Reverse Engineered MPC for Tracking with Systems That Become Uncertain

Edward N. Hartley and Jan M. Maciejowski

***Abstract*— A constrained model predictive control technique for tracking is proposed for systems whose models become uncertain (for example after a sensor failure). A linear time invariant robust controller with integral action is used as a baseline and "reverse engineered" into the form of a reduced order observer, steady state target calculator and control gain, based on a nominal model, augmented with integrating disturbance states. Constraints are enforced by optimising over perturbations to the nominal control action. Clean transition between a nominal, high performance mode of operation when parameters are known, to a safe and recursively feasible robust mode when parameters are unknown can be facilitated by using the same steady state target in both cases.**

I. Introduction

Model predictive control (MPC) is a popular [1] technique for control of constrained multivariable systems. MPC controllers can be designed to be stabilising and recursively feasible [2] for systems with additive (e.g. [3]) and/or parametric (e.g. [4], [5]) uncertainty. For tracking of non-zero setpoints, there has been substantial focus on systems with additive uncertainty [6]–[9]. One approach is to regulate the error dynamics between the plant state and a target equilibrium state/input pair. For systems with parametric uncertainty, the equilibrium state and inputs may be unknown *a priori*. In [10] multiple disturbance observers are implemented for each vertex of a polytopic uncertainty set, augmented with integrating disturbances and a single steady-state target estimate is obtained by adjusting the disturbance estimation. The min-max robust feedback MPC approach of [11], is then applied to regulate to this target. In [12], a dead-beat disturbance observer is designed using a nominal model, and combined with a target calculator and LQR feedback gain based on the same model. Offline, an optimisation is performed over the space of nominal models to find a controller that simultaneously stabilises all realisations whilst minimising a quadratic cost. Following [5], the online MPC minimises deviations from the nominal control law whilst maintaining feasibility by means of polytopic robust controlled invariant set constraint.

An alternative approach uses the *linear velocity form*, which considers an augmented model in terms of the input and state *increments*. In [13] a robust tracking MPC is proposed, based on the linear velocity form with ellipsoidal input constraints, building upon the LMI-based infinite-horizon robust feedback MPC method of [4]. Ellipsoidal constraint-admissible invariant sets are computed, tightened based on bounds on the set of possible steady-state equilibria. [14] proposes a tube-MPC scheme based on the linear velocity form with polytopic constraints, and shows that the effect of parametric uncertainties can be modelled as an additive disturbance if they are "small".

The present work proposes a method complementary to [12]. Similarities are that the constraints are polytopic, and that the implementation relies only on quadratic programming. There is no differencing of states, nor an exponentially growing tree of predictions over a long horizon, nor online solution of LMIs. The key *dissimilarity* is that the proposed method does not rely upon the existence of a single model for which a *nominal* LQR controller and dead-beat disturbance observer simultaneously stabilise all possible realisations of the uncertain plant. Instead, a linear (baseline) controller that already exists, or can be designed using standard methods to achieve adequate small-signal responses, is *reverse engineered* into a reduced-order observer-based form by application of the theory of [15], [16], and then further transformed into an observer/target-calculator-based form by development of the methods proposed in [17]. When there is no uncertainty, the MPC can optimise regulation to the target state and input from this construction. When there is uncertainty (e.g. due to a fault), the online MPC maintains recursive feasibility through perturbations to the reference and control signals of the reverse engineered baseline controller. The use of reverse-engineering provides the system designer the flexibility to design the closed-loop small-signal behaviour of the uncertain system using a range of methods and then cast the unconstrained design into a form where constraints can be handled in a computationally efficient manner.

This paragraph codifies the contents of the paper: Section II states the standing assumptions; Section III presents the transformation of a pre-standing satisfactory linear controller into a reduced-order-observer-target-calculator form; Section IV describes the integration into an MPC-based framework; Section V presents a motivating example in the form of control of the short-period longitudinal dynamics of an aircraft after loss of airspeed measurements.

II. Assumptions and notation

Assumption 1 (Plant model): Letting $x(k) \in \mathbb{R}^{n_x}$, $u(k) \in \mathbb{R}^{n_u}$, $y_r(k) \in \mathbb{R}^{n_r}$, $y_m(k) \in \mathbb{R}^{n_m}$ denote the plant state, input, controlled output, and measured output at time k, the plant is described by a parameterised state-space model

$$\begin{aligned} x(k+1) &= A(\theta)x(k) + B(\theta)u(k) + d(\theta) \\ y_r(k) &= C_r x(k), \quad y_m(k) = Cx(k) \end{aligned} \qquad (1)$$

The research leading to these results has received funding from the European Union Seventh Framework Programme FP7/2007-2013 under grant agreement number 314 544, project "RECONFIGURE".

The authors are with the University of Cambridge (email: {enh20,jmm}@eng.cam.ac.uk).

where the matrices $A(\theta) \in \mathbb{R}^{n_x \times n_x}$, $B(\theta) \in \mathbb{R}^{n_x \times n_u}$, $d(\theta) \in \mathbb{R}^{n_x \times 1}$, $C_r \in \mathbb{R}^{n_r \times n_x}$, $C \in \mathbb{R}^{n_m \times n_x}$. The variable $\theta \in \Theta \subset \mathbb{R}^M$ may not be measurable and parameterises

$$A(\theta) = \sum_{i=0}^{M-1} \theta_i A^{\{i\}},\ B(\theta) = \sum_{i=0}^{M-1} \theta_i B^{\{i\}},\ d(\theta) = \sum_{i=0}^{M-1} \theta_i d^{\{i\}} \tag{2}$$

$$\Theta \triangleq \{\theta : \theta_i \geq 0 \quad i = 0, \ldots, M-1, \quad \textstyle\sum_{i=0}^{M-1} \theta_i = 1\}.$$

The number of controlled outputs is less than or equal to the number of manipulated inputs: $n_y \leq n_u$. $(A(\theta), B(\theta))$ is controllable and $(C(\theta), A(\theta))$ is observable $\forall \theta \in \Theta$.

Remark 1: A non-zero $d(\theta)$ implies that $x(k) = 0$ is not an unforced equilibrium. The input required to maintain equilibrium varies with the parameter θ.

Assumption 2: An LTI controller with state $x_K \in \mathbb{R}^{n_K}$ with integral action that provides satisfactory robust stabilisation, tracking and transient response exists or can be designed:

$$x_K(k+1) = x_K(k) - C_r x(k) + r(k) \tag{3a}$$
$$u(k) = K_1 x(k) + K_2 x_K(k). \tag{3b}$$

If $r(k)$ and θ are constant, then $\lim_{k \to \infty} y_r(k) \to r(k)$.

III. Reverse Engineering

A. Reduced observer-based form

This subsection re-states some key results from [16] for observer-based realisations of linear regulators in a discrete-time setting, and explains their relevance in the context of the remainder of the paper.

Definition 1 (Candidate nominal system and controller): Consider a candidate nominal linear plant model

$$\overline{x}(k+1) = \overline{A}\overline{x}(k) + \overline{B}\overline{u}(k),\ \overline{y}(k) = \overline{C}\overline{x}(k) \tag{4}$$

and a stabilising linear controller

$$\overline{x}_K(k+1) = A_K \overline{x}_K(k) + B_K \overline{y}(k) \tag{5a}$$
$$\overline{u}(k) = C_K \overline{x}_K(k) + D_K \overline{y}(k). \tag{5b}$$

Remark 2: The LTI model (4) used for observer design can be constructed to accommodate an uncertain integrating term by adding an additional n_d states in the same way as for estimation of uncertain persistent disturbances e.g.

$$\overline{A} = \begin{bmatrix} \hat{A} & I_{n_d} \\ 0 & I_{n_d} \end{bmatrix}, \quad \overline{B} = \begin{bmatrix} \hat{B} & 0 \end{bmatrix}, \quad \overline{C} = \begin{bmatrix} \hat{C} & 0 \end{bmatrix} \tag{6}$$

where $\hat{A}$, $\hat{B}$ and $\hat{C}$ are nominal estimates of the matrices $A(\theta)$, $B(\theta)$ and $C(\theta)$ (e.g. an average value.)

Remark 3: When $r(k) = 0$, (5) is equal to (3) when $A_K = I$, $B_K = -C_r$, $C_K = K_2$, $D_K = K_1$.

Lemma 1: The system:

$$\overline{z}(k+1) = F\overline{z}(k) + G\overline{y}(k) + T\overline{B}\overline{u}(k) \tag{7}$$

is an observer of the variable $\xi = T\overline{x}$ of the system (4) if

$$T\overline{A} - FT = G\overline{C} \text{ and } F \text{ stable.} \tag{8}$$

Lemma 2: The controller state $\overline{x}_K$ is an observer of $\xi = T\overline{x}$ if T satisfies the non-symmetric Riccati equation

$$T\overline{A} - (A_K - T\overline{B}C_K)T - (B_K - T\overline{B}D_K)\overline{C} = 0. \tag{9}$$

Proof: To make (7) equivalent to (5), let

$$F = A_K - T\overline{B}C_K, \text{ and } G = B_K - T\overline{B}D_K. \tag{10}$$

Then substitute into (8) and apply Lemma 1 [16]. ■

Remark 4: If a solution to (9) exists, it may not be unique.

Remark 5: When $n_K = \dim(x_K) < \dim(\overline{x}) = n_{\overline{x}} = n_x + n_d$, the observer-based realisation can include a full order observer with $n_{\overline{x}} - n_K$ freely positioned eigenvalues in its error dynamics [17], [18], or instead, a reduced order observer can be designed. The former was used to transform linear tracking controllers for LTI systems into a target-calculator based MPC controller form in [17], whilst the latter has also been exploited for MPC design for linear systems in [19]. Here, the main purpose of the "observer" is to estimate the agglomeration of the uncertain affine term and parameter uncertainty. To minimise the number of states in the closed-loop system, the reduced-order observer is therefore favoured. (Mismatch between the design model and the real plant precludes invoking the *separation principle* anyway.)

Lemma 3: Controller (5) can be re-written as:

$$\overline{z}(k+1) = F\overline{z}(k) + G\overline{y}(k) + T\overline{B}\overline{u}(k) \tag{11a}$$
$$\hat{\overline{x}}(k) = H_2\overline{z}(k) + H_1\overline{y}(k) \tag{11b}$$
$$\overline{u}(k) = K_c\hat{\overline{x}}(k) + D_Q\left(\overline{y}(k) - \overline{C}\hat{\overline{x}}(k)\right) \tag{11c}$$

where $K_c = C_K T + D_K \overline{C}$, $\quad \begin{bmatrix} H_1 & H_2 \end{bmatrix} \begin{bmatrix} \overline{C}^T & T^T \end{bmatrix}^T = I,$ (12)

F and G satisfy (10), T satisfies (9), $\hat{\overline{x}}$ denotes an estimate of the state of (4) and D_Q is a static Youla parameter, satisfying:

$$C_K = (K_c - D_Q C)H_2,\ D_K = (K_c - D_Q C)H_1 + D_Q. \tag{13}$$

Remark 6: If $n_K + n_x \leq n_{\overline{x}}$ then the solution for H_1 and H_2 is non-unique and can be exploited to affect the behaviour of the internal signals to the controller (Sec. III-D).

B. Handling the reference input

Controller (11) assumes a zero-valued reference input. In [17] a controller whose input was $\overline{y}(k) - r(k)$ was reverse engineered by exploiting linear superposition, and implementing an observer on $\overline{x}(k)$ excited by $\overline{y}(k)$ and a prefilter with the same dynamics, but excited by $r(k)$. Here, the reference enters into the controller (3) differently from the measured output (state), and the tracked output is only a subset of the measured output.

Theorem 1: A controller that gives equivalent input/output behaviour to (3) can be parameterised as:

$$\overline{z}(k+1) = F\overline{z}(k) + G\overline{y}(k) + TB\overline{u}(k) \tag{14a}$$
$$\overline{z}_2(k+1) = F\overline{z}_2(k) + G_r r(k) \tag{14b}$$

$$\begin{aligned}\overline{u}(k) &= K_c\left(H_2\overline{z}(k) + H_1\overline{y}(k) + H_3\overline{z}_2(k)\right) \\ &\quad + D_Q(\overline{y} - \overline{C}(H_1\overline{y}(k) + H_2\overline{z}(k) - H_3\overline{z}_2(k)))\end{aligned} \tag{14c}$$

where F and G satisfy (10) and T satisfies (9). For the baseline controller parameters given in Remark 3, $G_r = I$ and $H_3 = H_2$.

Proof: Substituting (14c) into (14a) gives

$$\begin{aligned}\overline{z}(k+1) &= (F + T\overline{B}(K_cH_2 - D_Q\overline{C}H_2))\overline{z}(k)\\ &+ (G + T\overline{B}(K_cH_1 + D_Q(I - \overline{C}H_1)))\overline{y}(k)\\ &+ T\overline{B}(K_cH_3 + D_QCH_3)\overline{z}_2(k).\end{aligned} \quad (15)$$

Let $\zeta(k) = \overline{z}(k)+\overline{z}_2(k)$ and substitute $\overline{z}(k) = \zeta(k)-\overline{z}_2(k)$, let $H_2 = H_3$ and $G_r = I$ then:

$$\begin{aligned}\zeta(k+1) &= (F + T\overline{B}(K_cH_2 - D_Q\overline{C}H_2))\zeta(k)\\ &+ (G + T\overline{B}(K_cH_1 + D_Q(I - \overline{C}H_1)))\overline{y}(k) + r(k),\end{aligned} \quad (16)$$

$\overline{u}(k) = (K_c-D_QC)H_2\zeta(k)+(K_cH_1+D_Q(I-CH_1))y(k)$. Comparing (13) and (10) term-by-term,

$$F + T\overline{B}(K_cH_2 - D_Q\overline{C}H_2) = A_K \quad (17a)$$

$$G + T\overline{B}(K_cH_1 + D_Q(I - \overline{C}H_1)) = B_K \quad (17b)$$

Therefore, by (17) and (13), it is shown that (16) (which gives the same control action as (14c) with $G_r = I$ and $H_3 = H_2$) is identical to: $\zeta(k+1) = A_K\zeta(k) + B_K\overline{y}(k) + r(k)$, $\overline{u}(k) = C_K\zeta(k)+D_K\overline{y}(k)$. With values from Remark 3 this is the same as (3). ∎

C. Target calculator design

When the model structure (6) is applied, the state estimate can be partitioned into an "estimated controlled state" and "estimated uncontrolled disturbance", and the state feedback gain K_c can be partitioned into corresponding components

$$\hat{\overline{x}}(k) = \begin{bmatrix}\hat{x}(k)^T & \hat{d}(k)^T\end{bmatrix}^T, \quad K_c = \begin{bmatrix}K_{cx} & K_{cd}\end{bmatrix}. \quad (18)$$

As in [17] a steady-state target calculator is to be implemented, such that the input computation (14c) is

$$\overline{u}(k) = K_{cx}\left(\hat{x}(k) - x_s(k)\right) + u_s(k) \quad (19)$$

$$\text{where} \quad (\hat{A} - I)x_s(k) + \hat{B}u_s(k) = -\hat{d}(k) \quad (20a)$$

$$C_r x_s(k) = r_p(k). \quad (20b)$$

Proposition 1: A pair $(x_s(k), u_s(k))$ is an equilibrium pair satisfying $Cx_s(k) = CH_2\overline{z}_2$, and the control action (19) is equivalent to (14c) if it also holds that

$$-K_{cx}x_s(k)+u_s(k) = K_{cd}\hat{d}(k)+K_cH_2\overline{z}_2+d_q+d_P, \quad (21)$$

where $d_q = -D_Q(\hat{x}(k) - \overline{y}(k))$, $d_P = -D_QCH_2\overline{z}_2$, and $r_p(k)$ is a filtered reference signal derived from the state $z_2(k)$ and the static Youla parameters from the observer and prefilter d_q and d_P respectively. Let the superscript $\bullet^+$ denote the Moore-Penrose Pseudoinverse of $\bullet$.

Definition 2: Define:

$$M = \begin{bmatrix}M_{11} & M_{12}\\ M_{21} & M_{22}\end{bmatrix} = \begin{bmatrix}(\hat{A} - I) & \hat{B}\\ C_r & 0\end{bmatrix}^+ \quad (22)$$

Proposition 2: If there can be found an $x_{\text{ref}}(k)$ such that

$$Vx_{\text{ref}}(k) = K_cH_2\overline{z}_2(k) + d_q(k) + d_P(k) \quad (23)$$

where $V = (M_{22} - K_{cx}M_{12})C_r$, then (assuming existence of a solution), control action (19) is equivalent to (14c) with $(x_s(k), u_s(k))$

$$\begin{bmatrix}(\hat{A} - I) & \hat{B}\\ C_r & 0\\ -K_{cx} & I\end{bmatrix}\begin{bmatrix}x_s(k)\\ u_s(k)\end{bmatrix} = \begin{bmatrix}-I & 0\\ 0 & C_r\\ K_{cd} & V\end{bmatrix}\begin{bmatrix}\hat{d}(k)\\ x_{\text{ref}}(k)\end{bmatrix}. \quad (24)$$

Proof: Equation (23) enforces that the "contribution" to the final control action from the reference $C_rx_{\text{ref}}(k)$ is equal to the combined contribution of $\overline{z}_2(k)$, $d_q(k)$ and $d_P(k)$ to that of (14c). The matrix V is also used in the third row of (24) to enforce (21). ∎

Proposition 3: From equivalence to the baseline controller, the solution to (24) exists by construction when the baseline controller is designed to track the controlled output without offset. Similarly, by the same observation, $\lim_{k\to\infty} C_rx_{\text{ref}}(k) \to r(k)$.

Corollary 1: If $n_r = n_u$ and a unique solution exists for (22) then the target calculator (24) can be implemented in the standard way simply solving

$$\begin{bmatrix}(\hat{A} - I) & \hat{B}\\ C_r & 0\end{bmatrix}\begin{bmatrix}x_s(k)\\ u_s(k)\end{bmatrix} = \begin{bmatrix}-I & 0\\ 0 & I\end{bmatrix}\begin{bmatrix}\hat{d}(k)\\ r_p(k)\end{bmatrix} \quad (25)$$

where $r_p(k) = C_rx_{\text{ref}}(k)$ with x_{ref} satisfying Proposition 2.

D. Enforcing steady-state consistency

Even when a solution to (24) exists, and is used to compute control action (19) from application of (24), (23), (14a), (14b), (18), (11), it is not automatically guaranteed that $\lim_{k\to\infty}(Cx_s(k), u_s(k)) = \lim_{k\to\infty}(Cx(k), u(k))$ for plant (1) for all constant $\theta \in \Theta$, even though the net control action is identical to that from (3) by construction and $\lim_{k\to\infty} C_rx_s(k) = \lim_{k\to\infty} C_rx(k)$ for a constant reference $r(k)$.

This section contains the key enabling contribution of the present paper: a method to use the degrees of freedom available in (12) when $n_K + n_x < n_{\overline{x}}$ to produce a design for which the measured outputs and target calculator are consistent in steady state *despite the inevitable mismatch* between the plant and the candidate design model due to possible uncertain knowledge of the former.

For brevity, the following derivation assumes that the number of tracked outputs is equal to the number of inputs and the number of controller states, and the static Youla parameter $D_Q = 0$ (satisfied by the example in Section V).

Definition 3: When $[\overline{C}^T, T^T]^T$ is not of full row rank, the solution to (12) is not unique. Letting the superscript $\bullet^\perp$ denote an orthogonal basis for the nullspace of $\bullet$ (of rank $n_\perp$), define

$$Z = \begin{bmatrix}Z_{11} & Z_{12}\\ Z_{21} & Z_{22}\end{bmatrix} \triangleq \begin{bmatrix}C\\ T\end{bmatrix}^+, \quad W = \begin{bmatrix}W_1\\ W_2\end{bmatrix} \triangleq \begin{bmatrix}C\\ T\end{bmatrix}^\perp \quad (26)$$

such that for some $X \triangleq \begin{bmatrix}X_1 \in \mathbb{R}^{n_\perp \times n_m} & X_2 \in \mathbb{R}^{n_\perp \times n_K}\end{bmatrix}$

$$H_2 = \begin{bmatrix}Z_{12} + W_1X_2\\ Z_{22} + W_2X_2\end{bmatrix}, \quad H_1 = \begin{bmatrix}Z_{11} + W_1X_1\\ Z_{21} + W_2X_1\end{bmatrix}. \quad (27)$$

Definition 4: The unconstrained target calculator can be expressed in the closed form:

$$\begin{bmatrix} x_s(k) \\ u_s(k) \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} \begin{bmatrix} r_p(k) \\ \hat{d}(k) \end{bmatrix} \tag{28}$$

The objective of the following is to obtain conditions on X_1 and X_2 such that $Cx_s(k) \to \overline{y}(k)$ at equilibrium.

Theorem 2: For the observer (14a) and target calculator (24), a constant reference $r(k) = r_\infty$, and steady state measurement/input signals $\overline{y}_\infty$ and $\overline{u}_\infty$, the target output corresponding to the the steady state equilibrium pair, and the target input will be equal to the true measured plant output and input in steady state if X_1 and X_2 satisfy:

$$\begin{aligned} I_{n_m+n_u} &= \begin{bmatrix} CY_{11}C_r & CY_{12} \\ Y_{21}C_r & Y_{22} \end{bmatrix} \times \\ &\left(\begin{bmatrix} Z_{12} & Z_{11} \\ Z_{22} & Z_{21} \end{bmatrix} + \begin{bmatrix} W_1 & W_1 \\ W_2 & W_2 \end{bmatrix} \begin{bmatrix} X_2 & 0 \\ 0 & X_1 \end{bmatrix} \right) \times \\ &\left(\begin{bmatrix} I_{n_K+n_m} & 0_{(n_K+n_m)\times n_u} \end{bmatrix} \begin{bmatrix} (I-F)^{-1}[G, T\overline{B}] \\ I_{n_m+n_u} \end{bmatrix} \right). \end{aligned} \tag{29}$$

Proof: For given steady values of $\overline{y}_\infty, \overline{u}_\infty$, the steady state value of the state and disturbance estimate using (14a) and (11b) is, by application of the Final Value Theorem and some algebraic rearrangement:

$$\begin{bmatrix} \hat{x}_\infty \\ \hat{d}_\infty \end{bmatrix} = \left(\left(\begin{bmatrix} Z_{12} & Z_{11} \\ Z_{22} & Z_{21} \end{bmatrix} + \begin{bmatrix} W_1 & W_1 \\ W_2 & W_2 \end{bmatrix} \begin{bmatrix} X_2 & 0 \\ 0 & X_1 \end{bmatrix} \right) \times \begin{bmatrix} I_{n_K+n_m} & 0_{n_K+n_m\times n_u} \end{bmatrix} \times \begin{bmatrix} (I-F)^{-1}[G, T\overline{B}] \\ I_{n_m+n_u} \end{bmatrix} \right) \begin{bmatrix} \overline{y}_\infty \\ \overline{u}_\infty \end{bmatrix} \tag{30}$$

In steady state, the controller will have, by design, driven the plant to a state where $C_r x_\infty = r_\infty$, and the corresponding output from the target calculator will be:

$$\begin{bmatrix} x_{s,\infty} \\ u_{s,\infty} \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} \begin{bmatrix} C_r & 0_{n_r\times n_d} \\ 0_{n_d\times n_x} & I_{n_d} \end{bmatrix} \begin{bmatrix} \hat{x}_\infty \\ \hat{d}_\infty \end{bmatrix}. \tag{31}$$

The required steady state behaviour is expressed as

$$\begin{bmatrix} C & 0_{n_r\times n_u} \\ 0_{n_u\times n_x} & I_{n_u} \end{bmatrix} \begin{bmatrix} x_{s,\infty} \\ u_{s,\infty} \end{bmatrix} = \begin{bmatrix} \overline{y}_\infty \\ \overline{u}_\infty \end{bmatrix}. \tag{32}$$

Condition (29) in the theorem is thus constructed by combining (32), (31) and (30). ■

Corollary 2: If $n_K = n_r$, then (29) can be re-written as $\mathbf{ABC} = \mathbf{F}$, where

$$\mathbf{A} = \begin{bmatrix} CY_{11}C_r & CY_{12} \\ Y_{21}C_r & Y_{22} \end{bmatrix}, \mathbf{B} = \begin{bmatrix} W_1 & W_1 \\ W_2 & W_2 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} X_2 & 0 \\ 0 & X_1 \end{bmatrix}$$

$$\mathbf{F} = \begin{bmatrix} (I-F)^{-1}G & (I-F)^{-1}T\overline{B} \\ I_{n_m} & 0 \end{bmatrix}^{-1} - \mathbf{A} \begin{bmatrix} Z_{12} & Z_{11} \\ Z_{22} & Z_{21} \end{bmatrix}.$$

Letting $\mathbf{E} \triangleq \left\{ \mathbf{E} : \mathbf{E} \begin{bmatrix} \text{vec}(X_2)^T & \text{vec}(X_1) \end{bmatrix}^T = \text{vec}(\mathbf{C}) \right\}$, condition (29) can be achieved if there exists a solution to

$$(I_{n_r+n_K} \otimes \mathbf{AB})\, \mathbf{E}[\text{vec}(X_2)^T, \text{vec}(X_1)^T]^T = \text{vec}(\mathbf{F}) \tag{33}$$

Remark 7: (33) may have redundant equalities (e.g. constraining a tracked output to be equal to the reference, or its equivalent measured output at equilibrium).

IV. CONSTRAINED MPC

A. Inner loop

Defining $\overline{K}_x = K_{cx}(H_{11} - Y_{12}H_{21})\hat{C} + Y_{22}H_{21}\hat{C}$, $\overline{K}_z = K_{cx}(H_{12} - Y_{12}H_{22}) + Y_{22}H_{22}$, $\overline{K}_r = (Y_{21} - K_{cx}Y_{11})$, the closed-loop system with the uncertain plant and the unconstrained reverse-engineered controller based on the nominal plant model can be re-written as two separate systems. The first is the prestabilised plant (now including the affine term),

$$\mathbf{x}(k+1) = \mathcal{A}(\theta)\mathbf{x}(k) + \mathcal{B}(\theta)\mathbf{u}(k) + \mathbf{d}(\theta), \quad y(k) = \mathcal{C}\mathbf{x}(k), \tag{34}$$

where $\mathbf{x}(k) = [x(k)^T, \overline{z}(k)^T]^T$, $\mathbf{u}(k) = [r_p(k)^T, v(k)^T]^T$, $v(k)$ is an additive input perturbation that an outer loop can manipulate, $\mathbf{d}(\theta) = [d(\theta)^T, 0^T]^T$,

$$\mathcal{A}(\theta) = \begin{bmatrix} A(\theta) + B(\theta)\overline{K}_x & B(\theta)\overline{K}_z \\ G\hat{C} + T\overline{B}\overline{K}_x & F + T\overline{B}\overline{K}_z \end{bmatrix}$$

$$\mathcal{B}(\theta) = \begin{bmatrix} B(\theta)\overline{K}_r & B(\theta) \\ T\overline{B}\overline{K}_r & T\overline{B} \end{bmatrix}, \quad \mathcal{C} = \begin{bmatrix} \hat{C} & 0 \end{bmatrix}.$$

The steady state target estimate can be seen as an additional output of this system:

$$\begin{bmatrix} x_s(k) \\ u_s(k) \end{bmatrix} = \begin{bmatrix} Y_{12}H_{21}\hat{C} & Y_{12}H_{22} \\ Y_{22}H_{21}\hat{C} & Y_{22}H_{22} \end{bmatrix} \begin{bmatrix} x(k) \\ \hat{z}(k) \end{bmatrix} + \begin{bmatrix} Y_{11} \\ Y_{21} \end{bmatrix} r_p(k). \tag{35}$$

The second system is the prefilter $\overline{z}_2(k+1) = F\overline{z}_2(k) + r(k)$,

$$r_p^*(k) = C_r V^+ K_c H_2 \overline{z}_2(k). \tag{36}$$

For the unconstrained baseline, $r_p(k) = r_p^*(k)$.

B. Constraints and MPC formulation

The purpose of applying MPC in this scenario is to enforce input and state constraints in a manner that will be recursively feasible (within the confines of the modelling assumptions) for the constraints $H_{xu} \begin{bmatrix} x(k)^T & u(k) \end{bmatrix}^T \le h_{xu}^T$. This can be re-written as (more usefully):

$$H_{xu} \begin{bmatrix} I & 0 & 0 & 0 \\ \overline{K}_x & \overline{K}_z & \overline{K}_r & I \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{u}(k) \end{bmatrix} \le h_{xu}. \tag{37}$$

Tracking MPC for LTI systems often uses a terminal constraint that is a maximal positively invariant set $\mathcal{O}_\infty$ for an autonomous system under a candidate controller, augmented with integrating states parameterising the reference [20]. With parametric uncertainty, the steady state setpoint is not known a priori, and the system (34) augmented with an integrating state representing $r_p(k)$ does not admit a unique transformation that partitions the augmented state space into stable and marginally stable modes [21], so obtaining a finitely determined robust polytopic $\mathcal{O}_\infty$ for the uncertain augmented system is difficult. The affine component of the model, and the desire to non-conservatively enforce asymmetric constraints also makes ellipsoids unattractive.

Instead, the method proposed in [12] (also advocated in [22]) is applied, and a polyhedral robust controlled invariant set $\mathcal{C}$ [23] is computed for (34) assuming $v(k) = 0$:

$$\mathcal{C} \triangleq \big\{(x(k), \overline{z}(k)) : \exists r_p \text{ satisfying (37) with } v(k) = 0, \\ \text{such that } (x(k+1), \overline{z}(k+1)) \in \mathcal{C}, \quad \forall \theta \in \Theta\big\}.$$

When the variable θ is unknown, at each time step the online MPC formulation can compute $v(k)$ and $r_p(k)$ as:

$$\min_{\mathbf{u}(k)} v(k)^T R_v v(k) + (r_p(k) - r_p^*(k))^T S (r_p(k) - r_p^*(k))$$

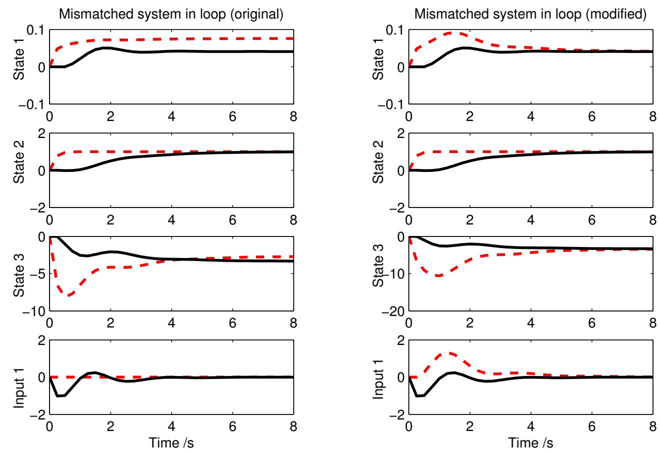
subject to (37) and

$$\mathcal{A}(\theta)\mathbf{x}(k) + \mathcal{B}(\theta)\mathbf{u}(k) + \mathbf{d}(\theta) \in \mathcal{C}, \forall \theta \in \Theta. \tag{38}$$

The target-calculator based formulation also means that when θ is known (e.g. absence of sensor failure), a "standard" MPC optimisation problem based on the same target can subsequently be performed over a finite horizon (c.f. [6], [7], [9]) as a heuristic to "improve" performance:

$$\min_{x_i, v_i} v_0^T R_v v_0 + R_r \|r_p(k) - r_p^*(k)\|_1 + \|x_i - x_s(k)\|_P^2 \\ + \sum_{i=0}^{N-1} \left(\|x_i - x_s(k)\|_Q^2 + \|u_i - u_s(k)\|_R^2\right)$$

subject to (38), $u_i = \overline{K}_x x_i + \overline{K}_z \overline{z}_i + \overline{K}_r r_i + v_i, \forall i \in \mathbb{Z}_0^{N-1}$, $\mathbf{x}_0 = \mathbf{x}(k)$, $\mathbf{x}_{i+1} = \mathcal{A}(\theta)\mathbf{x}_i + \mathcal{B}\mathbf{u}_i + \mathbf{d}(\theta), \forall i \in \mathbb{Z}_0^{N-1}$, $H_{xu}[x_i^T, u_i^T]^T \leq h_{xu}, \forall i \in \mathbb{Z}_0^{N-1}$, $H_{xu}[x_i^T, 0^T]^T \leq h_{xu}$, $i = N$. Q, P, R, R_r, R_v, S are appropriately sized weighting matrices. If θ becomes unknown (e.g. loss of sensor) (38) remains feasible and thus the "robust" MPC can be used, but lower performance is achieved.



(a) $X_1, X_2 = 0$ (b) Using (33)

Fig. 1. Effect of H_1, H_2 solution on target estimate (solid line is the measured state, dashed line is the steady state target estimate)

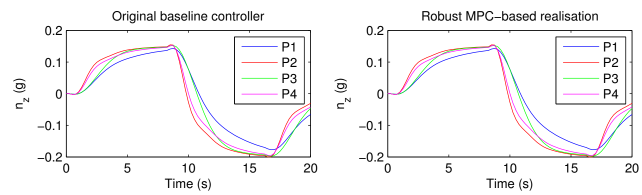


Fig. 2. Closed-loop behaviour matching for small deviations of n_z

V. Motivating Example

The developed MPC is applied to the control of the linearised short-period dynamics (e.g. [24]) of an aircraft [25] following a detected loss of airspeed measurement. It is assumed that it is possible to use other measurements to maintain groundspeed such that the airspeed is within a given range, and that the aircraft remains within a bounded altitude range. These dynamics are nonlinear, but can be approximated by local linear models (affine if the "input" and "state" are the true values rather than deviation from equilibrium), which vary as a function of the airspeed and altitude (which are not considered states in the short-period model). We consider a three-state discrete-time system sampled at $t_s = 0.25$ s, with states q (pitch rate), n_z (vertical acceleration in body frame), e (elevator position). The input u is the elevator increment, delayed by one time step. (It is more usual to use angle of attack α as a state variable, with n_z considered a measured output — the model used here is obtained by simple coordinate change.)

Linearisations of the nonlinear model are taken at straight and level equilibrium flight points, and the short period approximation is made in the usual way [24], transformed into the desired coordinate system and discretised. The flight points chosen, **P1**–**P4** are 5000 m altitude at 160 ms^{-1} and 260 ms^{-1} airspeed and 7500 m altitude at 180 ms^{-1} and 260 ms^{-1} airspeed. Since each of these equilibrium points requires different value of e, and the exact point is not known a priori, e retains its physical value and the non-equilibrium dynamics are encoded by the affine disturbance term in the uncertain model form (2). The control objective is to track references in n_z. To obtain a baseline controller, the system is augmented with an integrating state with dynamics $\zeta(k+1) = \zeta(k) - n_z(k) + r(k)$, and the unconstrained version of the robust MPC method of [4] is used to obtain a static feedback gain for the linear parts of the uncertain model giving satisfactory robust performance. The state $\zeta(k+1)$ is then detatched, and along with the computed gain rearranged to form a controller of form (3).

Figure 1 shows the unconstrained response to a unit step in reference for the second state, for plant **P4** using **P1** as the nominal model for design with $X_1 = 0$, $X_2 = 0$, and for X_1 and X_2 designed using (33). In the former case, the "target" is inconsistent, whilst in the latter case, the steady state target is consistent with the plant state by design.

Constraints are defined as $-2 \leq n_z \leq 1.5$, $-17 \leq e \leq 23$, $-37t_s \leq u \leq 37t_s$, $-2 \leq r_p \leq 1.5$. A controlled invariant set $\mathcal{C}$ for the uncertain plant is computed (c.f. [23]), but slightly tightening constraints to avoid lots of "almost redundant" constraints (c.f. [26]) before computing each projection, and the MPC (38) is implemented using YALMIP [27]. To demonstrate the equivalence of the unconstrained behaviour, Figure 2 shows the response to a multi-step trajectory for each of the flight points, starting from equilibrium states, chosen so that no constraints are active, firstly for the baseline controller of form (3), and then for the reverse engineered MPC (38). The responses are identical by design. Meanwhile Figure 3

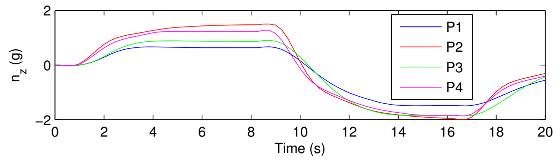


Fig. 3. Robust (conservative) enforcement of output constraints

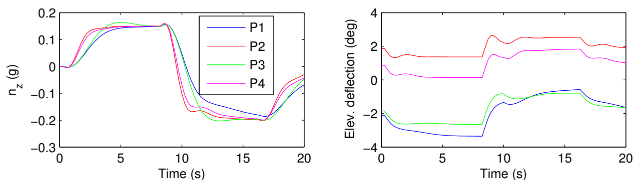


Fig. 4. Switchover from nominal MPC to robust MPC at $t = 10$ s

shows robust handling of nominal output constraints for a larger reference signal. The constraint (38) means that the constraint on n_z is treated more conservatively for the lower-speed flight points **P1** and **P3** than for **P2** and **P4** and instead of reaching the setpoint, the nearest safe value inside $\mathcal{C}$ is achieved. Figure 4 shows transition from a nominal "high performance" MPC to the robust MPC (both using the same steady state target from the reverse-engineered controller) at $t = 10$ s. The output trajectory is slower after the transition, but there is minimal jump in the commanded input.

VI. Conclusions

This paper has proposed an MPC method for constrained tracking control for a system with parametric uncertainty, which relies online only on constrained quadratic programming (for which efficient embedded solvers exist), based on reverse-engineering of an existing LTI controller into an observer-target calculator-gain form. This can give more design flexibility than prior approaches, and whilst the observer is still based on a nominal model, the unconstrained control law inherits the robust stabilising properties of the original controller. Algebraic conditions are given to ensure that the steady-state target computed within the reverse-engineered controller is consistent with the setpoint achieved by the plant, allowing the same steady state target to be used in a "nominal" setup based on an arbitrary cost function and full model information and a "robust" setup based on minimising deviation from the behaviour of the existing controller. The proposed method could be an enabling ingredient in a fault-tolerant system, for example facilitating a fallback in case of loss of scheduling parameter in a time-varying system through sensor failure.

References

[1] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Eng Pract*, vol. 11, no. 7, pp. 733–764, Jul. 2003.

[2] 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, Jun. 2000.

[3] D. Q. Mayne, M. M. Seron, and S. V. Rakovic, "Robust model predictive control of constrained linear systems with bounded disturbances," *Automatica*, vol. 41, no. 2, pp. 219–224, Feb. 2005.

[4] 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.

[5] B. Kouvaritakis, J. A. Rossiter, and J. Schuurmans, "Efficient robust predictive control," *IEEE Trans. Automat. Control*, vol. 45, no. 8, pp. 1545–1549, Aug. 2000.

[6] G. Pannocchia and J. B. Rawlings, "Disturbance models for offset-free model predictive control," *AIChE J.*, vol. 49, no. 2, pp. 426–437, Feb. 2003.

[7] K. R. Muske and T. A. Badgwell, "Disturbance modeling for offset-free linear model predictive control," *J. Process Control*, vol. 12, no. 5, pp. 617–632, 2002.

[8] I. Alvarado, D. Limón, T. Alamo, M. Fiacchini, and E. F. Camacho, "Robust tube based MPC for tracking of piece-wise constant references," in *Proc. IEEE Conf. Decision and Control*, New Orleans, LA, Dec. 12–14 2007, pp. 1820–1825.

[9] U. Maeder, F. Borrelli, and M. Morari, "Linear offset-free model predictive control," *Automatica*, vol. 45, no. 10, pp. 2214–2222, Oct. 2009.

[10] Y. J. Wang and J. B. Rawlings, "A new robust model predictive control method I: theory and computation," *J. Process Control*, vol. 14, no. 3, pp. 231–247, 4 2004.

[11] P. O. M. Scokaert and D. Q. Mayne, "Min-max feedback model predictive control for constrained linear systems," *IEEE Trans. Automat. Control*, vol. 43, no. 8, pp. 1136–1142, Aug. 1998.

[12] G. Pannocchia, "Robust model predictive control with guaranteed setpoint tracking," *J. Process Control*, vol. 14, no. 8, pp. 927–937, Dec. 2004.

[13] J. S. Lim, J.-S. Kim, and Y. I. Lee, "Robust tracking model predictive control for input-constrained uncertain linear time invariant systems," *Int. J. Control*, vol. 87, no. 1, pp. 120–130, 2014.

[14] G. Betti, M. Farina, and R. Scattolini, "A robust MPC algorithm for offset-free tracking of constant reference signals," *IEEE Trans. Automat. Control*, vol. 58, no. 9, pp. 2394–2400, 2013.

[15] D. J. Bender and R. A. Fowell, "Computing the estimator–controller form of a compensator," *Int. J. Control*, vol. 41, no. 6, pp. 1565–1575, 1985.

[16] D. Alazard and P. Apkarian, "Exact observer-based structures for arbitrary compensators," *Int. J. Robust Nonlinear Control*, vol. 9, no. 2, pp. 101–118, Feb. 1999.

[17] E. N. Hartley and J. M. Maciejowski, "Designing output-feedback predictive controllers by reverse engineering existing LTI controllers," *IEEE Trans. Automat. Control*, vol. 58, no. 11, pp. 2934–2939, Nov. 2013.

[18] F. Delmond, D. Alazard, and C. Cumer, "Cross standard form: a solution to improve a given controller with H2 or H-infinity specifications," *Int. J. Control*, vol. 79, no. 4, pp. 279–287, Apr. 2006.

[19] M. Foo and E. Weyer, "On reproducing existing controllers as model predictive controllers," in *Proc. Australian Control Conf. (AUCC)*, Melbourne, VIC, Nov. 10–11 2011, pp. 303–308.

[20] D. Limon, I. Alvarado, T. Alamo, and E. F. Camacho, "MPC for tracking piecewise constant references for constrained linear systems," *Automatica*, vol. 44, no. 9, pp. 2382–2387, 2008.

[21] E. G. Gilbert and K. T. Tan, "Linear systems with state and control constraints: the theory and application of maximal output admissible sets," *IEEE Trans. Automat. Control*, vol. 36, no. 9, pp. 1008–1020, Sept. 1991.

[22] R. Gondhalekar, J.-i. Imura, and K. Kashima, "Controlled invariant feasibility —a general approach to enforcing strong feasibility in MPC applied to move-blocking," *Automatica*, vol. 45, no. 12, pp. 2869–2875, 12 2009.

[23] F. Blanchini, "Ultimate boundedness control for uncertain discrete-time systems via set-induced Lyapunov functions," *IEEE Trans. Automat. Control*, vol. 39, no. 2, pp. 428–433, Feb. 1994.

[24] B. L. Stevens and F. L. Lewis, *Aircraft control and simulation*, 2nd ed. Hoboken, New Jersey: Wiley, 2003.

[25] C. van der Linden, H. Smaili, A. Marcos, G. Balas, D. Breeds, S. Runham, C. Edwards, H. Alwi, T. Lombaerts, J. Groeneweg, R. Verhoeven, and J. Breeman, "GARTEUR RECOVER benchmark," 2011. [Online]. Available: http://www.faulttolerantcontrol.nl

[26] B. Pluymers, "Robust model based predictive control - an invariant set approach," Ph.D. dissertation, Katholieke Universiteit Leuven, 2006.

[27] J. Löfberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004. [Online]. Available: http://control.ee.ethz.ch/~joloef/yalmip.php