Constrained Predictive Pole-placement Control with Linear Models

Wen-Hua Chen [a] Peter J Gawthrop [b]

[a] *Department of Aeronautical and Automotive Engineering, Loughborough University, Leicestershire, LE11 3TU, UK*
W.Chen@lboro.ac.uk

[b] *Centre for Systems and Control and Department of Mechanical Engineering, University of Glasgow, GLASGOW. G12 8QQ Scotland.*
P.Gawthrop@eng.gla.ac.uk

Abstract

Predictive Pole-Placement (PPP) control is a continuous-time MPC using a particular set of basis functions leading to pole-placment behaviour in the unconstained case. This paper presents two modified versions of the PPP controller which are each shown to have desirable stability properties when controlling systems with input, output and state constraints.

Key words: Predictive control; pole-placement control; constrained control; quadratic programming

1 Introduction

Predictive pole-placement control (PPP) (Gawthrop and Ronco, 2002; Gawthrop, 2000) is a *continuous-time* model-based predictive control scheme which represents the moving-horizon control signal as a weighted sum of a finite set of *basis functions* which are the state impulse response of a stable linear time-invariant system of the same order as the controlled system. Under mild conditions Gawthrop and Ronco (2002) show that the unconstrained closed-loop system has the same poles as the system generating the control signal components. Hence the name "predictive pole-placement" is used to describe the method. Insofar as the method uses a small number of impulse-reponse like basis functions, the method is related to the MPC of Richalet *et al.* (1978). As with all model-based predictive controllers, the ability of the method to incorporate input, output and state constraints is an important feature of the method. The *first purpose* of this paper is to show explicitly that constraints can, indeed, be easily incorporated into the method and the use of such constraints is illustrated.

Stability is an important property for all control algorithms; the *second purpose* of this paper is to provide a stability proof for two modified versions of the PPP algorithm. The main challenge for establishing stability of PPP is that the moving horizon control signal is not arbitrary but a weighted sum of a set of basis functions, which is different from conventional MPC, where the only constraint on control signal is its magnitude or rate. This implies that a stabilising control profile might not be a linear combination of the pre-specified basis functions. Two alternative modified versions of linear PPP (Gawthrop and Ronco, 2002) are presented in this paper, each of which guarantees stability of the closed-loop system. The first approach is similar to that of Rawlings and Muske (1993) insofar as the algorithm of Gawthrop and Ronco (2002) is modified to include appropriate weighting functions to embed the optimisation in an infinite horizon LQ formulation and in a similar fashion, stability of the approach is shown. The second approach, instead of including terminal weighting in the performance index, embeds a constraint in the on-line optimisation so that the stability can be established under mild conditions. Chen and Allgöwer (1998) examine a *continuous-time* algorithm and show its stability for *non-linear* systems. However the control signal is piecewise constant and thus retains a discrete-time character. Moreover, since the control signal in the PPP is a linear combination of a finite set of the basis functions, i.e., linearly parameterised by the basis functions, the number of on-line optimisation variables is the same as the order of the controlled system. For conventional MPC algorithms for continuous-time system, the number of the optimisation variables depends on the sampling time and the receding horizon, which may be much larger than the system order. Therefore, the number of optimisation variables and online optimisation burden is significantly reduced. A

detailed comparison between PPP and other MPC methods has not been made and would be beyond the scope of this paper; but it seems that PPP uses a particularly effective set of basis functions.

2 Unconstrained PPP

This section extends the unconstrained algorithm of Gawthrop and Ronco (2002) to include input and terminal state weighting thus embedding the PPP approach into an infinite horizon LQ optimisation. The linear systems considered in this paper are described by:

$$\begin{cases} \frac{d}{dt}x(t) & = Ax(t) + Bu(t) \\ y(t) & = Cx(t) \\ x(0) & = x_0 \end{cases} \tag{1}$$

where $x \in \Re^{n_x}$, $y \in \Re^{n_y}$ and $u \in \Re^{n_u}$. x_0 is the system initial condition. Given the state $x(t)$ at time t, we are interested in the evolution of the *moving horizon* state $x^\star(t,\tau)$ and output $y^\star(t,\tau)$ [1] where

$$\begin{cases} \frac{d}{d\tau}x^\star(t,\tau) & = Ax^\star(t,\tau) + Bu^\star(t,\tau) \\ y^\star(t,\tau) & = Cx^\star(t,\tau) \\ x^\star(t,0) & = x_0^\star \end{cases} \tag{2}$$

The differential equations 1 and 2 are related by having the *same* state space matrices and by imposing the *cross-coupling* conditions:

$$\begin{cases} x_0^\star & = x(t) \\ u(t) & = u^\star(t,0) \end{cases} \tag{3}$$

In this paper, the *moving horizon* control signal $u^\star(t,\tau)$ is *linearly parameterised* by the n_U components of the column vector $U(t)$ so that:

$$u^\star(t,\tau) = U^\star(\tau)U(t) \tag{4}$$

where $U^\star(\tau)$ is a $n_u \times n_U$ matrix of functions of τ. The components of $U^\star(\tau)$ can be regarded as a set of *basis functions* for the control signal $u^\star(t,\tau)$ and the components of $U(t)$ the corresponding weights. For the purposes of this paper, the basis functions $U^\star(\tau)$ are generated from the LTI system with transition matrix A_u given by:

$$\begin{cases} \frac{d}{d\tau}\widetilde{U}^\star(\tau) & = A_u\widetilde{U}^\star(\tau) \\ \widetilde{U}^\star(0) & = \widetilde{U}_0^\star \end{cases} \tag{5}$$

where A_u has non-positive eigenvalues; in other words (5) represents a stable system. Because Equation 4 generates the moving-horizon control $u^\star(t,\tau)$ with no feedback from $x^\star(t,\tau)$, it will be referred to as an *open-loop* control in the sequel.

The vector $U(t)$ is to be chosen to minimise (at a given time t) the (unconstrained) quadratic cost function:

$$\begin{aligned} &J(U(t),x(t),W(t)) = \\ &\frac{1}{2}\int_{\tau_1}^{\tau_2}(y^\star(t,\tau)-w^\star(t,\tau))^T Q(y^\star(t,\tau)-w^\star(t,\tau))d\tau \\ &+\frac{1}{2}\int_{\tau_1}^{\tau_2}u^\star(t,\tau)^T Ru^\star(t,\tau)d\tau \\ &+(x^\star(t,\tau_2)-x_w(\tau))^T P(x^\star(t,\tau_2)-x_w(\tau)) \end{aligned} \tag{6}$$

where $Q \in \Re^{n_y \times n_y}$, $R \in \Re^{n_u \times\ n_u}$, $P \in \Re^{n_x \times n_x}$ are positive definite matrices weighting the system output, input and terminal state respectively. $x_w(\tau) \in \Re^{n_x}$ is the state corresponding to the the steady-state solution of (2) corresponding to $y^\star(t,\tau) = w(t)$. To streamline the notation: $J(U(t),x(t),W(t))$ will be written as J in the sequel. Following Gawthrop and Ronco (2002), the derivatives of this cost function are denoted by $J_U = \frac{\partial}{\partial U}J$, $J_{UU} = \frac{\partial^2}{\partial U^2}J$, $J_{Ux} = \frac{\partial^2}{\partial U \partial x}J$ and $J_{UW} = \frac{\partial^2}{\partial U \partial W}J$. However, these subscripts do *not* imply derivatives when attached to other symbols apart from J.

There are two special cases of the PPP algorithm considered in this paper:

(1) The version considered by Gawthrop and Ronco (2002) where $R = 0$ and $P = 0$ in (6).
(2) A modified version of 1 where the terminal weight P is retained in (6) to reflect the terminal value of the performance index of version 1. It is assumed that the system (1) is controllable and $\tau_1 = 0$. P is chosen as the (unique) positive-definite solution of the Algebraic Ricatti Equation (ARE) corresponding to (1) and the output and input weighting matrices Q and R; A_u (5) is then chosen to have the corresponding eigenvalues.

3 Constrained PPP

A major reason for using predictive control is the possibility of including input, output and state constraints within the optimisation procedure. For the usual discrete-time predictive control, it is known that it is possible to formulate such constraints, together with the optimisation, as a Quadratic Programme (QP). For the continuous-time PPP algorithm of this paper to be useful, it is important that such constraints can be included in a similar fashion. The purpose of this section is to show that it is indeed possible to embed linear PPP together with input, output and state constraints in a QP. The result is contained in Lemma 1.

Lemma 1 (Constrained optimisation) *Consider the PPP algorithm where the input functions $U^\star(\tau)$ are given by*

[1] More generally, $y^\star(t,\tau)$ can be regarded as a vector of *performance variables*

Equation 5. Then given n_{cu} input inequality constraints $\bar{u}^\star(t,\tau_{uk})$ at the n_{cu} times τ_{uk} of the form:

$$u^\star(t,\tau_{uk}) \leq \bar{u}^\star(t,\tau_{uk}) \tag{7}$$

and the n_{cy} output inequality constraints $\bar{y}^\star(t,\tau_{yk})$ at the n_{cy} times τ_{yk} of the form:

$$y^\star(t,\tau_{yk}) \leq \bar{y}^\star(t,\tau_{yk}) \tag{8}$$

the minimisation of the cost function J of Equation 6 subject to the constraints 7 and 8 is equivalent to the solution of the QP:

$$\min_{U(t)} \left\{ \frac{1}{2} U(t)^T J_{UU} U(t) + U(t)^T \left(J_{Ux} x(t) - J_{UW} W(t) \right) \right\} \tag{9}$$

subject to $\Gamma U(t) \leq \gamma$

$$\Gamma = \begin{pmatrix} \Gamma_u \\ \Gamma_y \end{pmatrix};\ \gamma = \begin{pmatrix} \gamma_u \\ \gamma_y \end{pmatrix} \tag{10}$$

where

$$\Gamma_u = \begin{pmatrix} U^\star(\tau_{u1}) \\ U^\star(\tau_{u2}) \\ \cdots \\ U^\star(\tau_{un_{cu}}) \end{pmatrix};\ \gamma_u = \begin{pmatrix} \bar{u}^\star(t,\tau_{u1}) \\ \bar{u}^\star(t,\tau_{u2}) \\ \cdots \\ \bar{u}^\star(t,\tau_{un_{cu}}) \end{pmatrix} \tag{11}$$

and

$$\Gamma_y = \begin{pmatrix} y_U^\star(\tau_{y1}) \\ y_U^\star(\tau_{y2}) \\ \cdots \\ y_U^\star(\tau_{yn_{cy}}) \end{pmatrix};\ \gamma_y = \begin{pmatrix} \bar{y}^\star(t,\tau_{y1}) - y_x^\star(\tau_{y1})x(t) \\ \bar{y}^\star(t,\tau_{y2}) - y_x^\star(\tau_{y2})x(t) \\ \cdots \\ \bar{y}^\star(t,\tau_{yn_{cy}}) - y_x^\star(\tau_{yn_{cy}})x(t) \end{pmatrix} \tag{12}$$

the ith column $y_{Ui}^\star(\tau)$ of $y_U^\star(\tau)$ is the solution of the ode:

$$\begin{cases} \frac{d}{d\tau} x_{Ui}^\star(\tau) & = A x_{Ui}^\star(\tau) + B U_i^\star(\tau) \\ y_{Ui}^\star(\tau) & = C x_{Ui}^\star(\tau) \\ x_{Ui}^\star(0) & = 0_{n_x} \end{cases} \tag{13}$$

where 0_{n_x} is a column vector with all of its n_x elements zero and $U_i^\star(\tau)$ is the $n_u \times 1$ vector forming the ith column of the matrix $U^\star(\tau)$; the ith column of $y_x^\star(\tau)$ is the solution of:

$$\begin{cases} \frac{d}{d\tau} x_{x_i}^\star & = A x_{x_i}^\star \\ y_{x_i}^\star & = C x_{x_i}^\star \\ x_{x_i}^\star & = I_i \end{cases} \tag{14}$$

where I_i is a column vector with the ith element unity and all others zero,

PROOF. The cost function J of Equation 6 can be rewritten as:

$$J = J_{QP} + J_0(x(t), W(t)) \tag{15}$$

where

$$J_{QP} = \frac{1}{2} U(t)^T J_{UU} U(t) + U(t)^T \left(J_{Ux} x(t) - J_{UW} W(t) \right) \tag{16}$$

and J_{UU}, J_{Ux}, J_{UW} and $J_0(x(t), W(t))$ are all independent of U. Hence the $U(t)$ which minimises J_{QP} is the same as that which minimises J. Equation 11 follows from Equation 4 and Equation 12 follows from the fact that the system of Equation 2 is linear, and so that the solution can be rewritten as (Gawthrop and Ronco, 2002):

$$y^\star(t,\tau) = y_U^\star(\tau) U(t) + y_x^\star(\tau) x(t) \tag{17}$$

□

4 Stability Analysis

This section studies the stability of the constrained PPP algorithm. To this end, similar to most of the existing MPC algorithms, only regulation problems are considered. Two stability guaranteed constrained PPP algorithms are proposed in this section.

4.1 Stability Guaranteed Algorithm 1

The online optimisation problem in Lemma 1 is modified as:

$$\min_{U(t)} \frac{1}{2} \int_{\tau_1}^{\tau_2} y^\star(t,\tau)^T Q y^\star(t+\tau) + u^\star(t,\tau)^T R u^\star(t,\tau) dt + x^\star(t,\tau_2)^T P x^\star(t,\tau_2) \tag{18}$$

subject to system dynamics,

$$\Gamma U(t) \leq \gamma \tag{19}$$

and

$$2(Ax^\star(t,\tau) + Bu^\star(t,\tau))^T P x^\star(t,\tau) + y^\star(t,\tau)^T Q y^\star(t,\tau) + u^\star(t,\tau)^T R u^\star(t,\tau) \leq 0 \tag{20}$$

for $\tau \in [0,\delta]$ where δ denotes the sampling time interval, $R > 0$, $\widetilde{Q} = C^T Q C \geq 0$ and non-zero x is detectable in the performance index (Sepulchre *et al.*, 1996) and $P > 0$. Γ and γ are defined in Equations 10-12.

Lemma 2 *The constraint (20) is convex in the optimisation variable $U(t)$ if the matrix P satisfies (21)*

$$x_U^\star(\tau)^T (A^T P + P A + \widetilde{Q}) x_U^\star(\tau) + U^\star(\tau)^T B^T P x_U(\tau) + x_U^\star(\tau)^T P B U^\star(\tau) + U^\star(\tau)^T R U^\star(\tau) \geq 0 \tag{21}$$

where the ith column of $x_U^\star$ is defined in Eq. (13).

PROOF. The constraint (20) can be written as

$$\begin{pmatrix} x(t,\tau)^\star \\ u(t,\tau)^\star \end{pmatrix}^T \begin{pmatrix} A^TP + PA + \widetilde{Q} & PB \\ B^TP & R \end{pmatrix} \begin{pmatrix} x(t,\tau)^\star \\ u(t,\tau)^\star \end{pmatrix} \leq 0 \tag{22}$$

Since the state and the control can be expressed as (Gawthrop and Ronco, 2002)

$$x^\star(t,\tau) = x_x^\star(\tau)x(t) + x_U^\star(\tau)U(t) \tag{23}$$

$$u^\star(t,\tau) = U^\star(\tau)U(t), \tag{24}$$

it follows that

$$\begin{pmatrix} x^\star(t,\tau) \\ u^\star(t,\tau) \end{pmatrix} = \begin{pmatrix} x_x^\star(\tau) & x_U^\star(\tau) \\ 0 & U^\star(\tau) \end{pmatrix} \begin{pmatrix} x(t) \\ U(t) \end{pmatrix} \tag{25}$$

Substituting (25) into (22) gives

$$\begin{pmatrix} x(t) \\ U(t) \end{pmatrix}^T \begin{pmatrix} x_x^\star(\tau) & x_U^\star(\tau) \\ 0 & U^\star(\tau) \end{pmatrix}^T \begin{pmatrix} A^TP + PA + \widetilde{Q} & PB \\ B^TP & R \end{pmatrix} \begin{pmatrix} x_x^\star(\tau) & x_U^\star(\tau) \\ 0 & U^\star(\tau) \end{pmatrix} \begin{pmatrix} x(t) \\ U(t) \end{pmatrix} \leq 0 \tag{26}$$

The above constraint is convex in terms of the variable $U(t)$ if

$$\begin{pmatrix} x_U^\star(\tau) \\ U^\star(\tau) \end{pmatrix}^T \begin{pmatrix} A^TP + PA + \widetilde{Q} & PB \\ B^TP & R \end{pmatrix} \begin{pmatrix} x_U^\star(\tau) \\ U^\star(\tau) \end{pmatrix} \geq 0 \tag{27}$$

which is implied by condition (21).

Therefore it can be shown that the constraint (26), i.e. (20), is convex in terms of the variables $U(t)$ if condition (21) is satisfied. □

Remark 1 *In the online optimisation the constraint (20) is replaced by (26) where all the matrices can be calculated off-line. Since the original optimisation problem in the PPP is a QP, together with the convex constraint (26), the new reformed optimisation problem in the PPP algorithm is convex and the global minimum can be found. Introduction of extra constraints for stability changes the optimisation problem in PPP from original QP to a convex one which certainly adds the computational burden. However this shall not be comparable to benefits gained by significantly reducing the number of optimisation variables. The parameterisation of the control signal with few variables is very important for engineering implementation of MPC, in particular for fast processes; for example see recent work in Milam* et al. *(2005).*

Theorem 1 *Suppose that the online optimisation problem in PPP algorithm is feasible at the time $t \geq t_0$. Then the closed-loop system under the PPP algorithm is asymptotically stable.*

PROOF. Choose $V(x(t)) = x(t)^TPx(t)$ as an Lyapunov candidate for the PPP algorithm. Suppose that the control profile yielded by solving the optimisation problem is given by $u^\star(t,\tau)$, $0 \leq \tau \leq T$ and the corresponding state trajectory and the performance cost are denoted by $x^\star(t,\tau)$ and $J^\star(t)$. After implementation of this control profile for a period of time $[t, t+\delta]$ where δ could be sufficiently small in continuous fashion or a time interval in intermittent fashion (Gawthrop, 2004), the state is measured as $x(t+\delta)$. The corresponding Lyapunov function is given by

$$V(x(t+\delta)) = x(t+\delta)^TPx(t+\delta) \tag{28}$$

The difference of the Lyapunov function along the state trajectory is given by

$$V(x(t+\delta)) - V(x(t)) = x(t+\delta)^TPx(t+\delta) - x(t)^TPx(t) \tag{29}$$

Since the control profile $u^\star(t,\tau)$ is implemented in the time interval $[t, t+\delta]$, we have

$$\begin{aligned} u(t+\tau) &= u^\star(t,\tau), \tau \in [0,\delta] \\ x(t+\tau) &= x^\star(t,\tau), \tau \in [0,\delta] \end{aligned} \tag{30}$$

Condition (20) becomes

$$\begin{aligned} &2(Ax(t+\tau) + Bu(t+\tau))^TPx(t+\tau) \\ &+ x(t+\tau)^T\widetilde{Q}x(t+\tau) + u(t+\tau)^TRu(t+\tau) \leq 0 \end{aligned} \tag{31}$$

for $\tau \in [0,\delta]$.

Integrating (31) over the time period $[0,\delta]$ along the state trajectory gives

$$\begin{aligned} &\int_0^\delta 2(\dot{x}(t+\tau)^TPx(t+\tau)d\tau \\ &\leq -\int_0^\delta x(t,\tau)^T\widetilde{Q}x(t,\tau) + u(t,\tau)^TRu(t,\tau)d\tau \end{aligned} \tag{32}$$

The left side of the above inequality can be written as

$$\int_0^\delta 2\dot{x}(t+\tau)^TPx(t+\tau)d\tau = x(t+\delta)^TPx(t+\delta) - x(t)^TPx(t) \tag{33}$$

Combining (29),(32) and (33) obtains

$$\begin{aligned} &V(x(t+\delta)) - V(x(t)) \\ &\leq -\int_0^\delta x(t+\tau)^T\widetilde{Q}x(t+\tau) + u(t+\tau)^TRu(t+\tau)d\tau \\ &\leq 0 \end{aligned} \tag{34}$$

where the latter inequality follows from $\widetilde{Q} \geq 0$ and $R > 0$. Hence the Lyapunov function $V(x)$ is non-increasing along the closed-loop system's trajectory.

Repeatedly using the inequality (34) yields

$$\begin{aligned} V(x(\infty)) - V(x(t_0)) &\leq -\int_{t_0+\delta}^{\infty} x(\tau)^T \widetilde{Q} x(\tau) + u(\tau)^T R u(\tau) d\tau \\ &\leq -\int_{t_0+\delta}^{\infty} x(\tau)^T \widetilde{Q} x(\tau) d\tau. \end{aligned} \quad (35)$$

Since $V(x)$ is monotonically non-increasing and bounded from below by zero, the integral on the right side of the inequality (35) converges and hence exists and is bounded.

In addition, it can be shown that $x(t)$ is uniformly continuous in t on $[t_0, \infty)$. This implies that $x^T \widetilde{Q} x = y^T Q y \to 0$ as $t \to \infty$ (Hahn, 1967, proof of Theorem 26.4) or (Khalil, 1992). Hence the equilibrium point $x = 0$ of the system (1) is asymptotically stable (Sepulchre *et al.*, 1996) □

Remark 2 *In order to use the PPP algorithm, the matrix P has to be determined off-line. The following procedure can be used to determine P.*

- *Choose the desired pole placement according to performance requirement.*
- *Determine the linear control gain using pole placement technique.*
- *Calculate the matrix P by solving the Lyapunov equation:*

$$(A + BK)^T P + P(A + BK) + C^T Q C + K^T R K = 0 \quad (36)$$

Remark 3 *Similar to other MPC algorithms with basis functions such as general PFC (Rossiter and Richalet, 2002), PPP may suffer recursive feasibility problem. However, as long as the matrix P in the algorithm is chosen using the procedure in Remark 2, its recursive feasibility is guaranteed. By following this procedure, Eq. (37) is satisfied and the closed-loop system under the control $u = Kx$ has the required poles which form the basis functions. Further more the closed-loop response and thus the control profile can be expressed as a linear combination of the chosen basis functions. Together with (37), this implies that at each optimisation step t there at least exists a control profile which can be expressed as the linear combination of the chosen basis functions such that constraint (21) is feasible. Therefore, the optimisation problem is always feasible if it is feasible at $t = t_0$.*

4.2 Stability guaranteed algorithm 2

In Stability Guaranteed PPP Algorithm 1, similar to standard practice in conventional MPC algorithms, a terminal weighting term is added in the performance index to ensure the stability. In this section, Stability guaranteed PPP algorithm 2 is introduced where instead of modifying the performance index, a stability constraint is added in the on-line optimisation. In this algorithm, the on-line optimisation problem in the constrained PPP is modified as

$$\min_{U(t)} \frac{1}{2} \int_0^T y^\star(t,\tau)^T Q y^\star(t+\tau) + u^\star(t,\tau)^T R^\star u(t,\tau) dt \quad (37)$$

subject to system dynamics,

$$\begin{aligned} \Gamma U(t) &\leq \gamma \\ x^\star(t,\delta)^T P x^\star(t,\delta) &\leq \alpha x^\star(t,0)^T P x^\star(t,0) \end{aligned} \quad (38)$$

where P is positive definite, $\alpha \in (0,1)$ and δ could be sufficiently small in continuous fashion or a time interval in intermittent fashion.

Remark 4 *The constraint (38) is convex in terms of the variable $U(t)$. This can be shown by the fact that*

$$x^\star(t,\delta) = x_x^\star(\delta) x(t) + x_U^\star(\delta) U(t) \quad (39)$$

After substituting (39) into (38) and following the fact that P is positive definite, one can conclude that (38) is convex in terms of $U(t)$.

Theorem 2 *Suppose that the online optimisation problem in the PPP algorithm is feasible for all $t \geq t_0$. Then the closed-loop system under PPP algorithm is exponentially stable*

PROOF. Following from the assumption that the on-line optimisation problem is feasible, after implementation of the PPP algorithms for one sampling interval, the state trajectory satisfies

$$x(t+\delta)^T P x(t+\delta) \leq \alpha x(t)^T P x(t) \quad (40)$$

since $x(t+\delta) = x^\star(t,\delta)$ as discussed in the proof of Theorem 1.

It can be shown that for all $\alpha \in (0,1)$

$$\alpha \leq e^{-(1-\alpha)} \quad (41)$$

Using the relationship in (40), one obtains

$$x(t+\delta)^T P x(t+\delta) \leq e^{-(1-\alpha)} x(t)^T P x(t) \quad (42)$$

Repeated the above process for k steps from $t = t_0$, it can be shown that

$$x(t_0 + k\delta)^T P x(t_0 + k\delta) \leq e^{-(1-\alpha)k} x(t_0)^T P x(t_0) \quad (43)$$

Hence all the state on the discrete-time points are exponentially convergent to the origin. Since the controlled system

is linear and the control is constrained in its magnitude, it can be shown that there exists a constant $\beta \in [1, \infty)$ such that the transient state between t and $t + \delta$ is bounded by

$$x(t+\tau)^T P x(t+\tau) \leq \beta x(t)^T P x(t), 0 < \tau < \delta \quad (44)$$

As above, one can conclude that

$$x(t+\tau)^T P x(t+\tau) \leq \beta e^{-(1-\alpha)k} x(t_0)^T P x(t_0), 0 < \tau \leq \delta, \quad (45)$$

where $t = t_0 + (k-1)\delta$.

Therefore all the state along the trajectory is exponentially convergent to the origin. □

Remark 5 *In the original PPP algorithm (Gawthrop and Ronco, 2002), there is no control weighting in the performance index. This can be considered as a special case of the algorithm proposed in this case (with $R = 0$). In the same fashion as above, it can be shown the stability of the original PPP algorithm can be guaranteed if the condition (38) is added in the on-line optimisation.*

5 Conclusion

The continuous-time predictive pole-placement control algorithm has been extended to include input and output constraints. Two modified versions of the original algorithm have been shown to be stable under mild conditions. The effectiveness of the predictive pole-placement control algorithms for constrained systems is demonstrated by an illustrative numerical example.

References

Chen, H. and F. Allgöwer (1998). A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica* **34**(10), 1205–1217.

Gawthrop, P. J. (2004). Intermittent constrained predictive control of mechanical systems. In: *Proceedings of the 3rd IFAC Symposium on Mechatronic Systems* (I R Petersen, Ed.).

Gawthrop, P.J. (2000). Linear predictive pole-placement control: practical issues. In: *Proceedings of the 39th IEEE Conference on Decision and Control*. IEEE.

Gawthrop, P.J. and E. Ronco (2002). Predictive pole-placement control with linear models. *Automatica* **38**(3), 421–432.

Hahn, W. (1967). *Stability of Motion*. Springer-Verlag. New York.

Khalil, H. K. (1992). *Nonlinear Systems*. Macmillan. New York.

Milam, M.B., R. Franz, J.E. Hauser and R.M. Murray (2005). Receding horizon control of vectored thrust flight experiment. *IEE Proc.-Part D: Control theory and Applications* **152**(3), 340–348.

Rawlings, J. B. and K. R. Muske (1993). The stability of constrained receding horizon control. *IEEE Trans. on Automatic Control* **38**(10), 1512–1516.

Richalet, J., A. Rault, J. L. Testud and J. Papon (1978). Model predictive heuristic control: Applications to industrial processes. *Automatica* **14**(5), 413–428.

Rossiter, J.A. and J. Richalet (2002). Handling constraints with predictive fucntional control of unstable process. In: *Proceedings of American Control Conference*. United States. pp. 4746–4751.

Sepulchre, R., M. Jankovic and P. V. Kokotovic (1996). *Constructive Nonlinear Control*. Springer-Verlag. Berlin.