network (ANN) [20] and Model Predictive Control (MPC) [21]. The MPC has been used to control the DTS because of its robustness in many industrial systems using linear or non-linear systems [7, 10, 22].

This paper introduces an industrial internet platform to control the DTS by the MPC in real-time operation. The performance of the MPC is compared to the PID controller which is designed in the previous work of the authors in [23]. The PID controllers for the DTS comprising different configurations such as P, PI, PID controllers that are tuned by various empirical and heuristic methods. The best controller had been chosen here to compare it with the MPC designed in this paper. The main contribution of this paper is to develop and investigate real-time MPC for software platforms applicable in the Internet of Things and industrial internet. The real-time MPC has been designed and implemented on a Galileo embedded board and Node.js as a server for a networked control. This Node.js platform has been used to control the industrial internet DTS.

This paper is formulated as follows. Section 2 describes and illustrates the linear and non-linear mathematical model of the DTS. In Section 3, the PID is designed with GA to tune the parameters of the PID controller. MPC is described and designed in Section 4. Simulation results and real-time implementation are presented in Section 5 and 6, respectively. Finally, conclusions are clarified in Section 7.

## 2. DOUBLE-TANK SYSTEM DESCRIPTION

### A. Double-Tank System Model

The DTS consists of two cascaded tanks, one pump and a basin (reservoir), the two cascaded tanks have a small orifice outlet in the bottom that allows each tank to discharge the liquid through it. As shown in

Fig.1, Tank 1 (the upper tank) is supplied by the pump with varying speed, Tank 1 feeds Tank 2 (the lower tank) while Tank 2 feeds the reservoir at the bottom. There is a liquid level sensor that measures the liquid level in Tank2[5].

The non-linear model of the DTS is defined by the differential equations in Equations 1, 2 and 3, which describe the time rate of change of liquid level of the two tanks and the output equation of the system respectively. The parameters values of the DTS are given in Table 1.

$$\frac{dh_1}{dt} = -a_1 \frac{\sqrt{2gh_1}}{A_1} + \frac{Ku}{A_1} \quad (1)$$

$$\frac{dh_2}{dt} = -a_2 \frac{\sqrt{2gh_2}}{A_2} + a_1 \frac{\sqrt{2gh_1}}{A_2} \quad (2)$$

$$y = h_2 \quad (3)$$

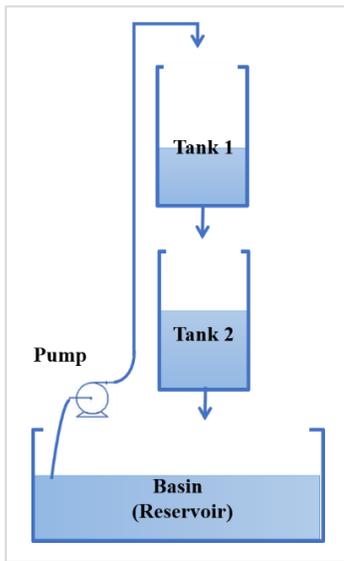


Fig. 1 Double-tank System Diagram

The linear mathematical model of the DTS can be represented by the continuous-time state-space representation in Equations 4 and 5 :

$$\dot{h}(t) = A_c h(t) + B_c u(t) \quad (4)$$

$$y(t) = C_c h(t) \quad (5)$$

where  $u$  is the manipulated variable (Pump speed),  $y$  is the measured variable (tank level),  $h$  is the state variable vector and  $A_c, B_c$  and  $C_c$  are the state  $n \times n$ , input-to-state  $n \times p$  and output-to-state  $m \times n$

respectively of the continuous-time state-space model, where  $n$  is a number of states,  $p$  is a number of inputs and  $m$  is a number of outputs. The final approximation of the linear continuous-time state-space model of the DTS that calculated to an equilibrium point is given in Equations 6 and 7 .

$$\begin{bmatrix} \dot{h}_1 \\ \dot{h}_2 \end{bmatrix} = \begin{bmatrix} -\frac{a_1}{A_1} \sqrt{\frac{g}{2h_1^\circ}} & 0 \\ \frac{a_1}{A_1} \sqrt{\frac{g}{2h_1^\circ}} & -\frac{a_2}{A_2} \sqrt{\frac{g}{2h_2^\circ}} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} + \begin{bmatrix} K \\ 0 \end{bmatrix} u \quad (6)$$

$$y = [0 \quad 1] \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \quad (7)$$

TABLE 1 DOUBLE-TANK SYSTEM PARAMETERS

Parameter	Description	Value
$A_1, A_2$	Tanks cross-sections	$225 \text{ cm}^2$
$a_1$	Orifice cross-section of the upper tank	$0.3933 \text{ cm}^2$
$a_2$	Orifice cross-section of the lower tank	$0.2576 \text{ cm}^2$
$g$	Acceleration of gravity	$980 \text{ cm/s}^2$
$K$	Pump Constant	$185.74 \text{ cm}^3/Vs$
$h_1, h_2$	Limits of the liquid level of the tanks	$0 - 27 \text{ cm}$
$h_1^\circ$	The equilibrium point of the upper tank	$6 \text{ cm}$
$h_2^\circ$	The equilibrium point of the lower tank	$14 \text{ cm}$

### B. IoT Control Platform

IoT has become a backbone of the access control system of the IoT devices. Each device has an IP address that is used to communicate the devices with each other over the network. These devices can be managed, monitored or controlled with a Graphical User Interface (GUI) that is connected to the network[24]. The overall system consists of two main parts: server-side and client-side as illustrated in Fig. 2. The server-side has been developed with the Node.js server that handles the requests with an event-driven handler. This event-driven gives the server scalability and fast response for the real-time controller. The client-side is consisting of the

process-side and the user-side. The process-side is presented in the DTS system that is connected to the embedded Galileo board to give the connection to the network. Also, the calculations of the control signals have been done through it. The user-side that provides the GUI for the user has been developed with HTML, CSS and JavaScript. Bootstrap is also used to give a responsive web application to the user. The user can monitor and control the process-side from the user-side by starting/stopping the process and setting the controller parameters. The server-side is connected to the client-side through the HTTP protocol, which transmits the requests and responses between them. This platform was developed by the authors in previous work for a hybrid controller that has been implemented for a benchmark Quadruple Tank Process (QTP) in [25].

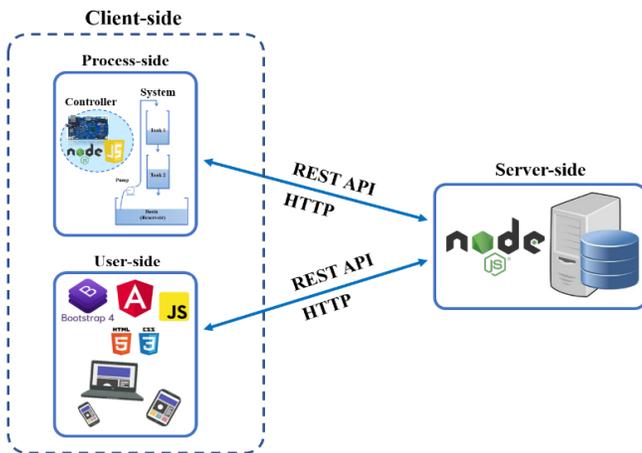


Fig. 2 The Architecture of the Implemented Control IoT Platform

### 3. PID-CONTROLLER DESIGN

The PID controller is one of the most common, simple and efficient control techniques that is used in industrial control systems. More than 95% of the industrial processes controllers are designed using the PID controller, especially the Single-Input Single-Output (SISO) systems and gives a better performance with processes of low to medium order [26]. The PID controller continuously calculates an error value as the difference between

the desired output and the current process variable. The controller tries to minimize the error over time by adjustment of a control variable to a new value determined by a weighted sum as described in the ideal form of the PID controller presented in Equation 8 [27]:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (8)$$

where,  $K_p$ ,  $K_i$  and  $K_d$  are the proportional, integral and derivative gains respectively,  $u$  is the control action and  $e$  is the error between the current and desired output of a controlled system.

The GA algorithm is used to obtain the optimum values of the PID controller parameters to control the DTS [11, 28]. Using GA in designing the PID controller helps to configure the dynamic behavior of the control system. The GA needs little information about the type and the area of the parameters of the controller which are suitable for the system behavior [29]. The idea of GA based on generating new offspring as candidate solutions until the population is completed. The process of minimizing the fitness function or called (cost function) of the GA is executed using crossover, mutation and selection to find the best values of the PID parameters [30]. The fitness function has been determined in this paper, by using the Integral of the Absolute value of the Error (IAE) performance index, which is defined in Equation 9 [31]. All the integral performance indices have been benchmarked and compared to the DTS in [23]. IAE has shown better performance results compared to the ISE and ITAE for tuning the PID controller. The optimization of the PID parameters is done with the linear model of the DTS using the Global Optimization Toolbox on MATLAB R2018a and Simulink [32]. The configurations of the GA parameters are given in Table 2. The optimizations and simulation results are performed using PC with Intel Core i7-7500U @ 2.8 GHz and 8.00 GB RAM.

TABLE 2 CONFIGURATIONS OF GENETIC ALGORITHM PARAMETERS

Option	Configuration
Population Size	50

<b>Number of Generations</b>		60		
<b>Elite count</b>		3		
<b>Crossover Fraction</b>		0.8		
<b>Function Tolerance</b>		1e-06		
-----		$K_p$	$K_i$	$K_d$
<b>Initial populations</b>		3.3	0.06	30
<b>Initial range</b>	Lower bounds	0	0	0
	Upper bounds	10	5	50

$$IAE = \int_0^{\infty} |e(t)| dt \quad (9)$$

**4. MODEL PREDICTIVE CONTROL DESIGN**

MPC strategies have an acquired important role in both theory and applications in control systems. The notion of MPC is depending on analyzing the system dynamics of the open-loop system and describe the relations between the variables of the system, which include manipulated variables (inputs), internal states and measured variables (outputs). An optimal control problem is solved at each sampling time in a receding horizon depending on the current measured or estimated states. The construction of the optimal problem is based on the system dynamics (system model), constraints, weights, initial states and reference signals[4, 33].The designing of MPC is starting by predicting the output of the system from the future control signal inside the prediction horizon or optimization window which is defined as the number of samples  $N_p$ . The future control trajectory has been obtained by the vector of themanipulated variables  $\Delta U$  which the number of its values known as the control horizon  $N_c$ .

According to the historical story of an MPC development, there is a variety of models that have been addressed, such as step-response models, input-output models, state-space models, etc. The state-space model is used in this paper to design the MPC because of the harmony of controller design and system analysis. Using the state-space model in designing MPC gives more system information that makes it easier to predict the further forward in the receding horizon by the current information at the current time. During the past decades, the state-space model has proved its worth in developing MPC on the theory and applications [34].

Although the state-space model is preferred for Linear Time-Invariant (LTI) systems, the accurate results might not be promised. To get accurate results, it should usually scale or normalize the matrices ( $A_c, B_c, C_c$ ) of the system to enhance their numerical conditioning. The normalized continuous-time state-space model shown in Equations 10 and 11 has been transformed into the discrete-time state-space with a sample time  $T_s = 1$ .

$$h(k + 1) = A_d h(k) + B_d u(k) \quad (10)$$

$$y(k) = C_d h(k) \quad (11)$$

where,  $A_d, B_d$  and  $C_d$  are the state, input-to-state and output-to-state matrices respectively of the discrete-time state-space model.The state-space model that is used to design the MPC is called the augmented model, which has been described in Equations 12 and 13 .

$$\bar{h}(k + 1) = A \bar{h}(k) + B \Delta u(k) \quad (12)$$

$$y(k) = C \bar{h}(k) \quad (13)$$

where  $\bar{h}(k + 1), \bar{h}(k)$  are defined as shown in Equation (14 ),  $\Delta u(k)$  is calculated as presented in Equation 15 and the matrices  $A, B$  and  $C$  are calculated by the Equation 16 .

$$\bar{h}(k + 1) = \begin{bmatrix} \Delta h(k + 1) \\ y(k + 1) \end{bmatrix}, \quad \bar{h}(k) = \begin{bmatrix} \Delta h(k) \\ y(k) \end{bmatrix} \quad (14)$$

$$\Delta u(k) = u(k) - u(k - 1) \quad (15)$$

$$A = \begin{bmatrix} A_d & o_d^T \\ C_d A_d & 1 \end{bmatrix}, B = \begin{bmatrix} B_d \\ C_d B_d \end{bmatrix}, C = [o_d \quad 1] \quad (16)$$

where  $o_d = \overbrace{[0 \quad 0 \quad \dots \quad 0]}^n, n$  is the number of states of the system,  $\Delta h(k)$  and  $\Delta h(k + 1)$  are calculated as presented in Equations 17 and 18 respectively.

$$\Delta h(k) = h(k) - h(k - 1) \quad (17)$$

$$\Delta h(k + 1) = h(k + 1) - h(k) \quad (18)$$

The objective of the MPC in most control problems is to minimize the error of the system by carrying the predicted output as near as possible to the set-

point. Also, MPC aims to find the best control value of vector  $U$  to apply the first value of the vector on the system. The cost function that is defined in Equation 19 is combined the two objectives of minimizing the error and taking the interest to find the best  $\Delta U$  when the cost function  $J$  is minimized as possible. The  $\Delta U$  is a vector that contains the future trajectory of control has defined in Equation 20 .

$$J = \min_{1 \leq k \leq N_p} \left( (R - F\bar{h}(k))^T (R - F\bar{h}(k)) - 2\Delta U^T \varphi^T (R - F\bar{h}(k)) + \Delta U^T (\varphi^T \varphi + Q) \Delta U \right) \quad (19)$$

$$\text{s.t.} \quad \bar{h}(k+1) = A\bar{h}(k) + B\Delta u(k)$$

$$\Delta u_{min} \leq \Delta u(k) \leq \Delta u_{max}$$

$$u_{min} \leq u(k) \leq u_{max}$$

$$y_{min} \leq y(k) \leq y_{max}$$

where,  $u_{min} = 0$ ,  $u_{max} = 1$ ,  $\Delta u_{min} = -1$ ,  $\Delta u_{max} = 1$ ,  $y_{min} = 0$  and  $y_{max} = 27$ .

$$\Delta U = \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \Delta u(k+2) \\ \dots \\ \Delta u(k+N_c-1) \end{bmatrix} \quad (20)$$

The necessary condition of the minimum  $J$  is obtained by  $\frac{\partial J}{\partial \Delta U} = 0$ , then, the optimal  $\Delta U$  that will minimize  $J$  is expressed in Equation 21. The predicted output can be calculated as described in Equation 22 using the current state  $\bar{h}(k)$  and the future control movement  $\Delta u(k+i)$  where,  $i = 0, 1, 2, \dots, N_c - 1$ .

$$\Delta U = (\varphi^T \varphi + Q)^{-1} (\varphi^T R - \varphi^T F \bar{h}(k)) \quad (21)$$

$$Y = F \bar{h}(k) + \varphi \Delta U \quad (22)$$

where, the matrix  $(\varphi^T \varphi + Q)^{-1}$  is known as the Hessian matrix in the optimization literature and  $Q, R, F$  and  $\varphi$  are weight matrices in the cost function of predictive control [10].

The implementation of the MPC in real-time is performed by applying the algorithm that is illustrated by the flowchart shown in Fig. 3. The state-space model of DTS is introduced to the

optimizer of MPC and the prediction and control horizon are determined according to the system behavior and its dynamics. Long or short prediction horizon effects on increasing or decreasing the condition of the Hessian matrix. The short prediction and control horizons have generally decreased system stability. If the prediction and control horizons are increased, the stability and the performance of the closed-loop system will be increased. This would increase the execution time and the response might be delayed, or the high-performance hardware will be needed [35]. The prediction horizon has been chosen to meet both objectives of the speed of the execution and the stability of the system. The control horizon should be less than or equal to the prediction horizon. The values of the prediction horizon and control horizon are chosen to meet two of these objectives over several phases of empirical simulations. Therefore,  $N_p = 50$  and  $N_c = 5$  are used in the MPC optimization window.

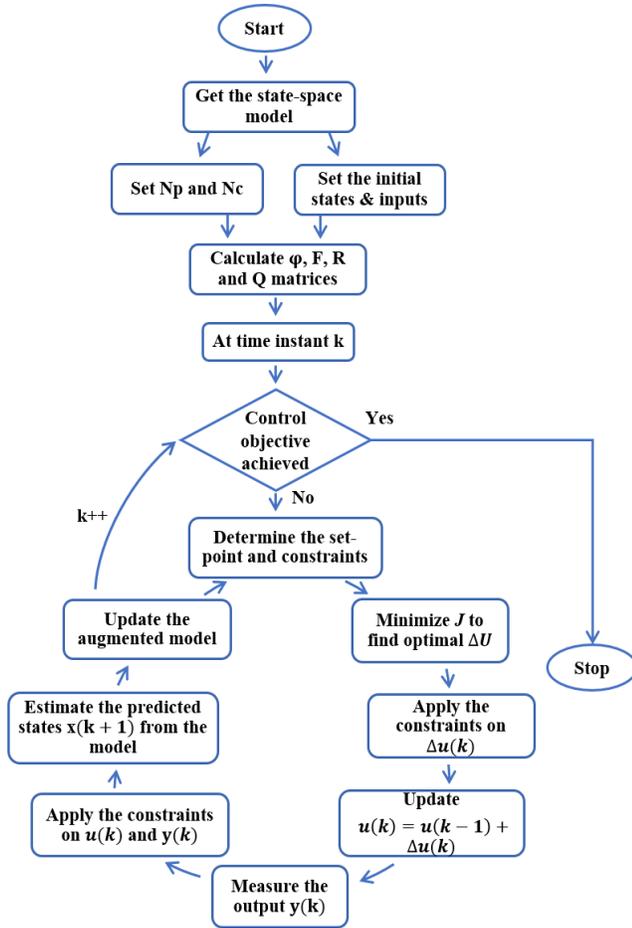


Fig. 3 MPC Algorithm Flow Chart

### 5. SIMULATION RESULTS

The simulation comparison of the PID and MPC is performed on MATLAB Simulink as shown in Fig. 4. The PID controller block is constructed of the three parts of the proportional, integral and derivative of the error, then summed together to get the control signal. The implementation of the MPC is carried out with the Model Predictive Control Toolbox of MATLAB[36], the toolbox gives the advantages of using state-space matrices, limitations of the plant and any delays could be caused by the system. The comparison is done between the two controllers using corresponding timeresponse characteristics and the values of the Integral of the Squared Error (ISE) of the system responses. ISE is used in comparison to get the best minimum values for large errors, especially at the beginning of the

response. These time characteristics are compared for the step responses of the two controllers at 15 cm tank level. The output responses of the simulation of the PID and MPC controllers are shown in Fig. 5. Fig. 6 shows the difference between the control signals generated with the PID and MPC controllers. Table 3 presents the response time characteristics of the PID and MPC controllers. From the results, the MPC gives overshoot smaller than the PID controller with approximately 15% and smaller in settling time and the ISE value. Another two test cases are applied to the system, the first test case starts from 0 cm initial value of the level and steps up to 10 cm then steps up to 20 cm then finally steps down to 15 cm as shown in Fig. 7. The second test case starts from 10 cm initial tank level and steps up to 18 cm then steps down to 7 cm as shown in Fig. 9. The behaviors of the controllers are shown for the two test cases in Fig. 8 and Fig. 10 respectively. All test cases provide that the MPC gives better performance than the PID controller.

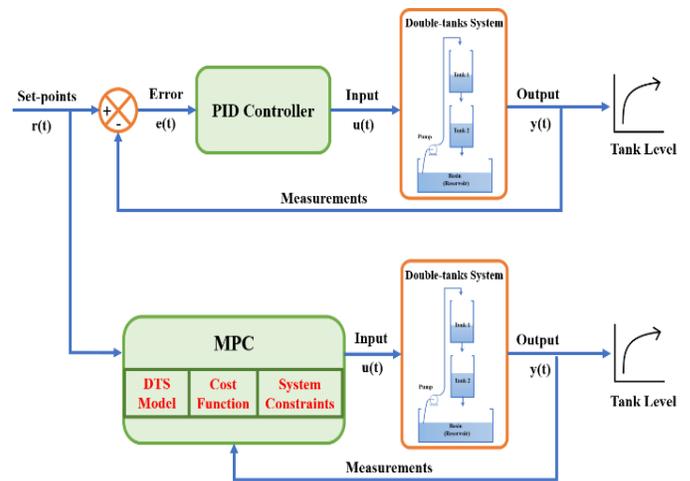


Fig. 4 Simulink Block Diagram of PID and MPC Controllers

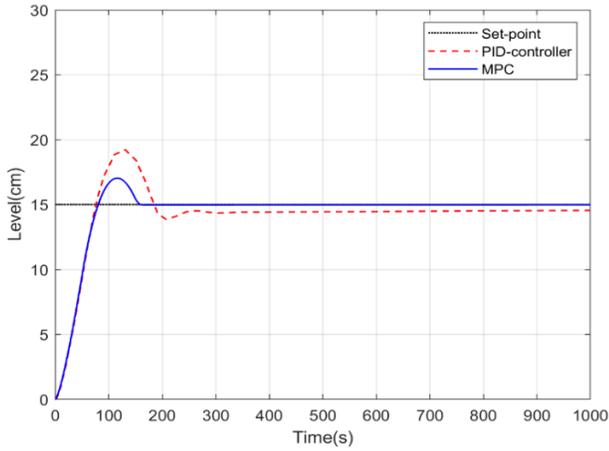


Fig. 5 Simulation of the Step Responses of the PID and MPC Controllers

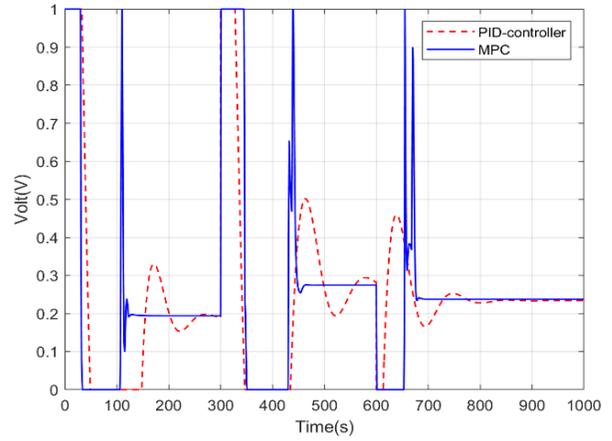


Fig. 8 Simulation Control Signals of the PID and MPC Controllers Test Case 1

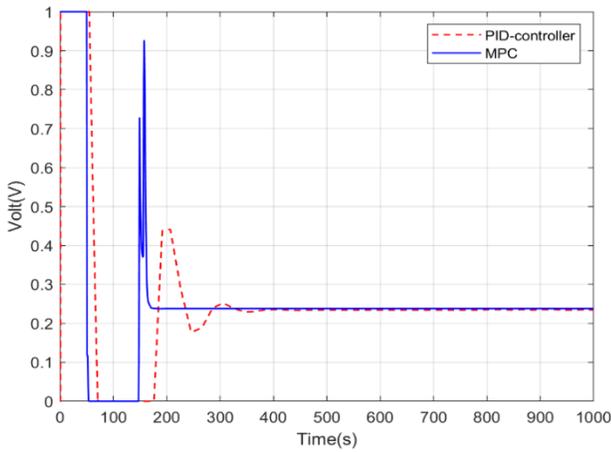


Fig. 6 Simulation Control Signals of the Step Responses of the PID and MPC Controllers

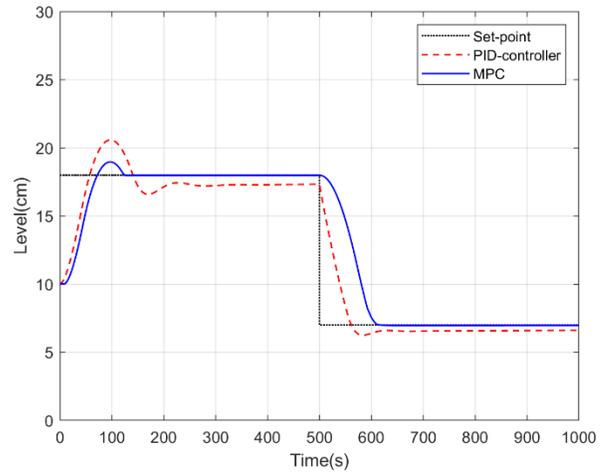


Fig. 9 Simulation Results of the PID and MPC controllers Test Case 2

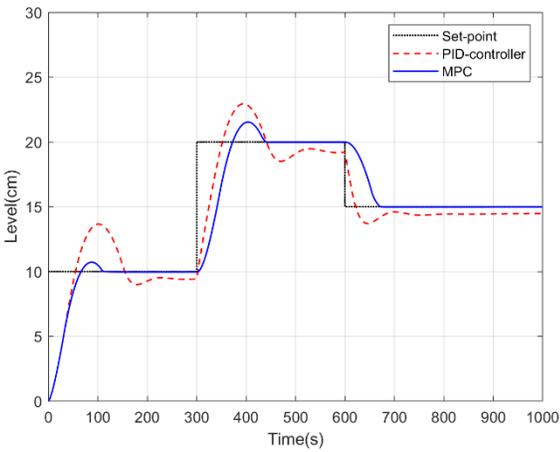


Fig. 7 Simulation Results of the PID and MPC controllers Test Case 1

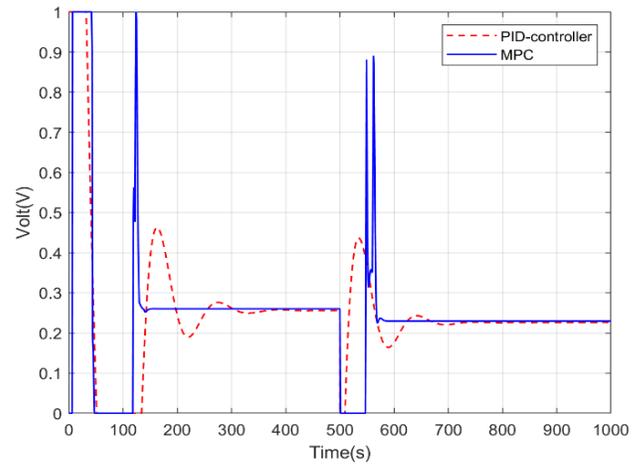


Fig. 10 Simulation Control Signals of the PID and MPC Controllers Test Case 2

TABLE 3 SIMULATION RESULTS OF THE STEP RESPONSES OF THE PID AND MPC CONTROLLERS

Controller	Rising Time (s)	Overshoot (%)	Settling Time (s)	ISE Value
<b>PID</b>	55.00	28.16	755.12	11.15
<b>MPC</b>	56.91	13.60	152.13	9.31

**6. REAL-TIME IMPLEMENTATION AND RESULTS**

The real-time implementation of the industrial internet DTS is illustrated in Fig. 11. The control of the industrial internet platform is built on the Node.js framework using Intel Galileo Development Board. The algorithms of the controllers are developed using JavaScript to perform all mathematical operations of the algorithm and to manage the internal architecture of the Galileo board. The PID control algorithm is generated to calculate the desired control signal according to the instant error and the restrictions of the hardware are considered. The MPC control algorithm is generated to calculate the optimum  $\Delta U$  which gives the minimum objective function of the predictive control. The constraints on the control variable, incremental control variable and output are applied. Frameworks in [37] and [38] have used a MATLAB script to generate the PID and MPC controllers in JavaScript code respectively. The two controllers are performed on the Node.js platform within a network. The measured values

from the plant that is controlled with the PID and MPC are refined by using a Kalman filter to reduce noises from the readings of the level sensor. The Kalman filter is an algorithm that calculates the best estimate of inaccurate data which contains noises then updates the current state of the system continuously[39, 40]. Real-time results are taken place in the Laboratory of Control Systems at Computers and Systems Engineering Department, Faculty of Engineering at Minia University, Egypt. The responses and the control signals of the PID and MPC controllers are shown in Fig. 12 and Fig. 13 respectively. The control signal provided by the MPC has smaller oscillations than the signal provided by the PID which is reduced the effort on the hardware. Fig. 14 shows that the values of  $\Delta u$  of the MPC are in the required limits during the optimization window. From Table 4 that presents the response characteristics of each controller, the MPC proved its efficiency also in the real-time results. This overshoot occurs in real-time due to over-approximation of actuators and sensors in the system. The pump and the level sensor have delay time with nonlinear characteristic curve. To simplify the modeling process for the predictive model, both the pump and the sensor are modelled as a linear gain. This simplifies the implementation of MPC and a linear type at the expense of overshooting at wider step commands.

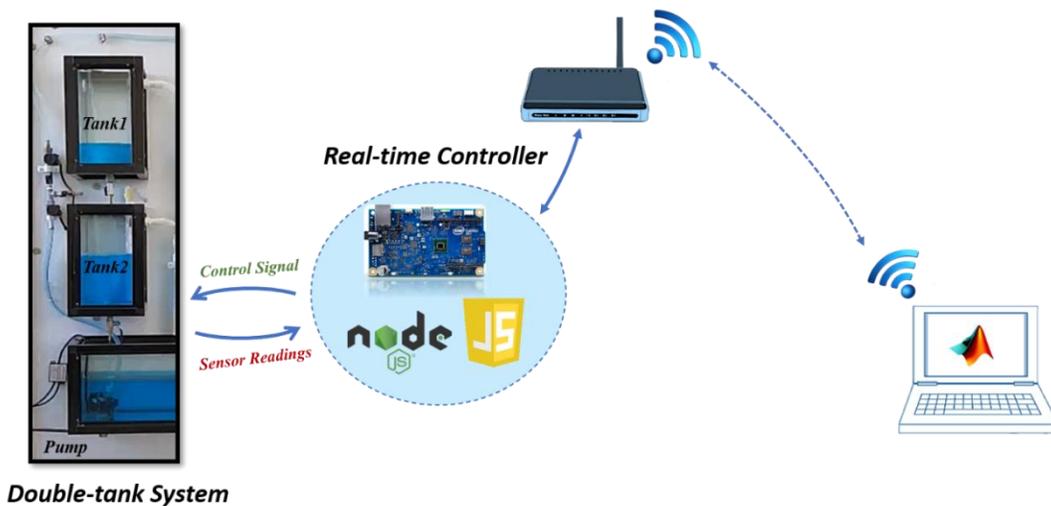


Fig. 11 Real-time implementation of the Industrial Internet DTS

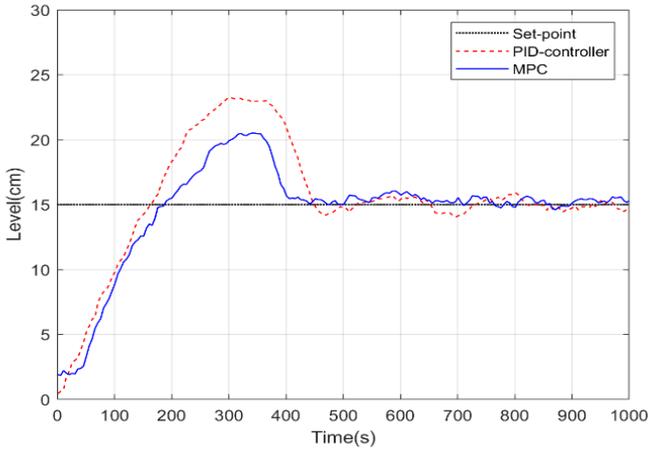


Fig. 12 Real-time Responses of the PID and MPC Controllers

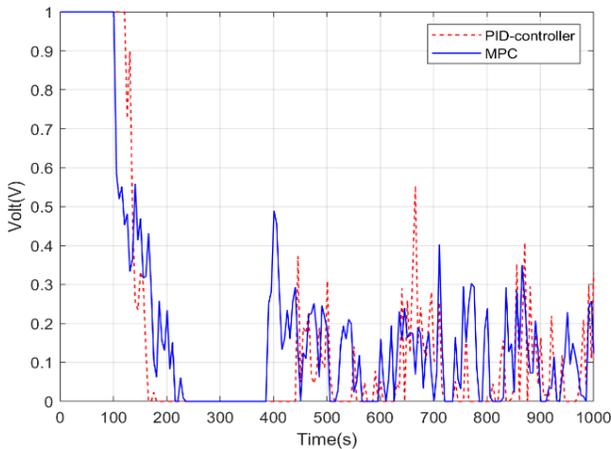


Fig. 13 Real-time Control Signals Applied to the pump with PID and MPC Controllers

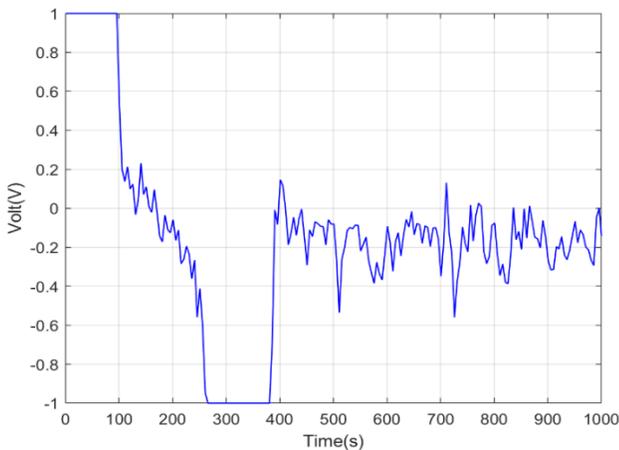


Fig. 14  $\Delta u$  signal of the real-time MPC

TABLE 4 REAL-TIME RESULTS OF THE PID AND MPC CONTROLLERS

Controller	Rising Time (s)	Overshoot (%)	Settling Time (s)	ISE Value
<b>PID</b>	120.84	55.16	1015.3	30.35
<b>MPC</b>	118.37	36.77	1011.3	22.40

## 7. CONCLUSION

This paper introduced a real-time platform built in Node.js for an I<sup>2</sup>oT system that had been controlled with the MPC. The MPC controller was compared with the PID controller to get the best liquid-level regulation of the DTS. Simulation and real-time results were presented to illustrate the improved performance of the submitted controllers. The PID controller was tuned with the GA to get the optimal parameters of the PID controller. The GA was chosen to find the PID parameters because of the double integrating plus dead-time dynamics of the DTS. The IAE performance index was used in calculating the objective function of the GA. The MPC was designed to control the DTS using the state-space model which had given more information to produce the predictive output. Node.js frameworks were developed to implement the PID and MPC controllers of the industrial internet platform. From the experimental results, the MPC provided better performance compared to the PID and had smaller time response characteristics. In the future work, the real-time implementation of the MPC on the industrial internet DTS built on the Node.js platform will be applied to a more complex system. Quadruple Tank Process (QTP) will be used as an extended work of the DTS to investigate the performance of this platform to a MIMO system.

## 8. REFERENCES

- [1] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial Internet of Things: Challenges, Opportunities, and Directions," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4724-4734, 2018, doi: 10.1109/TII.2018.2852491.
- [2] H. Boyes, B. Hallaq, J. Cunningham, and T. Watson, "The industrial internet of things (IIoT): An analysis framework," *Computers in Industry*, vol. 101, pp. 1-12, 2018/10/01/ 2018, doi: <https://doi.org/10.1016/j.compind.2018.04.015>.

- [3] A. Kiam Heong, G. Chong, and L. Yun, "PID control system analysis, design, and technology," *IEEE Transactions on Control Systems Technology*, vol. 13, no. 4, pp. 559-576, 2005.
- [4] A. Bemporad, "Model Predictive Control Design: New Trends and Tools," in *Proceedings of the 45th IEEE Conference on Decision and Control*, 13-15 Dec. 2006 2006, pp. 6678-6683, doi: 10.1109/CDC.2006.377490.
- [5] K. Astrom and A. Ostberg, "A teaching laboratory for process control," *IEEE Control Systems Magazine*, vol. 6, no. 5, pp. 37-42, 1986, doi: 10.1109/MCS.1986.1105142.
- [6] H. Pan, H. Wong, V. Kapila, and M. S. de Queiroz, "Experimental validation of a nonlinear backstepping liquid level controller for a state coupled two tank system," *Control Engineering Practice*, vol. 13, no. 1, pp. 27-40, 2005.
- [7] H. Gouta, S. H. Said, and F. M'sahli, "Predictive and backstepping control of double tank process: A comparative study," *IETE Technical Review*, vol. 33, no. 2, pp. 137-147, 2016.
- [8] A. Tijjani, M. Shehu, and A. Alsabari, "Performance Analysis for Coupled-Tank System Liquid Level Control Using MPC, PI and PI-plus-Feedforward Control Scheme," *J Robotics Autom*, vol. 1, no. 1, pp. 42-53, 2017.
- [9] M. F. Wahyudi and M. Shazri, "Robust anti-windup PID control of a couple industrial tank system," *Proceeding of the International Conference on Mechanical Engineering*. No. 62. 2007, 62 ed., pp. 1-4.
- [10] I. A. Shehu and N. A. Wahab, "Applications of MPC and PI controls for liquid level control in coupled-tank systems," in *2016 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)*, 22-22 Oct. 2016 2016, pp. 119-124, doi: 10.1109/I2CACIS.2016.7885300.
- [11] K. Zhou, B. Yan, Y. Jiang, and J. Huang, "Double-tank liquid level control based on genetic algorithm," in *Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2012 4th International Conference on*, 2012, vol. 2: IEEE, pp. 354-357.
- [12] I. M. Mohamed, "Direct power control of SRG-based WECSs using optimised fractional-order PI controller," *IET Electric Power Applications*, vol. 14, pp. 409-417(8), 2020.
- [13] M. I. Mosaad, "Whale Optimization Algorithms-based PI controllers of STATCOM for Renewable Hybrid Power System," *World Journal of Modelling and Simulation*, vol. 16, no. 1, pp. 26-40, 2020.
- [14] Z. Xiang, D. Ji, H. Zhang, H. Wu, and Y. Li, "A simple PID-based strategy for particle swarm optimization algorithm," *Information Sciences*, vol. 502, pp. 558-574, 2019/10/01/ 2019, doi: <https://doi.org/10.1016/j.ins.2019.06.042>.
- [15] I. M. Mohamed, A. Ahmed, and A.-S. Ahmed, "Enhancing the performance of wind energy conversion systems using unified power flow controller," *IET Generation, Transmission & Distribution*, vol. 14, pp. 1922-1929(7), 2020.
- [16] L. Li, Z. Yong, and L. Ai-Lian, "The application of fuzzy smith-PID controller in coupled-tank liquid-level control system," in *Applied Mechanics and Materials*, 2013, vol. 241: Trans Tech Publ, pp. 1076-1080.
- [17] J. Malmborg and J. Eker, "Hybrid control of a double tank system," in *Control Applications, 1997., Proceedings of the 1997 IEEE International Conference on*, 1997: IEEE, pp. 133-138.
- [18] H. K. Thae and D. Van Hung, "A case study on formal design of hybrid control systems," in *Computer Software and Applications Conference, 2001. COMPSAC 2001. 25th Annual International*, 2001: IEEE, pp. 423-428.
- [19] M. M. Ali, A.-R. Youssef, A. S. Ali, and G. T. Abdel-Jaber, "Variable step size PO MPPT algorithm using model reference adaptive control for optimal power extraction," *International Transactions on Electrical Energy Systems*, vol. 30, no. 1, p. e12151, 2020/01/01 2020, doi: 10.1002/2050-7038.12151.
- [20] S. Gulshan, P. Akhilesh, A. Yogendra, and K. Manoj, "Integrating layered recurrent ANN with robust control strategy for diverse operating conditions of AGC of the power system," *IET Generation, Transmission & Distribution*, 2020.
- [21] F. Salem and M. I. Mosaad, "A Comparison Between MPC and Optimal PID Controllers: Case Studies," *IET Conference Proceedings*, pp. 11 (7 .)-11 (7 .)(1), 2015.
- [22] M. Essahafi, "Model Predictive Control (MPC) Applied To Coupled Tank Liquid Level System," *arXiv preprint arXiv:1404.1498*, 2014.
- [23] A. A. Younis, A. M. Moustafa, and M. Moness, "Experimental Benchmarking of PID Empirical and Heuristic Tuning for Networked Control of Double-tank System," in *2019 15th International Computer Engineering Conference (ICENCO)*, 29-30 Dec. 2019 2019, pp. 162-167.

- [24] R. Carli, G. Cavone, S. Ben Othman, and M. Dotoli, "Iot based architecture for model predictive control of hvac systems in smart buildings," *Sensors*, vol. 20, no. 3, p. 781, 2020.
- [25] M. Moness, A. M. Moustafa, A. H. Muhammad, and A. A. Younis, "Hybrid controller for a software-defined architecture of industrial internet lab-scale process," in *2017 12th International Conference on Computer Engineering and Systems (ICCES)*, 19-20 Dec. 2017 2017, pp. 266-271, doi: 10.1109/ICCES.2017.8275316.
- [26] A. O'Dwyer and W. Scientific, *Handbook of PI and PID Controller Tuning Rules*. Imperial College Press, 2009.
- [27] M. Moness and A. M. Moustafa, "Tuning a digital multivariable controller for a lab-scale helicopter system via simulated annealing and evolutionary algorithms," *Transactions of the Institute of Measurement and Control*, vol. 37, no. 10, pp. 1254-1273, 2015.
- [28] S. M. Nawi, A. N. Abdalla, and M. S. Ramli, "Improved coupled tank liquid levels system based on Hybrid Genetic-Immune adaptive tuning of PI controller," in *International Conference on Electrical, Control and Computer Engineering 2011 (InECCE)*, 21-22 June 2011 2011, pp. 247-252, doi: 10.1109/INECCE.2011.5953885.
- [29] M. Mjahed, "PID controller design using genetic algorithm technique," *Proceedings of the ICICR*, pp. 1-4, 2013.
- [30] O. Kramer, *Genetic algorithm essentials*. Springer, 2017.
- [31] J.-W. Perng, S.-C. Hsieh, L.-S. Ma, and G.-Y. Chen, "Design of robust PI control systems based on sensitivity analysis and genetic algorithms," *Neural Computing and Applications*, vol. 29, no. 4, pp. 913-923, 2018.
- [32] MathWorks. "Global Optimization Toolbox." <https://www.mathworks.com/products/global-optimization.html> (accessed Jul 25, 2020).
- [33] S. J. Qin and T. A. Badgwell, "An overview of industrial model predictive control technology," 1997, vol. 93: New York, NY: American Institute of Chemical Engineers, 1971-c2002., 316 ed., pp. 232-256.
- [34] K. Patan, "Model predictive control," in *Robust and Fault-Tolerant Control*: Springer, 2019, pp. 77-129.
- [35] J. Espinoza, J. Buele, E. X. Castellanos, M. PilatÁSig, P. Ayala, and M. V. GarcÍA, "Real-Time Implementation of Model Predictive Control in a Low-Cost Embedded Device," *Order*, vol. 2, pp. 94-63, 2018.
- [36] MathWorks. "Model Predictive Control Toolbox." <https://www.mathworks.com/products/mpc.html> (accessed Jul 25, 2020).
- [37] Al-Shaimaa A. Younis. "Github Repository: PID Node.js Framework." [https://github.com/AlshaimaaYounis/PID\\_Node.js\\_Framework](https://github.com/AlshaimaaYounis/PID_Node.js_Framework) (accessed Jul 15, 2020).
- [38] Al-Shaimaa A. Younis. "Github Repository: MPC Node.js Framework." [https://github.com/AlshaimaaYounis/MPC\\_Node.js\\_Framework](https://github.com/AlshaimaaYounis/MPC_Node.js_Framework) (accessed Jul 15, 2020).
- [39] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35-45, 1960.
- [40] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry, "Kalman filtering with intermittent observations," *IEEE transactions on Automatic Control*, vol. 49, no. 9, pp. 1453-1464, 2004.

## المتحكم التنبؤي النموذجي لمنصة الانترنت الصناعي في الوقت الحقيقي

### الملخص:

تصدر انترنت الأشياء الصناعي ( $I^2OT$ ) معظم الأنظمة والعمليات الصناعية، والذي أعطى المزيد من إمكانية الوصول والتحكم والمراقبة في تلك العمليات عن طريق الإنترنت. جميع الأنظمة الصناعية تحتاج لأجهزة تحكم تقوم بتنفيذ العمليات الصناعية التي تتم بها بشكل تلقائي ودقيق. خلال عقود طويلة هناك نوعان رئيسيان من وحدات التحكم التي أظهرت فاعليتها في أنظمة التحكم الصناعية، هما وحدة المتحكم النسبي-المتكامل-المشتق (PID) والمتحكم التنبؤي النموذجي (MPC). تقدم هذه الورقة البحثية تصميم وحدات تقوم بالتحكم في نظام الخزان المزدوج والذي يعتبر من أهم العمليات الصناعية لكثرة تطبيقاته واستخداماته في المجال الصناعي. تم استخدام الخوارزمية الجينية (GA) لضبط معاملات وحدة التحكم PID باستخدام مؤشر أداء تكامل الخطأ بين القيمة الحالية للنظام والقيمة المرجعية (IAE). وتم تصميم وحدة التحكم MPC باستخدام نموذج فضاء الحالة والذي يمكن من خلاله الحصول على المزيد من المعلومات حول النظام أثناء عملية البحث عن الحل الأمثل. وتم إجراء عملية المحاكاة لوحدة التحكم وعرض النتائج التي توضح أداء وحدات التحكم على نظام الخزان المزدوج. تم تطوير منصتين باستخدام بيئة Node.js لتنفيذ خوارزميتي وحدتي التحكم PID و MPC وتنفيذ العمليات في الوقت الحقيقي. وتم عرض نتائج التجارب العملية للخزان المزدوج باستخدام كلتا وحدتي التحكم لتحديد الأداء الأفضل بينهما. وأظهرت نتائج استجابة النظام تفوق وحدة التحكم MPC في تحقيق مهمة استقرار النظام عند المستوى المطلوب بكفاءة أعلى.