Real-Time Implementation of Model Predictive Control in a Low-Cost Embedded

John Espinoza
11 PUBLICATIONS **62** CITATIONS
SEE PROFILE

Jorge Buele
Universidad Tecnológica Indoamérica
75 PUBLICATIONS **336** CITATIONS
SEE PROFILE

Marco Pilatasig
Universidad de las Fuerzas Armadas ES¨PE
34 PUBLICATIONS **137** CITATIONS
SEE PROFILE

Marcelo Garcia
Universidad Técnica de Ambato (UTA)
110 PUBLICATIONS **826** CITATIONS
SEE PROFILE

Real-Time Implementation of Model Predictive Control in a Low-Cost Embedded Device

John ESPINOZA
Departamento de Eléctrica y Electrónica, Universidad de las Fuerzas Armadas ESPE
Latacunga, 050104, Ecuador

Jorge BUELE
Facultad de Ingeniería en Sistemas, Electrónica e Industrial, Universidad Técnica de Ambato
Ambato, 180103, Ecuador

Esteban X. CASTELLANOS
Departamento de Eléctrica y Electrónica, Universidad de las Fuerzas Armadas ESPE
Latacunga, 050104, Ecuador

Marco PILATÁSIG
Departamento de Eléctrica y Electrónica, Universidad de las Fuerzas Armadas ESPE
Latacunga, 050104, Ecuador

Paulina AYALA
Facultad de Ingeniería en Sistemas, Electrónica e Industrial, Universidad Técnica de Ambato
Ambato, 180103, Ecuador

Marcelo V. GARCÍA
Departamento de Ingeniería de Sistemas y Automática, University of the Basque Country
Bilbao, 48013, Spain

ABSTRACT

This document presents a general description of the implementation of a low-cost system for flow control. A preconditioning of the signal coming from the process to be controlled is carried out and, through Arduino UNO, data is acquired. Using this data in the embedded board Raspberry Pi 3, a predictive control by model (MPC) has been implemented in real time. The plant model is obtained using the identification tool of the MATLAB software. The design of the controller which are in the latest version of this software, has been developed in Simulink using the new complements made for this board. The experimental results show that despite the technical limitations and the high computational cost that this controller represents, both devices can work together and present high performance, robustness and good response in the presence of disturbances.

Keywords: Predictive control, Embedded device, Process control.

1. INTRODUCTION

In the industry, controllers are highly necessary to manipulate and monitor a large number of variables because they directly influence in production, therefore, controllers have evolved progressively, each time seeking better performance in development and time response. It means that they have traveled from basic controllers, to classic controllers and finally to advanced controllers.

The Model Predictive Controller MPC, was implemented in chemical plants and refineries in the 80s, nowadays it is very applied at an industrial level, but it requires a high computational cost. The MPC due to its predictive capacity is very useful for modular systems, linear and non-linear systems, mono variables and multi variables with an excellent dynamic response [1-4], its use is very varied, so much so that in [2] and [5] its use is proposed in the implementation of level converters with load control and three-phase inverters, in [6] an application is presented in smart networks with multiple electric vehicle charging stations, in [7] its use in an altitude controller for an unmanned four-engined helicopter, in [8] it is used to control the temperature of a refrigeration station and in general in several applications of an industrial nature [9-13].

Due to an MPC can not exist without a computer, developers created computers with embedded low-cost systems and its application has been widely studied: in [14] the Beaglebone is used to perform closed-loop control of the speed of a DC motor, in [15] Raspberry Pi is used for the control of household devices (Internet of Things), in [16] BeagleBoard xM is used for latency reduction in transmissions of live music, in [17] the Beaglebone Black is applied for the navigation control of a robot analyzing the image information, in [18] the use of Arduino UNO with Raspberry Pi for the control of flow using a controller with fuzzy logic is presented, while in [19] it is used to control a wheelchair by moving the iris for paralyzed people. In [20] it is used for monitoring the air quality, and in [21] the combination of Arduino and Raspberry Pi is used for the control of appliances with intelligent communication. All this works developed can demonstrate that embedded systems are currently a computational alternative for controllers and user's applications so even for the implementation of an MPC.

This paper presents the flow control in a didactic station by developing an MPC controller using for data acquisition the Arduino UNO board and for the system implementation the Raspberry Pi 3 board which contains the appropriate embedded system for the aforementioned controller. The tests of performance, robustness and safety against disturbances are demonstrated through dynamic graphs while conclusions about their operation are detailed at the end of this document.

This work is presented as follows: first the Introduction, then the System Identification, later the Controller Design. After that the Hardware Setup is showed for the implementation, then the Real Time Implementation and finally section 6 and 7 for Results and Conclusions respectively.

2. SYSTEM IDENTIFICATION

The schematic diagram of the flow process to be controlled is shown in Figure 1.

Where:
FIT – Flow Indicator Transmitter
FZ – Flow Driver
I/E – Current/Voltage Converter
FI – Flow Indicator
SP – Set point

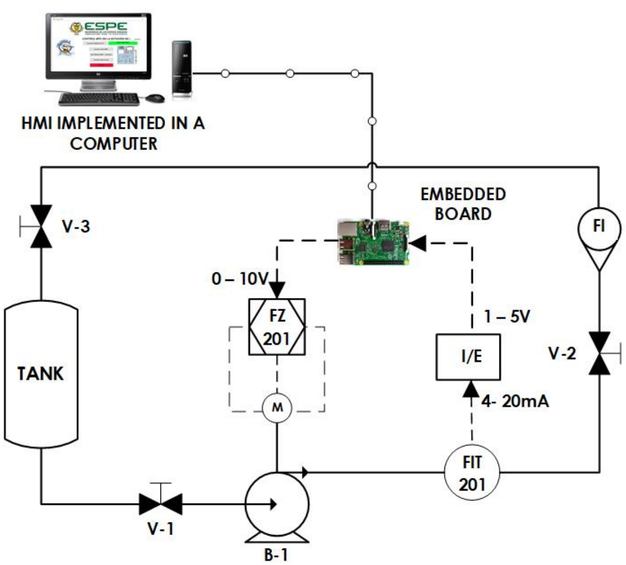


Figure 1. Flow control system

2.1. Process model in state space form

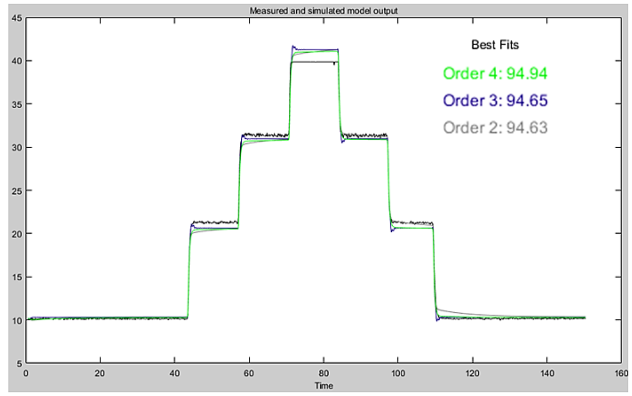
The model of the plant to be controlled is represented in a state space because of its facility to analyze the internal structure of the process, it is represented as shown below:

$$x(k+1) = Ax(k) + Bu(k) \quad (1)$$
$$y(k) = Cx(k) + Du(k) \quad (2)$$

With the MATLAB identification toolbox, it can be obtained the necessary model for the control [21]; the data is imported in domain time specified in the sampling period of 0.108s. In order to get the best result, several samples were taken with models of different order; Table 1 shows a summary of the percentages in each model of the sample. For the application, the one which has the higher percentage of similarity with the open loop test signal used for the validation is chosen. In Figure 2 it can be seen the state matrices and the response of the models obtained.

Table 1 Model comparison table

Validation model	Model	Percentage similarity (%)
Sample	Order 2	94.63
	Order 3	94.65
	Order 4	94.94



$$A = \begin{bmatrix} -1.3118 & 0.8394 & 0.1933 & -2.0197 \\ -2.1835 & -0.1438 & -2.1793 & 9.7886 \\ -0.0540 & 3.2032 & 1.0284 & 11.6407 \\ -5.3352 & -9.0445 & -7.4187 & -19.3374 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.0292 \\ -0.0326 \\ -0.2020 \\ 0.5903 \end{bmatrix}$$

$$C = [-53.7225 \quad 0.4858 \quad -1.8423 \quad 0.4425] \qquad D = [0]$$

Figure 2. Similarity percentages of the obtained models and state matrices.

3. CONTROLLER DESIGN

The control algorithm of the MPC incorporates a model of the process to be controlled to handle all the singularities. In addition, it takes into account the future behavior of the plant and incorporates input and output restrictions that avoid control violations. This type of control solves optimization problems at each instant of time through three essential elements:

- Prediction Model.
- Objective or Cost Function.
- Control Law.

The process model is one of the necessary requirements for the MPC. There are different control strategies to obtain the model that allows the representation of the relationship between the input with the measurable output; in practice and for the reasons mentioned in section 2, the state-space method shown in equations (1) and (2) is used.

Every system has its restrictions on input or output, these can be physical in the case of actuators or safety given by the work limits of the process, equations (3), (4) and (5) represents these restrictions:

$$y_{min} \leq y \leq y_{max} \quad (3)$$
$$u_{min} \leq u \leq u_{max} \quad (4)$$
$$\Delta u_{min} \leq u \leq \Delta u_{max} \quad (5)$$

The objective or cost function is used to optimize the MPC controller. In equation (6) this function is represented which is associated with the evolution of the system through the prediction horizon.

$$J = \sum_{k=0}^{Np} (\hat{y} - r)^T \, Q(\hat{y} - r) + \sum_{k=0}^{Np} \Delta u^T \, R \Delta u \qquad (6)$$

Where, *Np* is the prediction horizon, *k* is every instant of time, $\hat{y}$ is the future output, *r* is the reference, *Q* is the weight matrix of the error in the output, Δu is the predicted change in the control variable and *R* is the weight matrix of control.

In Figure 3 it can be seen the block diagram of the general structure that has a predictive control based in the model.

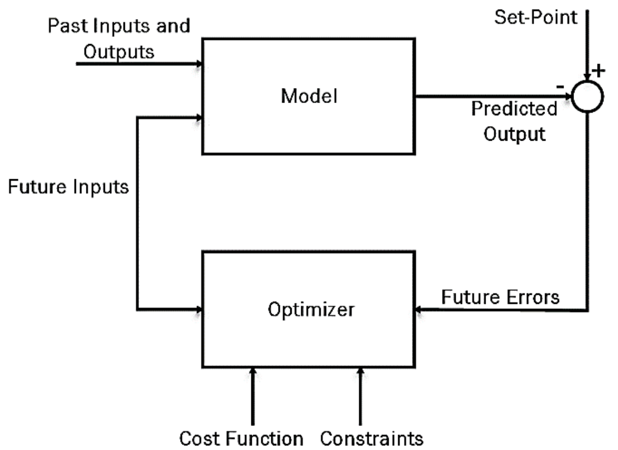


Figure 3. Block Diagram of the general structure of the predictive controllers

4. HARDWARE SETUP

The flow station is available inside the control process and industrial networks laboratory of the Universidad de las Fuerzas Armadas ESPE. It specifications are shown in Table 2.

The process begins with a cylindrical metal tank with a capacity of 25 gallons, coupled to a galvanized steel pipe of ¾ (since the transmitter requires it). The passage of fluid from the tank to a THEBE centrifugal pump of ½ Hp is carried out by a manual bypass valve. The primary element is a Rosemount magnetic sensor of the 8700 series, which is in contact with the process variable. It is connected to a Rosemount 8732E four-wire flow indicator transmitter, energized with 24VDC that produces a standard current signal at its output. This signal oscillates in a 4-20mA loop and acts as an input to the controller.

Table 2 Flow process station specifications

Part name	**Function details**
Process tank (capacity)	25 gal
Rotameter (range)	10 – 40 lpm
I/E converter	4 - 20 mA input, 0 - 5 V output
Flow Indicator Transmitter	Type four wires, 0– 30 ft/s input, 4–20 mA DC output, operating voltage 12 - 42 V DC
Flow Driver	0 – 10 V input, 3-Phase output, operating voltage 230 V 3-Phase.

For the subsequent acquisition of this signal it is required to convert it to a voltage range of 1 to 5V, through a simple converter circuit (I/E), which consists of a resistance of 250Ω placed in series with the process. Additionally, a low pass filter is added to eliminate signals of high frequency and a capacitor in the power supply.

The platform used for data acquisition is Arduino Uno R3. An open source platform that is based on the ATmega328P microcontroller, which has 32 KB of flash memory (with 0.5 KB used for the boot manager) [23]. It has 14 digital I/O pins, 6 of them are used as a PWM signal and 6 for analog input pins. Its software consists of a standard programming language and a firmware that runs on the board, it is programmed using a simplified C++ language in a development environment called IDE [24].

Arduino incorporates in its hardware an internal analog/digital converter (ADC) that operates between 0 - 5V with a resolution of 10 bits. Taking this in advantage, the oscillation between 1-5V of the signal coming from the station is assigned to numerical values between 0 - 1023. Through an escalation process these values are reassigned to another numerical range between 0 - 255, which is transmitted through communication I^2C to the Raspberry Pi 3 board for the control stage.

After the control stage and since Raspberry Pi in all its versions does not have a digital/analog converter (DAC), the integrated circuit (C.I.) PCF8591 is used, which receives data through I^2C communication and operates at 8 bits of resolution. It should be taken into account that this C.I. provides a maximum voltage of 5V and the variable frequency drive of the process operates between 0 - 10V. Therefore, an amplifier circuit with gain 2 and an impedance matching stage is implemented by the C.I. LF353. To perform all the conditioning discussed in this section, an electronic board has been made with conventional elements and all the C.I.s mentioned, the design is shown in Figure 4.

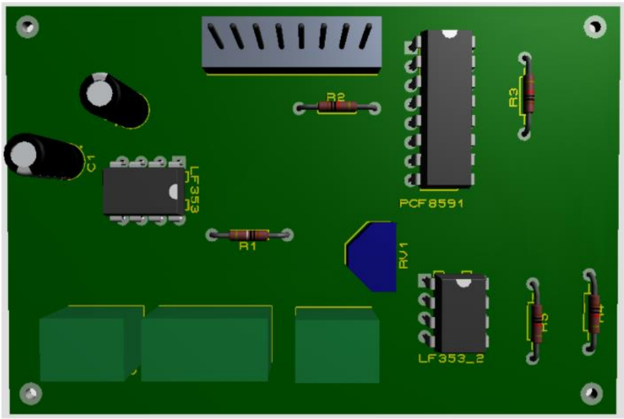


Figure 4. Design of the electronic board

5. REAL TIME IMPLEMENTATION

The implementation of the MPC controller is done with the predictive control toolbox of MATLAB, the toolbox empowers the use of state matrices, specified restrictions by the plant and even delays that allow the emulation of the performance of the controller in a more real way.

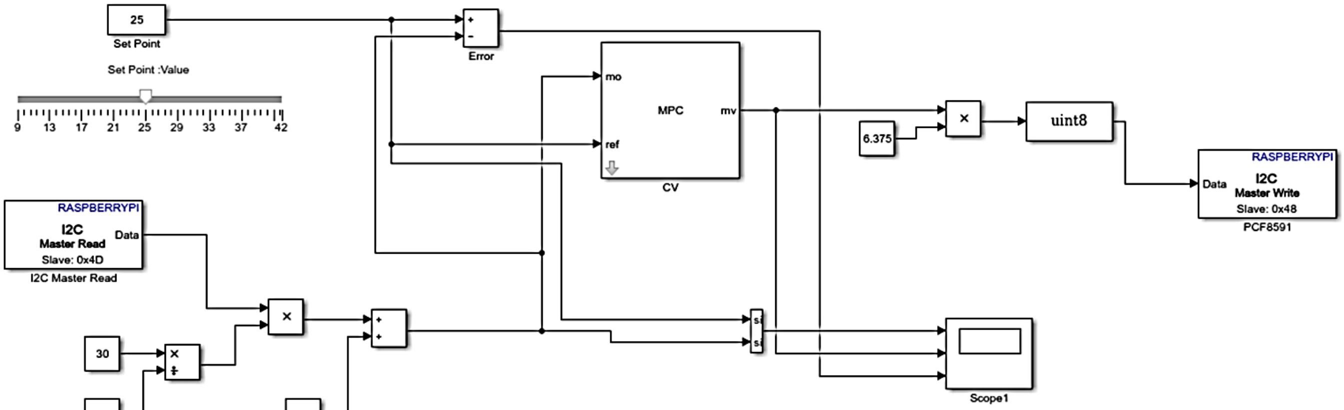


Figure 5. Block diagram of a MPC controller

Figure 5 shows the block diagram designed in Simulink, this design consists of three stages: data acquisition, control and controller output. In the first stage, Arduino UNO R3 is used as a data acquisition board and this data is sent through the I^{2}C port to Raspberry Pi 3. In the second stage, the relevant conditioning is carried out and the parameters that are necessary to obtain the desired response of the controller are defined. In the final block, the respective scaling is performed by the technical requirements of the integrated circuit used, for which it is operated with data type uint8 (0-255).

The simulations performed at the time of designing the controller provide the following information:

- A small prediction horizon causes a slow response of the controller to a set point change.
- A large prediction horizon, causes the controller to act faster in the presence of changes, i.e., the controller's prediction capacity increases.
- A large control horizon causes the response of the controller to be very aggressive and therefore overshooted; what does not happen when opting for a small control horizon.
- The lack of weights at the input and output, causes a total loss of control.

The whole setup for the experimental validation is shown in Fig. 6.

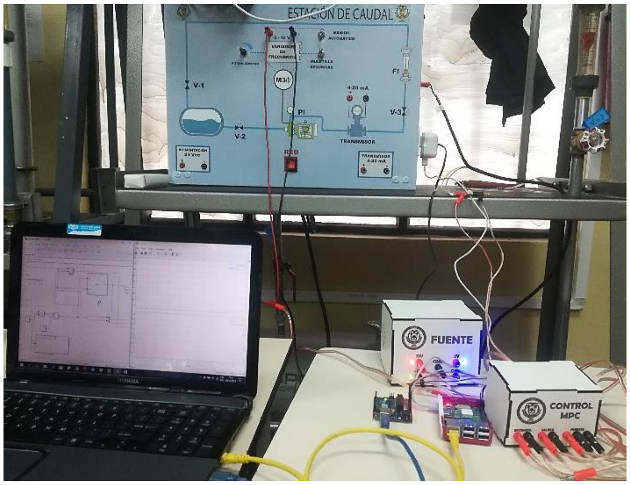


Figure 4. Complete setup of the flow control station.

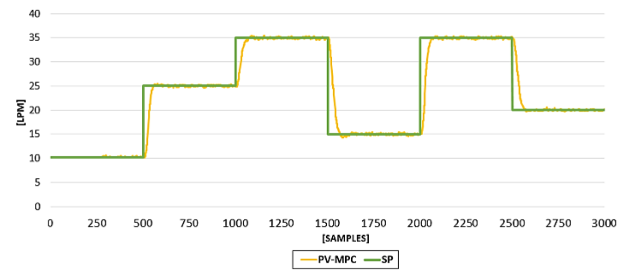
6. RESULTS

6.1. Controller performance

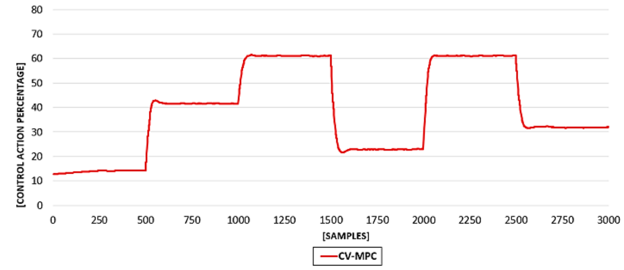
Figure 7 shows the response curves produced by the process variables when the control algorithm is implemented in the embedded board. The process starts at the lower limit, thus means 10 LPM, and then modifies its value to 25, 35, 15, 35 and 20 LPM every 20 seconds, respectively. This curve evidence that the response of the controller is fast; the percentage of the control action without the presence of oscillations is less than 62%, which lengthens the useful life of the final control element. The temporal and over-oscillation characteristic values of the system are presented in Table 3.

Table 3 Performance in different Setpoint changes

Indicators	10-25	25-35	35-15	15-35	35-20
Dead time (s)	0.28	0.24	0.28	0.16	0.12
Rise time (s)	1.08	1.40	1.44	1.36	1.32
Settling time (s)	1.92	2.12	2.36	2.10	2
Overshoot (%)	3.76	1.91	2.53	0.37	1.17



a)



b)

Figure 7. a) Response curves of PV vs. SP. b) CV response curves.

6.2. Controller performance with disturbance

Figure 8 shows the response curves when there is a disturbance in the process (opening of V-3), where the selected disturbance percentages are 69.3% and 77% with a SP = 25 LPM. If the disturbance is greater than 77% despite the fact that the control action is 100%, the process variable does not reach the expected value, which could cause damage to the station. Despite applying the maximum possible percentage, the controller could eliminate subsequent effects of this application and the flow was quickly reestablished to the setpoint without presenting an error in steady state.

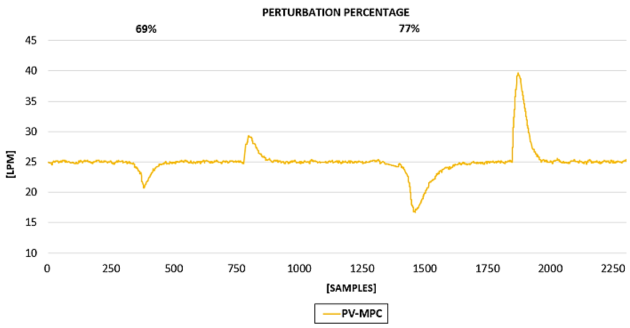


Figure 8. Response curve of the controller to disturbances

6.3. Robustness analysis of controller

The robustness of the controller is determined by its ability to override any undesired effect when a mismatch occurs in the process model. In order to support this, a parameter has been changed in the equations of the state space. It produces that the controller offers a stable response despite how this mathematical model has been modified. In Figure 9, the response of the controller is shown when the SP changes from 10 to 30 LPM.

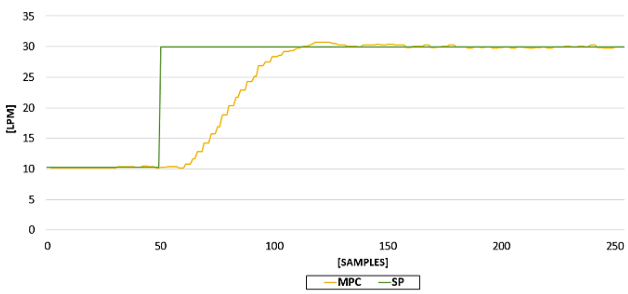


Figure 9. MPC robustness

7. CONCLUSIONS

The main contribution of this work is to corroborate the utility of embedded boards in the implementation of a reduced cost control system that can work in an industrial process. Thus, Arduino UNO was used as a data acquisition board, which working together with the conditioning circuit for the input and output signals of the controller perform the physical part of the system.

As for software, the implementation in real time of a MPC controller has been demonstrated in the microcomputer Raspberry Pi 3 despite the high computational cost that it represents. The response of the controller to sudden changes in SP were fast and with a minimum overshoot percentage. In addition, this controller presents robustness and a well behavior in the presence of disturbance.

For future work, these authors propose the implementation of multiple advanced controllers in this kind of boards for make a comparison between the behavior of each one of them inside an industrial environment.

ACKNOWLEDGMENTS

To the Universidad de las Fuerzas Armadas ESPE extensión Latacunga because of the facilities gave for the development of this work. This research was financed in part by Universidad Técnica de Ambato (UTA) under project CONIN-P-0167-2017 and Government of Ecuador through grant SENESCYT-2013.

8. REFERENCES

[1] C. Petersen, A. Jaunzemis, M. Baldwin, M. Holzinger and I. Kolmanovsky, "Model predictive control and Extended Command Governor for improving robustness of relative motion guidance and control", **Advances in the Astronautical Sciences**, Vol. 152, 2014, pp. 701-718

[2] B. Gutierrez and S. Kwak, "Model Predictive Control with Preselection Technique for Reduced Calculation Burden in Modular Multilevel Converters", **IEEE Transactions on Power Electronics**, Vol. PP, No. 99, 2018, pp. 1-1.

[3] M. Short and F. Abugchem, "A Microcontroller-Based Adaptive Model Predictive Control Platform for Process Control Applications", **Electronics**, Vol. 6, No. 4, 2017, pp. 1-17.

[4] S. Lucia, D. Navarro, Ó. Lucía, P. Zometa and R. Findeisen, "Optimized FPGA Implementation of Model Predictive Control for Embedded Systems Using High-Level Synthesis Tool", **IEEE Transactions on Industrial Informatics**, Vol. 14, No. 1, 2018, pp. 137-145.

[5] C. Fu, J. Lin, Y. Song, Y. Zhou and S. Mu, "Model predictive control of an integrated energy microgrid combining power to heat and hydrogen", **2017 IEEE Conference on Energy Internet and Energy System Integration (EI2)**, Beijing, China, 2017, pp. 1-6.

[6] Y. Shi, H. D. Tuan, A. V. Savkin, T. Q. Duong and H. V. Poor, "Model Predictive Control for Smart Grids with Multiple Electric-Vehicle Charging Stations", **IEEE Transactions on Smart Grid**, Vol. PP, No. 99, pp. 1-1.

[7] L. Jiajin, L. Rui, S. Yingjing and Z. Jianxiao, "Design of attitude controller using explicit model predictive control for an unmanned quadrotor helicopter", **2017 Chinese Automation Congress (CAC)**, Jinan, China, 2017, pp. 2853-2857.

[8] W. Dong, L. Kun and W. Yi, "Model predictive control of a large temperature difference refrigerating station with ice cold thermal energy storage", **2017 Chinese Automation Congress (CAC)**, Jinan, China, 2017, pp. 7340-7345.

[9] C. Lv, X. Chen, D. Yongkao, A. Geng and H. Wang, "Correlation analysis and validation on model predictive control method and SVPWM control method of asynchronous motor", **2017 Chinese Automation Congress (CAC)**, Jinan, China, 2017, pp. 2673-2677.

[10] D. M. Lima, M. V. Americano da Costa and J. E. Normey-Rico, "A flexible low cost embedded system for Model

Predictive Control of industrial processes", **European Control Conference (ECC)**, Zurich, 2013, pp. 1571-1576.

[11] D. Liang, F. Xu, H. Chen and S. Yu, "Support vector machine based model predictive control for engine idle speed control", **2017 Chinese Automation Congress (CAC)**, Jinan, China, 2017, pp. 2772-2777.

[12] M. Aftab, Ch. Chen, Ch. Chau and T. Rahwan, "Automatic HVAC control with real-time occupancy recognition and simulation-guided model predictive control in low-cost embedded system", Energy and Buildings, Vol. 154, 2017, pp. 141-156.

[13] J. Ma, W. Song, S. Wang and X. Feng, "Model Predictive Direct Power Control for Single Phase Three-Level Rectifier at Low Switching Frequency", **IEEE Transactions on Power Electronics**, Vol. 33, No. 2, 2018. pp. 1050-1062.

[14] I. Jaziri, L. Charaabi and K. Jelassi, "A closed Loop DC Motor Control using low cost single-board microcontroller based on embedded Linux", **2016 International Conference on Electrical Sciences and Technologies in Maghreb (CISTEM)**, Marrakech, Morocco, 2016, pp. 1-5.

[15] S. J. Johnston, M. Apetroaie-Cristea, M. Scott and S. J. Cox, "Applicability of commodity, low cost, single board computers for Internet of Things devices", **2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)**, Reston, USA, 2016, pp. 141-146.

[16] L. Gabrielli, M. aBussolotto and S. Squartini, "Reducing the latency in live music transmission with the BeagleBoard xM through resampling", **2014 6th European Embedded Design in Education and Research Conference (EDERC)**, Milano, Italy, 2014, pp. 302-306.

[17] I. Siradjuddin, S. P. Tundung, A. S. Indah and S. Adhisuwignjo, "A real-time Model Based Visual Servoing application for a differential drive mobile robot using Beaglebone Black embedded system", **2015 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS)**, Langkawi, Malaysia, 2015, pp. 186-192.

[18] M. Pilatásig et al., "Implementation of fuzzy controller in low cost embedded boards for a flow system", **2017 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)**, Pucón, Chile, 2017, pp. 1-7.

[19] S. Mathur, B. Subramanian, S. Jain, K. Choudhary and D. R. Prabha, "Human detector and counter using raspberry Pi microcontroller", **2017 Innovations in Power and Advanced Computing Technologies (i-PACT)**, Vellore, India, 2017, pp. 1-7.

[20] S. Kumar and A. Jasuja, "Air quality monitoring system based on IoT using Raspberry Pi", **2017 International Conference on Computing, Communication and Automation (ICCCA)**, Greater Noida, India, 2017, pp. 1341-1346.

[21] P. Kumar and Umesh Chandra Pati, "Arduino and Raspberry Pi based smart communication and control of home appliance system," **2016 Online International Conference on Green Engineering and Technologies (IC-GET)**, Coimbatore, India, 2016, pp. 1-6.

[22] V. Bagyaveereswaran, T. D. Mathur, S. Gupta and P. Arulmozhivarman, "Performance comparison of next generation controller and MPC in real time for a SISO process with low cost DAQ unit", **Alexandria Engineering Journal**, Vol. 55(3), 2016, pp. 2515-2524**.**

[23] Y. A. Badamasi, "The working principle of an Arduino", **2014 11th International Conference on Electronics, Computer and Computation (ICECCO)**, Abuja, Nigeria, 2014, pp. 1-4.

[24] T. Kaur, S. Mahajan, S. Verma, Priyanka and J. Gambhir, "Arduino based low cost active dual axis solar tracker",**2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)**, Delhi, India 2016, pp. 1-5.