Fault-tolerant reference generation for model predictive control with active diagnosis of elevator jamming faults

文章考虑了两种故障：

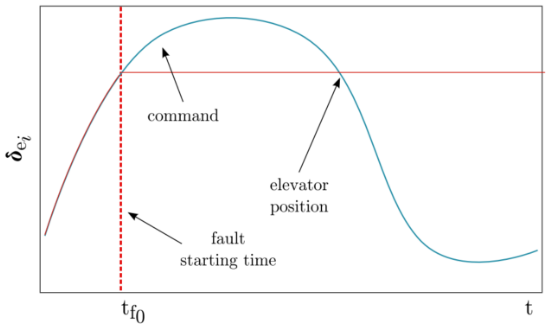

FIGURE 1 Stuck fault [Colour figure can be viewed at wileyonlinelibrary.com]

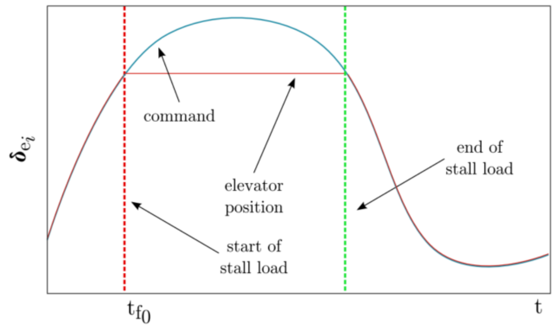

FIGURE 2 Stall load [Colour figure can be viewed at wileyonlinelibrary.com]

设计了对于这两种故障的辨识方法，容错方法只有一种，不针对故障区分。

容错MPC：

$$\mathcal{V}^*(v, x_{\text{init}}) := \underset{x,u,\theta}{\text{minimize}} \sum_{t=0}^{N} l_t(v, x_t, u_t, \theta_t) \tag{7a}$$

$$\text{subject to: } Ax_t + Bu_t = x_{t+1}, \quad t = 0, \ldots, N, \tag{7b}$$

$$\begin{bmatrix} \hat{x}_t \\ \hat{u}_t \end{bmatrix} = M_\theta \theta_t, \quad t = 0, \ldots, N, \tag{7c}$$

$$G_x x_t + G_u u_t + g \leq 0 \quad t = 0, \ldots, N, \tag{7d}$$

$$G_x \hat{x}_t + G_u \hat{u}_t + g_\theta + E\varepsilon_{\text{nl}} \leq 0 \quad t = 0, \ldots, N, \tag{7e}$$

$$\hat{y}_t = N_\theta \theta_t \quad t = 0, \ldots, N, \tag{7f}$$

$$x_0 := x_{\text{init}}, \tag{7g}$$

where $x_t \in \mathbb{R}^n$ and $u_t \in \mathbb{R}^{n_u}$ indicate the t-step-ahead state and control predictions, respectively. In addition, (7d) represents the constraints on the predicted state, input, and output ($G_x \in \mathbb{R}^{c \times n}$, $G_u \in \mathbb{R}^{c \times n_u}$, and $g_\theta = g$ in fault-free operating conditions) that follow from the definition of $\mathcal{X}_{\text{MPC}}$, $\mathcal{U}$, and $\mathcal{Y}_{\text{MPC}}$. Furthermore, $\theta_t \in \mathbb{R}^{n_u}$ is the vector of parameters used to generate the *artificial* steady state, input, and output $\hat{x}_t$, $\hat{u}_t$, and $\hat{y}_t$, respectively. M_θ and N_θ are suitable matrices (refer to the work of Limón et al[23] for details). For a prediction horizon of length N, the cost l_t in (7a) is described as follows:

$$l_t(v, x_t, u_t, \theta_t) := \|x_t - \hat{x}_t\|_Q^2 + \|u_t - \hat{u}_t\|_R^2 + \rho_1 \|\hat{y}_t - v\|_2^2, \tag{8}$$

与 Limon 的方法相同：

$$\begin{aligned}
V_N^*(x, \hat{\mathrm{x}}_s) = \; & \min_{\mathbf{u},\theta} V_N(x, \hat{\mathrm{x}}_s, \mathbf{u}, \theta) \\
& s.t. \;\; x(0) = x, \\
& \qquad x(j+1) = Ax(j) + Bu(j), \\
& \qquad (x(j), u(j)) \in Z, \quad j = 0, \ldots, N-1 \\
& \qquad (\mathrm{x}_s, \mathrm{u}_s) = M_\theta \theta, \\
& \qquad (x(N), \theta) \in \mathcal{X}_f^w.
\end{aligned}$$

文章中的约束 (7 e) 对应 Limon 的 $(x(N), \theta) \in X_f^w$ 。

An Active Fault-Tolerant MPC for Systems with Partial Actuator Failures

An active fault-tolerant MPC for systems with partial actuator failures

Publisher: IEEE Cite This PDF

Haokun Wang; Aipeng Jiang All Authors

线性系统：

$$\begin{cases} x_{k+1} = Ax_k + Bu_k + w_k \\ y_k = Cx_k + v_k \end{cases} \tag{1}$$

有三种执行器故障：partial actuator failure, the actuator outage and the actuator stuck（部分执行器故障，断电，卡死）。

文章考虑第一种故障：

$$n_k = \begin{bmatrix} f_{1,k} & 0 & \cdots & 0 \\ 0 & f_{2,k} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & f_{n_u,k} \end{bmatrix} u_k = F_k u_k, \tag{2}$$

where F_k is a diagonal matrix, and $0 < f_{i,k} \leq 1, (i = 1, 2, \cdots, n_u)$. Considering actuator faults are often uncertain in time, value, and pattern, here we assume that F_k is not known.

This paper aims to design an active fault-tolerant MPC for system (1) with unknown actuator faults (2) to track a specified piece-wised reference r without steady-state tracking error (offset-free control).

F_k 是不知道的。

对于故障后系统：

$$\begin{cases} x_{k+1} = Ax_k + BF_k u_k + w_k \\ y_k = Cx_k + v_k \end{cases}. \tag{3}$$

执行器输入 u_k 和故障 F_k 耦合，需要将其分开，以观测器的形式辨识故障的数值。

Since F_k is unknown, it is difficult to design a controller directly according to model (3). If we define a new variable d_k as

$$d_k = (F_k - I)u_k \tag{4}$$

Model (3) can then be expressed as

$$\begin{cases} x_{k+1} = Ax_k + Bu_k + Bd_k + w_k \\ y_k = Cx_k + v_k \end{cases} \tag{5}$$

设计最小方差观测器：

$$\begin{cases} \hat{x}_{k|k-1} = A\hat{x}_{k-1} + Bu_{k-1} \\ \hat{d}_{k-1} = L_d(y_k - C\hat{x}_{k|k-1}) \\ \hat{x}_k^* = \hat{x}_{k|k-1} + B\hat{d}_{k-1} \\ \hat{x}_k = \hat{x}_k^* + L_x(y_k - C\hat{x}_k^*) \end{cases}. \tag{6}$$

To obtain an unbiased estimate of d_{k-1}, the condition

$$L_dCB = I \tag{7}$$

文章的MPC使用两层控制。

第一层：

$$\begin{aligned} \min_{x_s,u_s} \quad & (u_s - u_t)^T R_s (u_s - u_t) \\ \text{s.t.} \quad & \begin{cases} \begin{bmatrix} I-A & -B \\ C & 0 \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} B\hat{d}_k \\ r_k \end{bmatrix}, \\ x_s \in \Omega_x, \\ u_s \in \Omega_u. \end{cases} \end{aligned} \tag{9}$$

如果无法满足等式约束，可以使用：

$$\begin{aligned} \min_{x_s,u_s} \quad & (r_k - Cx_s)^T Q_s (r_k - Cx_s) \\ \text{s.t.} \quad & \begin{cases} \begin{bmatrix} I-A & -B \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = B\hat{d}_k, \\ x_s \in \Omega_x, \\ u_s \in \Omega_u. \end{cases} \end{aligned} \tag{10}$$

第二层:

$$
\begin{aligned}
&\min_{\boldsymbol{U}_P} \sum_{i=k}^{k+P-1} \left(\|\hat{x}_{i+1} - x_s\|_{W_x}^2 + \|u_i - u_s\|_{W_u}^2 \right) \\
&\text{s.t.} \begin{cases} \hat{x}_{i+1} = A_k \hat{x}_i + B_k u_i + B_k \hat{d}_i \\ u_i \in \Omega_u \end{cases}
\end{aligned}
\tag{11}
$$

这篇文章的优点是做了一整套工作：从故障诊断到容错控制，考虑的是执行器效率降低的故障，能够实现故障情况下的无偏跟踪。

这种方法没有优化故障情况下的动态性能。

Reconfigurable Fault Tolerant Flight Control based on Nonlinear ModelPredictive Control

NPMC 与LMPC的性能对比：`In [8], NMPC is shown to providebetter performance compared to a Linear Parameter Varying(LPV) control approach for the F-16 model used in the workpresented here.`

The results for three test cases are presented. Case *1* assumes the existence of an FDD system that updates the NMPC autopilot with new control surface position limits or rate limits when faults occur (according to the FDD scheme in [17]), but it does not implement any external reconfiguration mechanism. Case *2* assumes no FDD system exists, and therefore simulates the case where the control system relies on only the inherent fault detection and reconfigurable capabilities of NMPC (enabled through the effective use of measured actuator position feedback). Case *3* assumes the existence of an FDD system and, in addition, employs the following simple reconfiguration mechanism.

1) *for each actuator*, scan for locked/jammed actuator by checking for changes in lower and upper limits (worst case $\delta_{min} = \delta_{max}$)
2) if an actuator is locked, decrease the weight of all the 'healthy' actuators in the same channel (longitudinal or lateral), by a predefined factor.
3) finally, reduce the weight on any secondary objectives that share the same remaining 'healthy' actuators (that are capable of achieving similar effects as the jammed actuator), by a predefined factor.

- case 1：根据 FDD 的信息更新约束，不重构。
- case 2：没有 FDD，根据内环的故障检测和重构能力
- case 3：根据 FDD 进行重构。