A FAST, EASILY TUNED, SISO, MODEL PREDICTIVE CONTROLLER

Gabriele Pannocchia *,1 **Nabil Laachi** **
James B. Rawlings **

* *Department of Chemical Engineering – Univ. of Pisa*
Via Diotisalvi 2, 56126 Pisa (Italy)
** *Department of Chemical and Biological Engineering – Univ. of Wisconsin 53706 Madison, WI (USA)*

Abstract: It is commonly believed that for SISO systems, well tuned PID controllers work as well as model-based controllers and that PID controllers are more robust to model errors. In this paper we present a novel offset-free constrained LQ controller for SISO systems, which is implemented in an efficient way so that the total controller execution time is similar to that of a PID. The proposed controller has three modules: a state and disturbance estimator, a target calculation, and a constrained dynamic optimization. It is shown that the proposed controller outperforms PID both in setpoint changes and disturbance rejection, it is robust to model errors, it is insensitive to measurement noise, it handles constraints much better than common anti-windup PID. Tuning the proposed controller is simple. In principle, there are three tuning parameters to choose, but in all examples presented only one was actually varied, obtaining a clear and intuitive effect on the closed-loop performance.

Keywords: Model predictive control (MPC), linear quadratic regulation (LQR), PID control, tuning, constraints.

1. INTRODUCTION

Digital PID control for single-input single-output (SISO) systems shows up everywhere in chemical process applications and process control education. Tuning rules are presented in numerous texts and, surprisingly, remain a topic of current control research (Chen and Seborg, 2001; Skogestad, 2003). In this paper we would like to raise the issue of whether this popularity is due to any concrete, technological advantage of digital PID controllers, or whether PID's continuing popularity is simply a historical accident stemming from the success of analog PID controllers. The main technical advantages ascribed to PID control are: PID is simple, fast, and easy to implement in hardware and software; it is easy to tune; it provides good nominal control performance; and it is robust to model errors.

Model-based control methods, such as model predictive control (MPC) of constrained systems, on the other hand, are regarded by many in process control as complex to implement and tune. MPC has become the advanced controller of choice by industry mainly for the economically important, large-scale, multivariable processes in the plant. The rationale for MPC in these applications is that the complexity of implementing MPC is justified only for the important loops with large payoffs.

To address this perception of complexity, we propose a constrained, SISO linear quadratic controller (CLQ) with the following features: it is essentially as fast to execute as PID (within a factor of five regardless of system order), it is easy to implement in software and hardware, and it displays both higher performance and better robustness than PID controllers.

Other researchers have explored the following, related topics. A SISO model predictive controller based on a first-order plus time delay model, with input horizon of one, is proposed by Mukati and Ogunnaike (2004). Soroush and Muske (2000) show that a particular MPC algorithm with input horizon of one, has PI or PID form when the system is first order or second order (without delay), respectively.

[1] Corresponding author. Email: g.pannocchia@ing.unipi.it, Fax: +39 050 511266.

2. OFFSET-FREE CONSTRAINED LQ CONTROLLER FOR SISO SYSTEMS

As most tuning rules for PID controllers require simple transfer function process models, we assume such a model is available. The offset-free constrained linear quadratic (CLQ) control algorithm has three main modules that use a state-space model of the system: a state and disturbance estimator, a constrained target calculation, and a constrained dynamic optimization.

2.1 Model and estimator

We assume that a state-space discrete-time model of the system to be regulated is known:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_{k-m} \\ y_k &= Cx_k \ , \end{aligned} \tag{1}$$

in which $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}$ is the input, $y \in \mathbb{R}$ is the output and m is a non-negative integer, the time delay.

Assumption 1. (General). The pair (A,B) is controllable, the pair (C,A) is observable and

$$\text{rank} \begin{bmatrix} I-A & -B \\ C & 0 \end{bmatrix} = n+1 \ . \tag{2}$$

The input u is assumed to be constrained as follows:

$$u_{\min} \leq u \leq u_{\max} \ , \tag{3}$$

in which $u_{\min} < u_{\max}$.

In order to guarantee offset-free control of y in the presence of plant/model mismatch and/or unmeasured nonzero disturbances, the system model (1) is augmented with an integrating disturbance according to the general methodology proposed in (Pannocchia and Rawlings, 2003). In this work we choose the so-called "input disturbance model", i.e. we add an integrating state d that enters the system at the same place as the input u. The resulting augmented system is:

$$\begin{aligned} \begin{bmatrix} x \\ d \end{bmatrix}_{k+1} &= \begin{bmatrix} A & B \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ d \end{bmatrix}_k + \begin{bmatrix} B \\ 0 \end{bmatrix} u_{k-m} \\ y_k &= [C \ \ 0] \begin{bmatrix} x \\ d \end{bmatrix}_k \ . \end{aligned} \tag{4}$$

Several studies have pointed out that such a disturbance model is an appropriate choice for rejecting unmeasured disturbances efficiently (Muske and Badgwell, 2002), and it provides good robustness to plant/model mismatch (Pannocchia, 2003).

The state x and the disturbance d are estimated from the plant measurement y by means of a steady-state Kalman filter. At each sampling time, an estimate of the state $\hat{x}_{k|k-1}$ and of the disturbance $\hat{d}_{k|k-1}$ based on previous measurements and inputs are available. Thus, the current filtered state and disturbance are:

$$\begin{aligned} \hat{x}_{k|k} &= \hat{x}_{k|k-1} + L_x(y_k - C\hat{x}_{k|k-1}) \\ \hat{d}_{k|k} &= \hat{d}_{k|k-1} + L_d(y_k - C\hat{x}_{k|k-1}) \ , \end{aligned} \tag{5}$$

in which the filter gains $L_x \in \mathbb{R}^n$ and $L_d \in \mathbb{R}$ are computed offline as described later in this paragraph. Given the input u_{k-m} (stored if $m > 0$, or computed as described in the next paragraph, if $m = 0$), the state and disturbance estimates for the next sampling time are:

$$\begin{aligned} \hat{x}_{k+1|k} &= A\hat{x}_{k|k} + Bu_{k-m} + B\hat{d}_{k|k} \\ \hat{d}_{k+1|k} &= \hat{d}_{k|k} \ . \end{aligned} \tag{6}$$

In order to compute the filter gains L_x and L_d, let

$$\hat{A} = \begin{bmatrix} A & B \\ 0 & 1 \end{bmatrix}, \qquad \hat{Q} = \begin{bmatrix} q_x I_n & 0 \\ 0 & 1 \end{bmatrix}, \qquad \hat{C} = [C \ \ 0] \ ,$$

in which q_x is a non-negative scalar. Also let R_v be a positive scalar that represents the output noise variance. Then, the estimator steady-state Riccati equation is

$$\Pi = \hat{Q} + \hat{A}\Pi\hat{A}^T - \hat{A}\Pi\hat{C}^T \left(\hat{C}\Pi\hat{C}^T + R_v\right)^{-1} \hat{C}\Pi\hat{A}^T \ ,$$

in which $\Pi \in \mathbb{R}^{(n+1)\times(n+1)}$ is symmetric semi-definite. Finally, the filter gain is:

$$L = \begin{bmatrix} L_x \\ L_d \end{bmatrix} = \Pi\hat{C}^T \left(\hat{C}\Pi\hat{C}^T + R_v\right)^{-1} \ . \tag{7}$$

Strictly speaking, q_x and R_v should be regarded as the estimator tuning parameters and more details about their effect are discussed in (Pannocchia *et al.*, 2003). In this paper, fixed values for q_x (0.05) and R_v (0.01) are used for all examples.

2.2 Constrained target calculation

At each sampling time, given the current disturbance estimate $\hat{d}_{k|k}$, we compute the steady-state targets for input and state such that the output ultimately reaches the setpoint. If the input were unconstrained, these targets would simply be the solution to the following square system:

$$\begin{bmatrix} I-A & -B \\ C & 0 \end{bmatrix} \begin{bmatrix} \bar{x}_k \\ \bar{u}_k \end{bmatrix} = \begin{bmatrix} B\hat{d}_{k|k} \\ \bar{y} \end{bmatrix} \ , \tag{8}$$

which exists and is unique because of Assumption 1. However, the solution to (8) may be such that the input target $\bar{u}_k$ violates (3). Moreover, for integrating processes it is possible that a steady state does not exist because of (3). For these reasons, we compute $(\bar{x}_k, \bar{u}_k)$ from the following QP:

$$\begin{aligned} (\bar{x}_k, \bar{u}_k) = \arg\min_{(\bar{x},\bar{u})} & (C\bar{x} - \bar{y})^2 + \\ & \eta \left\| (I-A)\bar{x} - B(\bar{u} + \hat{d}_{k|k}) \right\|_2^2 \end{aligned} \tag{9a}$$

subject to:

$$u_{\min} \leq \bar{u} \leq u_{\max} \ , \tag{9b}$$

in which η is a large positive number.

2.3 Constrained dynamic optimization problem: case of reachable setpoint

Given the current state and disturbance estimates, and given the current steady-state targets $(\bar{x}_k, \bar{u}_k)$, for the cases in which the desired setpoint is reachable, i.e. $C\bar{x}_k = \bar{y}$, we compute the control input by means of the following constrained dynamic optimization problem:

$$\min_{\{v_j\}_{j=0}^{N-1}} \left\{ \sum_{j=0}^{N-1} w_j^T Q w_j + s(v_j - v_{j-1})^2 \right\} + \begin{bmatrix} w_N \\ v_{N-1} \end{bmatrix}^T P \begin{bmatrix} w_N \\ v_{N-1} \end{bmatrix} , \quad (10a)$$

subject to:

$$w_0 = \hat{x}_{k+m|k} - \bar{x}_k, \qquad v_{-1} = u_{k-1} - \bar{u}_k \ , \quad (10b)$$

$$w_{j+1} = Aw_j + Bv_j \ , \quad (10c)$$

$$u_{\min} - \bar{u}_k \leq v_j \leq u_{\max} - \bar{u}_k \ , \quad (10d)$$

in which N is a positive integer, s a positive scalar and $Q = C^T C$. The matrix $P \in \mathbb{R}^{(n+1)\times(n+1)}$ is chosen as the positive semidefinite solution of the Riccati equation:

$$P = \tilde{Q} + \tilde{A}^T P \tilde{A} - \tilde{A}^T P \tilde{B}(\tilde{B}^T P \tilde{B} + s)^{-1} \tilde{B} P \tilde{A} \ , \quad (11)$$

in which

$$\tilde{A} = \begin{bmatrix} A & B \\ 0 & 1 \end{bmatrix}, \qquad \tilde{B} = \begin{bmatrix} B \\ 1 \end{bmatrix}, \qquad \tilde{Q} = \begin{bmatrix} Q & 0 \\ 0 & 0 \end{bmatrix} . \quad (12)$$

Let $v := (v_0, v_1, \ldots, v_{N-1})$ be a column vector of length N. We can write (10) as a strictly convex QP:

$$\min_v \frac{1}{2} v^T H v + v^T \tilde{c} \quad (13a)$$

subject to:

$$\mathbf{1} \cdot (u_{\min} - \bar{u}_k) \leq v \leq \mathbf{1} \cdot (u_{\max} - \bar{u}_k) \ , \quad (13b)$$

in which

$$H = \mathscr{B}^T \mathscr{Q} \mathscr{B} + \mathscr{D}^T \mathscr{R} \mathscr{D}, \ \tilde{c} = \mathscr{B}^T \mathscr{Q} \mathscr{A} w_0 + \mathscr{D}^T \mathscr{R} \mathscr{C} v_{-1} \quad (14)$$

and the constant matrices $\mathscr{A}$, $\mathscr{B}$, $\mathscr{C}$, $\mathscr{D}$, $\mathscr{Q}$, $\mathscr{R}$ are not shown the sake of space (Pannocchia *et al.*, 2003). Let $v^* = (v_0^*, \ldots, v_{N-1}^*)$ denote the optimal solution to (13). Then, the current control input is defined by using a receding horizon implementation, that is:

$$u_k = \bar{u}_k + v_0^* \ . \quad (15)$$

2.4 Constrained dynamic optimization problem: case of unreachable setpoint

When the setpoint is not reachable, i.e. when

$$\bar{y}_k = C\bar{x}_k \neq \bar{y} \ , \quad (16)$$

the optimization problem (10) needs to be modified because the corresponding optimal input would drive the controlled variable to the reachable target $\bar{y}_k$ as quickly as possible. There are important cases in which this behavior is undesirable. These situations occur when a "large" disturbance enters the system, and the input constraints are such that the input will asymptotically saturate without rejecting completely the disturbance, and hence offset will occur. It is clear that, if the disturbance keeps affecting the system, steady-state offset is unavoidable but even in such cases it is desirable to keep the controlled variable close to the desired setpoint $\bar{y}$ as long as possible. This goal can be achieved by modifying the optimization problem (10) with a linear penalty (Bonné *et al.*, 2003)

$$\min_{\{v_j\}_{j=0}^{N-1}} \left\{ \sum_{j=0}^{N-1} w_j^T \left(Qw_j + 2q\right) + s(v_j - v_{j-1})^2 \right\} + \begin{bmatrix} w_N \\ v_{N-1} \end{bmatrix}^T \left(P \begin{bmatrix} w_N \\ v_{N-1} \end{bmatrix} + 2p \right) , \quad (17a)$$

$$\text{subject to (10b)–(10d)} \ , \quad (17b)$$

in which P is given in (11) and

$$q = C^T(\bar{y}_k - \bar{y}), \quad p = \left(I - (\tilde{A} + \tilde{B}\tilde{K})^T\right)^{-1} \begin{bmatrix} q \\ 0 \end{bmatrix} , \quad (18)$$

and $\tilde{K} = -(s + \tilde{B}^T P \tilde{B})^{-1} \tilde{B}^T P \tilde{A}$. It is important to remark that if $\bar{y}_k = \bar{y}$ (i.e. the setpoint is reachable), we obtain the same formulation as in (10).

Let $v := (v_0, v_1, \ldots, v_{N-1})$ be a column vector of length N. We can write (17) as the same strictly convex quadratic program (QP) in (13) in which H is still given in (14) while $\tilde{c}$ is:

$$\tilde{c} = \mathscr{B}^T \mathscr{Q} \mathscr{A} w_0 + \mathscr{B}^T \mathscr{P} + \mathscr{D}^T \mathscr{R} \mathscr{C} v_{-1} \ , \quad (19)$$

in which $\mathscr{P} = \begin{bmatrix} q^T & \cdots & q^T & p^T \end{bmatrix}^T$. Let $v^* = (v_0^*, \ldots, v_{N-1}^*)$ denote the optimal solution to (13) with $\tilde{c}$ given in (19). Then, the current control input is still defined by (15).

3. PROPERTIES AND IMPLEMENTATION

If constraints are not present, the proposed controller reduces to an infinite horizon LQ controller with target calculation and origin "shifting". Hence, it is easy to show that it is nominally stable for any choice of the tuning parameters. It is also possible to derive a simple sufficiency test for nominal constrained stability using ellipsoid invariant set theory. The details are omitted for the sake of brevity (Pannocchia *et al.*, 2003). This test can be used online to detect if the terminal state is not in the output admissible set and to flag a warning for the operator. If one wishes to have a guarantee of nominal stability, we can easily formulate the regulator with the terminal constraint $w_N = 0$. But since N is chosen fairly small for computational speed, we find the terminal state constraint controller not as robust as the one presented here, and therefore do not recommend it for industrial practice.

Another important property of CLQ is that it guarantees offset-free control whenever the closed-loop system reaches a steady state in which the input is not saturated. This property holds independently of the plant dynamics, and is due to the presence of the integrating state d in (4) (Pannocchia and Rawlings, 2003, Th.1). Unlike PID, the proposed controller does not integrate the tracking error (i.e. $\bar{y} - y_k$). In fact from (5) and (6), one can write

$$\hat{d}_{k+1|k} = \hat{d}_{k|k-1} + L_d(y_k - C\hat{x}_{k|k-1}) \ ,$$

from which it is clear that there is integration of the prediction error (i.e. $y_k - C\hat{x}_{k|k-1}$). This approach is significantly different from integration of the tracking error as in PID control, and it does not require any anti-windup strategy when the input saturates.

Furthermore, unlike PID control, CLQ is a "two-degree of freedom" controller and it can provide both efficient setpoint tracking and disturbance rejection simultaneously. This feature is due to the use of the state and disturbance estimator, which can also be designed to be insensitive to measurement noise.

Two modules of CLQ – the target calculation and the constrained dynamic optimization – require one to solve a quadratic program at each sampling time. In order for the proposed method to be applicable to simple hardware and programming languages, we have developed an efficient method for solving these QPs (Pannocchia *et al.*, 2003). For space limitations details are omitted, but it is worth remarking that the proposed method for solving the constrained dynamic optimization problem (13) differs from the well known explicit LQR solutions (Bemporad *et al.*, 2002), and it is specifically tailored to SISO systems with input bound constraints only. This new method allows one to efficiently solve (13) on-line without using a "proper" QP solver, and also without using LP and QP solvers in the offline table generation as in (Bemporad *et al.*, 2002). Unlike previous research, the dimension of the state vector has no influence on the size of the stored solution table in this new method. Only the control horizon has an effect. This new method has two basic steps:

(1) The offline generation of a solution table using H, $u_{\min}$ and $u_{\max}$: this step involves some matrix inversions, multiplication and sums.
(2) The online table scanning given the current value of c: this step only involves multiplications and sums and checking conditionals. These same operations are required in PID control.

4. ILLUSTRATIVE EXAMPLES

In this section we present a number of examples of common processes to show that the proposed CLQ controller is simpler to tune than a PID controller, is robust to model errors, insensitive to noise measurements and guarantees superior performance both for setpoint changes and disturbance rejections. Since constraints are present, the common anti-windup "velocity" algorithm for PID is used (Ogunnaike and Ray, 1994). Also notice that if $T_d \neq 0$, the derivative action is suppressed when a setpoint change occurs, which is the common industrial practice to avoid the "derivative kick".

4.1 FOPTD system

The first example is a first order plus time delay (FOPTD) system:

$$G_1(s) = \frac{e^{-2s}}{10s+1},$$

sampled with $T_s = 0.25$. The input is constrained: $|u| \leq 1.5$, a horizon of $N = 4$ is used, and in all simulations the setpoint is changed from 0 to 1 at time zero. At time 25 a load disturbance of magnitude -0.25 enters the system; then at time 50 the disturbance magnitude becomes -1 (which makes the setpoint 1 unreachable); finally at time 75 the disturbance magnitude becomes -0.25 again. Fig. 1 shows the simulation results in the nominal case for two CLQ controllers

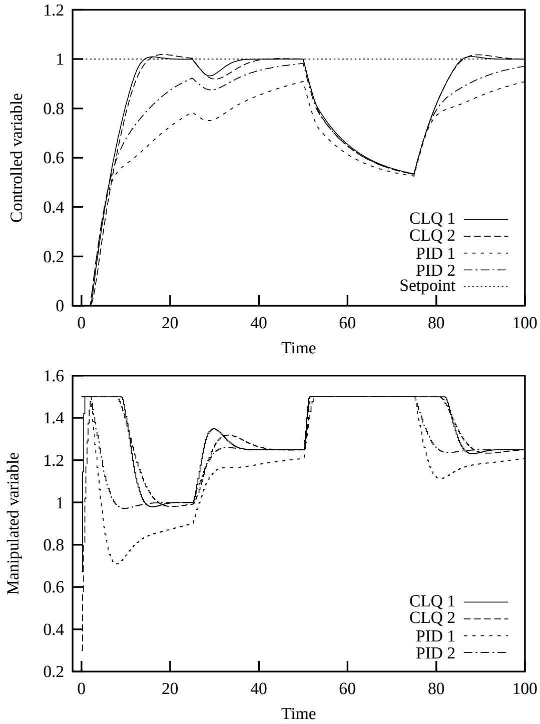

Fig. 1. FOPTD system: nominal case.

and two PID controllers. CLQ 1 uses a regulator input penalty of $s = 1$, while CLQ 2 uses $s = 10$. The tuning parameters for PID 1 are chosen according to Luyben's rules (Luyben and Luyben, 1997, p. 97): $K_c = 2.51$, $T_i = 17.3$, $T_d = 0$. The tuning parameters for PID 2 are chosen according to Skogestad's IMC rules (Skogestad, 2003): $K_c = 2.35$, $T_i = 10$, $T_d = 0$. Fig. 2 shows the simulation results for CLQ 1 and

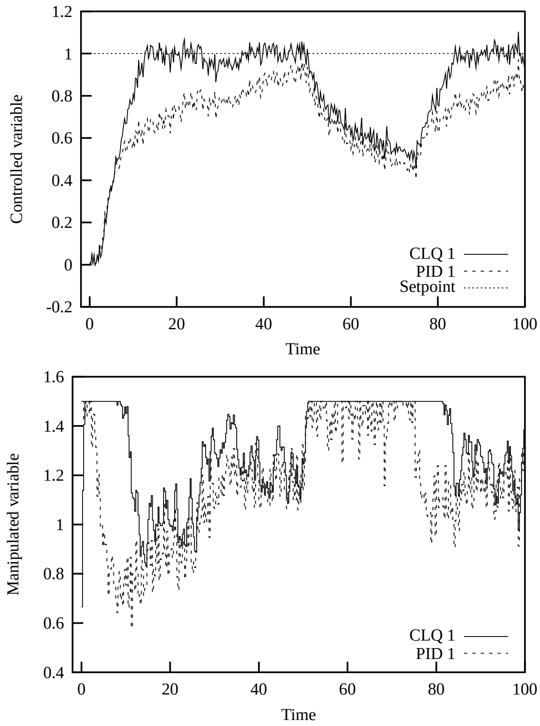

Fig. 2. FOPTD system: noisy case.

PID 1 in the presence of random output noise (with variance $\sigma^2 = 0.001$). Fig. 3 shows a comparison of

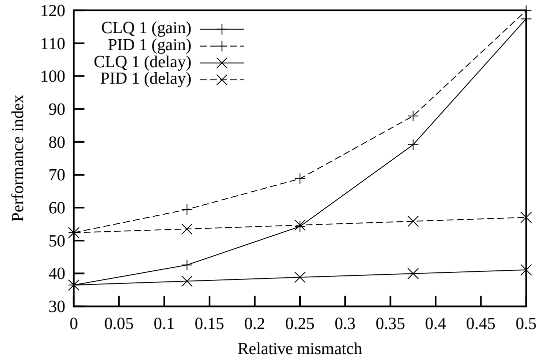

Fig. 3. FOPTD system: effect of plant/model mismatch.

the performance index $\Phi = \sum_{k=0}^{\infty}(y_k - \overline{y})^2 + (u_k - u_{k-1})^2$ *vs* the gain and delay relative plant/model mismatch, respectively, for CLQ 1 and PID 1.

4.2 Integrating system

The second example is an integrating system:

$$G_2(s) = \frac{e^{-2s}}{s} \ ,$$

sampled with $T_s = 0.25$. The same input constraints, horizon, setpoint change and disturbances as in the first example are considered. CLQ 1 uses a regulator input penalty of $s = 500$, while CLQ 2 uses $s = 5000$. The tuning parameters for PID 1 are chosen according to Luyben's rules: $K_c = 0.23$, $T_i = 18.7$, $T_d = 0$. The tuning parameters for PID 2 are chosen according to Skogestad's IMC rules: $K_c = 0.23$, $T_i = 17$, $T_d = 0$. Simulation results in the nominal case are reported

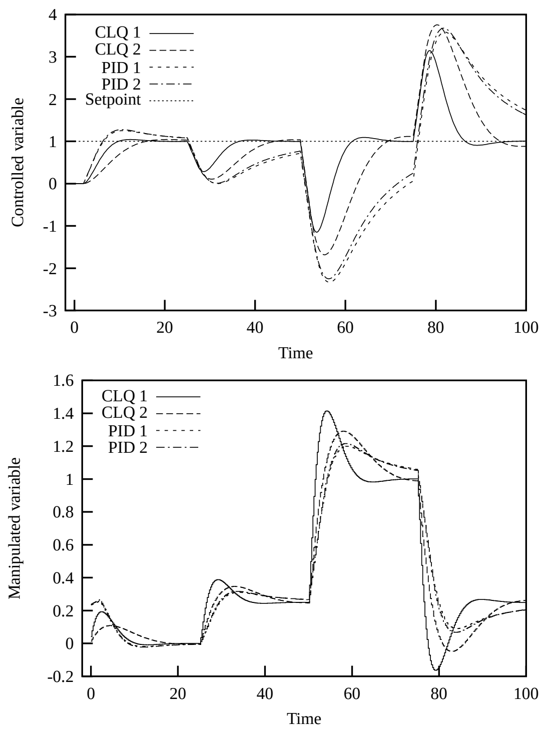

Fig. 4. Integrating system: nominal case.

in Fig. 4, while Φ *vs* the gain and delay relative plant/model mismatch is reported in Fig. 5.

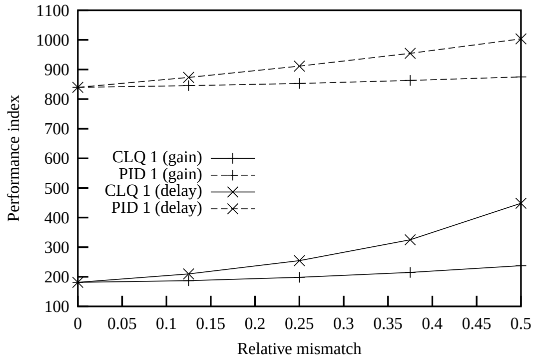

Fig. 5. Integrating system: effect of plant/model mismatch.

4.3 Under-damped system

The last example is a second order under-damped system:

$$G_3(s) = \frac{K}{\tau^2 s^2 + 2\tau\xi s + 1} \ ,$$

sampled with $T_s = 0.25$, and with nominal parameters of $K = 1$, $\tau = 5$ and $\xi = 0.2$. The same input constraints, horizon, setpoint change and disturbances as in the first example are considered. CLQ 1 uses a regulator input penalty of $s = 1$, while CLQ 2 uses $s = 10$. The tuning parameters for PID 1 are chosen according to Luyben's rules: $K_c = 5.0$, $T_i = 16.8$, $T_d = 0$. The tuning parameters for PID 2 are chosen using the same IMC approach as in (Skogestad, 2003): $K_c = 0.4$, $T_i = 2$, $T_d = 12.5$. Simulation results for the

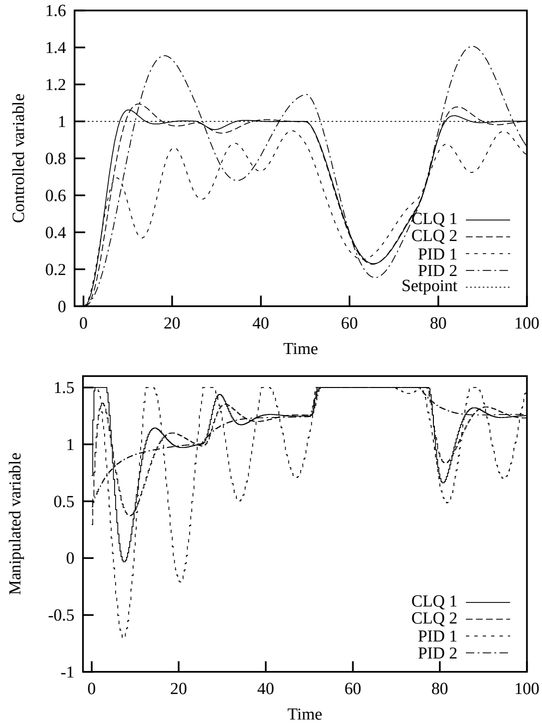

Fig. 6. Under-damped system: nominal case.

nominal case are reported in Fig. 6, while Φ *vs* the gain and damping factor relative uncertainties is reported in Fig. 7.

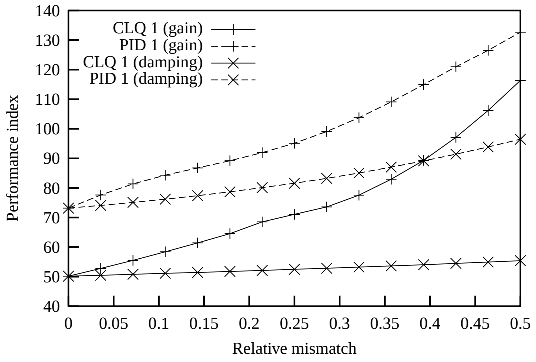

Fig. 7. Under-damped system: effect of plant/model mismatch.

4.4 Discussion

The results presented in the previous paragraphs clearly show that in all examples (as well as in several others not shown for the sake of space) CLQ outperforms PID both for setpoint changes and disturbance rejections. In the presence of constraints CLQ understands much better than PID "when" and "for how long" to saturate the input. Notice that CLQ does not require any anti-windup strategy.

Tuning CLQ is simple: basically one has to choose only the input penalty s, which trades off between tracking error and input usage. The effect of this tuning parameter is intuitive: the lower s the more aggressive the controller. CLQ is robust to plant/model mismatch: one can obtain high performance closed-loop response in the nominal case and still have robust performance and large stability margins. CLQ can be easily designed to be insensitive to measurement noise by adjusting the output noise variance R_v. If one detects high frequency oscillations in the manipulated variable it is sufficient to increase R_v to suppress this undesirable behavior. In this way, it is not necessary to "slow down" the controller's setpoint response (i.e. to increase s) when the measurement is noisy.

Finally, it is important to remark that CLQ is efficient and the computational burden is comparable to that of PID. The average CPU time required to compute the control input has been 0.22 ms for CLQ and 0.05 ms for PID (on a 1.7 GHz Athlon PC running Octave [2]). The maximum CPU time has been 0.55 ms for CLQ and 0.10 ms for PID. The computational efficiency comes about because only a small number of simple operations (addition, multiplication and comparison) are required at each sample time.

5. CONCLUSIONS

In this paper, a novel, offset-free, constrained, linear quadratic (CLQ) controller for SISO system was presented. The purpose of this work was to propose an alternative to digital PID controllers that are commonly available on the DCS. CLQ has three main modules based on a state-space model of the system: a state and disturbance estimator, a target calculation, a constrained dynamic optimization. Each module is implemented in an efficient way so that the overall CLQ algorithm has little computational cost and can be applied using simple hardware and software. As shown, the proposed controller outperforms PID controllers in all situations (setpoint changes or disturbance rejections, nominal case or in the presence of relevant model errors, noise-free or noisy measurements). Moreover, CLQ is much simpler to tune than PID. Strictly speaking, there are two parameters to choose for the estimator and one parameter for the dynamic optimization module. However, as shown in the examples, CLQ can achieve excellent results by varying only this last parameter, whose effect on the closed-loop performance is clear and intuitive. The proposed controller is "scalable", in the sense that it can be extended to larger multivariable systems in a straightforward fashion. Other possible extensions are:

- the use of feed-forward to reject measured disturbances even more efficiently,
- the coupling of several SISO CLQ controllers by appropriate exchange of information.

[2] Octave (`http://www.octave.org`) is freely distributed under the terms of the GNU General Public License.

ACKNOWLEDGMENT

The authors would like to thank Drs. Badgwell, Brambilla, Downs and Ogunnaike for helpful discussion of this paper. The authors gratefully acknowledge the financial support of the industrial members of the Texas-Wisconsin Modeling and Control Consortium and NSF through grant #CTS-0105360.

REFERENCES

Bemporad, A., M. Morari, V. Dua and E. N. Pistikopoulos (2002). The explicit linear quadratic regulator for constrained systems. *Automatica* **38**, 3–20.

Bonné, D., J.B. Jørgensen, J.B. Rawlings and S.B. Jørgensen (2003). Model predictive control with infeasible targets. In preparation.

Chen, D. and D. E. Seborg (2001). PID controller design based on direct synthesis and disturbance rejection. In: *AIChE Annual Meeting*. Reno (NV), USA. p. paper 276i.

Luyben, W. L. and M. L. Luyben (1997). *Essentials of Process Control*. McGraw-Hill Int. Editions.

Mukati, K. and B. Ogunnaike (2004). Stability analysis and tuning strategies for a novel next generation regulatory controller. In: *Amer. Cont. Conf.*. Boston (MA), USA.

Muske, K. R. and T. A. Badgwell (2002). Disturbance modeling for offset-free linear model predictive control. *J. Proc. Cont.* **12**, 617–632.

Ogunnaike, B. A. and W. H. Ray (1994). *Process Dynamics, Modeling, and Control*. Oxford University Press.

Pannocchia, G. (2003). Robust disturbance modeling for model predictive control with application to multivariable ill-conditioned processes. *J. Proc. Cont.* **13**, 693–701.

Pannocchia, G. and J. B. Rawlings (2003). Disturbance models for offset-free model predictive control. *AIChE J.* **49**, 426–437.

Pannocchia, G., N. Laachi and J. B. Rawlings (2003). A candidate to replace PID control: SISO constrained LQ control. Submitted for publication in *AIChE J.*

Skogestad, S. (2003). Simple analytic rules for model reduction and PID controller tuning. *J. Proc. Cont.* **13**, 291–309.

Soroush, M. and K. R. Muske (2000). Analytical model predictive control. In: *Progress in Systems and Control Theory* (F. Allgower and A. Zheng, Eds.). pp. 163–179.