Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2015) 23: 641 – 653

doi:10.3906/elk-1302-207

Predictive control of a constrained pressure and level system

Erkan KAPLANOĞLU[1,*], **Taner ARSAN**[2], **Hüseyin Selçuk VAROL**[3]

[1]Department of Mechatronics, Faculty of Technical Education, Marmara University, İstanbul, Turkey
[2]Computer Engineering Department, Faculty of Engineering and Natural Sciences, Kadir Has University, İstanbul, Turkey
[3]Department of Electronics, Faculty of Technical Education, Marmara University, İstanbul, Turkey

Received: 28.02.2013 • **Accepted/Published Online:** 04.04.2013 • **Printed:** 30.04.2015

Abstract: The focus of this paper is the implementation of a constrained predictive control algorithm implemented in Multi-Parametric Toolbox (MPT), which is a free MATLAB toolbox for design, analysis, and implementation of controllers for constrained linear, nonlinear, and hybrid systems. In general, MPT is used for modeling systems offline. The novelty of this study is that real-time mode MPT is used in process control. We also combined the Model Predictive Control Toolbox with MPT. This novel controller is considered a real-time controller of level-pressure systems. In this study, a special type of model predictive control algorithm, the constrained continuous-time generalized control, is used as a controller. The advantages of the controller are illustrated by comparing it to a decoupling PI control.

Key words: Decoupling of systems, level control, predictive control, pressure control, proportional control

1. Introduction

Model Predictive Control (MPC) is a commonly used and well-known method in control engineering. There are a lot of studies available for extending MPC from linear systems to nonlinear systems, and from SISO systems to MIMO systems. Engineers have taken stability, nonlinearity, and computational burden into consideration as the main problems of model predictive control. Some recently proposed nonlinear model predictive control algorithms tried to remove the computational burden and stability problems. Although these methods were applied to very difficult problems [1], such methods have not been completely implemented, and some have not been implemented at all [1,2].

Multi-Parametric Toolbox (MPT) is a tool for multi-parametric optimization that includes optimal control, modeling, analysis, and computational geometry options. The aim of MPT is to provide an efficient computational means of obtaining feedback controllers for linear and piecewise affine (PWA) constrained optimal control problems in a MATLAB programming environment [3]. The source code for MPT was developed for real-time applications in cooperation with the Swiss Federal Institute of Technology (ETH) in Zurich. In this study, firstly, continuous-time two input/two output (TITO) level-pressure control system transfer matrix parameters have been estimated in real time with the least-squares method. Experimental time domain input/output data were utilized in a gray-box modeling approach. Prior knowledge of the form of the system transfer function matrix elements is assumed. Continuous-time system transfer function matrix parameters were estimated in real time [4,6]. Then the MPC was obtained by using MPT, and this controller was applied to the TITO

*Correspondence: ekaplanoglu@marmara.edu.tr

level-pressure control system for real-time control. The strength of our work lies not only in applying a real-time model for predictive control to a TITO level-pressure control system, but also in obtaining a model predictive controller by using MPT. Unlike other toolboxes, efficiency of the code is guaranteed by the extensive library of algorithms from the field of computational geometry and multi-parametric optimization. The toolbox offers a broad spectrum of algorithms compiled in a user-friendly and accessible format, starting from different performance objectives (linear, quadratic, minimum time) to the handling of systems with persistent additive and polytopic uncertainties. Users can add custom constraints, such as polytopic, contraction, or collision avoidance constraints, or create custom objective functions [5].

The rest of this paper is organized as follows. In the next section, the level-pressure control system is described and the details of the system are given. We introduce the MPC with MPT in Section 3, and we give the experimental results of the system in Section 4. We conclude the paper with research contributions in Section 5.

2. Level-pressure control system description

A TITO level-pressure system controlled by MPC is used to control pressure in a tank and the liquid level in a connected container. It is very difficult to control such a system by using classical multivariable control algorithms because of its nonlinear characteristics. The output variables of the system are the liquid level of the container and the liquid pressure of the cylindrical metal tank. The input signals are applied to the pump (DC Motor-U1) for control of the flow rate, and to the proportional valve (U2) for the position of the valve. Both input signals are voltages in the range of 0–10 VDC. The liquid level feedback is obtained from an ultrasonic level sensor (Y1), and pressure feedback is obtained from a pressure-current converter (Y2). The real-time system is shown in Figure 1, and a general diagram of the system is given in Figure 2.

The level-pressure system is a Process Control System from Festo Didactic GmbH, used for experiments in the Mechatronic Education Department control laboratories of Marmara University. In our work, the experimental setup was modified to be a level-pressure system. The aim of this modification is to obtain a constrained TITO system with interaction between inputs and outputs.

MATLAB Realtime Windows Target Toolbox and Data Acquisition card (NI DAQ PCI 6024E) are used for the real-time control of the system. We have used an analog low-pass filter, because the output of an ultrasonic sensor and the output of a pressure-current converter are very noisy. We have selected 0.4 s as the sampling time for the system inertia.

In this study, the model of the system was obtained by the real-time nonparametric experimental method, proposed in [4]. An experimental method was proposed for the modeling of the level-pressure system. A gray-box model along with a curve-fitting approach was used to identify the input-output behavior of the system. Continuous-time system transfer function matrix parameters were estimated in real time. The TITO model of the system is given as:

$$\begin{bmatrix} Y_1(s) \\ Y_2(s) \end{bmatrix} = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \begin{bmatrix} U_1(s) \\ U_2(s) \end{bmatrix} \tag{1}$$

The correlation between inputs and outputs, which is the transfer function G(s), can be written as:

$$G(s) = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} = \begin{bmatrix} \frac{16.97638}{78.74016\,s+1} & \frac{4.30343}{24.09639\,s+1} \\ \frac{-57.1756s+0.98}{141.2236s^2+106.193s+1} & \frac{9.995875}{2.426595\,s+1} \end{bmatrix} \tag{2}$$

We obtained the state-space equations of the system for using the controller, which works with MPT. Due to

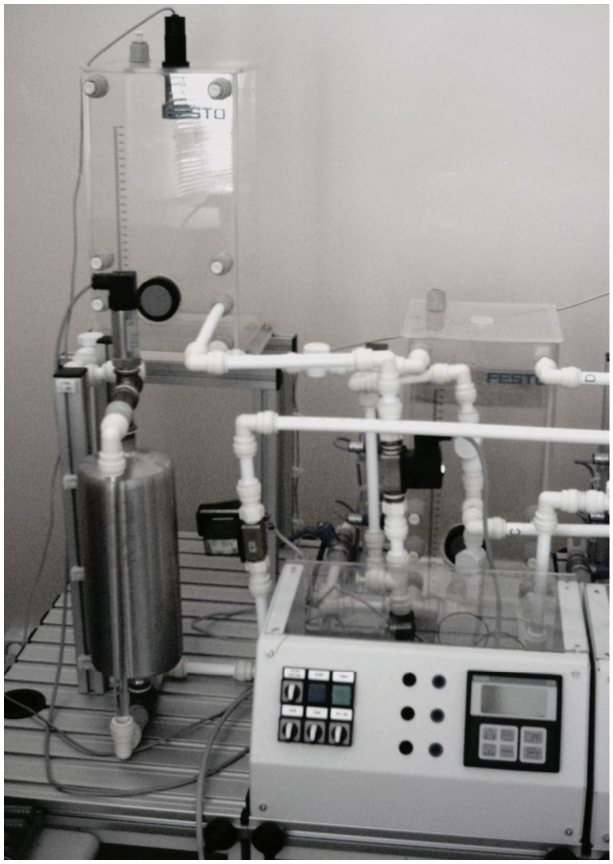

Figure 1. Real-time level-pressure system.

the fact that the level-pressure system has 2 inputs and 2 outputs, the system has 5-state variables.

$$A = \begin{bmatrix} -0.0127 & 0 & 0 & 0 & 0 \\ 0 & -0.7520 & -0.1133 & 0 & 0 \\ 0 & 0.0625 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.0415 & 0 \\ 0 & 0 & 0 & 0 & -0.4121 \end{bmatrix} B = \begin{bmatrix} 0.5 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0.5 \\ 0 & 2 \end{bmatrix}$$

$$C = \begin{bmatrix} 0.4312 & 0 & 0 & 0.3576 & 0 \\ 0 & -0.4049 & 0.111 & 0 & 2.060 \end{bmatrix} D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{3}$$

One of the most important factors, common to all process control applications, is the correct (best) pairing of the manipulated and controlled variables. A number of quantitative techniques are available to assist in the selection process. One of the earliest methods proposed was the Relative Gain Array (RGA) [7]. The original technique is based upon the open-loop steady-state gains of the process and is relatively simple to interpret. The RGA can be expressed as

$$G(0) = \begin{bmatrix} \lambda & 1-\lambda \\ 1-\lambda & \lambda \end{bmatrix} \lambda = \frac{\gamma_1\gamma_2}{(\gamma_1+\gamma_2-1)} \tag{4}$$

The relative gain can be interpreted as follows:

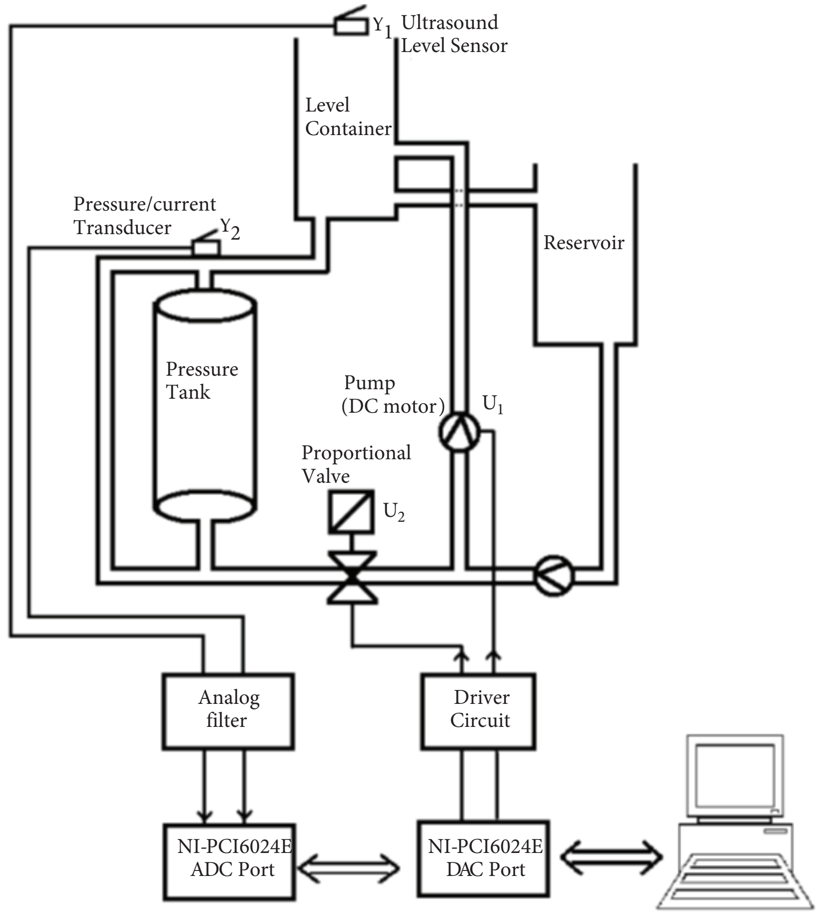

Figure 2. General diagram of a level-pressure system.

1. $\lambda_{ij} = 1$. There is no interaction with other control loops.

2. $\lambda_{ij} = 0$. Manipulated input i, has no affect on output j.

3. $\lambda_{ij} = 0.5$. There is a high degree of interaction. The other control loops have the same effect on output j as the manipulated input i.

4. $0.5 < \lambda_{ij} < 1$. There is interaction between the control loops. However, this would be the preferred pairing as it would minimize interactions.

5. $\lambda_{ij} > 1$. The interaction reduces the effect gain of the control loop. Higher controller gains are required.

6. $\lambda_{ij} > 10$. The pairing of variables with large RGA elements is undesirable. It can indicate a system sensitive to small variations in gain and possible problems applying model-based control techniques.

7. $\lambda_{ij} < 0$. Care must be taken with negative RGA elements. Negative off-diagonal elements indicate that closing the loop will change the sign of the effective gain.

The RGA was calculated using G(s) transfer function matrix via MATLAB. The RGA matrix that describes the input–output interaction of the system is shown as

$$RGA_{system} = \begin{bmatrix} 1.0255 & -0.0255 \\ -0.0255 & 1.0255 \end{bmatrix} \tag{5}$$

It is easy to understand from the elements of the RGA matrix that interaction between the input and output is relatively high, and that it can be reduced by a relative gain constant [7].

3. Design of controllers for a level-pressure system

In this study, we consider both a multivariable decoupling PI controller and the constrained MPC for controlling the level-pressure system. The following sections give all the details for these two controllers.

3.1. Design of multivariable decoupling PI controller

The most commonly used methods for annihilation of loop interaction effects are noninteracting and decoupling control scheme designs. Decouplers (compensation networks) are the key point of these designs for blocking interactions. Decouplers make ongoing parallel control processes disintegrated through separated branches of an independent single-loop subsystem. By ideal or complete design, decouplers let multivariable systems be controlled independently with single loops [8–10].

Figure 3 shows the whole block diagram of the control circuit. The controller is broken down into a decoupling block and a PI controller block. In the first design step, the decoupling block is evaluated for steady-state decoupling. In the second step, the two PI controllers are designed for each of the processes' two decoupled aim paths. The whole controller block can be regarded as a multivariable PI controller with four PI controllers by applying the block transformation technique.

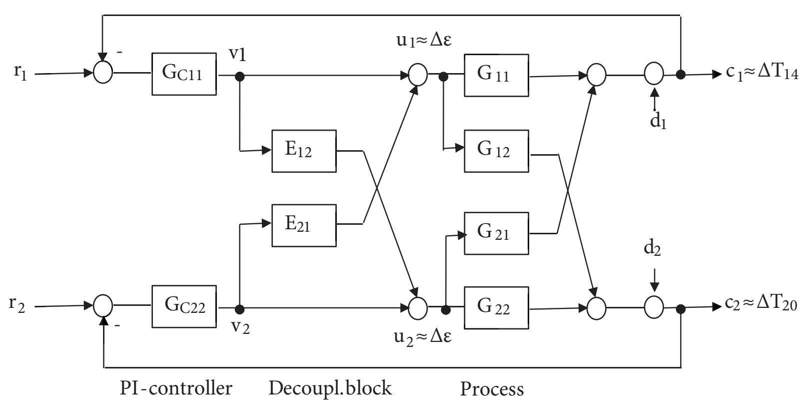

Figure 3. Block diagram of a multivariable control circuit with two inputs and two outputs.

The decoupling block is designed for steady-state decoupling so that there is no influence from the decoupled process input v_1 to output c_2 and from input v_2 to output c_1. From this, through the steady-state conditions for s $\rightarrow$ 0, we obtain

$$\begin{aligned} G_{12}(0) + E_{12}G_{22}(0) = 0 \\ G_{21}(0) + E_{21}G_{11}(0) = 0 \end{aligned} \tag{6}$$

The values of proportional decoupling blocks can be presented as

$$E_{12} = \frac{G_{12}(0)}{G_{22}(0)}, E_{21} = \frac{G_{21}(0)}{G_{11}(0)} \tag{7}$$

The cross-over from Branch II to Branch I is rather small, resulting in a small decoupling coefficient E_{21}.

The input variables of the decoupled process now are v_1 and v_2, which control the two resulting main paths of the process. The transfer functions of the resulting main paths are given as

$$\begin{aligned} G_{1m}(s) &= G_{11}(s) + E_{12}G_{21}(s) \\ G_{2m}(s) &= G_{22}(s) + E_{21}G_{12}(s) \end{aligned} \tag{8}$$

Due to the stationary decoupling, there will be a dynamic crossover between the circuits. The controller design will be carried out neglecting this fact. The final design will prove whether this will be of significance. Of course a dynamic decoupling can be introduced, but in many cases the dynamic decoupling blocks have to be realized by lead-blocks and introduce additional noise into the control loop.

In the second step, the two PI controllers for the main paths of the stationary decoupled process are designed. During the design procedure the PI controller is regarded as a cascade of an I controller and a correcting lead-filter (PD filter):

$$\begin{aligned} G_{c11} &= K_{p1}\left(1 + \tfrac{1}{sT_{r1}}\right) = \tfrac{K_{Ic1}}{s}(1 + sT_{r1}) \\ G_{c22} &= K_{p2}\left(1 + \tfrac{1}{sT_{r2}}\right) = \tfrac{K_{Ic1}}{s}(1 + sT_{r2}) \end{aligned} \tag{9}$$

We used the Ziegler–Nichols step response method for obtaining the values of K_p and K_i [11]. We used MATLAB Windows Target Toolbox for the controller model. The model for the real-time decoupling PI controller is shown in Figure 4.

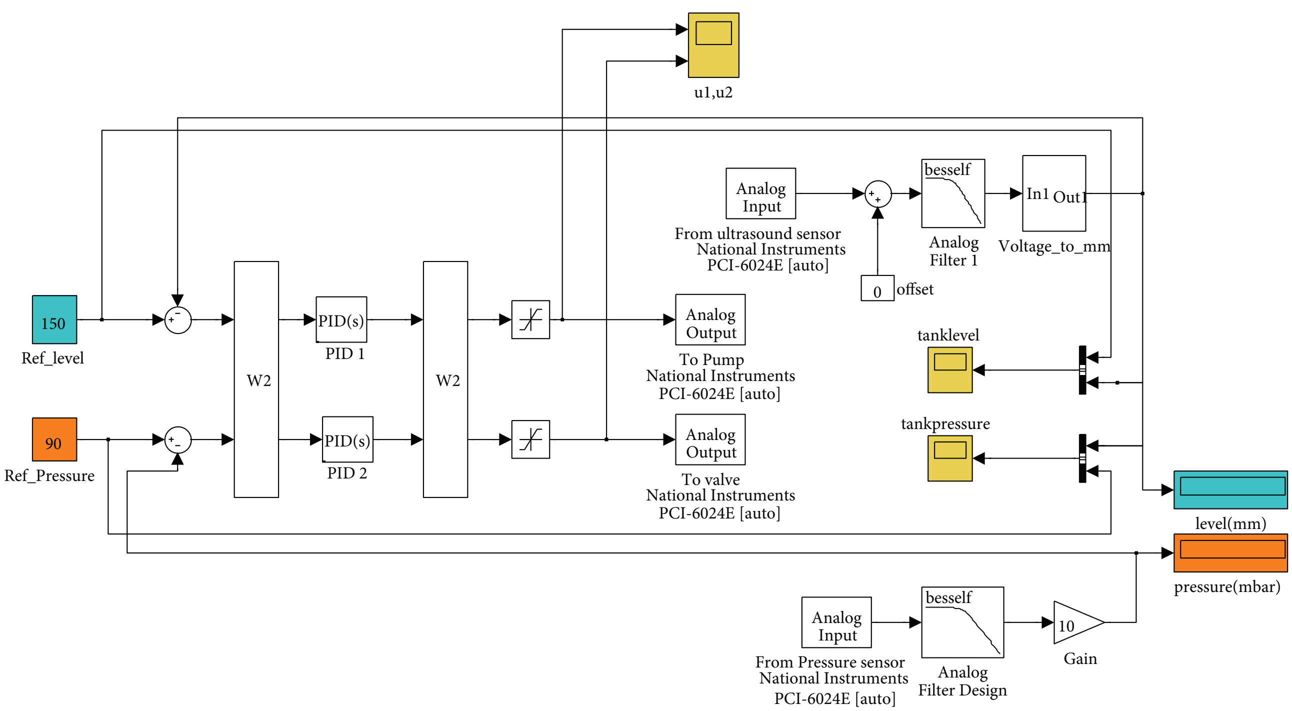

Figure 4. Real-time model of level-pressure system for decoupling PI control.

3.2. Design of the model predictive controller

Several predictive control algorithms are described in control engineering literature [12–17]. In this study, a special type of MPC algorithm, constrained continuous-time generalized control, proposed in [18], is used as a level-pressure controller. In a continuous-time generalized predictive control formulation, output prediction can be obtained by using the expansion of a truncated Taylor series of the system output y. Moving horizon control is necessary for prediction, and it is applied to the system. The main idea is to design the MPC with a moving horizon time frame. The moving time frame is an initial condition for state trajectory at time t. It is obtained from input value and related predicted output values. Within the moving time frame, control constraints are taken into consideration. Nevertheless, this leads to an optimization problem. It can be formulated as the minimization of the performance index function with respect to the control vector.

As described in [16,18], we also consider continuous-time systems in the Laplace domain, that is,

$$Y(s) = \frac{D(s)}{E(s)}U(s) + \frac{O(s)}{E(s)}V(s) \tag{10}$$

In Eq. (10), $D_{(s)}$, $E_{(s)}$, and $O_{(s)}$ are polynomials in the s-domain. $Y_{(s)}$ is the system output, $U_{(s)}$ is the control input, and $V_{(s)}$ is the disturbance input. Here, we describe $O_{(s)}$ as a first order observer polynomial. It is stable and chosen by the designer. As described above, output prediction can be obtained by using an expansion of a truncated Taylor series of the system output y in continuous-time generalized predictive control. The output prediction can be described as

$$\hat{y}(t+T) = \sum_{n=0}^{p} \frac{d^n y(t)}{dt^n}\frac{T^n}{n!}, \tag{11}$$

where p can be defined as the prediction order (order of the highest derivative of the system output y) and T can be expressed as the future time variable. As can be seen in the equation, it is necessary to obtain the output derivatives, but instead of taking derivatives, we use emulated values because of the noise effect of taking derivation. Using these emulated values, we finally obtain output predictor in the time of (t+T) as

$$y^*(t+T) = T_p H u + T_p Y_e, \tag{12}$$

where

$$H \hat{=} \begin{bmatrix} h_0 & 0 & 0 & . & . & 0 \\ h_1 & h_0 & 0 & . & . & 0 \\ h_2. & h_1 & h_0 & . & . & 0 \\ . & . & . & . & . & . \\ . & . & . & . & . & h_0 \\ . & . & . & . & . & . \\ \mathbf{h}_p & \mathbf{h}_{p-\mathbf{1}} & \mathbf{h}_{p-\mathbf{2}} & . & . & \mathbf{h}_{(p-k)} \end{bmatrix} \tag{13}$$

$$T_p = \begin{bmatrix} 1 & T & \frac{T^2}{2!} & \dots & \frac{T^p}{p!} \end{bmatrix} \tag{14}$$

$$u = \begin{bmatrix} u(t) & u_1(t) & . & . & u_k(t) \end{bmatrix}^T \tag{15}$$

$$Y_e = \begin{bmatrix} y_{e0}(t) & y_{e1}(t) & . & . & y_{ek}(t) \end{bmatrix}^T \tag{16}$$

In these equations, H is the Markov parameters matrix of $D_{(s)}/E_{(s)}$. $u_k(t)$ represents the input's kth derivative. $y_{ek}(t)$ is the output emulators of the output's kth derivative. We can describe k as the order of highest derivative of u.

As mentioned above, the aim is to design and implement the MPC with a moving horizon time frame, which is an initial condition for state trajectory at time t. It is obtained from the input value and related predicted output values. Within the moving horizon time frame, control constraints are taken into consideration. Then, an optimization problem occurs. Continuous-time generalized predictive control takes a performance index function into consideration as

$$J = \int_0^{T_h} \left[(y^*(t+T) - w^*(t+T))^2 + \lambda u^*(t+T)^2 \right] dT, \tag{17}$$

where λ is the control weighting and T_h is the prediction horizon. We also consider the truncated Taylor series expansion of the future set point as $w^*(t+T) = T_p w$ and the truncated Taylor series expansion of predicted future input as $u^*(t+T) = T_r u$, where $T_r = \begin{bmatrix} 1 & T & \frac{T^2}{2!} & ... & \frac{T^r}{r\,!} \end{bmatrix}$. The vector w is considered as $w = \begin{bmatrix} w(t) & 0 & ... & 0 \end{bmatrix}^T$, and it contains the set point and derivatives of the set point. The continuous-time generalized predictive controller cost function can be shown as

$$\begin{aligned} J = {} & u^T (H^T (\int_0^{T_h} T_p^T T_p dT)\, H + \lambda \int_0^{T_h} T_r^T T_r dT) u + 2(H^T (\int_0^{T_h} T_p^T T_p dT)(y_e - w))^T \\ & + (y_e - w)^T (\int_0^{T_h} T_p^T T_p dT)(y_e - w) \end{aligned} \tag{18}$$

When $u = -(H^T (\int_0^{T_h} T_p^T T_p dT)\, H + \lambda \int_0^{T_h} T_r^T T_r dT)^{-1} H^T (\int_0^{T_h} T_p^T T_p dT)(y_e - w)$, the performance index function J is the minimum under-constraints free conditions. At this point, it is possible to describe input constraints and output constraints as given in [18,19].

Input Constraints:

Limitations of an actuator may cause control input signals to have magnitude constraints, and also their derivatives [19]. All these constraints can be expressed as:

$$Q_{u_j} u \le p_{u_j} \tag{19}$$

$$p_{u_j} = \begin{bmatrix} u_{j0}^{\max} & -u_{j0}^{\min} & \cdots & u_{jk_j}^{\max} & -u_{jk_j}^{\min} \end{bmatrix}^T \quad Q_{u_j} = \begin{bmatrix} 1 & 0 & . & . & . & 0 \\ -1 & 0 & . & . & . & 0 \\ 0 & 1 & . & . & . & 0 \\ 0 & -1 & . & . & . & 0 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 0 & 0 & . & . & . & 1 \\ 0 & 0 & . & . & . & -1 \end{bmatrix},$$

$u_{jk}^{\min}$ is the minimum limitations and $u_{jk}^{\max}$ is the maximum limitations on the input's kth derivative.

Output Constraints:

Consideration of output constrains may be required in many conditions. As in the level-pressure system, to provide safe working conditions, we must consider that the pressure value is limited to 95 mbar in the tank.

These are the safety limitations for the output constraint. $y_i^{\min}$ is considered as the lower limit and $y_i^{\max}$ is considered as the upper limit. These constraints can be written via predicted output as

$$y_i^{min} \leq y^*(t+T) \leq y_i^{\max}, \quad T_{l_i} \leq T \leq T_{u_i}, \tag{20}$$

where T_{l_i} is the lower constraint horizon and T_{u_i} is the upper constraint horizon. As shown in [19], a single matrix inequality is obtained by:

$$Q_{y_i} u \leq p_{y_i}, \tag{21}$$

where

$$Q_{y_i} = \begin{bmatrix} Q_{y_i}(T_{l_i}) \\ Q_{y_i}(T_{l_i+1}) \\ . \\ . \\ Q_{y_i}(T_{u_i}) \end{bmatrix}, p_{y_i} = \begin{bmatrix} p_{y_i}(T_{l_i}) \\ p_{y_i}(T_{l_i+1}) \\ . \\ . \\ p_{y_i}(T_{u_i}) \end{bmatrix}, \quad Q_{y_i}(T) = \begin{bmatrix} T_p \\ -T_p \end{bmatrix}, p_{y_i}(T) = \begin{bmatrix} y_i^{\max} - T_p Y_{ie} \\ -y_i^{\min} - T_p Y_{ie} \end{bmatrix}$$

It is also possible to see some constraints on the derivative of a system output. These constraints are expressed as

$$Q_{y_i}^l u \leq p_{y_i}^l, \tag{22}$$

where $Q_{y_i}^l$ and $p_{y_i}^l$ are the same as Q_{y_i} and p_{y_i}, respectively, except that T_p, $y_i^{\min}$, and $y_i^{\max}$ are replaced by T_p^l, $y_{il}^{\min}$, and $y_{il}^{\max}$, respectively, by considering the lower and upper limits for the derivatives of the outputs, and $T_p^l = \begin{bmatrix} 0 & 1 & T & \frac{T^2}{2} & \ldots & \frac{T^{p-1}}{(p-1)!} \end{bmatrix}$.

The real-time control system is given in Figure 5. We used MATLAB Windows Target Toolbox for the controller model. In addition to MPT, the controller block includes analog input and output blocks (NI DAQ PCI 6024E), converters that convert voltage (V) to level (mm) and current (mA) to pressure (mbar), a low-pass filter that minimizes the noise level, and reference signal blocks and scopes. The controller parameters can be obtained as an m-file, which has MPC formulation, in MATLAB. This m-file is a preloaded file and related to the MPT toolbox.

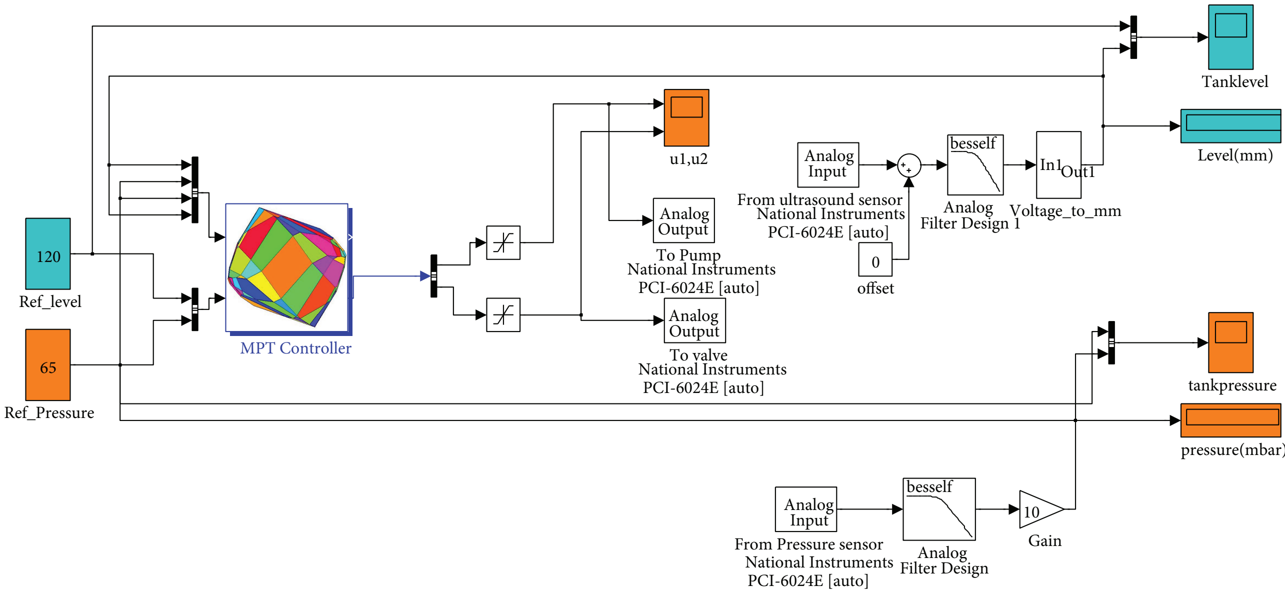

Figure 5. Real-time control schema of level-pressure System in MATLAB with Simulink.

Physical constraints exist in many control problems, especially in process control. Due to actuator limitations, we can see some constraints on inputs. In the same way, we can see some constraints on outputs, due to efficiency considerations, safety limitations, and product quality requirements. Our system also has these types of constraints. Firstly, input voltages, applied to the DC-motor and proportional valve, are selected as input-constrained such that the minimum and maximum values are 0–10 VDC. The level tank measurement values are selected between 15 mm and 200 mm instead of 0–300 mm, because of the placement of the liquid input pipe and the linear working range of the ultrasonic sensor. This is the output constraint of the system. To provide safe working conditions, we must consider that the pressure value is limited to 95 mbar in the tank. These are the safety limitations for the output constraint.

As mentioned in [12], constraints are taken into account in any control problem in two ways. The first is the control problem without considering constraints, and we try to keep the controller parameters within the constraints limits. In this case, if we try to keep the system within the constraint limits after the solution, this may cause a longer rising time and settling time. The second method is for the control problem to be solved subject to constraints. This method looks understandable, and we try to keep system performance in the presence of constraints. However, some control methods may not permit this. The formulation of predictive control methods mostly provides a natural framework for including constraints in the problem. All these constraints are easily applied to the system by means of the m-files in the MATLAB command window.

4. Experimental results

We propose several experiments to show the effectiveness of the MPC in this section. Firstly, we try to understand the input/output interaction of the level-pressure system. For this purpose, we set the level value minimum (15 mm) while the pressure value is 90 mbar as shown in Figure 6. The pressure value minimum is 0 mbar, while the level value is 180 mm, as shown in Figure 7. In the first application, although the chosen level value is 15 mm, the liquid level of the system reaches 40 mm. In the second application, although the chosen pressure value is 0 mbar, the liquid pressure of the system reaches 5 mbar. As shown in the applications, this system has a high interaction between level and pressure. Therefore, the reference values cannot reach the desired values.

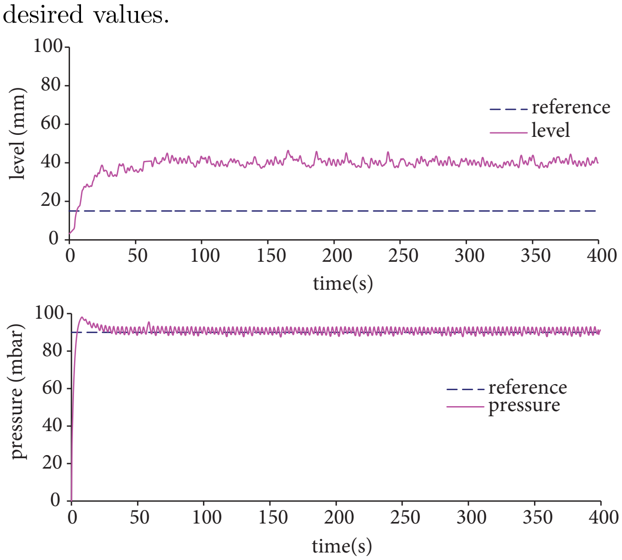

Figure 6. System response for level value is selected as the minimum under MPC (Ref_level = 15 mm, Ref_pressure = 90 mbar).

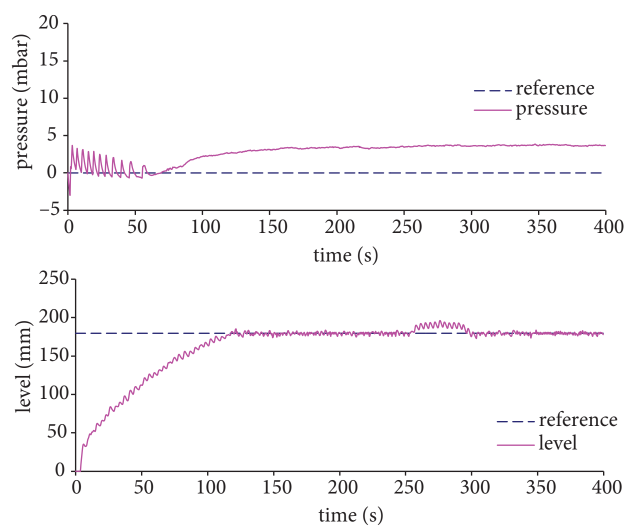

Figure 7. System response for pressure value is selected as minimum under MPC (Ref_level = 180 mm, Ref_pressure = 0 mbar).

The second experiment examines the performance of the MPC under constraints. For this purpose, the chosen pressure value is 92 mbar as this is very close to the safety constraint value of 95 mbar. The level value is also selected as 110 mm. As a result, the pressure value has set the reference value at an error rate of only 3%, as shown in Figure 8. It is important to note that the controller has successfully kept the system within the constraints, and that a 95 mbar pressure value has not been reached at this point. MPC has been used in experiments with constrained and unconstrained conditions. We have considered the same reference values for both conditions. In the first experiments, the chosen liquid level reference value is 130 mm for the first 250 s and then 180 mm until the total time of 400 s (the duration of the experiment). The chosen reference value for the pressure is 80 mbar for the first 150 s and then 65 mbar for the rest. Figure 9 shows the system response for varying reference values for the pressure and level systems under the MPC. The liquid level of the system has been set to the reference value (130 mm) after a small 6% overshoot and a 1.8% steady-state error, and settling time of 100.8 s. After changing the reference value to 180 mm, the liquid level of the system has been set to the new reference value after a 4% steady-state error for 314 s. Since the pressure reference has changed from 80 mbar to 65 mbar, the liquid level has changed to just around 5% of the reference, but the controller has provided the system output at the reference value within 2 s. If we consider the pressure-time graphic in Figure 9, the system has reached the first reference value of 80 mbar with a small 9% overshoot and 5% steady-state error. After choosing the second reference value of 65 mbar at 150 s, the pressure value of the system has been set to the new reference value after a 7% steady-state error at 153.6 s. As shown in Figure 9, when the pressure changes, the liquid level is affected. On the other hand, when the liquid level is changed, there is no remarkable change in the pressure value.

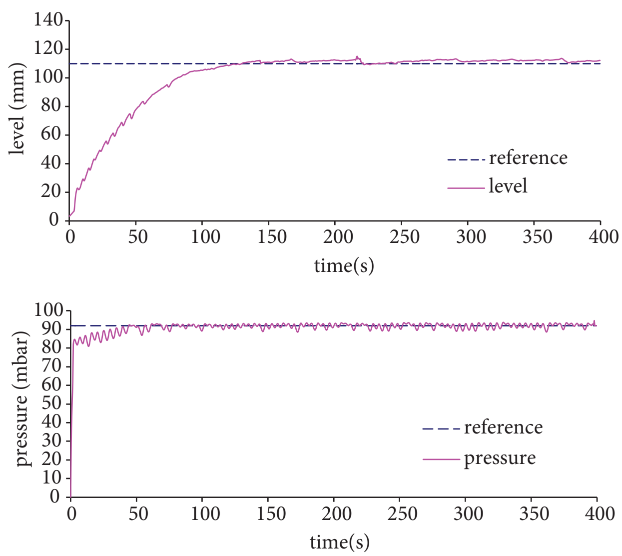

Figure 8. System response for very close values to constraint values of pressure and level under MPC (Ref_level = 110 mm, Ref_pressure = 92 mbar).

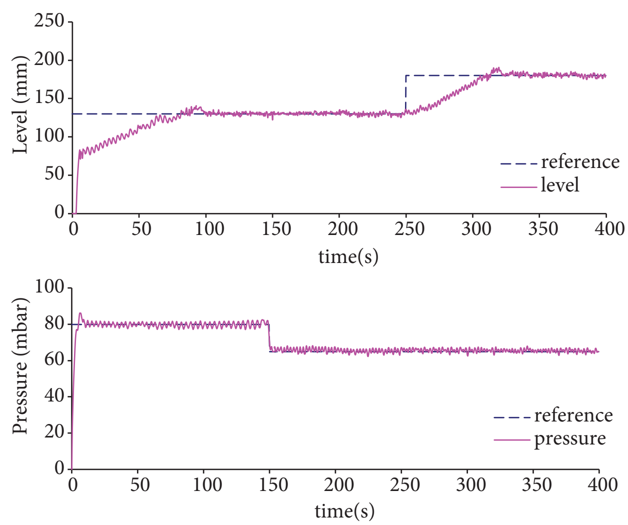

Figure 9. System response for varying reference values of pressure and level system under Model Predictive Controller (Ref_level=130-180mm, Ref_pressure=80-65mbar).

For the purpose of obtaining the system response with disturbance, the manual valve that stores the output of the level tank has been changed. In this experiment, the reference value of the level has been chosen as 180 mm and the reference value of the pressure has been chosen as 80 mbar. After the value of the liquid level has reached the reference value of around 250 s, the manual valve has been reduced to 50% of the current value. As shown in Figure 10, the liquid level reduces to the reference value after 50 s with a small 15 mm overshoot. Figure 11 shows the comparison of the control performance of the MPC and the decoupling PI controller. The coefficients of the PI controller (Kp and Ki) have been obtained first by the Ziegler–Nichols step response method and then the adaptive tuning method. These coefficients have been considered as optimal values, and they are capable of controlling the system with optimum time and tolerances. Both controllers are very effective in the same initial conditions and same reference values. Therefore, there are no big differences between these two controllers and it is not possible to say one controller provides a better solution than the other.

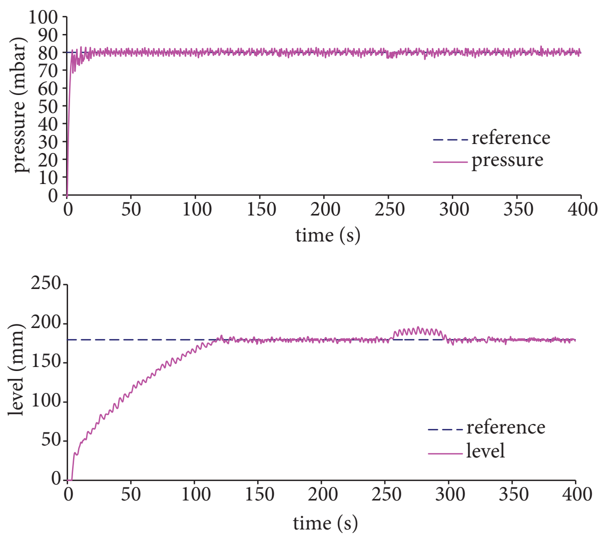

Figure 10. System response with disturbance for pressure and level system.

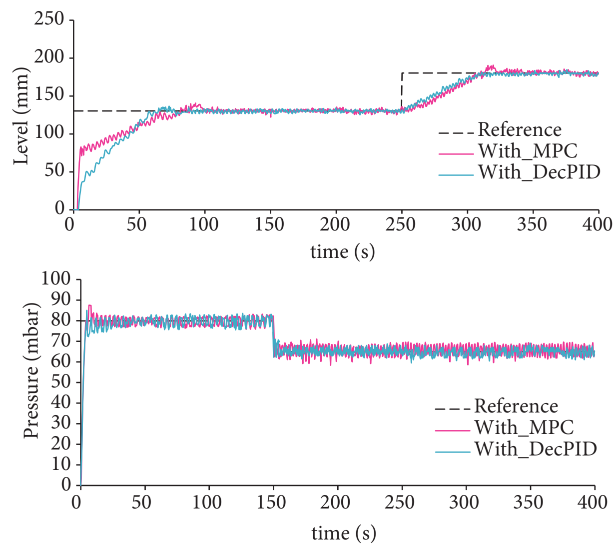

Figure 11. Comparison of the control performance of MPC and decoupling PI (Ref_level = 130–180 mm, Ref_pressure = 80–65 mbar).

5. Conclusion

The experimental results show that the MPC can control the level-pressure system under various working conditions. If we compare the results obtained from the MPC and the decoupling PI controller, we can say that both controllers are very successful. On the other hand, if we consider the constraints, the MPC is much more successful than the decoupling PI controller.

It is possible to control not only pressure and level but also flow and temperature by extending the system with flow control sensors and pressure control transducers in forthcoming research.

In the future, this experimental setup can be used for new model predictive control techniques such as multiagent MPC, neurofuzzy predictive control, and networked MPC.

References

[1] Martinsen F, Biegler L, Foss B. A new optimization algorithm with application to nonlinear MPC. Journal of Process Control 2004; 14: 853–865.

[2] Magni L, Raimondo DM, Allgöwer F. Nonlinear Model Predictive Control: Towards New Challenging Applications. Berlin, Germany: Springer, 2009.

[3] Kvasnica M, Grieder P, Baotic M, Morari M. Stabilizing low complexity feedback control of constrained piecewise affine systems. Automatica 2005; 41: 1863–1694.

[4] Kaplanoglu E, Safak KK, Varol HS. Real-time parameter estimation method applied to a MIMO process and its comparison with an offline identification method. IAENG Transactions on Engineering Technologies 2009; 1: 35–45.

[5] Kvasnica M. Real-time Model Predictive Control via Multi-parametric Programming. Saarbrücken, Germany: Verlag VDM, 2009.

[6] Zhu Y. Multivariable System Identification for Process Control. Oxford, UK: Elsevier, 2001.

[7] Skogestad S, Postlethwaite I. Multivariable Feedback Control Analysis and Design. New York, NY, USA: Wiley, 1996.

[8] Recica V. Automatic tuning of multivariable controllers. MSc, Lund Institute of Technology, Sweden, 1998.

[9] Kaplanoglu E, Varol HA, Varol HS. Trajectory tracking multivariable decoupling-pi control for a quadruple-tank system. IKS Proceeding of 2nd International Conference Intelligent Knowledge Systems; 6–8 June 2008; İstanbul, Turkey: pp 68–73.

[10] Johansson KH. The quadruple-tank process: a multivariable laboratory process with an adjustable zero. IEEE Transaction on Control Systems Technology 2000; 3: 456–465.

[11] Astrom KJ, Johansson KH, Wang QG. Design of decoupled PI controllers for two-by-two systems. Control Theory and Applications 2002; 149: 74–81.

[12] Demircioglu H, Gawthrop PJ. Continuous-time generalized predictive control (CGPC). Automatica 1991; 1: 55–74.

[13] Gawthrop PJ, Demircioğlu H, Siler-Alaca I. Multivariable continuous-time generalized predictive control: A state space approach to linear and nonlinear systems. IEEE Proc-Control Theory and Applications 1998; 145: 241–250.

[14] Camacho FE, Bordons C. Model Predictive Control. London, UK: Springer, 2000.

[15] Mayne DQ, Raković SV, Findeisen R, Allgöwer F. Robust output feedback model predictive control of constrained linear systems. Automatica 2006; 42: 1217–1222.

[16] Ronco E, Arsan T, Gawthrop PJ. Open-loop intermittent feedback control practical continuous-time GPC. IEE Proceedings of Control, Theory and Applications 1999; 146: 426–434.

[17] Gökçe D. Model predictive controller design of hydrocracker reactors. Turk J Elec Eng & Comp Sci 2011; 19: 817–825.

[18] Demircioglu H. Constrained continuous-time generalized predictive control. IEE Proc Control Theory Application 1999; 146: 470–476.

[19] Demircioglu H, Yavuzyılmaz C. Constrained predictive control in continuous time. IEEE Control Systems Magazine 2002; 22: 57–67.