

REVERSE-ENGINEERING EXISTING CONTROLLERS FOR MPC DESIGN

Jan Maciejowski *

* *Department of Engineering, University of Cambridge,
Cambridge, CB2 1PZ, UK,*

jmm@eng.cam.ac.uk

Abstract: We propose a method for finding the cost function and state estimator to be used in MPC, in order to obtain the same controller as a given low-order controller when constraints are inactive, and hence when the MPC controller is operating within its baseline linear regime. This is a very useful starting point for the development of an MPC controller when a successful existing controller is known, but it is desired to add the constraint-handling capabilities of MPC, and perhaps other functionality such as fault-tolerance. *Copyright © IFAC 2007.*

Keywords: Model Predictive Control, Inverse problem, Output feedback

1. INTRODUCTION

Model Predictive Control (MPC) is often applied to systems for which conventional feedback controllers exist, with the objective of improving performance. In such cases an existing controller is given, and it is convenient to begin the design of an MPC controller by replicating the performance of this existing controller. This gives an initial controller tuning. One can add constraint specifications, and observe whether the use of MPC gives benefits as a result of its constraint-aware functionality. One can also change the tuning of the cost function in the initial design in order to improve the performance benefits.

In this paper we assume that we are given a linear plant and a linear (output-feedback) stabilising controller, and we develop a procedure for obtaining the parameters of an infinite-horizon, quadratic-cost MPC controller which gives the same behaviour as the given controller when all constraints are inactive. This controller is the same as an infinite-horizon LQG controller. We assume that both the plant and the given controller have proper rational transfer functions.

In many cases the existing controller will be given in continuous-time form — for example as a set of PI controllers for an industrial process. We discuss alternative procedures for moving to the discrete-time domain required by MPC. (The properness assumption implies that the ‘D’ term of a 3-term PID controller needs to be combined with a low-pass filter — which is usually the case in implementations — before the results of this paper can be used.)

In earlier work (Rowe and Maciejowski, 2000*b*; Rowe and Maciejowski, 2000*a*) we showed how an MPC controller could be obtained which replicated an H_∞ controller obtained by the *loop-shaping* procedure of McFarlane and Glover (McFarlane and Glover, 1990), and hence was likely to have desirable performance and robustness properties. In this paper, however, there is no restriction on the given controller, except that it should stabilise the given plant model. We make use of the works of Alazard et al on finding observer-based realisations of stabilising controllers (Alazard and Apkarian, 1999; Alazard et al., 2004; Delmond et al., 2006).

2. MOTIVATION AND APPROACH

The unique feature of Model Predictive Control (MPC), among practical control strategies, is that constraints are taken account of explicitly from the start of the problem formulation (Maciejowski, 2002). The control problem is defined as a finite-dimensional constrained optimisation problem, with a multi-stage cost defined over a finite horizon, and with constraints on inputs, states and outputs. Most commonly, the cost function is quadratic and the constraints are linear, resulting in a convex quadratic programming problem. Less commonly, 1-norm or ∞ -norm cost functions are used, but still with linear inequality constraints, in which case a linear programming problem results. It is known that when used with a linear time-invariant internal model, the resulting MPC controller is piecewise-linear; in particular, in some neighbourhood of the set-point, in which none of the constraints is active, the resulting controller is a linear time-invariant system.

Whereas it is relatively straightforward to define appropriate constraints, it is often difficult to find a suitable cost function for the optimiser. However, in many cases a satisfactory controller already exists, or can be designed quite easily using other design approaches. In such cases, it would be very useful to find a cost function for which the existing controller is optimal, and use this cost function in the initial MPC problem formulation. In some cases, MPC will give a significant improvement over the existing controller, even with the same cost function, because of its constraint-handling capabilities — that is, it will allow the plant to operate nearer to some constraints in the presence of disturbances, or it will deal with actuator saturation without running into any ‘wind-up’ problems. In other cases, the designer will wish to obtain further benefits by changing the cost function; but in this case, it will be beneficial to proceed by making incremental changes to the cost function corresponding to the pre-existing controller, rather than by starting from nothing.

Note that the existing controller is usually an output feedback controller, so that it may correspond to some augmentation of the plant dynamics (which models disturbances), an observer, and some state feedback gain. So in order to find the cost function optimised by the controller, one also has to determine any model augmentation, and an observer gain.

A procedure for constructing an observer-based realisation of an existing stabilising controller has been developed in (Alazard and Apkarian, 1999), and refined further in (Delmond *et al.*, 2006), particularly for the case that the controller order is lower than the plant order. The output of this

procedure is an augmented model, an observer gain, and a state feedback gain. We require a further step, namely the determination of the cost function optimised by the state feedback gain — which is the classical inverse optimal control problem. The motivation for the development by Alazard *et al.* was similar to ours — to take an existing controller, and to improve it incrementally, either to enhance its robustness properties (Alazard *et al.*, 2004), or to meet some further design objectives (Alazard *et al.*, 2006).

An existing controller is usually specified in continuous time. This is particularly likely if it is a low-order controller with a simple structure, such as a PI or PID controller. MPC controllers, on the other hand, operate in discrete time. The question arises, therefore, of whether one should obtain a discrete-time approximation of the existing controller, and then proceed entirely in discrete time, or whether one should work with the continuous-time controller, but find a discrete-time equivalent cost at the end of the procedure. In this paper we outline the second method.

3. THE CROSS STANDARD FORM

It was remarked in (Kreindler and Jameson, 1972) that the linear state feedback law $u = K_c x$ is optimal for the cost function

$$\int_0^\infty (u - K_c x)^T R (u - K_c x) dt = \int_0^\infty [x^T, u^T] \begin{bmatrix} K_c^T R K_c & -K_c^T R \\ -R K_c & R \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} dt \quad (1)$$

for any $R = R^T > 0$, and thus that the inverse optimal regulator problem (Kalman, 1964; Molinari, 1973) is trivial if quadratic costs with cross-terms between x and u are allowed.

In a kind of generalisation of this observation, in (Delmond *et al.*, 2006) a *generalised plant* (in the sense of (Zhou *et al.*, 1996)) is constructed for which a given stabilising controller is the optimal H_2 and H_∞ controller, because it yields cost 0 in each case. To be specific, given a linear plant $G(s)$ and a stabilising feedback controller $K_0(s)$, (Delmond *et al.*, 2006) define a *cross standard form* to be the state-space realisation of a generalised plant $P = \mathcal{F}_\ell(W, G)$, for some W (where $\mathcal{F}_\ell(\cdot, \cdot)$ denotes the lower fractional transformation), such that $\mathcal{F}_\ell(P, K_0) = 0$, K_0 stabilises P , and $P_{yu} = G$ (see Fig.1). By construction, such a cross standard form yields P such that

$$K_0 = \arg \min_K \|\mathcal{F}_\ell(P, K)\|_2 \quad (2)$$

and thus in a sense solves the inverse H_2 problem for the pair (G, K_0) (and similarly for $\|\cdot\|_\infty$, but that does not concern us in this paper).

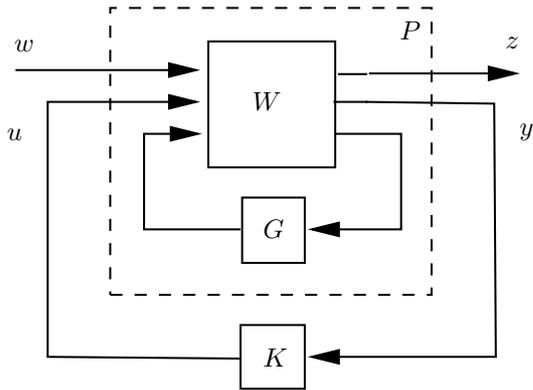


Fig. 1. Structure of generalised plant P .

Suppose that G has a minimal realisation

$$\begin{bmatrix} \dot{x} \\ y \end{bmatrix} = \begin{bmatrix} A & B \\ C & 0 \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} \quad (3)$$

(we assume $D = 0$ for simplicity, but (Delmond *et al.*, 2006) give the general case), and that the controller has minimal realisation

$$\begin{bmatrix} \dot{x}_K \\ u \end{bmatrix} = \begin{bmatrix} A_K & B_K \\ C_K & D_K \end{bmatrix} \begin{bmatrix} x_K \\ y \end{bmatrix} \quad (4)$$

Let $n = \dim x$ and $n_K = \dim x_K$. It is shown in (Delmond *et al.*, 2006) that, in the case of a *low-order controller* — specifically, if $n_K \leq n$ — then the cross standard form has the realisation:

$$\begin{bmatrix} \dot{x} \\ z \\ y \end{bmatrix} = \begin{bmatrix} A & T^\dagger B_K - B D_K & B \\ -C_K T - D_K C & -D_K & I \\ C & I & 0 \end{bmatrix} \begin{bmatrix} x \\ w \\ u \end{bmatrix} \quad (5)$$

where T^\dagger denotes a right-inverse of T , and T is a full-row-rank solution of

$$[-T, I] A_{cl} \begin{bmatrix} I \\ T \end{bmatrix} = 0 \quad (6)$$

where

$$A_{cl} = \begin{bmatrix} A + B D_K C & B C_K \\ B_K C & A_K \end{bmatrix} \quad (7)$$

is the closed-loop dynamic matrix of $\mathcal{F}_\ell(G, K_0)$ (which is stable, by assumption). Note that (6) is a generalised Riccati equation, which can be solved by a numerically stabilised version of the MacFarlane-Potter algorithm (Laub, 1979; van Dooren, 1981). Its solution is not unique, but it is shown in (Delmond *et al.*, 2006) that the corresponding optimal controller is unique, providing that the solution T is chosen so as to satisfy a standard regularity condition on the H_2 problem represented by (5).

A notable feature of (5) is that the cross standard form has the same state vector as the plant.

4. MPC PROBLEM FORMULATION FROM H_2 PROBLEM

We will assume an MPC cost function of the form

$$V_{MPC} = x_N^T P_M x_N + \sum_{k=0}^{N-1} [x_k^T, u_k^T] \begin{bmatrix} Q_M & S_M \\ S_M^T & R_M \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix} \quad (8)$$

with

$$\begin{bmatrix} Q_M & S_M \\ S_M^T & R_M \end{bmatrix} \geq 0 \quad \text{and} \quad P_M \geq 0. \quad (9)$$

Here we assume that the MPC controller operates in discrete time, that x_0 represents the state (estimate) at the time when the MPC controller is deciding which input signal to apply, and N is the length of the horizon over which the controller is ‘planning’, namely over which it evaluates and optimises performance. Most formulations of MPC assume that $S_M = 0$, but this is not necessary (Rao *et al.*, 1998). Also, most practical formulations of MPC replace u_k by

$$\Delta u_k = u_k - u_{k-1} \quad (10)$$

because the steady-state values of the inputs are usually non-zero, and penalising deviations from incorrect steady-state values — which would result from using a model with incorrect steady-state gains — leads to steady-state errors on the controlled variables. The use of cost function (8) with a linear model and linear inequality constraints enforced over the finite horizon of length N leads to a convex quadratic programming optimisation problem, providing that (9) holds. Implementation of an MPC controller requires the repeated on-line solution of this quadratic programming problem, and application of the initial segment of the solution as the input to the plant.

Note that if constraints are not active and a linear model is used, then (8) is equivalent to an infinite-horizon cost if P_M is chosen as the solution of an appropriate Riccati equation (Sznaier and Damborg, 1987; Chmielewski and Manousiouthakis, 1996).

The LQG cost function has the form

$$V_{LQG} = \mathbb{E} \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T [x(t)^T, u(t)^T] \begin{bmatrix} Q_L & S_L \\ S_L^T & R_L \end{bmatrix} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} dt \quad (11)$$

where \mathbb{E} denotes the expected value. It is assumed that $\{x^T, u^T\}$ is a Gaussian stochastic process, whereupon the optimal control is the same state feedback as that which minimises the LQR cost function

$$V_{LQR} = \int_0^\infty [x(t)^T, u(t)^T] \begin{bmatrix} Q_L & S_L \\ S_L^T & R_L \end{bmatrix} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} dt \quad (12)$$

and this optimal state feedback should act on the optimal estimate $\hat{x}(t)$ of the state $x(t)$ (Anderson and Moore, 1990; Kwakernaak and Sivan, 1972).

Finally, the H_2 cost function is

$$V_{H_2} = \|\mathcal{F}_\ell(P, K)\|_2^2 \quad (13)$$

$$= \|C_1\zeta + D_{12}u\|_2^2 \quad (14)$$

when w is an impulse (vector) of unit energy, ζ is the state of the generalised plant P :

$$\begin{bmatrix} \dot{\zeta} \\ z \\ y \end{bmatrix} = \begin{bmatrix} A_P & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{bmatrix} \begin{bmatrix} \zeta \\ w \\ u \end{bmatrix} \quad (15)$$

and

$$\|z\|_2^2 = \int_0^\infty z(t)^T z(t) dt. \quad (16)$$

Since we are considering the case $n_K \leq n$, we have $\zeta = x$ in the Cross Standard Form, and hence we can deduce, from (14), the cost

$$\int_0^\infty [x^T, u^T] \begin{bmatrix} C_1^T C_1 & C_1^T D_{12} \\ D_{12}^T C_1 & D_{12}^T D_{12} \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} dt \quad (17)$$

with, from (5), $C_1 = -C_K T - D_K C$ and $D_{12} = I$. This cost is not unique, since T is not unique (though most solutions for T yield the same optimal controller, as already remarked).¹

The optimal controller K_0 which solves problem (2) has a (generally non-minimal) observer-based realisation

$$K_0 = \left[\begin{array}{c|c} A + BD_K C + K_c B + K_f C & -K_f \\ \hline K_c & D_K \end{array} \right] \quad (18)$$

where the state-feedback gain K_c solves the LQR problem for the cost (17), and hence corresponds to the cost (1). Note that this implies that $K_c = C_1 = -C_K T - D_K C$. The transposed observer gain K_f^T solves the dual LQ problem with cost

$$\int_0^\infty [x^T, w^T] \begin{bmatrix} B_1 B_1^T & B_1 D_{21}^T \\ D_{21}^T B_1^T & D_{21}^T D_{21} \end{bmatrix} \begin{bmatrix} x \\ w \end{bmatrix} dt \quad (19)$$

where $B_1 = T^\dagger B_K - BD_K$ and $D_{21} = I$.

An MPC cost function can be obtained from any of the cost functions (17) or (1) in three steps:

- (1) Obtain an approximately equivalent infinite-horizon cost function in discrete time.
- (2) Obtain an equivalent finite-horizon cost function with terminal cost, of the form (8).
- (3) (Optional.) Obtain an equivalent cost function with Δu_k instead of u_k — see (10).

The MPC problem formulation is completed by using the observer gain K_f to obtain state estimates, and using these estimates in place of x in the MPC cost function and in any state-related constraints (again making appropriate approximations for discrete-time operation of the MPC controller).

¹ Note that this shows that, if the Cross Standard Form exists, then one can always find a state weighting matrix Q_L with $\text{rank}(Q_L) = m$ which yields the given controller K_0 .

To obtain discrete-time equivalents we use the standard assumption of piecewise-constant inputs, changing only at the sample intervals of duration T_s , giving $x_{k+1} = F(T_s)x_k + G(T_s)u_k$, where

$$F(t) = e^{At}, \quad G(t) = \int_0^t e^{A\tau} d\tau B. \quad (20)$$

The cost function equivalent is found as described in (Franklin *et al.*, 1990, section 9.4.4), namely as

$$\sum_{k=0}^{\infty} [x_k^T \ u_k^T] \mathcal{Q} \begin{bmatrix} x_k \\ u_k \end{bmatrix} \quad (21)$$

where

$$\mathcal{Q} = \int_0^{T_s} \begin{bmatrix} F^T(t) & 0 \\ G^T(t) & I \end{bmatrix} \begin{bmatrix} C_1^T C_1 & C_1^T \\ C_1 & I \end{bmatrix} \begin{bmatrix} F(t) & G(t) \\ 0 & I \end{bmatrix} dt \quad (22)$$

since $D_{12} = I$.

An alternative procedure is to obtain an approximately equivalent discrete-time controller first, then use the discrete-time version of the Cross Standard Form (Voinot *et al.*, 2003). This would obviously be preferred if one were given a discrete-time controller initially.

Finally one replaces the infinite-horizon cost (21) by a finite-horizon cost with terminal cost:

$$\sum_{k=0}^{N-1} [x_k^T \ u_k^T] \mathcal{Q} \begin{bmatrix} x_k \\ u_k \end{bmatrix} + x_N^T P_M x_N \quad (23)$$

where P_M solves the discrete algebraic Riccati equation corresponding to the discrete-time LQ problem defined by (21) — since the optimal cost from time N onwards is then given by $x_N^T P_M x_N$. Writing F and G instead of $F(T_s)$ and $G(T_s)$, $P_M \geq 0$ is the stabilising solution of

$$P_M = Q_M + F^T P_M F - (F^T P_M G + S_M)(G^T P_M G + R_M)^{-1}(F^T P_M G + S_M)^T \quad (24)$$

where Q_M, S_M, R_M are sub-blocks of \mathcal{Q} , as in (8). This last step assumes that N is sufficiently large that, in the absence of disturbances and modelling errors, the constraints become inactive by the end of the N -step horizon when the system operates under MPC control with the initial condition x_0 . This allows a finite-dimensional optimisation problem to be solved, despite costing over an infinite horizon.

5. CONTROLLERS WITH DIRECT FEED-THROUGH

Many controllers used in practice, such as PI controllers, have direct “feed-through”, that is, they are not strictly proper ($D_K \neq 0$). But any continuous-time LQG controller obtained as

the solution to a problem with realisable plant, disturbance and noise models, is strictly proper.

We therefore make the standard transformation to an equivalent plant-controller pair (\tilde{P}, \tilde{K}_0) in which both \tilde{P} and \tilde{K}_0 are strictly proper, by defining $\nu = u - D_K y$, which will be the input to the new plant \tilde{P} and the output of the controller $\tilde{K}_0 = K_0 - D_K$ (Zhou *et al.*, 1996). The state-space realisation of \tilde{P}_0 is

$$\tilde{P}_0 = \left[\begin{array}{c|c} \frac{A + BD_K C}{C} & B \\ \hline & 0 \end{array} \right] \quad (25)$$

The procedure described above is now applied to (\tilde{P}, \tilde{K}_0) to find the matrices P_M, Q_M, R_M, S_M . The MPC controller is then applied with the cost function (8) and u_k replaced by $\nu_k = u_k - D_K y_k$ (with transformation to the form involving $\Delta \nu_k$ if desired). Note that, since this is a linear transformation, the MPC problem remains in the standard form, with linear inequality constraints on $u_k, \Delta u_k$ and x_k remaining as linear inequalities on $\nu_k, \Delta \nu_k$ and x_k . Furthermore, since D_K is known exactly, no additional error is introduced into the constraint definitions by this transformation.

It is assumed here that computation time is small compared with the sampling time T_s , so that the control input u_k (or ν_k) is applied a short time after the measurement y_k is obtained, and hence the discrete-time implementation is capable of approximating the continuous-time controller with $D_K \neq 0$.

If the given controller K_0 is given in discrete time, and the whole process of finding an equivalent MPC controller is pursued in discrete time, the same method had to be applied to deal with $D_K \neq 0$. It is possible to have an LQG controller with $D_K \neq 0$ by using the ‘‘estimator’’ form of the controller, namely one which applies state-estimate feedback of the form $u_k = K_c \hat{x}_{k|k}$ (rather than $u_k = K_c \hat{x}_{k|k-1}$). However in this case the direct feed-through term is restricted to be $D_K = K_c K_f$ (Alazard *et al.*, 2004), which does not allow arbitrary stabilising controllers to be put into this form. It is therefore necessary to use the same approach as in the continuous-time case.

6. EXAMPLE

We consider a linearised model for the lateral dynamics of the Lockheed C-5A transport aircraft in a particular flight condition, with 2 inputs, 4 states and 4 outputs. The inputs are the aileron deflection and the rudder angle, respectively, and the states are the lateral velocity, the roll rate, the yaw rate, and the roll angle. All the states are measured, so the outputs are the same as the states. The model is taken from (Cook, 1997, Example 5.6).

A simple stability augmentation controller provides PI action from the roll angle to the ailerons:

$$K_{14}(s) = -0.28 - \frac{0.02}{s} \quad (26)$$

and proportional action only from the yaw rate to the rudder:

$$K_{23}(s) = 1.4 \quad (27)$$

All other elements of the controller transfer function matrix are 0. This controller is used in a positive feedback loop, since the gains in the transfer function from rudder to yaw rate are negative, because of the sign conventions that are used. This controller gives an asymptotically stable closed loop. It is a typical example of a simple but highly structured controller which has been arrived at using application-specific expertise.

Constructing the Cross Standard Form for this plant-controller pair — after transformation to the equivalent strictly proper pair — gives, as one possibility (which depends on the choice of solution T to (6)):

$$-K_c = C_1 = \begin{bmatrix} 3.5088 \times 10^{-3} & -0.32201 & -1.0886 & -0.36136 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (28)$$

The fastest pole of the open-loop aircraft is -1.106 , and the fastest pole of the closed loop is -1.04 . A sampling rate of 2 Hz ($= 12.6$ rad/sec) should therefore be adequate for a discrete-time controller, so we choose $T_s = 0.5$ sec. This gives

$$Q_M = \begin{bmatrix} 0 & -0.0005 & -0.0020 & -0.0006 \\ -0.0005 & 0.0471 & 0.1855 & 0.0542 \\ -0.0020 & 0.1855 & 0.7355 & 0.2136 \\ -0.0006 & 0.0542 & 0.2136 & 0.0623 \end{bmatrix} \quad (29)$$

$$S_M = \begin{bmatrix} 0.0016 & 0.0002 \\ -0.1468 & -0.0213 \\ -0.5781 & -0.0884 \\ -0.1688 & -0.0247 \end{bmatrix} \quad (30)$$

$$R_M = \begin{bmatrix} 0.4754 & 0.0665 \\ 0.0665 & 0.5136 \end{bmatrix} \quad (31)$$

Note that Q_M is close to being a rank-2 matrix. Solving (24) gives

$$P_M = \begin{bmatrix} 7 \times 10^{-7} & -0.0002 & -0.0007 & -0.0002 \\ -0.0002 & 0.0086 & 0.3124 & 0.1101 \\ -0.0007 & 0.3124 & 1.1735 & 0.4029 \\ -0.0002 & 0.1101 & 0.4029 & 0.1415 \end{bmatrix} \quad (32)$$

7. CONCLUSION

A method has been demonstrated for finding the cost function and state estimator to be used in MPC, in order to obtain the same controller as

a given low-order controller when constraints are inactive, and hence when the MPC controller is operating within its baseline linear regime. This is a very useful starting point for the development of an MPC controller when a successful existing controller is known, but it is desired to add the constraint-handling capabilities of MPC, and perhaps other functionality such as fault-tolerance.

This paper is essentially an exploitation of the Cross Standard Form developed by (Delmond *et al.*, 2006) and in related works, and does not introduce any new theory. The exploitation is for similar purposes as those proposed by (Delmond *et al.*, 2006); their main motivation was improving the robustness of existing controllers. It is only our application to MPC that is new here.

It remains to explore the degrees of freedom offered by the choice of T when solving (6). Although each solution leads to the same controller in the linear case, it can be expected that different cost functions will give different control actions when constraints become active. Alternative discretisation strategies when given an initial continuous-time controller also remain to be explored. Generalisation to high-order controllers, with $n_K > n$, should be straightforward, since the Cross Standard Form for that case is already available from (Alazard and Apkarian, 1999).

REFERENCES

- Alazard, D. and P. Apkarian (1999). Exact observer-based structures for arbitrary compensators. *International Journal of Robust and Non-linear Control*.
- Alazard, D., C. Cumer and F. Delmond (2006). Improving flight control laws for load alleviation.
- Alazard, D., O. Voinot and P. Apkarian (2004). A new approach to multi-objective control design from the viewpoint of the inverse optimal control problem. In: *Proc. IFAC Symp. System Structure and Control SSSC'04*. Oaxaca, Mexico. pp. 476–481.
- Anderson, B.D.O. and J.B. Moore (1990). *Optimal Control, Linear Quadratic Methods*. Prentice Hall.
- Chmielewski, D. and V. Manousiouthakis (1996). On constrained infinite-time linear quadratic optimal control. *Systems and Control Letters* **29**, 121–129.
- Cook, M.V. (1997). *Flight Dynamics Principles*. Arnold. London.
- Delmond, F., D. Alazard and C. Cumer (2006). Cross standard form: a solution to improve a given controller with H_2 or H_∞ specifications. *International Journal of Control* **79**(4), 279–287.
- Franklin, G.F., J.D. Powell and M.L. Workman (1990). *Digital Control of Dynamic Systems*. 2nd ed.. Addison-Wesley. Reading, MA.
- Kalman, R.E. (1964). When is a linear system optimal?. *Trans. ASME, J. Basic Engineering* **86**, 56–60.
- Kreindler, E. and A. Jameson (1972). Optimality of linear control systems. *IEEE Transactions on Automatic Control* **AC-17**, 349–351.
- Kwakernaak, H. and R. Sivan (1972). *Linear Optimal Control Systems*. Wiley. New York.
- Laub, A.J. (1979). A Schur method for solving algebraic Riccati equations. *IEEE Transactions on Automatic Control* **AC-24**, 913–921.
- Maciejowski, J.M. (2002). *Predictive Control with Constraints*. Prentice-Hall. Harlow UK.
- McFarlane, D.C. and K. Glover (1990). A loop-shaping design procedure using H_∞ synthesis. *IEEE Transactions on Automatic Control* **37**(6), 759–769.
- Molinari, B.P. (1973). The stable regulator problem and its inverse. *IEEE Transactions on Automatic Control* **AC-18**(5), 454–459.
- Rao, C.V., S.J. Wright and J.B. Rawlings (1998). Application of interior point methods to model predictive control. *Journal of Optimization Theory and Applications* **99**(3), 723–757.
- Rowe, C.A. and J.M. Maciejowski (2000a). Robust finite horizon model predictive control without terminal constraints. In: *Proc. IEEE CDC Conf.*. Sydney.
- Rowe, C.A. and J.M. Maciejowski (2000b). Tuning MPC using H-infinity loop shaping. In: *Proc. American Control Conference*. Chicago.
- Sznaier, M. and M.J. Damborg (1987). Suboptimal control of linear systems with state and control inequality constraints. In: *Proc. IEEE CDC Conf.*. Los Angeles. pp. 761–762.
- van Dooren, P.M. (1981). A generalized eigenvalue approach for solving Riccati equations. *SIAM J. Scientific and Statistical Computing* **2**, 121–135.
- Voinot, O., D. Alazard, P. Apkarian, S. Maufrey and B. Clément (2003). Launcher attitude control: discrete-time robust design and gain scheduling. *Control Engineering Practice* **11**, 1243–1252.
- Zhou, K., J.C. Doyle and K. Glover (1996). *Robust and Optimal Control*. Prentice-Hall.