Reconfigurable Predictive Control for Redundantly Actuated Systems with Parameterised Input Constraints

Edward N. Hartley[a,*], Jan M. Maciejowski[a]

[a] *University of Cambridge, Department of Engineering, Trumpington Street, Cambridge. CB2 1PZ. United Kingdom.*

Abstract

A method is proposed for on-line reconfiguration of the terminal constraint used to provide theoretical nominal stability guarantees in linear model predictive control (MPC). By parameterising the terminal constraint, its complete reconstruction is avoided when input constraints are modified to accommodate faults. To enlarge the region of feasibility of the terminal control law for a certain class of input faults with redundantly actuated plants, the linear terminal controller is defined in terms of virtual commands. A suitable terminal cost weighting for the reconfigurable MPC is obtained by means of an upper bound on the cost for all feasible realisations of the virtual commands from the terminal controller. Conditions are proposed that guarantee feasibility recovery for a defined subset of faults. The proposed method is demonstrated by means of a numerical example.

Keywords: model predictive control, reconfigurable control, fault-tolerant control

1. Introduction

It has long been advocated that model predictive control (MPC) has advantages in the field of reconfigurable and fault tolerant control [1–10]. In MPC, at each time step, the control input applied to the plant is obtained by solution of an optimisation problem [11, 12], which minimises a function of future inputs and states over a finite horizon, subject to physical and operational constraints. The first part of the computed input trajectory is applied to the plant, and at the next time step the process is repeated using new measurements. Since the optimisation problem can be modified given knowledge about a fault, physical and analytical redundancy within plant can be exploited to recover an acceptable level of performance.

For many applications, theoretical guarantees of recursive feasibility and stability can assist in acceptance and clearance procedures. One way to enforce recursive feasibility and stability by design is through employment of an appropriate terminal control law, terminal cost and terminal constraint [13]. A special case of this setup employs a terminal equality constraint. Here, the terminal cost is zero, and no terminal control law is used, but when the prediction horizon is short, this limits the region of feasibility of the controller, and leads to aggressive performance.

Whilst this methodology is sufficient (rather than necessary) for stability, its application is commonplace in contemporary literature. For example, in [4, 5], a reconfigurable linear MPC design is proposed for aircraft control. The predictions are pre-stabilised as in [14], and a terminal cost is used. It is indicated that the maximum admissible set (MAS) for the closed-loop system with just the pre-stabilising controller is large, but no explicit form for the constraint is presented. More recently, [7, 9] proposes a fault-tolerant MPC controller for flight control, with an explicit stabilising terminal constraint, corresponding to the MAS for a precomputed LTI controller around a steady-state set-point. The MAS is parameterised by augmenting the state vector with the target equilibrium state and input setpoints (as also in [15–17]). The authors of [7, 9] propose to pre-compute the MAS offline for each combination of input faults. The number of possible combinations grows combinatorially with the number of failures accounted for simultaneously, and these sets must be computed for every applicable trim condition, a large number of sets could be required. Moreover, if it is possible for an input to jam at any value and not just zero, then the issue of pre-computing terminal sets is even more problematic.

On-line re-calculation of an appropriate admissible set for a change in constraints, terminal control law or plant model would be computationally costly, and would substantially complicate an embedded implementation. Alternatively, post-fault, the terminal set could be replaced with an equality constraint, but then there is the risk of unnecessary infeasibility if the required (or even the closest [9, 15–17]) equilibrium point cannot be reached within the prediction horizon. Performance on channels unaffected by the fault would also suffer.

These issues motivate the contribution of the present paper. A certain class of input faults can be accommodated by an MPC controller by modelling them as a change

*Corresponding author
Email addresses: enh20@eng.cam.ac.uk (Edward N. Hartley), jmm@eng.cam.ac.uk (Jan M. Maciejowski)

in the input constraints. A change in input range (e.g. a stuck actuator) can be modelled by adjusting the input constraints. A constraint bounding the (weighted) difference between two inputs can be tightened to represent actuators locking together, whilst a constraint bounding the (weighted) sum of inputs could be tightened to represent an energy deficit. Since the MAS depends on the input constraints, it is proposed to extend the methods proposed in the aforementioned references to parameterise an inner approximation of the MAS for the terminal control law in terms of the input inequality constraints.

Whilst the finite-horizon MPC problem will be specified in terms of the *actual* plant inputs, to increase the admissible region for the terminal control law in the event of a faulty plant input, this is specified in terms of "virtual commands". A "virtual command" describes the effect of a non-zero input on the system rather than the specific realisation: in a redundantly actuated system (e.g. an aircraft with multiple ailerons and multiple elevators) the same "virtual command" can be delivered by different input realisations. The terminal control law is a theoretical construct and never actually applied to the plant, so the control allocation mapping the it to an input realisation is also never implemented online. Nevertheless, an upper bound on the worst-case cost incurred must be obtained to design a suitable terminal cost.

A summary of the paper is presented thus: Section 2 develops the constraint-parameterised MPC for linear time invariant (LTI) systems, with nominal stability guarantees assuming initial feasibility; Section 3 proposes conditions which guarantee that feasibility can be recovered during the transition from a nominal to plant to a class of faulty plant; Section 4 summarises the computation of setpoint targets; Section 5 presents a simple numerical example; and Section 6 concludes.

2. Reconfigurable MPC

Assumption 1. *The plant is described by the discrete-time linear time invariant state space model*

$$x(k+1) = Ax(k) + Bu(k), \quad y(k) = Cx(k) \tag{1}$$

where state $x(k) \in \mathbb{R}^{n_x}$*, input* $u(k) \in \mathbb{R}^{n_u}$*, controlled output* $y(k) \in \mathbb{R}^{n_y}$*,* $A \in \mathbb{R}^{n_x \times n_x}$*,* $B \in \mathbb{R}^{n_x \times n_u}$*,* $k \in \mathbb{Z}$*,* $n_u \geq n_x \in \mathbb{Z}$*, the pair* (A, B) *is stabilisable and all states are measurable.* $(A - I)$ *is invertible.*

Assumption 2. *Let* $\mathbb{X} \subseteq \mathbb{R}^{n_x}$*,* $\mathbb{U} \subseteq \mathbb{R}^{n_u}$*,* $\mathbb{T} \subseteq \mathbb{X}$ *be convex, closed polytopic sets, specified in the form:* $\mathbb{X} \triangleq \{x \colon G_x x \leq g_x\}$*,* $\mathbb{U} \triangleq \{u : G_u u \leq g_u\}$*,* $\mathbb{T} \triangleq \{x : G_t x \leq g_t\}$*.*

Assumption 3. *Let* $x_s \in \operatorname{int}(\mathbb{T})$ *and* $u_s \in \operatorname{int}(\mathbb{U})$ *be a state/input pair satisfying* $(A - I)x_s + Bu_s = 0$*.*

Let $x_i(k)$ and $u_i(k)$ denote the state and input predictions i time steps into the future from the current time step k, up to prediction horizon N. Define $\| \bullet \|_P^2 \triangleq \bullet^T P \bullet$, $\delta x = (x - x_s)$, $\delta u = (u - u_s)$, $\mathbf{u} = (u_0, \ldots, u_{N-1})$, $\mathbf{x} = (x_0, \ldots, x_N)$ and cost function

$$J(x, \mathbf{u}) = \|\delta x_N\|_P^2 + \textstyle\sum_{i=0}^{N-1} \left(\|\delta x_i\|_Q^2 + \|\delta u_i\|_R^2\right) \tag{2}$$

for matrices $Q \in \mathbb{R}^{n_x \times n_x} > 0$, $R \in \mathbb{R}^{n_u \times n_u} > 0$ and $P \in \mathbb{R}^{n_x \times n_x}$. Letting $\bullet^*$ indicate an optimal value, at every time step k, an MPC controller solves Problem 1 and applies $u(k) = u_0^*(x(k))$ to the plant:

Problem 1.

$$(\mathbf{u}^*(x), \mathbf{x}^*(x)) = \arg\min_{(\mathbf{u},\mathbf{x})} J(x, \mathbf{u}) \tag{3a}$$
$$s.t. \quad x_0 = x \tag{3b}$$
$$x_{i+1} = Ax_i + Bu_i, \; \forall i \in \{0, \ldots, N-1\} \tag{3c}$$
$$x_i \in \mathbb{X}, \quad \forall i \in \{0, \ldots, N-1\} \tag{3d}$$
$$u_i \in \mathbb{U}, \quad \forall i \in \{0, \ldots, N-1\} \tag{3e}$$
$$x_N \in \mathbb{T}. \tag{3f}$$

Let $\mathcal{S}_N(\mathbb{X}, \mathbb{U}, \mathbb{T})$ be the set of all $x(k)$ such that there exists $\mathbf{u}(x(k))$ satisfying constraints (3b) to (3f). A well-known set of sufficient additional conditions that guarantees nominal stability and recursive feasibility if $x(k) \in \mathcal{S}_N(\mathbb{X}, \mathbb{U}, \mathbb{T})$, requires the terminal cost weighting matrix P and the terminal constraint set $\mathbb{T}$ to have specific properties [13]. Sufficient conditions for recursive feasibility are presented in a linear setting in Assumptions 4 and 5.

Assumption 4. *An unconstrained state feedback controller* $u(k) = K_N x(k)$ *exists that renders the matrix* $(A + BK_N)$ *Schur (i.e.* $|\rho(A + BK_N)| < 1$*).*

Assumption 5. $\mathbb{T}$ *is positively invariant with respect to the closed-loop system comprised of (1) with the controller* $u(k) = K_N \delta x(k) + u_s$ *and the state and input constraints* $\mathbb{X}$ *and* $\mathbb{U}$*, i.e.* $x \in \mathbb{T} \implies x \in \mathbb{X}$*,* $K_N \delta x + u_s \in \mathbb{U}$ *and* $Ax + B(K_N \delta x + u_s) \in \mathbb{T}$*.*

Given recursive feasibility, the additional conditions sufficient for $J^*(x(k)) = J(x(k), \mathbf{u}^*(x(k)))$ to be a Control Lyapunov Function about the equilibrium (x_s, u_s) for the closed-loop system comprised of (1) and controller $u(k) = u_0^*(k)$ are summarised in Assumption 6.

Assumption 6. *The terminal cost* P *is chosen so that:* $-\|\delta x_N\|_P^2 + \|\delta x_N\|_Q^2 + \|K_N \delta x_N\|_R^2 + \|(A + BK_N)\delta x_N\|_P^2$ *for all* $x_N \in \mathbb{T}_g(g_u)$*. This holds if* $(A + BK_N)^T P (A + BK_N) - P + Q + K_N^T R K_N \leq 0$*.*

2.1. Implications of reconfiguration

Since the subsequent developments consider changing constraints, a simple parameterisation of the set $\mathbb{U}$ is introduced in terms of the vector g_u. Let $\mathbb{U} = \mathbb{U}_g(g_u) \triangleq \{u : G_u u \leq g_u\}$. Suppose that under normal operating conditions, $g_u = g_u^{(0)}$. Following a fault (assumed to be instantly detected), the value of g_u changes to $g_u^{(1)}$ to enable the MPC to redirect commands to the remaining degrees of freedom (possibly with reduced performance).

The realisation of $\mathbb{T}$ satisfying Assumption 5 is dependent on $\mathbb{U}$, and $\mathbb{U} = \mathbb{U}_g(g_u)$ is parameterised by g_u, so it follows that for a given (x_s, u_s), $\mathbb{T}$ should also a function of g_u: $\mathbb{T} = \mathbb{T}_g(g_u)$. A difficulty here is that the terminal control law $u = K_N \delta x + u_s$ may not be admissible with respect to the modified input constraint set $\mathbb{U}_g(g_u^{(1)})$ within a set $\mathbb{T}(g_u^{(0)})$ computed to satisfy Assumption 5 for $\mathbb{U} = \mathbb{U}_g(g_u^{(0)})$. Computing a set satisfying Assumption 5 can be a demanding process involving the solution of a large number of linear programs, and is therefore usually computed off-line. Since g_u is continuously valued, it is not possible to pre-compute a set of discrete $\mathbb{T}(g_u^{(i)})$ offline.

The second difficulty is that the set of x on which $u = K_N x$ is admissible with respect to $\mathbb{U}_g(g_u)$ can become small, and in some cases unnecessarily so, limiting the size of $\mathbb{T}_g(g_u)$, despite for some plants the same control "effect" $B(K_N \delta x + u_s)$ being achievable through a different value of u due to redundancy.

Assumption 7. *The matrix B can be naturally decomposed into two parts $B = \hat{B}M$, i.e.*

$$x(k+1) = Ax(k) + \hat{B}v(k) \qquad v(k) = Mu(k). \tag{4}$$

The columns of $\hat{B} \in \mathbb{R}^{n_v}$ are formed by orthogonal basis vectors defined by the column space of B, $n_v = \mathrm{Rank}(B)$ and the matrix $M \in \mathbb{R}^{n_v \times n_u}$, and the signal $v(k) = Mu(k)$ is termed a "virtual command".

This parameterisation of the state space model is particularly intuitive in cases where the virtual commands correspond to (for example) accelerations, or angular accelerations, and the matrix M defines the effects of the real system inputs in terms of these virtual commands. One approach is for the whole control system to be designed in terms of these virtual commands, and for a control allocation algorithm to then map these back to real inputs [18]. In the present work it is assumed that one wishes to design an MPC controller that considers all physical inputs, and thus the control and allocation tasks are combined.

2.2. Virtual terminal control law

To obtain a terminal control law that can be feasible in the presence of a number of constraint-restricting faults, we propose to specify it in terms of virtual commands.

Assumption 8 (Relaxes Assumption 5)**.** *$\mathbb{T}$ is positively invariant with respect to the closed-loop system (4) with the virtual controller $v(k) = M(K_N \delta x(k) + u_s)$, and the state constraints $\mathbb{X}$ and virtual control constraints $\mathbb{V}$, where $\mathbb{V} \triangleq \{v : \exists u \in \mathbb{U}, s.t\ v = Mu\}$. In other words $x \in \mathbb{T} \implies x \in \mathbb{X}$, $M(K_N \delta x + u_s) \in \mathbb{V}$ and $Ax + \hat{B}M(K_N \delta x + u_s) \in \mathbb{T}$.*

Lemma 1. *The set of all feasible virtual commands $\mathbb{V}_g(g_u) \triangleq \{v = Mu : G_u u \leq g_u\}$ is affine in g_u.*

Proof. By replacing the equality by back-to-back inequalities, the sets $\mathbb{U}_g(g_u)$ and $\mathbb{V}_g(g_u)$ can be related as:

$$\begin{bmatrix} G_u & 0 & -I \\ M & I & 0 \\ -M & -I & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ g_u \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{5}$$

By projection (e.g. Fourier-Motzkin elimination), a set $\hat{\mathbb{V}}$ can be computed, containing all v and g_u such that there exists a feasible corresponding $u \in \mathbb{U}_g(g_u)$:

$$\hat{\mathbb{V}} \triangleq \left\{ \begin{bmatrix} v^T & g_u^T \end{bmatrix}^T : \exists u \text{ s.t. } v = Mu,\ G_u u \leq g_u \right\}. \tag{6}$$

$\hat{\mathbb{V}}$ can be expressed in the form: $\begin{bmatrix} V_{gv} & V_{gg} \end{bmatrix} \begin{bmatrix} v^T & g_u^T \end{bmatrix}^T \leq v_g$. By substituting in a value of parameter g_u, the set of feasible virtual commands is:

$$\mathbb{V}_g(g_u) = \{v : V_{gv} v \leq v_g - V_{gg} g_u\}. \tag{7}$$

□

Assumption 9. *Let the elements of $v \triangleq \begin{bmatrix} v_{\{1\}}, \dots, v_{\{n_v\}} \end{bmatrix}^T$. Deviations $\pm\tilde{\epsilon}_i$, $i \in \{1, \dots, n_v\}$ in virtual commands from the equilibrium must always be feasible, where*

$$\tilde{\epsilon}_i \triangleq \left\{ v \in \mathbb{R}^{n_v} : x_{\{j\}} = \epsilon_i \text{ if } j = i,\ v_{\{j\}} = 0 \text{ if } j \neq i \right\}.$$

The values $\epsilon_i > 0$ may be chosen as the minimum independent virtual command deviations from equilibrium required for realistic operation of the plant.

Lemma 2. *Defining*

$$G_\epsilon = \begin{bmatrix} I \\ I \\ \vdots \end{bmatrix}, \quad g_\epsilon = \begin{bmatrix} v_g - V_{gg}\tilde{\epsilon}_1 \\ v_g + V_{gg}\tilde{\epsilon}_1 \\ \vdots \end{bmatrix}, \tag{8}$$

the condition required by Assumption 9 can be expressed as a constraint on allowable values of g_u in the (possibly not irredundant) form $G_\epsilon V_{gg} g_u \leq g_\epsilon$.

Proof. For a given $\tilde{\epsilon}_i$, to be inside the set (7) for a given g_u it is necessary by definition that $V_{gv}\tilde{\epsilon}_i \leq v_g - V_{gg} g_u$. This can be re-written as $V_{gg} g_u \leq v_g - V_{gv}\tilde{\epsilon}_i$. The proposition follows by imposing this for each $i \in \{1, \dots, n_v\}$. □

Consider a virtual control allocator $\mathcal{C}(v)\colon \mathbb{V}_g(g_u) \to \mathbb{U}_g(g_u)$, that for any $v \in \mathbb{V}_g(g_u)$ computes $u \in \mathbb{U}_g(g_u)$, such that $v = Mu$. By allowing control reallocation for a terminal control law, the set of states for which a the control law is admissible for a given g_u does not shrink, and is enlarged if there is sufficient redundancy in the system.

Proposition 1. *Following Assumptions 1 and 3, the equilibrium state can be expressed as a function of the equilibrium input or virtual input: $x_s = -(A - I)^{-1} B u_s = -(A - I)^{-1} \hat{B} v_s$.*

Lemma 3. *Define*

$$\mathbb{X}_u(g_u, u_s) = \{x : x \in \mathbb{X},\ K_N\delta x + u_s \in \mathbb{U}_g(g_u)\}$$
$$\mathbb{X}_v(g_u, v_s) = \{x : x \in \mathbb{X},\ MK_N\delta x + v_s \in \mathbb{V}(g_v)\}\,.$$

The set $\mathbb{X}_v(g_u, Mu_s) \supseteq \mathbb{X}_u(g_u, u_s)$ for given admissible values of x_s, u_s and g_u.

Proof. $x \in \mathbb{X}_u(g_u) \implies K_N\delta x + u_s \in \mathbb{U}_g(g_u)$. From the definition of $\mathbb{V}_g(g_u)$, $K_N\delta x + u_s \in \mathbb{U}_g(g_u) \implies M(K_N\delta x + u_s) \in \mathbb{V}(g_u)$. □

In [7, 15, 16], a positively invariant set, admissible with respect to $\mathbb{X}$ and $\mathbb{U}$, parameterised by a basis for the target equilibrium pair (x_s, u_s) was proposed. The following extends this concept with a positively invariant set for a re-allocatable terminal control law, that is also parameterised by the input constraint g_u. Recall that $B = \hat{B}M$.

Lemma 4. *An admissible positively invariant set for the virtual control law $v = MK_N\delta x + v_s \in \mathbb{V}_g(g_u)$, i.e.*

$$\begin{aligned}\mathbb{T}_g(g_u, v_s) \triangleq \{x : x \in \mathbb{X}_v(g_u, v_s),\\ Ax + BK_N\delta x + v_s) \in \mathbb{T}_g(g_u, v_s)\}\end{aligned} \quad (10)$$

can be computed that is affine in terms of g_u, x_s and u_s.

Proof. First consider the dynamics of δx:

$$\begin{aligned}\delta x(k+1) &= Ax(k) + \hat{B}MK_N\delta x(k) + \hat{B}v_s - x_s\\ &= (A + BK_N)\delta x(k).\end{aligned}$$

Consider an augmented autonomous state space system:

$$\begin{bmatrix}\delta x(k+1)\\ v_s(k+1)\\ g_u(k+1)\end{bmatrix} = \begin{bmatrix}(A+BK_N) & 0 & 0\\ 0 & I & 0\\ 0 & 0 & I\end{bmatrix}\begin{bmatrix}\delta x(k)\\ v_s(k)\\ g_u(k)\end{bmatrix} \quad (12)$$

Let $\Gamma = (I - A)^{-1}\hat{B}$. Constraints that must hold $\forall k \in \{0, \ldots, \infty\}$ are:

$$\begin{bmatrix}V_{gv}MK_N & V_{gv} & V_{gg}\\ G_x & G_x\Gamma & 0\\ 0 & G_\epsilon V_{gv} & G_\epsilon V_{gg}\\ 0 & \alpha^{-1}G_x\Gamma & 0\end{bmatrix}\begin{bmatrix}\delta x(k)\\ v_s(k)\\ g_u(k)\end{bmatrix} \le \begin{bmatrix}v_g\\ g_x\\ g_\epsilon\\ g_x\end{bmatrix}. \quad (13)$$

The first row of (13) constrains the virtual command from the terminal control law. The second row is the constraint on state $x = x_s + \delta x$. The third row enforces Assumption 9, and also helps ensure finite-determination of the positive invariant set [19], since $g_u(k)$ is a marginally stable state. The value $\alpha \in (0, 1)$ is a constant, again to ensure finite-determination [15, 19]. The MAS can be computed for system (12) with constraints (13) using algorithms from e.g. [19, 20]. By re-ordering the rows of the computed set, and performing a coordinate transformation $x = \delta x + x_s$, the admissible invariant set for the augmented system can be expressed in the form:

$$\begin{bmatrix}T_{xx} & T_{xv1} & T_{g1}\\ 0 & T_{xv2} & T_{g2}\\ 0 & 0 & T_{g3}\end{bmatrix}\begin{bmatrix}x^T & g_u^T & v_s^T\end{bmatrix}^T \le \begin{bmatrix}t_1\\ t_2\\ t_3\end{bmatrix}. \quad (14)$$

From this, two affinely parameterised sets are obtained. The first is the the parameterised MAS for the plant with virtual terminal control law, parameterised constraints, and parameterised setpoint:

$$\mathbb{T}_g(g_u, v_s) = \{x :\ T_{xx}x \ \le t_1 - T_{g1}g_u - T_{xv1}v_s\ \}. \quad (15)$$

The second is a set of admissible target virtual inputs pairs for which $\mathbb{T}_g(g_u, v_s)$ is defined

$$\mathbb{V}_s(g_u) = \{v_s : T_{xv2}v_s \le t_2 - T_{g2}g_u\}. \quad (16)$$

Finally, the set of g_u for which the parameterisation applies is given by $T_{g3}g_u \le t_3$. □

Following Lemma 3, for a given g_u, $\mathbb{T}_g(g_u, Mu_s)$ is either equal to or a superset of an invariant set computed instead using $\mathbb{X}_u(g_u, u_s)$. A larger terminal set enables a larger domain of feasibility for Problem 1. Note that the control allocator for the terminal control law is not implemented online (although computation of the target u_s is) — the subsequent analysis only relies upon its feasibility.

Lemma 5. *If $x(k) \in \mathcal{S}_N(\mathbb{X}, \mathbb{U}_g(g_u), \mathbb{T}_g(g_u, Mu_s))$, g_u is constant, and (x_s, u_s) are constant and satisfy Assumption 3, then the control law $u(k) = u_0(k)$ is recursively feasible.*

Proof. Feasibility at time $k+1$ can be shown by constructing a candidate solution $(\tilde{\mathbf{u}}(k+1), \tilde{\mathbf{x}}(k+1))$ by truncating and extending the solution from time k:

$$\begin{aligned}\tilde{\mathbf{u}}(k+1) &= \left(u_1^*(k)\cdots u_{N-1}^*(k),\quad \mathcal{C}(M(K_N\delta x_N^*(k) + u_s))\right)\\ \tilde{\mathbf{x}}(k+1) &= \left(x_1^*(k),\quad \cdots,\quad x_N^*(k),\quad (A+BK_N)\delta x_N^*(k)\right)\end{aligned}$$

This candidate solution satisfies (3b), (3c), (3d), and (3e) trivially, and (3f) following from definition of $\mathbb{T}_g(g_u, v_s)$ in (10). The theorem therefore holds by induction. □

2.3. Stabilising terminal cost

Theorem 1. *If for constant g_u, u_s and x_s, it holds that $x(k) \in \mathcal{S}_N(\mathbb{X}, \mathbb{U}_g(g_u), \mathbb{T}_g(g_u, Mu_s))$ then a sufficient condition for $J^*(x(k))$ to be a Control Lyapunov function for the closed-loop system $x(k+1) = Ax(k) + Bu_0^*(k)$ with $u_0^*(k)$ satisfying Problem 1 is that $-\|\delta x_N^*(k)\|_P^2 + \|\delta x_N^*(k)\|_Q^2 + \|\mathcal{C}(MK_N\delta x_N^*(k))\|_R^2 + \|(A + B\mathcal{C}(MK_N\delta x_N^*(k)))\|_P^2 \le 0$.*

Proof. $J^*(x(k)) = 0$ if $x(k) = 0$ and $J^*(x(k)) > 0$ if $x(k) \neq 0$. Considering the same feasible candidate solution as in the proof of Lemma 5, for $x(k) \neq 0$,

$$\begin{aligned}J(x(k+1), \tilde{\mathbf{u}}(k+1)) = J^*(x(k)) - \|\delta x_0^*(k)\|_Q^2 - \|\delta u_0^*(k)\|_R^2\\ - \|\delta x_N^*(k)\|_P^2 + \|\delta x_N^*(k)\|_Q^2\\ + \|\mathcal{C}(MK_N\delta x_N^*(k))\|_R^2 + \|(A + B\mathcal{C}(MK_N\delta x_N^*(k)))\|_P^2\end{aligned}$$

So, $J^*(x(k+1)) \le J(x(k+1), \tilde{\mathbf{u}}(k+1)) \le J^*(x(k)) - \|\delta x_0^*(k)\|_Q^2 - \|\delta u_0^*(k)\|_R^2$, if the stated conditions hold. □

Proposition 2. *The condition of Theorem 1 will be satisfied if P is chosen to satisfy $(A+BK_N)^T P(A+BK_N) - P \leq -Q-Z$, and Z satisfies $\|\delta x_N\|_Z^2 \geq \|\mathcal{C}(MK_N\delta x_N)\|_R^2$ $\forall x_N \in \mathbb{T}_g(g_u)$ for all admissible g_u.*

One method is proposed here to find such a Z (for all g_u). Initially, assume that the virtual control allocator chooses u_N as [18]:

$$\mathcal{C}(MK_N x_N) = \arg\min_{u_N} \quad u_N^T R u_N \tag{17a}$$

$$\text{s.t. } G_u u_N \leq g_u, \quad M u_N = MK_N x_N. \tag{17b}$$

$\mathcal{C}(MK_N x_N)$ is a piecewise-affine function of x_N and g_u, and the optimal cost is piecewise quadratic [21] in x_N and g_u. The value of $u_N^T R u_N$ in terms of x_N can be found by computing the explicit solution of (17) with the vector $[x_N^T, g_u^T]^T$ as a parameter, using multi-parametric programming [22], e.g. using the MPT Toolbox [23]. The solution to the mp-QP control allocator partitions the parameter space into M_p polyhedral regions $\mathcal{P}_i$, each with a local quadratic value function:

$$\begin{bmatrix} x_N \\ g_u \end{bmatrix}^T \begin{bmatrix} \mathcal{A}_{i,11} & \mathcal{A}_{i,12} \\ \mathcal{A}_{i,21} & \mathcal{A}_{i,22} \end{bmatrix} \begin{bmatrix} x_N \\ g_u \end{bmatrix} + \mathcal{B}_i \begin{bmatrix} x_N \\ g_u \end{bmatrix} + \mathcal{C}_i. \tag{18}$$

For the problem of form (17) it turns out that $\mathcal{B}_i$ and $\mathcal{C}_i$ are zero. Since g_u only parameterises inequality constraints, an upper bound on the cost of the allocated terminal control input of the form $x_N^T Z x_N$ where $Z \in \mathbb{R}^{n_x}$ can be computed by considering the marginal cost with respect to the state in each region of the convex, continuous, piecewise-quadratic value function from the mp-QP:

$$\min_Z \text{Trace}(Z) \text{ s.t. } Z \geq \mathcal{A}_{i,11}, \quad i \in \{1, \ldots, M_p\}. \tag{19}$$

This convex semi-definite program [24] can be solved using widely available tools (e.g. SDPT3 [25] or SeDuMi [26]).

3. Feasibility recovery for piecewise constant g_u

If a fault event happens at time k_f, g_u is piecewise constant and the target may need to change in reaction even if the reference r remains constant: $g_u = g_u^{(i)}$, $u_s = u_s^{(i)}$, where $i = 0$ if $k < k_f$ and $i = 1$ if $k \geq k_f$. Lemma 5 only guarantees recursive feasibility for constant g_u. Therefore, recursive feasibility is guaranteed before the transition. It is also guaranteed after the transition *if the transition is feasible* but feasibility of the transition at time k_f is not guaranteed. Since u_s (and thus x_s and v_s) may change, we propose conditions such that *there exists* a feasible (x_s, u_s) at time k_f. (It is unreasonable to guarantee feasibility for all values of g_u — i.e. up until complete input loss.)

Assumption 10. *Define $\mathbb{V}_0 = \{v : V_{g0} v \leq v_{g0}\}$ as a set of virtual commands, such that if $\mathbb{V}_g(g_u) \supseteq \mathbb{V}_0$ feasibility should be recoverable when g_u changes.*

Note that this contrasts to Assumption 9, which defines instead a guaranteed "margin" between the steady state virtual command and the boundary of $\mathbb{V}_g(g_u)$. Define $\mathbb{X}_0$ as the set of original state constraints stemming from a basic MPC design. Consider the set

$$\mathbb{X}_{\mathbb{V}_0}(v_s) \triangleq \{x \in \mathbb{X}_0 : MK_N \delta x + v_s \in \mathbb{V}_0\} \tag{20}$$

Proposition 3. *A set*

$$\mathbb{T}_0(v_s) = \left\{ x \in \mathbb{X}_{\mathbb{V}_0}(v_s) : Ax + \hat{B}(K_N \delta x + v_s) \in \mathbb{T}_0(v_s) \right\} \tag{21}$$

can be computed by considering the MAS for system

$$\begin{bmatrix} \delta x(k+1) \\ v_s(k+1) \end{bmatrix} = \begin{bmatrix} (A+BK_N) & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \delta x(k) \\ v_s(k) \end{bmatrix} \tag{22}$$

with constraints

$$\begin{bmatrix} V_{g0} MK_N & V_{g0} \\ G_{x0} & G_{x0}\Gamma \\ 0 & G_\epsilon V_{g0} \\ 0 & \alpha^{-1} G_{x0}\Gamma \end{bmatrix} \begin{bmatrix} \delta x(k) \\ v_s(k) \end{bmatrix} \leq \begin{bmatrix} v_{g0} \\ g_x \\ g_\epsilon \\ g_x \end{bmatrix} \tag{23}$$

and performing the change of variables $x = \Gamma v_s + \delta x$ and expressed in the form

$$\begin{bmatrix} T_{0x} & T_{0v} \end{bmatrix} \begin{bmatrix} x^T & v_s^T \end{bmatrix}^T \leq t_0. \tag{24}$$

The projection of (24) onto x defines a set of states for which a $\tilde{v}_s$ exists such that $x \in \mathbb{T}_0(\tilde{v}_s)$:

$$\mathbb{X}_{\mathbb{T}_0} \triangleq \{x : \exists v_s \text{ s.t. (24) holds}\}. \tag{25}$$

Express this as $G_{xt} x \leq g_{xt}$. Define the m-step virtual controllable set

$$\mathcal{S}_m^{\mathbb{V}}(\mathbb{X}, \mathbb{V}, \mathbb{E}) \triangleq \{x_0 : \exists v_i \in \mathbb{V} \text{ s.t. } x_i \in \mathbb{X} \text{ for } x_{i+1} = Ax_i + \hat{B}v_i,\, i \in \{0, \ldots, m-1\},\, x_m \in \mathbb{E}\}. \tag{26}$$

Proposition 4. *Let $\mathbb{X} = \mathcal{S}_m^{\mathbb{V}}(\mathbb{X}_0, \mathbb{V}_0, \mathbb{X}_{\mathbb{T}_0})$ for some $m < N$. This ensures that there is a feasible v_s corresponding to a feasible equilibrium, reachable by an m-step input sequence that is admissible with respect to $\mathbb{V}_0$.*

The difficulty with Proposition 4 is that since $\mathbb{T}_g(g_u, v_s)$ is now computed using more stringent state constraints than $\mathbb{T}_0(v_s)$. Therefore, there may exist $g_u^{(i)}$, $i = \{0, 1\}$ satisfying $\mathbb{V}_g(g_u^{(i)}) \supseteq \mathbb{V}_0$ for which the initial state $x_0 \in \mathcal{S}_N(\mathbb{X}, \mathbb{U}_g(g_u^{(0)}), \mathbb{T}_g(g_u^{(0)}, v_s^{(0)})) \;\not\Rightarrow\; \exists v_s^{(1)}$ such that $x_0 \in \mathcal{S}_N(\mathbb{X}, \mathbb{U}_g(g_u^{(1)}), \mathbb{T}_g(g_u^{(1)}, v_s^{(1)}))$. We propose a condition that can be tested *a posteriori* by computing a set analogous to that described in Proposition 3 for the constraint set $\mathbb{X} = \mathcal{S}_m^{\mathbb{V}}(\mathbb{X}_0, \mathbb{V}_0, \mathbb{X}_{\mathbb{T}_0})$ instead of $\mathbb{X} = \mathbb{X}_0$.

Define $\mathbb{X}_{\mathbb{V}_0}^{\mathcal{S}} \triangleq \{x \in \mathbb{X} : MK_N \delta x + v_s \in \mathbb{V}_0\}$ and $\mathbb{T}_0^{\mathcal{S}} \triangleq \{x \in \mathbb{X}_{\mathbb{V}_0}^{\mathcal{S}}(v_s) : Ax + \hat{B}(K_N \delta x + v_s) \in \mathbb{T}_0^{\mathcal{S}}(v_s)\}$, and the projection of this only x: $\mathbb{X}_{\mathbb{T}_0}^{\mathcal{S}}$

Theorem 2. *Let $q \leq N - m$. If $\mathcal{S}_q^{\mathbb{V}}(\mathbb{X}, \mathbb{V}_0, \mathbb{X}_{\mathbb{T}_0}^{\mathcal{S}}) \supseteq \mathbb{X}_{\mathbb{T}_0}$ for $\mathbb{V}_g(g_u) \supseteq \mathbb{V}_0$, then $x_0 \in \mathcal{S}_N(\mathbb{X}, \mathbb{U}_g(g_u^{(0)}), \mathbb{T}_g(g_u^{(0)}, v_s^{(0)})) \implies \exists v_s^{(1)}$ s.t. $x_0 \in \mathcal{S}_N(\mathbb{X}, \mathbb{U}_g(g_u^{(1)}), \mathbb{T}_g(g_u^{(1)}, v_s^{(1)}))$.*

Proof. $x_0 \in \mathbb{X}$ implies $x_0 \in \mathcal{S}_m^{\mathbb{V}}(\mathbb{X}_0, \mathbb{V}_0, \mathbb{X}_{\mathbb{T}_0})$. The set $\mathbb{X}_{\mathbb{T}_0}$ is control invariant, so $\mathcal{S}_{m-1}^{\mathbb{V}}(\mathbb{X}_0, \mathbb{V}_0, \mathbb{X}_{\mathbb{T}_0}) \subseteq \mathcal{S}_m^{\mathbb{V}}(\mathbb{X}_0, \mathbb{V}_0, \mathbb{X}_{\mathbb{T}_0})$, therefore, $x_0 \in \mathbb{S}_m^{\mathbb{V}}(\mathbb{X}, \mathbb{V}_0, \mathbb{X}_{\mathbb{T}_0})$. If the proposed test condition holds then this means that

$$x_0 \in \mathcal{S}_N(\mathbb{X}, \mathbb{U}_g(g_u^{(0)}), \mathbb{T}_g(g_u^{(0)}, v_s^{(0)})) \implies x_0 \in \mathbb{X}$$

$$\implies x_0 \in \mathcal{S}_m^{\mathbb{V}}(\mathbb{X}, \mathbb{V}_0, \mathcal{S}_q^{\mathbb{V}}(\mathbb{X}, \mathbb{V}_0, \mathbb{X}_{\mathbb{T}_0}^{\mathcal{S}})) \tag{27a}$$

$$\implies x_0 \in \mathcal{S}_{m+q}^{\mathbb{V}}(\mathbb{X}, \mathbb{V}_0, \mathbb{X}_{\mathbb{T}_0}^{\mathcal{S}}) \tag{27b}$$

$$\implies x_0 \in \mathcal{S}_N^{\mathbb{V}}(\mathbb{X}, \mathbb{V}_g(g_u), \mathbb{X}_{\mathbb{T}_0}^{\mathcal{S}})\ \forall g_u \colon \mathbb{V}_g(g_u) \supseteq \mathbb{V}_0 \tag{27c}$$

$$\implies x_0 \in \mathcal{S}_N(\mathbb{X}, \mathbb{U}_g(g_u), \mathbb{X}_{\mathbb{T}_0}^{\mathcal{S}})\ \forall g_u \colon \mathbb{V}_g(g_u) \supseteq \mathbb{V}_0 \tag{27d}$$

$$\implies \exists v_s^{(1)} \text{ s.t. } x_0 \in \mathcal{S}_N(\mathbb{X}, \mathbb{U}_g(g_u^{(1)}), \mathbb{T}_g(g_u^{(1)}, v_s^{(1)})). \tag{27e}$$

□

4. Target computation

To track a reference setpoint, the steady state target (x_s, u_s) should be chosen to minimise function of $\|Cx_s - r\|$, where Cx_s is the controlled output and r is the (piecewise constant) reference setpoint [9, 27]. A suitable target calculator may for example compute:

Problem 2.

$$\min_{x_s, u_s} \quad \|F_0(C\Gamma M u_s - r)\|_1 + \|u_s\|_{F_1}^2 + \|x_s\|_{F_2}^2 \tag{28a}$$

$$\textit{subject to: } u_s \in \mathbb{U}_g(g_u), M u_s \in \mathbb{V}_s(g_u),\ x_s = \Gamma M u_s \tag{28b}$$

$$x(k) \in \mathbb{S}_N(\mathbb{X}, \mathbb{U}_g(u_g), \mathbb{T}_g(g_u, v_s)). \tag{28c}$$

The objective (28a) is for the target output to be close to the reference and for preferred inputs to be used if possible. Matrices F_0, F_1, F_2 are weights chosen by the designer, with $F_0 \gg F_1$ and F_2. Constraint (28c) ensures feasibility of Problem 1 (implemented with slack variables rather than projections if required). Problem 2 is a parametric QP in terms of r and g_u since $\mathbb{V}_s$ and $\mathcal{S}_N$ are affine in g_u. Feasibility recovery for changing g_u is accommodated by choosing $\mathbb{X}$ as shown in Section 3. Target calculation can alternatively be performed simultaneously with Problem 1 [15–17].

5. Example

The proposed method is demonstrated on a simple state space system with 4 states, and 4 inputs:

$$[A|\hat{B}] = \left[\begin{array}{cccc|cc} 1.010 & 0.180 & 0 & 0 & 0.019 & 0 \\ 0.001 & 0.840 & 0 & 0 & 0.183 & 0 \\ 0 & 0 & 1.010 & 0.190 & 0 & 0.019 \\ 0 & 0 & 0.001 & 0.880 & 0 & 0.187 \end{array}\right] \tag{29a}$$

$$M = \begin{bmatrix} -0.10 & 0.10 & -0.03 & 0.03 \\ -0.02 & -0.02 & -0.12 & -0.12 \end{bmatrix} \tag{29b}$$

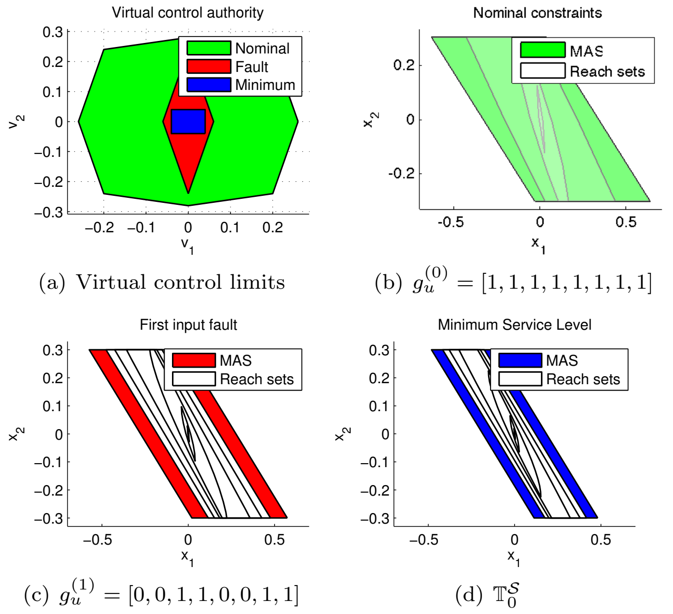
Asymmetric use of inputs (1,2) and (3,4) controls states (1,2), whilst symmetric use of inputs (1,2) and (3,4) controls states (3,4). However, inputs (1,2) have a larger magnitude of effect on states (1,2) and inputs (3,4) have a larger magnitude effect on states (3,4). This setup is representative of longitudinal and lateral dynamics of an aircraft with two elevators (normally used in common mode, mostly affecting the former with slight effect on the latter) and two ailerons (normally used differentially, mostly affecting the latter). The eigenvalues of A are slightly outside the unit disc. A candidate stabilising terminal controller K_N is designed as the unconstrained infinite horizon LQR for the plant (A, B) with weighting matrices: $Q = \text{diag}(0.2,\ 0.0001,\ 0.2,\ 0.0001)$ and $R = \text{diag}(1,\ 1,\ 1,\ 1)$:

$$K_N = \begin{bmatrix} 0.9959 & 0.9870 & 0.1373 & 0.1782 \\ -0.9959 & -0.9870 & 0.1373 & 0.1782 \\ 0.2988 & 0.2961 & 0.8238 & 1.0693 \\ -0.2988 & -0.2961 & 0.8238 & 1.0693 \end{bmatrix} \tag{30}$$

and places the eigenvalues of $(A + BK)$ on the real axis at 0.9761, 0.8304, 0.9686 and 0.8682. (Recall, that whilst we have specified this in the form, $u = K_N x$, we may realise this with any u such that $v = MK_N x$). The input constraints are described by the matrix $G_u = [I, -I]^T \in \mathbb{R}^{8\times 4}$, and the parameter $g_u \in \mathbb{R}^{8\times 1}$. The "fourier" solver from [23] is used to project (5) onto $[v^T, u_g^T]^T$ to obtain $\mathbb{V}_g(g_u)$ (7). Candidate state constraints $\mathbb{X}_0$ of $-1.5 \leq x_1 \leq 1.5$, $-0.3 \leq x_2 \leq 0.3$, $-1.5 \leq x_3 \leq 1.5$ and $-0.3 \leq x_4 \leq 0.3$, are considered.

We define the "minimum service level" on the virtual inputs for which feasibility should be guaranteed recoverable $\mathbb{V}_0$ by the vertices $(0.05, 0)$, $(-0.05, 0)$, $(0, 0.12)$, $(0, -0.12)$. Together with the state constraints and parameters $\alpha = 0.90$ and $\epsilon_i = 0.01$, this enables computation of $\mathbb{T}_0(v_s)$ (21) and hence $\mathbb{X}_{\mathbb{T}_0}$ (25) and $\mathbb{X}$ from Proposition 4, for $m = 10$. The condition of Theorem 2 is satisfied for $q = 0$ for this example! The parameterised set $T_g(g_u, v_s)$ is then computed. $\mathbb{T}_g(g_u, v_s)$ is parameterised by 80 linear inequalities, and $\mathbb{V}_s(g_u)$ by 12. A further 4 constraints only apply to g_u.

To reiterate the advantage of a terminal set in comparison to a terminal equality, slices of $\mathbb{T}_g(g_u, 0)$ with $x_3 = x_4 = 0$ are presented in Figure 1 for nominal conditions $g_u^{(0)} = [1, 1, 1, 1, 1, 1, 1, 1]^T$, the first input pair blocked out, $g_u^{(1)} = [0, 0, 1, 1, 0, 0, 1, 1]$ in addition to slices of the "minimum service level" terminal set $\mathbb{T}_0^{\mathcal{S}}(0)$. These are compared with slices of the 25, 20, 15, 10, 5 and 2-step backwards reach sets from the origin under the same constraints. From the figure, it is evident that a constrained MPC with a terminal equality constraint will require a long horizon to achieve a useful region of feasibility. Moreover, the benefit of terminal control law reallocation is emphasised, since for $g_u^{(1)}$, without allowing terminal control re-allocation, the terminal set for the original terminal



(a) Virtual control limits (b) $g_u^{(0)} = [1,1,1,1,1,1,1,1]$

(c) $g_u^{(1)} = [0,0,1,1,0,0,1,1]$ (d) $\mathbb{T}_0^{\mathcal{S}}$

Figure 1: Admissible $T_g(g_u, 0)$ and reach sets of the origin

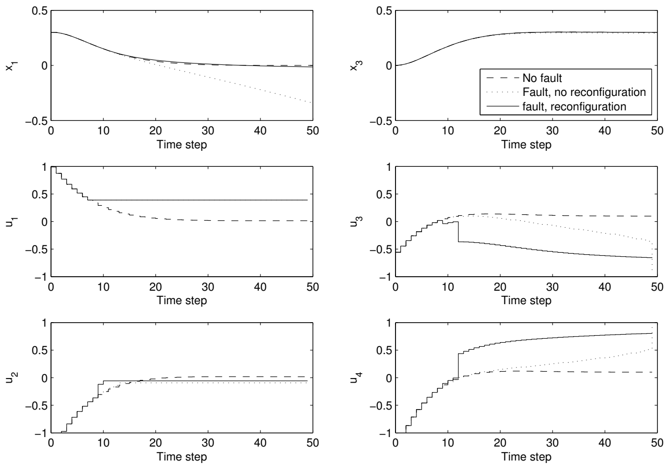


Figure 2: Simulation trajectories with faults on u_1 and u_2

control law would by necessity be the origin, whereas the computed terminal set with re-allocation is larger than the 25-step backwards reach set.

The parameterised terminal set is comprised of 80 half-space constraints. At each stage in the MPC prediction horizon, 8 state and 8 input half space constraints are imposed. Therefore, the terminal constraint has an overhead in terms of constraints in the optimisation problem of 5 prediction horizon steps. Yet the set itself is greater than the region of feasibility of a 25 step MPC with a terminal equality constraint for $g_u^{(1)}$. For $g_u^{(0)}$ the feasibility recoverability constraint $\mathbb{X} = \mathcal{S}_m^{\mathbb{V}}(\mathbb{X}_0, \mathbb{V}_0, \mathbb{X}_{\mathbb{T}_0})$ dominates.

The quadratic weighting matrices for the MPC controller are chosen as: $Q = \text{Diag}(20, 0.01, 20, 0.01)$, and $R = \text{Diag}(1,1,1,1)$. The upper bound Z on the worst-case marginal cost contributions from the inputs applied by the terminal controller, and the corresponding upper bound on the cost-to-go, P are computed using MPT [23] and YALMIP [28] as:

$$Z = \begin{bmatrix} 26.1858 & 25.9515 & 0.0000 & 0.0000 \\ 25.9515 & 25.7193 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 51.6101 & 66.9908 \\ 0.0000 & 0.0000 & 66.9908 & 86.9553 \end{bmatrix} \tag{31a}$$

$$P = \begin{bmatrix} 1.1199 & 1.0070 & 0.0000 & 0.0000 \\ 1.0070 & 0.9584 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 1.3396 & 1.5918 \\ 0.0000 & 0.0000 & 1.5918 & 2.0067 \end{bmatrix} \times 10^3. \tag{31b}$$

To analyse the potential change in closed-loop performance, the unconstrained infinite horizon cost with respect to Q, R is computed for three controllers: the infinite horizon LQR, and unconstrained MPC with $N = 10$ firstly for P satisfying $(A+BK_N)^T P(A+BK_N) - P - K_N^T R K_N - Q = 0$, and then with above conservative terminal cost P. Denote these $x^T P_{lqr} x$, $x^T P_{K_N} x$ and $x^T P_Z x$ respectively. Letting $\lambda_{\max}(.)$ and $\lambda_{\min}(.)$ denote the maximal and minimal eigenvalue respectively, $\lambda_{\max}(P_{K_N} - P_{lqr})/\lambda_{\min}(P_{lqr}) = 0.0409$, and $\lambda_{\max}(P_Z - P_{lqr})/\lambda_{\min}(P_{lqr}) = 0.0698$. The conservative terminal cost is anticipated therefore to not significantly modify closed-loop performance.

Using YALMIP [28], MPC solving Problem 1 and target calculator solving Problem 2 are implemented for $N = 10$ with the costs and constraints specified in this section. The key point, and contribution of this paper is that to reconfigure the terminal set, the only recomputation needed is the calculation of $t_1 - T_{g1} g_u - T_{xv1} v_S$ and $t_2 - T_{g2} g_u$ (where $v_s = M u_s$). Back-to-back inequality constraints may be present which can lead to numerical difficulties with some interior-point QP solvers, but this can be mitigated by using active set methods or by using a pre-solver to transform these linearly dependent inequality constraints into equality constraints.

Figure 2 shows closed loop trajectories for the controlled system, starting from $x(0) = [0.3, 0, 0, 0]^T$ controlled to a new setpoint $[0, 0, 0.3, 0]^T$. The dashed line shows the system with no fault. Also considered is a fault on control pairs 1 and 2 during the transient. When u_1 falls below 0.4, it jams. When u_2 rises above -0.1 it also jams. The dotted line shows the behaviour without reconfiguration. The solid line shows the behaviour when it is assumed that these faults can be detected instantaneously and used to reconfigure g_u. Without parameterisation, to work within this setting for enforcing stability, it would have been necessary to have computed a separate terminal set for the particular values of g_u used (and separately for all other possible values of g_u).

6. Conclusions

A method has been proposed for a stabilising predictive controller for LTI plants with redundant inputs, and parameterised input constraints. To guarantee stability, even in the case of certain classes of fault, the terminal constraint is parameterised by the input constraints to enable on-line reconfiguration in case of fault. By constructing the terminal set in this way, it is not necessary to construct a separate terminal set *a priori* for every combination of input failures. The formulation is extended to accommodate non-zero input faults, such as being stuck at a constant value, or the inability to decrease below a positive threshold. The efforts of the terminal controller that is used to guarantee the existence of a feasible solution at each time step, are implicitly re-distributable between the remaining degrees of freedom when a fault occurs. A method is proposed for obtaining an upper bound in the stage cost that would be incurred, even when an unusual combination of inputs is required to deliver the desired control effect. This is then used to compute a suitable stabilising terminal cost for the MPC. Additional constraints are imposed to guarantee existence of a feasible solution given a minimum level of input availability. The effectiveness of the method is demonstrated with a numerical example.

This stabilising terminal set parameterisation could also be used to complement the constraint-parameterised explicit MPC of [29]. The integration of the proposed reconfiguration methodology with robust MPC techniques that maintain guarantees of recursive feasibility and convergence over some region of the state space in presence of disturbances, even in fault scenarios remains a topic for future investigation, as does integration with fault detection methods.

Acknowledgements

The research leading to these results has received function from the European Union Seventh Framework Programme FP7/2007–2013 under grant agreement no. 314 544.

References

[1] J. M. Maciejowski, Reconfigurable control using constrained optimization, in: Proc. European Control Conf., Brussels, 1997.

[2] J. M. Maciejowski, The implicit daisy-chaining property of constrained predictive control., Applied Mathematics and Computer Science 8 (4) (1998) 101–117.

[3] J. M. Maciejowski, C. N. Jones, MPC fault-tolerant flight control case study: Flight 1862, in: IFAC Safeprocess Conf., 2003.

[4] M. M. Kale, A. J. Chipperfield, Robust and stabilized MPC formulations for fault tolerant and reconfigurable flight control, in: Proc. IEEE Int. Symp. Intelligent Control, Taipei, Taiwan, 2004, pp. 222–227.

[5] M. M. Kale, A. J. Chipperfield, Stabilized MPC formulations for robust reconfigurable flight control, Control Eng. Pract. 13 (6) (2005) 771–788.

[6] Y. Zhang, J. Jiang, Bibliographical review on reconfigurable fault-tolerant control systems, Annual Reviews in Control 32 (2) (2008) 229–252. doi:10.1016/j.arcontrol.2008.03.008.

[7] F. A. De Almeida, D. Leißling, Fault-tolerant model predictive control with flight-test results, J. Guidance, Control, and Dynamics 33 (2) (2010) 363–375.

[8] E. Camacho, T. Alamo, D. de la Pena, Fault-tolerant model predictive control, in: Proc. IEEE Conf. Emerging Technologies and Factory Automation, 2010, pp. 1–8. doi:10.1109/ETFA.2010.5641226.

[9] F. A. De Almeida, Reference management for fault-tolerant model predictive control, J. Guidance, Control, and Dynamics 34 (1) (2011) 44–56. doi:10.2514/1.50938.

[10] A. Yetendje, M. M. Seron, J. De Doná, Robust multiactuator fault-tolerant MPC design for constrained systems, International Journal of Robust and Nonlinear Controldoi:10.1002/rnc.2854.

[11] J. M. Maciejowski, Predictive Control with Constraints, Pearson Education, 2002.

[12] J. B. Rawlings, D. Q. Mayne, Model predictive control: Theory and design, Nob Hill Publishing, 2009.

[13] D. Q. Mayne, J. B. Rawlings, C. V. Rao, P. O. M. Scokaert, Constrained model predictive control: Stability and optimality, Automatica 36 (6) (2000) 789–814. doi:10.1016/S0005-1098(99)00214-9.

[14] J. A. Rossiter, B. Kouvaritakis, M. J. Rice, A numerically robust state-space approach to stable-predictive control strategies, Automatica 34 (1) (1998) 65–73. doi:10.1016/S0005-1098(97)00171-4.

[15] D. Limon, I. Alvarado, T. Alamo, E. F. Camacho, MPC for tracking piecewise constant references for constrained linear systems, Automatica 44 (9) (2008) 2382–2387. doi:10.1016/j.automatica.2008.01.023.

[16] A. Ferramosca, D. Limon, I. Alvarado, T. Alamo, E. F. Camacho, MPC for tracking with optimal closed-loop performance, Automatica 45 (8) (2009) 1975–1978. doi:10.1016/j.automatica.2009.04.007.

[17] M. Zeilinger, C. Jones, M. Morari, Robust stability properties of soft constrained MPC, in: Proc. 49th IEEE Conf. Decision and Control, Atlanta, GA, 2011, pp. 5276–5282. doi:10.1109/CDC.2010.5717488.

[18] M. Bodson, Evaluation of optimization methods for control allocation, J. Guidance, Control, and Dynamics 25 (4) (2002) 703–711. doi:10.2514/2.4937.

[19] E. G. Gilbert, K. T. Tan, Linear systems with state and control constraints: the theory and application of maximal output admissible sets, IEEE Trans. Autom. Control 36 (9) (1991) 1008–1020. doi:10.1109/9.83532.

[20] E. C. Kerrigan, Matlab invariant set toolbox 0.10.5 (2005).

[21] A. Bemporad, M. Morari, V. Dua, E. N. Pistikopoulos, The explicit linear quadratic regulator for constrained systems, Automatica 38 (2002) 3–20. doi:10.1016/S0005-1098(01)00174-1.

[22] T. Johansen, T. Fossen, F. Tøndel, Efficient optimal constrained control allocation via multiparametric programming, J. Guidance, Control, and Dynamics 28 (3) (2005) 506–515. doi:10.2514/1.10780.

[23] M. Kvasnica, P. Grieder, M. Baotić, Multi-Parametric Toolbox (MPT) (2004).

[24] S. Boyd, L. E. Ghaoui, E. Feron, V. Balakrishnan, Linear Matrix Inequalities in System and Control Theory, SIAM, 1994.

[25] K. C. Toh, R. H. Tütüncü, M. J. Todd, On the implementation and usage of SDPT3 – a MATLAB software package for semidefinite-quadratic-linear programming, version 4.0 (Jul. 2006).

[26] D. Peaucelle, D. Henrion, Y. Labit, K. Taitz, User's Guide for SeDuMi Interface (2002).

[27] K. R. Muske, J. B. Rawlings, Model predictive control with linear models, AIChE J. 39 (2) (1993) 262–287.

[28] J. Löfberg, YALMIP: A toolbox for modeling and optimization in MATLAB, in: Proc. CACSD Conference, Taipei, Taiwan, 2004.

[29] V. Puig, A. Roisch, C. Ocampo-Martinez, R. Sarrate, Fault-tolerant explicit MPC of PEM fuel cells, in: Proc. 46th IEEE Conf. Decision and Control, New Orleans, LA, 2007, pp. 2657–2662. doi:10.1109/CDC.2007.4434655.