Nonlinear Optimal Generalized Predictive Functional Control of Piecewise Affine Systems

Sultan Alotaibi, M. Grimble and L. Cavanini

***Abstract*—An algorithm for nonlinear optimal generalized predictive functional control is defined for controlling discrete-time piecewise affine systems. The main piecewise affine form is transferred into a corresponding state-dependent system form. The principal piecewise affine system's hybrid properties are retained, and both the continuous and switching dynamics are combined in the same system description. This method enables the use of nonlinear generalized predictive functional control for this hybrid system class. In the generalized predictive functional control method, different classical controller structures can be employed in the feedback loop, such as PI, PID or other classical transfer-function structures. The loop controller is defined here to have a PI structure, and its selected linear transfer-functions set is multiplied by gains that are found to minimize a GPC type of cost-index. The simulation results are shown using a model of a continuous stirred tank reactor.**

I. INTRODUCTION

Systems represented by continuous and discrete signals are called Hybrid dynamical Systems (HS) and increasingly have sparked interest from the computer science and control communities. Numerous authors have concentrated on some HS subclasses in which the analysis and/or the control design methods have been established. Some of these subclasses are Linear Complementarity (LC) systems [1], [2], Mixed Logical Dynamical (MLD) systems [3], first-order linear HS with saturation [4], Piecewise Affine (PWA) systems [5], and Max-Min-Plus Scaling (MMPS) systems [6]. Computer researchers have suggested many models, such as the essential Hybrid Automata (HA) [7]. Equivalence relations among MLD, LC, MMPS, ELC and PWA form have been established satisfying moderate conditions in [8]. The equivalence relations between PWA systems and State-dependent (Sd) systems established in [9] and were extended in [10] to show the link between Sd systems and Linear Hybrid Automata (LHA) as in Fig. 1. Noting that an arrow pointing from class A to class B implies that A is a subset of B. and a star * sign on the arrow means some conditions are involved in the creation of the specified inclusion.

The initial work in [9], [11] used a Nonlinear Generalized Minimum Variance (NGMV) controller in controlling PWA systems, which was translated into Non-linear (NL) Sd system by inserting a custom logic state to be able to describe the intersection of switching sections. The merit of using the Sd model of PWA systems is because Sd systems are much simpler to construct since all state, input constraints and logic state can be incorporated in the Sd system model.

S. Alotaibi & M. Grimble are with Electronic & Electrical Engineering Dep., Uni. of Strathclyde, Glasgow, UK. Email:sultan.m.alotaibi@strath.ac.uk; m.j.grimble@strath.ac.uk
L. Cavanini is with Industrial Systems and Control Ltd, Culzean House, Glasgow, UK. Email: l.cavanini@isc-ltd.com.

This work is the Nonlinear Generalized Predictive Functional Control (NGPFC) [12], [13] extension for PWA systems. PWA is a main class of HS and a proper structure to describe or approximate various physical processes as approximating NL systems using multiple linearizations over different operating points. This work follows the previous methodology to control Sd systems by first obtaining the PWA system in Sd form and constructing the NGPFC, which is easy to calculate and implement. Noting that, hard constraints can be included using a quadratic programming solution.

This paper is arranged as: The augmented system description outlined in Section II. The PI controller design procedures for PWA systems is discussed in section III. Section IV shows a potential application in simulation, and Section V summarizes the paper.

II. SYSTEM DESCRIPTION

A. PWA Systems

The state-space model of delayed discrete-time PWA systems is described as:

$$\begin{aligned} x(t+1) &= A_i x(t) + B_i u(t-k) + D_i \xi(t) + f_i \\ y(t) &= C_i x(t) + E_i u(t-k) + g_i \end{aligned} \tag{1}$$

where $x \in R^n$, $u \in R^m$, $y \in R^p$ and $d \in R^n$ are the states, inputs, outputs and disturbances, respectively. The k represents the value of a common delay component.

Definition: every $(A_i, B_i, C_i, D_i, E_i)$, $i=1,\dots,s$ situated affine sub-system in the polyhedron cell $\Omega_i \subset R^n \times R^m$ are assumed a polyhedral set specified by matrices G_{ix}, h_{ix}, G_{iu} and h_{iu}:

$$\Omega_i = \left\{ \begin{bmatrix} x \\ u \end{bmatrix} \mid G_{ix} x \le h_{ix} \wedge G_{iu} u \le h_{iu} \right\} \tag{2}$$

The cells satisfy $\Omega_i \cap \Omega_j = \emptyset$, $\forall i \ne j$ and states along with inputs admissible set $\Omega = \cup_{i=1}^{s} \Omega_i$ is represented by their union.

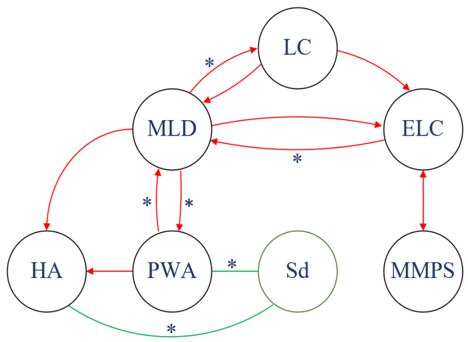

Fig. 1: Hybrid systems classes link

B. Sd System

The Sd system includes time-varying state equation matrices as they depend on states, inputs, external parameters and/or control signal. The system matrices can be described as:

$$\begin{aligned} x_{sd}(t+1) &= A\big(x(t),u(t)\big)x(t) + B\big(x(t),u(t)\big)u(t-k) \\ &\quad + D\big(x(t),u(t)\big)\xi(t) \\ y_{sd}(t) &= C\big(x(t),u(t)\big)x(t) + E\big(x(t),u(t)\big)u(t-k) \end{aligned} \tag{3}$$

Introduce supplementary logic variable $\delta_i(t)\in\{0,1\}$ as defined in (4):

$$\delta_i(t) = \begin{cases} 1 & if\ G_{ix}x(t) \le h_{ix} \wedge G_{iu}u(t) \le h_{iu} \\ 0 & otherwise \end{cases} \tag{4}$$

Then the PWA system in (1), with the segment in (2), can be transformed into this format:

$$\begin{aligned} x_{sd}(t+1) &= \sum_{i=1}^{s} \delta_i(t)\left[A_i x(t) + B_i u(t-k) + D_i\xi(t) + f_i\right] \\ y_{sd}(t) &= \sum_{i=1}^{s} \delta_i(t)\left[C_i x(t) + E_i u(t-k) + g_i\right] \end{aligned} \tag{5}$$

Noting that the bias term f_i and g_i in (1), can be considered as a measured disturbance in (3).

The logic variable values $\delta_i(t)\in\{0,1\}$ in (4) relying on the state $x(t)$ and the input $u(t)$ variables. Assigning, *less than or equal* ($\le$) function $LE\,(x, m)$ as:

$$LE(x,m) = \begin{cases} 1 & if\ x \le m \\ 0 & otherwise \end{cases}$$

where the limit $m\in R^n$. Then,

$$\delta_i(t) = \prod_j LE(G_{ix}^j x(t), h_{ix}^j) \prod_l LE(G_{iu}^l u(t), h_{iu}^l) \tag{6}$$

and j and l represent j^{th} and l^{th} rows, correspondingly.

To simplify the notation in (3), constitute $A_{sd}=A(x(t),u(t))$ and for matrices: B_{sd}, C_{sd}, D_{sd} and E_{sd}, then substituting (6) in (5), obtain:

$$\begin{aligned} x_{sd}(t+1) &= \overbrace{\left[\sum_{i=1}^{s}\prod_j LE(G_{ix}^j x(t), h_{ix}^j)\prod_l LE(G_{iu}^l u(t), h_{iu}^l)A_i\right]}^{A_{sd}} x_{sd}(t) \\ &+ \overbrace{\left[\sum_{i=1}^{s}\prod_j LE(G_{ix}^j x(t), h_{ix}^j)\prod_l LE(G_{iu}^l u(t), h_{iu}^l)B_i\right]}^{B_{sd}} u(t-k) \\ &+ \overbrace{\left[\sum_{i=1}^{s}\prod_j LE(G_{ix}^j x(t), h_{ix}^j)\prod_l LE(G_{iu}^l u(t), h_{iu}^l)D_i\right]}^{D_{sd}} \xi_{sd}(t) \end{aligned} \tag{7}$$

$$\begin{aligned} y_{sd}(t) &= \overbrace{\left[\sum_{i=1}^{s}\prod_j LE(G_{ix}^j x(t), h_{ix}^j)\prod_l LE(G_{iu}^l u(t), h_{iu}^l)C_i\right]}^{C_{sd}} x_{sd}(t) \\ &+ \overbrace{\left[\sum_{i=1}^{s}\prod_j LE(G_{ix}^j x(t), h_{ix}^j)\prod_l LE(G_{iu}^l u(t), h_{iu}^l)E_i\right]}^{E_{sd}} u(t-k) \end{aligned} \tag{8}$$

Consequently, the PWA system (1) is converted to the NL Sd system (9) that has the form in (3).

$$\begin{aligned} x_{sd}(t+1) &= A_{sd}x_{sd}(t) + B_{sd}u(t\text{-}k) + D_{sd}\xi_{sd}(t) \\ y_{sd}(t) &= C_{sd}x_{sd}(t) + E_{sd}u(t\text{-}k), \qquad x_{sd}(t) \in R^{n_{sd}} \end{aligned} \tag{9}$$

C. Total Linear Sub-System and Sd System

The general system described in [14] was used to obtain the control algorithm for Sd systems as in Fig. 2.

Reference Model:

$$\begin{aligned} x_r(t+1) &= A_r x_r(t) + D_r\omega(t), \qquad x_r(t) \in R^{n_r} \\ r(t) &= C_r x_r(t) \end{aligned} \tag{10}$$

Disturbance Model:

$$\begin{aligned} x_d(t+1) &= A_d x_d(t) + D_d\xi_d(t), \qquad x_d(t) \in R^{n_d} \\ d(t) &= C_d x_d(t) \end{aligned} \tag{11}$$

Error Weighting:

$$\begin{aligned} x_p(t+1) &= A_p x_p(t) + B_p\big(r(t) - d(t) - y_{sd}(t)\big), \ \ x_p(t) \in R^{n_p} \\ e_p(t) &= C_p x_p(t) + E_p\big(r(t) - d(t) - y_{sd}(t)\big) \end{aligned} \tag{12}$$

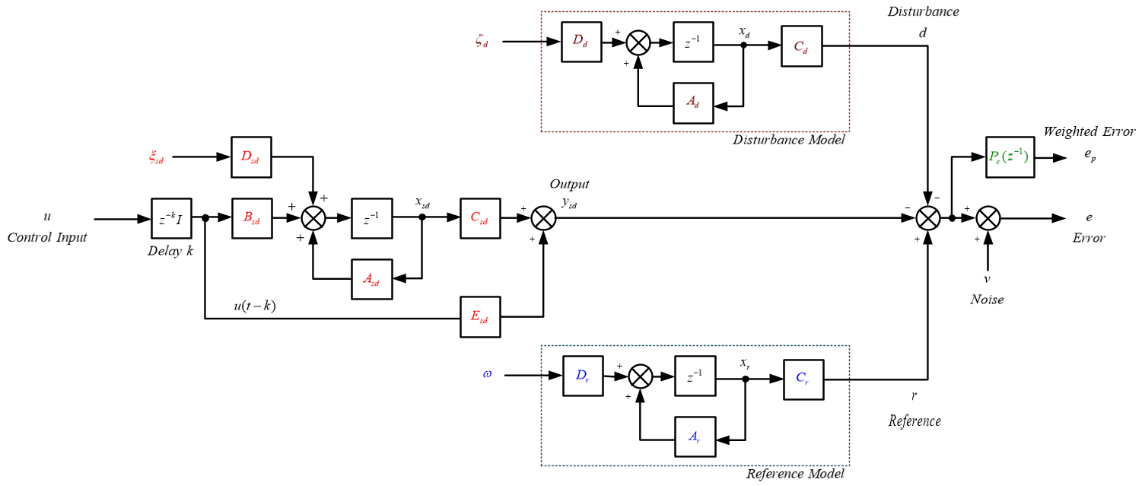

Fig. 2: System description

The error weighting in (12) is given using (9) as:

$$
\begin{aligned}
&x_p(t+1) = A_p x_p(t) \\
&+B_p\left(C_r x_r(t) - C_d x_d(t) - C_{sd} x_{sd}(t) - E_{sd} u(t-k)\right)
\end{aligned} \tag{13}
$$

$$
\begin{aligned}
&e_p(t) = C_p x_p(t) \\
&+E_p\left(C_r x_r(t) - C_d x_d(t) - C_{sd} x_{sd}(t) - E_{sd} u(t-k)\right)
\end{aligned} \tag{14}
$$

The linear sub-system augmented description can be obtained by augmenting the disturbance, the linear reference and error-weighting models as:

$$x_{ls}(t) = \begin{bmatrix} x_d^T(t) & x_p^T(t) & x_r^T(t) \end{bmatrix}^T$$

Noting the equation (10), (11) and (12), the linear sub-system augmented description is:

$$
\begin{aligned}
x_{ls}(t+1) = \\
&\overbrace{\begin{bmatrix} A_d & 0 & 0 \\ -B_p C_d & A_p & B_p C_r \\ 0 & 0 & A_r \end{bmatrix}}^{A_{ls}} \overbrace{\begin{bmatrix} x_d \\ x_p \\ x_r \end{bmatrix}}^{x_{ls}(t)} + \\
&\overbrace{\begin{bmatrix} 0 \\ -B_p \\ 0 \end{bmatrix}}^{B_{ls}} y_{sd}(t) + \overbrace{\begin{bmatrix} D_d & 0 \\ 0 & 0 \\ 0 & D_r \end{bmatrix}}^{D_{ls}} \overbrace{\begin{bmatrix} \xi_d(t) \\ \omega(t) \end{bmatrix}}^{\xi_{ls}(t)}
\end{aligned} \tag{15}
$$

Then the overall augmented state-vector of linear and Sd sub-system model can be outlined as:

$$x(t) = \begin{bmatrix} x_{sd}(t) \\ x_{ls}(t) \end{bmatrix}$$

Therefore, the augmented linear and the Sd sub-systems state-vector and the associated disturbances become:

$$
\begin{aligned}
x(t+1) &= A_t x(t) + B_t u(t\text{-}k) + D_t \xi(t) \\
y(t) &= C_t x(t) + E_t u(t\text{-}k) \\
z(t) &= y(t) + v(t)
\end{aligned} \tag{16}
$$

Clearly from equations (9), (13),(14) and (15), the augmented Sd and the weighted error system models format are:

$$
\begin{aligned}
x(t+1) = \\
&\overbrace{\begin{bmatrix} A_{sd} & 0 & 0 & 0 \\ 0 & A_d & 0 & 0 \\ -B_p C_{sd} & -B_p C_d & A_p & B_p C_r \\ 0 & 0 & 0 & A_r \end{bmatrix}}^{A_t} \overbrace{\begin{bmatrix} x_{sd} \\ x_d \\ x_p \\ x_r \end{bmatrix}}^{x(t)} + \\
&\overbrace{\begin{bmatrix} B_{sd} \\ 0 \\ -B_p E_{sd} \\ 0 \end{bmatrix}}^{B_t} u(t\text{-}k) + \overbrace{\begin{bmatrix} D_{sd} & 0 & 0 \\ 0 & D_d & 0 \\ 0 & 0 & 0 \\ 0 & 0 & D_r \end{bmatrix}}^{D_t} \overbrace{\begin{bmatrix} \xi_{sd}(t) \\ \xi_d(t) \\ \omega(t) \end{bmatrix}}^{\xi(t)}
\end{aligned} \tag{17}
$$

$$y(t) = \overbrace{\begin{bmatrix} C_{sd} & 0 & 0 & 0 \end{bmatrix}}^{C_t} \overbrace{\begin{bmatrix} x_{sd} \\ x_d \\ x_p \\ x_r \end{bmatrix}}^{x(t)} + \overbrace{E_{sd}}^{E_t} u(t\text{-}k) \tag{18}$$

$$
\begin{aligned}
e_p(t) = &\overbrace{\begin{bmatrix} -E_p C_{sd} & -E_p C_d & C_p & E_p C_r \end{bmatrix}}^{C_{pt}} \overbrace{\begin{bmatrix} x_{sd} \\ x_d \\ x_p \\ x_r \end{bmatrix}}^{x(t)} \\
&+\overbrace{\begin{bmatrix} -E_p E_{sd} \end{bmatrix}}^{E_{pt}} u(t-k)
\end{aligned} \tag{19}
$$

D. State-Space Prediction Models

The prediction of the weighted error is needed for the control solution. Both the states and weighted error future values at time *t* may be found by iterative utilization of the state model in (17) and the *i-steps-ahead* state is acquired as:

$$
\begin{aligned}
&x(t+i) = A_t^i x(t) \\
&+\sum_{j=1}^{i} A_{t+j}^{i-j}\left(B_{t+j-1} u(t+j-1-k) + D_{t+j-1}\xi(t+j-1)\right)
\end{aligned} \tag{20}
$$

The controlled weighted error e_p at future times $i \geq 0$ follows similarly as:

$$
\begin{aligned}
&e_p(t+i) = C_{pt+i} A_t^i x(t) \\
&+C_{pt+i}\sum_{j=1}^{i}\begin{pmatrix} A_{t+j}^{i-j} B_{t+j-1} u(t+j-1-k) \\ +D_t \xi(t+j+k-1) \end{pmatrix} \\
&+E_{pt+i} u(t+i-k)
\end{aligned} \tag{21}
$$

Equation (21) can be introduced in vector-matrix notation as:

$$
\begin{aligned}
&\overbrace{\begin{bmatrix} e_p(t+k) \\ e_p(t+1+k) \\ e_p(t+2+k) \\ \vdots \\ e_p(t+N+k) \end{bmatrix}}^{E_{P_{t+k,N}}} = \overbrace{\begin{bmatrix} C_{pt} I \\ C_{pt} A_t \\ C_{pt} A_t^2 \\ \vdots \\ C_{pt} A_t^N \end{bmatrix}}^{C_{PN} A_N} x(t+k) + \\
&\overbrace{\begin{bmatrix} E_{pt} & 0 & \cdots & 0 & 0 \\ C_{pt} B_t & E_{pt} & \ddots & 0 & 0 \\ \vdots & C_{pt} B_t & \ddots & \vdots & 0 \\ C_{pt} A_t^{N-1} B_t & \vdots & \ddots & E_{pt} & \vdots \\ C_{pt} A_t^{N-1} B_t & C_{pt} A_t^{N-2} B_t & \cdots & C_{pt} B_t & E_{pt} \end{bmatrix}}^{V_{PN}} \overbrace{\begin{bmatrix} u(t) \\ u(t+1) \\ \vdots \\ \vdots \\ u(t+N-1) \end{bmatrix}}^{U_{t,N}} + \\
&\underbrace{\begin{bmatrix} 0 & 0 & \cdots & 0 \\ C_{pt} D_t & 0 & \ddots & 0 \\ C_{pt} A_t D_t & C_{pt} D_t & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ C_{pt} A_t^{N-1} D_t & C_{pt} A_t^{N-2} D_t & \cdots & C_{pt} D_t \end{bmatrix}}_{C_{PN} D_N} \underbrace{\begin{bmatrix} \xi(t+k) \\ \xi(t+1+k) \\ \vdots \\ \vdots \\ \xi(t+N-1+k) \end{bmatrix}}_{W_{t+k,N}}
\end{aligned} \tag{22}
$$

E. Kalman Filter

The estimated weighted error (22) is arranged in the next *N+1* vector structure (23).

$$\hat{E}_{P_{t+k,N}} = \overbrace{\overbrace{C_{P_{t+k,N}} A_{t+k,N}}^{C_{PN}A_N} \hat{x}(t+k\,|\,t)}^{\tilde{D}_{P_{t+k,N}}} + \overbrace{(C_{P_{t+k,N}} B_{t+k,N} + E_{P_{t+k,N}})}^{V_{PN}} U_{t,N} \tag{23}$$

where $\hat{x}(t+k|t)$ are a least-squares state-estimate resulted from Time-Varying Kalman Filter (TVKF). This filter is employed alongside the Sd model, and delays are adjusted in the input channels [15].

The time-varying matrices A_t, B_t, C_t and E_t produce a time-varying error covariance matrix P_t and accordingly, a TVKF gain factor Kf_t derivation [16]. The controlled plant is stated observable and controllable from the plant noise inputs.

Also, the *k-steps-ahead* weighted error in (22) is accordingly determined in this vector form:

$$E_{P_{t+k,N}} = C_{PN} A_N x(t+k) + V_{PN} U^0_{t,N} + C_{PN} D_N W_{t+k,N} \tag{24}$$

and founded on (23) and (24), the prediction error:

$$\tilde{E}_{P_{t+k,N}} = C_{PN} A_N \tilde{x}(t+k\,|t) + C_{PN} D_N W_{t+k,N} \tag{25}$$

The state estimation error (26) doesn't depend on the selection of the control signal.

$$\tilde{x}(t+k\,|t) = x(t+k) - \hat{x}(t+k\,|\,t) \tag{26}$$

III. Hybrid GPFC in PI Structure

The optimal control solution utilizing Kalman filter state-estimation and prediction is:

$$\begin{aligned} J = E\{&(\hat{E}_{P_{t+k,N}} + \tilde{E}_{P_{t+k,N}})^T (\hat{E}_{P_{t+k,N}} + \tilde{E}_{P_{t+k,N}}) \\ &+ U^{0^T}_{t,N} \Lambda^2_N U^0_{t,N} + k^T_c \Lambda^2_K k_c \mid t\} \end{aligned} \tag{27}$$

Remarking that the optimal error estimate is orthogonal to the estimation error can simplify terms in the cost index (27).

$$J = \hat{E}^T_{P_{t+k,N}} \hat{E}_{P_{t+k,N}} + U^{0^T}_{t,N} \Lambda^2_N U^0_{t,N} + k^T_c \Lambda^2_K k_c + J_0 \tag{28}$$

where

$$\Lambda^2_N = diag\{\lambda^2_0, \lambda^2_1, ..., \lambda^2_N\}, \quad \Lambda^2_K = diag\{\rho^2_0, \rho^2_1, ..., \rho^2_N\}$$

$$J_0 = E\{\tilde{E}^T_{P_{t+k,N}} \tilde{E}_{P_{t+k,N}} \mid t\}$$

$E\{.|t\}$ is the conditional expectation on measurements till time t, λ_j is control action weighting factor, ρ_j are controller gains cost-weightings. The multi-step cost-function (28) is:

$$\begin{aligned} J = &\tilde{D}^T_{P_{t+k,N}} \tilde{D}_{P_{t+k,N}} + U^{0^T}_{t,N} V^T_{PN} \tilde{D}_{P_{t+k,N}} + \tilde{D}^T_{P_{t+k,N}} V_{PN} U^0_{t,N} \\ &+ U^{0^T}_{t,N} \left(V^T_{PN} V_{PN} + \Lambda^2_N\right) U^0_{t,N} + k^T_c \Lambda^2_K k_c + J_0 \end{aligned} \tag{29}$$

The controller within the loop is assumed in the next example to have a PI structure. For the optimal control computing, the PI gains were optimized using a unique approach to the usual unconstrained model-based predictive control where a future predicted controls vector is determined, and the control at time t is executed.

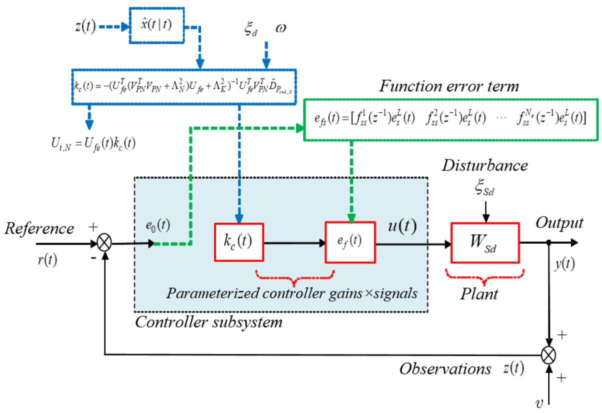

Fig. 3: NGPFC for Sd systems

The controller structure applied here has the classical cascade feedback structure. For a scalar system, functions $f_i(z^{-1})$ can be picked as in (30) utilizing two linear dynamic functions N_e to construct a PI controller:

$$u_0(t) = \overbrace{k_{c_1}}^{k_p} \overbrace{f_1(z^{-1})}^{1} e_0(t) + \overbrace{k_{c_2}}^{k_i} \overbrace{f_2(z^{-1})}^{\frac{1}{1-z^{-1}}} e_0(t) \tag{30}$$

and then the control signal can be obtained as in (31).

$$u_0(t) = \sum_{j=1}^{N_e} f_i(z^{-1}) k_j(t) e_0(t) \tag{31}$$

A suitable matrix representation is needed to calculate the gain through the optimization and (31) is arranged as:

$$u_0(t) = [\overbrace{\underbrace{f_1(z^{-1})e_0(t)}_{P} \quad \underbrace{f_{N_e}(z^{-1})e_0(t)}_{I}}^{e_{f_s}(t)}] \overbrace{\begin{bmatrix} k_{c_1} \\ k_{c_{N_e}} \end{bmatrix}}^{k_{c_s}} \tag{32}$$

where $s \in [1, N_e]$ and the PI time-varying gains are gathered as in (33).

$$k_{c_s}(t) = \begin{bmatrix} k_{c_1} \\ k_{c_{N_e}} \end{bmatrix} = \begin{bmatrix} k_p \\ k_i \end{bmatrix} \tag{33}$$

Compute the gains in (33) utilizing GPC related approach is simple as the future control action vector was formed as:

$$\overbrace{\begin{bmatrix} u_0(t) \\ u_0(t+1) \\ \vdots \\ u_0(t+N) \end{bmatrix}}^{U^0_{t,N}} = \overbrace{\begin{bmatrix} e_{f_s}(t) \\ e_{f_s}(t+1) \\ \vdots \\ e_{f_s}(t+N) \end{bmatrix}}^{U_{fe}} k_{c_s}(t) \tag{34}$$

where U_{fe} is a matrix of $N+1$ rows and N_e columns.

A receding horizon strategy type [17] is essential, where the gain k_c can be assumed constant within the period $[0, N]$ and gains computed at time t can be used to compute the optimal control action. At the subsequent sampling time, the process can be repeated.

By setting the cost-function (29) gradient to zero [15] and swapping for control input from (34), the future optimal controls vector is calculated from (35) that follows. Note that hard constraints on gains or signals can be added using the same matrices involved in (29) using Quadratic Programming (QP) methods.

$$k_{c_s}(t) = -\begin{pmatrix} U_{fe}^T \begin{pmatrix} V_{PN}^T V_{PN} \\ +\Lambda_N^2 \end{pmatrix} U_{fe} \\ +\Lambda_K^2 \end{pmatrix}^{-1} U_{fe}^T V_{PN}^T \tilde{D}_{P_{t+k,N}} \qquad (35)$$

Finally, constraints can be applied on the value of a controller gains or the gains rate of change by utilizing QP [18], and the controller is realized as in Fig. 3

IV. CSTR SYSTEM EXAMPLE

Continuous stirred tank reactors (CSTRs) are prevalent in chemical and pharmaceutical systems. The CSTR system has extremely NL behaviour and often has a wide operating range. They may sometimes be run in different operating regions to produce a variety of separate manufactured goods to achieve flexible manufacturing to cope with market competition [19]. A crucial control goal is to reduce the product transition time by reducing off-specs products volume produced throughout any transition [20]. A basic two-state CSTR with exothermic irreversible first-order reaction $A \rightarrow B$ is illustrated in Fig. 5 and used as a case-study in this simulation.

The C_A, T, q_c and T_{cf} are resultant concentration, reactor temperature, coolant flow rate, and coolant temperature correspondingly. CSTR system output is C_A, the input is T_{cf} and the system states are:

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} C_A \\ T \end{bmatrix}$$

The following NL equations can define the dynamics of the CSTR [21]:

$$\begin{aligned} \frac{dx_1}{dt} &= -\theta x_1 \kappa(x_2) + q(x_{1f} - x_1) \\ \frac{dx_2}{dt} &= \beta\theta x_1 \kappa(x_2) - (q+\delta)x_2 + \delta u + q x_{2f} \\ y &= x_1 \end{aligned} \qquad (36)$$

where

$$\kappa(x_2) = \exp\left(\frac{x_2}{1+\frac{x_2}{\lambda}}\right)$$

$$\delta = 0.3,\ \lambda = 20,\ \theta = 0.072,\ \beta = 8,\ q = 1,\ x_{1f} = 1,\ x_{2f} = 0$$

The CSTR has three steady-states at $u=0$:

$$x_{s_1} = \begin{bmatrix} 0.856 \\ 0.886 \end{bmatrix},\ x_{s_2} = \begin{bmatrix} 0.5528 \\ 2.7517 \end{bmatrix},\ x_{s_3} = \begin{bmatrix} 0.2353 \\ 4.7050 \end{bmatrix} \qquad (37)$$

The NL system has been linearized at each steady-state region and then discretized by sampling time of 100 milliseconds to obtain the CSTR system PWA model in [19] as:

$$x(k+1) = \begin{cases} A_1 x(k) + B_1 u(k) + b_1,\ x \in \Omega_1 \\ A_2 x(k) + B_2 u(k) + b_2,\ x \in \Omega_2 \\ A_3 x(k) + B_3 u(k) + b_3,\ x \in \Omega_3 \end{cases} \qquad (38)$$
$$y(k) = Cx(k)$$

where

$$A_1 = \begin{bmatrix} 0.8889 & -0.0123 \\ 0.1254 & 0.9751 \end{bmatrix},\ B_1 = \begin{bmatrix} -0.0002 \\ 0.0296 \end{bmatrix},\ b_1 = \begin{bmatrix} 0.1060 \\ -0.0852 \end{bmatrix}$$
$$A_2 = \begin{bmatrix} 0.8241 & -0.0340 \\ 0.6365 & 1.1460 \end{bmatrix},\ B_2 = \begin{bmatrix} -0.0005 \\ 0.0322 \end{bmatrix},\ b_2 = \begin{bmatrix} 0.1907 \\ -0.7537 \end{bmatrix}$$
$$A_3 = \begin{bmatrix} 0.6002 & -0.0463 \\ 2.4016 & 1.2430 \end{bmatrix},\ B_3 = \begin{bmatrix} -0.0007 \\ 0.0338 \end{bmatrix},\ b_3 = \begin{bmatrix} 0.3119 \\ -1.7083 \end{bmatrix}$$
$$C = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

Using (7) and (8), the PWA model (38) can be written in Sd form as follow:

$$x_{sd}(t+1) = \overbrace{\left\{ \begin{matrix} \delta_1 \times \begin{bmatrix} 0.8889 & -0.0123 \\ 0.1254 & 0.9751 \end{bmatrix} + \\ \delta_2 \times \begin{bmatrix} 0.8241 & -0.0340 \\ 0.6365 & 1.1460 \end{bmatrix} + \\ \delta_3 \times \begin{bmatrix} 0.6002 & -0.0463 \\ 2.4016 & 1.2430 \end{bmatrix} \end{matrix} \right\}}^{A_{sd}} x_{sd}(t)$$
$$+ \overbrace{\left\{ \begin{matrix} \delta_1 \times \begin{bmatrix} -0.0002 \\ 0.0296 \end{bmatrix} + \\ \delta_2 \times \begin{bmatrix} -0.0005 \\ 0.0322 \end{bmatrix} + \\ \delta_3 \times \begin{bmatrix} -0.0007 \\ 0.0338 \end{bmatrix} \end{matrix} \right\}}^{B_{sd}} u(t-k) + \overbrace{\left\{ \begin{matrix} \delta_1 \times \begin{bmatrix} 0.1060 \\ -0.0852 \end{bmatrix} + \\ \delta_2 \times \begin{bmatrix} 0.1907 \\ -0.7537 \end{bmatrix} + \\ \delta_3 \times \begin{bmatrix} 0.3119 \\ -1.7083 \end{bmatrix} \end{matrix} \right\}}^{b_{sd}}$$
$$y_{sd}(t) = \overbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}^{C_{sd}} x_{sd}(t)$$

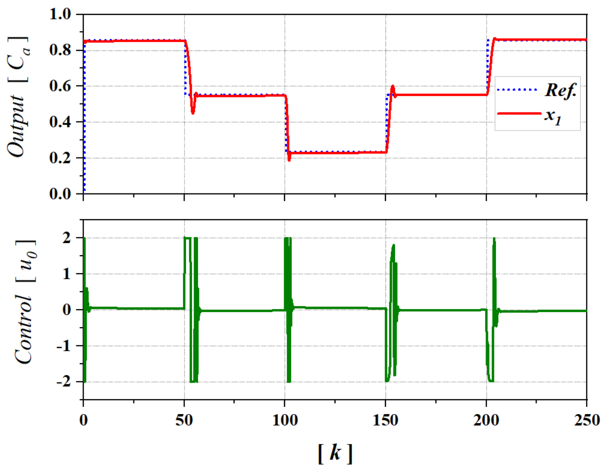

Fig. 4: Operating points output tracking

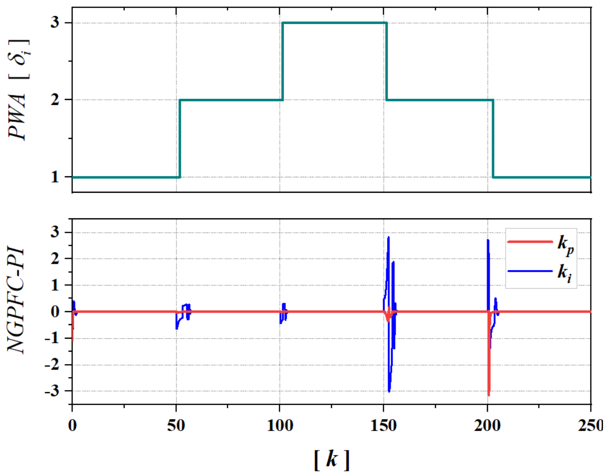

Fig. 6: Evolution of the PWA regions

where

$$\delta_1 = \begin{cases} 1 & if \quad 0.78 < x_1 \le 1 \\ 0 & otherwise \end{cases}$$
$$\delta_2 = \begin{cases} 1 & if \quad 0.35 < x_1 \le 0.78 \\ 0 & otherwise \end{cases} \tag{39}$$
$$\delta_3 = \begin{cases} 1 & if \quad 0 \le x_1 \le 0.35 \\ 0 & otherwise \end{cases}$$

The next constraints must be satisfied during the simulation:

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} \le x_{sd} \le \begin{bmatrix} 1 \\ 6 \end{bmatrix}, \quad -2 \le u \le 2$$

The need to build the CSTR flexible operating strategies and optimal switching amongst operating regions motivates the design objective. The NL plant was moved between the different steady-states in (37) by tracking the concentration set-point throughout the operating regions. Satisfying tracking performance was obtained from the simulation results as shown in Fig. 5, with a slight overshoot around the unstable steady state x_{s_2}. The controller tracking performance showed no oscillation during the PWA model switching from one region to another. The auxiliary logic variable in (39), acted as supervisor and was responsible for selecting the associated PWA model based on the concentration measurement and allowed the NGPFC to update the loops' PI controller's gains accordingly, as shown in Fig. 6.

V. Conclusion

An NGPFC controller has been designed based on a PWA model, in which switching dynamics are altered by the state and/or the control input. This controller has been set to have a PI structure and was used to control an NL model of CSTR in a reference tracking problem. The controller optimized time-varying gains were consistently modified. Utilizing future prediction, the NGPFC controller can adjust to the reference trajectory variations and system operating regions' changes. Satisfactory tracking performance was obtained.

Acknowledgement

The author, S. Alotaibi, would like to thank SACB UK for his studies' sponsorship and thank Professor Yang Pang, Dalian University and Dr Pawel Majecki, ISC Ltd, for the software support.

References

[1] W. P. M. H. Heemels, J. M. Schumacher, and S. Weiland, 'Linear Complementarity Systems', *SIAM J. Appl. Math.*, 2000.

[2] A. J. van der Schaft and J. M. Schumacher, 'Complementarity modeling of hybrid systems', *IEEE Trans. Automat. Contr.*, vol. 43, no. 4, pp. 483–490, Apr. 1998.

[3] A. Bemporad and M. Morari, 'Control of systems integrating logic, dynamics, and constraints', *Automatica*, 1999.

[4] B. De Schutter, 'Optimal Control of a Class of Linear Hybrid Systems with Saturation', *SIAM J. Control Optim.*, vol. 39, no. 3, pp. 835–851, Jan. 2000.

[5] E. Sontag, 'Nonlinear regulation: The piecewise linear approach', *IEEE Trans. Automat. Contr.*, vol. 26, no. 2, pp. 346–358, Apr. 1981.

[6] B. De Schutter, B. De Schutter, and T. Van Den Boom, 'On Model Predictive Control for Max-Min-Plus-Scaling Discrete Event Systems', 2000.

[7] T. A. Henzinger, 'The theory of hybrid automata', in *Proceedings 11th Annual IEEE Symposium on Logic in Computer Science*, pp. 278–292.

[8] W. P. M. H. Heemels, B. De Schutter, and A. Bemporad, 'Equivalence of hybrid dynamical models', *Automatica*, 2001.

[9] Y. Pang and M. Grimble, 'State dependent NGMV control of delayed piecewise affine systems'. pp. 7192–7197, 2009, doi: 10.1109/CDC.2009.5399927.

[10] P. Yan, li Weiliang, and X. Hao, 'Equivalent state-dependent models of linear hybrid automata | Request PDF', *Control Theory Appl.*, vol. 30, no. 3, 2013.

[11] M. Grimble and Y. Pang, 'NGMV control of state dependent multivariable systems', in *2007 46th IEEE Conference on Decision and Control*, 2007, pp. 1628–1633.

[12] S. Alotaibi and M. Grimble, 'Optimal Generalized Predictive Functional Control with Applications to Extended PID Control for Quadruple Tank', Sep. 2019.

[13] S. Alotaibi and M. Grimble, 'Nonlinear Optimal Generalized Predictive Functional Control Applied to Quasi-LPV Model of Automotive Electronic Throttle', Jul. 2019.

[14] M. Grimble and P. Yan, 'NGMV control of state dependent multivariable systems'. pp. 1628–1633, 2007, doi: 10.1109/CDC.2007.4434359.

[15] M. Grimble and M. A. Johnson, *Optimal control and stochastic estimation : theory and applications*. Wiley, 1988.

[16] M. Grimble and P. Majecki, 'NGMV Control Using Unstable State-Dependent Multivariable Models'. 2016.

[17] W. Kwon and A. Pearson, 'A modified quadratic cost problem and feedback stabilization of a linear system', *IEEE Trans. Automat. Contr.*, vol. 22, no. 5, Oct. 1977, doi: 10.1109/TAC.1977.1101619.

[18] M. Grimble and P. Majecki, 'Restricted structure predictive control for linear and nonlinear systems', *Int. J. Control*, vol. 93, no. 8, pp. 1775–1799, Oct. 2020, doi: 10.1080/00207179.2018.1533252.

[19] J. DU, C. SONG, and P. LI, 'Modeling and Control of a Continuous Stirred Tank Reactor Based on a Mixed Logical Dynamical Model', *Chinese J. Chem. Eng.*, vol. 15, no. 4, pp. 533–538, Mar. 2007.

[20] L. Özkan, M. V. Kothare, and C. Georgakis, 'Control of a solution copolymerization reactor using multi-model predictive control', *Chem. Eng. Sci.*, vol. 58, no. 7, pp. 1207–1221, Apr. 2003.

[21] P. B. Sistu and B. W. Bequette, 'Nonlinear predictive control of uncertain processes: Application to a CSTR', *AIChE J.*, vol. 37, no. 11, pp. 1711–1723, Nov. 1991.