On the Heuristic Design of Common PI Controllers for Multi-Model Plants

Fotis N. Koumboulis
Department of Automation
Halkis Institute of Technology
34400, Psachna, Evia, Greece
koumboulis@teihal.gr

Abstract

A generic heuristic algorithm is introduced for the derivation of "common" PI controllers, that is PI controllers that achieve specific performance requirements simultaneously for all members of a set of linear models, called the target set. Common PI controllers find application for the control of processes with multi-linear description, since they can achieve safe and satisfactory performance for all switching events between the linear models of the target set. The heuristic algorithm is generic, simple to use and can be extended for other classes of controllers. The performance of the heuristic algorithm is illustrated for the case of a double effect evaporator process.

1. Introduction

Multi-linear model descriptions are used in many engineering applications, in order to describe switched systems. A switched (or hybrid) system consists of a number of subsystems, either continuous-time or discrete-time ordinary dynamic systems, and a rule that orchestrates the switching between them [1]-[5]. Switching can be activated by environmental factors, by control commands or by changes in the mode of operation of the process. For example, in the case of a wheeled mobile robot, switching between different dynamic models occurs when the motion of the wheels changes from rolling to sliding. Typical examples of such systems include batch processes, power systems, relay systems, transmission and stepper motors, internal combustion engine control, constrained robotics, etc.

Multi-linear model descriptions are also used in many cases as approximate models of nonlinear processes. Then each linear model can be considered as the linearization of the nonlinear model at a corresponding operating point of the process. The linear model is accurate in a range around the respective operating point. The union of all ranges is assumed to cover the total area of operation of the process. As the process trajectories move between the ranges of different operating points, the process description switches between the corresponding linear models of the multi-linear model.

A significant control objective for the control of multi-linear models is to design controllers that achieve satisfactory performance not only when the process trajectories lie in the range of a specific linear model, but also during switching between different linear models.

Many of the control techniques, which are proposed for controlling multi-linear models, are based on designing for each linear model a controller that achieves specific performance requirements and then propose a controller switching scheme activated by the process switching sequence. Switching control has attracted much research attention (see e.g. [6]–[12] and the references therein).

Multi-linear models may also be controlled with the application of gain scheduling, where a parameter varying feedback controller is applied, whose parameters vary on-line as functions of the operating conditions (e.g. [13]-[14] and the references therein). Scheduling variables are signals that indicate the current state of operating conditions, e.g. the measurement signal, the controller output signal, an external command, etc. When the variation of the parameters is scheduled by a fuzzy supervisor, we refer to fuzzy gain scheduling techniques (e.g. [15]-[16] and the references therein). Fuzzy gain scheduling appears in several approaches, e.g. changing the controller parameters according to the control error, or applying a fuzzy controller with Sugeno type rules.

Switching control as well as gain scheduling techniques require a supervisory scheme that decides on-line the required variation of the controller based on measurements of process signals, thus increasing the complexity of the controller. Moreover, design techniques based on on-line controller variation may appear operational problems in case of abrupt changes due to model switching or to controller switching.

The present work is co-financed by the Hellenic Ministry of Education and Religious Affairs' and the ESF of European Union within the framework of the "Operational Programme for Education and Initial Vocational Training" (Operation "Archimedes-")

0-7803-9402-X/05/$20.00 © 2005 IEEE

Control of multi-linear models can be significantly simplified by using "common" controllers, that is controllers which achieve specific performance requirements not only for one of the linear models of the target set, but for two or even more "adjacent" linear models. The term "adjacent" is used to denote models that are only one switch event away from each other. The use of common controllers reduces or even eliminates the need for on-line controller variations, and consequently for the corresponding scheduling supervisory scheme. Moreover, the performance of common controllers is independent from abrupt changes that may appear in process behaviour due to switching.

The derivation of common controllers is a difficult task that belongs to the field of robust control. In the field of robust control, a variety of control design problems have been solved, namely stabilizability, model matching, disturbance rejection, input-output decoupling and pole placement (e.g. [17]-[18] and the references therein). For some of these problems the set of controllers have been determined analytically and explicitly but many related problems remain to be solved.

The selection of the robust control algorithm that has to be applied for the derivation of common controllers is strongly related to the specific characteristics of the problem under consideration, e.g. the form and the degree of the nominal process model, the form and the extent of uncertainty, the desired design goal, etc. Thus any algorithm, based on robust control techniques, for the derivation of common controllers, should include a decision tree for the selection of the robust control technique to be applied dependent on the specific process characteristics. This would increase significantly the complexity of the algorithm.

The present paper proposes a heuristic approach to the design of "common" PI controllers, that is PI controllers that achieve specific performance requirements simultaneously for two or more models. The proposed algorithm for the derivation of common controllers is performed off-line, thus significantly reducing, in comparison with switching or gain scheduling techniques, the computational burden of the controller's implementation.

The proposed form of the algorithm concerns the case of multi-linear models, since this is the most common case in practice. However, it can be extended for multi-models of generic form. The proposed algorithm is generic, in the sense that is does not depend on the degree or the specific structure of the process model, or even the design goal under consideration. Moreover, the algorithm is very simple to use and it can be extended for the derivation of common controllers of other type, e.g. PID controllers, or other general order dynamic controllers. However, the increase of the number of controller parameters will increase the numerical complexity of the algorithm.

2. Process Description

Consider a process which can be described satisfactorily by a set of discrete-time linear models

$$S_j : A_j(q^{-1})y_j(k) = B_j(q^{-1})u_j(k) \ , \ j = 1,2,\ldots,m \quad (1)$$

where $A_j(q^{-1}), B_j(q^{-1})$ are polynomials of the delay operator q^{-1} and $y_j(k), u_j(k)$ are the fluctuations of the outputs and the inputs, respectively, of the linear model S_j. When the linear models S_j are derived through linearization of a nonlinear process around a corresponding operating point $\ell_j = (Y_j, U_j)$, $y_j(k)$ and $u_j(k)$ denote the deviations of the process output y and the process input u, respectively, from the corresponding operating point, that is $y_j(k) = y(k) - Y_j$ and $u_j(k) = u(k) - U_j$.

Let Ω_j denote a set of discrete-time PI controllers with incremental description of the general form

$$u_j(k) = u_j(k-1) + c_j e_j(k) + g_j e_j(k-1) \quad (2)$$

where the controller parameters c_j and g_j are selected in a way that the closed-loop system derived by the application of the controller (2) to the linear model S_j satisfies a set of design requirements, let say $\wp_j$. Note that $e_j(k) = w_j(k) - y_j(k)$ denotes the error between the reference signal w_j and the output y_j of S_j.

To help the reader to clarify the form of the performance requirements $\wp_j$ characterizing the set of admissible controllers for each of the corresponding linear models, we will present the following indicative proposition:

$\wp_j =$

{

a) The settling time of the step response of the closed loop system must be p_1 % better than the corresponding settling of the linear model S_j.

b) The overshoot of the step response of the closed loop system is less than p_2 %.

c) The steady state gain of the closed-loop system is equal to 1.

d) The magnitude of the maximum eigenvalue of the closed loop system is not allowed to be more than p_3 % larger than the corresponding of the linear model S_j.

}

The values of p_1, p_2, p_3 are set by the designer according to the desired characteristics for the closed-loop process. Note the fact that the eigenvalues of the closed-loop system may be slightly larger than the corresponding one of the linear model S_j, since the PI controller cannot achieve arbitrary pole assignment for linear models with order greater than 1. However, this fact does not affect significantly the response of the closed loop process, since the settling time of the step response of the closed loop system will be improved.

The above requirements (a-d) in $\wp_j$ appear to be some of the most common control design requirements. However, the proposed algorithm can easily be applied for other performance requirements, such as stability improvement, command following and/or disturbance attenuation through H_∞ tests. Checking whether a candidate controller satisfies the performance requirements can be performed either in an analytic way or using simulations. The latter case provides the possibility to extend significantly the class of performance requirements, as well as the class of multi-model plants under consideration.

Note, that the incremental form of the PI controller can be derived through discretization of a continuous-time PI description of the form

$$u_j(t) = K_j e_j(t) + \frac{K_j}{T_{I,j}} \int_0^t e_j(\tau) d\tau \tag{3}$$

For example, when the discrete-time description (2) is derived through backward discretization, the parameters c_j, g_j are related with the proportional gain K_j, the parameter $T_{I,j}$ and the sampling period T through the equations

$$c_j = K_j \left(1 + \tfrac{T}{T_{I,j}}\right) \; , \; g_j = -K_j \tag{4}$$

Note that since $T > 0$ and $T_{I,j} > 0$, equations (4) imply that the parameters c_j and $-g_j$ have always the same sign with the proportional gain K_j and moreover $|g_j| < |c_j|$. Similar relations hold when other discretization methods are used.

3. Common PI Controller Search Algorithm

Consider, now, a set

$$\overline{S} = \left\{ S_{j_i}, i = 1, \ldots, \mu \right\} \tag{5}$$

of "adjacent" operating models, called the *target set*. The target set $\overline{S}$ is a subset of the multi-linear model. The term "adjacent" is used to denote that for any S_{j_i} of the target set $\overline{S}$, there exists at least one model S_{j_n} of $\overline{S}$ that is only one switch event away from S_{j_i}, that is there are single switch events during which the process switches from S_{j_i} to S_{j_n} and/or vice versa.

Consider, now, that a PI controller of the form (2) can be found, that satisfies the performance requirements $\wp_{j_1} \cup \cdots \cup \wp_{j_\mu}$. The application of such a controller to the process described by the multi-linear model (1) would achieve: a) satisfactory performance of the corresponding closed-loop system within the range of validity of each linear model S_{j_i} of the target set $\overline{S}$ and b) safe and satisfactory performance of the corresponding closed-loop system for all transitions between any two models of $\overline{S}$.

In order to determine a controller satisfying the requirements $\wp_{j_1} \cup \cdots \cup \wp_{j_\mu}$, it is necessary to determine the set $\Omega = \Omega_{j_1} \cap \cdots \cap \Omega_{j_\mu}$ (or at least a subset of Ω). In the following a heuristic algorithm is presented for the determination of the set Ω, when the applied controller is a PI controller in discrete-time incremental form (2).

The algorithm searches within rectangles of the (c, g)-plane, where

$$u(k) = u(k-1) + ce(k) + ge(k-1) \tag{6}$$

is the description of the common controller to be determined, where $e = w - y$ and u, y, w denote the input, the output and the external command of the process, respectively. The incremental form of the PI controller can be used indeed as a common controller, since the variation $\Delta u(k) = u(k) - u(k-1)$ of the process input, as well as the error e between the external command and the process output remains the same, whatever the linear model that describes the process at each specific instant of time. However, special considerations have to be made in order to use other type of controllers as common controllers for more than one model.

At the first steps of the proposed algorithm, search is performed within an initial search area of the form

$$P = \left\{ (c, g), c_{\min} \le c \le c_{\max}, g_{\min} \le g \le g_{\max} \right\} \tag{7}$$

the determination of which will be further discussed in the following. The points of search on the (c, g)-plane are determined by a web of $(N_{1,0} + 1) \times (N_{2,0} + 1)$ points, where $N_{1,0}$ and $N_{2,0}$ are parameters of the algorithm. Thus the search step in the c-direction is equal to $\delta_c = (c_{\max} - c_{\min}) / N_{1,0}$, while the search step in the g-direction is equal to $\delta_g = (g_{\max} - g_{\min}) / N_{2,0}$. The search within the initial area will be repeated twice by duplicating the density of the web. If this second search also fails to determine a set of common controllers, the algorithm determines for each linear model S_{j_i} the rectangles

$$R_{j_i} = \{(c,g) : c_{j_i,\min} \le c \le c_{j_i,\max}, g_{j_i,\min} \le g \le g_{j_i,\max}\} \quad (8)$$

within which controllers that satisfy the performance requirements $\wp_{j_i}$ have been found. Then the algorithm proceeds with searching repeatedly within a rectangle, which will be called the *union rectangle* and is determined as the smallest rectangle that includes in its interior all the rectangles R_{j_i}, $i = 1,\ldots,\mu$. The density of the web is duplicated at each repetition of the search. Moreover, at each repetition of the search within the union rectangle, the *intersection rectangle* of the rectangles R_{j_i} is determined and compared with the corresponding intersection rectangle determined at the previous repetition. The search within the union rectangle will be repeated until a set Ω of common controllers is determined, or until the size of the intersection rectangle does no longer increase. This repeated search within the union rectangle intends to determine with a satisfactory accuracy the intersection rectangle, that is the area inside which common controllers are expected to be found. It is important to determine the whole extent of the intersection area, since otherwise points of the (c,g)-plane corresponding to common controllers may be missed.

Once the intersection rectangle has been determined, the algorithm proceeds with searching within the intersection rectangle. The search within the intersection rectangle will be repeated twice, with a web of double density at the second repetition. If the algorithm fails to determine a set of common controllers after two consecutive searches within the intersection rectangle, then the algorithm stops. In any case, the algorithm will stop if any of the following occurs: a) if a set of common controllers has been found, b) if the steps δ_c and δ_g become smaller than a threshold value ε, or if the total number of search repetitions exceeds a maximum value $I_{\max}$.

The density of the web, determined by the parameters $N_{1,0}$ and $N_{2,0}$, is an important design parameter of the algorithm, since large values of these parameters could result in a very time consuming algorithm, while small values could result to failure of the algorithm.

The heuristic algorithm in pseudo-code form is presented in the following.

Common PI Heuristic Algorithm

Data:

Linear models: $S_{j_i}, i = 1,\ldots,\mu$

Design Specifications: $\wp_{j_i}, i = 1,\ldots,\mu$

Initial Area of Search:

$$P = \{(c,g), c_{\min} \le c \le c_{\max}, g_{\min} \le g \le g_{\max}\}$$

Initial grid parameters: $N_{1,0}, N_{2,0}$

Maximum Accuracy: ε

Maximum Number of Iterations: $I_{\max}$

Step 0: Set $h_1 = 0, h_2 = 0$, $h_3 = 0$, $h_4 = 0$, $R = \varnothing$

Step 1: Set $N_1 = N_{1,0}$, $N_2 = N_{2,0}$

Step 2: Set

$$\delta_c = (c_{\max} - c_{\min}) / N_1,$$
$$\delta_g = (g_{\max} - g_{\min}) / N_2,$$
$$h_2 = h_2 + 1$$

Step 3:

$$c_{j_i,\min} = c_{\max}, c_{j_i,\max} = c_{\min}, i = 1,\ldots,\mu$$
$$g_{j_i,\min} = g_{\max}, g_{j_i,\max} = g_{\min}, i = 1,\ldots,\mu$$
$$\Omega = \varnothing,\ \Omega_{j_i} = \varnothing, i = 1,\ldots,\mu$$

Step 4:

For $\kappa = 0,\ldots,N_1$

For $\lambda = 0,\ldots,N_2$

For $i = 1,\ldots,\mu$

Check if the PI controller with parameters $c = c_{\min} + \kappa\delta_c$ and $g = g_{\min} + \lambda\delta_g$ satisfy the requirements $\wp_{j_i}$. If yes then

a) if $c < c_{j_i,\min}, c_{j_i,\min} = c$,

b) if $c > c_{j_i,\max}, c_{j_i,\max} = c$,

c) if $g < g_{j_i,\min}, g_{j_i,\min} = g$,

d) if $g > g_{j_i,\max}, g_{j_i,\max} = g$,

e) $\Omega_{j_i} = \Omega_{j_i} \cup \{(c,g)\}$

End (if)

End (for)

If all requirements $\wp_{j_i}$, $i = 1,\ldots,\mu$ were found to be satisfied then $\Omega = \Omega \cup \{(c,g)\}$.

End (for)

End (for)

Step 5: If $\Omega \ne \varnothing$ or $\min\{\delta_c, \delta_g\} < \varepsilon$ or $h_2 = I_{\max}$, then Stop.

Step 6: If $\Omega = \varnothing$ and $h_1 < 1$ then

$h_1 = h_1 + 1$, $N_1 = 2N_{1,0}$, $N_2 = 2N_{2,0}$

Go to Step 2.

End (if)

Step 7: If $\Omega_{j_i} = \varnothing$ for any $i = 1,\ldots,\mu$ and $h_1 = 1$ then Stop.

Step 8: If $h_4 = 1$ then Stop.

Step 9: If $\Omega_{j_i} \ne \varnothing$ for all $i = 1,\ldots,\mu$ then determine the rectangles

$$R_{j_i} = \{(c,g) : c_{j_i,\min} \le c \le c_{j_i,\max}, g_{j_i,\min} \le g \le g_{j_i,\max}\}$$

If $R = \varnothing$ or $R \subset R_{j_1} \cap \cdots \cap R_{j_\mu}$ then

a) $R = R_{j_1} \cap \cdots \cap R_{j_\mu}$

b) $c_{\min} = \min_{i=1,\ldots,\mu}\{c_{j_i,\min}\}, c_{\max} = \max_{i=1,\ldots,\mu}\{c_{j_i,\max}\}$

c) $g_{\min} = \min_{i=1,\ldots,\mu}\{g_{j_i,\min}\}, g_{\max} = \max_{i=1,\ldots,\mu}\{g_{j_i,\max}\}$

d) $P = \{(c,g) : c_{\min} \le c \le c_{\max}, g_{\min} \le g \le g_{\max}\}$

e) $N_1 = 2^{h_3} N_{1,0}, N_2 = 2^{h_3} N_{2,0}$

f) $h_3 = h_3 + 1$

g) Go to Step 2.

End (if)

End (if)

Step 10: If $R_{j_1} \cap \cdots \cap R_{j_\mu} \neq \varnothing$ then

a) $h_1 = 0$, $h_4 = 1$

b) $c_{\min} = \max_{i=1,\ldots,\mu}\{c_{j_i,\min}\}, c_{\max} = \min_{i=1,\ldots,\mu}\{c_{j_i,\max}\}$

c) $g_{\min} = \max_{i=1,\ldots,\mu}\{g_{j_i,\min}\}, g_{\max} = \min_{i=1,\ldots,\mu}\{g_{j_i,\max}\}$

d) $P = \{(c,g) : c_{\min} \le c \le c_{\max}, g_{\min} \le g \le g_{\max}\}$

e) Go to Step 1.

End (if)

Step11: If $R_{j_1} \cap \cdots \cap R_{j_\mu} = \varnothing$ then Stop.

3.1. Initialization of the algorithm

As already mentioned, the common PI heuristic algorithm requires the knowledge of an initial area of search within the (c,g)-plane, inside which the algorithm will start seeking for common controllers. The initial search area represents an estimation, that should be available before the implementation of the algorithm, of the intervals inside which the common controller parameters are expected to be found.

The determination of this initial search area should be based on any available a priori information about the process. For example, in case a safe but poorly tuned PI controller can be derived for the process through standard tuning techniques (e.g. Ziegler-Nichols), the initial area of search may be selected as a sufficiently large area around the parameter values of this controller.

Moreover, the derivation of the initial search area should exploit all available information about the controller parameters. For example, when the discrete-time incremental form (2) of the PI controller is derived through backward discretization of a corresponding continuous-time description, the parameter g of the discrete-time description must satisfy the condition $|g| < |c|$ (recall the relations between c, g and the parameters K, T_I, T). Thus, in order to determine the initial search area, it suffices to determine an upper bound, let $c_{\max}$, for the absolute value of the parameter c. In the special case, being the most common in practice, namely when the sign of the dc gain of the process remains the same, then the proportional gain K of the PI controller should also keep the same sign. In this case, the initial area of search of the algorithm can be determined taking into account that the signs of the parameters c and g are predetermined.

Consider now the case when the process input is subject to actuator constraints of the form

$$|\Delta u(k)| = |u(k) - u(k-1)| \le \varepsilon_u \tag{9}$$

which implies that the change of the input variable between two consecutive time instants can not exceed a threshold value ε_u. Then, assuming that the error $e(k)$ is equal to zero for $k < 0$, and $|e(k)| \le \varepsilon_e$ for $k \ge 0$, the upper bound $c_{\max}$ can be selected as

$$c_{\max} = \frac{\varepsilon_u}{\varepsilon_e} \tag{10}$$

This selection guarantees that at the first time instant ($k = 0$) of the controller's application, the variation Δu of the process input will not exceed the maximum value ε_u, provided that $e(k)$ is equal to zero for $k < 0$, and $|e(0)| \le \varepsilon_e$. The selection of the threshold value ε_e may be based on the available knowledge about the extent of the considered area of operation of the process, as well as the external commands that may be used.

4. Application for a Double Effect Evaporator

To illustrate our results in brevity, we will present the application of the proposed algorithm for the derivation of common controllers for three adjacent operating points of a typical industrial process, that is a double effect evaporator process with short-tube vertical calandria-type units. The double effect evaporator process is described in [19] and [20]. The feed solution which is pumped to the first effect is heated by a saturated steam flow rate u, which is the input of the process. The solution produced by the first effect is fed to the second effect, where it is in turn heated by the vapor flow produced by the first effect. The solution concentration produced by the second effect constitutes the output y of the process. The double effect evaporator process is a stable nonlinear process. Consider now that identification is performed around the three nominal operating points, deriving the following three corresponding linearized second order discrete-time models:

Operating Point $\ell_1 = [Y_1, U_1] = [0.09395, 1.34]$

$$S_1 : \begin{array}{c} y_1(k) - 1.7619 y_1(k-1) + 0.7756 y_1(k-2) = \\ 0.0045 u_1(k-1) - 0.0032 u_1(k-2) \end{array}$$

where $y_1 = y - Y_1$ and $u_1 = u - U_1$.

Operating Point $\ell_2 = [Y_2, U_2] = [0.10331, 1.43]$

$$S_2 : \begin{array}{c} y_2(k) - 1.7769 y_2(k-1) + 0.7888 y_2(k-2) = \\ 0.0051 u_2(k-1) - 0.0038 u_2(k-2) \end{array}$$

where $y_2 = y - Y_2$ and $u_2 = u - U_2$.

Operating Point $\ell_3 = [Y_3, U_3] = [0.11066, 1.49]$

$$S_3: \begin{array}{c} y_3(k) - 1.7870 y_3(k-1) + 0.7978 y_3(k-2) = \\ 0.0057 u_3(k-1) - 0.0042 u_3(k-2) \end{array}$$

where $y_3 = y - Y_3$ and $u_3 = u - U_3$.

The sampling period is $T = 10[\min]$.

The set of admissible controllers for each of the corresponding linearized models is determined to satisfy the specifications expressed as the following set of propositions:

$\wp_i =$

{

a) The settling time of the step response of the closed loop system must be 15% better than the corresponding settling time of the linearized model S_i.

b) The overshoot of the step response of the closed loop system is less than 5%.

c) The steady state gain of the closed-loop system is equal to 1.

d) The magnitude of the maximum eigenvalue of the closed loop system is not allowed to be more than 1% larger than the corresponding of the linearized model S_i.

}

The above specifications can be achieved by applying PI controllers, with discrete-time description of the form (2).

The determination of the set $\Omega_1 \cap \Omega_2 \cap \Omega_3$ of common controllers for the linear models S_1, S_2 and S_3 can be done with the use of the heuristic algorithm presented in Section 3.

The initial area of search is determined according to the following. The dc gain of the double effect evaporator is positive. Thus, the proportional gain K as well as the parameter c of the discrete-time description of the PI controller must be also positive, while the parameter g of the discrete-time description must be negative and with $|g| < c$ (recall the relations between c, g and the parameters K, T_I, T). Considering an actuator restriction of the form (9) with $\varepsilon_u = 0.3$, as well as the error e to be bounded by $\varepsilon_e = 0.02506$, the initial search area for the set $\Omega_1 \cap \Omega_2 \cap \Omega_3$ of common controllers is determined according to (10) to be: $P = \{(c,g) \in \mathbb{R}^2 : 0.0001 \le c \le 11.9697, -11.9696 \le g \le -0.0001\}$. Note that the selection of ε_e is based on an estimation of the expected difference between the output value of the initial operating point of the process and the command signal. For example, if the external command is a step signal whose value indicates the desired target operating point, then the distance between the output value of the target and the initial operating point should be less than or equal to $\varepsilon_e = 0.02506$.

The initial web parameters of the heuristic algorithm are $N_{1,0} = N_{2,0} = 100$. The heuristic algorithm determines a non empty set of common controllers at the first repetition of the search within the intersection rectangle. The results of the search are presented in Figures 1 and 2. As shown in Figure 2, the heuristic algorithm determines five common controllers:

1) $c = 8.5594,\ g = -7.7475$

2) $c = 8.5768,\ g = -7.7634$

3) $c = 8.5941,\ g = -7.7793$

4) $c = 8.6114,\ g = -7.7952$

5) $c = 8.6287,\ g = -7.8111$

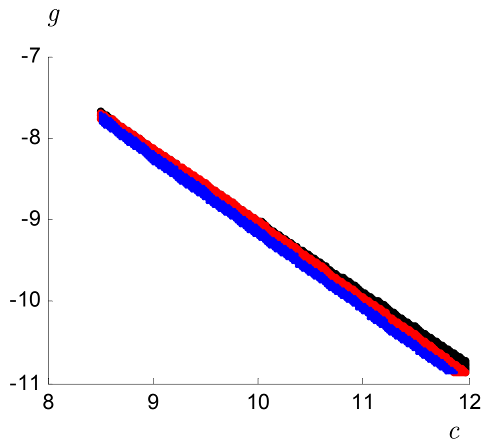

Figure 1. Results of Search Algorithm for Ω_1 (black dots), Ω_2 (red dots) and Ω_3 (blue dots) within the intersection rectangle

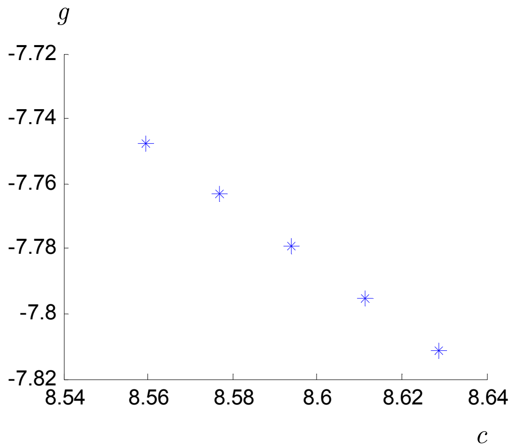

Figure 2. Results of Search Algorithm for $\Omega_1 \cap \Omega_2 \cap \Omega_3$

In the following, we select to use the controller that corresponds to $c = 8.5594,\ g = -7.7475$. Figures 3-6 present the simulation results derived from the application of the aforementioned controller, when the double effect evaporator process is described by the following nonlinear state equation ([19], [20])

$$\dot{x}_1(t) = d_1 F_0 (C_0 - x_1(t)) + d_2 x_1(t) u(t)$$
$$\dot{x}_2(t) = d_3 F_0 (x_1(t) - x_2(t)) + (d_4 x_1(t) + d_5 x_2(t)) u(t)$$
$$y(t) = x_2(t)$$

where x_1 and x_2 are the output concentrations of the first and the second effect respectively, $F_0 = 2.525[\mathrm{kg/min}]$ is the feed flow to the first effect, $C_0 = 0.04[\mathrm{kg(sugar)/kg(water)}]$ is the feed concentration to the first effect and $d_1, \ldots, d_5$ are parameters of the process with values $d_1 = 0.010526[1/\mathrm{kg}]$, $d_2 = 0.008510[1/\mathrm{kg}]$, $d_3 = 0.009524[1/\mathrm{kg}]$, $d_4 = -0.007700[1/\mathrm{kg}]$, and $d_5 = 0.010306[1/\mathrm{kg}]$.

Figure 3 presents the closed-loop response when the process trajectories move from operating point ℓ_1 to operating point ℓ_3, while Figure 4 presents the corresponding controller output. Figures 5 and 6 present the corresponding signals when moving from ℓ_3 back to ℓ_1. Note that the controller output is applied to the process with the use of a zero-order hold. In Figures 3 and 5, we also present for comparison reasons the corresponding response of the nonlinear model for input function determined, respectively, by $u(t) = U_3$ and $u(t) = U_1$ for $t \geq 0$.

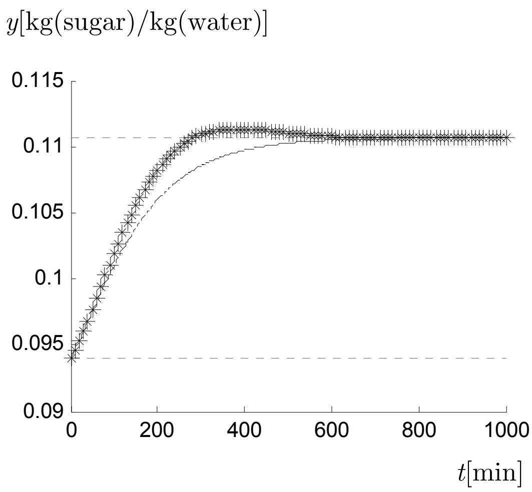

Figure 3. Open loop (-) and closed loop trajectories (*) from ℓ_1 to ℓ_3

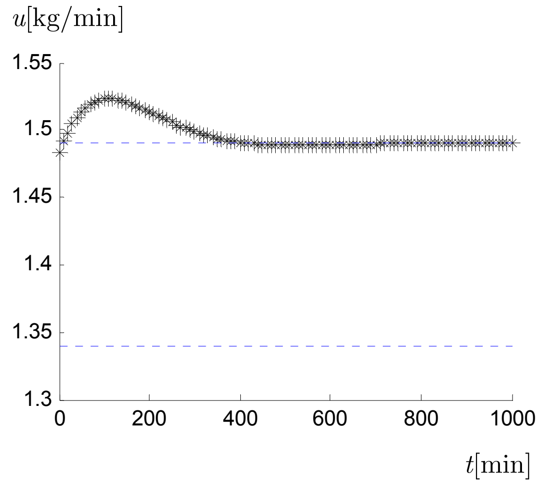

Figure 4. Controller output for the transition from ℓ_1 to ℓ_3

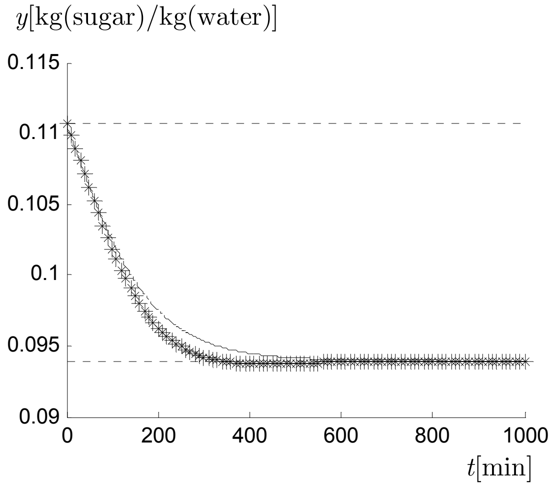

Figure 5. Open loop (-) and closed loop trajectories (*) from ℓ_3 to ℓ_1

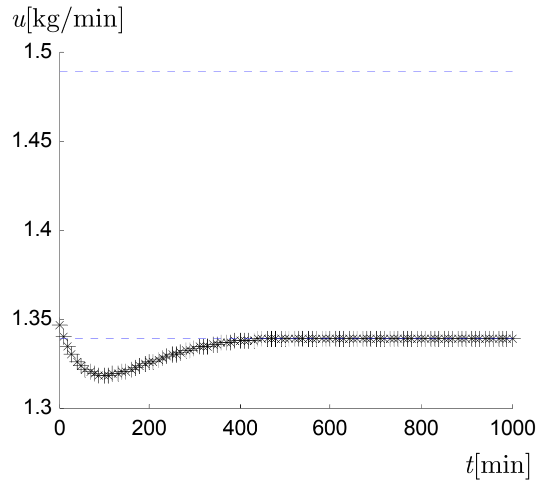

Figure 6. Controller output for the transition from ℓ_3 to ℓ_1

5. Conclusions

A heuristic algorithm has been introduced for the derivation of "common" PI controllers, that is PI controllers that achieve specific performance requirements simultaneously for all members of a set of linear models, called the target set. Common PI controllers find application for the control of processes, with multi-linear description, since they can achieve safe and satisfactory performance for all switching events between the linear models of the target set. The heuristic algorithm is generic, simple to use and can be extended for other classes of controllers, as well as for multi-model plants of generic form. The extension of the algorithm for other types of controllers and systems is currently under investigation. The performance of the heuristic algorithm is illustrated for the case of a double effect evaporator process.

References

[1] J. Zhao, G. Dimirovski, "Quadratic stability of a class of switched nonlinear systems", *IEEE Trans. On Automatic Control*, vol, 49, no.4, pp. 574-578, 2004.

[2] D. Liberzon, J. P. Hespanha and A. Stephen Morse, "Stability of switched systems: a Lie-algebraic condition", *Systems & Control Letters*, vol. 37, no. 3, pp. 117-122, 1999

[3] D. Liberzon and A.S. Morse, "Basic problems in stability and design of switched systems", *IEEE Control Systems Magazine*, vol. 19, pp. 59-70, Oct. 1999.

[4] X. Xu and P. J. Antsaklis, "Stabilization of second-order LTI switched systems", *Proceedings of the 38th IEEE Conference on Decision and Control*, pp. 1339-1344, 1999

[5] M. Egerstedt and V. D. Blondel, "How hard is it to control switched systems?", *Proceedings of the 2002 American Control Conference*, pp. 1869-1873, 2002

[6] P.V. Zhivoglyadov, R.H. Middleton, "Switching controller design via convex polyhedral Lyapunov fumctions", *Automatica*, vol. 38, pp. 1439-1448, 2002.

[7] A. Leonessa, W. M. Haddad and V. S. Chelllaboina, "Nonlinear system stabilization via hierarchical switching control", *IEEE Trans. on Automatic Control*, vol. 46, no. 1, pp. 17-28, 2001

[8] D. Prattichizzo and D. Borrelli, "Supervisory switching control in robotic manipulation", *Proceedings of the 38th IEEE Conference on Decision and Control*, pp. 2957-2962, 1999

[9] Biao Yu and P. R. Pagilla, "A switching control scheme for constrained robot tasks", *Proceedings of the 2001 IEEE International Conference on Control Applications*, pp. 1105-1110, 2001

[10] R.E. King, F.N. Koumboulis and A. Stathaki, "Intelligent hybrid industrial control", *Proceedings of the 2002 First International IEEE Symposium Intelligent Systems*, pp. 2-6, 2002.

[11] A. Stathaki, F.N. Koumboulis and R. E. King, "An application of logic-based switching for a class of hybrid industrial controllers" *Proceedings of the 12th IEEE Mediterranean Conference on Control & Automation*, Kusadasi, Aydin, Turkey, June 6-9, 2004

[12] F. N. Koumboulis, M. P. Tzamtzi, G. E. Chamilothoris, "Iterative feedback tuning safe switching controllers", 13th *IEEE Mediterranean Control Conference on Control & Automation (MED 2005)*, Limassol, Cyprus, June 2005, pp. 938-945.

[13] D. A. Lawrence and W. J. Rugh, "Gain scheduling dynamic linear controllers for a nonlinear plant", *Automatica*, vol. 31, pp. 381-390, 1995.

[14] S.-H. Lee and J.-T. Lim, "Fast gain scheduling on tracking problems using derivative information", *Automatica*, vol. 33, pp. 2265-2268, 1997.

[15] Z. Y. Zhao, M. Tomizuka, S. Isaka, "Fuzzy gain scheduling of PID controllers", *IEEE Trans. on Systems Man and Cybernetics*, vol. 23, no. 5, pp. 1392-1398, 1993.

[16] P. Viljamaa, H. N. Koivo, "Fuzzy logic in PID gain scheduling", *Third European Congress on Fuzzy and Intelligent Technologies EUFIT'95,* Aachen, Germany, August 28-31, 1995, ELITE-foundation, Vol. 2, pp. 927-931, 1995.

[17] F. N. Koumboulis, M. G. Skarpetis, B. G. Mertzios, "Robust regional stabilisation of an electropneumatic actuator", *IEE Proceedings - Control Theory and Applications*, vol. 145, no. 2, pp. 226-230, 1998.

[18] F. N. Koumboulis, M. G. Skarpetis, Robust triangular decoupling with application to 4WS cars, *IEEE Trans. on Automatic Control*, vol. 45, no. 2, Feb. 2000, pp. 344-352, 2000.

[19] H. Sira-Ramirez, O. Llanes-Santiago, "Dynamical discontinuous feedback strategies in the regulation of nonlinear chemical processes"*, IEEE Trans. on Control Systems Technology*, vol. 2, no. 1, pp. 11-21, 1994.

[20] A. Montano, G. Silva, "Design of a nonlinear control for a double effect evaporator", pp. 2256-2261, *European Control Conference* (*ECC 1991)*, Grenoble, France, July 2-5, 1991.