Set-Theoretic Failure Mode Reconfiguration for Stuck Actuators

Huayi Li, Ilya Kolmanovsky, and Anouck Girard

***Abstract*— This paper proposes a set-theoretic Failure Mode and Effect Management (FMEM) strategy that handles stuck/jammed actuators and enforces pointwise-in-time state and control constraints. The approach exploits nesting between constraint admissible and recoverable sets to ensure the existence of a recovery sequence. A reference governor is applied to track reference commands while imposing constraint satisfaction using the remaining working actuators. Numerical results of an aircraft longitudinal flight application are reported.**

***Index Terms*— Constrained control, fault tolerant systems, reference governor, set-theoretic methods, stuck actuators**

I. INTRODUCTION

MAINTAINING the safety of systems in applications such as aircraft and autonomous vehicles is critical. These systems often operate in constrained environments while tracking reference commands. For example, an autonomous vehicle may be given a planned trajectory to follow but it must simultaneously avoid obstacles such as other vehicles in traffic or pedestrians. Such applications may require the system to operate without constraint violation at all times, even when failures occur.

Continuing to operate without constraint violations is challenging when there are failed hardware components. It is not uncommon for hardware components, such as sensors and actuators, to fail for various reasons. In particular, failures due to stuck actuators often occur in industrial applications. Jammed elevators or rudders, for example, are one of the most common reasons for the failure of aircraft flight control, for which consequences could be fatal [1]. To mitigate failures, redundant actuators are often used (e.g., dual steering systems in automotive applications). In addition, sensors can be added for diagnostic purposes. As the software and algorithmic content for handling failure modes can be large and complex, systematic methods for Failure Mode and Effect Management (FMEM) system design are very much in need. Such FMEM strategies must be able to reconfigure system operation to maintain safety and maximize system availability.

This paper presents an approach to designing an FMEM strategy for handling actuator failures, where actuators can fail by getting stuck in a constant position. The proposed FMEM system aims to guarantee that pointwise-in-time state and control constraints are satisfied during normal operation, in failure modes, and during mode transitions.

There is much literature on the design of fault-tolerant control systems and on the analysis of the ability to reconfigure, such as [2] and [3]; usually, state and control constraints are not considered. Constraint handling using control methods similar to ours in this paper is addressed in [4]–[6], but failure modes are not considered. In [7], the case of stuck actuators is handled using set-theoretic methods; however, the reference tracking problem is not treated. The use of reference governors for fault-tolerant control is described in [8] and references therein, but these approaches are different from ours and are not combined with the reconfiguration strategy. The use of recoverable sets for safe trim point to trim point transitions is considered in [9], but the reconfiguration is not addressed to handle sequential failures that involve multiple operating mode transitions. Fault-tolerant Model Predictive Control (FTMPC) methods proposed in [10] handle constraints and reference tracking; however, [10] addresses unknown fault intensity instead of stuck/jammed actuators in this paper and does not consider sequential failures. A comprehensive comparison between the FTMPC approaches and our strategy is left to future work.

In [11], an FMEM strategy is proposed for the case when actuator failures result in the corresponding control input being set to zero. This strategy relies on manipulating the reference command to a nominal controller based on the use of constraint admissible and recoverable sets constructed using discrete-time linear models of the closed-loop system in each mode. A maximum constraint admissible set $O_{\infty,M}$ is the set of all initial states x_0 of the system and reference commands v with which the ensuing response satisfies state and control constraints for all future time instants if operating in mode M. For each failure mode, a recoverable set $R^{N_M}_{\infty,M}$ is the set of all initial states x_0 of the system for which there exists a reference command sequence $\boldsymbol{v}$ that steers the states into the state projection of $O_{\infty,M}$ within N_M steps without constraint violations. Then, if the following conditions are satisfied,

$$\text{Proj}_x\ O_{\infty,M'} \subseteq R^{N_M}_{\infty,M} \quad \forall M' \in \text{pred}(M), \tag{1}$$

where M' is the predecessor mode of M, constraints can be satisfied in each mode and during mode changes. This result is established under the assumptions of a single point of failure (i.e., one actuator failure at a time), instantaneous fault detection and isolation, and large time between subsequent failures. Furthermore, [11] introduced three mechanisms by

This research is supported by the National Science Foundation under award number ECCS-1931738.

The authors are with the Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI 48109, USA. e-mail: {huayil, ilya, anouck}@umich.edu

which (1) can be ensured: (i) by adding extra state constraints in the preceding mode; (ii) by increasing time duration allowed for the reconfiguration; and (iii) by temporarily relaxing state constraints when determining the recovery sequence. The latter mechanism is suitable for systems with soft constraints when a temporary constraint violation is permissible.

A reference governor [8] is used in [11] for reference tracking in each mode after the reconfiguration is completed and until another failure occurs. The reference governor maintains the state and modified reference in $O_{\infty,M}$ while minimizing the difference between the modified references and reference commands. When a failure occurs and the reconfiguration begins, the reference governor operation is suspended. A constrained quadratic programming problem is solved to generate a recovery sequence of modified references, which is then applied until the states enter the constraint admissible set of the current mode, and the operation of the reference governor resumes.

In this paper, the approach of [11] is extended to address the practically important case when actuators can get stuck/jammed at a constant position. This requires modifying control laws in each mode to compensate for failed actuator positions, as well as re-formulating the constraint admissible sets, recoverable sets, and reconfiguration conditions to include the dependence on the failed actuator positions. The case treated in this paper is significantly more general than that in [11] where inputs corresponding to failed actuators were set to zero.

The proposed FMEM strategy (Figure 1), upon failure, first modifies the reference command by generating a recovery sequence, and then by using a reference governor once the state enters the maximum constraint admissible set (Figure 2). The controller responding to the modified reference command is also switched for each specific mode.

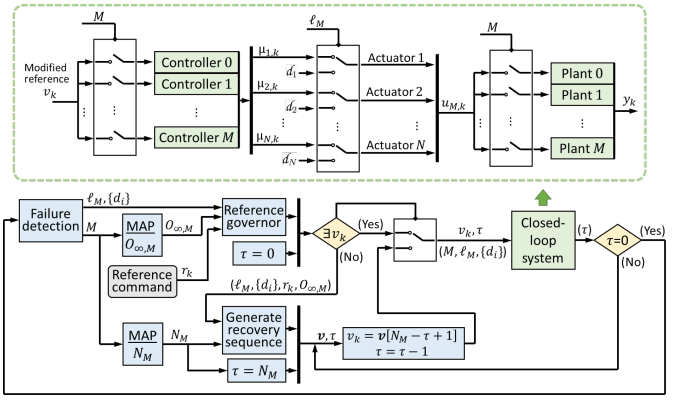

Fig. 1. A flowchart of the proposed FMEM strategy. Signals in round brackets are passed from previous blocks.

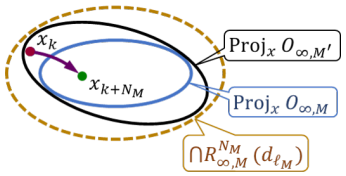

Fig. 2. The relation between offline designed constraint admissible and recoverable sets exploited in the FMEM strategy.

The rest of the paper is organized as follows. First, normal and failure modes, open and closed-loop discrete-time system models, as well as constraints are introduced in Section II. Section III defines the problem statement, constraint admissible and recoverable sets, reconfiguration conditions for different mode transitions, and reference governors for reference tracking. Numerical results for an aircraft longitudinal flight application are reported in Section IV. Finally, conclusions are drawn in Section V.

II. Operation Modes, System Dynamics, and Constraints

A. Normal and failure modes

Consider an over-actuated system with N actuators. Each actuator may fail because it gets stuck/jammed and generates a constant input equal to the value immediately before the failure happens. Failures of multiple actuators are possible, but only one actuator can fail at a time. The time between subsequent failures is assumed to be large. Furthermore, to simplify the exposition, we assume that the failure is detected instantaneously and the failed actuator position is accurately measured/estimated, while in practice these assumptions can be relaxed.

Figure 3 shows an example of potential failure paths for the case of $N = 2$ in the worst case scenario when every combination of actuator failures is possible. Each box represents a potential operating mode of the system. The numbers inside the box are labels of the actuators that are still working. Each mode is denoted by $M \in \{0, 1, \cdots, 2^N\}$, and a vector ℓ_M is defined to label each mode by

$$\ell_M = \begin{bmatrix} m_1 & m_2 & \cdots & m_N \end{bmatrix}^T,$$
$$m_i = \begin{cases} 0 & \text{if the } i^{th} \text{ actuator failed,} \\ 1 & \text{if the } i^{th} \text{ actuator works,} \end{cases}$$

for $i \in \{1, \cdots, N\}$. In particular,

$$\ell_0 = \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}^T,\ \ell_\Omega = \begin{bmatrix} 0 & 0 & \cdots & 0 \end{bmatrix}^T,$$

where ℓ_0 is the label of the normal mode when all actuators work, and ℓ_Ω with $\Omega = 2^N$ labels the mode when all actuators fail.

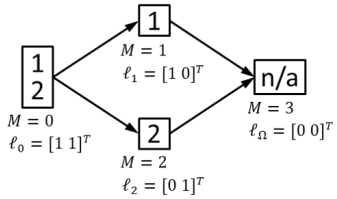

Fig. 3. Example of failure paths, modes M, and labels ℓ_M for $N = 2$.

B. Open-loop and closed-loop system models

We assume that the system in mode M is represented by the following discrete-time linear model:

$$x_{k+1} = Ax_k + Bu_{M,k}, \tag{2}$$
$$y_k = Cx_k, \tag{3}$$

where x_k is the state, y_k is the output, and $u_{M,k}$ is the input. We have

$$u_{M,k} = \ell_M \odot u_{0,k} + (\ell_0 - \ell_M) \odot u_{\Omega,k} \ \forall 0 \le M \le \Omega,$$

where

$$u_{0,k} = \begin{bmatrix} \mu_{1,k} & \mu_{2,k} & \cdots & \mu_{N,k} \end{bmatrix}^T,$$
$$u_{\Omega,k} = \begin{bmatrix} \mathscr{d}_1 & \mathscr{d}_2 & \cdots & \mathscr{d}_N \end{bmatrix}^T,$$

and where $\mu_{i,k}$ is the input in the i^{th} channel where the corresponding actuator is working properly, $\mathscr{d}_i$ is the constant input in the i^{th} channel where the corresponding actuator failed and got stuck, and $\odot$ is the element-wise product. For example, in the case with three actuators, if $\ell_1 = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}^T$, we have $u_{1,k} = \begin{bmatrix} \mu_{1,k} & \mu_{2,k} & \mathscr{d}_3 \end{bmatrix}^T$.

In the normal mode ($M = 0$), a stabilizing feedback plus feedforward controller is used, of the form,

$$u_{0,k} = K_0 x_k + G_0 v_k, \tag{4}$$

where K_0 is a stabilizing feedback gain, G_0 is a feedforward gain such that the tracking error in steady state is equal to zero, and v_k is the reference command.

In failure modes M, $0 < M < \Omega$, when there are actuators stuck at constant values, we first define $\mu_{\ell_M,k}$ as the vector of inputs of the working actuators and d_{ℓ_M} as the vector of constant inputs of the failed actuators by reducing $u_{M,k}$ to

$$\mu_{\ell_M,k} = \diamondsuit(\ell_M, u_{M,k}), \quad d_{\ell_M} = \diamondsuit(\ell_0 - \ell_M, u_{M,k}),$$

where $\diamondsuit(\ell, E)$ is an operator that reduces the dimension of the vector, matrix, or product of sets E by removing the $i^{th} \in \{1, \cdots, N\}$ element, row, or set if the corresponding i^{th} element in ℓ is zero.

It is assumed that in all failure modes the number of references is less than or equal to the number of working actuators. Then for each mode M, a stabilizing controller is assumed to have been designed, and given by

$$\mu_{\ell_M,k} = K_M x_k + G_M v_k + H_M d_{\ell_M}, \tag{5}$$

where K_M is the stabilizing feedback gain and G_M and H_M are the feedforward gains for the reference commands and stuck inputs. They are designed so that the system has zero steady-state tracking error.

In general, the closed-loop dynamics are represented by

$$x_{k+1} = \bar{A}_M x_k + \bar{B}_M U_{M,k}, \tag{6}$$

where $U_{M,k}$ is the closed-loop input, and

- for $M = 0$,

$$\bar{A}_0 = A + BK_0, \ \bar{B}_0 = BG_0, \ U_{0,k} = v_k,$$

- for $0 < M < \Omega$,

$$\bar{A}_M = A + B_{M,\mu} K_M,$$
$$\bar{B}_M = \begin{bmatrix} B_{M,\mu} G_M & B_{M,\mu} H_M + B_{M,d} \end{bmatrix},$$
$$U_{M,k} = \begin{bmatrix} v_k \\ d_{\ell_M} \end{bmatrix},$$

where

$$B_{M,\mu} = (\diamondsuit(\ell_M, B^T))^T, \ B_{M,d} = (\diamondsuit(\ell_0 - \ell_M, B^T))^T.$$

In the failure mode when all actuators fail and give constant inputs ($M = \Omega$), the system runs in open-loop. In this case we assume that the open-loop system is stable as otherwise state constraints cannot be handled. For consistency of notations, we let

$$\bar{A}_\Omega = A, \ \bar{B}_\Omega = B, \ U_{\Omega,k} = u_{\Omega,k}.$$

C. Constraints

To ensure safe operation, pointwise-in-time state and control constraints are imposed given by inequalities of the form,

$$x_k \in X_M^*(U_{M,k}) = \{x : \ \mathscr{A}_M^* \begin{bmatrix} x \\ U_{M,k} \end{bmatrix} \le \mathscr{b}_M^*\}. \tag{7}$$

It is assumed that individual open-loop input range constraints of the form $u_{M,k} \in D_1 \times \cdots \times D_N$ are reflected in (7) where D_i, $i = 1, \cdots, N$, is the feasible input range for the i^{th} actuator.

Modifications of constraints may be needed to facilitate the sequential failure mode reconfiguration design. Firstly, to satisfy subsequent conditions for the safe reconfiguration, it may be necessary to restrict the operation in preceding modes by artificially tightening constraints (7) to

$$\begin{aligned} x_k \in X_M(U_{M,k}) = \ & X_M^*(U_{M,k}) \cap \bar{X}_M(U_{M,k}) \\ = \ & \{x : \ \mathscr{A}_M \begin{bmatrix} x \\ U_{M,k} \end{bmatrix} \le \mathscr{b}_M\}, \end{aligned} \tag{8}$$

where the sets $\bar{X}_M(U_{M,k})$ need to be appropriately designed.

Secondly, in practical applications, some of the state constraints could be imposed conservatively to extend the system operating life and can be relaxed temporarily during the recovery to, for example, reduce the number of steps needed for the recovery. Thus, during the short period when the recovery sequence is applied, the constraints can be relaxed to

$$x_k \in X_{R,M}(U_{M,k}) = \ \{x : \ \mathscr{A}_{R,M} \begin{bmatrix} x \\ U_{M,k} \end{bmatrix} \le \mathscr{b}_{R,M}\}, \tag{9}$$

where $X_{R,M}(U_{M,k}) \supseteq X_M(U_{M,k})$, and $X_{R,M}(U_{M,k})$ also need to be appropriately designed.

III. Reconfiguration Strategy

The FMEM strategy has two basic goals. Firstly, constraints should not be violated at any time so that safety is preserved. The system needs to operate safely in all modes and during mode transitions. Secondly, the system output should follow the given reference command as closely as possible. The proposed strategy exploits constraint admissible sets, recoverable sets, and reference governors.

A. Constraint admissible sets

For $M \in \{0, \cdots, \Omega\}$, the maximum constraint admissible sets are defined by

$$\begin{aligned} O_{\infty,M} = & \{(U_M, x_0) : \ x_t \in X_M(U_M) \ \forall t \in \mathbb{Z}_{\ge 0}, \\ & x_{M,\mathrm{ss}}(U_M) \oplus \mathcal{B}_\epsilon \subset X_M(U_M)\} \\ = & \{(U_M, x_0) : \ \mathscr{A}_{O_{\infty,M}} x_0 \le \mathscr{b}_{O_{\infty,M}}(U_M)\}, \end{aligned} \tag{10}$$

where x_t is the response of (6) to the initial condition x_0 and constant closed-loop input U_M, $x_{M,\mathrm{ss}}(U_M)$ is the steady-state point given by

$$x_{M,\mathrm{ss}}(U_M) = (I - \bar{A}_M)^{-1}\bar{B}_M U_M,$$

and $\mathcal{B}_\epsilon$ is an open ball of radius $\epsilon > 0$. By adding the constraint $x_{M,\mathrm{ss}}(U_M) \oplus \mathcal{B}_\epsilon \subset X_M(U_M)$, one can ensure that, under mild additional assumptions, the set $O_{\infty,M}$ is positively invariant, finitely determined, and can be represented by a finite set of affine inequalities [12]. Positive invariance means that if $(U_M, x_0) \in O_{\infty,M}$, then $(U_M, x_t) \in O_{\infty,M}$ for all future time instants $t \geq 0$ as long as the mode remains equal to M. Being finitely determined means that there exists $T \in \mathbb{Z}_{\geq 0}$ such that $x_t \in X_M(U_M)\ \forall t \leq T$ is equivalent to $(U_M, x_0) \in O_{\infty,M}$. As a result, it is possible to represent $O_{\infty,M}$ by a finite set of inequalities as in (10) (see, e.g., [9] for derivation).

B. Recoverable sets

The recoverable sets for $M \in \{1, \cdots, \Omega - 1\}$ are defined as

$$\begin{aligned} R^{N_M}_{\infty,M}(d_{\ell_M}) = \{x_0 :\ & \exists \{v_0, \cdots, v_{N_M}\} \text{ such that} \\ & x_t \in X_{R,M}(U_{M,t})\ \forall t = 0, 1, \cdots, N_M - 1, \\ & (U_{M,N_M}, x_{N_M}) \in O_{\infty,M}\}, \end{aligned} \tag{11}$$

where N_M is the number of steps allowed for the reconfiguration, which is a design parameter. We let $\boldsymbol{v} = \begin{bmatrix} v_0 & \cdots & v_{N_M} \end{bmatrix}^T$ designate the reference sequence.

In mode Ω, when all actuators fail, the system runs open-loop, so it needs to be already running inside $O_{\infty,\Omega}$ in the immediate preceding modes. To keep the consistency of notations, we let

$$R^{N_\Omega}_{\infty,\Omega} = \mathrm{Proj}_x\ O_{\infty,\Omega} \quad \forall N_\Omega \geq 0, \tag{12}$$

where

$$\mathrm{Proj}_x\ O_{\infty,M} = \{x_0 :\ \exists U_M \text{ such that } (U_M, x_0) \in O_{\infty,M}\}.$$

C. Reconfiguration conditions

The safe reconfiguration conditions that follow are based on an extension of (1). If these conditions are satisfied, upon failure, the states of the system are in the state projection of the constraint admissible set of the predecessor mode, which is a subset of the recoverable set of the successor mode. Therefore, the states are guaranteed to be inside the recoverable set of the successor mode, and there exists a feasible sequence of references for the reconfiguration.

For transitions to mode $M \in \{1, \cdots, \Omega - 1\}$ from mode $M' \in \mathrm{pred}(M)$, where $\mathrm{pred}(M)$ is the set of all predecessor modes that can change to the successor mode M due to a single actuator failure, it suffices to require that

$$\mathrm{Proj}_x\ O_{\infty,M'} \subseteq \bigcap_{d_{\ell_M} \in D_{\ell_M}} R^{N_M}_{\infty,M}(d_{\ell_M}), \tag{13}$$

where

$$D_{\ell_M} = \Diamond(\ell_0 - \ell_M, D_1 \times \cdots \times D_N)$$

is the set of control constraints for d_{ℓ_M}.

For transitions to mode Ω from $M' \in \mathrm{pred}(\Omega)$, since none of the actuators work, we have to ensure that

$$O_{\infty,M'} \subseteq O_{\infty,\Omega}, \tag{14}$$

for all $M' \in \mathrm{pred}(\Omega)$.

D. Safe reconfiguration upon failure detection

At the beginning of the reconfiguration and mode transition to mode $M \in \{1, \cdots, \Omega - 1\}$, the recovery reference sequence $\boldsymbol{v}$ is computed by solving the following quadratic programming (QP) problem

$$\begin{aligned} &\text{Minimize } \|r_k \mathbf{1} - \boldsymbol{v}\|^2 \\ &\text{subject to } x_t \in X_{R,M}(U_{M,t})\ \forall t = k+1, \cdots, k + N_M - 1, \\ &\qquad (U_{M,k+N_M}, x_{k+N_M}) \in O_{\infty,M}, \end{aligned} \tag{15}$$

where r_k equals to the given reference command at the beginning of the transition (at time step k). Then the system runs for N_M steps using the recovery sequence of modified references $\boldsymbol{v}$ until the end of the reconfiguration.

E. Reference governor

A reference governor (RG) is used for reference tracking after the reconfiguration is completed. With the system running in mode $M \in \{0, \cdots, \Omega - 1\}$, the reference governor determines the modified reference based on the solution of the following optimization problem

$$\begin{aligned} &\text{Minimize } \|r_k - v_k\|^2 \\ &\text{subject to } (U_{M,k},\ x_k) \in O_{\infty,M}. \end{aligned} \tag{16}$$

The modified reference v_k is then updated at every time step. By construction, the following result is obtained:

Theorem: For system operation in arbitrary mode $M \in \{1, \cdots, \Omega - 1\}$, if (13) is satisfied, the time between failures is larger than N_M, and the safe reconfiguration strategy and reference governor based on (15) and (16) are used, then constraints given by (9) and (8) are satisfied respectively during the reconfiguration and after the recovery is completed.

Remark 1: Note that if (13) is satisfied, the system can operate safely not only when the input is stuck at a constant value immediately preceding the failure, but even if it instantaneously jumps to another constant value at the time of failure as long as $d_{\ell_M} \in D_{\ell_M}$, such as for the zero control considered in [11] where (13) coincides with (1) by having $d_{\ell_M} \in D_{\ell_M} = \{\mathbf{0}\}$ and $R^{N_M}_{\infty,M} = R^{N_M}_{\infty,M}(d_{\ell_M})$.

Remark 2: The FMEM strategy is developed offline and involves choosing variables N_M, $X_M(U_{M,k})$, and $X_{R,M}(U_{M,k})$ to satisfy (13) as in Figure 2 for each mode transition. Then during the online operation (Figure 1), the modified reference command is generated by (15) or (16). Note that the online chronometric load would not regularly increase with the increase in the number of redundant actuators or possible failure paths as the underlying QP problems are only relevant to the current mode. However, the read-only memory (ROM) size necessary to store sets for different modes can grow.

IV. Application to Aircraft Longitudinal Flight Control

A. System dynamics

1) Open-loop model: The aircraft model represents a Boeing 747-100 aircraft in steady level flight corresponding to Mach 0.5 cruise at 20,000 feet [13]. The linearized longitudinal flight dynamics under the normal operating conditions with the classical phugoid approximation are modeled by

$$\Delta\dot{u} = -0.0075\Delta u - 0.1713\Delta\theta + \Delta a_T - 0.0051\Delta\delta_e,$$
$$\Delta\dot{\theta} = 0.0436\Delta u - 0.0423\Delta\delta_e,$$

and the output is

$$\Delta\dot{h} = 2.7645\Delta\theta,$$

where Δ denotes the deviation from the trim value, u is the projection of the velocity vector on the x-axis of the body-fixed frame in m/s, θ is the pitch angle in $^\circ$, a_T is the thrust-to-mass ratio in N/kg, δ_e is the elevator deflection in $^\circ$, and $\dot{h}$ is the climb rate in m/s.

The system can be written compactly in the form,

$$\dot{x} = Ax + Bu, \quad y = Cx, \tag{17}$$

where $x = [\Delta u \;\; \Delta\theta]^T$ is the state vector, $u = [\Delta a_T \;\; \Delta\delta_e]^T$ is the input vector, and $y = \Delta\dot{h}$ is the output. This model is converted to discrete-time assuming a 5-second update period. Note that this system has two redundant actuators, hence $N = 2$ as in Figure 3.

2) Closed-loop model: The controllers (4) for $M = 0$ and (5) for $M \in \{1, 2\}$ are designed using Linear Quadratic Regulator (LQR) theory with the *dlqr* command in `Matlab` to obtain the feedback gain K_M. The feedforward gains G_M and H_M are computed so that the steady-state gain from the reference v to the output y for the climb rate deviation is equal to 1. The weights Q_M and R_M are chosen by trial and error and assuming that the use of thrust is more expensive than the use of the elevator. Their values and those of the resulting G_M and H_M matrices are as follows:

- In Mode 0, the controller is designed to track a given climb rate deviation from nominal and a zero deviation of the velocity magnitude from nominal, that is, to hold u, using

$$Q_0 = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}, \; R_0 = \begin{bmatrix}1000 & 0\\ 0 & 250\end{bmatrix},$$
$$G_0^* = \left(\begin{bmatrix}0 & 2.7645\\ 1 & 0\end{bmatrix}(I - \bar{A}_M)^{-1}B\right)^{-1}\begin{bmatrix}1 & 0\\ 0 & 0\end{bmatrix},$$
$$G_0 = (\Diamond([1\;0]^T, G_0^{*T}))^T.$$

- In Modes 1 and 2 (for $M \in \{1, 2\}$), the controller only tracks a given climb rate deviation reference, with

$$Q_1 = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}, \; R_1 = \begin{bmatrix}10 & 0\\ 0 & 0\end{bmatrix},$$
$$Q_2 = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}, \; R_2 = \begin{bmatrix}0 & 0\\ 0 & 2.5\end{bmatrix},$$
$$G_M = [C(I - \bar{A}_M)^{-1}B_{M,\mu}]^{-1},$$
$$H_M = -G_M[C(I - \bar{A}_M)^{-1}B_{M,d}]^{-1}.$$

The closed-loop systems are then obtained based on (6).

B. Constraints

Constraints are imposed on the climb rate, the thrust-to-mass ratio, and the elevator deflection as

$$|\Delta\dot{h}| \le y_{\max}, \;\; |\Delta a_T| \le u_{1_{\max}}, \text{ and } |\Delta\delta_e| \le u_{2_{\max}}, \tag{18}$$

where $y_{\max} = 20\ m/s$, $u_{1_{\max}} = 2\ N/kg$, and $u_{2_{\max}} = 45^\circ$. These constraints define $X_M^*(U_{M,k})$ given by (7) for all modes of $M \in \{0, 1, 2, 3\}$.

C. Design of constraint admissible and recoverable sets

1) Constraint admissible sets: The design begins from Mode 3. During simulation, constraints for the failed inputs d_1 and d_2 by (18) are automatically satisfied due to the same constraints being applied to the working inputs $\mu_{1,k}$ and $\mu_{2,k}$ in preceding modes. However, the input constraints are still used to define $X_3^*(U_{3,k})$ so that $O_{\infty,3}$ is a bounded set. Since Mode 3 is the last mode of the failure sequence, no subsequent conditions need to be considered, so for (8), let $X_3(U_{3,k}) = X_3^*(U_{3,k})$. Then, the constraint admissible set of Mode 3, as defined in (10), can be represented by

$$O_{\infty,3} = \{(d_1, d_2, x_0) : \mathcal{A}_{O_3,d_1}d_1 + \mathcal{A}_{O_3,d_2}d_2 + \mathcal{A}_{O_3,x_0}x_0 \le b_{O_3}\}. \tag{19}$$

Next, in Modes 1 and 2 ($M \in \{1, 2\}$), in order to satisfy the condition (14) for transitions to Mode 3, the state constraints can be tightened by imposing constraints from $O_{\infty,3}$, that is, to have

$$x_0 \in \{x_0 : \mathcal{A}_{O_3,d_M}\mu_{M,k} + \mathcal{A}_{O_3,d_{\ell_M}}d_{\ell_M} + \mathcal{A}_{O_3,x_0}x_0 \le b_{O_3}\}, \tag{20}$$

where $\mu_{M,k}$ is generated by the controller in (5), as a function of $U_{M,k}$. By substituting $\mu_{M,k}$ in (20) by (5), we define

$$\bar{X}_M(U_{M,k}) = \{x_0 : \mathcal{A}_{O_3,d_M}G_M v_k + (\mathcal{A}_{O_3,d_M}H_M + \mathcal{A}_{O_3,d_{\ell_M}})d_{\ell_M} + (\mathcal{A}_{O_3,d_M}K_M + \mathcal{A}_{O_3,x_0})x_0 \le b_{O_3}\}. \tag{21}$$

Then, $X_M(U_{M,k})$ and $O_{\infty,M}$ are defined based on (8) and (10).

Finally, to satisfy the recovery conditions of (13) for mode transitions between Modes 0 and 1 and 0 and 2, the constraints are tightened by a scaling coefficient of $\eta_O \in (0, 1]$ that needs to be tuned (beginning from 1 and decreasing until the recovery conditions are satisfied). Then, $X_0(U_{0,k})$ is defined by these tightened constraints as

$$X_0(U_{0,k}) = \{x_0 : \; \mathcal{A}_0^*\begin{bmatrix}x\\ U_{0,k}\end{bmatrix} \le \eta_O b_0^*\}, \tag{22}$$

and $O_{\infty,0}$ is defined based on (10).

2) Recoverable sets: The recoverable sets for Modes 1 and 2 ($M \in \{1, 2\}$) are designed based on (13).

The state constraints during recovery are temporarily relaxed by having

$$X_{R,M}(U_{M,k}) = \{x : \; \mathcal{A}_{M,X}\begin{bmatrix}x\\ U_{M,k}\end{bmatrix} \le \eta_{R,M} b_{M,X}\}, \tag{23}$$

where $\eta_{R,M} \ge 1$ is a design parameter. Then, the recoverable sets are constructed based on (11).

3) Choosing design parameters to satisfy recovery conditions: The final tuning results are $\eta_O = 0.7$, $N_1 = 5$, $\eta_{R,1} = 1.9$, $N_2 = 5$, and $\eta_{R,2} = 1$. Conditions (13) are checked using the toolbox `Bensolve` [14]. Figure 4 shows the state projections of the $O_{\infty,M}$ sets for all modes, compared with the intersection of sets $R^{N_M}_{\infty,M}(d_{\ell_M})$ for all $d_{\ell_M} \in D_{\ell_M}$ for $M = 1$ and 2. It can be seen that the reconfiguration conditions (13) are satisfied. By checking the set relations of $O_{\infty,1}$ and $O_{\infty,2}$ versus $O_{\infty,3}$ using the `Bensolve` command *le*, it has been confirmed that the conditions (14) are also satisfied.

D. Simulation results

Figure 4 shows the state trajectories and time-based trajectories of the major signals of a simulation with mode switching from 0 to 2 to 3 at times 0 and 225. In order to demonstrate the reconfiguration process, an initial condition is picked such that $([v_0 \ \mu_{1,0}]^T, x_0) \notin O_{\infty,2}$ but $x_0 \in R^{N_2}_{\infty,2}(\mu_{1,0})$, and the simulation starts at the transition from Mode 0 to Mode 2.

The system runs from 0 to 25 seconds with the modified reference sequence $\{v_t\}_0^{N_2}$ generated by the recovery sequence generator. The thrust is stuck while the elevator is working normally. At 25 seconds, the state is steered inside $O_{\infty,2}$, so after that, the reference governor updates the reference at every time step, until 225 seconds when the mode switches from 2 to 3, in which both actuators fail. Since $x_0 \in O_{\infty,2}$ prior to the transition, the system continues to operate without constraint violation in open-loop after it switches to Mode 3.

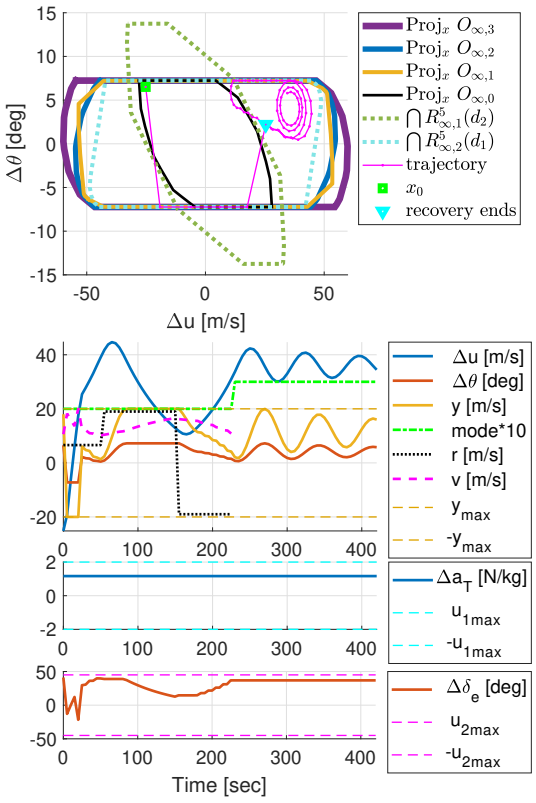

Fig. 4. Aircraft example simulation results (initial states are picked such that recovery is needed).

V. Concluding Remarks

Failure Mode and Effect Management (FMEM) systems need to be properly designed to avoid safety constraint violations when failures occur. Redundant actuators can be exploited to improve system reliability, but failures could happen due to one or multiple actuators being stuck. A set-theoretic based strategy for system reconfiguration was presented, using constraint admissible and recoverable sets together with the use of a reference governor that enables tracking of references. The strategy considers sequential transitions of normal and failure modes, guaranteeing safe operations in all modes as well as during mode transitions. Mechanisms to help satisfy the reconfiguration conditions were illustrated through a numerical example of an aircraft longitudinal flight control application. By requiring that the recovery sequence be simultaneously feasible for multiple modes, the present approach can be extended to the case when failure detection and isolation are not instantaneous. In such a setting, generating a recovery control sequence directly rather than a modified reference command sequence could be more straightforward.

References

[1] H. Williamson, *Air Crash Investigations: Jammed Rudder Kills 132, The Crash Of USAir Flight 427*. lulu.com, October 2011.

[2] M. Blanke, M. Kinnaert, J. Lunze, M. Staroswiecki, and J. Schröder, *Diagnosis and Fault-Tolerant Control*, vol. 2. Springer, 2006.

[3] W. Chen and J. Jiang, "Fault-tolerant control against stuck actuator faults," *IEE Proceedings-Control Theory and Applications*, vol. 152, no. 2, pp. 138–146, 2005.

[4] F. Blanchini and S. Miani, *Set-Theoretic Methods in Control*. Springer, 2008.

[5] C. Danielson, K. Berntorp, A. Weiss, and S. D. Cairano, "Robust motion planning for uncertain systems with disturbances using the invariant-set motion planner," *IEEE Transactions on Automatic Control*, vol. 65, no. 10, pp. 4456–4463, 2020.

[6] F. Blanchini, F. A. Pellegrino, and L. Visentini, "Control of manipulators in a constrained workspace by means of linked invariant sets," *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, vol. 14, no. 13-14, pp. 1185–1205, 2004.

[7] W. Lucia, D. Famularo, and G. Franze, "A set-theoretic reconfiguration feedback control scheme against simultaneous stuck actuators," *IEEE Transactions on Automatic Control*, vol. 63, no. 8, pp. 2558–2565, 2017.

[8] E. Garone, S. Di Cairano, and I. Kolmanovsky, "Reference and command governors for systems with constraints: A survey on theory and applications," *Automatica*, vol. 75, pp. 306–328, 2017.

[9] K. McDonough and I. Kolmanovsky, "Fast computable recoverable sets and their use for aircraft loss-of-control handling," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 4, pp. 934–947, 2017.

[10] A. Yetendje, M. M. Seron, and J. A. De Doná, "Robust MPC multicontroller design for actuator fault tolerance of constrained systems," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 4678–4683, 2011.

[11] H. Li, I. Kolmanovsky, and A. Girard, "A failure mode reconfiguration strategy based on constraint admissible and recoverable sets," in *2021 American Control Conference (ACC)*, pp. 4759–4764, IEEE, 2021.

[12] I. Kolmanovsky and E. G. Gilbert, "Theory and computation of disturbance invariant sets for discrete-time linear systems," *Mathematical Problems in Engineering*, vol. 4, pp. 317–367, 1998.

[13] A. Girard and I. Kolmanovsky, *Lecture Notes on Control of Aerospace Vehicles*. Department of Aerospace Engineering, The University of Michigan, Ann Arbor, January 2019.

[14] A. Löhne and B. Weißing, "The vector linear program solver bensolve–notes on theoretical background," *European Journal of Operational Research*, vol. 260, no. 3, pp. 807–813, 2017.