Proceedings of the 23rd IASTED International Conference
MODELLING, IDENTIFICATION, AND CONTROL
February 23-25, 2004, Grindelwald, Switzerland

PARAMETRIC LINEAR QUADRATIC CONTROL: CAN WE AVOID A STATE ESTIMATOR?

David I. Wilson
Department of Electrotechnology
Auckland University of Technology, New Zealand
email: diwilson@aut.ac.nz

ABSTRACT
Parametric linear quadratic (PLQ) control is an alternative to linear quadratic control that avoids the need to explicitly design and tune a state estimator. While the design computation for PLQ is higher than that for LQG, the performance is better for linear plants in the common case where the actual process noise is underestimated. For nonlinear plants, both LQG (using an EKF) and PLQ require very small sample times to maintain stability. It is postulated that the unscented transform (UT) could counter this disadvantage of the EKF, but it is not applicable in the PLQ case.

KEY WORDS
LQ, output optimal, state estimation, Kalman filter

1 Introduction

For practical application, the classical optimal control problem (LQG) has been criticized on at least two key points. The first is the need for a state estimator and the difficulties of tuning the estimator given the invariably poor state measurements required for validation, and the second reason is the lack of robustness in the design algorithm. Notwithstanding, optimal control has many advantages; it is mathematically elegant, readily extendible to nonlinear systems with constraints, and possesses a certain element of satisfaction given that it is, in a word, optimal.

However when it comes to implementation in the process industries, the track record for optimal control at the industrial scale is at best modest. From 1970 to the early 1990s, [1] notes only 22 reported industrial applications of LQG in the process industries although the survey was not intended to be exhaustive. Compared to MPC, the other popular candidate for advanced control in the process industries, this falls below the radar screen. So the question is both why has LQG failed to garner the attention and commercial support that MPC has, and if we were to avoid the state estimator part, but still retain some element of optimality, would LQG become a serious contender to MPC for advanced applications in the process industries? After all, MPC offers little more in the robustness field theoretically than does classical quadratic optimal control, so it behooves one only to address the first deficiency listed above.

A second motivation for this study is to assess the suitability of advanced control schemes such as EKFs or PLQ intended for embedded applications that typically run on digital signal processors (DSPs), or field programmable gate arrays (FPGAs).

2 The parametric LQ problem

Parametric linear quadratic control is a generalisation of the common linear quadratic optimisation control problem which has the potential of deftly side-stepping the state estimator. The optimisation problem is to find the gain, **K**, of an output feedback controller

$$\mathbf{u} = -\mathbf{K}\mathbf{y} \tag{1}$$

that minimises

$$\mathscr{J} = \frac{1}{2}\int_0^\infty \mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{u}^T\mathbf{R}\mathbf{u}\, dt, \quad \mathbf{Q} \geq 0, \mathbf{R} > 0 \tag{2}$$

subject to a linear dynamic plant

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{3}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x}. \tag{4}$$

Note that the output-feedback control law Eqn. 1 is more general than the classical LQR state-feedback case where $\mathbf{u} = -\mathbf{K}\mathbf{x}$.

This, quite naturally, has generated some research effort over the last 2 decades, in part because it avoids the need for a state estimator such as a Kalman filter, [2, 3]. In many industrial applications the dearth of state measurements is matched only by the lack of input variables.

The main problem with parametric LQ control, and probably the main reason for the lack of applications is that the algorithm is an order of magnitude more complex than that for the standard LQG implementation, and the worrying tendency for the design algorithm to fail to converge even under quite benign conditions. Until recently these sort of attributes precluded this algorithm being a reasonable candidate for optimal control implementation.

However now, with improved convergence results, coupled with better implementation schemes, this kind of optimal controller is feasible on a non-trivial piece of equipment such as a 2 degree of freedom bench-scale model helicopter, [4].

Listing 1. MATLAB routine to compute optimal output gain $\mathbf{K}$.

```
function [K,Jp,itn]=ylqr(A,B,C,K,Q,R,tol,alpha);
% [K,Jp,itn] = ylqr(A,B,C,K,Q,R,tol,alpha);
% Continuous output optimal control
% tol = tolerance (1e-4)
% alpha = step length parameter

X= eye(max(size(A))); % Assume E{x(0)x(0)'}
Kold = randn(size(K));   Ktrend = K(:)';
Jp = 1e6; dJ = 10*tol; % large # at start

while (norm(abs(K-Kold),1)>tol) & (abs(dJ)>tol)
   itn = itn+1; % iteration counter
   Kold=K;

   Ac = A - B*K*C; % closed loop
   C1 = C'*K'*R*K*C + Q;
   P = lyap(Ac',C1);
   S = lyap(Ac,X);

   deltaK = R\B'*P*S*C'/(C*S*C') - K;
   K = K + alpha*deltaK;

   dJ = Jp - 0.5*trace(P*X);
   Jp = 0.5*trace(P*X); % performance so far
end % for

return % end ylqr.m
```

2.1 Continuous design procedure

The design equations for the optimum gain $\mathbf{K}$ are

$$\mathbf{A}_c^T\mathbf{P}+\mathbf{P}\mathbf{A}_c = -\mathbf{C}^T\mathbf{K}^T\mathbf{R}\mathbf{K}\mathbf{C}-\mathbf{Q} \quad (5)$$

$$\mathbf{A}_c\mathbf{S}+\mathbf{S}\mathbf{A}_c^T = -\mathbf{X} \quad (6)$$

$$\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}\mathbf{S}\mathbf{C}^T\left(\mathbf{C}\mathbf{S}\mathbf{C}^T\right)^{-1} \quad (7)$$

where the closed loop $\mathbf{A}_c = \mathbf{A}-\mathbf{B}\mathbf{K}\mathbf{C}$, $\mathbf{X} = E\{\mathbf{x}_0\mathbf{x}_0^T\}$, and the performance index in Eqn. 2 is given by $J = \mathrm{tr}(\mathbf{P}\mathbf{X})$. A simple MATLAB routine given in Listing 1, adapted from [2, p370], computes $\mathbf{K}$ by starting with a stabilising trial gain $\mathbf{K}(0)$ and then iterating around Eqns 5–7 using a gradient descent-type algorithm with constant stepsize α.

Of course the naive implementation of the coupled Lyapunov system given in Listing 1 could be improved in a number of ways such as by incorporating an adaptive step size in the optimiser or by only partially computing the Lyapunov solutions in the intermediate iterations.

2.2 Discrete design procedure

For discrete models and objective functions, the design equations are similar and are typically solved via a variant of the descent Anderson-Moore algorithm (DAM). However [5] caution that the convergence properties of this coupled equation system is not well understood and the solution is comparatively computationally expensive although our tests indicate that under 10 iterations usually suffice. Modifications to improve convergence have been proposed by various authors, see e.g. [3].

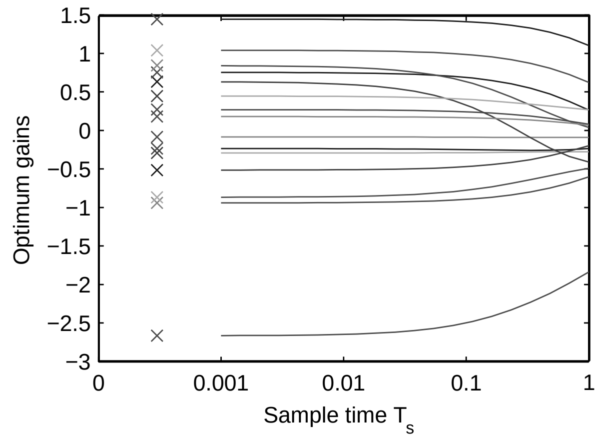


Figure 1. Continuous $\times$ and discrete optimal gains, (—), as a function of sample time for the aircraft model from [6].

One thing to note is that as the sample time tends to zero, the discrete optimal gain converges to the continuous case. Fig. 1 shows that the discrete gains for the continuous model to be used in section 3.1 computed via a discretised version do converge to the continuous gains as expected.

The incentive to use the continuous gain even when using a sampled controller is that then one needs not discretise the linearised model avoiding the expensive matrix exponential calculation. The trick is knowing when this approximation is reasonable, and what sort of speedup we can expect when avoiding the discretisation and matrix exponential.

It is interesting to note that the computationally naive implementation seems to slightly favour the discrete version over the continuous version partially compensating for the load in discretising the continuous system.

For small well-posed continuous or discrete problems it may be more efficient to use a general purpose unconstrained optimiser to search for the elements of $\mathbf{K}$ directly. This computational scheme avoids the strict necessity of finding a stable initial guess for $\mathbf{K}$ which can be a problem in an online implementation.

2.3 The Unscented KF

A recent alternative to the EKF for state estimation of nonlinear systems is the unscented Kalman filter (UKF), [7] which is further developed in [8] and available as a toolbox for MATLAB from [9]. The advantage of the UKF is that it better explicitly captures the possibly convoluted second order statistics of the uncertainty as opposed to the KF which assumes ellipsoid covariance regions. However

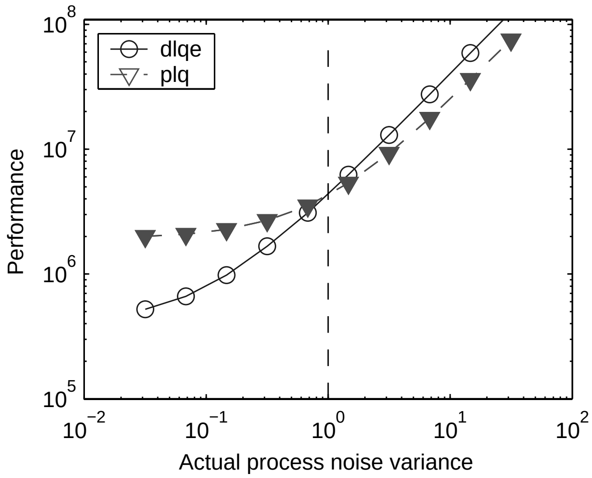


Figure 2. The performance cost, $\mathscr{J}$, of PLQ (▼) and LQG (○) for different process noise levels.

note that the UKF still does not address the issue of hiding the state estimator entirely.

The UKF shares the same computational complexity of the KF, and does not need to compute Jacobians or Hessians of the nonlinear system when used as a state estimator for nonlinear systems.

3 Numerical experiments

The following numerical simulations illustrate some of the points raised so far.

3.1 Linear plants

This section uses the 5 state, 3 input, 3 output aircraft model from [6, p31] as an example of a well-behaved linear multivariable system. The disturbance rejection simulation is run for 50,000 samples to obtain accurate statistics.

Fig. 2 compares the performance of an LQG controlled response with that achievable using PLQ. At the design specifications, (i.e. a process noise variance level of $\sigma_w^2 = 1$, vertical dashed line) both schemes deliver the same level of performance as quantified by the discrete equivalent of Eqn. 2. This gives a check on the numerical implementation of the algorithms.

However if the actual process noise is larger than anticipated (and designed for), then the PLQ scheme out performs the LQG. For smaller process noise sequences, the reverse is true. This makes intuitive sense because in the case of the PLQ scheme, the states are not explicitly calculated

Experience has shown that practictioners tend to underestimate the process noise under industrial conditions given that it is difficult to obtain reliable state measurements for validation, [10]. Furthermore the EKF is known

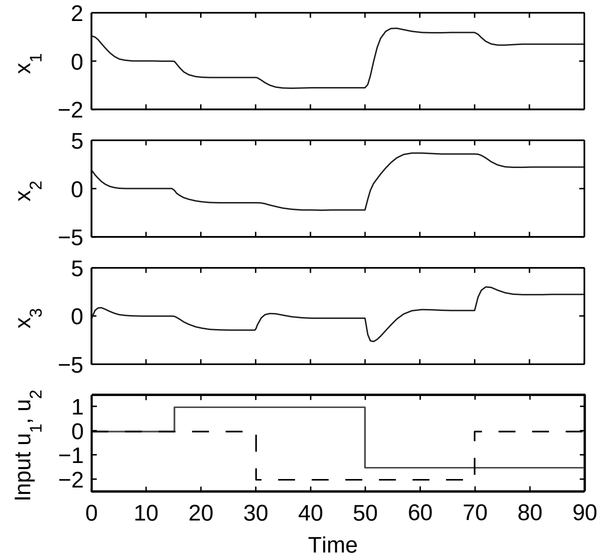


Figure 3. The open-loop response to input step changes for the nonlinear system in Eqns. 8–10.

to typically underestimate the state covariance when used with nonlinear models further compounding this problem.

This trend of the PLQ outperforming the EKF for situations where the process noise is underestimated occurs in most of the other linear plants studied.

3.2 A representative nonlinear plant

Notwithstanding the results for the linear system in section 3.1, the principle interest is to compare the performance of PLQ and LQG for nonlinear systems. The dynamic system

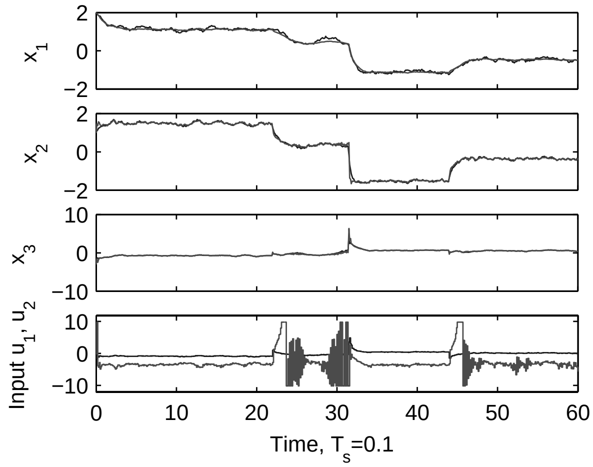
$$\frac{dx_1}{dt} = -x_1 - \frac{x_3}{5} + \tan^{-1}(x_2) \tag{8}$$

$$\frac{dx_2}{dt} = \frac{-x_2}{x_1^2+1} - u_1 \tag{9}$$

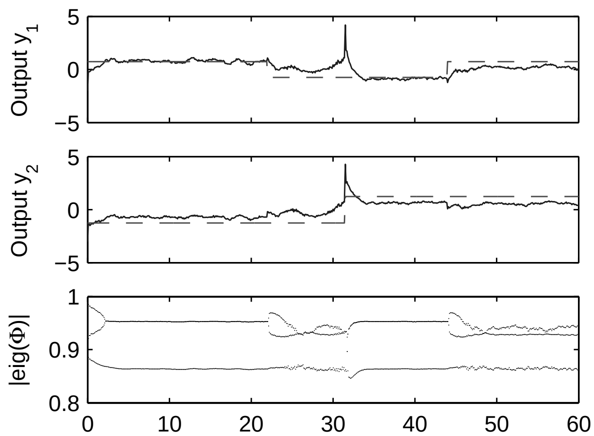
$$\frac{dx_3}{dt} = -x_3 + x_2 - u_2 u_1 \tag{10}$$

is stable and well behaved at low values of $\mathbf{x}$, but is unstable if either $\mathbf{u}$ and/or $\mathbf{x}$ get too large. These characteristics are representative of models used in the process industries where the nonlinearities are differentiable (e.g. Arrhenius expressions for reaction rates), as opposed to mechanical models that are often dominated by non-differentiable nonlinearities such as friction.

The degree of nonlinearity is evident in Fig. 3 which shows the open loop response to step changes in input $\mathbf{u}$. Note that the inverse response characteristics of x_3 around $t = 50$ and the dependance of the overshoot on the input magnitude. Based on the open-loop dynamics, a reasonable sample time would be $T_s \approx 1$ second.

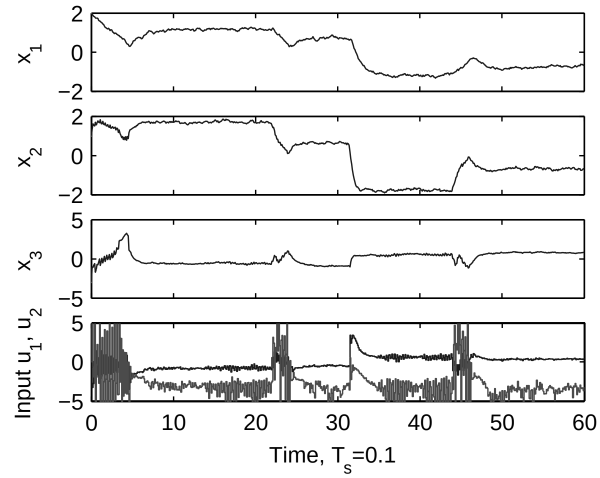


(a) True and estimated states and inputs.

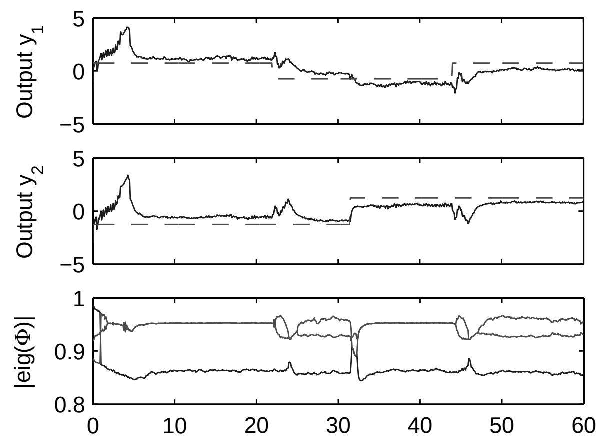


(b) Outputs (solid) and setpoints (dashed), and the instantaneous eigenvalues of the linearised model.

Figure 4. LQG response tracking a reference input.



(a) Actual states and inputs



(b) Outputs (solid) and setpoints (dashed), and the instantaneous eigenvalues of the linearised model.

Figure 5. PLQ response tracking a reference input.

The linearity of the measurement function

$$\mathbf{y} = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{x}$$

is not strictly necessary, but this trivially ensures that the output matrix $\mathbf{C}$ has always full row rank p where p is the number of outputs to be controlled. In practice this is unlikely to be a severe restriction, but must be checked for each particular application.

Fig. 4 illustrates the controlled response for the LQG while Fig. 5 shows the controlled response for the PLQ tracking a reference input with a sample time of $T_s = 0.1$s, considerable smaller than the open loop dynamics may suggest. The input is saturated at $|\mathbf{u}| < 10$ in Fig. 4(a) and < 5 for the PLQ case. Both the variance of the process noise, σ_w^2, and measurement noise, σ_v^2 is 10^{-3}. The magnitude of the eigenvalues of the instantaneously linearised plant is also shown to give an idea of the extent of the nonlinearities.

Fig. 4(a) indicates that the EKF rapidly converged to the true states. This is more a consequence of the benign noise characteristics employed (Gaussian white), the rapid sampling and the modest step changes in reference than any intrinsic robustness of the algorithm. However at the abrupt step changes, the state estimates do deviate causing chattering in the manipulated variable. The controlled response demonstrated in Fig. 4(b) is plausible, but hardly commendable.

Unfortunately the controlled response of the PLQ in Fig. 5(b) is arguably worse. Since the states are not explicitly computed in the PLQ algorithm, Fig. 5(a) only trends the actual, but unknown, states. The ISE of the output deviations for the PLQ case is about double that for the LQG simulation.

4 Conclusions

The title of this paper introduced the challenge of constructing a control algorithm that retained the advantages of optimal control, but avoids the need for an explicit estimator. Parametric linear quadratic control (PLQ) is one such candidate in that it has the advantage that one does not need to design and tune a state estimator explicitly, nor guess initial states and their covariances. The drawbacks are the choosing of the initial stabilising gain, and the computational requirements.

For linear plants, the performance of parametric linear control and LQG are very similar, indeed identical at the design conditions. The higher complexity of the PLQ design is partially compensated by the marginally simpler implementation. However the reduced dimension of the tuning for PLQ is matched by the separation principle which enables one to design the estimator independently from the regulator. Of course while the separation principle is not strictly valid for nonlinear systems, it proves adequate in practice.

For conditions that deviate from the design conditions, PLQ performed better than LQG when the process noise was larger than anticipated. Based on past experience in designing Kalman filters for industrial applications, (e.g. in [10]), this situation, rather than the reverse, is more common.

For nonlinear systems, not unexpectedly the EKF out performs the opposition. However in all cases the sample time must be kept small in order to keep the nonlinearities small, and the consequent errors in the state estimation. A small sample time, relative to the plant dynamics incurs an unnecessary computation cost which has implications for applications such as the increasing use of advanced control in embedded applications such as DSPs and FPGAs. On the other hand, at small sample times, it is possible to use the continuous optimal gain with little approximation in a discrete controller thereby saving the costly discretisation step, and, in the case of the PLQ, a marginally more problematic optimal gain solution procedure.

At the outset of this study, we had hoped to use PLQ in high speed applications intended for DSPs and FPGAs that avoided linearisation and subsequent discretisation at every step, had robust and rapid re-design algorithms for the controller gain, and can take relatively large time steps without incurring either a large loss of optimality, or linearisation errors. Unfortunately both the EKF and the PLQ fell short of those characteristics, but our hope is that the optimal state feedback using not an EKF, but the UT to enable us to take larger steps will suffice.

References

[1] A. Johnson. LQG Applications in the Process Industries. *Chemical Engineering Science*, 48(16):2829–2838, 1993.

[2] Frank L. Lewis and Vassilis L. Syrmos. *Optimal Control*. John Wiley & Sons, 2 edition, 1995.

[3] T. Rautert and E.W. Sachs. Computational design of optimal output feedback controllers. Technical Report Nr. 95-12, Universität Trier, FB IV, Germany, June 1995.

[4] Jonas Balderud and David Wilson. A comparison of optimal control strategies for a toy helicopter. In *4th Asian Control Conference, ASCC2002*, pages 1432–1437, SICEC, Singapore, 25–27 September 2002. ISBN: 981-04-6440-1.

[5] Pertti M. Mäkilä and Hannu T. Toivonen. Computational Methods for Parametric LQ Problems – A Survey. *IEEE transactions on Automatic Control*, AC-32(8):658–671, 1987.

[6] Peter Dorato, Chaouki Abdallah, and Vito Cerone. *Linear-quadratic Control: An Introduction*. Prentice–Hall, 1995.

[7] S. Julier, J. Uhlmann, and H.F. Durrant-Whyte. A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Trans. on Automatic Control*, 45(3):477–482, 2000.

[8] Eric A. Wan and Rudolph van der Merwe. The Unscented Kalman Filter. In Simon Haykin, editor, *Kalman filtering and Neural Networks*, chapter 7, pages 221–280. Wiley Interscience, 2001.

[9] Rudolph van der Merwe. *ReBEL: Recursive Bayesian Estimation Library*, 2003. `http://choosh.bme.ogi.edu/rebel/`.

[10] D. I. Wilson, M. Agarwal, and D.W.T. Rippin. Experiences implementing the extended Kalman filter on an industrial batch reactor. *Computers & Chemical Engineering*, 22(11):1653–1672, 1998.