BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 67, No. 5, 2019
DOI: 10.24425/bpasts.2019.130877

Hands-on MPC tuning for industrial applications

C. IONESCU[1,2,3] and D. COPOT[1,2*]

[1] Ghent University, Department of Electrical energy, Metals, Mechanical Constructions and Systems,
Research group on Dynamical Systems and Control, Tech Lane Science Park 125, Ghent B9052, Belgium
[2] EEDT core lab on Decision and Control, member of Flanders Make consortium, Tech Lane Science Park 131, Ghent 9052, Belgium
[3] Technical University of Cluj Napoca, Department of Automation, Memorandumului Street no 28, Cluj, Romania

Abstract. This paper proposes a practical tuning of closed loops with model based predictive control. The data assumed to be known from the process is the result of the bump test commonly applied in industry and known in engineering as step response data. A simplified context is assumed such that no prior know-how is required from the plant operator. The relevance of this assumption is very realistic in the context of first time users, both for industrial operators and as educational competence of first hand student training. A first order plus dead time is approximated and the controller parameters immediately follow by heuristic rules. Analysis has been performed in simulation on representative dynamics with guidelines for the various types of processes. Three single-input-single-output experimental setups have been used with no expert users available in different locations – both educational and industrial – these setups are representative for practical cases: a variable time delay dominant system, a non-minimum phase system and an open loop unstable system. Furthermore, in a multivariable control context, a train of separation columns has been tested for control in simulation, followed by experimental tests on a laboratory system with similar dynamics, i.e. a sextuple coupled water tank system. The results indicate the proposed methodology is suitable for hands-on tuning of predictive control loops with some limitations on performance and multivariable process control.

Key words: step response, first order plus dead time, predictive control, hands-on tuning, model based control, model uncertainty, robustness, control education, plant operator.

1. Introduction

Latest surveys from academia-industry platforms strongly agree that most commonly present proportional-integral-derivative (PID) control is closely followed by the model based predictive control (MPC) in terms of practical relevance [1–3]. Ever increasing demands on performance and specificity in product manufacturing yield great stress on improvements at plant-wide scale while accepting significant model uncertainty in the closed loop [4]. However, despite the interactions posed by the multimodal organization of nowadays plant structure, most control problems are viewed as decentralized control problems [5]. PID owns its success to the straightforward relation between parameters and closed loop performance, while often being unexploited to its full power when tuned optimally and fully (i.e. all three parameters) [6–9]. Automatic tuning of these controllers are traditional in industry and monitoring loops are commonly available for re-tuning whenever necessary [10–12]. Despite earlier efforts, automatic tuning of MPC has experienced little development over the last decades, while excelling in providing optimal solutions to complex systems based on good model predictors [13–21]. The tools commonly employed for relating process dynamics to controller parameters are based on academic insight, e.g. genetic algorithms [22], extremum-seeking [23, 24], etc. However, several textbook MPC works include some information on automatic tuning techniques [25, 26] and few works describe online tuning principles [27–29]. A recognized problem by control performance monitoring systems is the poor controller settings [11], often the result of inadequate operator training. The industry uses most of commercial packets and own development strategies to counteract these negative effects by surrogate solutions, but there is a clear need for better tools for non-expert users.

Relay based and combinations thereof has been employed for automatic tuning of MPC parameters [30]. A generalized minimum variance control and Smith Predictor based tuning indicated in [30, 31] the potential of extending autotuning concepts from PID to MPC. Later on, step response data based autotuning of PID has been revisited with intrinsic robustness guarantee and measurement noise attenuation [31–34]. The idea of using sensitivity functions to design controller parameters in an automated way revealed that relay based point identification may be suboptimal, improved by adding delays in the loop [35, 36]. From a practical point of view, it is appealing to allow the user to specify time-domain envelopes, which can be translated to frequency-domain specifications for sensitivity functions [37]. A recursive identification scheme for automatic re-tuning of MPC for SISO systems has been proposed in [38] and an extension to the classical first-order-plus-dead-time (FOPDT) to a fractional-order-plus-dead-time (FO2PDT) model identification has been proposed in [39]. It was suggested that input perturbation in the form of sinusoidal signals with adequate amplitude may be preferred to bump tests when operating the process.

*e-mail: dana.copot@ugent.be

Manuscript submitted 2019-05-03, revised 2019-06-05, initially accepted for publication 2019-06-09, published in October 2019

Accurate model development accounts for 70% of the development costs when tuning control loops [2, 11]. To speed up the task completion, one could firstly tune MPC hands-on process operation using the method proposed in this paper, possibly followed by further improvement of the process model. The industry uses bump tests to gather step response data for tuning of SISO loops [40], which suggests that if one would make an effort to ease the inflow of MPC used in industry loops, one should use this ubiquitously available data. Hence, we propose to make use solely of this information when tuning the MPC parameters and to extract from it the necessary information. When designing control loops, one must take into account that expertise in (optimal) tuning of controller parameters may be limited as plant operators work essentially with straightforward concepts as settling time, overshoot, sampling period. Some performance indicators may include gain margin as a robustness measure.

Often students rely on off-the-shelf tools to tune controllers due to lack of expertise and/or poor training. Such an example is the commonly used blocks of PID (automatic) tuning and MPC (automatic) tuning in Simulink, within the Matlab platform used for educational purposes. Control education is however much more than using blindly such tools, and a first hand help is always welcomed by inexperienced users. For this, an excellent textbook on MPC is [25]. However, students who do not take a course in MPC may wish to learn to tune an MPC controller with heuristic rules. The educational axis of this paper is thus to emphasize the possibility to teach MPC in a non-control oriented discipline in higher education studies.

This paper focuses on the automatic tuning methodology for MPC based on simple step response models and hands-on tuning rules. The second section introduces the context of process data extraction, followed by the section on controller tuning rules. In the fourth section, a set of typical processes is used for the SISO loop tuning analysis, followed by experimental tests on some of these available in our laboratory. Analysis on MIMO loop hands-on tuning has been briefly investigated, with both simulation and experimental test examples. Limitations for the tuning of MIMO loops are enumerated, followed by a conclusion section.

2. Data-Driven Model Approximation

The information required apriori from the process operator is twofold: i) the sampling time of the loop and ii) bump test results (i.e. step response data) [39, 41–44]. The data may be available as a vector input u and output y sampled at a sampling period T_s. The sampled step response coefficients are then used to graphically fit a FOPDT model structure:

$$\hat{P}(s) = \frac{K}{\tau s + 1} e^{-sL} \tag{1}$$

with K the gain, τ the time constant and L a possible delay (at least one sample). For this FOPDT model, various methodologies for automatic tuning of feedback controllers (mostly PID-type controllers) are readily available [6, 10, 12, 32, 35, 45].

Another use for the directly available step response input-output data vectors is to scale it to a unit response and use it in the MPC controller in the step response coefficients matrix (details in [25, 26, 46, 47]). For linear systems, this step response matrix is defined offline and prediction is based on the superposition principle. For nonlinear systems, this step response matrix is updated at every sampling time, as the process depends on the operating point and superposition principle is no longer valid. In the extreme case that simple feasibility of MPC is envisaged, a step response approximation in the form of (1) suffices to obtain the step response coefficients necessary in the MPC scheme for prediction of future outputs. The lack of persistent excitation leading to limitations in model accuracy can be compensated by realignment of model output to process output in the MPC scheme. The remainder of this paper uses this minimal process information in the MPC algorithm, with an input-output data formulation.

3. MPC Control

3.1. Input-Output MPC Formulation for SISO Systems. The MPC principles are mature and broadly available [14–18]. Among all algorithms for MPC realization, an input-output formulation may hold greater relevance when plug-and-play objectives are aimed. Process data comes in the form of input-output vectors and if one could directly use the information to tune an MPC strategy it may facilitate its usage in more process industry applications. Such input-output formulation for MPC is that of the extended prediction self-adaptive control (EPSAC) algorithm, developed in the early 80s [48, 49]. A comprehensive tutorial has been published in [47], whereas only an essential summary will be given hereafter.

Assume the process output is given by:

$$y(t) = x(t) + u(t) \tag{2}$$

in which relation past model output and past process inputs are used:

$$x(t) = f\left[x(t-1), x(t-2), \ldots, u(t-1), u(t-2), \ldots\right] \tag{3}$$

and a term $n(t)$ containing disturbances, noise and model mismatch. Notice that these functions may take any model structure: transfer function, state space, linear, nonlinear, neural networks, etc. This model is also known as the parallel model, whereas a realigned model would use past measured process outputs [47]. Both schemes for model updates will be used in this study.

The term in $n(t)$ denotes the disturbance and modelling errors effects, modelled by coloured noise:

$$n(t) = \frac{C\left(q^{-1}\right)}{D\left(q^{-1}\right)} e(t) \tag{4}$$

with $e(t)$ white noise signal and the simplest prediction is to have an integrator to ensure zero steady state error. Other disturbance filters are possible, but this is out of the scope of this study, for details see [50]. The predicted future response in linear MPC is the cumulative result of a base and an optimal response:

$$y(t+k|t) = y_{base}(t+k|t) + y_{opt}(t+k|t) \tag{5}$$

where t is the discrete-time index (will be omitted in the remainder of the paper for simplicity of notation) and k is the sample index for the prediction horizon to be estimated. The notation $(t+k|t)$ denotes here the future values, postulated at time t.

The base response can be calculated with the process and noise model for a generic control scenario u_{base} (e.g. constant value), and for linear systems, the choice for this vector values is not important as superposition principle applies. In case of nonlinear system MPC, without the linearization of the process model, it is recommended to define them as being the previously applied input value to the process: $u(t-1)$.

The second component, y_{opt}, is the effect of optimizing the future control actions defined as $\delta u(t+k|t) = u(t+k|t) - u_{base}(t+k|t)$, with $u(t+k|t)$ the optimal control input. The controller has N_u degrees of freedom, defined by the control horizon. The postulated optimal output can be calculated using the step g and impulse h response coefficients matrix G:

$$\begin{bmatrix} y_{opt}(t+1|t) \\ y_{opt}(t+2|t) \\ \dots \\ \dots \\ y_{opt}(t+N_2|t) \end{bmatrix} = \begin{bmatrix} h_1 & 0 & \dots & 0 \\ h_2 & h_1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ h_{N_2} & h_{N_2-1} & \dots & g_{N_2-N_u+1} \end{bmatrix} = \begin{bmatrix} \delta u(t|t) \\ \delta u(t+1|t) \\ \dots \\ \dots \\ \delta u(t+N_u-1|t) \end{bmatrix}.$$

Substituting the above relation in (5) the key equation for unconstrained EPSAC is obtained:

$$\mathbf{Y} = \bar{\mathbf{Y}} + \mathbf{G} \cdot \mathbf{U} \tag{6}$$

which in the special case of $N_u = 1$ reduces to a scalar. In the remainder of the paper, only cases for $N_u = 1$ will be discussed. In the EPSAC formulation, linearization of the process model is not necessary, and thus can be applied directly to nonlinear process models. In this case, the G-matrix coefficient values are updated at every sampling time by applying a small amplitude step to the process. To converge to optimal solution by minimizing the values of $\delta u(t+k|t)$, iterations within the sampling period are then applied. A schematic flowchart is given in the rectangular enclosed area denoted EPSAC in Fig. 1 which is the iterated part of the algorithm when nonlinear EPSAC is applied. The EPSAC method has been extensively applied in a manifold of applications, both simulation and real-life processes. This scheme is also used to separate the delays from the prediction model output to ease the implementation and matrix manipulation of the algorithm, as explained hereafter is a separate subsection.

There are two versions of the EPSAC algorithm, or in fact, of any MPC algorithm. First, there is the classical scheme based on model predictions used to compute the future process output as a result of various control scenarios. This is called the parallel scheme, as it assumes all differences between model prediction and measured process output as disturbances and tackles them using the noise model. As such, the predictor runs in parallel with the process, hence the name of *parallel scheme*, and it is often used in practice. Secondly, we have the series/parallel scheme, in which the model predictions are directly updated with the measured process outputs. This is a more robust scheme to dynamical changes in the plant, and suitable for difficult processes and unstable open loop processes. The updating of the predictions is called then *realignment*, as the model output realigns with the measures process output.

The cost function used in this work is based solely on the performance, i.e. minimizing the errors between the setpoint and process output. Naturally, other cost functions may be used, e.g. where also the control effort is penalized. However, the weight tuning of the error vs control effort in the cost function is yet another tuning parameter of the MPC algorithm. Similarly, the weight on the reference trajectory is another tuning parameter. These additional degrees of freedom in the user specifications may increase the complexity of the tuning procedure and may be confusing fo the first time users of MPC control. For this reason, we do not tackle these tuning parameters in this work. Even so, additional tuning of these extra parameters will only improve the results, either in terms of performance (i.e. minimize errors), or in terms of energy consumption (i.e. minimize the control effort).

3.2. Minimal Hands-on Tuning Rule. Heuristic rules for explicit min-max formulation of MPC to a scaled laboratory process have been successfully applied in [51]. In the current work, heuristic rules for model approximation and controller tuning are also envisaged. Given the available sampling period value, T_s, the prediction horizon N_p can be chosen between 10 T_s and 30 T_s interval [52, 53]. This rule gives sufficient performance for nominal processes. Large values of N_p will likely provide more stable, conservative performance of the closed loop. The control horizon N_u is only one sample, as larger value increase the (on-line) computational effort without justification in significant performance improvements. Some specific dynamics may have improved closed loop performance for higher values fo the control horizon, but this is discussed later on. As aforementioned, two implementations for the prediction updates are possible: parallel and series-parallel (realignment) scheme. We discuss here cases of process dynamics which require either one of the schemes. Disturbance rejection is not discussed in this work.

Notice that due to the fact that a FOPDT is approximated from the real step response data, there will always be modelling errors, i.e. model mis-match. The degree of model mis-match is then considered as a model uncertainty. This is not induced artificially in the paper, but it is calculated as the difference between the area of the real process step response and the area of the FOPDT approximation. The approximation to a FOPDT model is made in this paper by non-expert users, hence a first hand approximation. Some dynamics allow a relatively straightforward FOPDT approximation and some others described in this paper not. This is why the model uncertainty vary among the given examples.

3.3. Implementation for Time Delay. The prediction becomes complex in terms of matrix manipulation for situations with (variable) time delay. Since we are using FOPDT, we always have time delay in the prediction model. For SISO systems with constant time delay values this is not a problem, as the matrix size remains constant throughout the calculations. However, if time delay value varies from sample to sample, the size of matrices is dynamic and thus implementation becomes complex. Additionally, for multivariable systems with time delays, and assuming that the individual loops have different time delay values, the prediction is again complicated by manipulations of matrices with different sizes. Again, in this case, the implementation becomes difficult.

A simple solution to the matrix manipulation in presence of time delays is to assume a Smith-Predictor scheme [54] for the EPSAC algorithm, as applied in previous works [45, 55–57]. The scheme is visible in Fig. 1. In this figure we do not estimate the delay, but we assume it known. if delay is known, then the MPC predictions can be done on the model without delay, and all matrices remain of constant size. This is a reasonable decision, taking into account that any changes in the controlled variable (i.e. in the output of the process) are only visible once the time delay has passed. Hence, the delay horizon is defined as $N_d = 1 + delay$.

In particular for the complete EPSAC formulation, but valid also for the generic MPC case, is illustrated in Fig. 1. At each sampling instant, the delay-free model output $x(t)$ is calculated using the stored values $[x(t1), \ldots, u(t1), \ldots]$. At the same sampling instant, the variable time delay is estimated/computed. If the delay value in samples Nd is known, $x(t * Nd)$ can be selected out of the stored x-values, such that an intermediate variable $z(t) = x(t * Nd)$.

4. Analysis on Representative SISO Process Dynamics – Simulation Examples

4.1. Processes Requiring Only Prediction Model Data. A set of process types has been analysed for automatic tuning of MPC prediction horizon, control horizon and approximation to FOPDT process model structure. These processes are successfully controlled with the parallel implementation of MPC (i.e. only model updates for prediction), a control horizon of one sample and a minimal value of the prediction horizon of 10 samples.

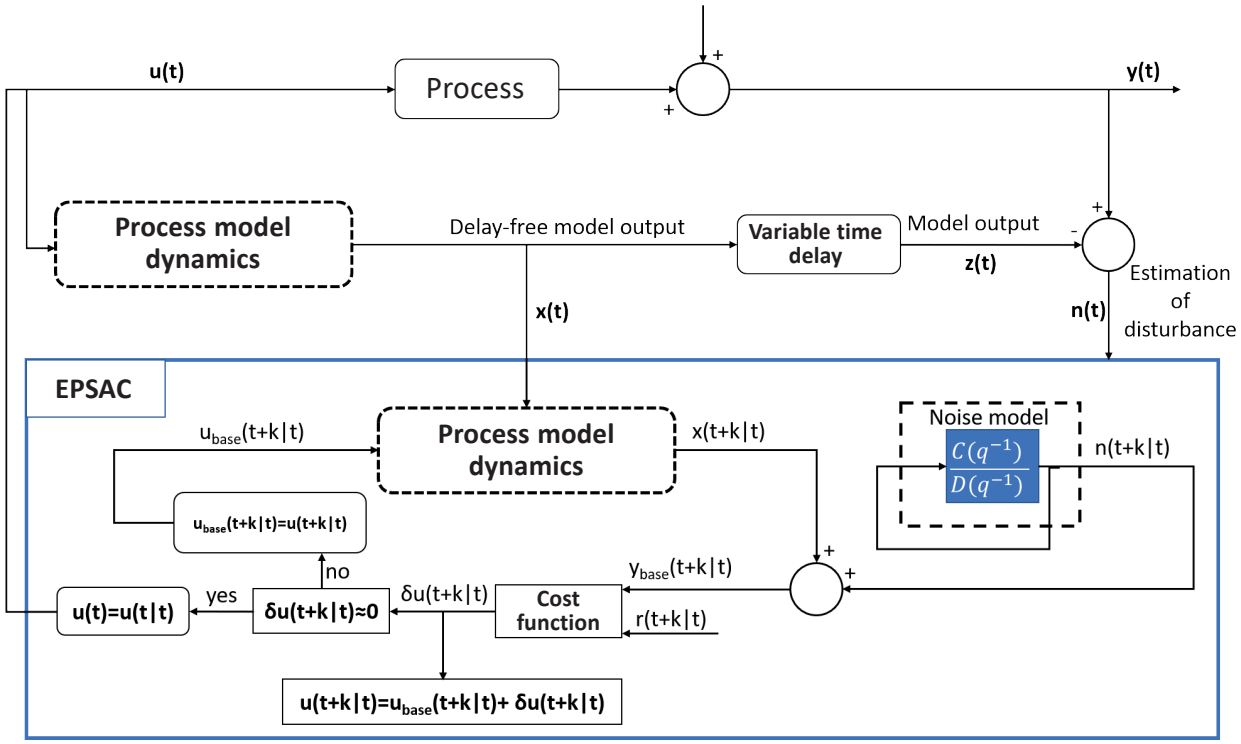

Fig. 1. Smith-Predictor like EPSAC-MPC formulation for processes with variable time delays (SISO) or processes with multiple time delays functions (MIMO)

Lag dominant process:

$$P_1(s) = \frac{15}{(s+10)(s+5)} \tag{7}$$

with FOPDT approximation:

$$P_1^*(s) = \frac{1}{(s+5)} e^{-0.1s} \tag{8}$$

has been tested for $T_s = 0.1$ seconds, $N_u = 1$, $N_d = 2$, $N_p = = N_d + 10$ samples, with the step response and closed loop result given in Fig. 2. Analysis of prediction horizon effect on performance without model mis-match is given in Fig. 3.

Balanced process:

$$P_2(s) = \frac{15}{(s+10)(s+5)} e^{-0.3s} \tag{9}$$

with FOPDT approximation:

$$P_2^*(s) = \frac{1.5}{(s+5)} e^{-0.4s} \tag{10}$$

has been tested for $T_s = 0.1$ seconds, $N_u = 1$, $N_d = 4$, $N_p = = N_d + 10$ samples, with the step response and closed loop result given in Fig. 4. Analysis of prediction horizon effect on performance without model mis-match is given in Fig. 5.

Delay dominant process:

$$P_3(s) = \frac{15}{(s+10)(s+5)} e^{-1s} \tag{11}$$

with FOPDT approximation:

$$P_3^*(s) = \frac{3}{(s+5)} e^{-1s} \tag{12}$$

has been tested for $T_s = 0.1$ seconds, $N_u = 1$, $N_d = 11$, $N_p = = N_d + 10$ samples, with the step response and closed loop result given in Fig. 6. Analysis of prediction horizon effect on performance without model mis-match is given in Fig. 7.

High order dynamics process:

$$P_4(s) = \frac{1}{(s+1)^6} \tag{13}$$

with FOPDT approximation:

$$P_4^*(s) = \frac{1}{(s+1)} e^{-4s} \tag{14}$$

has been tested for $T_s = 0.5$ seconds, $N_u = 1$, $N_d = 9$, $N_p = = N_d + 10$ samples, with the step response and closed loop result given in Fig. 8. Analysis of prediction horizon effect on performance without model mis-match is given in Fig. 9.

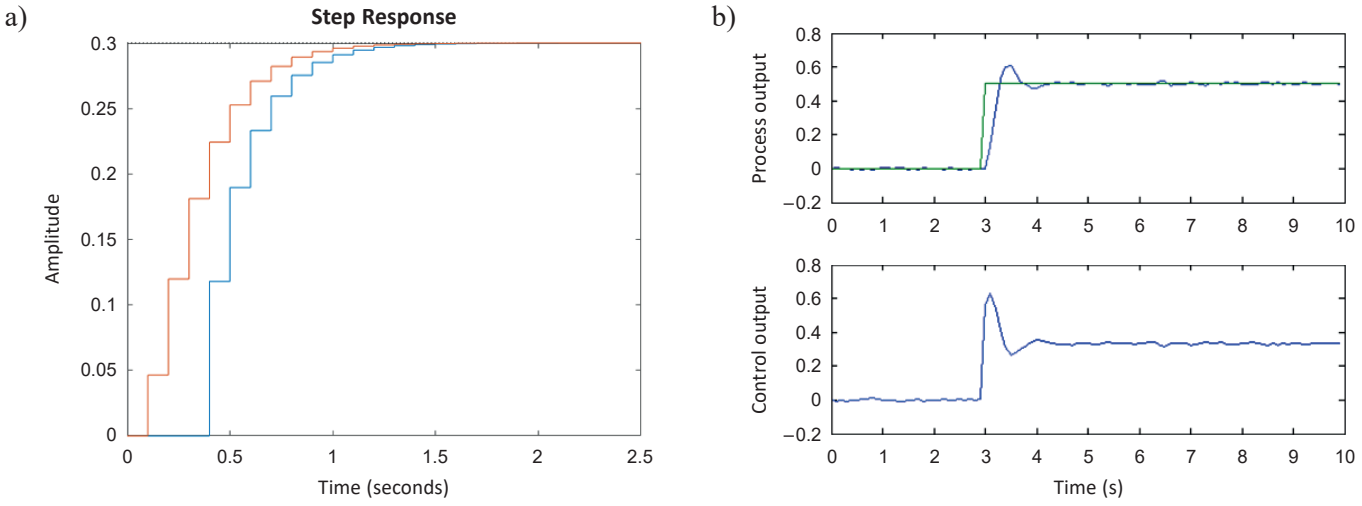

Fig. 2. Lag Process: step response approximation with 13% model uncertainty (a) and the corresponding closed loop performance (b)

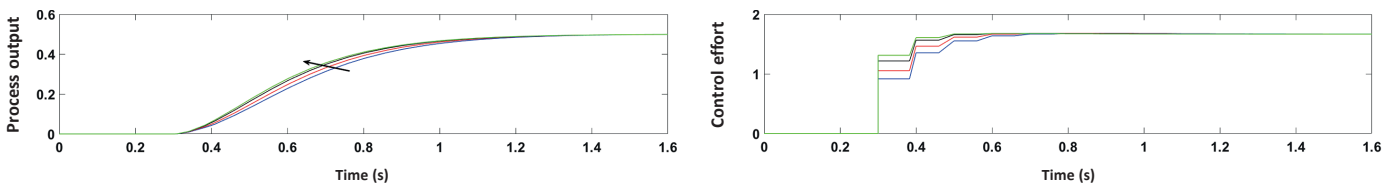

Fig. 3. Lag Process: arrow indicates the effect of N_p on closed loop performance

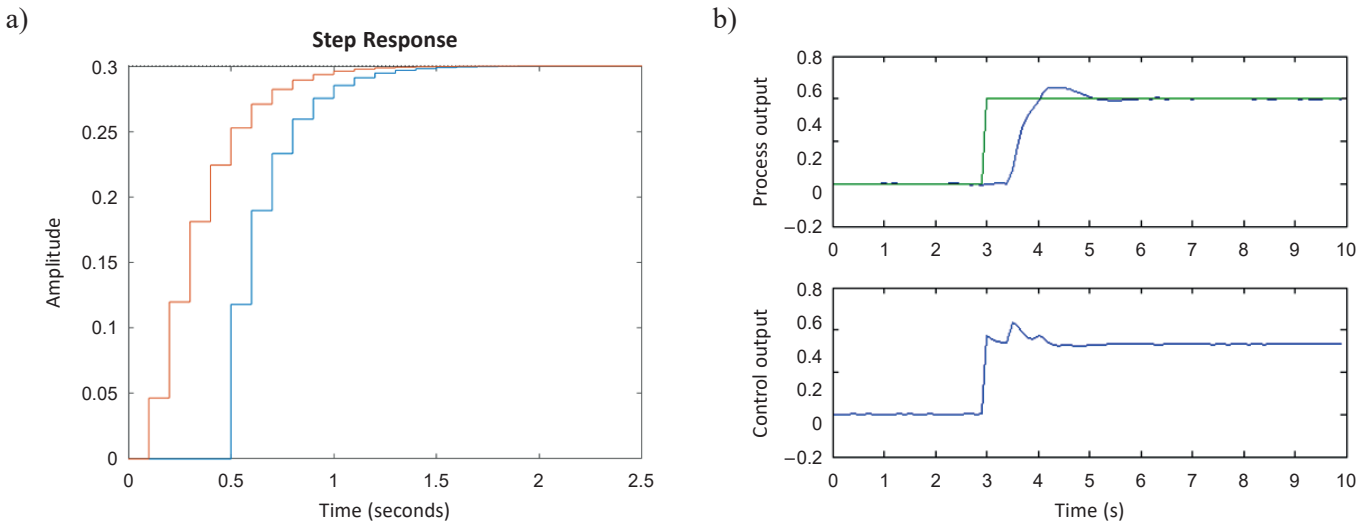

Fig. 4. Balanced Process: step response approximation with 21% model uncertainty (a) and the corresponding closed loop performance (b)

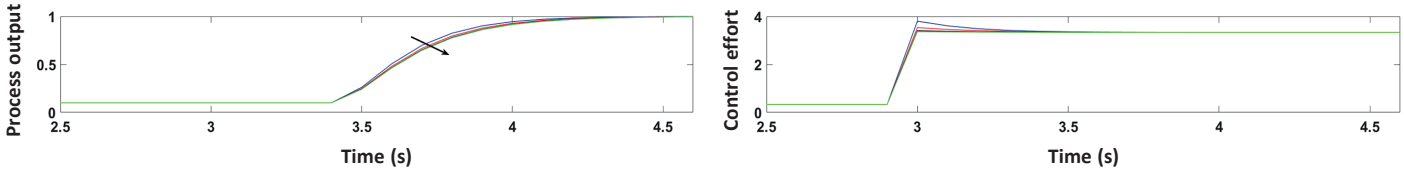

Fig. 5. Balanced Process: arrow indicates the effect of N_p on closed loop performance

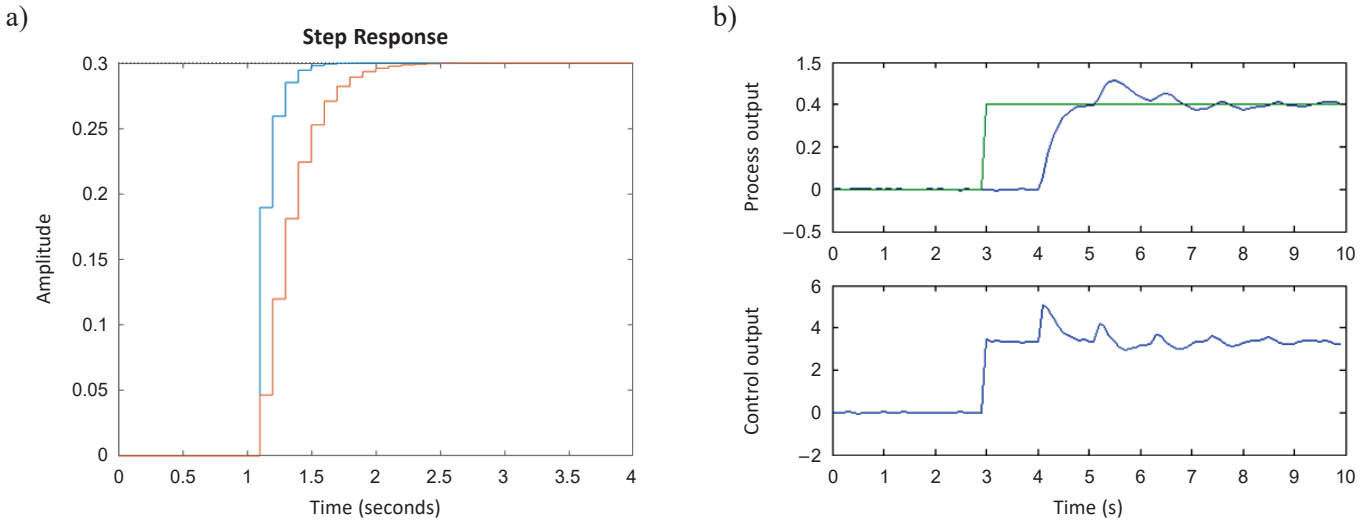

Fig. 6. Delay Dominant Process: step response approximation with 13% model uncertainty (a) and the corresponding closed loop performance (b)

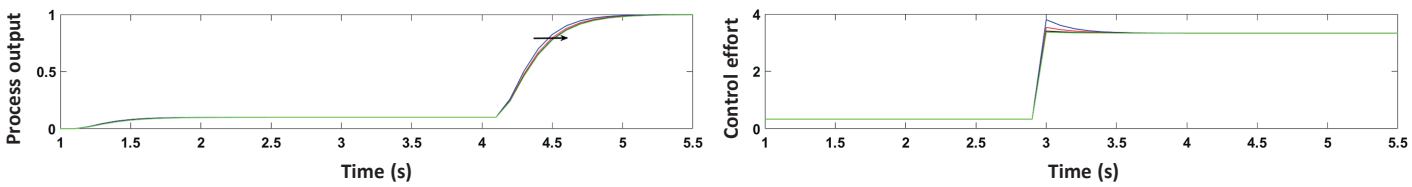

Fig. 7. Lag Process: arrow indicates the effect of N_p on closed loop performance

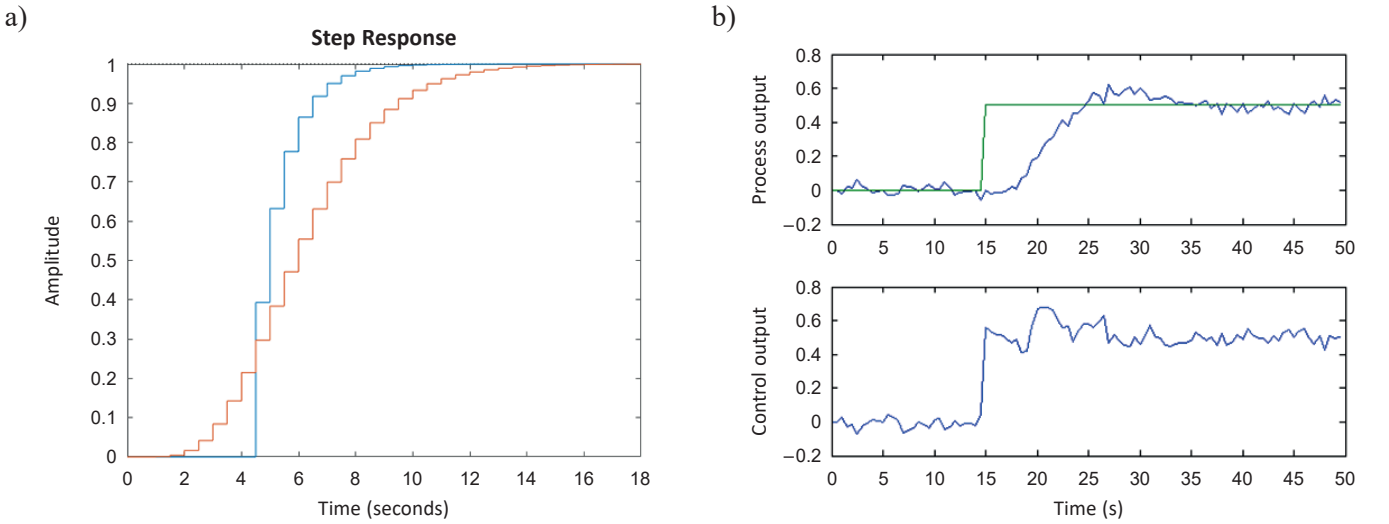

Fig. 8. High Order Process: step response approximation with 8% model uncertainty (a) and the corresponding closed loop performance (b)

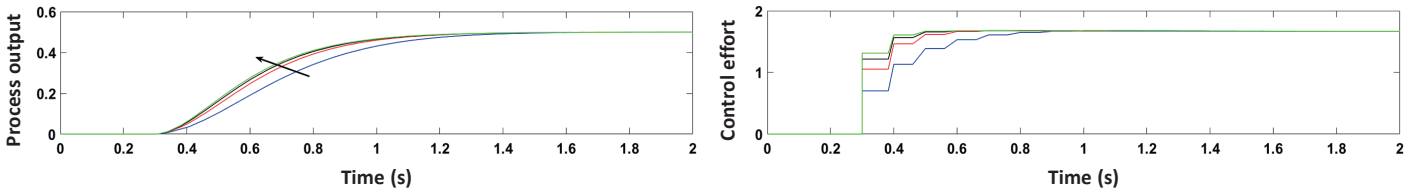

Fig. 9. Lag Process: arrow indicates the effect of N_p on closed loop performance

Underdamped dynamics process:

$$P_5(s) = \frac{15}{s^2 + 10s + 290} \tag{15}$$

with FOPDT approximation:

$$P_5^*(s) = \frac{0.6}{(s+10)} e^{-0.01s} \tag{16}$$

has been tested for $T_s = 0.01$ seconds, $N_u = 1$, $N_d = 2$, $N_p = = N_d + 20$ samples, with the step response and closed loop result given in Fig. 10. Analysis of prediction horizon effect on performance without model mis-match is given in Fig. 11.

Integrating dynamics process:

$$P_6(s) = \frac{32}{s^3 + 20s^2 + 64s} \tag{17}$$

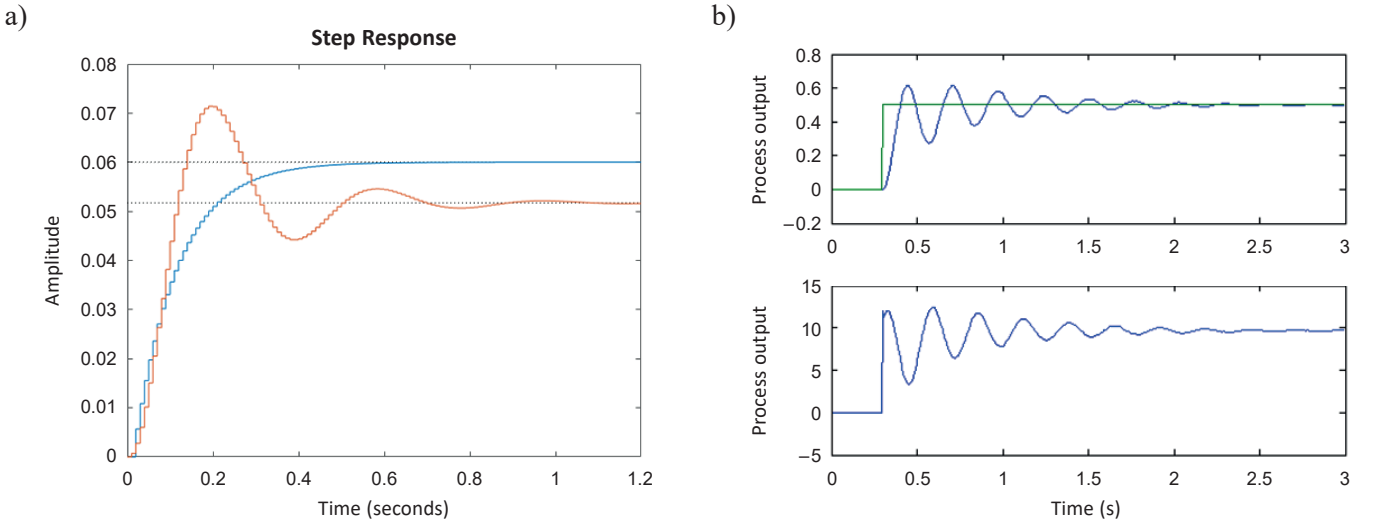

Fig. 10. Underdamped Process: step response approximation with 8% model uncertainty (a) and the corresponding closed loop performance (b)

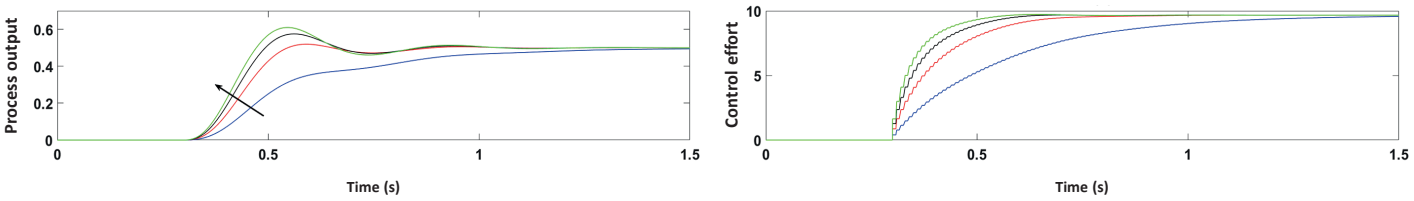

Fig. 11. Underdamped Process: arrow indicates the effect of N_p on closed loop performance

with FOPDT approximation:

$$P_6^*(s) = \frac{1}{s(s+2)} e^{-0.02s} \tag{18}$$

has been tested for $T_s = 0.02$ seconds, $N_u = 1$, $N_d = 2$, $N_p = = N_d + 30$ samples, with the step response and closed loop result given in Fig. 12. Analysis of prediction horizon effect on performance without model mis-match is given in Fig. 13.

4.2. Processes Requiring Realignment of Prediction Model to Output Data. The following is a series of systems which require the process output to be realigned to model output, due to significant model mis-match or unstable dynamics. The results are for control horizon of one sample, and a longer prediction horizon than for the processes in previous subsection.

Poorly damped dynamics process:

$$P_7(s) = \frac{160\,000}{2.4s^4 + 16.65s^3 + 4473s^2 + 14\,000s + 1300\,000} \tag{19}$$

with FOPDT approximation:

$$P_7^*(s) = \frac{1}{(s+5)} e^{-1s} \tag{20}$$

has been tested for $T_s = 0.01$ seconds, $N_u = 1$, $N_d = 101$, $N_p = = N_d + 30$ samples, with the step response and closed loop result given in Fig. 14. Analysis of prediction horizon effect on performance without model mis-match is given in Fig. 15.

a) b)

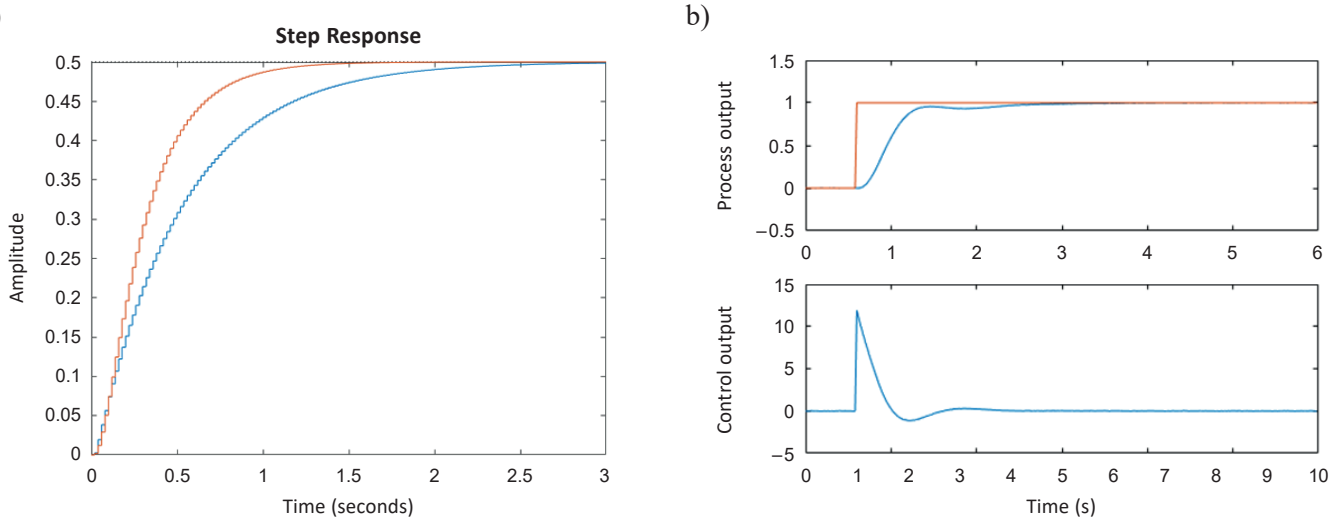

Fig. 12. Integrating Process: step response approximation with 6% model uncertainty (a) and the corresponding closed loop performance (b)

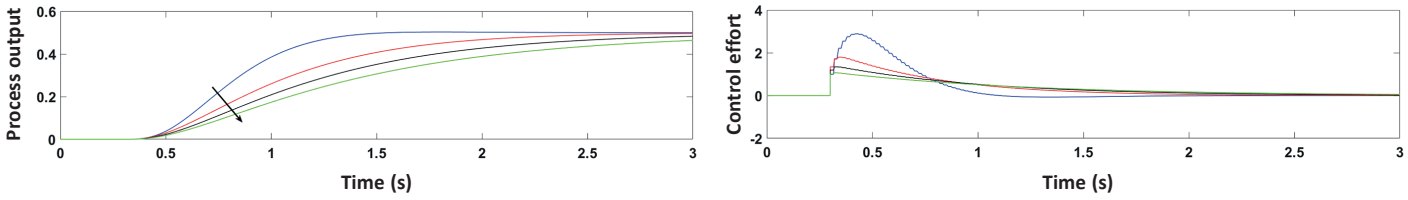

Fig. 13. Integrating Process: arrow indicates the effect of N_p on closed loop performance

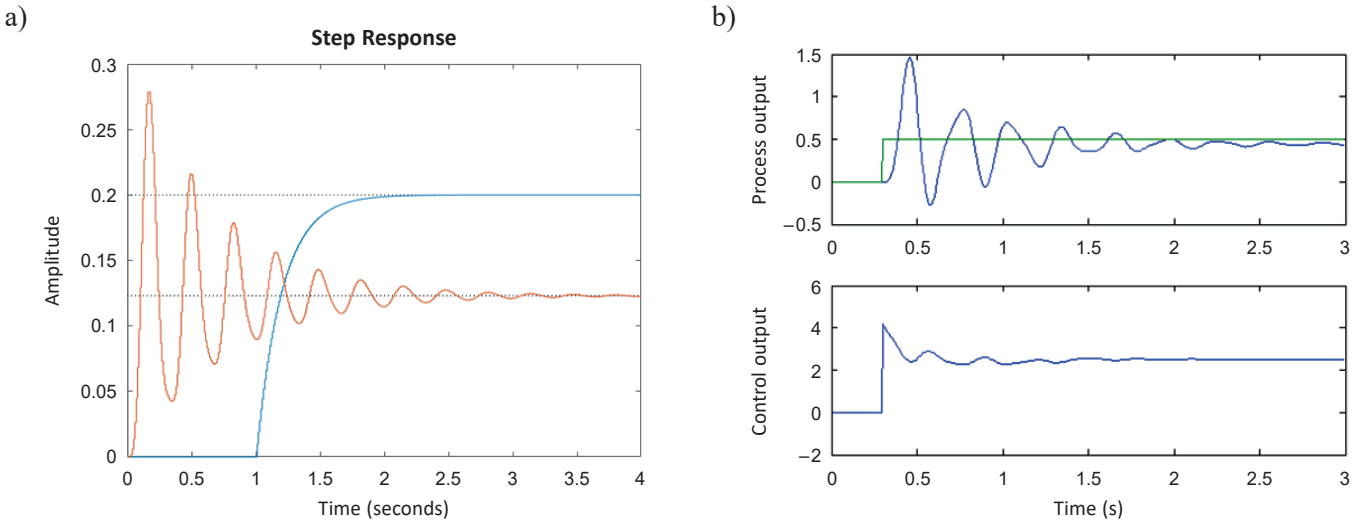

Fig. 14. Poorly Damped Process: step response approximation with 7% model uncertainty (a) and the corresponding closed loop performance (b)

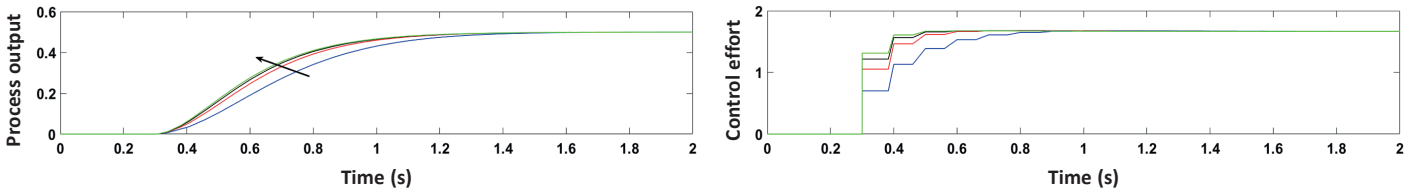

Fig. 9. Lag Process: arrow indicates the effect of N_p on closed loop performance

Non-minimum phase dynamics process:

$$P_8(s) = \frac{-15(s-2)}{(s+1)(s+10)} \tag{21}$$

with FOPDT approximation:

$$P_8^*(s) = \frac{3}{(s+1)} e^{-0.5s} \tag{22}$$

has been tested for $T_s = 0.1$ seconds, $N_u = 1$, $N_d = 6$, $N_p = N_d + 30$ samples, with the step response and closed loop result given in Fig. 16. Analysis of prediction horizon effect on performance without model mis-match is given in Fig. 17.

Unstable in open loop dynamics process:

$$P_9(s) = \frac{1}{(s+10)(s-10)} \tag{23}$$

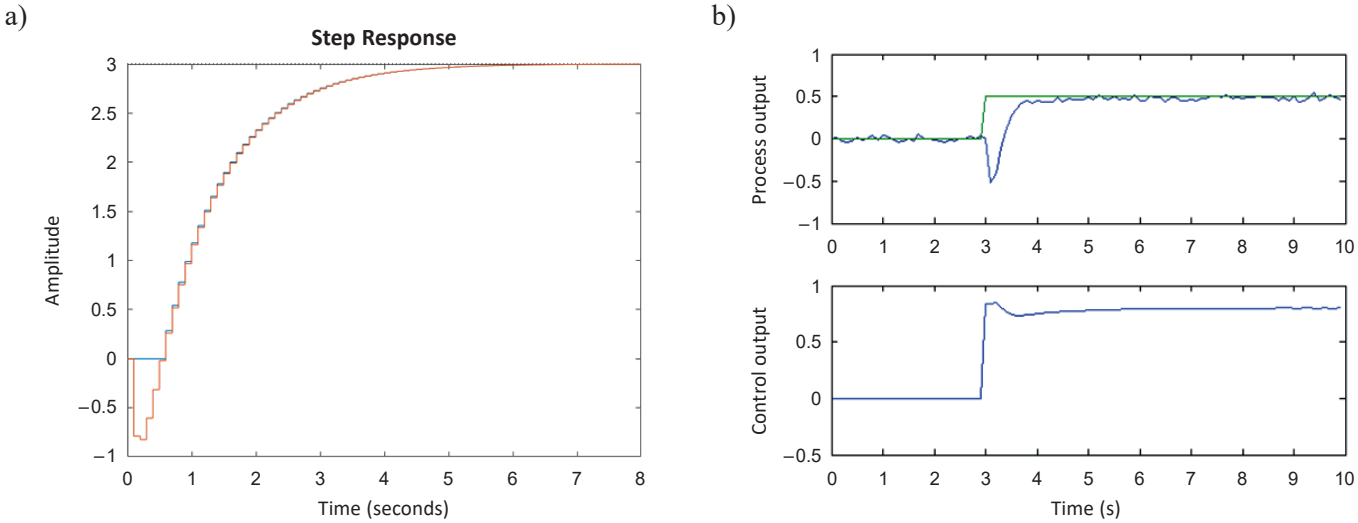

Fig. 16. Non-minimum Phase Process: step response approximation with 2% model uncertainty (a) and the corresponding closed loop performance (b)

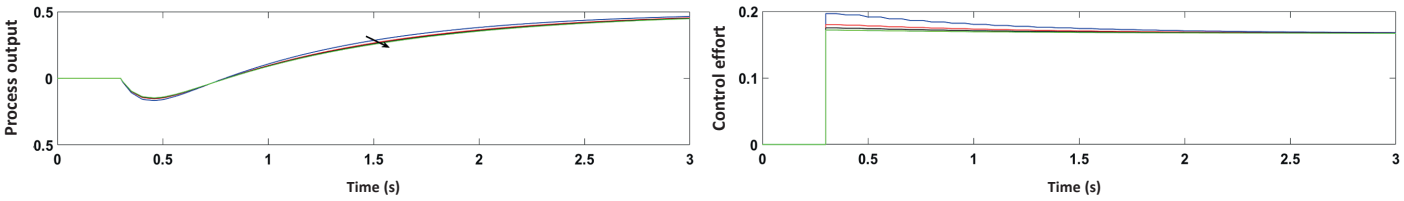

Fig. 17. Non-minimum Phase Process: arrow indicates the effect of N_p on closed loop performance

with FOPDT approximation:

$$P_9^*(s) = \frac{1}{(s+10)} e^{-0.001s} \tag{24}$$

has been tested for $T_s = 0.001$ seconds, $N_u = 1$, $N_d = 2$, $N_p = N_d + 30$ samples, with the step response and closed loop result given in Fig. 18. Analysis of prediction horizon effect on performance without model mis-match is given in Fig. 19.

4.3. Analysis Outline. The study on the effect of prediction horizon on the closed loop dynamics in absence of modelling errors allows a clear analysis for the selected processes.

In general, the prediction horizon is chosen according to the settling time of the closed loop system. If the settling time t_s of the closed loop system goes to t_s when N_p increases, then N_p can be chosen as a ratio between the settling time and the sampling period T_s. A drawback in practice is that due to noise and disturbances, it may be difficult to determine the exact value of the settling time, but an interval for the process output within 5–10% around the averaged final value is acceptable. Assuming the closed loop system settles faster than the open-loop, then the rule works fine in most of the cases. For poorly damped systems or unstable processes, the prediction horizon should be linked to the rise time t_r instead. As a rule of thumb, the relation $N_p = 3t_r/T_s$ may be applied. It has been shown that these rules are quite effective in manifold SISO and MIMO cases [46].

The choice of a good sampling period remains of crucial importance and often a challenge. In our simulation cases, the sampling time has been chosen 10–20 times smaller than the settling time of the step response and the prediction horizon between 10–30 samples. In this case, as observed from the analysis in absence of model uncertainty, the sensitivity of the closed loop to changes in the prediction horizon is small and the problem of tuning this parameter no longer poses problems.

Exception are some specific processes. Processes with complex conjugated poles dynamics exhibit periodic drops in the gain margin for increasing values of prediction horizon [46]. This is in case of poorly damped processes (15) and (19), or

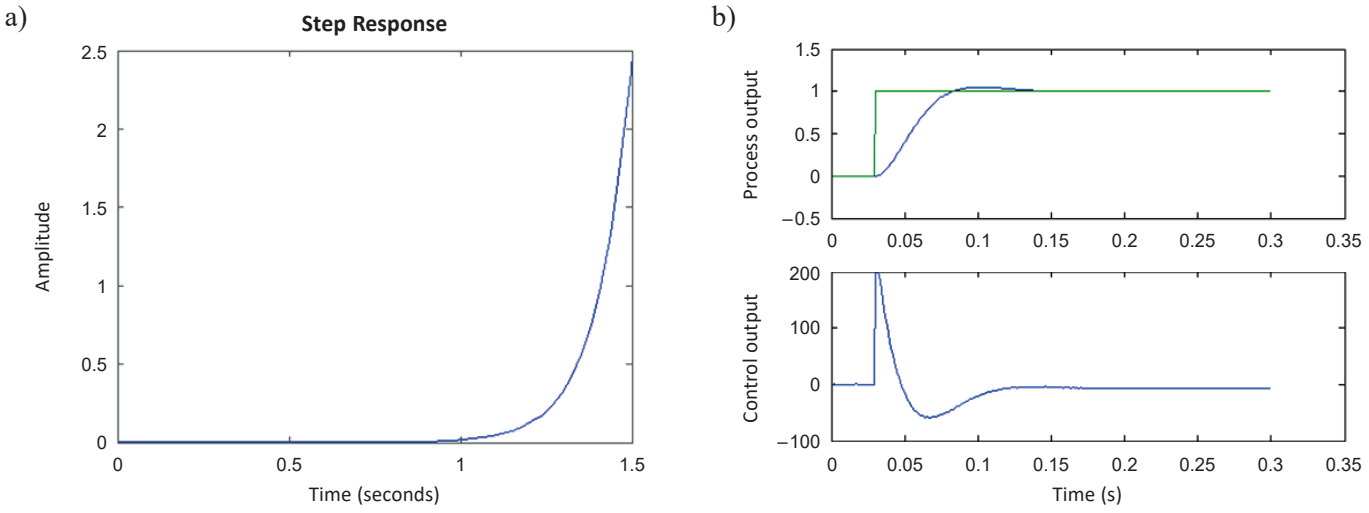

Fig. 18. Unstable Process: step response (a) and the corresponding closed loop performance (b)

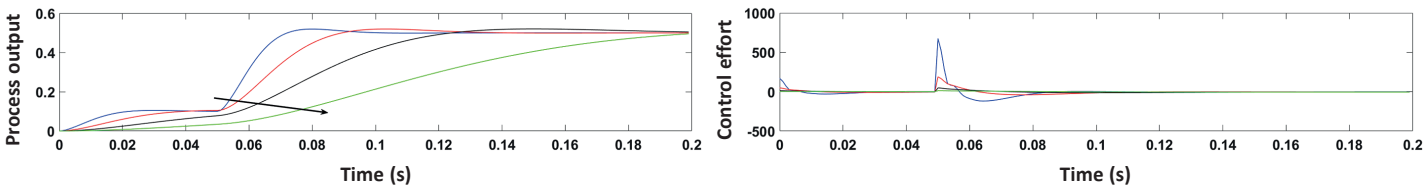

Fig. 19. Unstable Process: arrow indicates the effect of N_p on closed loop performance

processes showing non-minimum phase and complex poles dynamics. In these cases the prediction horizon must take higher values, and, for very low damping values, the process model must be updated with real output data for realignment. Processes with time delay in general are tunable with these rules, but if they are coupled with non-minimum phase dynamics, the closed loop is very sensitive to model mis-match. In this case, the realignment scheme must be used. Unstable open loop dynamics have periodic increase and decrease in gain margin robustness as prediction horizon increases, and essentially require realignment; loop sensitivity may be improved if control horizon is larger than one.

The basic algorithm for hands-on tuning of MPC control loops based on step response data can be summarized as follows:

- obtain the step response of the process and decide on a suitable sampling period,
- approximate a FOPDT model,
- verify dynamics of the real process and evaluate if you need realignment or not,
- based on previous step use a prediction horizon between 10–30 samples,
- for realignment case use larger prediction horizon than 30 samples.

An additional set of conclusions can be extracted to aid the operator tune the MPC easily.

- even in case of parallel scheme, if important modelling errors are present, the prediction horizon must be enlarged,
- if the process with enlarged prediction horizon and significant modelling errors does not give satisfactory performance in closed loop, then model with realignment MPC scheme must be used,
- the time of dynamics which cannot be captured by FOPDT structure, e.g. non-minimum phase, poorly damped oscillations, must be approximated by delay,
- if the process has non-minimum phase dynamics which are rather small compared to the overall dynamics, then the MPC scheme without model realignment gives satisfactory results; if the non-minimum phase part in the dynamic response is significant, then realignment scheme must be used,
- open loop unstable systems must be approximated by a prediction model in the form of limit stability (i.e. complex conjugated poles without real part), instead of FOPDT.

Notice all this analysis has been made for control horizon of one sample. As explained beforehand, this is the least costly implementation of predictive control and in absence of constraints it delivers a direct analytical solution. Beyond the scope of our paper, a complete and rather practical analysis of all available tuning parameters in predictive control is given in [25, 46].

5. Experimental Validation for SISO Process Control

In this section, three existing setups in our laboratory have been chosen for experimental validation of the proposed heuristic MPC tuning approach: i) a variable time delay process; ii) a non-minimum phase dynamic process and iii) an open loop unstable process.

5.1. Temperature Control. This is a process with variable time delay as the flow is the manipulated variable to control the output temperature. The setup depicted in Fig. 20 has a water tank

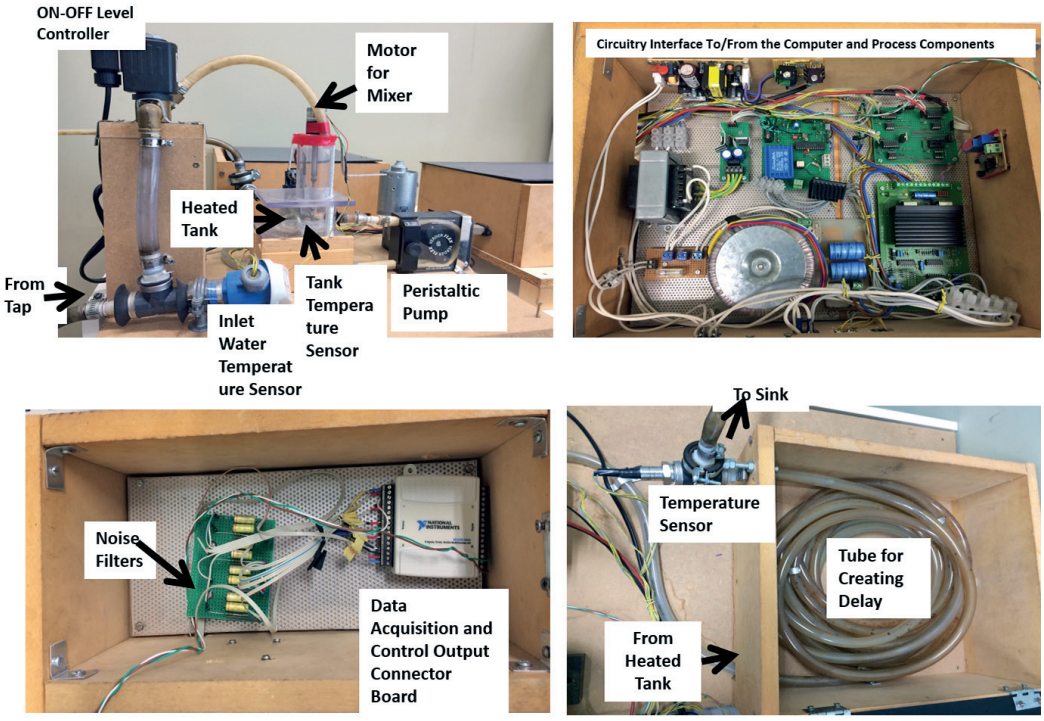

Fig. 20. Photos of the various components of the thermal regulatory process with variable time delay

with constant heat source and a regulated level for a constant inlet flow of cold water. A pump applies changes to the warm tank water outflow to regulate the temperature at the output measured at the end of a pipeline. Due to the fact that the flow is manipulated to regulate outlet temperature, the time delay values are varying during operation.

The approximation of the step response of the process is given by

$$P_{THERMAL}(s) = \frac{-950}{67s+1} e^{-15s} \tag{25}$$

and the closed loop response of the MPC algorithm for $N_u = 1$, $N_d = 4$ and $N_p = N_d + 20$, with a sampling period of $T_s = 4$ seconds is given in Fig. 21.

5.2. Level Control. This example is a single unit of connected water basin system for level control and manipulated pump output from a reservoir. A photo and scheme of the full sextuple (three units) tank system is given in Fig. 22. The system exhibits non-minimum phase dynamics. The input to this system is denoted as the voltage Vp supplied to the volumetric pump motor, while the output is denoted as $H2$ and represents the

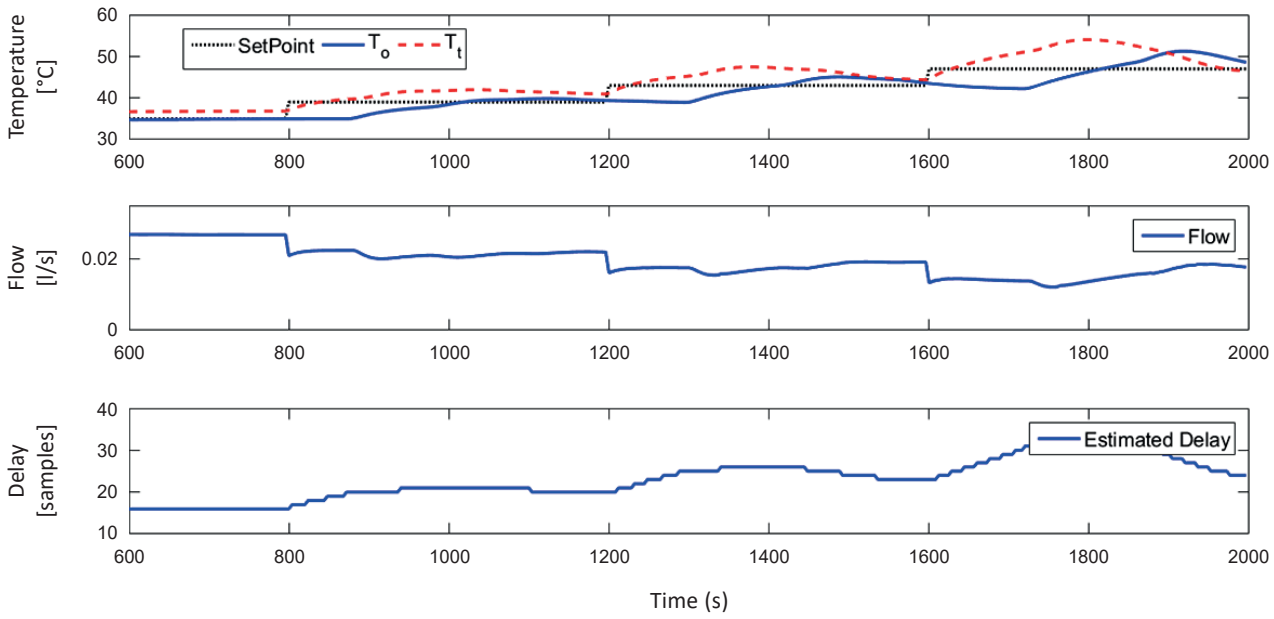

Fig. 21. Temperature Control: closed loop response. T_o is the outlet temperature to be controlled; T_t is the water temperature in the tank

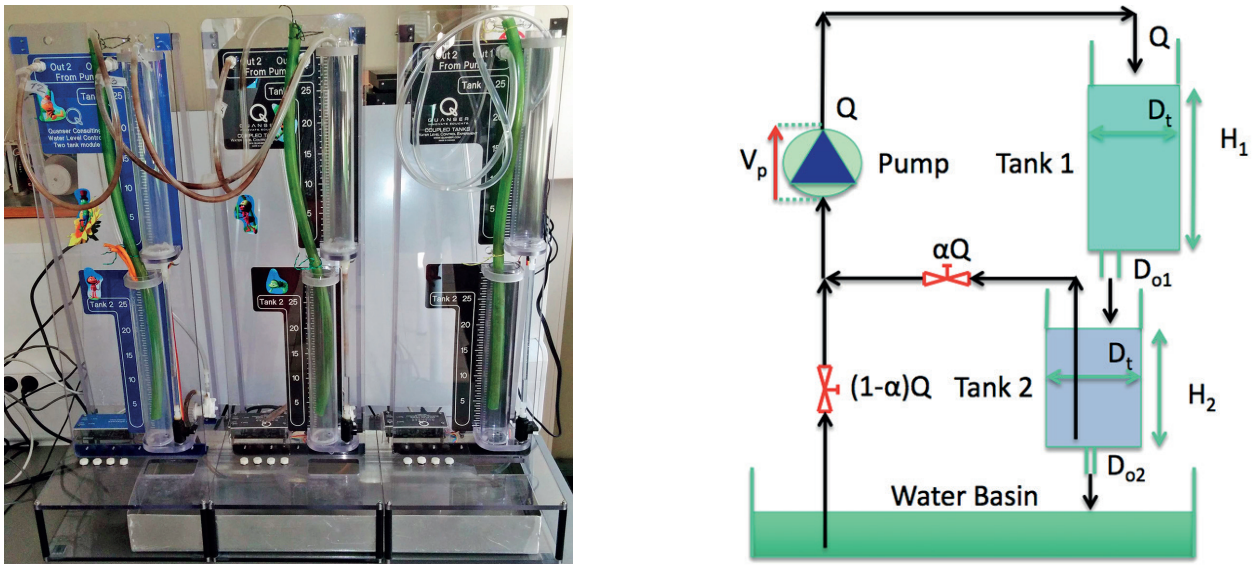

Fig. 22. Photo of the setup with the components and equivalent scheme for one unit: D_t is the tank diameter; D_o is the outlet diameter from tank; α is the percentage opening ratio valve for the water flow denoted by Q

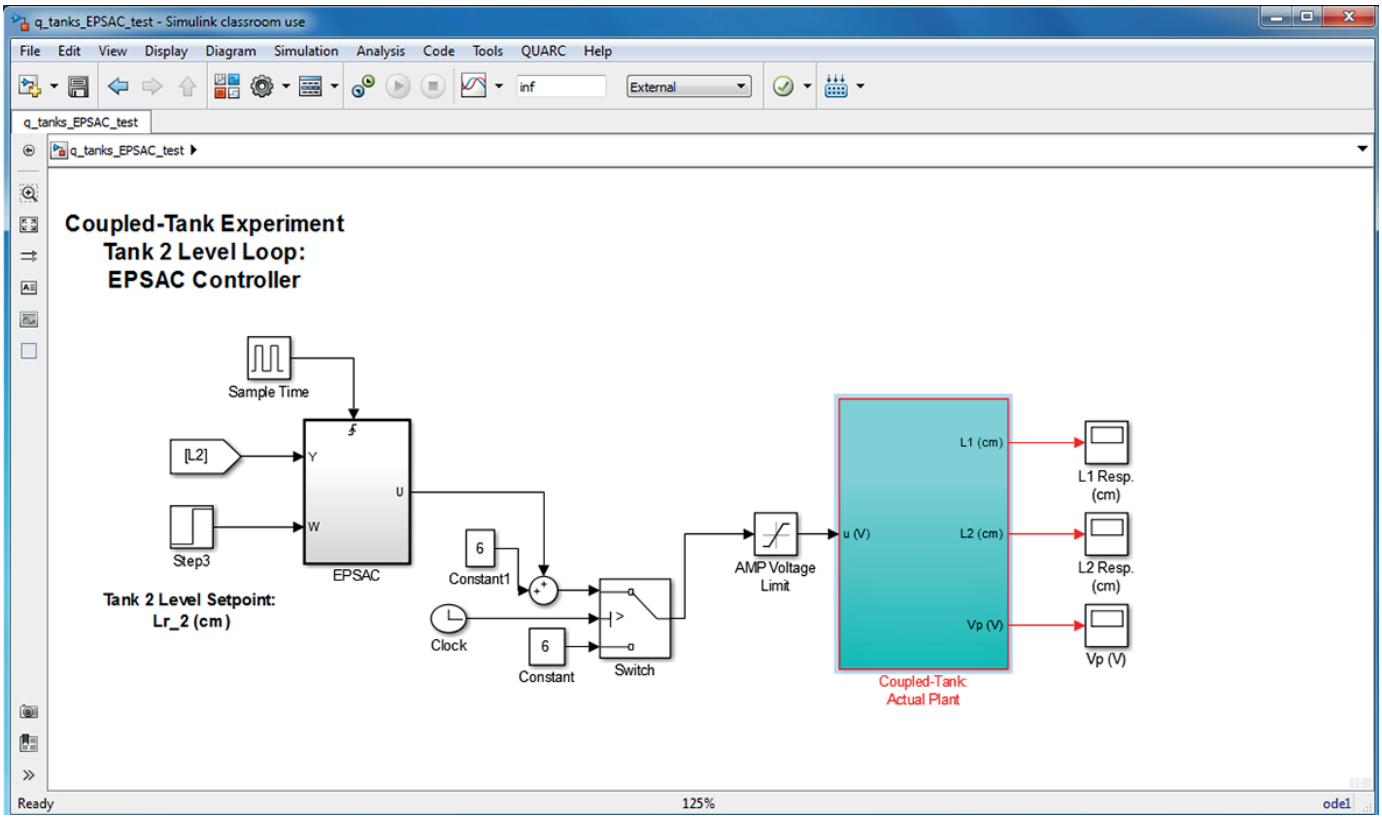

Fig. 23. The Simulink scheme with communication interface to Quark for the Quanser platform for real time deployment of the control system

level in the lower tank. The process is connected to Matlab via Quanser interface and Simulink block scheme as in Fig. 23. The MPC algorithm is the one based solely on the prediction model information (i.e. no realignment).

The step response is given in Fig. 24, with the FOPDT approximation as:

$$P_{NMP}(s) = \frac{4.25}{50s+1} e^{-15s} \tag{26}$$

The closed loop response for $N_u = 1$, $N_d = 16$ and $N_p = N_d + 50$ and sampling time of $T_s = 1$ second is given in Fig. 24.

5.3. Position Control. The process is a ball and plate system, consisting of a 6 degree of freedom platform for position control. The system in Fig. 25 has been used to test the hands-on tuning of EPSAC-MPC. Other types of control have been successfully tested on this nonlinear open loop unstable system [58–60]. The dynamics of the ball and plate platform vary sig-

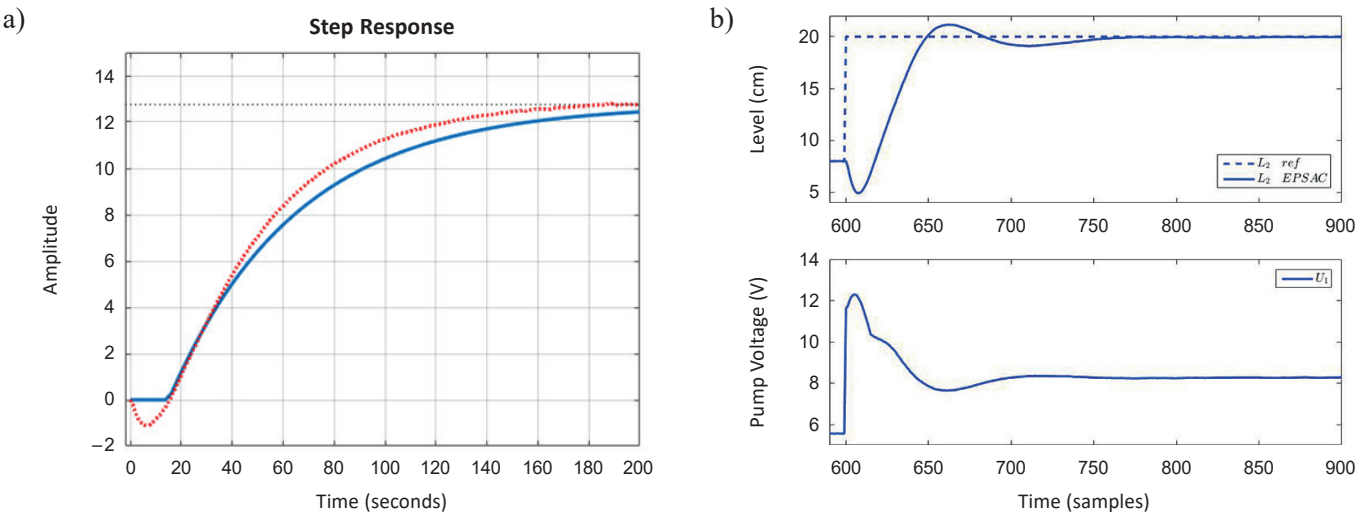

Fig. 24. Level Control: step response with FOPDT approximation (a) and the closed loop response (b)

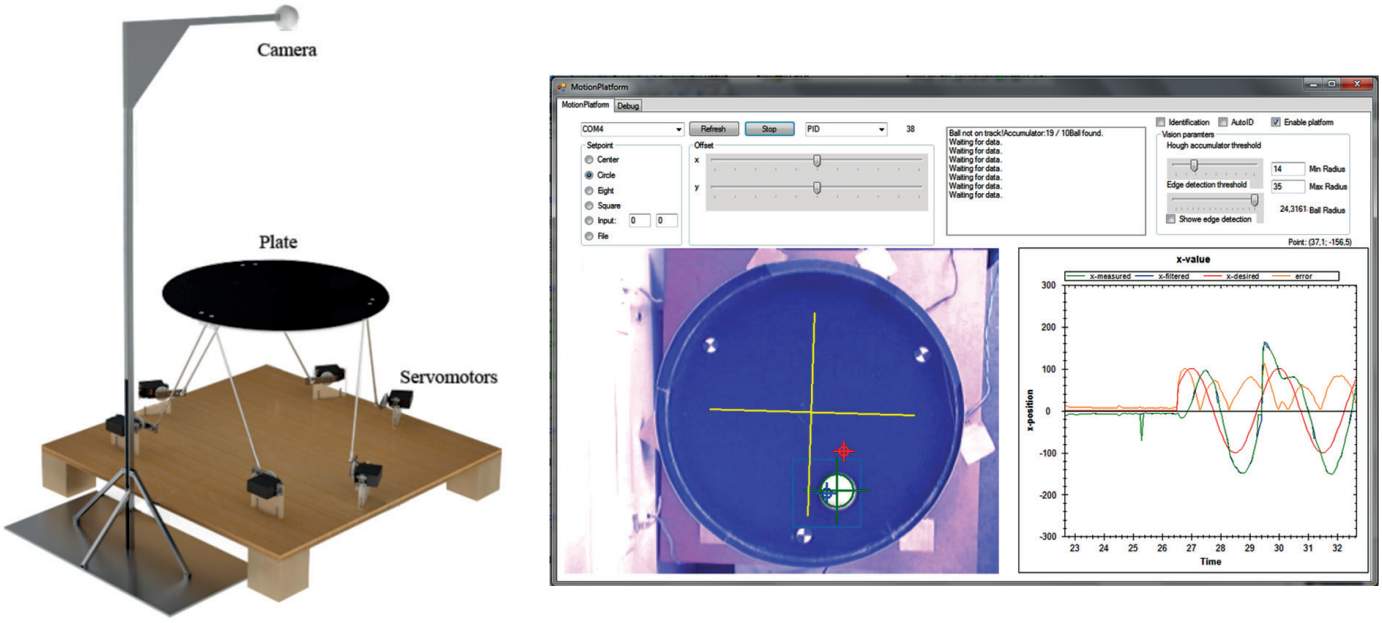

Fig. 25. Photo of the ball and plate setup with the corresponding camera-based feedback interface for testing

nificantly as a function of the ball characteristics, as discussed in [61]. Here, we use one ball and an approximated FOPDT model from the impulse response test, as the true system contains an integrator and a close to origin pole:

$$P_{BP}(s) = \frac{-15}{15s+1} e^{-s}. \tag{27}$$

Realignment scheme and prediction horizon of 30 samples for a sampling time of 0.1 seconds has been implemented with the results depicted in Fig. 26.

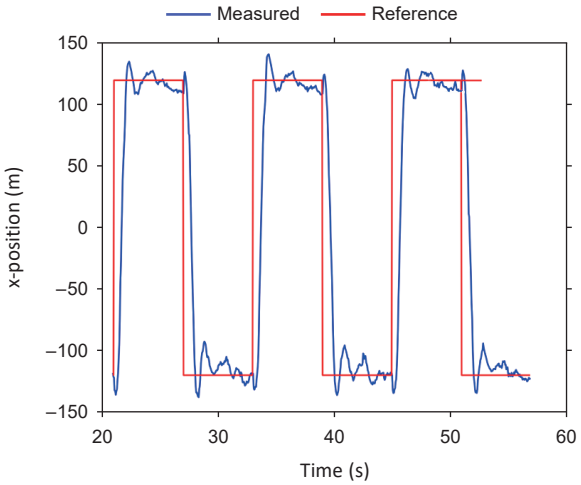

Fig. 26. Closed loop performance of the ball position for a step train test

6. Exploring MIMO Process Control

6.1. MPC Formulation for MIMO Processes. The MIMO formulation for EPSAC-MPC follows the SISO formulation. The basic equation for MIMO EPSAC for a 2×2 process is given by:

$$\begin{aligned} y_1(t) &= x_1(t) + n_1(t) \\ y_2(t) &= x_2(t) + n_2(t) \end{aligned} \tag{28}$$

where

$$\begin{aligned} x_1(t) &= f_1[x_1(t-1), x_1(t-2), \ldots, u_1(t-1), u_1(t-2), \ldots] \\ x_2(t) &= f_2[x_2(t-1), x_2(t-2), \ldots, u_2(t-1), u_2(t-2), \ldots] \end{aligned} \tag{29}$$

The term in $n(t)$ denote the disturbance and modelling errors effects, modelled by coloured noise:

$$\begin{aligned} n_1(t) &= \frac{C_1(q^{-1})}{D_1(q^{-1})} e_1(t) \\ n_2(t) &= \frac{C_2(q^{-1})}{D_2(q^{-1})} e_2(t) \end{aligned} \tag{30}$$

with $e(t)$ white noise signals. The algorithm presented in detail in [47] introduces the concepts of base response and optimizing response:

$$\begin{aligned} y_1(t+k|t) &= y_{1base}(t+k|t) + y_{1opt}(t+k|t) \\ y_2(t+k|t) &= y_{2base}(t+k|t) + y_{2opt}(t+k|t) \end{aligned} \tag{31}$$

or in matrix format:

$$\begin{bmatrix} y_{1opt}(t+1|t) \\ y_{1opt}(t+2|t) \\ \dots \\ \dots \\ y_{1opt}(t+N_2|t) \end{bmatrix} = \begin{bmatrix} h_1^{11} & 0 & 0 & \dots & 0 \\ h_2^{11} & h_1^{11} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ h_{Np}^{11} & h_{Np-1}^{11} & h_{Np-2}^{11} & \dots & g_{Np-Nu+1}^{11} \end{bmatrix} \begin{bmatrix} \delta u_1(t|t) \\ \delta u_1(t+1|t) \\ \dots \\ \dots \\ \delta u_1(t+N_u-1|t) \end{bmatrix} + \\ + \begin{bmatrix} h_1^{12} & 0 & 0 & \dots & 0 \\ h_2^{12} & h_1^{12} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ h_{Np}^{12} & h_{Np-1}^{12} & h_{Np-2}^{12} & \dots & g_{Np-Nu+1}^{12} \end{bmatrix} \begin{bmatrix} \delta u_2(t|t) \\ \delta u_2(t+1|t) \\ \dots \\ \dots \\ \delta u_2(t+N_u-1|t) \end{bmatrix}. \tag{32}$$

In general, the prediction horizons N_p could be different for the two outputs, whereas the control horizons N_u could be different for the two inputs. In this study we assume control horizon to one sample and same prediction horizon in all loops.

Extension from SISO case, the key equation for MIMO EPSAC is:

$$\begin{aligned} \mathbf{Y_1} &= \bar{\mathbf{Y}}_\mathbf{1} + \mathbf{G_{11}} \cdot \mathbf{U_1} + \mathbf{G_{12}} \cdot \mathbf{U_2} \\ \mathbf{Y_2} &= \bar{\mathbf{Y}}_\mathbf{2} + \mathbf{G_{21}} \cdot \mathbf{U_1} + \mathbf{G_{22}} \cdot \mathbf{U_2}. \end{aligned} \tag{33}$$

Two implementations are possible for MIMO systems, described in detail elsewhere [62, 63]. Essentially, one optimization finds the best strategy for each loop, while using the information from the other loops from its own benefit. By contrast, the other strategy finds the best solution for all loops. Consider the cost function is given by:

$$\sum_{k=N_d}^{N_p} \left[r_1(t+k|t) - y_1(t+k|t)\right]^2 + \\ + \sum_{k=N_d}^{N_p} \left[r_2(t+k|t) - y_2(t+k|t)\right]^2 \tag{34}$$

with r_j denoting reference signals for all j controlled outputs and subject to $u_1(t+k|t) = u_1(t+N_u-1|t)$ and to $u_2(t+k|t) = u_2(t+N_u-1|t)$ for $k \geq N_u$. Which is also valid for the case when the number of inputs differs from the number of outputs. In (34), the predicted control errors summed over all process outputs are minimized. In practice, this implies that some variables may deliberately allow more errors to help other variables reach the setpoint in order to minimize the global cost. In EPSAC-MPC formulation, this is known under the name of *solidary control*.

Relation (34) can be re-written as:

$$\mathbf{J(U)} = \mathbf{U^T H U} + 2\mathbf{f^T U} + c \tag{35}$$

it follows that:

$$\begin{aligned} \mathbf{H} &= \mathbf{G_1^T G_1} + \mathbf{G_2^T G_2} \\ \mathbf{f} &= -\left[\mathbf{G_1^T}\left(\mathbf{R_1} - \bar{\mathbf{Y}}_\mathbf{1}\right) + \mathbf{G_2^T}\left(\mathbf{R_2} - \bar{\mathbf{Y}}_\mathbf{2}\right)\right] \\ c &= \left(\mathbf{R_1} - \bar{\mathbf{Y}}_\mathbf{1}\right)^\mathbf{T}\left(\mathbf{R_1} - \bar{\mathbf{Y}}_\mathbf{1}\right) + \left(\mathbf{R_1} - \bar{\mathbf{Y}}_\mathbf{1}\right)^\mathbf{T}\left(\mathbf{R_1} - \bar{\mathbf{Y}}_\mathbf{1}\right) \end{aligned} \tag{36}$$

with $G_1 = [G_{11}\ G_{12}]$ and $G_2 = [G_{21}\ G_{22}]$. In the situation there are no constraints active, the exact solution is given by $\mathbf{U}^* = -\mathbf{H}^{-1}\mathbf{f}$, as:

$$\mathbf{U}^* = -\left[\mathbf{G_1^T G_1} + \mathbf{G2^T G_2}\right]^{-1}\left[\mathbf{G_1^T}\left(\mathbf{R_1} - \bar{\mathbf{Y}}_\mathbf{1}\right) + \right. \\ \left. + \mathbf{G_2^T}\left(\mathbf{R_2} - \bar{\mathbf{Y}}_\mathbf{2}\right)\right]. \tag{37}$$

6.2. Simulation Analysis on a Train of Cryogenic 13C Separation Columns. For distillation columns, the chemical process industry has a direct matrix control type of MPC [64], but other versions of MPC have been implemented [65] and MPC-EPSAC as well [66]. Figure 27 depicts the distillation columns described in detail in [67]. For a column to operate, the key elements are represented: i) the condenser (C1) at the top of the column cooled with liquid nitrogen and ii) the electrical boiler at the bottom of the column and vacuum jacket for thermal isolation. Carbon monoxide (CO) is fed as a gas at an intermediary point in the column. Flow transducers (FT) and flow controllers (FC)/pumps are installed on the feed, waste and product flows. A dedicated level transducer (LT) for liquid CO and a level controller (LC) are present at the bottom of the column. Pressure transducers (PT) are installed at the top and bottom of the column. The train of distillation columns operates

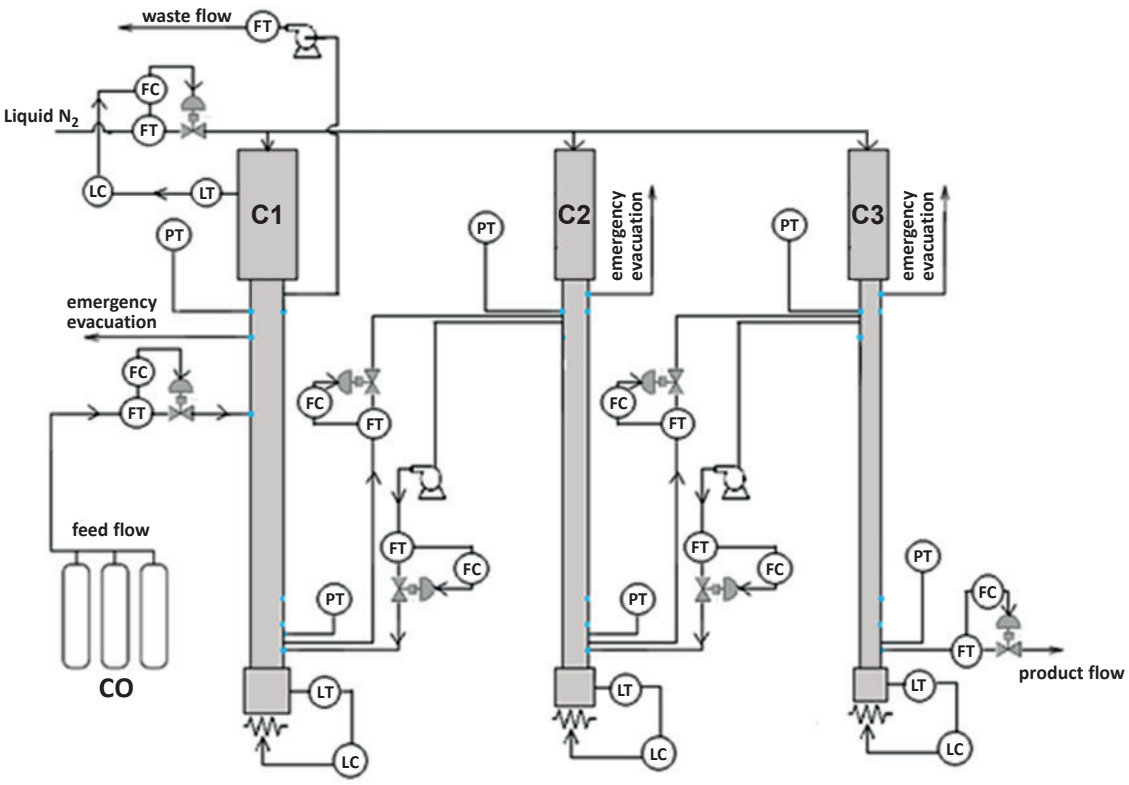

Fig. 27. Schematic of a train of distillation columns

as follows. The first column is fed with carbon monoxide at an intermediary point approximately one third from the top of the column. The enriched 13C gas from the bottom of the first column is taken as the feed to the top of the second column, and further sent to the third column in the same manner. The waste flow from the third column is recirculated to the bottom of the second column, with a similar recirculation loop from the second to the first column. The waste from the first column is stored into a reservoir.

The complex nonlinear model of a single column has been previously linearized around its equilibrium point and scaled in a $[-100\%, +100\%]$ range [55, 67]. The following models are used for simulating the real process in its operating point. First column model:

$$\begin{pmatrix} p_{t1} \\ h_{CO1} \\ p_{b1} \end{pmatrix}(s) = \begin{pmatrix} \dfrac{-0.1111}{s^2+1.094s+0.08423}e^{-10s} & \dfrac{0.1152}{s^2+1.211s+0.2021}e^{-32s} & 0 \\ \dfrac{-0.001731}{s^2+0.1343s+0.001961}e^{-10s} & \dfrac{0.003846}{s^2+0.1547s+0.004357}e^{-8s} & \dfrac{-0.104}{s+0.1176} \\ \dfrac{-0.009918}{s^2+1.056s+0.07036}e^{-18s} & \dfrac{0.006288}{s^2+1.085s+0.09851}e^{-35s} & \dfrac{8.457}{s+0.9851} \end{pmatrix} \begin{pmatrix} W_1 \\ F_1 \\ P_{el1} \end{pmatrix}(s) \tag{38}$$

Second column model:

$$\begin{pmatrix} p_{t2} \\ h_{CO2} \\ p_{b2} \end{pmatrix}(s) = \begin{pmatrix} \dfrac{-0.1111}{s^2+1.111s+0.1011}e^{-8s} & \dfrac{0.1152}{s^2+1.311s+0.3033}e^{-30s} & 0 \\ \dfrac{-0.001731}{s^2+0.13s+0.0022}e^{-8.5s} & \dfrac{0.003846}{s^2+0.15s+0.0044}e^{-7s} & \dfrac{-0.104}{s+0.12} \\ \dfrac{-0.009918}{s^2+1.06s+0.0784}e^{-16s} & \dfrac{0.006288}{s^2+1.105s+0.1225}e^{-30s} & \dfrac{8.457}{s+0.98} \end{pmatrix} \begin{pmatrix} W_2 \\ F_2 \\ P_{el2} \end{pmatrix}(s) \tag{39}$$

Third column model:

$$\begin{pmatrix} p_{t3} \\ h_{CO3} \\ p_{b3} \end{pmatrix}(s) = \begin{pmatrix} \dfrac{-0.1111}{s^2+1.131s+0.1213}e^{-7s} & \dfrac{0.1152}{s^2+1.361s+0.3538}e^{-24s} & 0 \\ \dfrac{-0.001731}{s^2+0.145s+0.003}e^{-6s} & \dfrac{0.003846}{s^2+0.17s+0.006}e^{-5s} & \dfrac{-0.104}{s+0.14} \\ \dfrac{-0.009918}{s^2+1.085s+0.0985}e^{-13s} & \dfrac{0.006288}{s^2+1.12s+0.133}e^{-27s} & \dfrac{8.457}{s+0.985} \end{pmatrix} \begin{pmatrix} W_3 \\ F_3 \\ P_{el3} \end{pmatrix}(s) \tag{40}$$

These models have defined as manipulated variables: the waste flow, feed-flow and electrical power supply in each column. The controlled variables are: the top pressure, bottom pressure and liquid CO level in each column. Simplified models describe the interaction between the column:

$$\begin{aligned} h_{CO1}(s) &= \frac{1}{10s+1}W_2(s),\ h_{CO1}(s) = \frac{-1}{10s+1}F_2(s) \\ h_{CO2}(s) &= \frac{1}{10s+1}W_3(s),\ h_{CO2}(s) = \frac{-1}{7.5s+1}F_3(s) \\ h_{CO3}(s) &= \frac{-1}{6s+1}P_3(s) \end{aligned} \tag{41}$$

where P_3 is the product flow from the third column.

This section presents the results of the EPSAC tests on the train of distillation columns afore mentioned. The sampling period is 1 minute, and all related designs are reported in samples. The model for prediction contains only the relations between the manipulated inputs: feed flow in the first, second and third column respectively, and the controlled outputs: carbon isotope in first, second and third column. Information upon reflux from third to second column and from second to first column is also used for prediction.

The control scenario is as follows: follow the desired trajectory in the output of the third column, while maintaining the other outputs around their operating point. All variables have been scaled between $\pm 100\%$ and the operating point is denoted by 0. A physically safe interval of operation has been established:

$$\begin{aligned} -10\% \le U_1, U_2, U_3 \le 30\% \\ -100\% \le Y_1, Y_2, Y_3 \le 100\% \end{aligned} \tag{42}$$

and a rate limiter in the control effort of 10%.

A selfish constrained EPSAC approach has been tested with a hands-on tuning for prediction horizon of 10 samples and FOPDT approximated models for prediction. The results are given in Fig. 28.

6.3. Experimental Test on a Sextuple Water Tank System. In this section we describe the sextuple tank system from Quanser

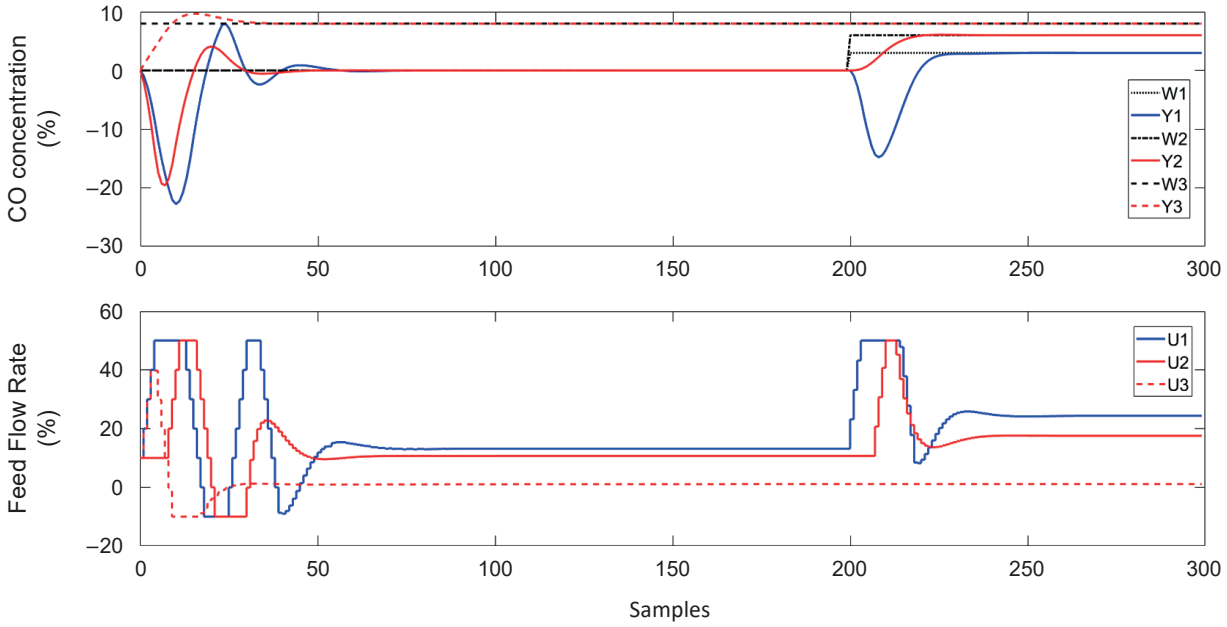

Fig. 28. Simulation result for hands-on tuning with MIMO EPSAC on the train of separation columns

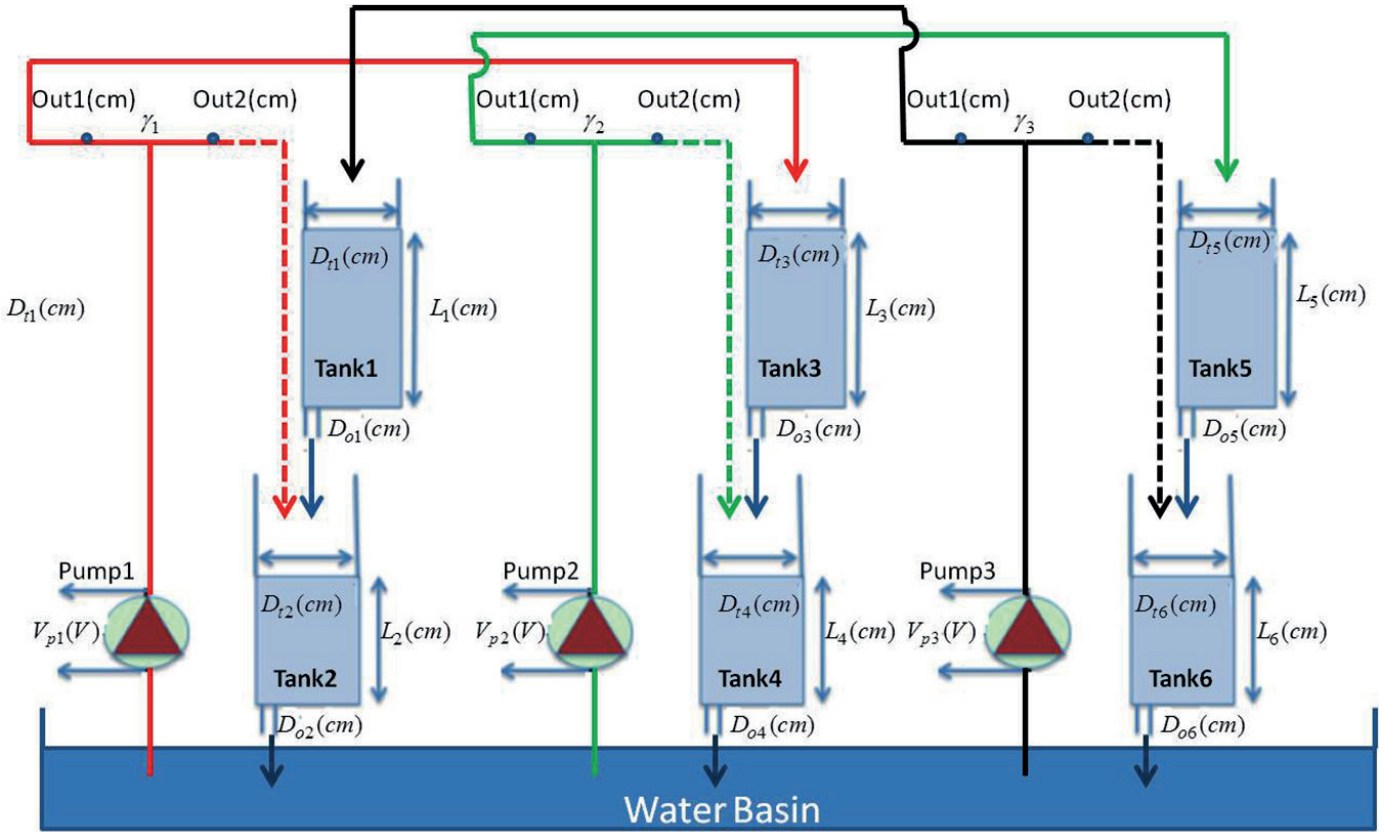

Fig. 29. Schematic diagram of the sextuple tank process

depicted in Fig. 22 with a schematic diagram given in Fig. 29. The control objective is to regulate the level of the water in the lower tanks by manipulating the inlet flows to the three way valves. The plant has three manipulated inputs, i.e. the voltages of the three pumps $V_{p1}(t)$, $V_{p2}(t)$ and $V_{p3}(t)$ (expressed in Volts) and three controlled outputs, i.e. the water levels of the three tanks below, $L_2(t)$, $L_4(t)$ and $L_6(t)$ the levels in tank2, tank4 and tank6 respectively (all expressed in cm). A detailed description of the process is given in [68].

Step response test has been performed and FOPDT model approximation extracted for prediction. A hands-on tuning of prediction horizon of 10 samples with sampling time 1 second has been used. The experimental results are given in Figs 30 and 31.

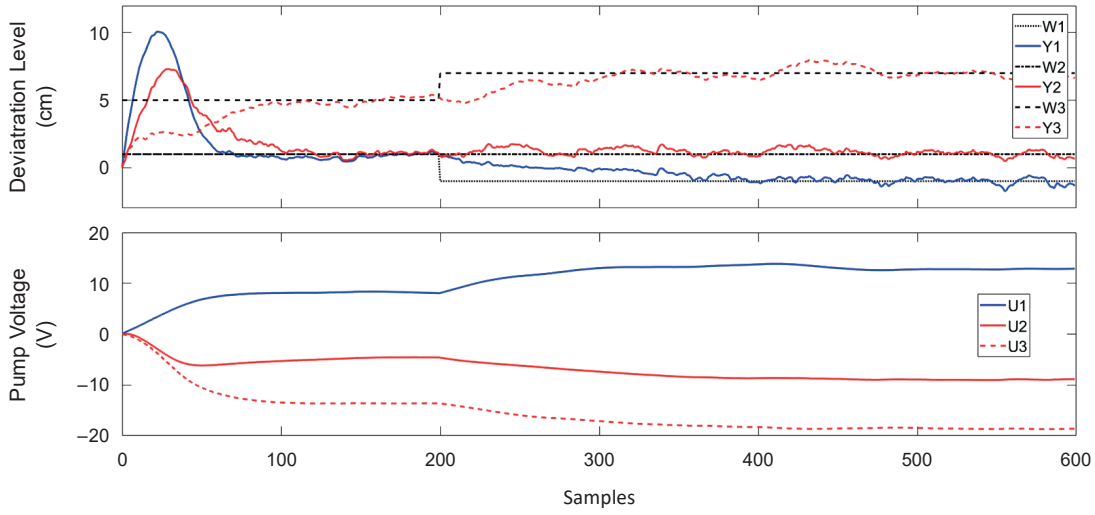

Fig. 30. Experimental result of the sextuple tank level control with hands-on MIMO EPSAC MPC tuning. Data depicts levels in the controlled water tanks 2, 4 and 6, and their respective manipulated variable pump voltages

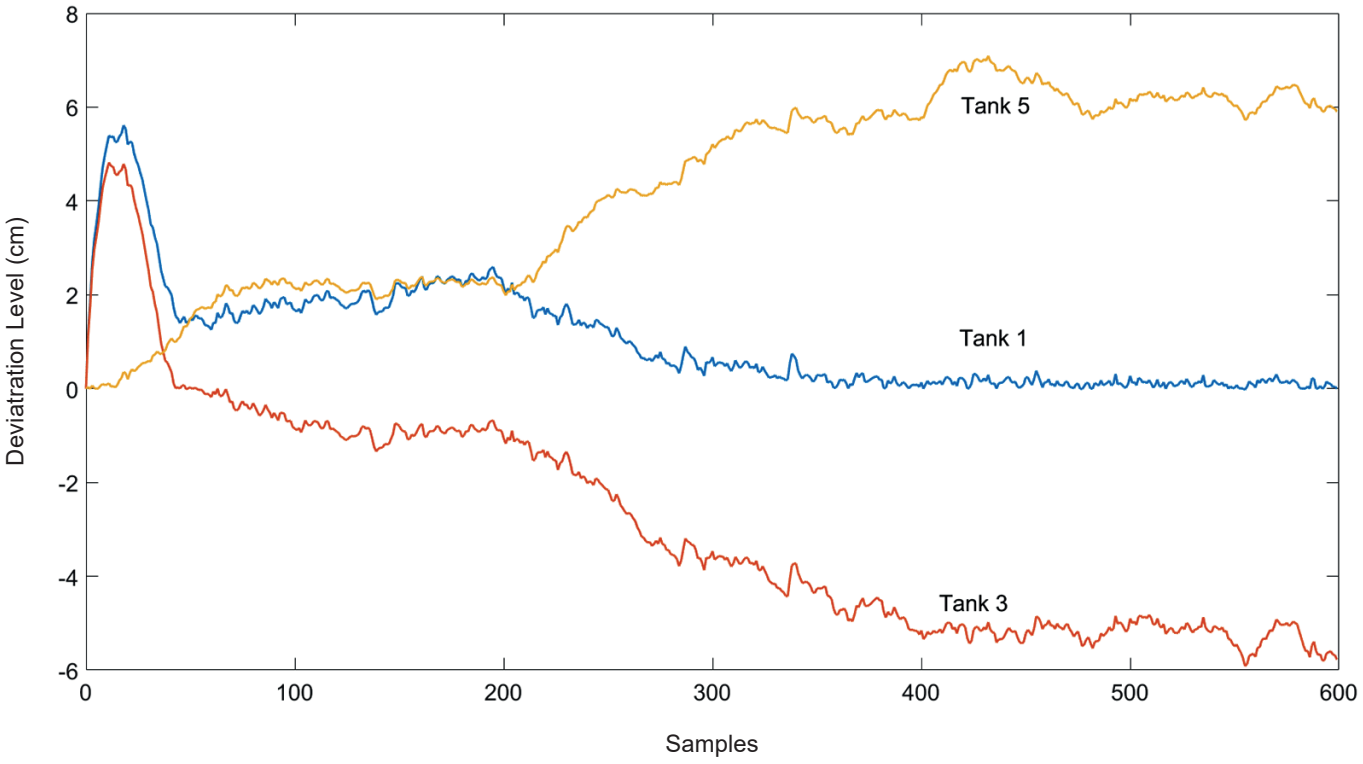

Fig. 31. Experimental result of the sextuple tank level control with hands-on MIMO EPSAC MPC tuning. Data depicts levels in the free level dynamics water tanks 1, 3 and 5

7. Limitations of this Study

The above methodology can be directly applied also to **nonlinear processes** by calculating the step response coefficient matrix at every sampling time T_s, within each optimization step. The amplitude of the step to the process is in this case not unitary, but relative to the actual expected control effort variations. In this way is ensured that the process is not disturbed from its nominal operation and the nonlinear dynamics are captured for prediction in the vicinity of current operation point. This approach avoids local linearization of the process, as explained in [47]. In this study we did not analyse the applicability of the hands-on tuning rules for EPSAC-MPC to strongly nonlinear processes. The experimental results presented here both SISO and MIMO have moderate nonlinear dynamics.

Constraints are not included in this study. The experimental tests have used control variables which do not demand accurate control. Tolerance intervals for output variables may be satisfied by tuning the prediction horizon, i.e. increasing it leads to slow and robust control performance. Input constraints are not explicitly imposed, but min-max tolerance intervals may be satisfied. Rate limiters and effects of resolution mismatch will pose more significant limitations on the performance, but we except this will not affect the hands-on tuning principles, as long as closed loop performance is stable and satisfactory.

Optimality in the sense of tuning the controller parameters is not included as it moves away from the hands-on tuning principles. The tuning is supposed to be in this context based on limited process information and in absence of an accurate prediction model.

8. Conclusions

A hands-on tuning method for predictive control based on minimal process data is presented. The following principles have been taken into account when proposing the EPSAC-MPC methodology with the tuning rules:

- applicable to large class of process types;
- does not require specialised control theory and insight training of process operators;
- easily implementable with short execution times.

The results obtained in both simulation and experimental validation suggest the method is applicable by non-expert users.

Acknowledgements. This work was supported by Ghent University Special Research Fund BOF-STG grant nr STG020–18 and Flanders Make ICON project CONACON grant nr HBC2018–0235 Context Aware Control.

References

[1] T. Samad, A survey on industry impact and challenges thereof, *IEEE Ctrl Syst Mag* 37(1), 17–18 (2017).

[2] A. Maxim, D. Copot, C. Copot, and C. Ionescu, The 5Ws for control as part of industry 4.0: why, what, where, who, and when – a PID and MPC control perspective, *Inventions* 4(1) (2019). DOI: 10.3390/inventions4010010.
[3] M. Forbes, R. Patwardhan, H. Hamadah, and R. Gopaluni, Model predictive control in industry: challenges and opportunities, in: IFAC papersOnline 48–8, IFAC, 2015, 531–538.
[4] A. Carvalho, S. Lefevre, G. Schildbach, J. Kong, and F. Borrelli, Automated driving: the role of forecasts and uncertainty – a control perspective, *European Journal of Control* 24, 14–32 (2015).
[5] G. Serale, M. Fiorentini, A. Capozzoli, D. Bernardini, and A. Bemporad, Model predictive control for enhancing building and HVAC system energy efficiency: problem formulation, applications and opportunities, *Energies* 11(631) (2018). DOI: 10.3390/en11030631.
[6] R. Vilanova and A. Visioli, PID control in the third millennium, Springer Advances in Industrial Control, London, 2012.
[7] A. Visioli, Practical PID control, Advances in Industrial Control, Springer Verlag, London, 2006.
[8] C.-C. Yu, Autotuning of PID controllers: a relay feedback approach, Springer Science & Business Media, London, 2006.
[9] K. Soltesz and A. Cervin, When is pid a good choice?, Proc of the IFAC – Conf on Advances in PID Control PID18, Ghent, Belgium, 250–255, May 9–11 (2018).
[10] K. Åström and T. Hägglund, The future of PID control, *Control Engineering Practice* 9(11), 1163–1175 (2001).
[11] M. Bauer, A. Horch, L. Xie, M. Jelali, and N. Thornhill, The current state of control loop performance monitoring – a survey of application in industry, *Journal of Process Control* 38, 1–10 (2016). DOI:10.1016/j.jprocont.2015.11.002.
[12] A. Leva and M. Maggio, Extending ideal PID tuning rules to the ISA real structure: two procedures and a benchmark campaign, *Ind Eng Chem Res* 50, 9657–9666 (2011).
[13] G.E.A. Astolfi, Special issue: The path of control, *European Journal of Control* 19, 339–435 (2013).
[14] S. Qin and T. Badgwell, A survey of industrial model predictive control technology, *Control Engineering Practice* 11, 733–764 (2003).
[15] T. Badgwell and S. Qin, Model-predictive control in practice, Encyclopedia of Systems and Control Springer Verlag London. doi:10.1007/978–1–4471–5102–9–8–1.
[16] M. Morari and J. Lee, Model predictive control: past, present and future, *Computers and Chemical Engineering* 23(4–5), 667–682 (1999).
[17] C. Garcia, D. Prett, and M. Morari, Model predictive control: theory and practice – a survey, *Automatica* 25(3), 335–348 (1989).
[18] R. Findeisen, L. Imsland, F. Allgower, and B. Foss, State and output feedback nonlinear model predictive control: an overview, *European Journal of Control* 9(2–3), 190–206 (2003).
[19] K. Antoniewicz and K. Rafal, Model predictive current control method for four-leg three-level converter operating as shunt active power filter and grid connected inverter, *Bull. Pol. Ac.: Tech.* 65(5), 601–607 (2017).
[20] P. Wiatrs and A. Krynski, Model predictive control of multilevel cascaded converter with boosting capability – experimental results, *Bull. Pol. Ac.: Tech.* 65(5), 589–599 (2017).
[21] P. Wiatrs and M. Kazmierkowski, Model predictive control of multilevel cascaded converter with boosting capability – a simulation study, *Bull. Pol. Ac.: Tech.* 64(3), 581–590 (2016).
[22] J. Lee, W. Svrcek, and B. Young, A tuning algorithm for model predictive controllers based on genetic algorithms and fuzzy decision making, *ISA Transaction* 47(1), 53–59 (2008).
[23] Y. Tan, D. Nesic, and I. Mareels, On non-local stability properties of extremum seeking control, *Automatica* 44(5), 889–903 (2006).
[24] Y. Tan, D. Nesic, and I. Mareels, On the choice of dither in extremum seeking systems: a case study, *Automatica* 44(5), 1446–1450 (2008).
[25] A. Rossiter, Model Predictive Control: a practical approach, CRC Press, 2003.
[26] E. Camacho and C. Bordons, Model Predictive Control, Springer Verlag London, 2005, 2nd edition.
[27] H. Waschl, D. Alberer, and L. del Re, Automatic tuning methods for MPC environments, In: Moreno-Díaz R., Pichler F., Quesada-Arencibia A. (eds) Computer Aided Systems Theory – EUROCAST 2011, Lecture Notes in Computer Science 6928 (Springer, Berlin, Heidelberg) 41–48 (2012).
[28] Q. Tran, J. Scholten, L. Ozkan, and A. Backx, A model-free approach for auto-tuning of model predictive control, *IFAC Proceedings Volumes* 47(3), 2189–2194 (2014).
[29] J. Garriga and M. Soroush, Model predictive control tuning methods: a review, *Industrial Engineering and Chemical Research* 49, 3505–3515 (2010).
[30] W. Cluett and E. Goderdhansingh, Autotuning for model based predictive control, *Automatica* 26(4), 691–697 (1990). DOI:10.1016/005–1098(90)90046-K.
[31] S. Skogestad and C. Grimholt, The SIMC method for smooth PID controller tuning, Chapter 5 in Book: R. Vilanova, A. Visioli (eds.), PID Control in the Third Millennium, *Advances in Industrial Control*, pages 147–175, Springer-Verlag, London, 2015.
[32] K. Åström and T. Hägglund, Revisiting the Ziegler-Nichols step response method for PID control, *Journal of Process Control* 14 (2004) 635–650. DOI:10.1016/j.jprocont.2004.01.002.
[33] V. Alfaro and R. Vilanova, Model-reference robust tuning of PID controllers, Advances in Industrial Control, Springer International Publishing, Switzerland, 2016.
[34] M. Johnson and M. Moradi, PID control, Springer Verlag, London, 2005.
[35] B. Kristiansson and B. Lennartson, Robust and optimal tuning of PI and PID controllers, IEE Proceedings Control Theory and Applications 149(1), 17–25 (2002). DOI:10.1049/ip-cta:20020088.
[36] I. Boiko, Non-parametric tuning of PID controllers. A modified relay-feedback test approach., Advances in Industrial Control, Springer Verlag, London, 2013.
[37] K. Papadopoulos, PID controller tuning using the magnitude optimum criterion, Springer International Publishing, Switzerland, 2015.
[38] C. Ionescu, D. Copot, A. Maxim, E. Dulf, R. Both, and R. De Keyser, Robust autotuning MPC for a class of process control applications, IEEE – Automation, Quality, Testing, Robotics. DOI: 10.1109/AQTR.2016.7501356.
[39] R. De Keyser and C. Ionescu, Minimal information based simple identification method of fractional order systems for model based control applications, in: 11th Asian Control Conference, IEEE, Gold Coast, 1411–1416, December 17–20, Australia, 2017.
[40] K. Starr, Single loop control methods, ABB Inc., Sweden, 2015.
[41] T. Liu, Q. Wang, and H. Huang, A tutorial review on process identification from step or relay feedback test, *J Process Control* 23(10), 1597–1623 (2013).
[42] T. Liu and F. Gao, Industrial process identification and control design, step test and relay-experiment based methods, Springer Advances in Industrial Control, London, 2012.
[43] Q. Bi, W. Cai, E. Lee, Q. Wang, C. Hang, and Y. Zhang, Robust identification of first order plus dead time model from step response, *Control Engineering Practice* 7, 71–77 (1999).
[44] A. Rossiter and R. Haber, The effect of coincidence horizon on predictive functional control, *Processes* 3, 25–45 (2015). DOI: 10.3390/pr3010025.

[45] D. Copot, M. Ghita, and C. Ionescu, Simple alternatives to PIDtype control for processes with variable time-delay, *Processes* 7 (2019) DOI:10.3390/pr7030146.
[46] R. Soeterboek, Predictive control: a unified approach, TU Delft – Academic Press, The Netherlands, 1990.
[47] R. De Keyser, Model based predictive control for linear systems, UNESCO Encyclopaedia of Life Support Systems, Control Systems, Robotics and Automation Vol. XI, Article contribution 6.43.16.1, Eolss Publishers Co Ltd, Oxford (2003) 30 pages. DOI: available online at: http://www.eolss.net/sample-chapters/c18/e6–43–16–01.pdf.
[48] R. De Keyser and A. Van Cauwenberghe, A self-tuning predictor as operator guide, In: R. Isermann (Ed.), *Identification and System Parameter Estimation*, 1249–1256 (1979).
[49] R. De Keyser and A. Van Cauwenberghe, A self-tuning multistep predictor application, *Automatica* 17(1), 167–174 (1981).
[50] R. De Keyser and C. Ionescu, The disturbance model in model based predictive control, in: IEEE Conf on Control Applications, IEEE, 446–451, Istanbul, June 23–25, Turkey, 20103.
[51] D. de la Pena, D. Ramirez, E. Camacho, and T. Alamo, Application of an explicit min-max MPC to a scaled laboratory process, *Control Engineering Practice* 13(12), 1463–1471 (2005). DOI:10.1016/j.conengprac.2004.12.008.
[52] W. Wojsznis, J. Gudaz, T. Blevins, and A. Mehta, Practical approach to tuning MPC, *ISA Transactions* 42, 149–162 (2003).
[53] A. Al-Ghazzawi, E. Ali, A. Nouh, and E. Zafiriou, On-line tuning strategy for model predictive controllers, *Journal of Process Control* 11, 265–284 (2001). DOI: PII: S0959–1524(00)00033–0.
[54] J. Normey-Rico, Control of Dead-Time processes, Springer, London, 2007.
[55] C. Ionescu, C. Muresan, D. Copot, and R. De Keyser, Constrained multivariable predictive control of a train of cryogenic 13C separation columns, in: 11th IFAC Symposium on Dynamics and Control of Process Systems, IFAC, 1103–1108, Trondheim, June 6–8, Norway, 2016.
[56] M. Sbarciog, R. De Keyser, S. Cristea, and C. De Prada, Nonlinear predictive control of processes with variable time delay. a temperature control case study, in: IEEE International Conference on Control Applications, IEEE, 601–606, San Antonio, CA Date: SEP 03–05, 2008.
[57] C. Ionescu, R. Hodrea, and R. De Keyser, Variable time-delay estimation for anesthesia control during intensive care, *IEEE Transactions on Biomedical Engineering* 58, 363–369 (2011).
[58] A. Chevalier, C. Copot, C. Ionescu, R. De Keyser, A three-year study of a remote laboratory used in control engineering studies, IEEE Transactions on Education 60 (2017) 127–133.
[59] C. Copot, Y. Zhong, C. Ionescu, and R. De Keyser, Tuning fractional PID controllers for a steward platform based on frequency domain and artificial intelligence methods, *Central European Journal of Physics* 11, 702–713 (2013).
[60] Y. Zhong, A. Dutta, C. Copot, C. Ionescu, and R. De Keyser, Implementation of a fractional PD controller tuned by genetic algorithm for a steward platform, in: 9th Asian Control Conference, IEEE, Istanbul, June 23–25, Turkey, 2013.
[61] A. Chevalier, C. Copot, C. Ionescu, and R. De Keyser, Automatic calibration with robust control of a six DoF mechatronic system, *Mechatronics* 35, 102–108 (2016).
[62] A. Maxim, D. Copot, R. De Keyser, and C. Ionescu, An industrially relevant formulation of a distributed model predictive control algorithm based on minimal process information, *Journal of Process Control* 68, 240–253 (2018).
DOI:10.1016/j.jprocont.2018.06.004.
[63] D. Fu, C. Ionescu, E. Aaghezzaf, and R. De Keyser, Decentralized and centralized model predictive control to reduce the bullwhip effect in supply chain management, *Computers and Industrial Engineering* 73, 21–31 (2014).
[64] C. Reverter, J. Ibarrola, and J. Cano Izquierdo, Tuning rules for a quick start up in dynamic matrix control, *ISA Transactions* 53, 612–627 (2014).
[65] B. Huyck, J. De Brabanter, B. De Moor, J. Van Impe, and F. Logist, Online model predictive control of industrial processes using low level control hardware: a pilot scale distillation column case study, *Control Engineering Practice* 28, 34–48 (2014).
[66] C. Muresan, C. Ionescu, E. Dulf, R. Rusu-Both, and S. Folea, Advantage of low cost predictive control: study case on a train of distillation columns, *Chemical Engineering and Technology* 41(10), 1936–1948 (2018).
[67] C. Pop, C. Ionescu, and R. De Keyser, Time delay compensation for the secondary processes in a multivariable carbon isotope separation unit, *Chemical Engineering Science* 80, 205–218 (2012).
[68] D. Copot, A. Maxim, R. De Keyser, and C. Ionescu, Multivariable control of sextuple tank system with non-minimum phase dynamics, in: IEEE Int Conf on Automation, Quality and Testing, Robotics, IEEE, 399–404, Cluj-Napoca, 19–21 May, Romania, 2016.