Robust Predictive Control with Feasible Contingencies for Fault Tolerance

Rohan C. Shekhar ∗ Jan M. Maciejowski ∗∗

∗ Department of Engineering, University of Cambridge, Cambridge, United Kingdom (e-mail: rcs48@cam.ac.uk).
∗∗ Department of Engineering, University of Cambridge, Cambridge, United Kingdom (e-mail: jmm@eng.cam.ac.uk).

Abstract: This paper introduces the notion of M-step robust fault tolerance for discrete-time systems where finite-time completion of a control manoeuvre is desired. It considers a scenario with two distinct objectives; a primary and secondary target are specified as sets to be reached in finite-time, whilst satisfying operating constraints on the states and inputs. The primary target is switched to the secondary target when a fault affects the system. As it is unknown when or if the fault will occur, the trajectory to the primary target is constrained to ensure reachability of the secondary target within M steps. A variable-horizon linear MPC formulation is developed to illustrate the concept. The formulation is then extended to provide robustness to bounded disturbances by use of tightened constraints. Simulations demonstrate the efficacy of the controller formulation on a double-integrator model.

Keywords: fault-tolerance, robust control, predictive control, autonomous control.

1. INTRODUCTION

In real world systems, known but unpredictable faults can adversely affect controller performance, altering system dynamics and constraints. When a fault occurs, it is possible that a new control objective will be required; in fact, the original objective may no longer be feasible for the altered system. In this instance, it is desirable that a new feasible objective exists. This could be specified a-priori or chosen dynamically depending on the nature of the fault. In this paper, the former case is considered, where the target region to be reached under fault is chosen in advance.

This paper builds upon ideas from prior work on nominal MPC formulations having additional constraints for fault-tolerance. Breger and How [2008] design a passive fault-tolerance strategy for spacecraft rendezvous that guarantees collision free abort trajectories in case of thruster failure. This strategy assumes a total loss of control authority, so it is possible to explicitly calculate spacecraft behaviour after failure by propagating open-loop dynamics. Schouveenaars et al. [2004], on the other hand, ensure the existence of a safe periodic trajectory at the end of every prediction horizon, as applied to vehicle path planning.

In combining elements of these strategies, the formulation outlined in the following sections ensures that the system state can be steered to a well-defined target region if a fault occurs. It does so by explicitly guaranteeing the existence of a controlled trajectory, or contingency under fault at every future state prediction. An extension provides robustness to bounded disturbances.

Section 1 defines M-step and robust M-step fault tolerance for a general discrete-time system. Section 2 details two linear predictive control algorithms which are proven to have the desired fault-tolerant properties for the nominal and robust cases respectively. Section 3 provides an illustrated example of the robust algorithm applied to a simple double-integrator vehicle model. Conclusions and future work are outlined in section 4.

2. FAULT-TOLERANCE

2.1 Problem Formulation

Consider a switched discrete-time nonlinear system of the form

\[ x(k + 1) = \begin{cases} f(x(k), u(k)) & k < k_f \\ \tilde{f}(x(k), u(k)) & k \geq k_f, \end{cases} \]

where \( k_f \) denotes the time at which a given fault occurs and the system dynamics change. The input vector \( u(k) \) and state vector \( x(k) \) are subject to the switched constraints

\[ u(k) \in \begin{cases} \mathcal{U} \subset \mathbb{R}^n & k < k_f \\ \tilde{\mathcal{U}} \subset \mathbb{R}^n & k \geq k_f \end{cases} \]

\[ x(k) \in \begin{cases} \mathcal{X} \subset \mathbb{R}^n & k < k_f \\ \tilde{\mathcal{X}} \subset \mathbb{R}^n & k \geq k_f, \end{cases} \]

where \( \mathcal{U}, \tilde{\mathcal{U}} \) are compact and \( \mathcal{X}, \tilde{\mathcal{X}} \) are closed. For consistency it is required that \( \mathcal{X} \subseteq \tilde{\mathcal{X}} \).

The control problem in the absence of a fault is to find an admissible feedback law \( u(k) = \kappa(x(k)) \), \( \kappa : \mathcal{U} \rightarrow \mathcal{X} \) that steers the state in some optimal way to a target region \( \Omega \subset \mathcal{X} \) in finite time \( k_c \). Note that there is no requirement imposed on the behaviour of the system once the state has reached \( \Omega \), so this set does not need to be positively invariant.

When the fault affects the system, that is if \( k_f < k_c \), the control objective is switched to a secondary target set \( \tilde{\Omega} \subset \tilde{\mathcal{X}} \), to be reached within \( M \) steps from the time the
fault has occurred. As the time \(k_f\) is unknown, the system
must be able to reach this secondary target admissibly,
regardless of when the fault occurs up to time \(k_c\). For
notational convenience, define the input vector \(\mu(j)\) and
state vector \(\phi(j)\) as

\[
\mu(j) \equiv u(k + k_f + j) \\
\phi(j) \equiv x(k + k_f + j),
\]

which will be used to represent the fault dynamics. The
index \(j \geq 0\), hence these variables only have meaning from
the time the fault occurs. Using this notation, define the
\(q\)-step controllable set [Kerrigan, 2000]

\[
\mathcal{K}_q(\Omega) = \left\{ \phi(0) \in \mathbb{R}^n \left| \exists \{\mu(j)\} \in \bar{U}^q_0, \right. \left. \phi(j) \in \hat{X}^q_0, \phi(q) \in \Omega \right\}
\]

which is the set of all states for which there exists a feasible input sequence under fault that, from \(k_f\) onwards, admissibly steers the system (1) to the secondary target set \(\Omega\) in exactly \(q\) steps. Criterion 1 uses the \(q\)-step controllable set to impose a constraint on the nominal system’s state trajectory that ensures the secondary target is reached in \(M\) steps or fewer.

**Criterion 1.** The controlled system (1) is considered (nomially) \(M\)-step fault-tolerant to the fault described by the switched dynamics and constraints if the set

\[
\mathcal{F}_M(\Omega) = \bigcup_{q=0}^{M} \mathcal{K}_q(\Omega)
\]

is non-empty and there exists an admissible input sequence that drives the system state to the set \(\Omega\) in finite time \(k_c\), where the state trajectory satisfies

\[
x(k) \in \mathcal{F}_M(\Omega)
\]

for all \(k < k_c\).

**Remark 2.** Full state information is assumed during normal operation and under fault.

### 2.2 Robustness

Bounded disturbances are now introduced into the nominal formulation (1). Define the disturbance

\[
w(k) \in \left\{ \mathcal{W} \subset \mathbb{R}^n \left| k < k_f \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \r

3. LINEAR PREDICTIVE CONTROL

Having introduced the criteria for fault-tolerance of a general nonlinear system, the specific case of a linear time-invariant system will be analysed. A variable horizon predictive controller will then be formulated to satisfy the criteria.

#### 3.1 Linear System

A linear system is described by the equation

\[
f(x(k), u(k)) = Ax(k) + Bu(k),
\]

where \(A \in \mathbb{R}^{n \times n}\) and \(B \in \mathbb{R}^{n \times m}\). The state and input constraints take the form of (2), except that sets are required to be polytopes. The primary objective is to admissibly reach the set \(\Omega\) in finite time, starting at state \(x(0)\), whilst minimising a \(p\)-norm cost function of the form,

\[
J(x(0), u) = \sum_{k=0}^{N(x(0)) - 1} \left(1 + \|Qx(k)\| + \|Rx(k)\|\right),
\]

where \(u = \{u(0), u(1), \ldots, u(N(x(1))-1)\}, Q \succeq 0\) and \(R > 0\). \(N(x(0))\) indicates when completion occurs, given initial state \(x(0)\) i.e.

\[
x(N(x(0))) \in \Omega.
\]

\(J(\cdot, \cdot)\) penalises time to completion, state-based operating costs and control effort.

In a similar way, the dynamics under the occurrence of a given fault can be described by the equation

\[
\tilde{f}(x(k), x(k)) = \tilde{A}x(k) + \tilde{B}u(k),
\]

where \(\tilde{A} \in \mathbb{R}^{n \times n}\) and \(\tilde{B} \in \mathbb{R}^{n \times m}\). The constraint sets under fault and the secondary objective are also required to be polytopes. The system must be able to reach the set \(\tilde{\Omega}\) in \(M\) steps regardless of when the fault occurs.

#### 3.2 Controller Formulation

An MPC controller can be formulated to meet all of these objectives. Define the collection of decision variables \(\Theta \equiv \{x(k + i|k), u(k + i|k), \phi_i(j|k), \mu_i(j|k), N(k), M_i(k)\}\), where the notation \(x(k + i|k)\) indicates the prediction of state variable \(x\) at time \(k+i\), given its value at \(k\). Variable
\( \phi_i \) represents the anticipated state trajectory under fault, propagated from the time \( k + i \) under control inputs \( \mu_i \). \( M_i(k) \) represents the time taken for the trajectory under a fault anticipated at time \( k+i \) to reach the secondary target set. It is bounded from above by a positive integer \( M \), to provide \( M \)-step fault-tolerant control. The time index \( j \) is used to propagate the dynamics after the anticipated fault, as in the simplified notation (3). Indices \( i \) and \( j \) take values conditional on the variables that they index:

\[
\begin{align*}
  i & \in \{0, \ldots, N(k) - 1\} & \text{for } u, \mu_i \text{ and } M_i \\
  j & \in \{0, \ldots, M_i(k) - 1\} & \text{for } \phi_i
\end{align*}
\]

The required optimisation problem is then given by

\[
J^*(x(k)) = \min_{\theta} \sum_{i=0}^{N(k)-1} \left( 1 + \|Qx(k+i[k]) + ||Ru(k+i[k])\| \right)
\]

subject to the constraints

\[
\begin{align*}
  x(k[k]) = x(k) \\
  \phi_i(0[k]) = x(k+i[k]) \\
  x(k+i + 1[k]) = Ax(k+i[k]) + Bu(k+i[k]) \\
  \phi_i(j+1[k]) = \bar{A}\phi_i(j[k]) + \bar{B}\mu_i(j[k]) \\
  x(k+i[k]) \in \mathcal{X}, & \quad \phi_i(j[k]) \in \bar{\mathcal{X}} \\
  u(k+i[k]) \in \mathcal{U}, & \quad \mu_i(j[k]) \in \bar{\mathcal{U}} \\
  x(k+N(k)[k]) \in \Omega, & \quad \phi_i(M_i(k)[k]) \in \bar{\Omega}
\end{align*}
\]

with

\[
0 \leq M_i(k) \leq M, \quad N(k) \geq 0, \tag{18c}
\]

for \( Q \geq 0 \) and \( R > 0 \).

Algorithm 1 describes the action of the MPC controller in terms of this optimisation problem.

**Algorithm 1** (\( M \)-Step Fault-Tolerant MPC).

1. Given the state \( x(k) \) at current time \( k \), solve the optimisation (17) subject to constraints (18) to obtain \( N(k) \) and \( \{u^*(k+i[k])\} \).
2. Apply the first element of \( \{u^*(k+i[k])\} \) as the input to the system i.e. \( x(k) = u^*(k) \).
3. If \( x(k) \in \Omega \) then terminate the optimisation. Else, if a fault has occurred, switch to an auxiliary controller to steer the state towards \( \bar{\Omega} \). If none of these cases apply, then set \( k \rightarrow k + 1 \) and go to step 1.

**Theorem 5.** The controller described by algorithm 1 ensures that the primary target is reached in at most \( [J^*(x(0))] \) steps if no fault occurs.

**Proof.** The proof follows the same reasoning as Richards and How [2003], as the fault trajectory predictions are not costed. Given the optimal solution to (17) at time \( k \), say \( \{x^*(k+i[k]), \{u^*(k+i[k])\} \} \) and \( N^*(k) \), a feasible solution at time \( k+1 \) is given by

\[
\begin{align*}
  \hat{x}(k+i+1[k+1]) &= x^*(k+i+1[k]) \\
  \hat{u}(k+i+1[k+1]) &= u^*(k+i+1[k]) \\
  \hat{N}(k+1) &= N^*(k) - 1
\end{align*}
\]

Then, it can be shown that the cost of the candidate solution, denoted \( \hat{J}^*(\cdot) \), satisfies

\[
\hat{J}(x(k+1)) = J^*(x(k)) - \|Qx(k)\| - \|Ru(k)\| - 1 \leq J^*(x(k)) - 1
\]

By optimality, \( \hat{J}^*(\cdot) \) is an upper bound on the optimal cost at time \( k+1 \), hence

\[
J^*(x(k)) - J^*(x(k+1)) \geq 1,
\]

which shows that the optimal cost reduces by at least 1 at each time step. As the cost must be non-negative by construction, the time to completion satisfies the bound

\[
N(x(0)) \leq [J^*(x(0))]
\]

**Theorem 6.** The controller described by algorithm 1 ensures that the closed loop system is \( M \)-step fault-tolerant to the fault specified by (16).

**Proof.** Theorem 5 shows that the controller steers the system state to the set \( \Omega \) in finite time. By construction, the constraints (18) ensure that for every predicted state, there exists a trajectory that steers the system state under fault to the set \( \bar{\Omega} \) in \( M \) steps or fewer, via the fault propagation variables \( \phi_i \) and \( \mu_i \). They implicitly impose the constraint \( x(k+i[k]) \in \mathcal{F}_M(\Omega) \).

**Remark 7.** This algorithm only ensures the existence of **feasible** trajectories to the secondary target. The fault propagation variables do not enter the cost function, so the actual trajectory taken by the system to reach \( \bar{\Omega} \) will depend on the controller and cost function used after the fault has occurred.

### 3.3 Robustness

Adding a bounded disturbance into the linear system gives the dynamics

\[
f(x(k), u(k), w(k)) = Ax(k) + Bu(k) + w(k), \tag{19}
\]

The set containing the disturbance, \( \mathcal{W} \) is assumed to be polytopic. When the fault occurs, the dynamics are altered to

\[
\hat{f}(x(k), u(k), w(k)) = \hat{A}x(k) + \hat{B}u(k) + w(k), \tag{20}
\]

where the set containing the disturbance under fault, \( \mathcal{W} \) is also polytopic. Using the notation of Richards and How [2006], recursively define the matrices

\[
\begin{align*}
  L(0) &= I_n, & L(i+1) &= (A + BK(i))L(i) \\
  \Lambda(0) &= I_n, & \Lambda(j+1) &= (\hat{A} + \hat{B}\hat{K}(j))\Lambda(j)
\end{align*}
\]

and the corresponding sets

\[
\begin{align*}
  \mathcal{X}(k) &= \mathcal{X}, & \mathcal{X}(i+1) &= \mathcal{X}(i) \sim L(i)\mathcal{W} \\
  \mathcal{U}(0) &= \mathcal{U}, & \mathcal{U}(i+1) &= \mathcal{U}(i) \sim K(i)L(i)\mathcal{W} \\
  \Omega(0) &= \Omega, & \Omega(i+1) &= \Omega(i) \sim L(i)\mathcal{W} \\
  \mathcal{X}_0(0) &= \mathcal{X}, & \mathcal{X}_0(j+1) &= \mathcal{X}_0(j) \sim \Lambda(j)\mathcal{W} \\
  \mathcal{U}_0(0) &= \mathcal{U}, & \mathcal{U}_0(j+1) &= \mathcal{U}_0(j) \sim \hat{K}(j)\Lambda(j)\mathcal{W} \\
  \Omega_0(0) &= \Omega, & \Omega_0(j+1) &= \Omega_0(j) \sim \Lambda(j)L(i)\mathcal{W} \\
  \mathcal{X}_{i+1}(j) &= \mathcal{X}_0(j) \sim \Lambda(j)L(i)\mathcal{W} \\
  \mathcal{U}_{i+1}(j) &= \mathcal{U}_0(j) \sim \hat{K}(j)\Lambda(j)L(i)\mathcal{W} \\
  \Omega_{i+1}(j) &= \Omega_0(j) \sim \Lambda(j)L(i)\mathcal{W}
\end{align*}
\]

Matrices \( K(i) \) and \( \hat{K}(j) \) represent candidate control polices, the existence of which ensures robustness to bounded disturbances. The operator \( \sim \) denotes the Pontryagin set difference, or morphological erosion, defined for sets \( \mathcal{A} \) and \( \mathcal{B} \) by the identity

\[
\mathcal{A} \sim \mathcal{B} \equiv \{x|x + b \in \mathcal{A}, \forall b \in \mathcal{B}\}. \tag{22}
\]
A matrix $D$ multiplying a set $W$ is to be interpreted as the new set
\[ D W = \{ y | y = Dx, x \in W \}. \]
The following algorithm describes how the contracted sets are used in order to obtain a robust controller.

**Algorithm 2 (Robust M-Step Fault-Tolerant MPC).**

(1) Given the state $x(k)$ at current time $k$, solve the optimisation (17) subject to (18), enforcing the constraints
\[
\begin{align*}
x(k + i)|k] \in X(i) \\
u(k + i)|k] \in U(i) \\
\phi_i(j)|k] \in \tilde{X}_i(j) \\
\mu_i(j)|k] \in \tilde{U}_i(j) \\
x(k + N(k)|k] \in \Omega(N(k)) \\
\phi_i(M_i(k)|k] \in \tilde{\Omega}_i(M_i(k))
\end{align*}
\]
in lieu of (18b).

(2) Apply the first element of $\{u^*(k + i)|k]\}$ as the input to the system i.e. $u(k) = u^*(k)|k]$. 

(3) If $x(k) \in \Omega$ then terminate the optimisation. Else, if a fault has occurred, switch to the auxiliary controller to steer the state towards $\tilde{\Omega}$. If none of these cases apply, then set $k \rightarrow k + 1$ and go to step 1.

**Remark 8.** The tightening of constraint sets in (21) for the inputs $\mu_i$ and states $\phi_i$ is essentially carried out twice, to account for the disturbances under normal operation as well as those under fault.

In order to show that this algorithm ensures fault-tolerant control action, it is necessary to prove that it is recursively feasible and guarantees finite-time completion. This will be established in the following two theorems.

**Theorem 9.** Given the solution to the optimisation problem described by algorithm 2 at time $k$, namely $\{x^*(k + i)|k]\}, \{u^*(k + i)|k]\}, \{\phi^*_i(j)|k]\}, \{\mu^*_i(j)|k]\}, N^*(k)$ and $M^*(i)|k]$), a feasible solution at time $k + 1$ is given by:
\[
\begin{align*}
\hat{x}(k + 1 + i)|k] & = x^*(k + i + 1|k] + L(i)|w(k) \\
\hat{u}(k + 1 + i)|k] & = u^*(k + i + 1|k] + K(i)|L(i)|w(k) \\
\hat{\phi}_i(j)|k] & = \phi^*_i+1(j)|k] + A(j)|L(i)|w(k) \\
\hat{\mu}_i(j)|k] & = \mu^*_i+1(j)|k] + \bar{K}(j)|L(i)|w(k) \\
\tilde{N}(k + 1) & = N^*(k) - 1 \\
\tilde{M}_i(k + 1) & = M^*_i+1(k)
\end{align*}
\]

**Proof.** The proof follows a similar methodology to Richards and How [2006], demonstrating that the candidate solution (24) satisfies the required equality and set membership constraints. One difference from Richards and How [2006] is the consideration of input as opposed to output constraints. A sketch of the proof is as follows.

Satisfaction of the dynamics constraints can be easily shown by using the fact that the optimal solution at time $k$ satisfies the dynamic constraints at prediction time $k + i + 1$. As an example, the fault states at time $k + 1$ satisfy
\[
\phi^*_i+1(j + 1)|k] = \hat{\phi}^*_i(j)(j + 1|k] + B\hat{\mu}^*_i(j)|k].
\]

Substituting the candidate solution gives
\[
\begin{align*}
\hat{\phi}_i(j + 1|k] & = \hat{\phi}(j)(j + 1|k] + B\hat{\mu}(j)|k] \\
& = \hat{A}\phi(j)(j + 1) + B\hat{\mu}(j)|k] - \lambda(j)\Lambda(j)|L(i)|w(k)
\end{align*}
\]

Therefore,
\[
\hat{\phi}_i(j + 1|k] = \hat{A}\phi(j)(j + 1) + B\hat{\mu}(j)|k] + \lambda(j)\Lambda(j)|L(i)|w(k).
\]

An example of the proof methodology for set membership constraints is given for the inputs under fault. From the optimal solution at time $k$,
\[
\mu^*_i+1(j)|k] \in \tilde{U}_i+1(j)
\]

Then, from the definition of the Pontryagin difference (22),
\[
\mu^*_i+1(j)|k] \in \tilde{U}_i(j) \cap \tilde{K}(i)\Lambda(j)|L(i)|w(k)
\]

Therefore,
\[
\mu^*_i+1(j)|k] \in \tilde{U}_i(j).
\]

In a similar manner, it can be shown that the remaining states and inputs are admissible. It can also be shown that the terminal set constraints are satisfied.

The above analysis shows that, from an initial feasible solution, there exists a feasible solution at the following time step. By induction, this means that the entire optimisation is feasible to completion, for any allowable disturbance sequence.

Finite-time completion is shown by extending the proof from the nominal case.

**Theorem 10.** If the cost weightings on the states and inputs are chosen such that
\[
\lambda \equiv 1 - \max_{w \in W} \max_{i=0}^{\infty} \left\| Q(i) \right\| + \max_{i=0}^{\infty} \left\| R(i) \right\| > 0,
\]
algorithm 2 ensures that for all allowable disturbance realisations, the set $\Omega$ is reached in $H$ steps or fewer if a fault does not occur, where
\[ H = \left[ \frac{J^*(x(0))}{\lambda} \right]. \]

**Proof.** By repeated application of the triangle inequality and maximising over all possible disturbance sequences and horizon lengths, it can be shown that the cost of (24), denoted \( \bar{J}(\cdot) \), satisfies
\[ \bar{J}(x(k + 1)) \leq J^*(x(k)) - \lambda. \]
Again from optimality, \( \bar{J}(\cdot) \) is an upper bound on the optimal cost at time \( k + 1 \), hence
\[ J^*(x(k)) - J^*(x(k + 1)) \geq \lambda, \]
which shows that the optimal cost reduces by at least \( \lambda \) at each step. As the cost must be non-negative by construction, the time to completion satisfies the bound
\[ N(x(0)) \leq H \]

Robust fault tolerance can now be proven.

**Theorem 11.** The controller specified by algorithm 2 with the tightened constraints (23) is robustly M-step fault tolerant.

**Proof.** Algorithm 2 provides a control law that steers the state to the set \( \Omega \) in finite time, for all allowable disturbance sequences, in the absence of a fault. For any predicted state \( x(k + i|k) \), the constraints (23) ensure that there exists a sequence of inputs under fault that satisfies
\[ \phi_i(j|k) \in \bar{X}_i(j) \subseteq \bar{X}_0(j), \]
\[ \mu_i(j|k) \in \bar{U}_i(j) \subseteq \bar{U}_0(j). \]
This follows from the set definitions (21). In particular, observe that the sets \( \bar{X}_0(j) \) and \( \bar{U}_0(j) \) are defined in such a way that if a variable horizon MPC controller is used to steer the system to the target set \( \tilde{\Omega} \) from the time when the fault occurs, it will be recursively feasible for any disturbance sequence \( \{u(k) \in \mathcal{U}\} \). This can be proven in a manner similar to theorem 9. Hence, there exists a control law that steers the system from \( x(k + i|k) \) to the secondary terminal set \( \tilde{\Omega} \), for all allowable disturbance sequences under fault. By construction, it does so in \( M \) steps or fewer, as specified in (18c). Each state prediction is therefore constructed to satisfy
\[ x(k + i|k) \in \bar{F}_M(\tilde{\Omega}) \]

4. **EXAMPLE**

A simple double-integrator model will be used to illustrate the action of the robust fault-tolerant MPC algorithm.

4.1 **Control Problem**

Consider a 1 kg point mass moving in two-dimensions, actuated by orthogonal forces in the \( x \) and \( y \) directions. Treating the mass as a fictitious vehicle, fuel usage is quantified as the integral of the absolute actuator forces over time. Hence, there are two inputs representing the force components and five states representing the position, velocity and fuel. In order to measure the fuel usage, the inputs are restructured to capture the positive and negative components separately, giving four inputs in the resulting system, constrained to be non-negative. Discretising with a sampling frequency of 1 Hz gives the state-space matrices

\[
\begin{bmatrix}
1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1
\end{bmatrix}, \quad
\begin{bmatrix}
0.5 & -0.5 & 0 & 0 \\
0 & 0 & 0.5 & -0.5 \\
1 & -1 & 0 & 0 \\
0 & 0 & 1 & -1 \\
-1 & -1 & -1 & -1
\end{bmatrix}
\]

The velocity of the vehicle is constrained to have a maximum magnitude of 2.5 m s\(^{-1}\). To restrict the maximum actuation and ensure non-negativity, the inputs satisfy the element-wise constraints
\[ 0 \leq u(k) \leq 1 \text{ N}. \]

The system is assumed to start with 12 N s of fuel, which limits the overall use of control authority (or impulse applied to the vehicle), as the fuel state is constrained to be non-negative. The control objective is to minimise the cost function
\[ J(x(0), \{u(k)\}) = \sum_{k=0}^{N(x(0))-1} 1 + \|u(k)\|_1, \]
whilst steering the system to a square target set having side length 3 m and lower-left vertex \((-1.5, 11)\) m.

The system is known to be vulnerable to a certain fault, where it begins to leak fuel at the rate of 10% of its current limits the overall use of control authority (or impulse applied to the vehicle), as the fuel state is constrained to be non-negative. The control objective is to minimise the cost function

4.2 **Simulation**

In order to simulate the controlled system, a mixed-integer formulation is used to make the horizon length a variable in the optimisation problem (refer to Richards and How [2003, 2006] for details). Essentially, binary variables are introduced to indicate completion. The resulting optimisation problem is a mixed-integer linear program (MILP), which must be solved at each time step until completion is achieved. To ensure that the problem remains linear and convex, the magnitude constraints on the vehicle speed and the wind disturbance are approximated by inscribed and circumscribed polygons respectively.

Robustness is ensured by tightening these polygonal constraint sets according to (21). The process is simplified by choosing nilpotent candidate controllers for constraint tightening, which greatly reduces the number of sets that need to be calculated. It can be verified that, for the system operating normally (26), the gain matrix

\[
K = \begin{bmatrix}
-0.5 & 0 & -0.75 & 0 & 0.25 \\
0.5 & 0 & 0.75 & 0 & 0.25 \\
0 & -0.5 & 0 & -0.75 & 0.25 \\
0 & 0.5 & 0 & 0.75 & 0.25
\end{bmatrix}
\]
renders the system degree-2 nilpotent. For the system under fault (27), the matrix
\[
\tilde{K} = \begin{bmatrix}
-0.5 & 0 & -0.75 & 0 & 0.2036 \\
0.5 & 0 & 0.75 & 0 & 0.2036 \\
0 & -0.5 & 0 & -0.75 & 0.2240 \\
0 & 0.5 & 0 & 0.75 & 0.2240
\end{bmatrix}
\]
also gives degree-2 nilpotency.

A Monte Carlo simulation of the controlled, discretised system is performed over 20 sequences of allowable disturbances. Figure 1 shows the trajectories under normal operation and anticipated fault. The solid lines represent the trajectories to the primary target and the dotted lines represent the optimised trajectories to the secondary target set if a fault occurs. It is clear that the distribution of trajectories entering the primary target is skewed towards the right, illustrating the effect of requiring robust reachability of \( \hat{\Omega} \) from any state prediction. Figures 2 and 3 show the actuator forces and speed respectively. These quantities satisfy the imposed constraints, indicated by the dashed lines, in the presence of the disturbance force.

![Fig. 1. Trajectories under normal operation and anticipated faults](image1)

![Fig. 2. Actuator inputs in x and y directions](image2)

5. CONCLUSION AND FUTURE WORK

This paper has introduced the concept of \( M \)-step robust fault tolerance, presenting a variable-horizon fault-tolerant MPC controller formulation for linear systems. This formulation has been shown to have the required properties of finite-time completion and recursive feasibility. Simulations have also demonstrated its efficacy when applied to a double-integrator model. Whilst the theory in this paper has been developed for a single fault, it can be easily generalised to provide tolerance to multiple faults at the expense of computational complexity.

Future research will consider alternative ways of ensuring fault tolerance without explicitly optimising for feasible trajectories. It will also analyse the more general problem of MPC with changing objectives, of which \( M \)-step fault tolerance is a special case. The general problem allows costs to be imposed on all trajectories and can include probabilistic information on when the objective changes.

6. ACKNOWLEDGEMENTS

Simulations were performed using MATLAB, YALMIP [Löfberg, 2004], the Invariant Set Toolbox [Kerrigan, 2000] and CPLEX [2009]. This research is funded by the Menzies Foundation and the Cambridge Commonwealth Trust.

REFERENCES


