

Initial Tuning of Predictive Controllers by Reverse Engineering

E. N. Hartley and J. M. Maciejowski

Abstract—This paper demonstrates a method for finding the cost function and state observer to be used in model predictive control (MPC) so that when constraints are inactive a pre-existing low order controller is reproduced. The MPC controller thereby inherits its desirable properties. This can be used as a baseline for further tuning. The available degrees of design freedom are explored, in order to facilitate, as appropriate, exploitation of constraint-handling, offset-free and redundancy management capabilities of MPC.

I. INTRODUCTION

Model predictive control (MPC) has some attractive properties with respect to other control technologies. Using constrained optimisation, constraints can be explicitly handled [1], allowing the domain of attraction of equilibrium points to be extended (yielding a possible expansion of plant capacity), or the use of cheaper, more tightly constrained actuators. Furthermore, with careful formulation, redundancy that might be present in the plant can be exploited, either to increase working capacity or alternatively to provide some degree of robustness to failures [2], [3].

Due to modern optimisation algorithms [4] and techniques such as explicit MPC [5], combined with ever improving computers, the relatively high computational complexity associated with MPC is becoming a less significant barrier to its adoption. Nevertheless, many applications where MPC could be a practical option already have an existing controller that exhibits desirable performance properties. A replacement MPC controller would have to be tuned to perform at least as well. Therefore, a practical starting point for controller re-design would be an MPC formulation that, in the absence of constraints, is equivalent to the original controller.

The work of [6], further developed in [7], derives an analytical method for obtaining an observer-based realisation of an arbitrary linear time-invariant, stabilising output feedback controller. The *Cross Standard Form* (CSF), introduced in [8], is a generalised plant model constructed with weightings such that the optimum H_2 and H_∞ controllers are both equal to the observer-based realisations of a pre-specified output feedback controller K_0 of order greater than or equal to the order of the plant with the goal of improving robustness through consideration of multiple objectives. A discrete-time variant is defined in [9], whilst in [10] the CSF is generalised to accommodate the case where the baseline controller is of lower order than the controlled plant.

It is proposed in [11] that such a realisation be used as the starting point for a predictive controller which, by

This work is supported by the EPSRC, the European Space Agency, and EADS Astrium.

The authors are with the Department of Engineering, University of Cambridge CB2 1PZ, United Kingdom {enh20, jmm}@eng.cam.ac.uk.

construction, will exhibit the same closed loop behaviour as the original feedback controller in the absence of active constraints.

This paper builds upon the work of [11]. However, it differs from previous work by addressing the effects that the choices of the non-unique realisation of the original controller have on a constrained predictive controller. The issues addressed include the decision between reverse engineering in continuous time, or discretising before reverse engineering; the allocation of the closed loop poles between the observer error dynamics and the pure state feedback dynamics; and the extra degrees of design freedom when the original controller is of lower order than the plant and constraints are enforced.

II. OBSERVER BASED CONTROLLER REALISATIONS

We use the results of [10] for low order controllers as the basis for our MPC controller design. However, it is convenient to retain the LQG formulation of [7] for generating an MPC controller.

Let us consider a plant model and existing conventional controller with discrete-time state-space representations

$$G(z) = \left[\begin{array}{c|c} A & B \\ \hline C & 0 \end{array} \right], \quad K_0(z) = \left[\begin{array}{c|c} A_K & B_K \\ \hline C_K & D_K \end{array} \right] \quad (1)$$

respectively, where the plant model is of order n and the controller is of order $n_K \leq n$.

There are two main structures for discrete-time observers. These differ in how the residual is calculated. The *predictor* structure provides an *a priori* prediction of the plant state by using the output measurements from the previous time step. By using $\hat{x}(k|k-1)$, the estimate of x at time k given measurements from time $k-1$ as the boundary condition, the optimisation associated with the MPC controller can commence before time k , thus allowing longer for the optimisation to complete. Given an observer gain K_f , the predictor formulation is:

$$\hat{x}(k+1|k) = (A - K_f C)\hat{x}(k|k-1) + Bu(k) + K_f y(k). \quad (2)$$

The state estimate $\hat{x}(k|k-1)$ is then used for control purposes.

A controller with non-zero D_K cannot be directly reproduced using the estimate from a discrete-time predictor. Techniques to avoid this limitation are described later on. However, let us assume that some transformation on (or modification to) the system has already been performed to ensure that the controller is strictly proper, yielding:

$$\tilde{G}(z) = \left[\begin{array}{c|c} \tilde{A} & \tilde{B} \\ \hline \tilde{C} & 0 \end{array} \right], \quad \tilde{K}_0(z) = \left[\begin{array}{c|c} \tilde{A}_K & \tilde{B}_K \\ \hline \tilde{C}_K & 0 \end{array} \right]. \quad (3)$$

By considering a transformation T such that the controller state $x_K(k) = T\hat{x}(k|k-1)$ and solving for the (positive feedback) gains in an observer-based realisation such that the closed loop system remains the same (see [6], [7] for derivation), it is possible to obtain controller and observer gains, respectively, of

$$K_c = \tilde{C}_K T \quad (4a)$$

$$K_f = T^\dagger \tilde{B}_K \quad (4b)$$

where T^\dagger is a right inverse (recalling that $n_K \leq n$) of T .

The *filter* structure provides an *a posteriori* filtered estimate of the current plant state by using measurements from the current time step to influence the estimate of the current state. It is possible to reproduce a controller with non-zero D_K using a discrete-time filter, conditional on there being a zero at $z = 0$ on every channel — i.e. that $D_K = C_K A_K^{-1} B_K$ [6]. In state space form:

$$\hat{x}(k+1|k) = (A - AK_f C)\hat{x}(k|k-1) + Bu(k) + AK_f y(k) \quad (5a)$$

$$\hat{x}(k|k) = (I - K_f C)\hat{x}(k|k-1) + K_f y(k) \quad (5b)$$

Given a valid $K_0(z)$ with feedthrough terms and zeros at $z = 0$ on all paths, and by considering the same transformation as for the predictor formulation, the feedback and observer gains (assuming no delays in the plant) are:

$$K_c = C_K T + D_K C \quad (6a)$$

$$K_f = A^{-1} (T^\dagger B_K - B D_K). \quad (6b)$$

In both cases T is the solution to the generalised Riccati equation,

$$[-T \quad I] A_{cl} \begin{bmatrix} I \\ T \end{bmatrix} = 0 \quad (7)$$

where A_{cl} is the “ A ” matrix for the original closed loop system, e.g. if the filter formulation is to be used, or a loop shifting transformation has been used

$$A_{cl} = \begin{bmatrix} A + B D_K C & B C_K \\ B_K C & A_K \end{bmatrix}. \quad (8)$$

There are multiple solutions to (7), obtainable using invariant subspace methods [12]. The partition of the invariant subspaces of A_{cl} used to obtain a solution T determines which poles of A_{cl} appear in the state feedback dynamics and which appear in the observer error dynamics [6], [7].

The resulting observer based controller is a (non-minimal when $n_K < n$) realisation of the original controller. The closed loop system using the observer based realisation of the controller will contain $n - n_K$ poles that did not exist in the original closed loop system. These correspond to observer error dynamics in the nullspace of T , and thus are “invisible” through K_c , although can affect the closed loop system if K_c were changed. They are freely assignable by the system designer through the choice of T^\dagger used to calculate K_f . Defining T^+ as the Moore-Penrose pseudo-inverse and T^\perp as the orthogonal complement to T , for any compatibly sized matrix X :

$$T^\dagger = T^+ + T^\perp X. \quad (9)$$

By separating the observer modes into those in the row-space of T and those in the null-space of T by using the similarity transform

$$N = \begin{bmatrix} T \\ T^\perp \end{bmatrix}, \quad N^{-1} = [T^+ \quad T^\perp] \quad (10)$$

on the observer matrix $A_{pre} = \tilde{A} - K_f \tilde{C}$ (for the predictor formulation) or $A_{fil} = A - AK_f C$ (for the filter formulation) and substituting from (7), the “free” poles can then be assigned, as shown in [10], by calculating X in (9) as a predictor-form observer gain for the system:

$$\left[\begin{array}{c|c} T^\perp \tilde{A} T^\perp & T^\perp \tilde{B} \\ \hline \tilde{B}_K \tilde{C} T^\perp & 0 \end{array} \right] \quad (11)$$

or

$$\left[\begin{array}{c|c} T^\perp (A + B D_K C) T^\perp & T^\perp B \\ \hline B_K C T^\perp & 0 \end{array} \right] \quad (12)$$

for the predictor and filter structures respectively.

III. MPC IMPLEMENTATION

The cost function for the MPC controller must be constructed so that the optimum solution is equivalent to the estimated state feedback $K_c \hat{x}$ obtained for the discrete-time observer based controller. A zero-value infinite horizon cost function can trivially be constructed to ensure that $u(k) = K_c \hat{x}(k)$ is the optimal solution by using a stage cost

$$\ell(k) = \begin{bmatrix} x(k) \\ u(k) \end{bmatrix}^T \begin{bmatrix} K_c^T R K_c & -K_c^T R \\ -R K_c & R \end{bmatrix} \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \quad (13)$$

where $R > 0$ is a weighting matrix [13].

A standard MPC implementation performs an optimisation over a finite, but rolling horizon. A finitely parameterised infinite horizon cost function can be obtained by using the candidate cost function over a finite horizon of length N and using the solution to the discrete algebraic Riccati equation P as a terminal cost weighting [14], [15]. The cost function is then:

$$\left(\sum_{k=0}^{N-1} \ell(k) \right) + x^T(N) P x(N). \quad (14)$$

Because this cost function is designed to have an optimum cost of zero, P will be a zero matrix of appropriate size. Therefore no terminal cost need be added to the finite horizon optimisation.

Polytopic or ellipsoidal constraints [16] can subsequently be added to the MPC controller as desired. In the presence of constraints, the stabilising properties of the original controller might not be inherited, therefore, the usual technique of applying a terminal constraint as in [17] should be applied if theoretical stability guarantees are required.

IV. DESIGN DECISIONS AND PROCEDURES

The route from the original plant and controller pair to an effective MPC controller involves a number of design decisions that significantly affect the performance of the closed loop system and how effectively constraints are enforced. If poor choices are made, then it is very easy to

obtain a structure that is little more useful than an elaborate realisation of the original controller. In this section, these decisions and their consequences are outlined.

A. Discretisation

An existing controller might be specified in continuous time, however most practical implementations of MPC operate in discrete-time with sampled data. The procedure of casting a feedback controller into an observer-based realisation can work for both continuous time and discrete-time systems. So far, we have assumed everything is already discretised. However, when the plant or the original controller is specified in continuous time, the question arises as whether to perform the reverse engineering procedure in continuous-time or in discrete-time.

In the continuous-time case, a continuous-time observer is obtained and the estimated state sampled. If the gain used to form stage cost (13) is the same as the continuous time state feedback gain, the closed loop system behaviour might be very different to that when using the original controller. Better matching can be achieved by finding an “equivalent” discrete-time cost weighting matrix [18] that minimises an integral cost function whilst being formulated as a discrete time problem as done in [11]. This however, changes the effective gain of the unconstrained controller such that its rows are no longer in the row space of T . As a consequence, the $n - n_K$ poles corresponding to error dynamics in the nullspace of T will affect even the unconstrained closed loop system. This added complication constitutes a strong argument for directly obtaining the discrete-time observer-based realisation.

If the discrete-time observer-based realisation is used, a zero-order hold method best models how a simple MPC controller will drive the real plant. For the controller, it might be preferable to use a first-order hold or a Tustin transformation, particularly at low sampling frequencies. These can introduce non-zero D_K terms even if none existed in the continuous time controller. This process should, therefore, be performed before any transformations to ensure strict properness.

B. Ensuring a Strictly Proper Controller

If a predictor structure is used for the observer, the controller to be reproduced must be strictly proper—the D_K matrix must be zero. We outline two methods for working around this restriction for the case where D_K is non-zero.

1) *Loop Shifting*: The controller feedthrough term can be applied directly to the plant [19], leading to the following modified plant and controller:

$$\tilde{G}(z) = \left[\begin{array}{c|c} A + BD_KC & B \\ \hline C & 0 \end{array} \right] \quad (15a)$$

$$\tilde{K}_0(z) = \left[\begin{array}{c|c} A_K & B_K \\ \hline C_K & 0 \end{array} \right]. \quad (15b)$$

The closed loop system is not modified. However, the direct feedthrough component incorporated into the plant model uses the measured output, whilst the MPC prediction uses the observer output (the output values cannot be extrapolated

over the prediction horizon without the estimates of unmeasured states). Input constraints might therefore be violated, or, control can be overly conservative when the observer error $y(k) - C\hat{x}(k|k-1)$ is large in magnitude.

2) *Unit Delay or Low Pass Filter*: Adding a unit delay or a low pass filter in series with the original controller prior to obtaining the observer-based realisation will ensure a strictly proper $\tilde{K}_0(z)$. By avoiding a direct feedthrough, the MPC controller directly manipulates the plant input $u(k)$ rather than an estimate of the input. This allows hard input constraints to be imposed. The conventional controller $K_0(z)$ must be sufficiently robust to tolerate the adverse effects on stability margins that the phase shift due to this modification might have.

C. Ensuring Correct Zeros in Controller

If a filter structure is used for the observer, the controller model must have a zero at $z = 0$ on every channel for correct replication. If $K_0(z)$ does not include these zeros, success might be achieved by introducing a (stable) dipole of the form,

$$\frac{Wz}{Wz - 1} \quad (16)$$

where W is a “large” number, into the open-loop controller model. This introduces the required zeros whilst at the same time has minimal effect on open loop gains and phase shifts of the unconstrained controller. As long as the new poles introduced into the closed loop system near the origin are placed in the observer error dynamics, acceptable performance and robustness to disturbances and parasitic delays can be achieved.

D. Fixed Poles (from A_{cl})

The allocation of the closed loop poles between the observer error and the (unconstrained) state feedback dynamics is an important design decision when implementing an MPC controller in this manner. When discretisation occurs before obtaining the observer-based realisation, all arrangements should give identical closed loop performance. Each combination is merely a different non-minimal realisation of the original discretised controller. However, there can be a marked difference in the observer error dynamics. As previously stated, in the presence of constraints, this error can manifest itself as a violation of constraints, or overly conservative control (depending on the sign of the error). These types of error are particularly marked when using a formulation where $D_k y$ is directly fed back to the input of the plant, bypassing the MPC controller, whilst the MPC controller has to enforce constraints using an estimate, $C\hat{x}$ from the observer.

For meaningful predictions of the state trajectory to be made, the quality of the observations must be high. Intuition would therefore suggest, subject to keeping conjugate pairs together, and not attempting to shift uncontrollable modes between the feedback dynamics and the observer dynamics [7], that the fastest of the available closed loop poles be allocated to the observer error dynamics. However, a modicum

of common sense must be applied, especially if poles close to the origin have been introduced into the controller to ensure that it be strictly proper — if these are driven into the left half of the unit disc in the closed loop system, the observer will become susceptible to high frequency noise. Additionally, if a dipole has been introduced into the controller, poles introduced into the closed loop system near the origin must appear in the observer error dynamics otherwise the closed loop system will become extremely sensitive to the delay inevitably introduced by the online calculations which must occur after each sampling instance.

E. Free Poles (from choice of X)

The poles introduced as a consequence of the observer being of higher order than the original controller affect the MPC controller performance, despite their associated modes being “invisible” through the initial calculated K_c . The solution to the conventional, linearly constrained MPC problem is piecewise affine with respect to the current measured state [5]. When constraints become active the gain component of this function will change, and the observer error modes which were previously invisible in the closed loop system will have effects. Thus the system designer must choose appropriate positions for these newly introduced closed loop poles.

Also, if a continuous time realisation is used, and an “equivalent cost” formulation obtained, the unconstrained feedback gain will not be equal to the continuous time feedback gain, thus error dynamics in the nullspace of T might affect the plant input.

Rather than direct pole placement, one might use prior knowledge about noise covariances to calculate a discrete-time Kalman filter for the system (11). The filter gain can then be used as X in (9). This allows a more intuitive design procedure if the measurement noise and input disturbance covariances can be estimated.

VI. CONTROLLERS WITH INTEGRAL ACTION

A controller might exhibit integral action as a means of ensuring offset-free control despite persistent disturbances. Naïve reproduction of such a controller through the reverse engineering procedure would reproduce the input/output characteristics, however the quality of the observer estimates would be severely compromised in the presence of a persistent disturbance which would manifest itself as a bias on each state estimate. This is particularly problematic for a constrained predictive controller, as poor predictions made from biased state estimates may lead to overly conservative control action, or control action that leads to inevitable infeasibility.

To remedy this, the plant model can be augmented with a persistent input disturbance (although other models are possible [20], [21]). By considering an augmented state vector $[x^T \ d^T]^T$ with plant dynamics of the form

$$\bar{G}(z) = \left[\begin{array}{c|c|c} \tilde{A} & \tilde{B} & \tilde{B} \\ \hline 0 & I & 0 \\ \hline \tilde{C} & 0 & 0 \end{array} \right] \quad (17)$$

(or equivalent if using a filter observer structure) and considering the original controller \tilde{K}_0 as an observer on the transformed state $x_K = T [\hat{x}^T \ \hat{d}^T]^T$, the reverse-engineering process can be directly applied, reproducing the original dynamics perfectly, but providing more accurate observations of \hat{x} under persistent disturbances, and an estimate of these disturbances, \hat{d} . Further tuning is then possible by replacing the persistent disturbance model with a more general model [20], [21] if required.

VI. ALTERNATIVE COST FUNCTION FORMULATIONS

A. Degrees of Freedom in R and Choice of Norms

The above MPC formulation minimises a weighted quadratic cost function of the input and the state. When constraints are active preventing this cost from reaching zero, there is a degree of design freedom in the choice of R , allowing a weighting to be placed on each input channel.

When $\dim(u) \geq 2$, the choice of R affects the constrained solution by prioritising which directions of u should be matched to $K_c x$ most closely when a perfect match is impossible without violating constraints. If $R = R(k)$ is allowed to vary throughout the prediction horizon, then weightings can be chosen to prioritise matches at different points in the horizon. It would also be valid [22] to minimise:

$$\sum_{k=0}^{N-1} \|\hat{R}(u(k) - K_c x(k))\|_p, \quad p \in \{1, \infty\}. \quad (18)$$

As long as no constraints are active, zero cost will still be achieved when $u(k) = K_c x(k)$. However, when constraints are active, minimisation of the 1-norm minimises the absolute sum of errors whilst minimisation of the ∞ -norm minimises the worst case error.

B. Trajectory Matching

It is questionable whether there is any meaningful answer to “which input channel do I want to match to the original controller output most closely?”. The design goal is to reproduce the time-domain response of the original controlled system as closely as possible in the absence of constraints, and reproduction of the control action is merely a means to an end. Instead, some measure of the error between the constrained trajectory and the prediction of the nominal trajectory under the unconstrained state feedback gain K_c can be penalised. By decoupling the desired trajectory from a prescribed control allocation, redundancy of actuators can also be exploited in a natural manner. An appropriate cost function could take the form

$$\left\| \begin{bmatrix} \hat{Q} \left(x(1) - \tilde{A}_{fb}^1 x(0) \right) \\ \hat{Q} \left(x(2) - \tilde{A}_{fb}^2 x(0) \right) \\ \vdots \\ \hat{Q} \left(x(N) - \tilde{A}_{fb}^N x(0) \right) \end{bmatrix} \right\|_{\infty} + \left\| \begin{bmatrix} \hat{R}u(0) \\ \hat{R}u(1) \\ \vdots \\ \hat{R}u(N-1) \end{bmatrix} \right\|_1 \quad (19)$$

where $\tilde{A}_{fb} = \tilde{A} + \tilde{B}K_c$ and with weightings \hat{Q} and \hat{R} chosen to prioritise minimising trajectory error over minimising

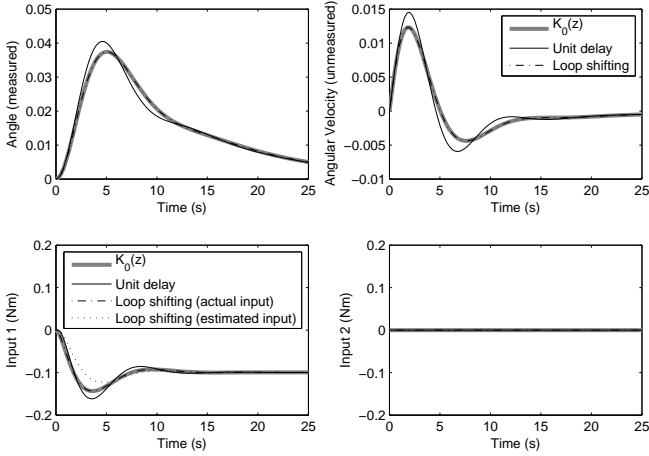


Fig. 1. Simulation results for reverse-engineered single axis attitude control system subject to a step disturbance of 0.1 Nm using a predictor-form observer

control inputs. \hat{Q} might also be chosen to prioritise the minimisation of errors in certain modes, for example, it might be preferable to prioritise minimisation of any unstable modes. The actuators can also be prioritised by the choice of \hat{R} . Redundancy can then be exploited automatically in normal operation, and under actuator failure all that is needed is an external fault detection, isolation and recovery (FDIR) algorithm to trigger a change of constraints to make a “best effort” at maintaining nominal performance.

VII. EXAMPLE

We demonstrate the reverse engineering technique on the discrete-time controller with integral action from [23, Example 9.4.1] for a single axis attitude control system with angle measurement only. The state-space realisation of the conventional controller with a sampling time of 0.25 s is

$$K_0(z)|_{T_s=0.25} = \left[\begin{array}{cc|c} 1.412 & -0.8235 & 32 \\ 0.5 & 0 & 0 \\ \hline -13.01 & 26.14 & 871 \end{array} \right] \quad (20)$$

whilst the plant is modelled as a double integrator with a persistent input disturbance. We introduce a redundant torque pair that is left idle by the original controller. The controller has poles at 1 and 0.41, zeros at 0.98 and 0.91, and the feedthrough component is large.

Figures 1 and 2 show the system response and observer error dynamics, respectively, to a step disturbance of 0.1 Nm simulated using the MPT Toolbox [24] and control weightings obtained by reverse engineering the original controller first by adding a unit delay to ensure strict properness (solid line), then by using loop shifting (dash-dot line). Additionally, when using loop shifting, the plant input estimated from the observer values (and thus the value on which constraints would be enforced) is plotted (dotted line) alongside the actual input to the plant (dash-dot line).

Adding a full unit delay reduces the damping of the closed loop system as can be seen in Figure 1. On the other hand, when using loop shifting, the error transient on the attitude

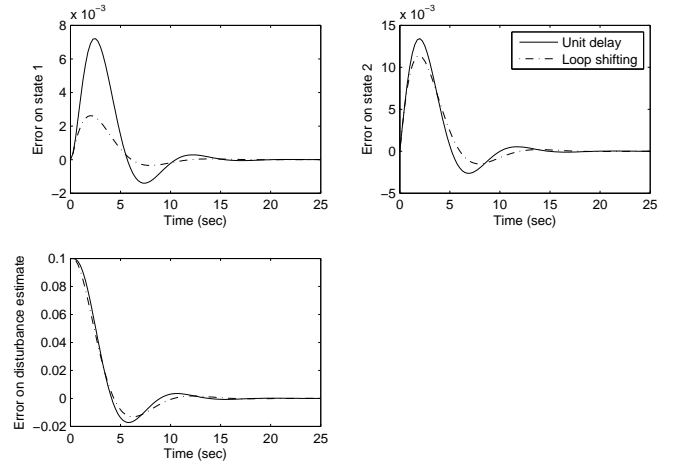


Fig. 2. Observer error transients for reverse-engineered single axis attitude control system subject to a step disturbance of 0.1 Nm using a predictor-form observer

angle estimate, shown as the error on state 1 in Figure 2, is large enough such that when amplified by D_K the error between estimated input and actual input is of an order of magnitude similar to that of the actual input. This means that constraints will be difficult to enforce correctly.

Therefore the filter observer form must be used and a dipole introduced (with subsequent care needed to ensure putting a pole near $z \approx 0$ in the observer). Figure 3 shows the response of the attitude angle to a step disturbance of 0.1 Nm for:

- unconstrained reproduction of the baseline controller in series with a dipole and a parasitic delay of 0.1 s to represent the MPC calculation time (dotted line);
- torque constraints of 0.125 Nm, only one thruster pair, with parasitic delay (dash-dot line);
- torque constraints of 0.125 Nm per thruster pair, with extra thruster pair allowed, but still penalising error from original single torque pair feedback, with parasitic delay (thin dashed line); and
- torque constraints of 0.125 Nm with trajectory matching as in (19) with $\hat{Q} = \text{diag}[1000, 1000, 1000]$ and $\hat{R} = \text{diag}[1, 10]$ and parasitic delay (solid line).

The time domain response with the dipole introduced and a delay of 0.1 s introduced to simulate the MPC processing time is virtually indistinguishable from the original closed loop system. Input constraints can be imposed. If there is redundancy and a saturation is incurred, even whilst penalising error from the original state feedback law, the second torque pair is used to reduce the cost (thin dashed line). Attempting to match a state trajectory rather than an input trajectory allows the second torque pair to be used in order to match the output response of the original closed loop system (solid thin line).

VIII. CONCLUSIONS

This paper has outlined some of the design decisions that must be made when using the observer based realisations

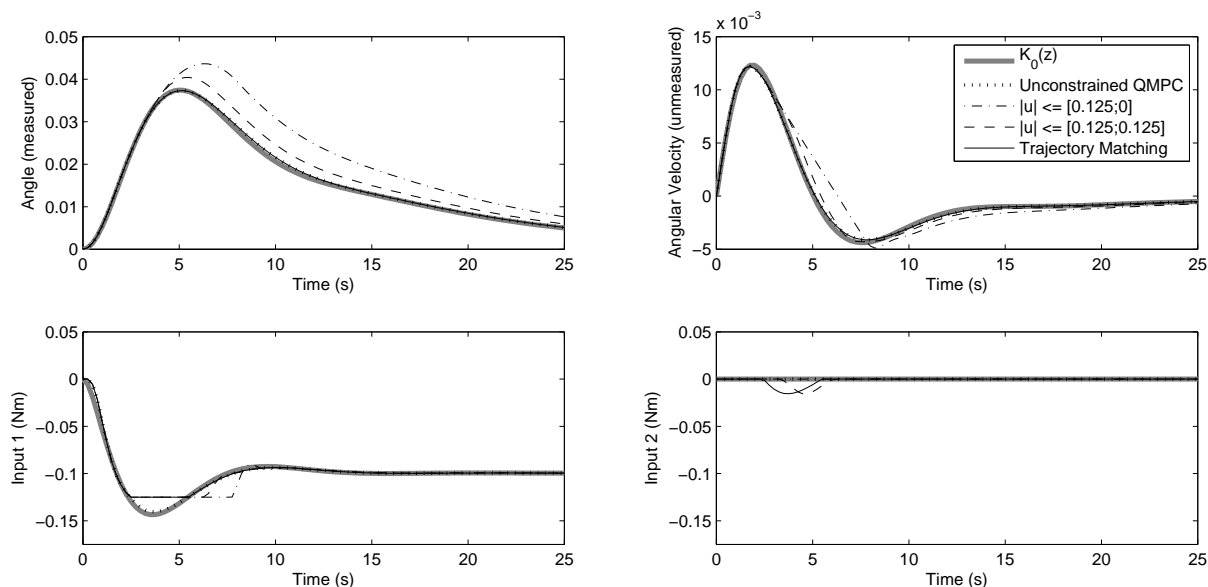


Fig. 3. Simulation results for reverse-engineered single axis attitude control system subject to a step disturbance of 0.1 Nm using a filter-form observer

of [6], [7], [10] to form the basis of a constrained MPC controller. For most predictable behaviour it is strongly recommended to discretise the plant model and original controller and obtain a complete discrete-time observer based realisation to work from. To enforce hard input constraints, either an observer structure with feedthrough must be used, or the controller $K_0(z)$ should be transformed to be strictly proper by adding a unit delay, or a low pass filter in series subject to the original controller being sufficiently robust to tolerate this. To ensure good predictions, the fastest closed loop poles must correspond to the modes of the observer error dynamics. Additionally, by aiming to match the state trajectory of the original controller rather than the input actions, we can exploit actuator redundancy.

Further work is needed to investigate the effectiveness of this procedure for obtaining an initial MPC controller design for large MIMO systems. Additionally, further work is required to investigate approximations of arbitrary controllers without using cross terms between input and state in the cost function.

REFERENCES

- [1] J. M. Maciejowski, *Predictive Control with Constraints*. Pearson Education, 2002. [Online]. Available: http://books.google.co.uk/books?id=HV_Y58c7KiWC
- [2] —, “Reconfigurable control using constrained optimization,” in *Proceedings of the European Control Conference*, Brussels, July 1997. [Online]. Available: <http://www-control.eng.cam.ac.uk/jmm/papers/ecc97.pdf>
- [3] —, “The implicit daisy-chaining property of constrained predictive control,” *Applied Mathematics and Computer Science*, vol. 8, no. 4, pp. 101–117, 1998. [Online]. Available: <http://www-control.eng.cam.ac.uk/Homepage/papers/cued.control.91.pdf>
- [4] S. J. Wright, “Applying new optimization algorithms to model predictive control,” *AIChE Symposium Series*, vol. 93, no. 316, pp. 147–155, 1997. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.55.5317>
- [5] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, “The explicit linear quadratic regulator for constrained systems,” *Automatica*, vol. 38, pp. 3–20, 2002.
- [6] D. J. Bender and R. A. Fowell, “Computing the estimator–controller form of a compensator,” *International Journal of Control*, vol. 41, no. 6, pp. 1565–1575, 1985.
- [7] D. Alazard and P. Apkarian, “Exact observer-based structures for arbitrary compensators,” *International Journal of Robust and Nonlinear Control*, vol. 9, no. 2, pp. 101–118, February 1999.
- [8] D. Alazard, “Cross standard form for generalized inverse problem: application to lateral flight control of a highly flexible aircraft,” in *Proceedings of International Conference on Nonlinear Problems in Aviation and Aerospace*, Melbourne (Florida), May 2002. [Online]. Available: <http://oatao.univ-toulouse.fr/2273/>
- [9] O. Voinot, D. Alazard, P. Apkarian, S. Mauffrey, and B. Clement, “Launcher attitude control: Discrete-time robust design and gain-scheduling,” *Control Engineering Practice*, vol. 11, no. 11, pp. 1243–1252, 2003.
- [10] F. Delmond, D. Alazard, and C. Cumer, “Cross standard form: a solution to improve a given controller with H2 or H-infinity specifications,” *International Journal of Control*, vol. 79, no. 4, pp. 279–287, April 2006.
- [11] J. M. Maciejowski, “Reverse engineering existing controllers for MPC design,” in *IFAC Symposium on System Structure and Control*, Iguassu Falls, Brazil, October 17–19 2007. [Online]. Available: <http://www.ifac-papersonline.net/Detailled/39313.html>
- [12] A. J. Laub, “A Schur method for solving algebraic Riccati equations,” *IEEE Transactions on Automatic Control*, vol. AC-24, no. 6, pp. 913–921, 1979. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1102178
- [13] E. Kreindler and A. Jameson, “Optimality of linear control systems,” *IEEE Transactions on Automatic Control*, vol. AC-17, no. 3, pp. 349–351, 1972. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1099985
- [14] J. B. Rawlings and K. R. Muske, “The stability of constrained receding horizon control,” *IEEE Transactions on Automatic Control*, vol. 38, no. 10, pp. 1512–1516, 1993.
- [15] D. Chmielewski and V. Manousiouthakis, “On constrained infinite-time linear quadratic optimal control,” *Systems & Control Letters*, vol. 29, no. 3, pp. 121–129, November 1996.
- [16] M. V. Kothare, V. Balakrishnan, and M. Morari, “Robust constrained model predictive control using linear matrix inequalities,” *Automatica*, vol. 32, no. 10, pp. 1361–1379, 1996.
- [17] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, “Constrained model predictive control: Stability and optimality,” *Automatica*, vol. 36, no. 6, pp. 789–814, June 2000.
- [18] C. F. Van Loan, “Computing integrals involving the matrix exponential,” *IEEE Transactions on Automatic Control*, vol. 23,

- no. 3, pp. 395–404, 1978. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1101743
- [19] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*. Prentice Hall, 1996.
- [20] K. R. Muske and T. A. Badgwell, “Disturbance modeling for offset-free linear model predictive control,” *Journal of Process Control*, vol. 12, no. 5, pp. 617–632, 2002.
- [21] G. Pannocchia, “Robust model predictive control with guaranteed setpoint tracking,” *Journal of Process Control*, vol. 14, no. 8, pp. 927–937, Dec. 2004.
- [22] C. V. Rao and J. B. Rawlings, “Linear programming and model predictive control,” *Journal of Process Control*, vol. 10, no. 2–3, pp. 283–289, April 2000.
- [23] M. J. Sidi, *Spacecraft dynamics and control: A practical engineering approach*. Cambridge University Press, 1997. [Online]. Available: <http://books.google.co.uk/books?id=xQpZJMtDehQC>
- [24] M. Kvasnica, P. Grieder, and M. Baotić, “Multi-Parametric Toolbox (MPT),” 2004. [Online]. Available: <http://control.ee.ethz.ch/~mpt/>