



Predictive pole-placement control with linear models[☆]

Peter J. Gawthrop^{a, *}, Eric Ronco^b

^aCentre for Systems and Control, Department of Mechanical Engineering, University of Glasgow, Glasgow G12 8QQ, Scotland, UK

^bSIMULOG, 1 rue James Joule, 78286 Guyancourt Cedex, France

Received 7 July 1999; revised 20 March 2001; received in final form 13 September 2001

Abstract

The predictive pole-placement control method introduced in this paper embeds the classical pole-placement state feedback design into a quadratic optimisation-based model-predictive formulation. This provides an alternative to model-predictive controllers which are based on linear–quadratic control. The theoretical properties of the controller in a linear continuous-time setting are presented and a number of illustrative examples are given. These results provide the foundation for novel linear and nonlinear constrained predictive control methods based on continuous-time models. © 2002 Elsevier Science Ltd. All rights reserved.

Keywords: Model-based control; Predictive control; Pole assignment; Continuous-time systems

1. Introduction

Most work on stabilising model-based predictive control (MPC) can be seen as the approximation of infinite horizon linear–quadratic control using a finite horizon optimisation with constraints. Stability results—for example those of Clarke and Scattolini (1991), Demircioglu and Clarke (1992), Muske and Rawlings (1993), Rawlings and Muske (1993), Chen and Allgöwer (1998) and Mayne, Rawlings, Rao, and Scokaert (2000)—are based on the idea of showing that, with suitable terminal constraints, this approximation is equivalent to the solution of a related infinite horizon cost function.

This paper takes a different approach. Although optimisation via model-based prediction is used, it is *not* an approximation to a linear–quadratic cost function but rather as a means of solving linear constrained problems by approximating the behaviour of the classical linear state feedback control of a linear system with chosen closed-loop pole locations. These pole locations can be determined by any linear design method (including linear–quadratic). For this reason, the algorithm is named *predictive pole placement* (PPP). In common with many

other MPC papers such as those of Muske and Rawlings (1993), Rawlings and Muske (1993), Gawthrop, Demircioglu, and Siller-Alcala (1998) and Chen and Allgöwer (1998), a state (as opposed to output) feedback approach is used; thus the method can be categorised as manipulating input–output behaviour using state feedback. In the linear context of this paper, output feedback may be readily accomplished using standard observer techniques.

Within a continuous-time setting, the basic PPP algorithm for a linear unconstrained system has the following features:

- (1) The open-loop control signal (within the moving horizon) is constrained to be the linear sum of pre-specified basis functions.
- (2) Optimisation is used to minimise the open-loop output response error with respect to a constant setpoint over a finite time at the end of the optimisation horizon. This can be viewed as forcing the response to become nearly constant at the end of the optimisation horizon.
- (3) A particular choice of basis functions, corresponding to the transient response of a stable linear dynamic system, leads to approximately equal open-loop (within the moving horizon) and closed-loop responses both corresponding to the regulator and tracking response of a stable closed-loop system with prespecified stable poles.

[☆] This paper was not presented at any IFAC meeting. This paper was recommended for publication in revised form by Associate Editor Pre-Olof Gutman under the direction of Editor Tamer Basar.

* Corresponding author.

E-mail address: p.gawthrop@eng.gla.ac.uk (P.J. Gawthrop).

Feature (1) is not new: it has been used by Richalet, Rault, Testud, and Papon (1978) and Richalet (1993); the usual discrete-time choice of the control (or control move) at each sample time can be viewed as one such choice (Rawlings & Muske, 1993); a polynomial (in time) set of basis functions has been used in the continuous context by Demircioglu and Gawthrop (1991) and Gawthrop et al. (1998) and Laguerre functions have been used by Wang (2000).

Feature (2) can be viewed as an output-orientated version of the terminal state constraint shown to be important for stability in a number of papers, including those of Clarke and Scattolini (1991), Demircioglu and Clarke (1992), Muske and Rawlings (1993), Rawlings and Muske (1993) and Chen and Allgöwer (1998); we believe that this output-orientated approach is appropriate given the input–output focus of the paper. Not surprisingly, it is likewise important in creating a stable moving-horizon controller.

Sokaert and Rawlings (1998) show that (discrete-time) “constrained linear quadratic regulation” has the property that nominal closed-loop performance is identical to the open-loop predictions and thus shares feature (3) with the (continuous-time) PPP algorithm. One contribution of our paper is to prove feature (3) for the PPP algorithm.

Of course, standard techniques are available to design controllers for the linear unconstrained case and, in this case, the PPP algorithm would be yet another way of achieving the same result. However, the strength of the method is in its extension to constrained systems using the *quadratic programming* (QP) approach to optimisation (Fletcher, 1987); the linear unconstrained case representing an ideal situation with the corresponding nice properties listed above. Although outside the scope of this paper, we note that the method extends in principle to the nonlinear case and some preliminary results are available elsewhere (Gawthrop & Ronco, 2000).

Although much of the MPC literature uses a discrete-time formulation, this paper uses a continuous-time formulation and thus builds on the previous work of Demircioglu and Gawthrop (1991, 1992), Demircioglu and Clarke (1992) and Gawthrop et al. (1998).

The paper is organised as follows. Section 2 considers the unconstrained optimisation problem and provides an explicit solution of the open- and closed-loop controllers. Section 3 gives conditions under which open- and closed-loop control are the same and shows that the PPP algorithm approximates this situation. Section 4 gives a selection of illustrative examples and Section 5 concludes the paper.

2. Unconstrained PPP

This section introduces the class of systems considered in this paper, the corresponding unconstrained optimisa-

Table 1
Some basis functions

	$U^\star(\tau)_{ij}$	Laplace transform
1	1	$1/s$
2	$e^{-\alpha_i \tau}$	$1/(s + \alpha_i)$
3	$e^{-\alpha_i \tau} \cos \omega_i \tau$	$(s + \alpha_i)/((s + \alpha_i)^2 + \omega_i^2)$
4	$e^{-\alpha_i \tau} \sin \omega_i \tau$	$\omega_i/(s + \alpha_i)^2 + \omega_i^2$
5	$[t^{k-1}/(k-1)!]e^{-\alpha_i \tau}$	$1/((s + \alpha_i)^k)$
6	Laguerre(t, α, N)	$\sqrt{2p}(s - \alpha)^{N-1}/(s + \alpha)^N$

tion problem and gives an explicit formula for its solution. Some special cases of the input and setpoint basis functions are considered in Sections 2.1 and 2.2.

The linear systems considered in this paper are described by

$$\begin{aligned} \frac{d}{dt}x(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t), \end{aligned} \quad (1)$$

$$x(0) = x_0,$$

where $x \in \mathfrak{R}^{n_x}$, $y \in \mathfrak{R}^{n_y}$ and $u \in \mathfrak{R}^{n_u}$. x_0 is the system's 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

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

Differential equations (1) and (2) are related by having the *same* state-space matrices and by imposing the *cross-coupling* conditions

$$\begin{aligned} x_0^\star &= x(t), \\ u(t) &= u^\star(t, 0). \end{aligned} \quad (3)$$

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

$$u^\star(t, \tau) = U^\star(\tau)U(t), \quad (4)$$

where $U^\star(\tau)$ is an $n_u \times n_U$ 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. A particular choice of such functions is given in Table 1.

Because (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.

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

Similarly, the *moving-horizon* setpoint $w^\star(t, \tau)$ is linearly parameterised by the n_w components of the column vector $W(t)$ so that

$$w^\star(t, \tau) = W^\star(\tau)W(t), \quad (5)$$

where $W^\star(\tau)$ is an $n_y \times n_w$ matrix of functions of τ . Typically the components $W_i^\star(\tau)$ of $W^\star(\tau)$ will be constant:

$$W_i^\star(\tau) = \begin{cases} 1 & \text{for tracking,} \\ 0 & \text{for regulation} \end{cases} \quad (6)$$

but they could also be the more general functions appearing in Table 1.

In a similar fashion to (3) the actual setpoint $w(t)$ is the value of the *moving-horizon* setpoint at $\tau = 0$:

$$w(t) = w^\star(t, 0). \quad (7)$$

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

$$J(U(t), x(t), W(t)) = \frac{1}{2} \int_{\tau_1}^{\tau_2} (y^\star(t, \tau) - w^\star(t, \tau))^T Q(\tau)(y^\star(t, \tau) - w^\star(t, \tau)) d\tau, \quad (8)$$

where $Q(\tau)$ is an $n_y \times n_y$ positive-definite output weighting matrix which is possibly a function of τ . To streamline the notation, here after $J(U(t), x(t), W(t))$ will be written as J .

The derivatives of this cost function are denoted by $J_U = \partial J / \partial U$, $J_{UU} = \partial^2 J / \partial U^2$, $J_{Ux} = \partial^2 J / \partial U \partial x$ and $J_{UW} = \partial^2 J / \partial U \partial W$. However, these subscripts do *not* imply derivatives when attached to other symbols apart from J .

With this notation, Lemma 1 gives the solution of this optimisation problem.

Lemma 1 (Explicit solution of unconstrained problem). *When the system (within the moving horizon) is given by (2), the cost function J has a global minimum with respect to $U(t)$ if J_{UU} is not singular. The corresponding minimising $U(t)$ is then given by*

$$U(t) = K_w W(t) - K_x x(t), \quad (9)$$

where

$$K_w = J_{UU}^{-1} J_{UW}, \quad K_x = J_{UU}^{-1} J_{Ux} \quad (10)$$

and

$$J_{UU} = \int_{\tau_1}^{\tau_2} y_U^\star(\tau)^T Q(\tau) y_U^\star(\tau) d\tau, \quad (11)$$

$$J_{Ux} = \int_{\tau_1}^{\tau_2} y_U^\star(\tau)^T Q(\tau) y_x^\star(\tau) d\tau, \quad (12)$$

$$J_{UW} = \int_{\tau_1}^{\tau_2} y_U^\star(\tau)^T Q(\tau) W^\star(\tau) d\tau, \quad (13)$$

where the i th column $y_{Ui}^\star(\tau)$ of $y_U^\star(\tau)$ is the solution of the ODE:

$$\begin{aligned} \frac{d}{d\tau} x_{Ui}^\star(\tau) &= A x_{Ui}^\star(\tau) + B U_i^\star(\tau), \\ y_{Ui}^\star(\tau) &= C x_{Ui}^\star(\tau), \end{aligned} \quad (14)$$

$$x_{Ui}^\star(0) = 0_{n_x},$$

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 i th column of the matrix $U^\star(\tau)$.

$y_x^\star(\tau)$ is the solution of

$$\begin{aligned} \frac{d}{d\tau} x_x^\star(\tau) &= A x_x^\star(\tau), \\ y_x^\star(\tau) &= C x_x^\star(\tau), \end{aligned} \quad (15)$$

$$x_x^\star(0) = 1_{n_x},$$

where 1_{n_x} is a column vector with all of its n_x elements unity.

The closed-loop control is given by

$$u(t) = k_w w(t) - k_x x(t), \quad (16)$$

where

$$k_x = U^\star(0) K_x \quad \text{and} \quad k_w = U^\star(0) K_w \quad (17)$$

and $w(t)$ is given by (7).

Proof. As the system of (2) is linear, the solution can be rewritten as

$$y^\star(t, \tau) = y_U^\star(\tau) U(t) + y_x^\star(\tau) x(t). \quad (18)$$

Differentiating the cost function J (8) with respect to $U(t)$, $J_U(U)$ then gives

$$\begin{aligned} J_U &= \int_{\tau_1}^{\tau_2} \frac{dy^\star(t, \tau)}{dU} Q(\tau)(y^\star(t, \tau) - w^\star(t, \tau)) d\tau \\ &= \int_{\tau_1}^{\tau_2} y_U^\star(\tau)^T Q(\tau)(y_U^\star(\tau) U(t) \\ &\quad + y_x^\star(\tau) x(t) - W^\star(\tau) W(t)) d\tau. \end{aligned} \quad (19)$$

Differentiating (19) again with respect to $U(t)$ gives (11)–(13). Reorganising these equations and setting $J_U = 0$ gives (9).

From (11), J_{UU} is a quadratic form and thus non-negative; linking this to the assumption of this lemma that J_{UU} is non-singular, it follows that J_{UU} is positive definite. From (18), the cost J is quadratic in $U(t)$ and so has a unique global minimum as J_{UU} is positive definite. Eq. (16) follows from (9), (4) and (3).

Remark 1. Equation (9), together with (4), gives the *open-loop control signal* as a function of τ , the initial

state $x(t)$ and the setpoint at time t and the basis functions $U^\star(\tau)$ as

$$u^\star(t, \tau) = U^\star(\tau)K_w W(t) - U^\star(\tau)K_x x(t). \quad (20)$$

Remark 2. The open-loop control signal has two independent terms: a *tracking* term $U^\star(\tau)K_w W(t)$ driven by the setpoint and a *regulation* term $-U^\star(\tau)K_x x(t)$ driven by the initial state $x(0)$.

Remark 3. Eq. (15) has the explicit solution

$$y_x^\star(\tau) = C \exp A\tau 1_{n_x}. \quad (21)$$

2.1. Special forms of $U^\star(\tau)$

It is convenient to choose the following special form of $U^\star(\tau)$:

$$U^\star(\tau) = \text{blockdiag}(U_1^\star(\tau)^T \quad U_2^\star(\tau)^T \quad \dots \quad U_{n_u}^\star(\tau)^T), \quad (22)$$

where $U_i^\star(\tau) \in \mathcal{R}^{n_i}$ (the set of functions defining the i th system input) is generated as the state of the autonomous system

$$\begin{aligned} \frac{d}{d\tau} U_i^\star(\tau) &= A_i U_i^\star(\tau), \\ U_i^\star(0) &= U_{i0}^\star. \end{aligned} \quad (23)$$

Note that (23) has the explicit solution

$$U_i^\star(\tau) = \exp A_i \tau U_{i0}^\star. \quad (24)$$

In this particular case, the open-loop control $u^\star(t, \tau)$ of (4) can be rearranged as

$$u^\star(t, \tau) = U^\star(\tau)U(t) = \tilde{U}(t)\tilde{U}^\star(\tau), \quad (25)$$

where

$$\tilde{U}(t) = \text{blockdiag}(U_1(t)^T \quad U_2(t)^T \quad \dots \quad U_{n_u}(t)^T), \quad (26)$$

$$\tilde{U}^\star(\tau) = (U_1^\star(\tau)^T \quad U_2^\star(\tau)^T \quad \dots \quad U_{n_u}^\star(\tau)^T)^T, \quad (27)$$

$$U(t) = (U_1(t)^T \quad U_2(t)^T \quad \dots \quad U_{n_u}(t)^T)^T. \quad (28)$$

In the single input case ($n_u = 1$):

$$\tilde{U}(t) = U^T(t) \quad \text{and} \quad \tilde{U}^\star(\tau) = U^{\star T}(\tau). \quad (29)$$

In the multi-input case, the i th control signal $u_i(t)$ is given by $u_i(t) = U_i^\star(\tau)^T U_i(t)$. The motivation for the special form of $U^\star(\tau)$ of (22) is the two alternative ways of writing $u^\star(t, \tau)$ displayed in (25). The first form is that used for optimisation in Section 2, the second form is used in the analysis of Section 3.

It is convenient to rewrite the n_u sets of equations (23) in more compact form as

$$\begin{aligned} \frac{d}{d\tau} \tilde{U}^\star(\tau) &= \tilde{A}_u \tilde{U}^\star(\tau), \\ \tilde{U}^\star(0) &= \tilde{U}_0^\star, \end{aligned} \quad (30)$$

where \tilde{A}_u is the block-diagonal matrix

$$\tilde{A}_u = \text{blockdiag}(A_1 \quad A_2 \quad \dots \quad A_{n_u}) \quad (31)$$

and the initial condition \tilde{U}_0^\star is given by

$$\tilde{U}_0^\star = (U_{10}^{\star T} \quad U_{20}^{\star T} \quad \dots \quad U_{n_u 0}^{\star T})^T. \quad (32)$$

Eqs. (30)–(32) can be explicitly solved to give

$$\tilde{U}^\star(\tau) = \exp \tilde{A}_u \tau \tilde{U}_0^\star, \quad (33)$$

$$y^\star(t, \tau) = (C \quad 0) \exp \begin{pmatrix} A & B\tilde{U}(t) \\ 0 & \tilde{A}_u \end{pmatrix} \tau \begin{pmatrix} x^\star(t, 0) \\ \tilde{U}^\star(0) \end{pmatrix}, \quad (34)$$

where \tilde{A}_u is the matrix with the n_u matrices A_i on the diagonal and zero elsewhere.

In a similar fashion, and differentiating (25) with respect to $\tilde{U}(t)$ and setting $x^\star(t, 0) = x(t) = 0$ gives

$$y_U^\star(\tau) = (C \quad 0) \exp \begin{pmatrix} A & B \\ 0 & \tilde{A}_u \end{pmatrix} \tau \begin{pmatrix} 0 \\ \tilde{U}^\star(0) \end{pmatrix}. \quad (35)$$

Eq. (35) is useful for computing $y_U^\star(\tau)$ needed in (11)–(13).

Computation is further simplified in the case where each input has the same set of basis functions—generated by the same transition matrix A_u and initial condition \tilde{U}_0^\star . In this case, $\tilde{U}(t)$ and $\tilde{U}^\star(\tau)$ can be redefined as:

$$\tilde{U}(t) = (U_1(t) \quad U_2(t) \quad \dots \quad U_{n_u}(t))^T, \quad (36)$$

$$\begin{aligned} \frac{d}{d\tau} \tilde{U}^\star(\tau) &= A_u \tilde{U}^\star(\tau), \\ \tilde{U}^\star(0) &= \tilde{U}_0^\star. \end{aligned} \quad (37)$$

Eqs. (36) and (37) are analysed in Section 3.

2.2. Special form of $W^\star(\tau)$

A constant setpoint is implied by the setpoint generating function $W^\star(\tau)$ given by

$$W^\star(\tau) = I_{n_w \times n_w} \quad (\text{the } n_w \times n_w \text{ unit matrix}) \quad (38)$$

and $W(t) = W$ a constant.

Using (5) and (7), the moving-horizon setpoint $w^\star(t, \tau)$ and the actual setpoint $w(t)$ become identical, constant vectors:

$$w(t) = w^\star(t, \tau) = W. \quad (39)$$

Using (1), (16) and (39), the closed-loop system can be written as the autonomous system

$$\begin{aligned} \frac{d}{dt}X(t) &= A_c X(t), \\ u(t) &= k_X X(t), \end{aligned} \quad (40)$$

$$X(0) = X_0,$$

where

$$A_c = \begin{pmatrix} A - Bk_x & Bk_w W \\ 0 & 0 \end{pmatrix}, \quad (41)$$

$$X(t) = (x(t)^T \quad 1)^T, \quad X_0 = (x_0^T \quad 1)^T, \quad (42)$$

$$k_X = (-k_x \quad k_w W). \quad (43)$$

Eq. (40) is useful for the analysis of Section 3.

3. Properties of unconstrained PPP

This section looks at the basic properties of the PPP algorithm and gives the fundamental result on the relationship between the *open-loop* control $u^\star(t, \tau)$ and the *closed-loop* control $u(t)$.

Lemma 1 presents a general algorithm parameterised by the choice of input functions $U^\star(\tau)$ and setpoint functions $W^\star(\tau)$. A key idea in this paper is to choose the input functions $U^\star(\tau)$ (rewritten as $\tilde{U}^\star(\tau)$ in (37) to be the solutions of the autonomous system of (37). Some appropriate functions appear in Table 1. It turns out that a suitable combination of such functions yields an open-loop system that has the same input, state and output as the corresponding closed-loop system. This relationship between open and closed-loop systems is given in the following lemma.

Lemma 2 (Open – and closed – loop responses). *The moving-horizon control signal of (4) is applied to the system of (2) and the closed-loop controller of the form of (16) is applied to the system of (1).*

If

- (1) $\tilde{U}(t)$ is given by (36),
- (2) $\tilde{U}^\star(\tau)$ is generated from (37),
- (3) $[A_u \tilde{U}_0^\star]$ is controllable,
- (4) $w^\star(t, \tau)$ is given by (39),
- (5) the basis-function generating matrix A_u appearing in (37) has the same n_x eigenvalues as the matrix A_c of (41) appearing in (40),

then there exists a unique control vector $U_{\text{opt}}(t)$ of the form of (28) and is given by

$$U_{\text{opt}}(t) = \mathcal{K}_X X(t), \quad (44)$$

such that the solution of the open-loop system (2) is the same as the solution of the closed-loop system (1) in the sense that:

$$u(t + \tau) = u^\star(t, \tau), \quad (45)$$

$$x(t + \tau) = x^\star(t, \tau), \quad (46)$$

$$y(t + \tau) = y^\star(t, \tau). \quad (47)$$

Proof. For compatibility with the open-loop setup, the evolution of the closed-loop system starting at time t and parameterised by τ is considered. Assumption (4) means that the setpoint can be incorporated into the closed-loop system as in (40).

The closed-loop system input $u(t)$ is written as the output of the dynamic system (48) with scalar input $\delta(\tau)$ by rewriting (40) as

$$\frac{d}{d\tau}X(t + \tau) = A_c X(t + \tau) + X_0 \delta(\tau), \quad (48)$$

$$u(t + \tau) = -k_X X(t + \tau),$$

where k_X is given by (43) and $\delta(\tau)$ is the Dirac delta function.²

On the other hand, using (25), the open-loop system can also be written as the output of another dynamic system (49) with output $u^\star(t, \tau)$ and the *same* scalar input $\delta(\tau)$:

$$\frac{d}{d\tau}\tilde{U}^\star(\tau) = A_u \tilde{U}^\star(\tau) + \tilde{U}_0^\star \delta(\tau), \quad (49)$$

$$u^\star(t, \tau) = \tilde{U}(t) \tilde{U}^\star(\tau).$$

The key to this proof is to note that (48) and (49) are linear state-space systems which are invariant with respect to time τ which can be written as the triples $(A_c, x(t), -k_X)$ and $(A_u, \tilde{U}_0^\star, \tilde{U}(t))$, respectively. In particular, when the output equations are discarded, they can be written as the pairs $(A_c, x(t))$ and $(A_u, U^\star(0))$, respectively. Standard system-theoretic results are then used to deduce that the pairs are similar and the corresponding transformation matrix \mathfrak{T} is found. The triples are then *constructed* to be similar by choosing $\tilde{U}(t)$ in terms of $x(t)$ and \mathfrak{T} . Because the inputs to the two systems of (48) and (49) are identical ($\delta(\tau)$), it then follows that the outputs are also identical: $u^\star(t, \tau) = u(t + \tau)$.

Finally, continuity arguments are used to remove one of the controllability assumptions needed for similarity but not for this lemma.

² This δ function has the effect of giving system (48) the initial value of X_0 at $\tau = 0$.

To aid in the understanding of the proof, it is firstly stated for the single-state ($n_x = 1$) case with zero setpoint and then for the general case.

Scalar case: $n_x = 1$. Initially, assume that $x(t) \neq 0$ and define the *transformation* scalar by

$$\mathfrak{T} = \frac{x(t)}{\tilde{U}_0^\star}. \quad (50)$$

It then follows from condition (5) and by comparing (48) and (49) that

$$\tilde{U}^\star(\tau) = \mathfrak{T}^{-1}x(t + \tau). \quad (51)$$

By choosing $\tilde{U}(t) = -k_x \mathfrak{T}$, it follows that $-k_x x(t + \tau) = \tilde{U}(t) \tilde{U}^\star(\tau)$ and thus that $u^\star(t, \tau) = u(t + \tau)$. In particular, it is possible to write

$$\tilde{U}(t) = \mathfrak{K}_x x(t) \quad \text{where } \mathfrak{K}_x = \frac{k_x}{\tilde{U}_0^\star}. \quad (52)$$

Eq. (51) only makes sense if \mathfrak{T} is invertible, i.e., it needs the extra assumption $x(t) \neq 0$. However, (52) gives \mathfrak{K}_x smoothly in terms of $x(t)$. Hence, when $x(t) = 0$, it can be replaced by $x(t) = \varepsilon$ and then (52) can be interpreted as the limit as $\varepsilon \rightarrow 0$.

Single input $n_u = 1$ case: In addition to assumption (3), initially assume that

$$(A_c, X(t)) \quad \text{is controllable} \quad (53)$$

(this assumption will be removed later). Using standard system-theoretic results (see, for example the book of Kailath, 1980) and equating (A_a, B_a) with $(A_c, x(t))$ and (A_b, B_b) with $(A_u, U^\star(0))$, a transformation matrix \mathfrak{T} exists (and is given by (54)), so that (A_a, B_a) and $(A_c, x(t))$ are similar

$$\mathfrak{T}C_0 = \mathfrak{C}_c, \quad (54)$$

where \mathfrak{C}_c and \mathfrak{C}_0 are the *controllability matrices* of the single-input systems (48) and (49), respectively and are given by the usual partitioned matrices:

$$\mathfrak{C}_c = (x(t) \quad A_c x(t) \quad \dots \quad A_c^{n_x-1} x(t)), \quad (55)$$

$$\mathfrak{C}_0 = (\tilde{U}_0^\star \quad A_u \tilde{U}_0^\star \quad \dots \quad A_u^{n_x-1} \tilde{U}_0^\star). \quad (56)$$

It is now possible to choose $\tilde{U}(t)$ as in (57):

$$\tilde{U}(t) = -k_x \mathfrak{T}, \quad (57)$$

which implies that the triples $(A_c, x(t), -k_x)$ and $(A_u, \tilde{U}_0^\star, \tilde{U}(t))$ are also similar; it follows from standard system-theoretic results (see, for example the books of Kailath, 1980; Polderman & Willems, 1997) that

$$u^\star(t, \tau) = u(t + \tau). \quad (58)$$

Furthermore, it is now shown that \mathfrak{T} is *linear* in $x(t)$. In particular, defining $\mathfrak{T}_i = d\mathfrak{T}/dx_i$ where x_i is the i th

component of $x(t)$ and defining α_i^k as the i th column of A_c^k , it follows that

$$\mathfrak{T}_i = (\alpha_i^0 \quad \alpha_i^1 \quad \dots \quad \alpha_i^{n_x-1}) \mathfrak{C}_0^{-1}. \quad (59)$$

This is independent of $x(t)$ for all i and thus \mathfrak{T} is *linear* in $x(t)$. Thus

$$\mathfrak{T} = \sum_{i=1}^{n_x} \mathfrak{T}_i X_i(t), \quad (60)$$

where $X_i(t)$ is the i th component of $X(t)$. Using (57), it follows that

$$\begin{aligned} \tilde{U}(t) &= -k_x \mathfrak{T} = - \sum_{i=1}^{n_x} k_x \mathfrak{T}_i X_i(t) \\ &= - \sum_{i=1}^{n_x} X_i(t) k_x \mathfrak{T}_i = -X(t)^T \mathfrak{K}_X^T, \end{aligned} \quad (61)$$

where \mathfrak{K}_X is defined as

$$\mathfrak{K}_X = (\kappa_1^T \quad \kappa_2^T \quad \dots \quad \kappa_{n_u}^T) \quad (62)$$

and the row vectors κ_i (k_X is a row vector when $n_u = 1$) are given by

$$\kappa_i = k_X \mathfrak{T}_i. \quad (63)$$

In this scalar case, (29) may be used to deduce $U(t)$ to give

$$U(t) = -\mathfrak{K}_X X(t). \quad (64)$$

Removing the extra controllability condition:

From (54), condition (3) is clearly necessary for the existence of \mathfrak{T} and hence \mathfrak{K}_X . However, although the additional condition of (53) that $(A_c, X(t))$ is controllable is required for similarity (which requires that \mathfrak{T} is invertible), this section shows that it is not required here.

Firstly, note that (64) is *linear* in $X(t)$ and therefore the function mapping $X(t)$ to $U(t)$ is *continuous* and therefore well behaved even when $(A_c, X(t))$ is not controllable.

Secondly, note that controllability is generic in that $(A_c, X(t))$ is controllable for almost all $(A_c, X(t))$.

Thus, for those times when $(A_c, X(t))$ is not controllable, (64) holds for a *controllable* system in the neighbourhood of $(A_c, X(t))$ and then, due to the continuity of the function mapping $X(t)$ to $U(t)$, (64) holds everywhere.

General case: In the general case, the previous arguments can be applied to the j th input and thus (64) is replaced by

$$U_j(t) = \mathfrak{K}_j X(t), \quad (65)$$

where \mathfrak{K}_j is given by (59)–(62) and (63) is replaced by

$$\kappa_i = k_{Xj} \mathfrak{T}_i, \quad (66)$$

where k_{Xj} is the j th row of k_X .

Noting from (28) that $U(t)$ comprises the n_u vectors $U_j(t)$ stacked into a single column and defining \mathfrak{K}_X to be the $n_U n_u \times n_X$ matrix comprising the n_u matrices \mathfrak{K}_j stacked vertically

$$\mathfrak{K}_X^T = (\mathfrak{K}_1^T \quad \mathfrak{K}_2^T \quad \dots \quad \mathfrak{K}_{n_X}^T), \quad (67)$$

it follows that (64) is also true for multiple-input systems. Thus (44) is proved. As \mathfrak{K}_X (67) is unique, it follows that U_{opt} in (44) is unique.

This proves (45). Eqs. (46) and (47) follow by observing that as the inputs to the systems (1) and (2) are the same as equation (58) and, from the condition contained in (3), the initial conditions are also the same, similarity implies that the states and outputs must also be the same.

Remark 4. Partitioning the matrix \mathfrak{K}_X as $\mathfrak{K}_X = (-\mathfrak{K}_X, \mathfrak{K}_W)$ where \mathfrak{K}_X is $n_U \times n_X$ and \mathfrak{K}_W is $n_U \times 1$, (64) can be rewritten as

$$U(t) = -\mathfrak{K}_X x(t) + \mathfrak{K}_W. \quad (68)$$

Remark 5. The matrix \mathfrak{K}_X depends, via \mathfrak{C}_c and A_c , on the closed-loop gain matrix k_X . If the pair (A, B) is controllable and $n_u = 1$ there is only *one* k_X corresponding to a given set of closed-loop poles (eigenvalues of A_c); however there are, in general, many possible values of k_X (Kailath, 1980). Therefore (except in the special case) Lemma 2 does *not* imply a unique \mathfrak{K}_X for a given set of closed-loop poles implied by A_u .

Remark 6. This result is independent of the optimisation-based design of Lemma 1.

Remark 7. Eq. (67) gives an explicit expression for \mathfrak{K}_X .

Remark 8. To satisfy conditions (4) and (5), it is necessary that A_u have at least one eigenvalue at zero—corresponding to the constant setpoint.

Lemma 2 paves the way for showing the properties of the algorithm contained in the following theorem.

Theorem 1 (Linear behaviour). *If*

- (1) *the pair $[AB]$ is controllable,*
- (2) *the assumptions of Lemma 2 hold,*
- (3) *the matrix A_u is chosen such that $(A_u - A)$ is negative definite,*
- (4) *the upper horizon τ_2 is given by $\tau_2 = \tau_1 + \Delta\tau$ where $\Delta\tau > 0$,*

then in the limit as $\tau_1 \rightarrow \infty$,

- (1) *the solution of the open-loop systems (2), (4) is the same as the solution of the closed-loop system (1);*
- (2) *the closed-loop system poles (eigenvalues of $A_c = A - Bk_X$) are the n_X eigenvalues of A_u ;*
- (3) $K_X = \mathfrak{K}_X$.

Proof. The general solution of the *moving-horizon* system of (2) with input given by (4) and the cross-coupling conditions of (3) is given by

$$\begin{aligned} x^\star(t, \tau) &= e^{A\tau} x(t) + \int_0^\tau e^{A(\tau-\tau')} U^\star(\tau) d\tau' U(t), \\ y^\star(t, \tau) &= Cx^\star(t, \tau). \end{aligned} \quad (69)$$

The result of Lemma 2 is that there exists a value of $U(t)$, $U_{\text{opt}}(t)$, such that the solution of the open-loop *moving-horizon* system of (2) only contains terms in $e^{A_u\tau}$, that is *no* terms in $e^{A\tau}$. Define $\Delta U(t)$ as the difference between an arbitrary $U(t)$ and the value resulting from Lemma 2, (44):

$$\Delta U(t) = U(t) - U_{\text{opt}}(t). \quad (70)$$

It follows that (69) can be rewritten as

$$\begin{aligned} x^\star(t, \tau) &= e^{A_u\tau} x(t) + \int_0^\tau e^{A(\tau-\tau')} U^\star d\tau' \Delta U(t), \\ y^\star(t, \tau) &= Cx^\star(t, \tau). \end{aligned} \quad (71)$$

Invoking assumption (3), it follows that, as τ increases, the convolution integral multiplying $\Delta U(t)$ will be large compared to the factor $e^{A_u\tau}$ multiplying $x(t)$. Hence, in the limit as $\tau_1 \rightarrow \infty$, the optimal choice of $U(t)$ is that which sets $\Delta U(t) = 0$, that is $U(t) = U_{\text{opt}}$. The result then follows from Lemma 2.

Remark 9. Choosing this particular horizon to weight the output is similar to the standard approach of using terminal state weighting.

Remark 10. Although the proof is stated for the asymptotic case, in practical terms τ_1 should be chosen to be somewhat greater than the slowest time constant of the desired closed-loop system—typically by a factor of 5. A detailed discussion of this is given by Gawthrop (2000).

Remark 11. This result can be extended in a number of ways.

- (1) If condition (3) is reversed ($A - A_u < 0$) then there is no feedback $k_X = 0$.
- (2) If $U^\star(\tau)$ contains more elements than necessary (that is the dimension of A_u is greater than that of A), the elements with the *slowest* time constants are discarded.

4. Examples

This section provides a number of illustrative examples of the main features of the PPP algorithm. Each example is chosen to illustrate a particular aspect of PPP as follows:

Example 4.1. This use a simple example to allow hand calculation of the PPP algorithm.

Example 4.2. This shows that PPP successfully controls a third order, unstable system with unstable inverse. The fact that PPP gives approximate pole-placement for finite horizon prediction is emphasised.

Example 4.3. This illustrates PPP applied to a system with more outputs than inputs ($n_y > n_u$).

Example 4.4. This illustrates the behaviour of PPP acting on a square multi-input, multi-output system in a situation where the conditions of Lemma 2 are *not* satisfied.

4.1. Simple worked example

This example is simple enough for the calculations to be performed by hand and therefore simply illustrates the main results. A simple integrator with gain b is described by

$$A = 0, \quad B = b, \quad C = 1. \quad (72)$$

Consider the regulation problem where the basis function is $U^\star(\tau) = e^{-\alpha\tau}$, that is

$$A_u = -\alpha, \quad (73)$$

that is the moving-horizon control signal $u^\star(t, \tau)$ of (4) becomes

$$u^\star(t, \tau) = U^\star(\tau)U(t) = e^{-\alpha\tau}U(t), \quad (74)$$

where $U(t)$ is a real number in this case. Eq. (2) then becomes

$$\frac{d}{d\tau}x^\star(t, \tau) = -\alpha x^\star(t, \tau) + be^{-\alpha\tau}U(t), \quad (75)$$

$$y^\star(t, \tau) = x^\star(t, \tau),$$

where all quantities are scalars. The solution to (75) is

$$y^\star(t, \tau) = x^\star(t, 0) + \frac{b}{\alpha}(1 - e^{-\alpha\tau})U(t) \quad (76)$$

$$= x(t) + \frac{b}{\alpha}(1 - e^{-\alpha\tau})U(t), \quad (77)$$

where the second equality follows from (3). By direct differentiation, the derivatives of $y^\star(t, \tau)$ with respect to $U(t)$ and $x(t)$, respectively are:

$$y_U^\star(\tau) = \frac{b}{\alpha}(1 - e^{-\alpha\tau}), \quad (78)$$

$$y_x^\star(\tau) = 1. \quad (79)$$

Substituting $y_U^\star(\tau)$ and $y_x^\star(\tau)$ into (11) and (12) gives

$$\begin{aligned} J_{UU} &= \int_{\tau_1}^{\tau_2} y_U^\star(\tau)^T Q(\tau) y_U^\star(\tau) d\tau \\ &= \left(\frac{b}{\alpha}\right)^2 \int_{\tau_1}^{\tau_2} (1 - e^{-\alpha\tau})^2 d\tau \\ &= \left(\frac{b}{\alpha}\right)^2 \int_{\tau_1}^{\tau_2} 1 - 2e^{-\alpha\tau} + e^{-2\alpha\tau} d\tau \\ &= \left(\frac{b}{\alpha}\right)^2 \left[(\tau_2 - \tau_1) + 2\frac{1}{\alpha}(e^{-\alpha\tau_2} - e^{-\alpha\tau_1}) \right. \\ &\quad \left. - \frac{1}{2\alpha}(e^{-2\alpha\tau_2} - e^{-2\alpha\tau_1}) \right] \end{aligned} \quad (80)$$

and

$$\begin{aligned} J_{Ux} &= \int_{\tau_1}^{\tau_2} y_U^\star(\tau)^T Q(\tau) y_x^\star(\tau) d\tau \\ &= \left(\frac{b}{\alpha}\right) \int_{\tau_1}^{\tau_2} (1 - e^{-\alpha\tau}) d\tau \\ &= \left(\frac{b}{\alpha}\right) \left((\tau_2 - \tau_1) + \frac{1}{\alpha}(e^{-\alpha\tau_2} - e^{-\alpha\tau_1}) \right). \end{aligned} \quad (81)$$

As in Theorem 1, let $\tau_2 = \tau_1 + \Delta\tau$. Then it follows that:

$$\lim_{\tau_1 \rightarrow \infty} J_{UU} = \left(\frac{b}{\alpha}\right)^2 \Delta\tau, \quad (82)$$

$$\lim_{\tau_1 \rightarrow \infty} J_{Ux} = \left(\frac{b}{\alpha}\right) \Delta\tau, \quad (83)$$

$$\lim_{\tau_1 \rightarrow \infty} K_x = J_{UU}^{-1} J_{Ux} = \frac{\alpha}{b}, \quad (84)$$

$$\lim_{\tau_1 \rightarrow \infty} k_x = U^\star(0)K_x = \frac{\alpha}{b}, \quad (85)$$

where (84) follows from (10) and (85) follows from (17) noting that $U^\star(0) = e^{-\alpha \cdot 0} = 1$ in this case.

The corresponding closed-loop system matrix is

$$A_c = A - Bk_x = 0 - b\frac{\alpha}{b} = -\alpha \quad (86)$$

giving a single closed-loop pole at $s = \alpha$ —that is the closed-loop pole is the same as that implied by (73). This verifies Theorem 1.

Substituting (85) into (76) gives

$$\begin{aligned} y^\star(t, \tau) &= x(t) - \frac{b}{\alpha}(1 - e^{-\alpha\tau})\frac{\alpha}{b}x(t) \\ &= e^{-\alpha\tau}x(t). \end{aligned} \quad (87)$$

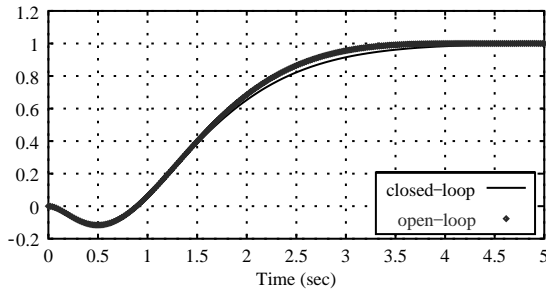


Fig. 1. Unstable system: open- and closed-loop outputs.

As implied by Lemma 2, (87) shows that the open-loop response of $y^*(t, \tau)$ is the same as the closed-loop response of $y(t)$.

It is also worth mentioning that Lemma 2 is explicitly proved for this scalar ($n_x = 1$) case in Section 3.

4.2. Unconstrained control—effect of τ_1

A third-order unstable system with unstable inverse is described by the transfer function

$$G(s) = \frac{1 - 0.5s}{(s - 1)^3}. \quad (88)$$

A state-space representation is

$$A = \begin{pmatrix} 3 & -3 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad (89)$$

$$C = (0 \quad -0.5 \quad 1).$$

The controller is designed with constant setpoint $W^*(\tau) = 1$ and to have three closed-loop poles at $s = -2$. An appropriate A_u matrix is

$$A_u = \begin{pmatrix} -2 & 0 & 0 & 0 \\ -4 & -2 & 0 & 0 \\ -4 & -4 & -2 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \quad (90)$$

In (90), the matrix A_u has two parts:

- (1) the upper left 3×3 matrix which generates the first three Laguerre functions corresponding to entry 6 of Table 1 with $\alpha = 2$. These basis functions appear in Fig. 2.
- (2) the lower right zero entry corresponding to the constant setpoint referred to in Remark 8.

The time horizon is specified by $\tau_1 = 4$; $\tau_2 = 5$. Note that $\tau_1 = 8\tau_d$ where τ_d is the desired time-constant of 0.5. This is in line with Remark 10. The corresponding open-loop (dashed) and closed-loop (firm) step responses appear in Fig. 1. The difference between these is due to

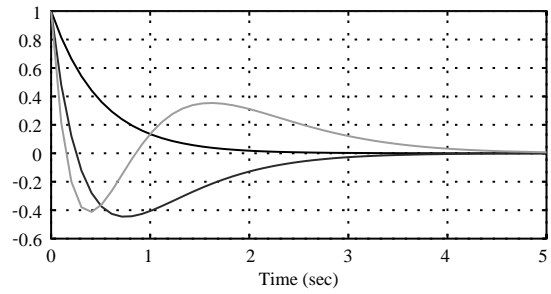


Fig. 2. Unstable system: basis functions.

τ_1 being such that the basis functions $U^*(\tau)$ of Fig. 2 still have significant value at $\tau = \tau_1$.

Remark 12. This example emphasises the point that the open- and closed-loop responses are (theoretically) identical precisely because the open-loop PPP controller removes the free response of the system. In this case, the free response of the system is *unstable* and clearly does not appear in the figure. In other words, Lemma 2 shows that such an open-loop control is always possible; Theorem 1 shows that this is what PPP achieves and this example illustrates the result.

4.3. A one-input two-output system

This example illustrates PPP applied to a system with more outputs than inputs ($n_y > n_u$). The system

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & -1 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad (91)$$

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

which has a two-output one-input transfer function

$$\frac{1}{s^2(s^2 + 2)} \begin{pmatrix} s^2 + 1 \\ 1 \end{pmatrix} \quad (92)$$

has been used by Middleton, Ward, Freudenberg, and Woodyatt (1999) to illustrate performance limitations in resonant systems.

The five basis functions are generated using

$$A_u = \begin{pmatrix} -\alpha & 0 & 0 & 0 & 0 \\ -2\alpha & -\alpha & 0 & 0 & 0 \\ -2\alpha & -2\alpha & -\alpha & 0 & 0 \\ -2\alpha & -2\alpha & -2\alpha & -\alpha & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \quad (93)$$

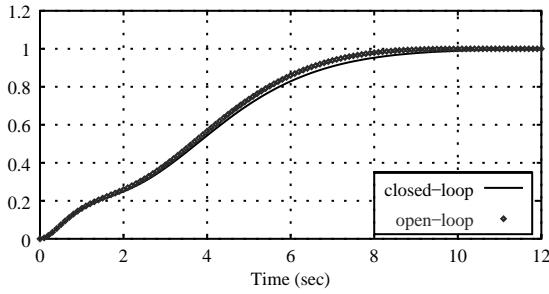


Fig. 3. A one-input two-output system: slow response.

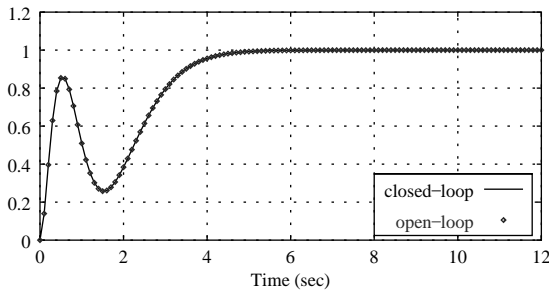


Fig. 4. A one-input two-output system: fast response.

In (93), the matrix A_u has two parts:

- (1) the upper left 4×4 matrix which generates the first four Laguerre functions corresponding to entry 6 of Table 1;
- (2) the lower right zero entry corresponding to the constant setpoint referred to in Remark 8.

Figs. 3 and 4 show step responses when $\alpha = 1$ and 2.5, respectively.

4.4. Simple multivariable system

This example illustrates the behaviour of PPP acting on a square multi-input multi-output system in a situation where the conditions of Lemma 2 are *not* satisfied.

A second-order, two-input two-output system is described by the state-matrices

$$A = \begin{pmatrix} -2 & -1 \\ 1 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 1 \\ 0 & 2 \end{pmatrix}, \quad C = \begin{pmatrix} 0 & 1 \\ 2 & 1 \end{pmatrix}. \quad (94)$$

This corresponds to the transfer-function matrix

$$G(s) = \frac{1}{(1+s)^2} \begin{pmatrix} 1 & 2s+5 \\ 2s+1 & 4s+1 \end{pmatrix}. \quad (95)$$

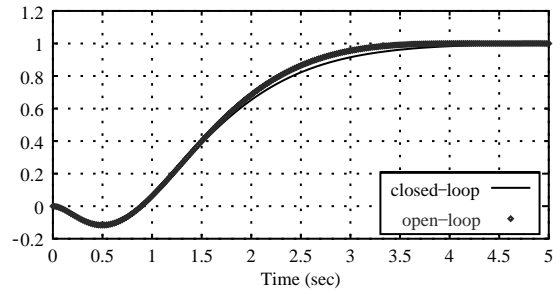


Fig. 5. Simple multivariable system: the two system outputs.

The controller is designed with constant setpoint $W = (1 \ 2)^T$ and A_u is chosen to be

$$A_u = \begin{pmatrix} -3 & 0 \\ 0 & 0 \end{pmatrix}. \quad (96)$$

In (96), the matrix A_u has two parts:

- (1) the upper left 1×1 matrix which generates a single decaying exponential corresponding to $\alpha_i = 3$ in entry 2 of Table 1;
- (2) the lower right zero entry corresponding to the constant setpoint referred to in Remark 8.

This choice of A_u does *not* correspond to the conditions of Lemma 2 as A_u has smaller dimension than A_c . The time horizon is specified by $\tau_1 = 3$; $\tau_2 = 4$. The corresponding closed-loop system outputs appear in Fig. 5. The upper pair line (represented by points) corresponds to the second output (with setpoint 2); the lower firm line corresponds to the first output (with setpoint 1). The corresponding open-loop response is identical.

The computed value of K_x is

$$K_x = \begin{pmatrix} 0.64 & -2.44 & 2.12 \\ 0.00 & 0.00 & -2.25 \\ 0.47 & 1.50 & -1.73 \\ 0.00 & 0.00 & 0.25 \end{pmatrix} \quad (97)$$

and the resultant closed-loop system matrix is

$$A_c = \begin{pmatrix} -3.02 & -0.01 \\ 0.01 & -2.99 \end{pmatrix}, \quad (98)$$

which corresponds to two closed-loop poles at $s = -3.01$.

Thus, in this case, the *single* pole defined by A_u translates into two closed-loop poles at the same location. Although the conditions of Lemma 2 are not satisfied, the resultant closed loop has the specified poles.

5. Conclusion

A new MPC algorithm, predictive pole placement, has been shown to have a number of useful features as

listed in the introduction and these features have been theoretically proved and illustrated by the examples of Section 4.

A popular approach to nonlinear control design is the *exact linearisation* approach of Isidori (1995). This has the disadvantage of cancelling zero dynamics and, in this sense, may be regarded as the nonlinear equivalent of linear design methods such as the minimum variance control of Åström (1970) and the generalised minimum variance control of Clarke and Gawthrop (1975) which cancel system zeros to achieve a closed-loop system with prescribed poles and zeros. In the linear case, the *pole-placement* approach gives a class of algorithms which avoid such cancellation by prescribing closed-loop poles (but not zeros). As yet, there is no nonlinear equivalent of such an approach; one reason being that it is impossible to achieve a linear closed-loop system with prescribed poles without zero cancellation. In this sense, the optimisation-based method presented in this paper provides one possible approach to nonlinear pole placement. Initial results (Gawthrop & Ronco, 2000) have shown that the method can indeed be extended to nonlinear systems; future work will build on the work presented here to further develop the PPP approach to nonlinear control.

Acknowledgements

This work was accomplished whilst the first author was a visitor at the Centre for Integrated Dynamics and Control, University of Newcastle, New South Wales. He would like to thank Professor Graham Goodwin for providing an excellent work environment and other members of the centre, in particular Liuping Wang, Rick Middleton and Will Heath, for helpful suggestions. Both authors would like to thank Professor David Hill of the University of Sydney for facilitating the collaboration between the two authors whilst the second author was a Research Fellow at the University of Sydney. Professor David Mayne and three anonymous referees provided helpful comments which substantially improved the presentation of the ideas in this paper.

References

- Åström, K. J. (1970). *Introduction to stochastic control theory*. New York: Academic Press.
- Chen, H., & Allgöwer, F. (1998). A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34(10), 1205–1217.
- Clarke, D. W., & Gawthrop, P. J. (1975). Self-tuning controller. *IEE Proceedings Part D: Control Theory and Applications*, 122(9), 929–934.
- Clarke, D. W., & Scattolini, R. (1991). Constrained receding horizon predictive control. *IEEE Proceedings*, 138, 347–354.

- Demircioglu, H., & Clarke, D. W. (1992). CGPC with guaranteed stability properties. *Proceedings of IEE*, 139(4), 371–380.
- Demircioglu, H., & Gawthrop, P. J. (1991). Continuous-time generalised predictive control. *Automatica*, 27(1), 55–74.
- Demircioglu, H., & Gawthrop, P. J. (1992). Multivariable continuous-time generalised predictive control. *Automatica*, 28(4), 697–713.
- Fletcher, R. (1987). *Practical methods of optimization* (2nd ed.). Chichester: Wiley.
- Gawthrop, P. J. (2000). Linear predictive pole-placement control: Practical issues. In *Proceedings of the 39th IEEE conference on decision and control*, Sydney, Australia (pp. 160–165).
- Gawthrop, P. J., Demircioglu, H., & Siller-Alcala, I. (1998). Multivariable continuous-time generalised predictive control: A state-space approach to linear and nonlinear systems. *Proceedings of IEE Part D: Control Theory and Applications*, 145(3), 241–250.
- Gawthrop, P. J., & Ronco, E. (2000). A sensitivity bond graph approach to estimation and control of mechatronic systems. *Control Engineering Practice*, 8(11), 1237–1248.
- Isidori, A. (1995). *Nonlinear control systems: An introduction* (3rd ed.). New York: Springer.
- Kailath, T. (1980). *Linear systems*. Englewood, NJ: Prentice-Hall.
- Mayne, D. Q., Rawlings, J. B., Rao, C. V., & Scokaert, P. O. M. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36(6), 789–814.
- Middleton, R. H., Ward, J. K., Freudenberg, J. S., & Woodyatt, A. R. (1999). *Performance limitations in the feedback control of a class of resonant systems*. Technical Report EE9908, Department of Electrical and Computer Engineering, University of Newcastle, 2308 Australia.
- Muske, K. R., & Rawlings, J. B. (1993). Model predictive control with linear models. *Process Systems Engineering*, 39(2), 262–287.
- Polderman, J. W., & Willems, J. C. (1997). *Introduction to mathematical system theory: A behavioral approach*, Texts in applied mathematics, Vol. 26. Berlin: Springer.
- Rawlings, J. B., & Muske, K. R. (1993). The stability of constrained receding horizon control. *IEEE Transactions on Automatic Control*, AC-38(10), 1512–1516.
- Richalet, J. (1993). *Pratique de la Commande Prédictive*. Paris: Hermes.
- Richalet, J., Rault, A., Testud, J. L., & Papon, J. (1978). Model predictive heuristic control: Applications to industrial processes. *Automatica*, 14, 413–428.
- Scokaert, P. O. M., & Rawlings, J. D. (1998). Constrained linear quadratic regulation. *IEEE Transactions on Automatic Control*, 43(8), 1163–1169.
- Wang, L. (2000). Continuous time model predictive controller design using orthonormal functions. In *Proceedings of the UKACC Conference on Control 2000*, Cambridge, UK.



Peter Gawthrop was born in Seascale, Cumberland, in 1952. He obtained his B.A. (first class honours), D.Phil. and M.A. degrees in Engineering Science from Oxford University in 1973, 1977, and 1979, respectively. Following a period as a Research Assistant with the Department of Engineering Science at Oxford University, he became W. W. Spooner Research Fellow at New College, Oxford. He then moved to the University of Sussex as a Lecturer, and later a Reader in control engineering. Since 1987, he has held the Wylie Chair of Control Engineering in the Department of Mechanical Engineering at Glasgow University where he was involved in founding

the Centre for Systems and Control—a cross-departmental research grouping at Glasgow. His research interests include self-tuning control, model-based predictive control, continuous-time system identification and system modelling—particularly using bond graphs in the context of partially-known systems. He is interested in applying modelling and control techniques to a number of areas, including process control, mechatronic systems, aerospace systems and anaesthesia. He has coauthored and authored numerous conference and journal articles and three books in these areas. He is a Fellow of the Institution of Electrical Engineers, a Fellow of the Institution of Mechanical Engineers, a Senior Member of the IEEE, a Chartered Engineer in the UK and a Eur.Eng. in the EU. In 1994 he was awarded the Honeywell International Medal by the Institute of Measurement and Control. In 1999 he spent a year in Australia at the Universities of Newcastle and Sydney.



Eric Ronco is director of a control group within a private company named SIMULOG, specialised in scientific computing. He has been academic researcher for several years focusing on the application of predictive control to large scale nonlinear systems such as power plants, human motor system, chemical process and various mechatronics systems. As a result he has developed a strong knowledge of system modelling and control in general. He is an expert in optimisation techniques for control, identification and system analysis. He is also an expert in physical modelling of systems, particularly through the use of the universal physical language of representation called “Bond Graphs”.