**Preprints of the 19th World Congress**
**The International Federation of Automatic Control**
**Cape Town, South Africa. August 24-29, 2014**

# Fault tolerant spacecraft attitude control by multiple control processors [1]

**Hao Yang** [*,**], **Bin Jiang** [*,**], **Vincent Cocquempot** [***],

[*] *College of Automation Engineering,*
*Nanjing University of Aeronautics and Astronautics, China*
[**] *State Key Laboratory of Synthetical Automation for Process*
*Industries (Northeastern University), China.*

[***] *LAGIS, UMR CNRS 8219,*
*Lille1 University: Sciences et Technologies, France*

**Abstract:** This paper considers fault tolerant attitude control problem of spacecraft under intermittent faults that occur in the control processor. A novel control framework based on multiple redundant control processors is provided, and a state-dependent switching law among these processors is proposed to stabilize the attitude dynamics without requiring control reconfiguration in each control processor when faults occur. Moreover, a probability-based method is provided to find the minimal number of control processors that are needed for attitude stabilization. Simulation results show the efficiency of the proposed methods.

## 1. INTRODUCTION

The potential faults in a complex system often range over a very large region. A single fault tolerant control (FTC) law (even an adaptive one) is often hard to design to stabilize all faulty situations effectively as indicated in Blanke et al. (2006); Yang et al. (2010). Supervisory FTC approaches assume that the plant model belongs to a pre-specified set of models, including the nominal situation and all possible faulty situations, and that there exists a finite family of candidate control laws such that the faulty system is stabilized when controlled by one of those candidate control laws as in Staroswiecki and Gehin (2011); Yang et al. (2009) or by switching among those control laws as in Yang et al. (2012).

Although multiple control laws are provided, the physical realization of supervisory FTC is often achieved by only one control processor (it will be called "processor" for short in the following if there is no confusion) which adopts the most appropriate control law. Such a supervisory FTC scheme obviously relies on the assumption that the processor is always healthy and available. In the presence of processor faults, most of (supervisory) FTC methods that are based on control reconfiguration are unavailable. Different from faults in actuator, sensors or the plant that are often permanent, most of processor faults are intermittent. An intermittent fault appears and disappears successively and randomly as described in Su et al. (1978), such faults can occur 10 to 30 times as often as the permanent faults and often exists in electronic equipments (Ismaeel and Bhatnagar (1997)).

This paper investigates the FTC problem of spacecraft attitude control system (ACS) and particulary focuses on a kind of intermittent processor faults ($\mathcal{I}$) that forces the torque inputs to be zero (the mechanism and formal model of $\mathcal{I}$ will be given later). Although FTC methods of spacecraft ACS have been researched for many years, e.g. Tafazoli and Khorasani (2006); Xiao et al. (2011), to name a few, most of these results assume that the faulty spacecraft is still controlled with three inputs and processor is always healthy. In the presence of complete failures such that torque inputs become zero, the spacecraft become underactuated, and the FTC design is more complicated as indicated in Tsiotras and Doumtchenko (2000). The fault tolerance of $\mathcal{I}$ deserves deep investigation due to two reasons:

1. For FTC design with hardware redundancy, multiple processors would be applied as backups. However, intermittent faults may occur in each processors, the reliability of the whole ACS may not be guaranteed even with multiple control processes. Moreover, too many processors obviously increase the hardware cost and computational burden of the spacecraft.
2. For FTC design with analytical redundancy, control reconfiguration has to be applied. However, it is difficult to adjust the controller to accommodate the fault in itself. Moreover such FTC takes time and control cost. Since intermittent faults may occur frequently, much control effort has to be made if we apply the FTC scheme every time when these faults occur. This is often not admissible in real situation of spacecraft operation.

This paper will answer two questions: 1) Is it possible to accommodate $\mathcal{I}$ by multiple processors without control reconfiguration in each processor? 2) how many processors are needed?

The main contributions and novelties are as follows:

[1] This work is supported by National Natural Science Foundation of China (61104116, 61273171), and Doctoral Fund of Ministry of Education of China (No.20113218110011).
Email: *haoyang@nuaa.edu.cn.* (H. Yang), *binjiang@nuaa.edu.cn* (B. Jiang), *vincent.cocquempot@univ-lille1.fr* (V. Cocquempot).

1. The ACS is modeled by a switched system where each mode represents the system with one of the processors. A novel switching scheme is proposed among such a family of redundant processors. It shows that if the period in which at least one processor is fault-free is long enough compared with that when all processors are faulty, then the attitude is stabilized without any control reconfiguration in each processor.
2. According to Markovian statistical property of intermittent faults, a probability-based method is provided to build a link between the fault tolerance analysis and the number of processors, under which the minimal number of processors that are needed for maintaining stability of ACS can be found.

In the rest of the paper: Section 2 presents some preliminaries. Section 3 analyzes the system behavior under single processor, Section 4 addresses the switching control issue with multiple processors. Section 5 provides simulation results, followed by conclusions in Section 6.

## 2. PRELIMINARIES

### *2.1 Rigid spacecraft model*

Consider a spacecraft whose principal axes of the body-fixed reference frame are along the direction of principal axes of the inertia moments. The kinematics equation is:

$$\dot{q} = \frac{1}{2}(q_4\omega - \omega^{\times} q), \quad \dot{q}_4 = -\frac{1}{2}\omega^{\top} q$$

where $\omega \in \Re^3 \triangleq [\omega_1\ \omega_2\ \omega_3]^{\top}$ represents the inertial angular velocity vector. $q \in \Re^3 \triangleq [q_1\ q_2\ q_3]^{\top}$, $q_4$ is a scalar, $q_1$, $q_2$, $q_3$ and $q_4$ denote the quaternions. $J = J^{\top}$ is the inertia matrix. The cross product is defined as:

$$\omega^{\times} \triangleq \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}$$

The dynamic equation is:

$$J\dot{\omega} = -\omega^{\times} J\omega + Du \qquad (1)$$

where $u \in \Re^3$ is the output of the processor, $D = diag[1, 1, 1]$ is the actuator distribution matrix. $Du$ represents the torque input generated by the thrusters. Eq. (1) can also be expressed as in Wertz (1995):

$$\ddot{q} = -\frac{1}{4}\omega^{\top}\omega q + \frac{1}{2}Q\left(-J^{-1}(\omega^{\times}J\omega) + J^{-1}Du\right) \qquad (2)$$

where $Q \triangleq \begin{bmatrix} q_4 & -q_1 & -q_2 \\ q_1 & q_4 & -q_3 \\ q_2 & q_3 & q_4 \end{bmatrix}$.

### *2.2 Model of $\mathcal{I}$*

Under $\mathcal{I}$, the torque inputs become zero, i.e., $Du = 0$. This includes three cases: 1) The fault brakes the programme running process of the processor and makes the command signals from the processor to thrusters be zero, i.e., $u = 0$; 2) The fault leads to the short circuit of the processor and makes $u = 0$; 3) The fault affects the processor such that the command signals deviate from normal, which is very dangerous, thus the actuators are automatically stopped, i.e., let $D = 0$. Such an operation is available since thrusters can work in both continuous and impulsive ways.

Denote $u^{no}$ as the nominal control law of ACS, one has

$$Du(t) = \begin{cases} u^{no}(t) & \text{if there is no fault} \\ 0 & \text{if fault appears} \end{cases}$$

The model of intermittent faults are often described by a transition system with two modes (one is for the healthy situation and the other is for the faulty situation). The transitions between these two modes, i.e. the appearance and disappearance of the faults follow the well known continuous-parameter Markov rule as in Su et al. (1978). Such a model is adopted for $\mathcal{I}$. It follows that

$$\mathbf{P}\{Du(t+\Delta t) = 0 | Du(t) = u^{no}(t)\} = \rho_{01}\Delta t \quad (3)$$

$$\mathbf{P}\{Du(t+\Delta t) = u^{no}(t+\Delta t) | Du(t) = 0\} = \rho_{10}\Delta t \quad (4)$$

where $\mathbf{P}$ denotes the probability, $0 \leq \rho_{01} < 1$ represents the fault appearance rates, and $0 \leq \rho_{10} < 1$ represents the fault disappearance rates, $\Delta t \geq 0$ is a period. Throughout the paper, it is supposed that the initial situation of the processor is healthy.

### *2.3 Problem formulation*

Define $x \triangleq [q^{\top}, \epsilon_1\dot{q}^{\top}]^{\top}$, where $\epsilon_1 > 0$ is a constant to be chosen. Note that

$$\omega = 2\begin{bmatrix} -q_1 & q_4 & q_3 & -q_2 \\ -q_2 & -q_3 & q_4 & q_1 \\ -q_3 & q_2 & -q_1 & -q_4 \end{bmatrix}\begin{bmatrix} \dot{q}_4 \\ \dot{q} \end{bmatrix} = -2\dot{q}_4 q + 2\bar{Q}\dot{q}$$

where $\bar{Q} \triangleq \begin{bmatrix} q_4 & q_3 & -q_2 \\ -q_3 & q_4 & q_1 \\ q_2 & -q_1 & -q_4 \end{bmatrix}$. Eq. (2) can be rewritten as:

$$\dot{x} = F(x) + G(x)u \qquad (5)$$

where $F$ and $G$ can be obtained from (2). It is clear that if $x \to 0$, then $q \to 0, q_4 \to 1, \omega \to 0$, i.e., the attitude is asymptotically stable at origin.

With $m$ ($m > 1$) redundant processors, the ACS switches among these processors and apply one of them at one time, thus the system (5) is rewritten as

$$\dot{x} = F(x) + G(x)u_{\sigma} \qquad (6)$$

where $\sigma(t) : [0, \infty) \to \mathcal{M} = \{1, ..., m\}$ denotes the *switching function*, and $u_i$ denotes the output of processor $i$, whose nominal control law is denoted as $u_i^{no}$ accordingly.

The problem to be solved in this paper is: *Given any $\rho_{01}$ and $\rho_{10}$ (the appearance and disappearance rates of $\mathcal{I}$), choose $m$ (the minimal number of processors), design $u_i^{no}(t)$ of each processor $i$ and a switching function $\sigma(t)$ among processors such that the origin of system (6) is asymptotically stable without reconfiguring $u_i$ of each processor $i$ in its faulty case.*

## 3. SYSTEM BEHAVIOR UNDER SINGLE PROCESSOR

Suppose that processor $i$, $i \in \mathcal{M}$, is applied to the ACS, i.e. $\sigma = i$.

*Lemma 1*: Consider the system (6) with $\sigma(t) = i$, $i \in \mathcal{M}$, and there is no fault. There exists an initial condition of $x(0)$ and $u_i^{no}$ such that the origin of system (6) is asymptotically stable.

*Proof*: Design the nominal control law

$$u_i^{no} = \left(\frac{1}{2}QJ^{-1}\right)^{-1}\left(\frac{1}{4}\omega^\top\omega q + \frac{1}{2}Q\Big(-J^{-1}(\omega^\times J\omega)\Big)\right.$$
$$\left. -k_1 q - k_2\epsilon_1\dot q\right) \qquad (7)$$

where $Q$ is defined in (2), $k_1$ and $k_2$ are two positive constants.

Substituting (7) into (6) yields

$$\dot x = \begin{bmatrix} 0_{3\times3} & \frac{1}{\epsilon_1}I_{3\times3} \\ -k_1 I_{3\times3} & -k_2 I_{3\times3}\end{bmatrix} x \qquad (8)$$

For any $\epsilon_1 > 0$, we can chose $k_1$ and $k_2$ such that the system (8) is asymptotically stable at origin. Consider a function $V = V_1 + V_2 + V_3$ where $V_i \triangleq [q_i\ \epsilon_1\dot q_i]^\top P[q_i\ \epsilon_1\dot q_i]^\top$ with $P$ being positive definite symmetric matrix, its time derivative along the solution of (8) satisfies

$$\dot V \le -\lambda_0 V \qquad (9)$$

for $\lambda_0 > 0$. It follows that $|x(t)| \le Me^{-\frac{\lambda_0}{2}t}|x(0)|$, where $M \triangleq \sqrt{\frac{\lambda_{\max}(P)}{\lambda_{\min}(P)}}$.

Note that the control law (7) is available if $Q$ is non-singular, this requires that $q_4 \ne 0$. If we choose the initial state satisfying

$$|x(0)| \le \frac{\alpha}{M}, \quad \alpha < 1 \qquad (10)$$

then $q_4(0) \ne 0$, control law (7) is available at $t = 0$. It follows from (9) that under (7), $|q(t)| \le \alpha\ \forall t \ge 0$, this means that $q^\top q(t) \le \alpha$ and $q_4^2(t) \ge 1-\alpha$, $\forall t \ge 0$. Therefore control law (7) always works and $\lim_{t\to\infty} V(t) = 0$. This completes the proof.

The initial condition (10) implies that if the initial Euler angle $\theta \in (-\pi, \pi)$, then under (7), $\theta \to 0$ and would never reach $\pi$. This does not restrict $\dot q(0)$ since $\epsilon_1$ can be chosen small.

*Lemma 2*: Consider the system (6) with $\sigma(t) = i$, $i \in \mathcal{M}$, and $Du_i = 0$. If $|x(t)| \le \alpha$, then $\dot V \le \lambda_1 V$ for $\lambda_1 \ge 0$.

*Proof*: Since $Du_i = 0$, the system (8) changes into

$$\dot x = \left[\frac{1}{\epsilon_1}x_4\ \ \frac{1}{\epsilon_1}x_5\ \ \frac{1}{\epsilon_1}x_6\ \ f_1(\omega,x)\ \ f_2(\omega,x)\ \ f_3(\omega,x)\right]^\top \qquad (11)$$

where

$$f_1(\omega,x) \triangleq -\frac{1}{4}\omega^\top\omega q_1 + \frac{1}{2}[q_4\ \ -q_1\ \ -q_2]\left(J^{-1}(\omega^\times J\omega)\right)$$
$$f_2(\omega,x) \triangleq -\frac{1}{4}\omega^\top\omega q_2 + \frac{1}{2}[q_1\ q_4\ \ -q_3]\left(J^{-1}(\omega^\times J\omega)\right)$$
$$f_3(\omega,x) \triangleq -\frac{1}{4}\omega^\top\omega q_3 + \frac{1}{2}[q_2\ q_3\ q_4]\left(J^{-1}(\omega^\times J\omega)\right)$$

The time derivative of $V$ along the solution of (11) is

$$\dot V \le 2|x||P|\left(\sum_{i=1}^{3}|f_i(\omega,x)| + \frac{|x|}{\epsilon_1}\right) \qquad (12)$$

Since $|x| \le \alpha$, $|q| \le \alpha$, one has that

$$|\omega| \le 2\alpha|\dot q_4| + 2\sqrt{2+\alpha}|\dot q| \qquad (13)$$

Also note that $|q_4\dot q_4| = |-q^\top\dot q| \le \alpha|\dot q|$, it follows that

$$|\dot q_4| \le \frac{\alpha}{\sqrt{1-\alpha}}|\dot q| \qquad (14)$$

Substituting (14) into (13) yields

$$|\omega| \le \underbrace{\left(\frac{2\alpha^2}{\sqrt{1-\alpha}} + 2\sqrt{2+\alpha}\right)}_{\Lambda}|\dot q| \qquad (15)$$

One further has that

$$|f_i(\omega,x)| \le \frac{1}{4}|\omega|^2|q| + \frac{1}{2}|J^{-1}||\omega^\times J\omega|$$
$$\le \underbrace{\left(\frac{\alpha}{4} + \frac{\sqrt{2}}{2}|J^{-1}||J|\right)\Lambda^2}_{\Psi}|\dot q|^2 \le \frac{\Psi\alpha}{\epsilon_1}|\dot q| \qquad (16)$$

Substituting (16) into (12) leads to

$$\dot V \le 2|x||P|\left(\frac{3\Psi\alpha}{\epsilon_1^2}|x| + \frac{1}{\epsilon_1}|x|\right)$$
$$\le 2|P|\frac{6\Psi\alpha + \epsilon_1}{\epsilon_1^2}|x|^2 \le \lambda_1 V \qquad (17)$$

This completes the proof.

Lemmas 1 and 2 mean that in the healthy situation, under initial condition satisfying (10) and nominal control law as in (7), the origin of ACS can be exponentially stabilized. In the presence of fault, the states may diverge no faster than exponential provided it is bounded within a region.

## 4. FTC VIA MULTIPLE REDUNDANT PROCESSORS

### *4.1 Switching control framework*

Onboard computers and processors of spacecraft often need the hot backups that always work even they are not used for the purpose of reliability. Inspired by such a setting, a switching control framework is proposed as shown in Fig. 1, where $m$ processors work in parallel, each one is a hot backup of others. Each processor $i$, $i \in \mathcal{M}$ is either connected with spacecraft body denoted as $\mathcal{B}$ or connected with its virtual body denoted as $\mathcal{E}_i$. At one time, only one of processors is chosen to be connected with $\mathcal{B}$, others are connected with $\mathcal{E}_i$. The control law $u_i$ of processor $i$ is always designed as $u_i^{no}$ whatever the processor is connected with $\mathcal{B}$ or with $\mathcal{E}_i$.

It is assumed that the appearance and disappearance of $\mathcal{I}$ can be detected rapidly by using certain fault diagnosis scheme which is not the main focus of the paper. Interested readers are referred to Su et al. (1978); Ismaeel and Bhatnagar (1997), Blanke et al. (2006) for detailed information. The real-time fault diagnosis information of processor $i$

is sent to $\mathcal{E}_i$ and the switching scheme. Based on these information, the switching scheme provides the switching function $\sigma(t)$, and chooses one of processors to connect with $\mathcal{B}$.

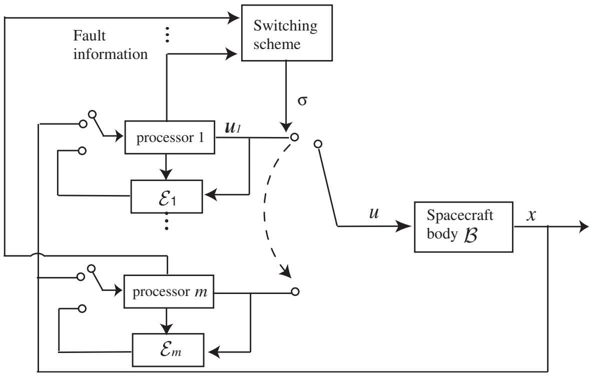

Fig. 1. Switching control framework

We will first discuss the design of $\mathcal{E}_i$, then propose a switching law among processors.

### 4.2 Design of $\mathcal{E}_i$

$\mathcal{E}_i$ works when processor $i$ is connected with it. The dynamics of $\mathcal{E}_i$ is also represented by a switched system with two modes

$$\dot{z}_i = F_{\varrho_i}(z_i) + G_{\varrho_i}(z_i)u_i \tag{18}$$

where $z_i \in \Re^6$ is the state, $\varrho_i(t) : [0, \infty) \to \{1, 2\}$ is a switching function, $\varrho_i(t) = 1$ if there is no fault in processor $i$, and $\varrho_i(t) = 2$ if fault occurs. The synchronization between the switchings of two modes of $\mathcal{E}_i$ and the appearance/disappearance of processor $i$'s fault can be achieved based on fault diagnosis information.

The dynamics of mode 1 is designed to be the same as $\mathcal{B}$, i.e., $F_1(z) = F(z)$ and $G_1(z) = G(z)$, where $F(\cdot)$ and $G(\cdot)$ are defined in (5). One has that

$$\frac{\partial V}{\partial z_i}\left(F_1(z_i) + G_1(z_i)u_i\right) \leq -\lambda_0 V(z_i) \tag{19}$$

where $V$ takes the same form as in Section 3.

The dynamics of mode 2 is designed as $F_2(z_i) = Az_i$ where $A$ is a Hurwitz matrix and $G_2(z_i) = 0$ such that

$$\frac{\partial V}{\partial z_i}A(z_i) \leq -\lambda_2 V(z_i) \quad \text{for } \lambda_2 > 0 \tag{20}$$

It can be seen from (19)-(20) that whatever processor $i$ is faulty or not, $z_i$ exponentially converges to zero if processor $i$ is connected with $\mathcal{E}_i$.

To guarantee the availability of nominal control law $u_i^{no}$ for $\mathcal{E}_i$. At every time instant $t_s$ after which processor $i$ is connected with $\mathcal{E}_i$, the state values $z_i(t_s)$ is chosen such that

$$|z_i(t_s)| \leq \frac{\alpha}{M}, \quad \alpha < 1 \tag{21}$$

### 4.3 Switching law design

Divide $\mathcal{M} = \mathcal{M}_h \bigcup \mathcal{M}_f$, where $\mathcal{M}_h$ denotes the set of healthy processors, and $\mathcal{M}_f$ denotes the set of faulty processors. Since under $\mathcal{I}$, the fault appears and disappears intermittently in each processor, both $\mathcal{M}_h$ and $\mathcal{M}_f$ are time variant.

The switching law among processors are given as:

*Switching law $\mathcal{S}$*

1. *At $t = 0$, apply an arbitrary processor $i$, $i \in \mathcal{M}$, to $\mathcal{B}$.*
2. *Until $t = t^\star$ such that $i \in \mathcal{M}_f(t^\star)$, if $\mathcal{M}_h(t^\star) \neq \emptyset$, disconnect processor $i$ from $\mathcal{B}$, go to step 3; else go to step 4.*
3. *Pick an arbitrary controller $j \in \mathcal{M}_h(t^\star)$, apply it to $\mathcal{B}$, let $i = j$, go to step 2.*
4. *Continue applying processor $i$ to $\mathcal{B}$, until $t = t^{\star\star}$ such that $\mathcal{M}_h(t^{\star\star}) \neq \emptyset$, let $t^\star = t^{\star\star}$, go to step 3.* ■

The main idea behind $\mathcal{S}$ is that at each time one healthy processor $i$ is connected with $\mathcal{B}$ until this processor can not stabilize $\mathcal{B}$ due to fault, then another healthy processor is connected with $\mathcal{B}$. If there is no healthy processor, processor $i$ is still applied until a healthy one appears.

It can be seen that such a switching law relies on real-time situations (healthy or faulty) of all processors. Thanks to the structure of $\mathcal{E}_i$ as described in Section 4.2, $\mathcal{S}$ is implementable since each processor always works whatever it is faulty or not by being connected with $\mathcal{E}_i$ or $\mathcal{B}$, its real situation is always known by fault diagnosis scheme.

### 4.4 Fault tolerance analysis

For any $t > 0$, divide the interval $[0, t)$ into two parts: $\Delta^h_{aoc}(t)$ and $\Delta^f_{allc}(t)$, where $\Delta^h_{aoc}$ denotes the period in which at least one healthy processor exists and $\Delta^f_{allc}$ denotes the period in which all processors are faulty.

*Theorem 1*: The origin of (6) with initial condition satisfying (10) is asymptotically stable by $m$ redundant processors under switching law $\mathcal{S}$ if

$$\lambda_0 \Delta^h_{aoc}(t) > \lambda_1 \Delta^f_{allc}(t), \quad \forall t > 0 \tag{22}$$

*Proof*: According to Step 1 of $\mathcal{S}$, apply an arbitrary processor $i$ to $\mathcal{B}$. Since the initial situation of each processor is healthy, based on lemma 1, applying processor $i$ with nominal control law as in (7) and choosing the initial condition satisfying (10) guarantee $\dot{V} \leq -\lambda_0 V$. It follows that $V(t) \leq e^{-\lambda_0 t}V(0)$ for $t < t_f$ where $t_f$ is the time when fault occurs in processor $i$.

At $t = t_f$, two cases are considered:

- *Case 1, $\mathcal{M}_h(t_f) = \emptyset$.*

  According to Step 4 of $\mathcal{S}$, processor $i$ is still applied to $\mathcal{B}$ until $t = t^{\star\star}$ such that $\mathcal{M}_h(t^{\star\star}) \neq \emptyset$. It follows that $\Delta^h_{aoc}(t^{\star\star}) = t_f$, $\Delta^f_{allc}(t^{\star\star}) = t^{\star\star} - t_f$.

  Note that $|x(t_f)| \leq \alpha < 1$, thus control law (7) is still available at $t_f$. According to Lemma 2, one has

  $$V(t) \leq e^{-\lambda_0 t_f + \lambda_1 (t - t_f)}V(0)$$

  for $t < t_{escape}$ where $t_{escape}$ denotes the time when $|x(t_{escape})| \geq 1$. Note that for $t \geq t_{escape}$, $q_4(t)$ may equal zero, which violates the control law (7).

On the other hand, Condition (22) guarantees that $V(t) < V(0)$, which means that $|x(t)| \leq \alpha$, $\forall t \leq t^{\star\star}$. Therefore the control law (7) and lemmas 2 is always available in $[0, t^{\star\star})$. It follows that

$$V(t^{\star\star}) \leq e^{-\lambda_0 t_f + \lambda_1 (t^{\star\star} - t_f)} V(0) < V(0)$$

Thus $|x(t^{\star\star})| \leq \alpha < 1$.

- *Case 2*, $\mathcal{M}_h(t_f) \neq \emptyset$.

According to Step 3 of $\mathcal{S}$, at $t = t_f$, apply another healthy processor $j$ to $\mathcal{B}$. We have that

$$V(t_f) \leq e^{-\lambda_0 t_f} V(0) < V(0)$$

It holds that $|x(t_f)| \leq \alpha < 1$.

Therefore, when another healthy processor $j$ is applied to $\mathcal{B}$, the nominal control law is always available.

$\mathcal{S}$ guarantees that for any $t > 0$, one of healthy processors is always being applied to $\mathcal{B}$ in $\Delta^h_{aoc}(t)$, while in $\Delta^f_{allc}(t)$, a faulty processor is applied, it follows that

$$V(x(t)) \leq e^{-\lambda_0 \Delta^h_{aoc}(t) + \lambda_1 \Delta^f_{allc}(t)} V(x(0)), \quad \forall t > 0$$

Condition (22) guarantees that $V(t)$ always decreases, therefore when each processor is connected with $\mathcal{B}$, the nominal control law is always available. Finally, $\lim_{t \to 0} V(t) = 0$. This completes the proof.

It is interesting to compare (22) with the stability condition of ACS under individual processor. Denote $\Delta^h_i(t)$ and $\Delta^f_i(t)$ respectively as the period in which the individual processor $i$ is healthy and faulty in $[0, t)$. It is clear that processor $i$ stabilizes $\mathcal{B}$ if

$$\lambda_0 \Delta^h_i(t) > \lambda_1 \Delta^f_i(t), \quad \forall t > 0 \tag{23}$$

In this case, there is no need to switch among multiple redundant processors. In the proposed multiple-processors switching scheme, even all processors do not satisfy (23), condition (22) may still hold. The more is the number of processors, the less restrictive is condition (22). This explicitly reveals the advantage of using multiple processors.

### *4.5 The minimal number of processors*

Condition (22) of Theorem 1 can be used for checking on-line whether the attitude is stable. However it is unavailable *a priori* for the determination of the number of redundant processors. This motivates us to further investigate the statistic properties of $\mathcal{I}$ which can build a link between the fault tolerance analysis and the number of processors as it will be shown.

For each processor, denote $\rho_h(t)$ and $\rho_f(t)$ respectively as the probability of the healthy and faulty situation at $t$. Since the initial situation is healthy, it follows from the Markovain jump theory (see Parzen (1962)) and (3)-(4) that $\forall t \geq 0$

$$\rho_f(t) = \frac{\rho_{01}}{\rho_{01} + \rho_{10}} \left(1 - e^{-(\rho_{01} + \rho_{10})t}\right) \leq \frac{\rho_{01}}{\rho_{01} + \rho_{10}} \tag{24}$$

$$\rho_h(t) = 1 - \rho_f(t) \geq \frac{\rho_{10}}{\rho_{01} + \rho_{10}} \tag{25}$$

Inequalities (24)-(25) implies that a large (small) fault appearance rate $\rho_{01}$ leads to a large (small) probability of being faulty at present time, while a large (small) fault disappearance rate $\rho_{10}$ leads to a large (small) probability of being healthy (faulty) at present time.

*Definition 1 :* The origin of the system (6) is *asymptotically stable* in probability if $\lim_{t\to\infty} \mathbf{E}(x) = 0$, where $\mathbf{E}$ denotes the mathematical expectation.

*Theorem 2*: The origin of (6) with initial condition satisfying (10) is asymptotically stable in probability by $m$ redundant processors under switching law $\mathcal{S}$ if

$$\lambda_0(1 - (\varrho_f)^m) > \lambda_1 (\varrho_f)^m, \quad \forall t > 0 \tag{26}$$

where $\varrho_f \triangleq \frac{\rho_{01}}{\rho_{01} + \rho_{10}}$.

*Proof*: It follows from (24) and (25) that

$$\mathbf{E}(\Delta^f_{allc}(t)) = t(\rho_f)^m \leq t(\varrho_f)^m$$
$$\mathbf{E}(\Delta^f_{aoc}(t)) = t - \mathbf{E}(\Delta^f_{allc}(t)) \leq t - t(\varrho_f)^m$$

Under switching law $S$, applying processor $i$ with control law designed as $u^{no}_i$ in (7) and choosing the initial condition satisfying (10) leads to

$$\begin{aligned}\mathbf{E}(V(x(t))) &\leq e^{-\lambda_0 \mathbf{E}(\Delta^h_{aoc}(t)) + \lambda_1 \mathbf{E}(\Delta^f_{allc}(t))} V(x(0)) \\ &\leq e^{t(-\lambda_0(1-(\varrho_f)^m) + \lambda_1(\varrho_f)^m)} V(x(0)), \quad \forall t > 0\end{aligned}$$

Condition (26) ensures that $\mathbf{E}(V(t)) < V(0)$, which means that when each processor is connected with $\mathcal{B}$, the nominal control law is always available in probability. Finally, with (26), $\lim_{t\to\infty} \mathbf{E}(V(t)) = 0$. This completes the proof.

Following condition (26), we can choose a minimal number of $m$ such that

$$(\varrho_f)^m < \frac{\lambda_0}{\lambda_0 + \lambda_1} \tag{27}$$

Condition (27) reveals that the appropriate selection of number $m$ depends on the decay rate of the system with the healthy processor, the diverging rate with the faulty processor, and the fault appearance and disappearance rates. If $\rho_{10}$ and $\lambda_0$ are large enough such that

$$\varrho_f < \frac{\lambda_0}{\lambda_0 + \lambda_1} \tag{28}$$

then one processor can stabilize the ACS in the presence of intermittent faults. Since $\varrho_f < 1$, one has that (28) ⇒ (27) while the converse may not be true.

## 5. SIMULATION RESULTS

In the simulation, the inertia matrix is chosen as in Xiao et al. (2011) :

$$J = \begin{bmatrix} 350 & 3 & 4 \\ 3 & 270 & 10 \\ 4 & 10 & 190 \end{bmatrix} \mathrm{kg} \cdot \mathrm{m}^2$$

The initial parameters are $(q_1, q_2, q_3, q_4) = (0.308, 0.218, -0.218, 0.9)$, $(\omega_1, \omega_2, \omega_3) = (0, 0, 0)$ (rad/s), therefore $\dot{q}(0) = 0$. Choose $\alpha = 0.4$, $\epsilon_1 = 1$. The feedback gains are $k_1 = k_2 = 1$. Simple calculations lead to $\lambda_0 = 1.5$, $\lambda_1 = 10.5$. Also choose $A$ in (20) such that $\lambda_2 = 1.5$.

The appearance rate and disappearance rate of $\mathcal{I}$ are supposed to be $\rho_{01} = 0.2$, $\rho_{10} = 0.6$. It follows that $\varrho_f = \frac{1}{4}$, while $\frac{\lambda_0}{\lambda_0+\lambda_1} = \frac{1}{12}$. According to condition (26), let $m = 2$, i.e., 2 processors are applied for the FTC purpose.

The first two sub-figures of Fig. 2 illustrate the healthy periods and faulty periods of two processors that are generated under $\rho_{01}$ and $\rho_{10}$, the function $\chi_i(t) = 1$ ($\chi_i(t) = 0$) when processor $i$ is healthy (faulty), $i = 1, 2$. It can be seen that in period $[0.50)s$, processor 1 is healthy in periods $[0, 7.7) \bigcup [12.4, 22.5) \bigcup [28.7, 41.3) \bigcup [45.1, 50)s$ and processor 2 is healthy in periods $[0, 3.6) \bigcup [8.7, 14.1) \bigcup [16.7, 30.8) \bigcup [37.8, 43.1) \bigcup [47.8, 50)s$.

The third sub-figure of Fig. 2 shows the trajectory of $\sigma(t)$ according to switching law $\mathcal{S}$. Processor 1 is applied to the spacecraft in periods $[0, 8.7) \bigcup [12.4, 22.5) \bigcup [30.8, 41.3) \bigcup [45.1, 50)s$ and processor 2 is applied in other periods.

Fig. 3 shows trajectories of $Du_\sigma$. Since in periods $[7.7, 8.7) \bigcup [43.1, 45.1)s$, both processors are faulty, there is no torque input in these periods. Fig. 4 shows the behaviors of $\omega$, $q$ and $q_4$. It can be seen that when there is no torque input, the states diverge, however, the attitude stability in the whole process is achieved under switching between two processors in spite of intermittent faults.

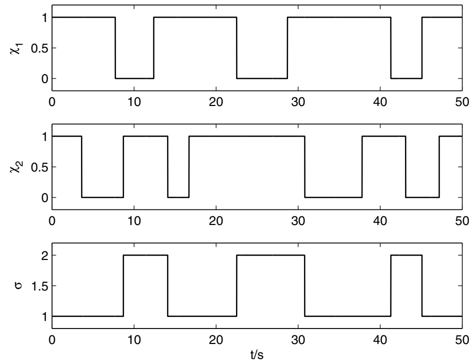

Fig. 2. The healthy and faulty periods of two processors and the trajectory of $\sigma(t)$

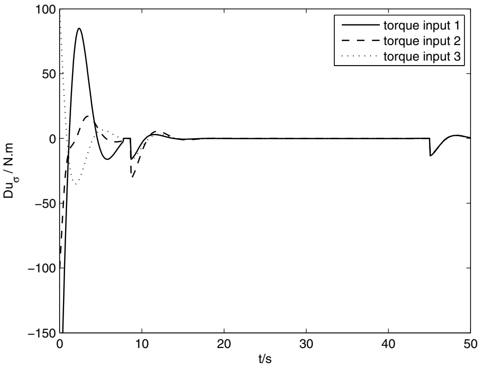

Fig. 3. the trajectories of $Du_\sigma$

## 6. CONCLUSION

This paper provides a new fault tolerant control method for spacecraft with intermittent faults. Such a method relies on the trade-off among multiple processors, and provides a new FTC clue in the case that control reconfigurations are difficult to be done. In this work, all states are available, output-feedback control together with observer design would be considered in the absence of full state measurements.

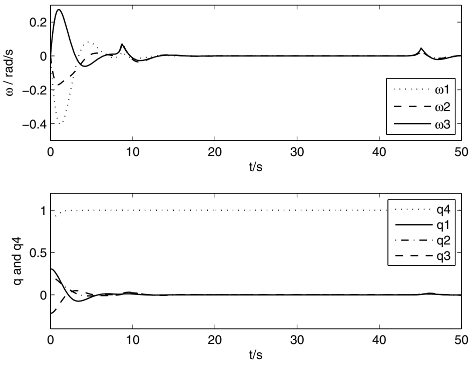

Fig. 4. The trajectories of $\omega$, $q$ and $q_4$

## REFERENCES

M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki. *Diagnosis and Fault-Tolerant Control, 2nd edition.* Springer Verlag, Berlin, 2006.

A. A. Ismaeel and R. Bhatnagar. Test for detection & location of intermittent faults in combinational circuits. *IEEE Transactions on Reliability*, 46(2):269–274, 1997.

E. Parzen. *Stochastic Processes.* Holden-Day, New York, 1962.

M. Staroswiecki and A. L. Gehin. From control to supervision. *Annual Reviews in Control*, 25(1):1–11, 2011.

S. Su, I. Koren, and Y. K. Malaiya. A continuous-parameter markov model and detection procedures for intermittent faults. *IEEE Transactions on Computers*, C-27(6):567–570, 1978.

S. Tafazoli and K. Khorasani. Nonlinear control and stability analysis of spacecraft attitude recovery. *IEEE Transactions on Aerospace and Electronic Systems*, 42(3): 825–845, 2006.

P. Tsiotras and V. Doumtchenko. Control of spacecraft subject to actuator failures: state-of-the-art and open problems. *Journal of the Astronautical Sciences*, 48(2-3):337–358, 2000.

J. R. Wertz. *Spacecraft Attitude Determination and Control.* Kluwer Academics Publishers, Boston, 1995.

B. Xiao, Q. Hu, and Y. M. Zhang. Fault-tolerant attitude control for flexible spacecraft without angular velocity magnitude measurement. *Journal of Guidance, Control, and Dynamics*, 34(5):1556–1561, 2011.

H. Yang, B. Jiang, and M. Staroswiecki. Supervisory fault tolerant control for a class of uncertain nonlinear systems. *Automatica*, 45(10):2319–2324, 2009.

H. Yang, B. Jiang, and V. Cocquempot. *Fault Tolerant Control Design For Hybrid Systems.* Springer Verlag, Berlin Heidelberg, 2010.

H. Yang, B. Jiang, V. Cocquempot, and Lingli Lu. Supervisory fault tolerant control with integrated fault detection and isolation: switched system approach. *International Journal of Applied Mathematics and Computer Science*, 22(1):87–97, 2012.