

# 复杂动态系统的故障检测与诊断

葛 彤 邓建华

(西北工业大学飞机系, 西安, 710072)

## COMPLEX DYNAMIC SYSTEM FAILURE DETECTION AND DIAGNOSIS

Ge Tong, Deng Jianhua

(Department of Aircraft Engineering, Northwestern Polytechnical University, Xi'an, 710072)

**摘 要** 提出元件模态转变的单向性假设以替代通常的诊断过程中元件模态不变的假设, 指出模型基础诊断方法是适合此类系统的基本诊断方法。由此提出一套用于复杂动态系统故障检诊的综合检诊策略 IFDDS (Integrated Failure Detection and Diagnosis Strategy), 针对飞控系统开发了其具体的检诊算法。

**关键词** 动态控制 故障 检测 诊断

**中图分类号** TP277, V249.1

**Abstract** Firstly, considering the failures' complicated combination modes and temporal characteristics, the authors raise a more practical assumption that the elements' modes can transfer but only from normal to fault at any time to replace the general assumption that the modes are unchanged in a diagnosis process. Moreover, based on this assumption, it is pointed out that the model-based diagnosis methods can be used as a general tool for this kind of system, and the diagnosis methods based on fault modes as an additional one. Secondly, based on the above consideration, an integrated failure detection and diagnosis strategy is supposed for a complex dynamic system which has the advantage of both artificial intelligence and mathematic algorithms, and it is also realized for a flight control system.

**Key words** dynamic control failure detection diagnosis

## 1 元件模态转变的单向性假设

复杂动态系统故障具有复杂的组合模式和时间特性, 为此, 提出元件模态转变的单向性假设: 系统的元件可以在任何时刻经历由正常至故障的模态转变。和通常的元件模态在诊断过程中不变的假设相比, 该假设更符合实际情况。这样, 先前做出的有关某元件故障的结论在以后的所有时刻都是有效的, 而先前做出的某元件正常的结论在以后的任何时刻都是无效的。因此, 检诊算法中大量应用的应该是能够推出元件故障的知识和方法, 对于此类系统中常用的一致性基础(consistency-based)<sup>[1]</sup>诊断, 表现为模型基础(model-based)的诊断方法。

## 2 故障检诊策略

### 2.1 基本诊断策略

基本诊断策略是一种模型基础的诊断方法<sup>[2]</sup>。任一元件 A 只有两种模态(行为模式):

正常模态  $A$  和故障模态  $X$ 。正常模态行为已知, 故障模态行为未知。以  $d_i$  表示当前的测量数据, 所有元件的每一种模态组合方式称为系统的实现, 则和  $d_i$  一致的系统实现称为  $i$  时刻的候选项 (candidate), 其集合记为  $CAN(i)$ , 是  $i$  时刻的诊断结果。为了产生  $CAN(i)$ , 必须提供一个领域专用的推理机  $C(d_i, ENV)$ , 其中  $ENV$  为某正常模态集, 称为环境, 它对应相应元件集的正常行为。推理机判断  $d_i$  和  $ENV$  的一致性, 若不一致, 则此时的  $ENV$  称为  $i$  时刻的矛盾项 (conflict) (以  $\langle \cdot \rangle$  表示)。

根据元件模态转变的单向性假设,  $i$  时刻的矛盾项必是以后所有时刻的矛盾项, 且任何  $i$  时刻矛盾项的扩集是  $i$  时刻矛盾项。这样, 一方面,  $i$  时刻只需产生以前未产生过的矛盾项其集合记为  $CON(i)$ , 且  $CON(i)$  可用其最小矛盾项以紧凑格式完全表示<sup>[2]</sup>; 另一方面, 可以采用序贯处理方法<sup>[2]</sup>, 利用  $CAN(i-1)$  和  $CON(i)$  构造  $CAN(i)$ , 且和  $CON(i)$  类似,  $CAN(i)$  可用其中最小候选项完全表示。

产生完备的  $CON(i)$  的机制记为  $CONG(d_i)$ , 产生  $CAN(i)$  的机制记为  $CANG(CAN(i-1), CON(i))$ 。诊断策略还需一个结束控制机制  $END$ , 当  $CAN(i)$  在一段相当的时间内不变时诊断结束。

## 2.2 引入故障模型

有时只凭模型基础诊断进行故障的精确定位是困难的, 必须引入元件的故障模型, 表现为推理机进行推理的模态集中出现元件的故障模态。这种模态集称为负环境, 相应的推理机记为  $NC(d_i, NENV)$ 。由此产生的不一致负环境称为负矛盾,  $i$  时刻的负矛盾集记为  $NCON(i)$ , 产生  $NCON(i)$  的机制记为  $NCONG(d_i)$ 。 $NCON(i)$  仍可用最小负矛盾表示成紧凑格式。但是,  $i$  时刻的负矛盾在以后的任何时刻可能不是负矛盾, 因而  $NCONG(d_i)$  必须对所有不包含  $i$  时刻负矛盾的负环境运行推理机。另外, 负矛盾出现后, 候选项的扩集可能不是候选项, 为了不破坏  $CAN(i)$  的紧凑格式,  $CAN(i)$  仍由  $CAN(i-1)$  和  $CON(i)$  产生, 而  $NCON(i)$  用来在  $CAN(i)$  基础上产生一个新的候选项集合  $CAN'(i)$ , 它为  $CAN(i)$  除去和  $NCON(i)$  不一致的候选项, 并取其中的最小候选项形成的集合。产生  $CAN'(i)$  的机制称为候选项的确认 (candidate validation), 记为  $VALID(CAN(i))$ 。

引入故障模型后, 结束控制机制根据  $CAN'(i)$  判断结束条件。完整的 IFDDS 故障诊断策略如图 1 所示, 其中  $i_0$  为诊断开始时刻,  $i$  为当前时刻,  $CAN_0$  (初始化中) 由检测算法提供。

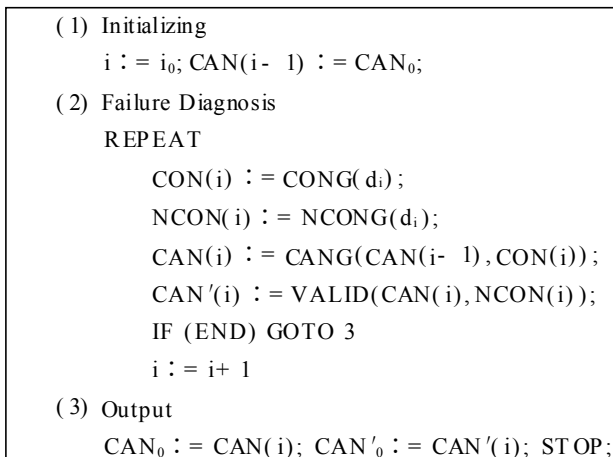


图 1 诊断策略

## 2.3 故障检测策略和全局检诊策略

检测过程实际上是诊断过程的一个子功能集。和诊断算法一样,检测算法也运行推理机  $C(d_i, ENV)$  对环境进行一致性判断,但只限于一部分环境。以  $DEEN$  表示这一部分环境的集合,这些环境要求在检测的初始时刻  $i_0$  之前是一致的,并且这些环境要能覆盖当前检测元件集  $\delta$ 。另外,检测算法也不必对负环境运行推理机。

全局检诊过程为由检测至诊断,诊断结束再开始检测的循环。

## 3 飞控系统的具体实现

IFDDS 用于具体领域必须开发领域专用的推理机算法及调度方法。

### 3.1 $C(d_i, ENV)$ 的设计

考虑飞控开环系统,包括  $m$  个作动器(以  $act$  表示)、 $l$  个传感器(以  $sen$  表示)和控制对象(以  $OBJ$  表示)。控制对象模型为

$$\begin{aligned} X(k+1) &= AX(k) + BU(k) \\ Y(k) &= CX(k) \end{aligned} \quad (1)$$

其中:  $X(k) \in R^n$ ,  $U(k) \in R^m$ ,  $Y(k) \in R^l$ 。ENV 中包含作动器和传感器数目分别为  $p$  和  $q$ 。

某  $act_i \in ENV$  时,其输出已知,而若  $act_i \notin ENV$ ,则其输出是不可知的,这样,  $U(k)$  被分成两部分,即

$$U(k) = [U_{ENV}^T(k), U_{ENV}^T(k)]^T$$

其中:  $U_{ENV}$  对应 ENV 中作动器;  $U_{ENV}$  对应 ENV 以外作动器,且  $U_{ENV}$  含  $p$  个分量。同样有

$$Y(k) = [Y_{ENV}^T(k), Y_{ENV}^T(k)]^T$$

利用未知输入观测器判断  $U_{ENV}$  和  $Y_{ENV}$  的一致性,方程(1)可写成以下形式

$$\begin{aligned} [Y^T(k-s), Y^T(k-s+1), \dots, Y^T(k)]^T = \\ FX(k-s) + H[U^T(k-s), U^T(k-s+1), \dots, U^T(k-1)]^T \end{aligned} \quad (2)$$

其中:

$$F = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^s \end{bmatrix}, \quad H = \begin{bmatrix} 0 & 0 & 0 \\ CB & 0 & 0 \\ CAB & CB & \\ & & 0 \\ CA^{s-1}B & \dots & CB \end{bmatrix}$$

$s$  为数据长度。因为  $Y_{ENV}$  不可知,去掉相应行,式(2)成为

$$\begin{aligned} [Y_{ENV}^T(k-s), Y_{ENV}^T(k-s+1), \dots, Y_{ENV}^T(k)]^T = \\ F_{ENV}X(k-s) + H_{ENV}[U_{ENV}^T(k-s), U_{ENV}^T(k-s+1), \dots, U_{ENV}^T(k-1)]^T + \\ H_{ENV}[U_{ENV}^T(k-s), U_{ENV}^T(k-s+1), \dots, U_{ENV}^T(k-1)]^T \end{aligned} \quad (3)$$

式中:  $F_{ENV}$ ,  $H_{ENV}$ ,  $H_{ENV}$  为相应的矩阵,若能选取向量  $v$ ,使得

$$v^T F_{ENV} = 0, \quad v^T H_{ENV} = 0 \quad (4)$$

$$v^T F_{ENV} \neq 0 \quad (5)$$

令

$$H_{ENV}[U_{ENV}^T(k-s), U_{ENV}^T(k-s+1), U_{ENV}^T(k-1)]^T \quad (6)$$

则当  $P(ENV, k) = 0$  时,  $C(d_i, ENV)$  一致, 否则不一致。考虑到系统噪声、计算误差以及防止虚警的要求, 以上判据中应组合一定的时序列表决方法, 如数据窗或序惯概率比法。

为了保证  $v$  存在,  $ENV$  中包含的作动器和传感器的数目有一下限, 具有此下限的环境称为最小环境。当  $ENV$  中包含余度传感器, 也即和  $ENV$  中所含传感器对应的  $C$  阵行向量线性相关时,  $ENV$  中可不包含作动器。此类环境的最小环境中只包含传感器且相应的  $C$  阵行向量构成最小相关组。当  $ENV$  中不包含余度传感器时, 必有

$$q \leq n \quad (7)$$

假设矩阵  $[F_{ENV}, H_{ENV}, H_{ENV}]$  列满秩, 为了保证满足式(4)的  $v$  存在, 要求  $[F_{ENV}, H_{ENV}]$  的行数大于列数, 有

$$s(p+q-m) + (q-n) > 0 \quad (8)$$

$$p+q = m+1 \quad (9)$$

矩阵  $[F_{ENV}, H_{ENV}, H_{ENV}]$  列不满秩时可类似处理。

只对最小环境运行推理机是足够的, 这是因为对于任何包含最小环境的环境  $ENV_i$ , 当  $C(d_i, ENV_i)$  一致时, 不能得到任何信息<sup>[2]</sup>, 反之说明  $ENV_i$  中必有故障元件, 这可以通过  $ENV_i$  包含的某些最小环境的不一致性体现出来。

### 3.2 NC( $d_i, NENV$ )的设计

$C(d_i, ENV)$  总是假定  $OBJ$  功能正常, 当将  $OBJ$  也作为可能故障元件时, 还必须引入一定的故障模态。本文假设: 如果  $OBJ$  故障, 则所有包含  $OBJ$  正常模态的环境  $ENV$  和  $d_i$  不一致。  $NC(d_i, NENV)$  利用此规则进行推理, 结论不满足时, 产生负矛盾 ( $\leftarrow OBJ$ )。

## 4 算 例

以某飞机纵向控制系统为例, 包括 2 个作动器 ( $m=2$ ) 和 3 个传感器 ( $i=3$ ), 分别为升降舵作动器 ( $A1$ )、襟翼作动器 ( $A2$ )、两个俯仰角速率传感器 ( $S1, S2$ ) 和攻角传感器 ( $S3$ )。其中  $S1, S2$  为余度传感器。现设在 4s 时,  $A2$  效率下降 50%, 8s 时  $S1$  产生固定偏差  $2(^{\circ})/s$ 。全局检诊过程如表 1 所示, 其中  $CAN, CAN'$  中只列出双元件以下故障情况。该过程包括 2 次循环, 第 1 次循环  $\delta = \{OBJ, A1, A2, S1, S2, S3\}$ ; 第 2 次  $\delta = \{OBJ, A1, S1, S2, S3\}$ , 2 次诊断结果分别为  $[A2]$  和  $[A2, S1]$ 。

## 5 总结与讨论

理论分析和仿真结果表明了本文算法的正确性和有效性。它组合了人工智能的高效性和数学检诊算法的精确性, 可成功用于如飞控系统一类复杂动态系统的多元件故障情况。还可以采取层次递进的诊断方式, 即将对象系统划分为多个层次, 由高至低在不同层次上依次运行检诊算法, 由此可进一步提高效率。

表1 全局检诊过程

过程	时刻/s	CON(i)	NCON(i)	CAN(i)	CAN'(i)	DEEN
检测	0~4.3	$\dot{A}$	-	[ ]	[ ]	{S1, A1, A2, OBJ}
	4.4	$\$1, A1, A2, OBJ$ $\$2, A1, A2, OBJ$	-	[OBJ], [A1] [A2][S1, S2]	[OBJ][A1] [A2][S1, S2]	{S2, A1, A2, OBJ} {S3, A1, A2, OBJ}
诊断	4.5	$\$3, A1, A2, OBJ$ $\$1, S3, A2, OBJ$ $\$2, S3, A2, OBJ$	$\leftarrow OBJ$	[OBJ], [A2] [A1, S3]	[A2][A1, S3]	-
	4.6~4.7	$\dot{A}$	$\leftarrow OBJ$	同上	同上	
检测	4.8~8.3	$\dot{A}$	-	同上	同左	
	8.4	$\$1, S3, A1, OBJ$	-	[OBJ][A2, S1] [A2, S3]	[OBJ] [A2, S1][A2, S3] [A1, A2][A1, S3]	{S1, S3, A1} {S2, S3, A1}
诊断	8.5	$\$1, S2$	$\leftarrow OBJ$	[OBJ, S1] [OBJ, S2][A2, S1]	[A2, S1]	-
	8.6~8.7	$\dot{A}$	$\leftarrow OBJ$	同上	同上	

## 参 考 文 献

- 1 Poole D. Normality and faults in logic-based diagnosis. In: Sridharan N S Ed. Proceedings of IJCAI-89, the 10th International Joint Conference on Artificial Intelligence, Detroit, 1989. California: Morgan Kaufmann Publishers Inc, 1989. 1304~1910
- 2 Kleer de, Williams B C. Diagnosing multiple faults. Artificial Intelligence, 1987, 32(1): 97~130