CONTINUOUS-TIME LQ PREDICTIVE POLE-PLACEMENT CONTROL

Peter J Gawthrop [*] **Wen-Hua Chen** [**] **Liuping Wang** [***]

[*] *Centre for Systems and Control and Department of Mechanical Engineering, University of Glasgow, GLASGOW. G12 8QQ Scotland.*
P.Gawthrop@eng.gla.ac.uk
[**] *Department of Aeronautical and Automotive Engineering, Loughborough University, Leicestershire, LE11 3TU, UK*
W.Chen@lboro.ac.uk
[***] *Discipline of Electrical Energy and Control Systems, School of Electrical and Computer Engineering, RMIT University, Melbourne, Victoria 3000, Australia*
liuping.wang@rmit.edu.au

Abstract: The previously developed Predictive Pole Placement (PPP) controller is modified to give enhanced numerical and stability properties by embedding the method in a linear-quadratic formulation to give a linear-quadratic PPP (LQPPP) controller. Input, output and state constraints are considered using an natural quadratic programming (QP) formulation of LQPPP. Illustrative examples are given.

Keywords: Predictive control; pole-assignment; constraint satisfaction problems; quadratic programming; continuous-time systems.

1. INTRODUCTION

Most approaches to model predictive control are derived on the basis of discrete time models, but their corresponding continuous counterpart is still in a relatively immature state of development because of obstacles in obtaining predictions and imposing constraints on the control variable. However, recent work in the continuous-time context (Wang, 2001; Wang, 2004) shows that by using orthonormal functions (Laguerre functions in particular) to describe the trajectory of the control variable, these obstacles can be readily overcome and continuous time predictive control can be solved in a similar framework to the corresponding discrete-time case. Importantly, because of the parsimonious representation of the control trajectory, the algorithm developed is computationally efficient. Cannon and Kouvaritakis (2000) consider quite general functions (including linear splines) to represent deviations about the unconstrained optimal control; Magni and Scattolini (2004) use a piecewise constant set of functions.

Polynomial basis functions have been used in the linear context (Demircioglu and Gawthrop, 1991; Demircioglu and Gawthrop, 1992) and in the nonlinear context (Gawthrop *et al.*, 1998). Although polynomials lead to a nice relationship (Gawthrop *et al.*, 1998) with exact linearisation, they have the disadvantage that they are unbounded. For this reason the use of basis functions which are the initial condition response of a *stable* linear time-invariant (LTI) system – the *basis generator* – have been suggested. As shown by Gawthrop and Ronco (2002) this leads, in the unconstrained case, to a close relationship between the closed-loop system and the basis generator; in particular, under mild conditions, they share the same poles

– hence the name predictive pole-placement control (PPP).

The main motivation for predictive control in general (and PPP in particular) is to handle constraints on the system input, output and states. This is done by embedding the unconstrained controller in a quadratic optimisation which is then converted into a Quadratic Programme (QP) to account for constraints. The real-time solution of the QP is critical to real-time implementation, particularly in the context of fast mechanical systems. As well as the choice of QP algorithm, the *conditioning* of the QP is crucial in obtaining a fast and accurate solution. Unfortunately the basic PPP algorithm does *not* give good conditioning (Gawthrop, 2000); however the extended PPP algorithm presented here, Linear-quadratic PPP (LQPPP), has significantly better conditioning.

The key idea behind LQPPP is that (for linear SISO systems) the solution of a steady-state linear-quadratic regulator problem for a given system with a given set of weighting matrices gives a unique set of closed-loop poles. These closed-loop poles can then be used to specify the basis generator and the original PPP cost function is modified to include the LQ weighting terms – thus the solution of the LQPPP (for the basis function weights) is indeed that of the LQ problem. In this paper, the resultant LQPPP controller is the same as the PPP of Gawthrop and Ronco (2002), but is embedded in the modified cost function. As will be seen by example, this gives a significantly better conditioned QP.

As discussed in more detail by Chen and Gawthrop (2004), LQPPP incorporates a key idea of Rawlings and Muske (1993): the LQ terminal weighting corresponds to the steady-state LQ solution. Because of this, it turns out that unlike PPP, LQPPP has guaranteed stability properties; this is the subject of current investigations (Chen and Gawthrop, 2004).

2. UNCONSTRAINED LQPPP

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. An important advantage of the LQPPP approach is that the lower time horizon $\tau_1 = 0$; this is not possible with PPP (Gawthrop, 2000). The choice of basis generator is then discussed.

The linear systems considered in this paper are described by:

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

where $x \in \Re^{n_x}$, $y \in \Re^{n_y}$ and $u \in \Re^{n_u}$. x_0 is the system initial condition. For simplicity, it is assumed hereafter that $n_u = n_y = 1$; the general case is discussed by Chen and Gawthrop (2004). It is assumed that the pair $[AB]$ is controllable. 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 $y^\star(t,\tau)$ where

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

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 row vector 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.

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.

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:

$$w^\star(t,\tau) = W^\star(\tau)W(t) \tag{5}$$

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

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

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

$$w(t) = w^\star(t,0) \tag{7}$$

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

$$\begin{aligned} &J(U(t),x(t),W(t)) = \\ &\frac{1}{2}\int_{\tau_1}^{\tau_2} (y^\star(t,\tau) - w^\star(t,\tau))^T Q(y^\star(t,\tau) - w^\star(t,\tau))d\tau \\ &+ \frac{1}{2}\int_{\tau_1}^{\tau_2} u^\star(t,\tau)^T R u^\star(t,\tau)d\tau \\ &+ (x^\star(t,\tau_2) - x_w(\tau))^T P(x^\star(t,\tau_2) - x_w(\tau)) \end{aligned} \tag{8}$$

where $Q \in \Re^{n_y \times n_y}$, $R \in \Re$, $P \in \Re^{n_x \times n_x}$ 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 $y^\star(t,\tau) = w(t)$. In the case of LQPPP, $\tau_1 = 0$.

To streamline the notation: $J(U(t),x(t),W(t))$ will be written as J in the sequel.

The derivatives of this cost function are denoted by $J_U = \frac{\partial J}{\partial U}$, $J_{UU} = \frac{\partial^2 J}{\partial U^2}$, $J_{Ux} = \frac{\partial^2 J}{\partial U \partial x}$ and $J_{UW} = \frac{\partial^2 J}{\partial U \partial W}$. 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 Equation 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:

$$U(t) = K_w W(t) - K_x x(t) \tag{9}$$

where

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

and the derivatives are given by

$$\begin{aligned} J_{UU} = &\int_{\tau_1}^{\tau_2} y_U^\star(\tau)^T Q y_U^\star(\tau) d\tau \\ &+ \int_{\tau_1}^{\tau_2} U^\star(\tau)^T R U^\star(\tau) d\tau \\ &+ x_U^\star(\tau_2)^T P x_U^\star(\tau_2) \end{aligned} \tag{11}$$

$$\begin{aligned} J_{Ux} = &\int_{\tau_1}^{\tau_2} y_U^\star(\tau)^T Q y_x^\star(\tau) d\tau \\ &+ x_U^\star(\tau_2)^T P x_x^\star(\tau_2) \end{aligned} \tag{12}$$

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

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

$$\begin{cases} \frac{d}{d\tau} x_{Ui}^\star(\tau) & = A x_{Ui}^\star(\tau) + B U_i^\star(\tau) \\ y_{Ui}^\star(\tau) & = C x_{Ui}^\star(\tau) \\ x_{Ui}^\star(0) & = 0_{n_x} \end{cases} \tag{14}$$

where 0_{n_x} is a column vector with all of its n_x elements zero and $U_i^\star(\tau)$ is the basis function forming the ith element of the matrix $U^\star(\tau)$.

Similarly the ith column $y_{xi}^\star(\tau)$ of $y_x^\star(\tau)$ is the solution of the ordinary differential equation [1] :

$$\begin{cases} \frac{d}{d\tau} x_{xi}^\star(\tau) & = A x_{xi}^\star(\tau) \\ y_{xi}^\star(\tau) & = C x_{xi}^\star(\tau) \\ x_x^\star(0) & = 1_i \end{cases} \tag{15}$$

where 1_i is a column vector with all of its n_x elements zero except for the ith which is unity.

Remark 1. The condition number of the matrix J_{UU} is crucial for determining the numerical accuracy of (10).

[1] The corresponding equation (15) of (Gawthrop and Ronco, 2002) is incorrect

PROOF. The corresponding proof in Gawthrop and Ronco (2002) can be readily modified to include the terms containing R and P. □

As discussed by Gawthrop and Ronco (2002), the closed-loop control is given by

$$u(t) = k_w w(t) - k_x x(t) \tag{16}$$

where:

$$k_x = U^\star(0) K_x,\ k_w = U^\star(0) K_w \tag{17}$$

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

2.1 Basis generator

In the SISO case, it is convenient to choose the following special form of $U^\star(\tau)$:

$$U^\star(\tau) = \tilde{U}^\star(\tau)^T \tag{18}$$

Where $\tilde{U}^\star(\tau) \in \Re^{n_U}$ is generated as the state of the basis generator

$$\begin{cases} \frac{d}{d\tau} \tilde{U}^\star(\tau) & = A_u \tilde{U}^\star(\tau) \\ \tilde{U}^\star(0) & = \tilde{U}_0^\star \end{cases} \tag{19}$$

Note that Equation 19 has the explicit solution:

$$\tilde{U}^\star(\tau) = e^{A_u \tau} \tilde{U}_0^\star \tag{20}$$

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

$$u^\star(t,\tau) = U^\star(\tau) U(t) = \tilde{U}(t) \tilde{U}^\star(\tau) \tag{21}$$

In the case of PPP, A_u is chosen to give the desired (unconstrained) closed-poles, the eigenvalues of A_u; in the case of LQPPP A_u is chosen as follows:

(1) Q and R are chosen, and P is computed as the unique positive definite solution of the Algebraic Riccati Equation (ARE):

$$A^T P + P A - P B R^{-1} B^T P + C^T Q C = 0 \tag{22}$$

together with the corresponding closed-loop system poles and state-feedback gain k_{lq}.

(2) A_u is chosen to have the eigenvalues corresponding to the closed loop poles from step 1. This choice of A_u is not unique, here we use

$$A_u = A - B k_{lq} \tag{23}$$

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 2.

Lemma 2. (Constrained optimisation). Consider the PPP algorithm where the input functions $U^\star(\tau)$ are given by (19). Then given n_{cu} input inequality constraints $\overline{u}^\star(t,\tau_{uk})$ at the n_{cu} times τ_{uk} of the form:

$$u^\star(t,\tau_{uk}) \leq \overline{u}^\star(t,\tau_{uk}) \tag{24}$$

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

$$y^\star(t,\tau_{yk}) \leq \overline{y}^\star(t,\tau_{yk}) \tag{25}$$

the minimisation of the cost function J of Equation 8 subject to the constraints 24 and 25 is equivalent to the solution of the QP:

$$\min_{U(t)} \left\{ \frac{1}{2} U^T(t) J_{UU} U(t) + U^T(t) \left(J_{Ux} x(t) - J_{UW} W(t) \right) \right\} \tag{26}$$

subject to $\Gamma U(t) \geq \gamma$ where J_{UU}, J_{Ux} and J_{UW} are given by Equations 11–13 and

$$\Gamma = \begin{pmatrix} \Gamma_u \\ \Gamma_y \end{pmatrix}, \gamma = \begin{pmatrix} \gamma_u \\ \gamma_y \end{pmatrix} \tag{27}$$

where

$$\Gamma_u = \begin{pmatrix} U^\star(\tau_{u1}) \\ U^\star(\tau_{u2}) \\ \cdots \\ U^\star(\tau_{n_c u}) \end{pmatrix}; \gamma_u = \begin{pmatrix} \overline{u}^\star(t,\tau_{u1}) \\ \overline{u}^\star(t,\tau_{u2}) \\ \cdots \\ \overline{u}^\star(t,\tau_{un_{cu}}) \end{pmatrix} \tag{28}$$

and

$$\Gamma_y = \begin{pmatrix} y_U^\star(\tau_{y1}) \\ y_U^\star(\tau_{y2}) \\ \cdots \\ y_U^\star(\tau_{n_c y}) \end{pmatrix} \tag{29}$$

$$\gamma_y = \begin{pmatrix} \overline{y}^\star(t,\tau_{y1}) - y_x^\star(\tau_{y1}) x(t) \\ \overline{y}^\star(t,\tau_{y2}) - y_x^\star(\tau_{y2}) x(t) \\ \cdots \\ \overline{y}^\star(t,\tau_{yn_{cy}}) - y_x^\star(\tau_{n_c y}) x(t) \end{pmatrix} \tag{30}$$

PROOF. The cost function J of Equation 8 can be rewritten as:

$$J = J_{QP} + J_0(x(t), W(t)) \tag{31}$$

where

$$J_{QP} = \frac{1}{2} U^T(t) J_{UU} U(t) + U^T(t) \left(J_{Ux} x(t) - J_{UW} W(t) \right) \tag{32}$$

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 28 follows from Equation 4 and Equation 29 follows from the fact that the system of Equation 2 is linear, and so that the solution can be rewritten as:

$$y^\star(t,\tau) = y_U^\star(\tau) U(t) + y_x^\star(\tau) x(t) \tag{33}$$

□

Remark 2. As in remark 1, it is clear that the condition number of J_{UU} is crucial to the numerical properties of the QP.

Remark 3. J_{UU}, J_{Ux}, J_{UW}, $u^\star(t,\tau)$ and $y_U^\star(\tau)$ are independent of $U(t)$, $x(t)$ and $W(t)$ and can therefore be precomputed. Γ_u can be precomputed; whereas Γ_y depends on the state $x(t)$ and so has to be computed online.

Remark 4. When no constraints are present, the QP has the same solution as given in Lemma 1. Thus the simple linear solution of Equation 9 provides a good starting point for the QP.

Remark 5. The simplest constraints are those on the input at $\tau = 0$. For example, constraining $u_{min} \leq u^\star(t,0) \leq u_{max}$ maps into

$$\Gamma = \begin{pmatrix} 1 \\ -1 \end{pmatrix} U^\star(0), \gamma = \begin{pmatrix} u_{max} \\ u_{min} \end{pmatrix} \tag{34}$$

4. EXAMPLES

The following two examples illustrate the superior numerical properties of LQPPP when compared with PPP. In particular, the parameter η defined as:

$$\eta = \log_{10} \mathrm{cond}(J_{UU}) \tag{35}$$

where $\mathrm{cond}(J_{UU})$ is the condition number of the symmetric matrix J_{UU} of (11), is a measure of the numerical properties the LQPPP algorithm.

4.1 Unconstrained control of 4 integrators

This section uses the system:

$$y = \frac{1}{s^4} u \tag{36}$$

to illustrate the comparative properties of PPP and LQPPP. The LQ weighting matrices are: $Q = 1$ and $R = 0.1^2$.

Table 1. State-feedback gains & η (35)

Method	k_1	k_2	k_3	k_4	η
1	10.00	14.69	10.80	4.65	-
2	5.65	8.16	7.24	4.08	15.08
3	9.99	14.68	10.79	4.64	7.58
4	14.98	21.42	14.71	5.62	7.40

Table 1 gives the state-feedback gains for four cases

(1) steady-state LQ solution (which gives closed-loop poles at $s = -0.68052 \pm j1.64292$ and $s = -1.64292 \pm j0.68052$),
(2) PPP with the basis function generator (19) having A_u given by (23) and with eigenvalues corresponding to item 1, $\tau_1 = 9$ and $\tau_2 = 10$,

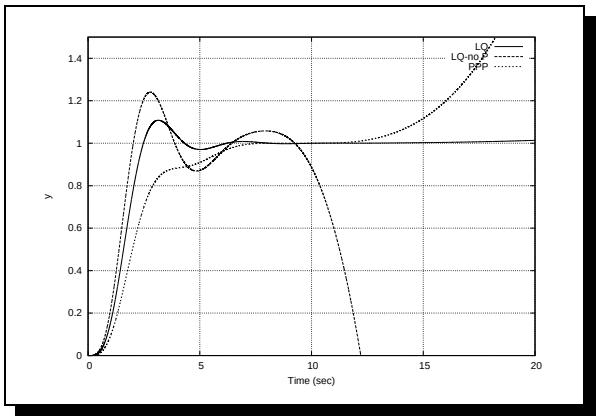


Fig. 1. Open-loop responses: LQPPP, LQPPP without terminal constraint, PPP

(3) LQPPP with the same A_u as item 2, $\tau_1 = 0$ and $\tau_2 = 10$ and
(4) as the LQPPP of item 3 but *without* the terminal weight P.

The gains for LQ and LQPPP are nearly the same; but the gains for PPP are quite different – this is due to the poor numerical conditioning consequent on the high condition number. The gains for the final case differ from LQ as the τ_2 is small enough for the terminal weight P to be significant.

In addition, the table shows η (35). Notice that LQPPP has a condition number about eight orders of magnitude better than that corresponding to PPP. In fact, the condition number for the final case is also small; thus the improvement in condition number is due to setting $\tau_1 = 0$ in (8).

Figure 1 shows the open loop response $y^\star(t,\tau)$ (2) for the latter three cases. The PPP control correctly sets the output $y = 1$ at $t = 10$ but diverges thereafter, the LQPPP control without P is unstable but the LQPPP open-loop control is stable.

4.2 Constrained-output control of non-minimum phase system

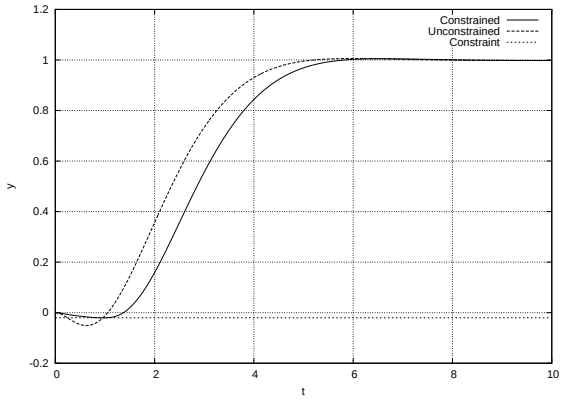
This example illustrates the use of output constraints to reduce the "backwards" response of a non-minimum phase system. It uses the stable, non-minimum phase system given by

$$G(s) = \frac{1-0.5s}{(s+1)^3} \tag{37}$$

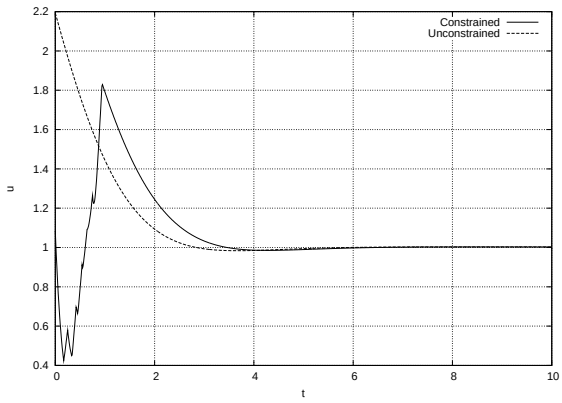
A state-space representation is:

$$A = \begin{pmatrix} -3 & -3 & -1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}; B = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}; C = \begin{pmatrix} 0 & -0.5 & 1 \end{pmatrix} \tag{38}$$

The controller is designed with constant setpoint $W^\star(\tau) = 1$ and using $Q = 1$ and $R = 0.1^2$. This gives the terminal weighting



(a) System output ($y(t)$)



(b) System input ($u(t)$)

Fig. 2. Constrained & unconstrained output control

$$P = \begin{pmatrix} 0.10880 & 0.35008 & 0.30902 \\ 0.35008 & 1.21997 & 1.17034 \\ 0.30902 & 1.17034 & 2.20985 \end{pmatrix} \tag{39}$$

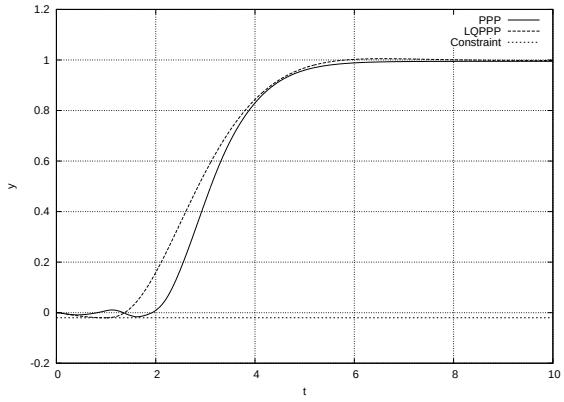
and basis generation matrix A_u (19)

$$A_u = \begin{pmatrix} -3.43520 & -4.40031 & -2.23607 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \tag{40}$$

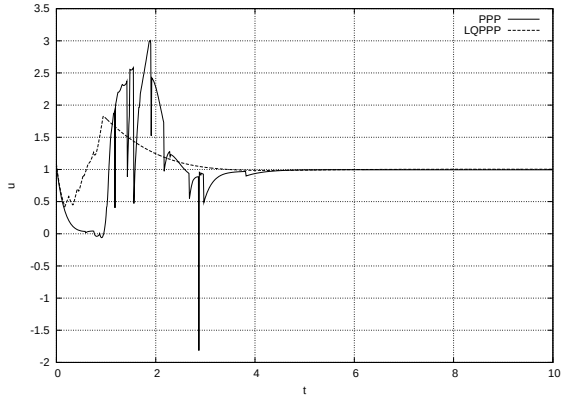
To limit the negative-going portion of the step response, the output constraints were set to be:

$$y^\star(t,\tau) > -0.02 : \tau = 0.1, 0.2, \ldots, 1 \tag{41}$$

Three simulations were performed: LQPPP with and without the constraint (41) and PPP with the constraint (41). Figures 2(a) and 2(b) show the system output $y(t)$ and input $u(t)$ response to a unit step setpoint whilst using LQPPP; for comparison, the unconstrained signals are shown dashed. It can be seen that the constrained response has decreased undershoot at the expense of a slower response. The control signal has a correspondingly complex shape. It is not possible to have zero undershoot with this process whilst actually reaching the setpoint.



(a) System output ($y(t)$)



(b) System input ($u(t)$)

Fig. 3. LQPPP & PPP constrained output control

Figures 3(a) and 3(b) show the system output $y(t)$ and input $u(t)$ response to a unit step setpoint when PPP is used, for comparison, the LQPPP result is shown with dashed lines. Many error messages were generated during the PPP simulation indicating that the no converging solution was found for the QP; this is reflected in the "spiky" form of the control signal; even when convergence occurred, it took about twice as many iterations as in the LQPPP case. Table 2

Table 2. Condition number

Method	PPP	LQPPP
η	8.0	2.1

shows η (35) for each method. The condition number for LQPPP is about 6 orders of magnitude less than that for the PPP method – this is the reason for the improved behaviour of the QP algorithm for LQPPP compared with PPP.

5. CONCLUSION

The continuous-time pole-placement predictive control algorithm has been extended to include input and output constraints. A modified version (LQPPP) of the original algorithm (PPP) has been shown by example to have much improved numerical properties leading to faster and more reliable QP solution. Future work will involve choosing poles and using inverse optimal control (Kalman, 1964) to find the weights.

6. ACKNOWLEDGEMENTS

Will Heath and Adrian Wills of the University of Newcastle, NSW, kindly provided the QP code used in the examples. This research was partially funded by the Royal Academy of Engineering under International Travel Grant ITG 03-553.

REFERENCES

Cannon, M. and B. Kouvaritakis (2000). Infinite horizon predictive control of constrained continuous-time linear systems. *Automatica* **36**(7), 943–955.

Chen, Wen-Hua and Peter J Gawthrop (2004). Constrained predictive pole-placement control with linear models. *Ms in preparation.*

Demircioglu, H. and P. J. Gawthrop (1991). Continuous-time generalised predictive control. *Automatica* **27**(1), 55–74.

Demircioglu, H. and P. J. Gawthrop (1992). Multivariable continuous-time generalised predictive control. *Automatica* **28**(4), 697–713.

Gawthrop, P. J., H. Demircioglu and I. Siller-Alcala (1998). Multivariable continuous-time generalised predictive control: A state-space approach to linear and nonlinear systems. *Proc. IEE Pt. D: Control Theory and Applications* **145**(3), 241–250.

Gawthrop, Peter J (2000). Linear predictive pole-placement control: Practical issues. In: *Proceedings of the 39th IEEE Conference on Decision and Control*. IEEE. Sydney, Australia. pp. 160–165.

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

Kalman, R.E. (1964). When is a linear control system optimal?. *Trans. ASME Journal of Basic Engineering* **86**, 51–60.

Magni, L. and R. Scattolini (2004). Model predictive control of continuous-time nonlinear systems with piecewise constant control. *IEEE Trans. on Automatic Control* **49**(6), 900–906.

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

Wang, L. (2001). Continuous time model predictive control using orthonormal functions. *Int. J. Control* **74**, 1588–1600.

Wang, Liuping (2004). Discrete model predictive controller design using laguerre functions. *Journal of Process Control* **14**, 131–142.