Increasing Performance of Spacecraft Active Fault-tolerant Control Using Neural Networks

Rouzbeh Moradi
Assistant Professor
Urmia University of Technology
Faculty of Aerospace Engineering
Iran

Actuator fault poses a challenge to the attitude control of spacecraft. Fault-tolerant control (active or passive) is often used to overcome this challenge. Active methods have better performance than passive methods and can manage a broader range of faults. However, their implementation is more difficult. One reason for this difficulty is the critical reaction time. The system may become unrecoverable if the actual reaction time becomes larger than the critical reaction time. This paper proposes using a feedforward neural network to reduce the actual reaction time in the active fault-tolerant control of spacecraft. Besides this improvement, using a feedforward neural network can increase the success percentage. Success percentage is the ratio of successful simulations to the total number of simulations. Simulation results show that for 200 simulations with random faults and initial conditions, the actual reaction time decreases by 73%, and the success percentage increases by 25%. Based on these results, the proposed controller is a good candidate for practical applications.

Keywords: *Attitude Control; Spacecraft; Fault-Tolerant Control; Neural Network.*

1. INTRODUCTION

Research into the conditions and preparation for a space flight to Mars has been especially actualized and intensified in recent years [1-3]. Successful spacecraft attitude control is crucial to the growth of this field. For instance, for Mars orbiters to effectively complete their missions, their attitude needs to be appropriately adjusted. The subsystem responsible for attitude adjustment is called ACS [4].

Despite taking measures to prevent faults, there are always possibilities of fault occurrence in ACS [4]. If not handled properly, the fault occurrence can lead to performance degradation and even mission failure. For these reasons, the literature studied spacecraft FTC extensively [5-14].

According to [4], the actuator fault is the most important reason for ACS failure. The partial loss of actuator effectiveness [15] is considered the fault model. Actuator saturation is considered in the controller design.

FTC is divided into two categories: active and passive. PFTCs have a fixed controller structure and need neither FDD nor RM. Therefore, their implementation is easier. However, they are too conservative (from a performance point of view) and can deal with a limited range of faults/failures. AFTCs have FDD and RM. They are less conservative and can handle many faults/failures. On the other hand, their implementation is more difficult [16-19].

According to the literature, several papers have considered the AFTC of rigid spacecraft [9-11]. These papers have considered sliding mode and backstepping methods and have proposed novel ideas for the spacecraft AFTC problem.

Due to constraints in the problem, e.g., actuator saturation and final time interval, FTC may not tolerate severe faults. Therefore, a quantitative measure should be defined for the efficiency of RM. SP is used for this purpose and is defined as the ratio of the number of successful simulations to the total number of simulations for various faults and initial conditions [20]. SP should be as large as possible.

When a fault occurs, there will be limited time to reconfigure the controller before the system becomes unrecoverable. This time is called CRT. The time required for FDD to detect and diagnose the fault, besides the time required for RM to reconfigure the controller, is ART. Therefore, ART should be as small as possible. If ART becomes larger than CRT, the system may enter an unrecoverable state [17,18].

As stated previously, hard implementation is a challenge in AFTC design. Large ART is one factor that leads to this challenge. As ART becomes larger, implementation becomes more difficult. Therefore, reducing ART will make the AFTC implementation easier.

To the author's knowledge, previous works have not considered ART and SP in the spacecraft AFTC design process. The main contribution is to consider these important parameters explicitly in the AFTC design of spacecraft and propose using feedforward NN [21] to increase SP and decrease ART to a large extent. This will make the proposed AFTC a suitable candidate for applications, e.g., Ultraviolet Spectroscopic Explorer Satellite accident, where the actuators malfunctioned during the mission [6].

The rest of the paper comprises the following sections: Section two describes the spacecraft attitude

Received: July 2022, Accepted: December 2022
Correspondence to: Rouzbeh Moradi. Assistant Professor, Faculty of Aerospace Department, Urmia University of Technology, P.O. Box: 57166-419, Urmia, Iran. E-mail: rouzbeh_moradi@uut.ac.ir
doi: 10.5937/fme2301039M

dynamics and base controller structure. Section three presents the CLS, RM, and stability analysis. Section four explains the fault scenario. Section five defines SP and ART. Section six presents simulations to show the advantages of the proposed method. Finally, section seven ends the paper with a conclusion.

2. SPACECRAFT ATTITUDE DYNAMICS AND CONTROLLER STRUCTURE

The rotational dynamic of rigid spacecraft in the principal coordinate system is described by (1-7) [22]:

$$\dot{q}_0 = 0.5\left(-\omega_1 q_1 - \omega_2 q_2 - \omega_3 q_3\right) \tag{1}$$

$$\dot{q}_1 = 0.5\left(\omega_1 q_0 + \omega_3 q_2 - \omega_2 q_3\right) \tag{2}$$

$$\dot{q}_2 = 0.5\left(\omega_2 q_0 - \omega_3 q_1 + \omega_1 q_3\right) \tag{3}$$

$$\dot{q}_3 = 0.5\left(\omega_3 q_0 + \omega_2 q_1 - \omega_1 q_2\right) \tag{4}$$

$$\dot{\omega}_1 = \left(\frac{I_2 - I_3}{I_1}\right)\omega_2\omega_3 + u_1' \tag{5}$$

$$\dot{\omega}_2 = \left(\frac{I_3 - I_1}{I_2}\right)\omega_1\omega_3 + u_2' \tag{6}$$

$$\dot{\omega}_3 = \left(\frac{I_1 - I_2}{I_3}\right)\omega_1\omega_2 + u_3' \tag{7}$$

Equations (8-10) show the relation between (u_1,u_2,u_3) and (u_1',u_2',u_3'):

$$u_1' = u_1/I_1 \tag{8}$$

$$u_2' = u_2/I_2 \tag{9}$$

$$u_3' = u_3/I_3 \tag{10}$$

The following saturation function represents the range of the control inputs:

$$\text{sat}\left(u_i\right) = \begin{cases} u_i & \text{if} \quad -u_{\max} \le u_i \le u_{\max} \\ u_{\max} & \text{if} \quad u_i > u_{\max} \\ -u_{\max} & \text{if} \quad u_i < -u_{\max} \end{cases} \tag{11}$$

According to [22], the following equation is present among the quaternions:

$$q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1 \tag{12}$$

Considering constraint (12), knowing q_1,q_2,q_3, the other elements of the quaternion vector (q_0) will be determined. Considering this fact, the output vector is selected as:

$$\mathbf{y} = \begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix}^T \tag{13}$$

In this case, the controller design will be easier because the number of controlled outputs (three) is equal to the number of inputs (three) [23].

The control inputs will appear in the second time derivative of **y**. Consequently, the total relative degree (six) and the number of states ($[q_1,q_2,q_3,\omega_1,\omega_2,\omega_3]$) will be equal. Therefore, internal dynamic does not exist, and it is possible to use input-output linearization easily [23].

Taking the second time derivative of **y** results in the following equations:

$$\ddot{q}_1 = \chi_1 + \frac{1}{2}\left(q_0 u_1' - q_3 u_2' + q_2 u_3'\right) \tag{14}$$

$$\ddot{q}_2 = \chi_2 + \frac{1}{2}\left(q_3 u_1' + q_0 u_2' - q_1 u_3'\right) \tag{15}$$

$$\ddot{q}_3 = \chi_3 + \frac{1}{2}\left(-q_2 u_1' + q_1 u_2' + q_0 u_3'\right) \tag{16}$$

where

$$\chi_1 = -\frac{1}{4}q_1\sum_{i=1}^{3}\omega_i^2 + \frac{1}{2}\left(G_1 q_0\omega_2\omega_3 - G_2 q_3\omega_1\omega_3 + G_3 q_2\omega_1\omega_2\right) \tag{17}$$

$$\chi_2 = -\frac{1}{4}q_2\sum_{i=1}^{3}\omega_i^2 + \frac{1}{2}\left(G_1 q_3\omega_2\omega_3 + G_2 q_0\omega_1\omega_3 - G_3 q_1\omega_1\omega_2\right) \tag{18}$$

$$\chi_3 = -\frac{1}{4}q_3\sum_{i=1}^{3}\omega_i^2 + \frac{1}{2}\left(-G_1 q_2\omega_2\omega_3 + G_2 q_1\omega_1\omega_3 + G_3 q_0\omega_1\omega_2\right) \tag{19}$$

And

$$G_1 = \left(I_2 - I_3\right)/I_1 \tag{20}$$

$$G_2 = \left(I_3 - I_1\right)/I_2 \tag{21}$$

$$G_3 = \left(I_1 - I_2\right)/I_3 \tag{22}$$

Using feedback linearization, these equations will be transformed into the following LTI form:

$$\ddot{q}_1 = u_1'' \tag{23}$$

$$\ddot{q}_2 = u_2'' \tag{24}$$

$$\ddot{q}_3 = u_3'' \tag{25}$$

Asymptotic stability of CLS will be guaranteed if the components of $\mathbf{u}''$ are [24]:

$$u_1'' = \ddot{q}_{1,d} - k_{\dot{q}_1}\left(\dot{q}_1 - \dot{q}_{1,d}\right) - k_{q_1}\left(q_1 - q_{1,d}\right) \tag{26}$$

$$u_2'' = \ddot{q}_{2,d} - k_{\dot{q}_2}\left(\dot{q}_2 - \dot{q}_{2,d}\right) - k_{q_2}\left(q_2 - q_{2,d}\right) \tag{27}$$

$$u_3'' = \ddot{q}_{3,d} - k_{\dot{q}_3}\left(\dot{q}_3 - \dot{q}_{3,d}\right) - k_{q_3}\left(q_3 - q_{3,d}\right) \tag{28}$$

Consequently, the state-feedback control laws will be:

$$\begin{bmatrix} u_1' \\ u_2' \\ u_3' \end{bmatrix} = \begin{bmatrix} q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix}^{-1}\begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} \tag{29}$$

where

$$f_1 = 2\left(u_1'' - \chi_1\right) \tag{30}$$

$$f_2 = 2\left(u_2'' - \chi_2\right) \tag{31}$$

$$f_3 = 2\left(u_3'' - \chi_3\right) \tag{32}$$

Euler angles and quaternions are related based on the following equations [22]:

$$q_0 = \cos\left(\frac{\psi}{2}\right)\cos\left(\frac{\theta}{2}\right)\cos\left(\frac{\phi}{2}\right) + \sin\left(\frac{\psi}{2}\right)\sin\left(\frac{\theta}{2}\right)\sin\left(\frac{\phi}{2}\right) \tag{33}$$

$$q_1 = \cos\left(\frac{\psi}{2}\right)\cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\phi}{2}\right) - \sin\left(\frac{\psi}{2}\right)\sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\phi}{2}\right) \tag{34}$$

$$q_2 = \cos\left(\frac{\psi}{2}\right)\sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\phi}{2}\right) + \sin\left(\frac{\psi}{2}\right)\cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\phi}{2}\right) \tag{35}$$

$$q_3 = \sin\left(\frac{\psi}{2}\right)\cos\left(\frac{\theta}{2}\right)\cos\left(\frac{\phi}{2}\right) - \cos\left(\frac{\psi}{2}\right)\sin\left(\frac{\theta}{2}\right)\sin\left(\frac{\phi}{2}\right) \tag{36}$$

$$\tan(\phi) = \frac{2\left(q_2 q_3 + q_0 q_1\right)}{q_0^2 - q_1^2 - q_2^2 + q_3^2} \tag{37}$$

$$\sin(\theta) = -2\left(q_1 q_3 - q_0 q_2\right) \tag{38}$$

$$\tan(\psi) = \frac{2\left(q_1 q_2 + q_0 q_3\right)}{q_0^2 + q_1^2 - q_2^2 - q_3^2} \tag{39}$$

3. AFTC MECHANISM

3.1 CLS

Fig. 1 shows the CLS. As shown in this figure, the controller inputs are the quaternion error (and its derivative), and angular velocity vectors (according to (29)). FDD uses the plant input and output to detect the fault.

RM receives FDD data and produces $\mathbf{q}_d$ and its derivatives via the procedure shown in Fig. 2.

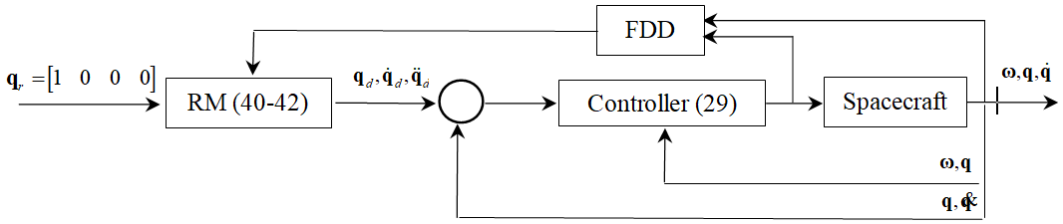
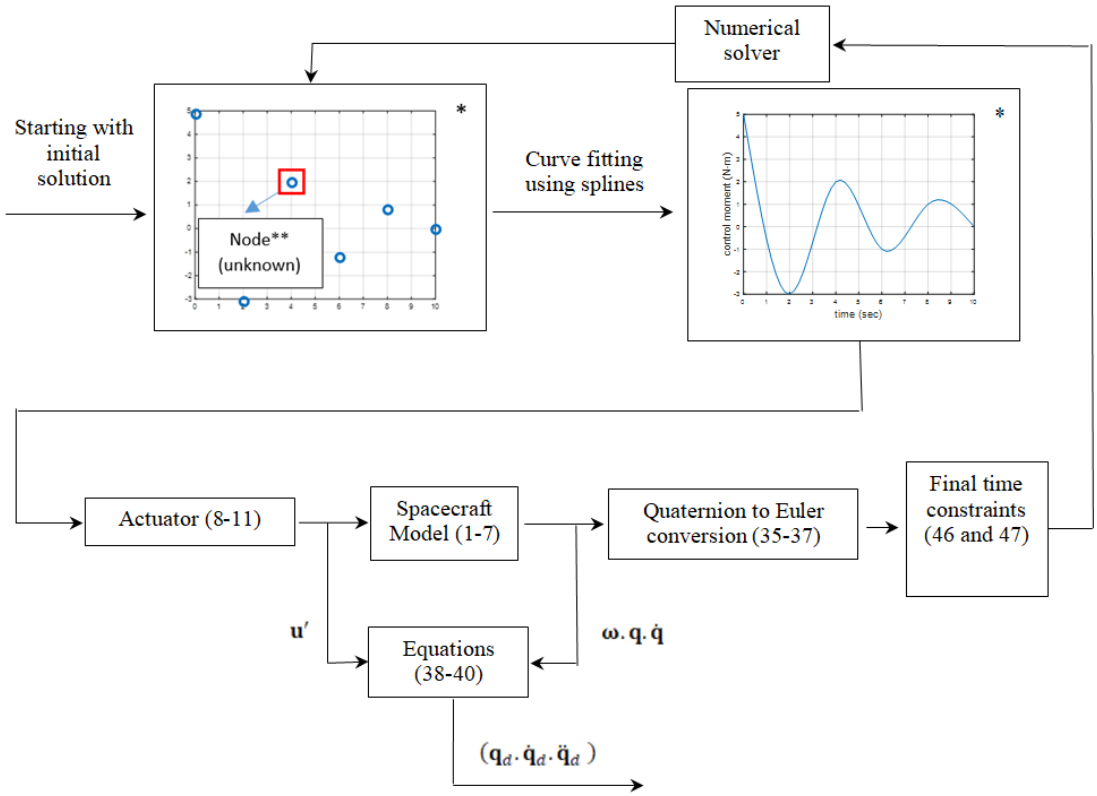


Figure 1. AFTC structure.



* Data shown in the figures are typical.

** The nodes are the unknowns of the problem.

Figure 2. RM structure.

3.2 RM structure

Fig. 2 shows the RM structure. According to Fig. 2, RM works as follows:

The process begins with an "initial solution". The numerical solver uses this "initial solution" to produce the continuous open-loop control commands via the cubic splines. Then, based on FDD data, an actuator model is constructed, and finally, based on the response of the spacecraft model, final time constraints are evaluated. One loop execution (LE) starts at $t = 0$ and ends at $t = t_f$. For example, in Fig. 2, one LE starts at t = 0 and ends at t = 10s. After the completion of one loop, the other loop begins. This procedure continues until the stopping criteria are satisfied. This process is equivalent to one simulation. Therefore, one simulation comprises several LEs.

Then, based on obtained open-loop control commands ($\mathbf{u}$), and therefore ($\mathbf{u}'$) and state trajectories ($\boldsymbol{\omega}, \mathbf{q}, \dot{\mathbf{q}}$), RM produces desired quaternions and their derivatives ($\mathbf{q}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d$) according to the following equations:

$$\begin{aligned}\ddot{q}_{1,d} &= k_{\dot{q}_1}\left(\dot{q}_1 - \dot{q}_{1,d}\right) + k_{q_1}\left(q_1 - q_{1,d}\right) \\ &+ \chi_1 + \frac{1}{2}\left(q_0 u_1' - q_3 u_2' + q_2 u_3'\right)\end{aligned} \tag{40}$$

$$\begin{aligned}\ddot{q}_{2,d} &= k_{\dot{q}_2}\left(\dot{q}_2 - \dot{q}_{2,d}\right) + k_{q_2}\left(q_2 - q_{2,d}\right) \\ &+ \chi_2 + \frac{1}{2}\left(q_3 u_1' + q_0 u_2' - q_1 u_3'\right)\end{aligned} \tag{41}$$

$$\begin{aligned}\ddot{q}_{3,d} &= k_{\dot{q}_3}\left(\dot{q}_3 - \dot{q}_{3,d}\right) + k_{q_3}\left(q_3 - q_{3,d}\right) \\ &+ \chi_3 + \frac{1}{2}\left(-q_2 u_1' + q_1 u_2' + q_0 u_3'\right)\end{aligned} \tag{42}$$

The value of each node and the value of the final time (t_f) are the unknowns of this problem. The interval (upper and lower values) of t_f is determined by the mission requirements (58). Equations (50) and (51) are used to determine the maximum and minimum values of the nodes.

According to this discussion, the initial solution has a direct influence on RM (and equivalently AFTC) performance. Using a mechanism to obtain an initial solution close to the final solution will increase the performance of AFTC.

Stopping criteria consist of two parts: 1-Nonlinear inequality constraints (48), (49), and 2-Maximum allo–wable NLE per simulation. Satisfaction of the stopping criteria shows that either inequality constraints are satisfied or NLE in a simulation exceeds the maximum allowable NLE per simulation.

Some points about cubic splines: The main reason to use cubic splines instead of other functions, such as polynomials, is that the coefficients of splines (nodes in Fig. 2) can be set within the upper and lower limits of the control commands. The accuracy of cubic spline interpolation is dependent on the number of nodes. The more nodes considered, the better will be the accuracy of splines. However, increasing the number of nodes will increase the number of variables and increase the complexity of the problem.

3.3 Stability analysis

If the following constraint equations are satisfied,

$$\mathbf{u}_{t=t_f} = \mathbf{0} \tag{43}$$

$$\omega_{t=t_f} = \mathbf{0} \tag{44}$$

$$\mathbf{q}_{t=t_f} = \begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix} \tag{45}$$

Then, based on (1-10), the following equalities will be satisfied:

$$\dot{\omega}_{t=t_f} = \mathbf{0} \tag{46}$$

$$\dot{\mathbf{q}}_{t=t_f} = \mathbf{0} \tag{47}$$

Therefore, angular velocity and quaternion vectors will reach the origin and remain there forever after $t = t_f$.

Fortunately, satisfying the constraint (43) is straightforward. It is sufficient to set the upper and lower bounds of the nodes at $t = t_f$ to zero. To satisfy the other two constraints (44) and (45), the following final time non-equality constraints are defined:

$$\max\left(|\omega_1|, |\omega_2|, |\omega_3|\right)_{t=t_f} \le \mathrm{TOL}_{\omega} \tag{48}$$

$$\max\left(|\phi|, |\theta|, |\psi|\right)_{t=t_f} \le \mathrm{TOL}_{Attitude} \tag{49}$$

Mission requirements specify angular velocity and attitude tolerances ((59) and (60)).

4. FAULT SCENARIO

As the fault model, it is assumed that the actuators have lost their effectiveness partially [15]:

$$\mathrm{sat}_n\left(u_i\right) = \begin{cases} u_i & \text{if } -u_{\max,\mathrm{n}} \le u_i \le u_{\max,\mathrm{n}} \\ u_{\max,\mathrm{n}} & \text{if } u_i > u_{\max,\mathrm{n}} \\ -u_{\max,\mathrm{n}} & \text{if } u_i < -u_{\max,\mathrm{n}} \end{cases} \tag{50}$$

$u_{\max,\mathrm{n}}$ is the post-fault actuator operating region and given by (51):

$$u_{\max,n} = a u_{\max} \tag{51}$$

a is a random number. More details about this parameter will be given in the simulation section.

5. SUCCESS PERCENTAGE AND ACTUAL RECONFIGURATION TIME

Two crucial parameters are introduced before analyzing the simulation results: SP and ART.

SP is the number of successful simulations per total number of simulations (for different faults and initial conditions) multiplied by 100:

$$\mathrm{SP} = \frac{\text{Number of successful simulations}}{\text{Total number of simulations}} \times 100 \tag{52}$$

Successful simulation is a simulation where the nonlinear inequality constraints (48), and (49) are satis–fied. As previously stated, each simulation consists of several LEs. A high value of SP shows a high proba–bility of success for RM.

NLE directly affects ART, i.e., as the NLE becomes larger, ART will become larger. Therefore, to make the AFTC easier to implement, NLE should be reduced.

In the simulation section, it will be shown that using the feedforward NN to approximate the initial solution will increase SP and decrease NLE to a large extent.

6. SIMULATION

The considered spacecraft has the following moments of inertia [25]:

$$I_1 = 449.5,\ I_2 = 264.6,\ I_3 = 312.5\ \text{kg.m}^2 \tag{53}$$

The controller coefficients are selected as follows:

$$k_{q_1} = k_{q_2} = k_{q_3} = 0.5 \tag{54}$$

$$k_{\dot{q}_1} = k_{\dot{q}_2} = k_{\dot{q}_3} = 1 \tag{55}$$

According to the appendix, these coefficients will make the tracking error converge to zero as time goes to infinity. Besides, based on the simulation results, these controller coefficients will lead to an acceptable system response.

Equation (56) gives the maximum torque that the healthy actuators can generate:

$$u_{\max} = 10\,\text{N.m} \tag{56}$$

According to the mission specifications, final time interval, angular velocity, and attitude tolerances are:

$$t_f \in [20 \quad 50]\text{sec} \tag{57}$$

$$\text{TOL}_{\omega} = 0.1\,\text{deg/ sec} \tag{58}$$

$$\text{TOL}_{Attitude} = 1\,\text{deg} \tag{59}$$

The maximum allowable NLE per simulation is 200. The spacecraft is assumed to be initially at rest i.e. $\boldsymbol{\omega}(0) = 0$. The initial conditions of desired quaternions (to integrate (40-42)) are:

$$\mathbf{q}_d(0) = [1 \quad 0 \quad 0 \quad 0]^T,\ \dot{\mathbf{q}}_d(0) = [0 \quad 0 \quad 0 \quad 0]^T$$

Equations (61) and (62) show the actuator effecti-veness coefficients and initial conditions considered for simulation:

$$a_1 = 0.11, a_2 = 0.78, a_3 = 0.29 \tag{60}$$

$$\phi_0 = -10.88\ \text{deg},\ \theta_0 = -2.29\ \text{deg}, \psi_0 = -3.43\ \text{deg} \tag{61}$$

Note 1: 200 simulations with random faults and initial conditions are used to train the NN. Therefore, the efficiency of the proposed method is evaluated for 200 simulations and not merely the values presented in (60) and (61).

First, the controller performance is demonstrated without adjusting the desired quaternions. Therefore, the desired quaternion vector is: $\mathbf{q}_d = [1 \quad 0 \quad 0 \quad 0]$. Based on (37-39), this is equivalent to $\phi_d = \theta_d = \psi_d = 0$. The controller's response, in terms of Euler angles, angular velocities, and control moments, is illustrated in Figs. 3-5:

As shown in Figs. 3 and 4, when $\phi_d = \theta_d = \psi_d = 0$, the controller is not able to satisfy the mission requirements (48) and (49). The reason is as follows: As seen in Fig. 5, u_2 and u_3 converge to the origin in a short time, while u_1 saturates all the time. Therefore, the controller will not be able to exploit u_1, u_2, u_3 to satisfy the mission requirements.

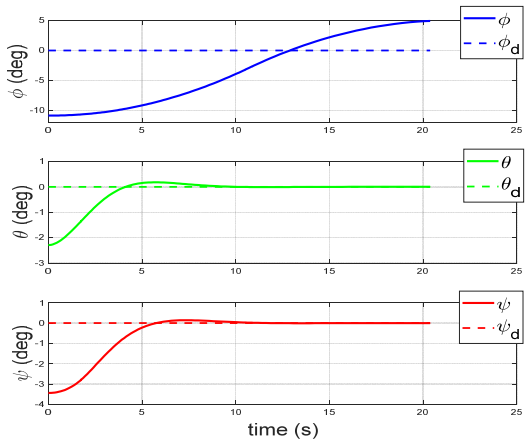


Figure 3. Euler angles for $\phi_d = \theta_d = \psi_d = 0$.

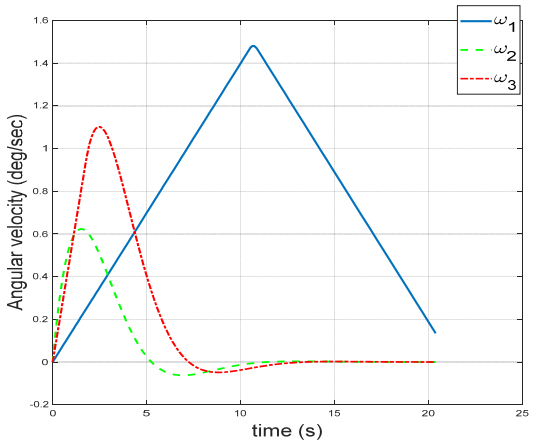


Figure 4. Angular velocities for $\phi_d = \theta_d = \psi_d = 0$.

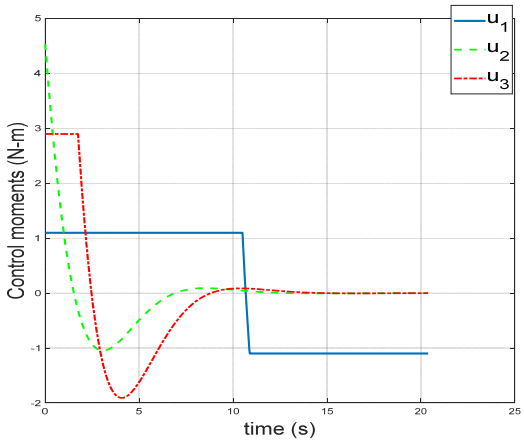


Figure 5. Control moments for $\phi_d = \theta_d = \psi_d = 0$.

In the second stage, RM (Fig. 2) is implemented. Five nodes in equal time intervals are considered in each axis between $t = 0$ and $t = t_f$. Since the value of nodes at $t = t_f$ should be zero (section 3-3), there will be 12 unknowns. The final time (t_f) is another unknown. Therefore, there will be 13 unknowns in this problem.

Euler angles, angular velocities, and control moments are illustrated in Figs. 6-8:

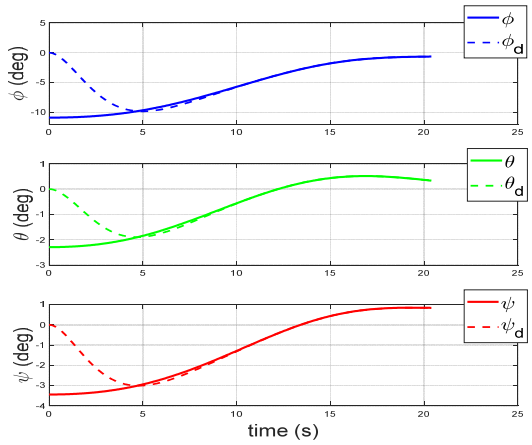


Figure 6. Corrected actual and desired Euler angles.

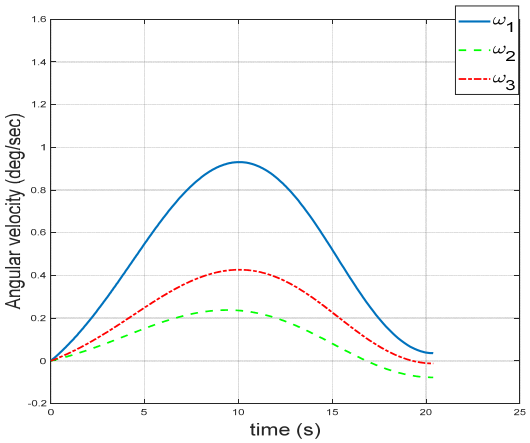


Figure 7. Corrected angular velocities.

According to Figs. 6-7, the proposed method can satisfy the mission constraints (48) and (49). Fig. 8 shows the control moments. Compared to Fig. 5, no saturation occurs, so the actuators operate normally.

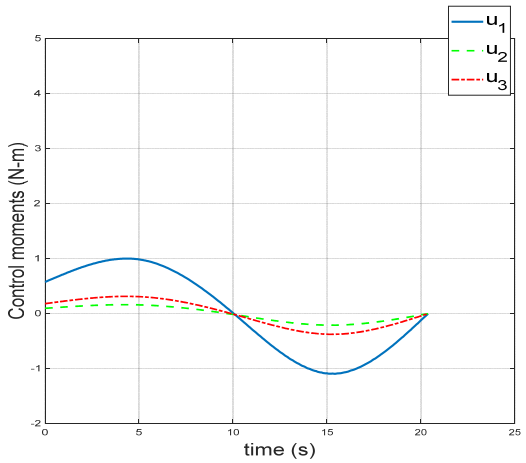


Figure 8. Corrected control moments.

The following will show the effect of using feedforward NN to increase the SP and decrease ART.

Note 2: R2020a neural network toolbox [21] is used to model and train the neural network.

Faults and initial conditions are considered to be normally distributed, with the following means and standard deviations:

For a_1,a_2,a_3: Mean = 0.5 and standard deviation = 0.1.

For initial Euler angles: Mean = 0 and standard deviation = 0.1 rad.

Note 3: The normal distribution assumption for faul–ts is considered in [26] and used in this paper.

6.1 Fault scenario (initial solution not adjusted)

In this scenario, the initial solution is: $[0_{1\times12}, 20s]^T$. The numerical solver uses this initial solution to obtain the desired quaternions (equivalently Euler angles) via the procedure shown in Fig. 2. After 200 simulations, SP and the average value of NLE per simulation will be 77% and 140, respectively. These data indicate that RM succeeded in 77% of the simulations.

After performing these 200 simulations, the set of inputs ($[a_1\ a_2\ a_3\ \square_0\ \theta_0\ \psi_0]^T$) and outputs (12 nodes and the final time) are used to train the feedforward NN. Training algorithm is Levenberg-Marquardt. This algo–rithm usually requires more memory and less time [21].

NN structure is shown in Fig. 9. As shown in this figure; there are 6 inputs, 1 hidden layer with 10 nodes, and 13 outputs.

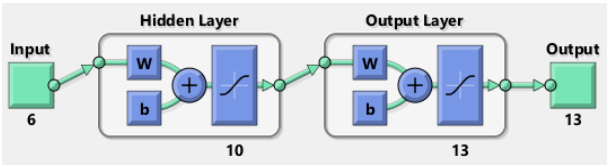


Figure 9. Structure of the trained feedforward NN

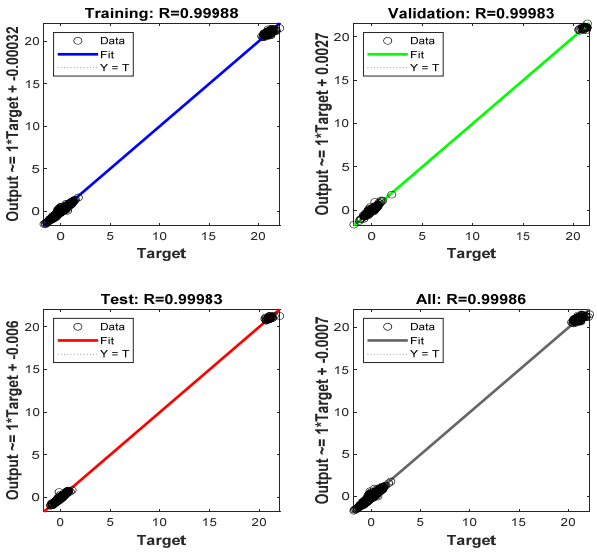


Figure 10. Neural network training regression

To evaluate NN performance, training regression is presented in Fig. 10. As shown in this figure; regression is near 1. NN training performance is also shown in Fig. 11. As shown in this figure; validation performance is best at epoch 11.

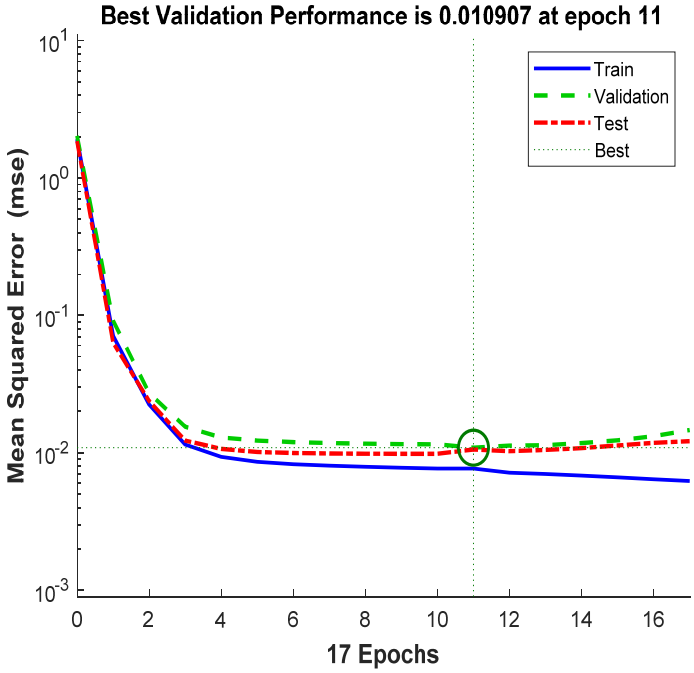


Figure 11. NN training performance

6.2 Second Scenario (NN Used to Adjust the Initial Solution)

The faults and initial conditions are identical to those in the previous scenario. However, trained feedforward NN is used in this scenario to determine the initial solu–tion. Based on the simulations, SP and the average value of NLE per simulation are 96.5% and 36.97, respectively.

Comparing the results of these scenarios, it is concluded that using the feedforward NN to approxi–mate the initial solution increases SP and decreases the average value of NLE per simulation by about 25% and 73%, respectively.

These improvements in SP and the average value of NLE (equivalently ART) per simulation are desirable from a practical point of view.

As stated in the introduction, the proposed method can be utilized in cases similar to Ultraviolet Spectro–scopic Explorer Satellite accidents where actuators malfunctioned.

7. CONCLUSION

Active fault-tolerant attitude stabilization of spacecraft was considered in this paper. In comparison to PFTCs, AFTCs are less conservative and can deal with a broader range of faults. However, AFTCs are harder to implement. To solve this challenge, ART and SP were explicitly considered in the AFTC design, and FEEDFORWARD NN was used to decrease the former and increase the latter. This contribution makes the proposed controller suitable for applications.

REFERENCES

[1] Dančuo, Z.Z, Rašuo, B.P., Bengin, A. Č., Zeljković, V.I.: Flight to Mars: Envelope Simulation in a Ground Based High-performance Human Centrifuge, FME Transactions, Vol. 46, No. 1, pp. 25-32, 2018. doi: 10.5937/fmet1801001D.

[2] Marčeta, D., Šegan, S., Rašuo, B.: Influence of seasonal cycles in Martian atmosphere on entry, descent and landing sequence, Acta Astronautica, Vol. 98, May–June 2014, doi. 10.1016/j.actaastro .2014.02.001

[3] Marčeta, D., Šegan, S., Rašuo, B., Racković Babić, K.: Meteoroid environment on the transfer trajectories to Mars, Aerospace Science and Technology, Vol. 56, September 2016, pp. 14-21, doi: 10.1016/j.ast.2016.05.007

[4] Tafazoli, M.: A study of on-orbit spacecraft failures, Acta Astronautica, Vol. 64, Issues 2–3, pp. 195-205, 2009.

[5] Shen, Q., Wang, D., Zhu, S., Poh, K.: Finite-time fault-tolerant attitude stabilization for spacecraft with actuator saturation, IEEE Trans Aerosp Electron Syst, Vol. 51, Issue: 3, 2015.

[6] Yin, S., Xiao, B., X. Ding, S., Zhou, D.: A Review on Recent Development of Spacecraft Attitude Fault Tolerant Control System, IEEE Trans. Ind. Electron, Vol. 63, Issue: 5, 2016.

[7] Safa, A., Baradarannia, M., Kharrati, H., Khanmohammadi, S.: Attitude stabilization of a rigid spacecraft with actuator delay and fault, Trans. Inst. Meas. Control, Vol. 40, Issue: 7, pp. 2340-2351, 2018.

[8] Zhu, S., Wang, D., Shen, Q and Poh, K.: Satellite Attitude Stabilization Control with Actuator Faults, J Guid Control Dyn, Vol. 40, No. 5, 2017.

[9] Zhang, X., Gao, Z., Qian, M., Zhou, Z.: Active Fault Tolerant Attitude Control for Rigid Spacecraft with Actuator LOE Fault and Saturation Constraint, in: *Chinese Control and Decision Conference (CCDC)*, pp. 9-11 June 2018.

[10] Gao, Z., Cheng, P., Qian, M., Jiang, G., Lin, J.: Active fault-tolerant control approach design for rigid spacecraft with multiple actuator faults,

Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, Vol. 232, Issue: 10, pp. 1365-1378, 2018.

[11] Shen, Q., Yue, C., Goh, C.H., Wang, D: Active Fault-Tolerant Control System Design for Spacecraft Attitude Maneuvers with Actuator Saturation and Faults, IEEE Trans. Ind. Electron, Vol. 66, Issue: 5, May 2019.

[12] Shao, X., Hu, Q., Shi, Y., Jiang, B.: Fault-Tolerant Prescribed Performance Attitude Tracking Control for Spacecraft Under Input Saturation, IEEE Trans Control Syst Technol, Vol. 28, Issue: 2, 2020.

[13] Lee, D: Fault-tolerant finite-time controller for attitude tracking of rigid spacecraft using intermediate quaternion, IEEE Trans Aerosp Electron Syst, Vol. 57, Issue: 1, 2021.

[14] Wang, Z., Su, Y., Zhang, L.: Fixed-Time Fault-Tolerant Attitude Tracking Control for Rigid Spacecraft, J. Dyn. Sys., Meas., Control, 142(2): 11 pages, 2019.

[15] Miksch, T., Gambier, A.: Fault-tolerant control by using lexicographic multi-objective optimization, in *8th Asian Control Conference (ASCC)*, 2011.

[16] Zhang, Y., Jiang, J.: Bibliographical review on reconfigurable fault-tolerant control systems. Annu Rev Control, Vol. 32, Issue 2, pp. 229-252, 2008.

[17] Abbaspour, A., Mokhtari, S., Sargolzaei, A., Yen, K.K.: A Survey on Active Fault-Tolerant Control Systems, Electronics, 9(9), 1513, 2020.

[18] Jiang, J., Yu, X: Fault-tolerant control systems: A comparative study between active and passive approaches, Annu Rev Control, Vol. 36, Issue 1, pp. 60-72, 2012.

[19] Lunze, J., Richter, J.H.: Reconfigurable Fault-tolerant Control: A Tutorial Introduction, Eur. J. Control, Vol. 14, Issue 5, pp. 359-386, 2008.

[20] T. Saeed et al.: Fault Adaptive Routing in Metasurface Controller Networks, in: *International Workshop on Network on Chip Architectures (NoCArc)*, pp. 1-6, 2018.

[21] MATLAB and Neural Network Toolbox Release 2020a, The MathWorks, Inc., Natick, Massachusetts, United States.

[22] Zipfel, P. H.: *Modeling and Simulation of Aerospace Vehicle Dynamics,* American Institute of Aeronautics and Astronautics, 2014.

[23] Slotine, J-J., Li, W.: *Applied Nonlinear Control*, Prentice Hall, 1991.

[24] Benosman, M., Lum, K.Y.: Online reference reshaping and control reallocation for nonlinear fault tolerant control, IEEE Trans. Control Syst. Technol, Vol. 17, Issue: 2, 2009.

[25] Wang, D., Jia, Y., Jin, L., Xu, Sh.: Control analysis of an underactuated spacecraft under disturbance, Acta Astronautica, Vol. 83, pp. 44-53, 2013.

[26] Zhang, Y., Li, X. R.: Detection and diagnosis of sensor and actuator failures using IMM estimator, IEEE Trans. Aerosp. Electron. Syst, Vol. 34, Issue: 4, pp. 1293-1313, 1998.

[27] Golnaraghi, F., Kuo, B.: *Automatic control systems*, McGraw Hill, 10th edition, 2017.

NOMENCLATURE

$[q_0, q_1, q_2, q_3]$	Quaternion vector
$[q_{1,d}, q_{2,d}, q_{3,d}]$	Desired quaternion vector
$[\omega_1, \omega_2, \omega_3]$	Angular velocity vector
I_1, I_2, I_3	Principal moments of inertia
$[u_1, u_2, u_3]$	Control moment vector
$[u_1', u_2', u_3']$	Control input vector
Sat	Saturation function
$[k_{q_1}, k_{q_2}, k_{q_3}] \in R^+$	Controller coefficients
$[k_{\dot{q}_1}, k_{\dot{q}_2}, k_{\dot{q}_3}] \in R^+$	
ϕ, θ, ψ	Euler angles (roll, pitch, yaw)
t_f	Final time
a	Actuator effectiveness Coefficient

Subscripts

d	Desired
f	Final

Acronyms

ACS: Attitude Control System
AFTC: Active Fault Tolerant Control
ART: Actual Reaction Time
CLS: Closed-loop system
CRT: Critical Reaction Time
FDD: Fault Detection and Diagnosis
FTC: Fault Tolerant Control
LE: Loop Execution
LTI: Linear Time Invariant
NLE: Number of Loop Executions
NN: Neural Network
PFTC: Passive Fault Tolerant Control
RM: Reconfiguration mechanism
SP: Success Percentage

APPENDIX

Considering (23-25) and (26-28):

$$\ddot{q}_1 = \ddot{q}_{1,d} - k_{\dot{q}_1}\left(\dot{q}_1 - \dot{q}_{1,d}\right) - k_{q_1}\left(q_1 - q_{1,d}\right) \tag{62}$$

$$\ddot{q}_2 = \ddot{q}_{2,d} - k_{\dot{q}_2}\left(\dot{q}_2 - \dot{q}_{2,d}\right) - k_{q_2}\left(q_2 - q_{2,d}\right) \tag{63}$$

$$\ddot{q}_3 = \ddot{q}_{3,d} - k_{\dot{q}_3}\left(\dot{q}_3 - \dot{q}_{3,d}\right) - k_{q_3}\left(q_3 - q_{3,d}\right) \tag{64}$$

The tracking error is defined as follows:

$$e_i = q_i - q_{i,d}, \quad i = 1,2,3 \tag{65}$$

Considering (65), (62-64) can be rewritten as:

$$\ddot{e}_i + k_{\dot{q}_i}\dot{e}_i + k_{q_i}e_i = 0, \quad i = 1,2,3 \tag{66}$$

The subscript i will be dropped for simplicity. Taking the Laplace transform of this equation:

$$E(s) = \frac{se(0) + \dot{e}(0) + k_{\dot{q}} e(0)}{s^2 + k_{\dot{q}} s + k_q} \tag{67}$$

Considering the proposed controller coefficients, $sE(s)$ will have no pole with zero or positive real part.

Therefore, according to the final value theorem [27]:

$$\lim_{t \to \infty} e(t) = \lim_{s \to 0} sE(s) = 0 \tag{68}$$

and the tracking error will converge to zero as time goes to infinity.

ПОВЕЋАЊЕ ПЕРФОРМАНСИ АКТИВНЕ КОНТРОЛЕ ОТПОРНЕ НА ГРЕШКЕ У СВЕМИРСКОЈ ЛЕТЕЛИЦИ КОРИШЋЕЊЕМ НЕУРОНСКИХ МРЕЖА

Р. Моради

Квар актуатора представља изазов за контролу положаја свемирске летелице. Контрола отпорна на грешке (активна или пасивна) се често користи за превазилажење овог изазова. Активне методе имају боље перформансе од пасивних метода и могу управљати ширим спектром грешака. Међутим, њихово спровођење је теже. Један од разлога за ову потешкоћу је критично време реакције. Систем може постати непоправљив ако стварно време реакције постане веће од критичног времена реакције. Овај рад предлаже коришћење неуронске мреже унапред да би се смањило стварно време реакције у активној контроли свемирских летелица која је толерантна на грешке. Поред овог побољшања, коришћење неуронске мреже унапред може повећати проценат успеха. Проценат успешности је однос успешних симулација према укупном броју симулација. Резултати симулације показују да се за 200 симулација са случајним грешкама и почетним условима, стварно време реакције смањује за 73%, а проценат успешности расте за 25%. На основу ових резултата, предложени контролер је добар кандидат за практичну примену.