Nonlinear Fault Recovery of a Quadrotor Using Progressive Accommodation

Hajer MLAYEH[1]*, Sahbi GHACHEM[2], Othman NASRI[3], Kamel Ben OTHMAN[4]

[1] Université de Sousse, Ecole Nationale d'Ingénieurs de Sousse, 4023, Sousse, Tunisia; University of Monastir, National Engineering School of Monastir, LASEE Research Unit, Ibn Jazzar 5019, Monastir, Tunisia
mlayehhajer@gmail.com (**Corresponding author*)

[2] LARA Automatic, National Engineering School of Tunis, University of Tunis. Tunis, BP37 Belvedere 1002, Tunisia

[3] Université de Sousse, Ecole Nationale d'Ingénieurs de Sousse, LATIS- Laboratory of Advanced Technology and Intelligent Systems, 4023, Sousse, Tunisia

[4] University of Monastir, National Engineering School of Monastir, LASEE Research Unit, Ibn Jazzar 5019, Monastir, Tunisia

Abstract: This paper proposes a nonlinear (NL) fault-tolerant control (FTC) strategy for stabilizing the attitude of a quadrotor affected by multiple rotor failures. The recovery process relies mainly on progressive accommodation (PA) strategy. The latter uses the Newton-Raphson (NR) algorithm in order to solve the state-dependent Riccati Equation (SDRE) in a recursive way. Simulations are performed using the NL model of the quadrotor in fault-free and post-fault cases. Both PA and the direct SDRE methods are employed so that the attitude control of the quadrotor can be stabilized. The results show that the PA strategy performs better than the classical method when it comes to convergence speed and to preserving the control objectives of the nominal system.

Keywords: NL FTC, PA, Quadrotor, SDRE, Rotor failures.

1. Introduction

The reliability and dependability of critical systems are of paramount importance especially for aerospace systems such as unmanned aerial vehicles (UAVs) and airplanes. Indeed, detecting and identifying a technical fault is no longer sufficient, UAVs must be able to instantly correct or tolerate the respective anomaly in order to fulfill the objectives it was designed for or at least to ensure stability. The system is a so-called fault-tolerant control system (FTCS).

Various FTC strategies were reported in specialised literature: linear matrix inequality (Andrade et al., 2017), the pseudo-inverse (Tchon & Janiak, 2009), multiple model (Pandey, Kar & Mahanta, 2017) and adaptive control methods (Yu et al., 2019), robust controls (Zhi & al., 2018), the Algebraic Riccati Equation (ARE), the Hamilton-Jacobi Equation (HJE), the sliding mode control (SMC) (Zhang et al., 2018) and intelligent controls based on artificial neural network (Yen & Ho, 2004). The "suitable" technique for FTC depends on the type of system considered and the nature/gravity of the fault.

(Andrade et al., 2017) employed the LMI technique to guarantee the stability and the robust performance of an F-16 aircraft and to improve its damping factor. Similarly, (Zhi & al., 2018), presented a robust FTC technique in order to deal with disturbances and unmodelled dynamics of an over-actuated system. However, despite the effectiveness and the robustness of these methods, they remain "passive". That means that they are pre-computed offline and specific to a presumed set of faults. Indeed, unlike "active" approaches, passive ones use neither a fault detection and identification (FDI) block, nor controller reconfiguration which puts the global stability of the system at risk. Various active approaches can be mentioned such as (Tchon & Janiak, 2009; Yen & Ho, 2004). However, they lack the important property of the SDRE-based technique which allows the designer to trade-off between control accuracy and control effort using weighting matrices.

Moreover, with regard to nonlinear systems, it is hardly feasible to implement FTC laws using the HJE, especially for complicated systems. Several numerical algorithms were proposed to provide an approximate solution such as the iterative "Heuristic Dynamic Programming" (HDP) approach (Mu, Wang & He, 2018). However, the latter suffers from a heavy computational load. The SDRE approach (Korayem & Nekoo, 2015),

https://doi.org/10.24846/v29i1y202007

on the other hand, does not require the resolution of the HJE. It was conceived as an extension of the ARE in order to solve nonlinear optimal control problems in real time while conserving the same state-space “linear aspect”.

Owing to the last property, the SDRE technique has the advantage of combining the simplicity of linear methods and the effectiveness of NL ones in dealing with complicated systems, which is generally not provided by other FTC strategies.

As UAVs are concerned, their field of application has expanded significantly in recent years to include industrial applications, agriculture, security (such as border and coastal surveillance, traffic surveillance and protection of sensitive sites), delivery (such as the transport of medicines in urban environments, food and postal deliveries), etc… Consequently, the reliability of UAVs has gained increased attention with a focus on the use of FTC strategies. (Emran & Najjaran, 2018) presented a high-quality review of FTC for UAVs. (Merheb, Noura & Bateman, 2014) employed the SMC in order to stabilize the quadrotor in the degraded mode, but they considered a partial loss of only one motor. (Saied et al., 2015) referred to a total failure of one motor which was compensated by controlling the built-in redundant motor instead of the failing one. This redundancy allows full controllability of the damaged UAV, but its drawbacks are the additional costs it incurs and the complexity of its architecture.

Moreover, the methods considered in the aforementioned papers apply the reconfigurable control right after the fault diagnosis is established, which is not realistic and assumes the system be “ideal”. During that delay (between the fault detection and the application of the new control), the post-fault system is still driven by the nominal control and can face the risk of instability. The theory of the PA was conceived for minimizing these risks.

(Ciubotaru & Staroswiecki, 2009) presented a comparison between two PA strategies: The NR algorithm and the Matrix Sign Function (MSF). Simulations have shown that the NR algorithm needs more steps to converge but has a more satisfactory performance concerning the cost functional value and the closed loop admissibility. The work of (Menon et al., 2002) also features a good comparison between the iterative methods of “Kleinman/NR” and “the discrete time transformation”. The authors used 3 different processors to evaluate the execution time of the 2 above-mentioned algorithms. The “Kleinman” or “NR” method yielded a minimum computing time which proved that it is suitable for real-time applications.

Specialised literature includes several works based on the PA such as (Ciubotaru & Staroswiecki, 2006; Staroswiecki, Yang & Jiang, 2007) where the authors used the NR algorithm in order to solve the ARE in a recursive way, and applied it to the linear model of an aircraft. However, this linear approximation, despite its simplicity, becomes non-valid for certain initial conditions (ICs). This happens when the fault occurs near the boundary of the linearization-validity domain. A solution to this problem was presented by (Raharijaona et al., 2009; Ghachem, Benothman & Benrejeb, 2012) who used the NL PA, but its application was limited to first- and second-order equations and never involved a real NL system. Besides, in all the aforementioned studies, only one actuator failure was considered.

This paper proposes a NL PA for a quadrotor’s recovery process when partial failures affect its four rotors simultaneously. The NL modelling of the aforementioned system is presented, as well as the simulations of the healthy, affected, and recovered system using both the PA and the direct SDRE methods.

The remainder of this paper is organized as follows. Section 2 describes the UAV and its dynamics as well as the control strategy adopted for stabilizing the attitude of the quadrotor in normal conditions. Section 3 presents the system recovery process based on the iterative strategy of PA. Before the conclusion, section 4 sets forth the simulations and analysis of the healthy, affected and recovered system, with a comparison between the iterative and direct methods.

2. Quadrotor Modelling and Control Strategy

First, this section presents an overview of the quadrotor and its dynamics. Then, it sets forth the control strategy employed for stabilizing the quadrotor attitude.

2.1 Quadrotor Dynamics

The UAV analysed in this paper is the "DraganFly" quadrotor. (Bresciani, 2008) presented a basic structure of the UAV with its body and global coordinate system, and the Euler angles of roll, pitch and yaw. The quadrotor is composed of counter-rotating motors arranged in pairs along each of its arms. If all motors are spinning at the same speed, this counter-rotation allows the vehicle to maintain a constant heading while hovering. Moreover, some assumptions have to be made for this model:

- The atmosphere or wind effects are negligible.
- The quadrotor is equipped with an inertial measurement unit (IMU) that provides the necessary output measurements related to the body frame.
- The Euler angles are small enough to consider that angular velocities (p,q,r) are equal to the derivatives of Euler angles $(\dot{\varphi},\dot{\theta},\dot{\psi})$.

Based on the Newton-Euler formalism (Bresciani, 2008), the following rotational dynamics can be obtained for the quadrotor:

$$\begin{cases} \ddot{\varphi} = \frac{I_y - I_z}{I_x}\dot{\theta}\dot{\psi} + \frac{L}{I_x}U_1 + \frac{I_r}{I_x}\dot{\theta}U_4 \\ \ddot{\theta} = \frac{I_z - I_x}{I_y}\dot{\theta}\dot{\varphi} + \frac{L}{I_y}U_2 + \frac{I_r}{I_y}\dot{\varphi}U_4 \\ \ddot{\psi} = \frac{I_x - I_y}{I_z}\dot{\theta}\dot{\varphi} + \frac{1}{I_z}U_3 \end{cases} \tag{1}$$

φ,θ,ψ denote the roll, pitch and yaw angles, respectively. I_x, I_y, I_z are the body moments of inertia and I_r is the rotational moment of inertia around the propeller axis. L is the length of the arm (measured from the center of mass) and $U_{1..4}$ are the control inputs of the system defined by:

$$\begin{cases} U_1 = b\left(W_2^2 - W_4^2\right) \\ U_2 = b\left(W_1^2 - W_3^2\right) \\ U_3 = cb\left(-W_1^2 + W_2^2 - W_3^2 + W_4^2\right) \\ U_4 = \left(-W_1 + W_2 - W_3 + W_4\right) \end{cases} \tag{2}$$

Where W_i denote the speed of the rotor i $(i = 1..4)$ and b, c are the thrust and drag coefficients, respectively.

2.2 Control Strategy

This section presents the SDRE method which is an optimal control strategy that was adopted for stabilizing the quadrotor. Let us first consider equation (1) in the following nonlinear form:

$$\dot{x} = F\left(x\right) + G\left(x\right)u \tag{3}$$

Where $x \in \Re^n$ and $u \in \Re^m$ are the state and control vectors respectively and F, G are nonlinear functions. The optimal control problem is designed using the cost function (4). The aim is to find a control law that ensures the stability of the closed-loop system and drives the state $x(t)$ to 0 (as $t \to \infty$), while minimizing the following performance index:

$$J\left(x\right) = \int_0^\infty \left(x^T Q_p\left(x\right)x + u^T R_p\left(x\right)u\right)dt \tag{4}$$

$Q_p(x)$ and $R_p(x)$ are penalization matrices that affect the state and control actions. The concept of the SDRE method is to rewrite equation (3) in a state-dependent coefficient (SDC) form (5) where all matrices become a function of the current state.

$$\dot{x} = A\left(x\right)x + B\left(x\right)u \tag{5}$$

Where $A(x)$ and $B(x)$ are matrices of appropriate dimensions. For simplicity, it is assumed that all states are known. Thus, no observer is incorporated in this study. It's also worth mentioning that the choice of $A(x)$ and $B(x)$ is not unique, a suitable parameterization must ensure the system (5) controllability for any state denoted by x i.e., $rank\left(C_{trb}\left(x\right)\right)=n$ where C_{trb} is the controllability matrix:

$$C_{trb}\left(x\right)= \left[B\left(x\right) \quad A\left(x\right)B\left(x\right)\ldots A\left(x\right)^{n-1}B\left(x\right)\right] \tag{6}$$

Since all matrices are "state-dependent", the concept of the SDRE method consists in computing the control law "on line" at each sample time. This aspect makes this method advantageous as it avoids heavy computations and generates simple algorithms that can be conveniently implemented to UAVs. The following steps can summarize the design theory:

1. Obtain the measured states x from sensors.

2. Compute the model states $A(x)$, $B(x)$ and the weighting matrices $Q_p(x)$ and $R_p(x)$.

3. Form and solve the SDRE (7) evaluated at the current state (for $T\left(x\right)>0$):

$$A\left(x\right)^{T}T\left(x\right)+T\left(x\right)A\left(x\right)- T\left(x\right)B\left(x\right)R_p^{-1}\left(x\right)B\left(x\right)^{T}T\left(x\right)=-Q_p\left(x\right) \tag{7}$$

4. Evaluate the feedback control law:

$$u\left(x\right)=-R_p^{-1}\left(x\right)B^{T}\left(x\right)T\left(x\right)x\left(t\right) \tag{8}$$

It can be noticed that, if all former conditions are provided, the SDRE method ensures local asymptotic stability of the system presented in this paper.

3. The iterative FTC

This section sets forth the concept of the iterative strategy (PA), adopted to stabilize the quadrotor after failure occurrence.

The majority of FTC methodologies apply the accommodated control right after the diagnosis is established, as in (Saied et al., 2015), which is not right for real-time applications. Indeed, as it was mentioned in the Introduction, the transient period needed to compute the new FTC is necessary and critical because it can lead to system instability. Therefore, it is useful to shorten this period as much as possible by applying an iterative scheme known as the PA. The concept of the aforementioned theory is to use an iterative algorithm (such as the "NR"), applied online, that converges to the final FTC. Each control law produced by this algorithm ensures the stabilization of the system and improves as computation time increases.

The concept of the NR algorithm, in a NL system case, consists in starting with an initial matrix $T_0\left(x\right)\in\Re^{n\times n}$ and in solving the "Lyapunov" equation (9) in order to obtain the next iteration $T_{k+1}\left(x\right)$ where $(k=0,1,2,3...)$ represent the Newton steps.

$$T_{(k+1)}(x)(A(x)-B(x)G_k(x))+ (A(x)-B(x)G_k(x))^{T}T_{(k+1)}(x)= -Q_p(x)-G_k^{T}(x)R_p(x)G_k(x) \tag{9}$$

Where:

$$G_k\left(x\right)=R_p^{-1}\left(x\right)B\left(x\right)^{T}T_k\left(x\right) \tag{10}$$

Each time a new iteration $T_{k+1}\left(x\right)$ is computed, the control law is applied online using equation (8). The algorithm stops when $T_{k+1}(x)=T_k\left(x\right)$. Let $T_\infty(x)$ be the final solution to which the algorithm converges. It represents also the unique solution of the SDRE as defined in (7) which leads in fact to the optimal control (8).

As convergence and stability are concerned, (Kleinman, 1968) proved that the above-mentioned algorithm implies a monotone quadratic convergence once certain conditions are fulfilled:

- There exists a matrix $T_0\left(x\right)\in\Re^{n\times n}$ that is positive semi-definite, such that $\left(A\left(x\right)-B\left(x\right)G_0\left(x\right)\right)$ is of the Hurwitz type i.e., all its eigenvalues have negative real parts.

- The stabilizability and detectability conditions must be verified.

If the above assumptions hold, then all produced iterations $T_{k+1}\left(x\right)$ defined in (9) will ensure the stability of the closed-loop system and converge monotonically to the stabilizable solution $T_\infty\left(x\right)$. It can also be stressed that the choice of an accurate starting matrix makes the convergence faster.

4. Simulation Results

This section will focus on applying the SDRE-based control strategy (section 2.2) so that the quadrotor attitude in the fault-free case can be stabilized. Then the affected system will be simulated with and without the recovery process. In the latter case, both the iterative method (PA), and the direct one will be employed.

4.1 Controlling the Fault-free System

Let us consider $x=\left[\varphi \quad \dot{\varphi} \quad \theta \quad \dot{\theta} \quad \Psi \quad \dot{\Psi}\right]^T$ as the state vector. In order to apply the SDRE method, one should first make up the SDC matrices so that the system (1) can take the form expressed by equation (5). All calculations worked out, the following parameterization (11) can be obtained:

$$A(x)=\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & c1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & c2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & c3 & 0 & 0 & 0 & 0 \end{bmatrix}, \qquad (11)$$

$$B(x)=\begin{bmatrix} 0 & 0 & 0 & 0 \\ L/I_x & 0 & 0 & (I_r/I_x)x4 \\ 0 & 0 & 0 & 0 \\ 0 & \mathrm{L}/I_y & 0 & (I_\mathrm{r}/I_y)x2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1/I_z & 0 \end{bmatrix}$$

Where: $c1=\left(\frac{I_y-I_x}{I_x}\right)x_4$, $c2=\left(\frac{I_z-I_x}{I_y}\right)x_2$ and $c3=\left(\frac{I_x-I_y}{I_z}\right)x_4$. The different parameters of the UAV were presented by (Bresciani, 2008).

Then, the controllability condition defined by (6) should be checked:

$$rank\left(C_{trb}\left(x\right)\right)=6 \ \forall x \in \Re^{6\times 1}$$

Further on, the above-mentioned model can take the following state-space form:

$$\dot{x}=A_n\left(x\right)x+B_n\left(x\right)u \qquad (12)$$

where the subscript n denotes the nominal system (before fault occurrence). Finally, in order to solve the SDRE in (7) and to obtain the nominal control u_n (8), the penalization matrices should be defined. The matrix $Q_p\left(x\right)$ is used for weighting the system states, while $R_p\left(x\right)$ is used for weighting the control inputs and they are commonly chosen as identity matrices. For this system, one chose $Q_p=I_6$ and $R_p=I_4$ (where I denotes the identity matrix). For simulations, a small step size $t_s=10^{-4}$ was chosen, which makes results more accurate. Starting with the following ICs:

$$IC=\left[0.12; \quad 0; \quad 0.08; \quad 0; \quad -0.5; \quad 0\right] \qquad (13)$$

The objective is to stabilize the attitude of the quadrotor by driving the Euler angles and the angular velocities to the equilibrium position $(x_{eq}=0)$ with respect to a speed constraint (the response time being ideally: $\left(4s \le t \le 7s\right)$. Here, one can consider that the outputs of the system are the variable states.

Figure 1 and Figure 2 show the responses of the Euler angles and the angular velocities in normal conditions.

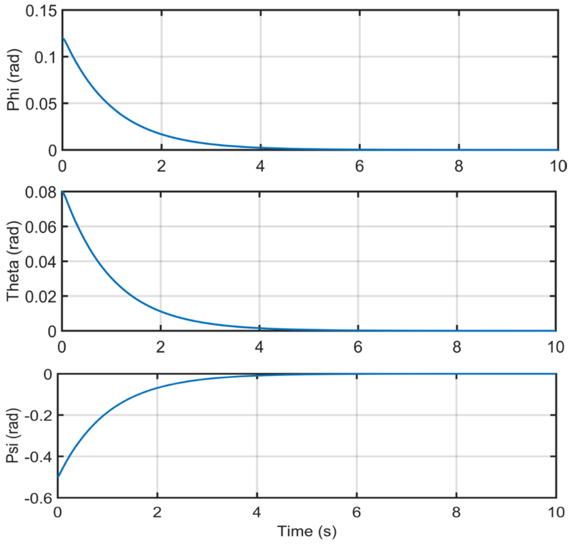

Figure 1. Stabilization of Euler angles before failure occurrence

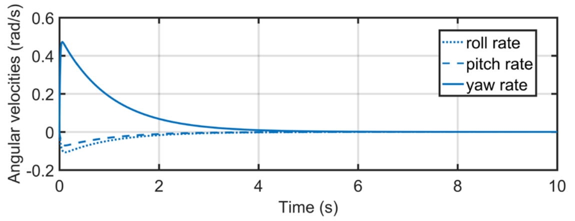

Figure 2. Stabilization of angular velocities before failure occurrence

It can be noticed that the quadrotor attitude is successfully stabilized during hovering. It's also vivid that the system needed no more than $5s$ to converge which makes the overall performance of the controller satisfactory.

4.2 Fault Injection

For quadrotors, it is more interesting to consider actuator faults rather than sensor faults due to their direct impact on system stability and the high probability of their occurrence. Moreover, a sensor fault does not require control modification, it requires only that redundant sensors or observers provide the correct estimation for the recovery process.

This study considers partial actuator failures occurring at time $t_f = 4s$, and affecting the 4 rotors of the UAV. Depending on the type of failure, the FDI module provides a pair of matrices $\left(A_f(x), B_f(x)\right)$ describing the affected system where stands for faulty. In this case, $\left(A_f(x) = A_n(x)\right)$.

The actuator failure can be modelled either by means of a multiplicative coefficient that indicates the control effectiveness, or by means of an additive term representing a deviation of the system parameters. For the system presented in this paper, the faulty control matrix becomes:

$$B_f(x) = B_n(x) + \delta B \tag{14}$$

Where: $\delta B = 10^3 \times I(6,4)$.

Figure 3 and Figure 4 illustrate the impact of the aforementioned failures. The affected system is simulated by using the same nominal control u_n without any accommodation procedure. The control sequence for the faulty system is described in Figure 5.

It can be seen that the outputs deviate quickly from their steady-state value after the failure occurrence. One can also notice that the yaw angle and yaw rate are not affected by failures because of the location of additive terms in δB.

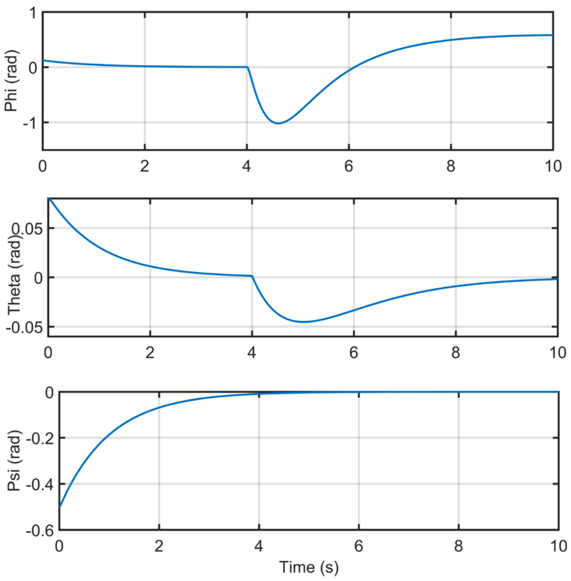

Figure 3. Euler angles after failure occurrence

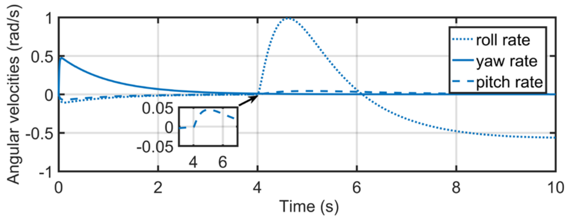

Figure 4. Angular velocities after failure occurrence

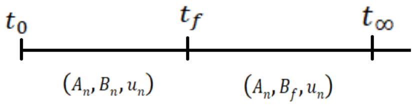

Figure 5. Control sequence for the system before recovery

4.3 Recovery Procedure

In section 4.2, the nominal control u_n was not sufficient to preserve the system stability in the presence of failures. Hence, the necessity to use the FTC law for the recovery process.

Although various studies have proven the advantages of the iterative method over the direct one, a real NL case was never approached. To this aim, both the NL PA and the direct SDRE method were employed in order to accommodate the faulty system.

The direct method of SDRE consists in applying the same steps as in section 2.2 and in taking into consideration the faulty input matrix in (14). A switching procedure is used to monitor the new computed control.

As regards the iterative strategy, the control sequence for the accommodated system is illustrated in Figure 6. For illustration purposes, it is assumed that the delay related to fault diagnosis be $t_{fdi} = 0{,}5s$ and $t_e = 0.5s$ be the time needed to compute each control law u_i .

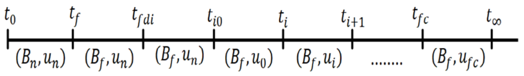

Figure 6. Control sequence for the system under the PA

As it was explained in Section 3, one starts applying $G_0(x)$ at time t_{i0}, then the Lyapunov equation (9) is solved and the obtained control u_i is applied online until the algorithm converges at time t_{fc}. When the gain matrix $G_i(x)$ remains constant (i.e., when $G_i(x) = G_{i+1}(x)$), $G_{fc}(x)$ (15) will correspond then to the final solution to which the algorithm converges. The matrix $G_{fc}(x)$ (of dimensions (4×6)), is computed using equation (10) and it ensures the stability of the closed-loop system $\left(A(x) - B_f(x) G_{fc}(x)\right)$.

$$G_{fc}(x) =$$

$$\begin{bmatrix} 0.9999 & 0.0158 & 0 & -0.0001 & 0.0001 & 0.0006 \\ -0.0148 & 0.9997 & 0.0058 & 0.0153 & -0.0142 & -0.0753 \\ 0 & 0 & 0.9258 & 0.0004 & 0.3779 & 1.0028 \\ 0.0003 & -0.0153 & 0.3779 & 1.0003 & -0.9257 & -4.9067 \end{bmatrix} \quad (15)$$

Finally, the FTC u_{fc} (16) for both methods is obtained as:

$$u_{fc} = -G_{fc} x \quad (16)$$

Figure 7, Figure 8 and Figure 9 illustrate the stabilization of the quadrotor attitude.

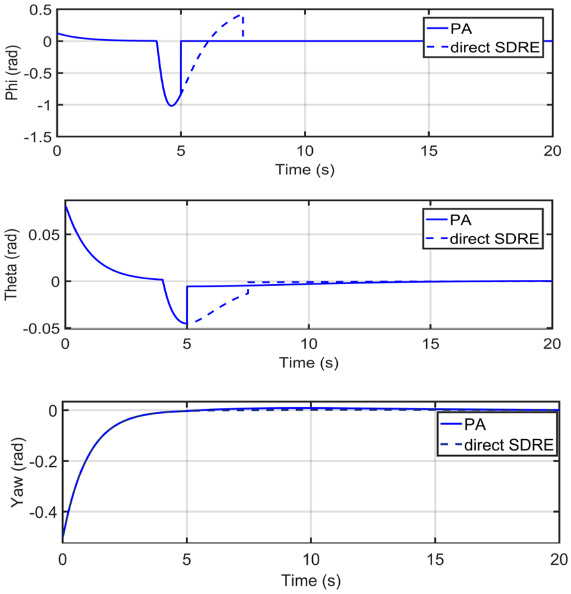

Figure 7. Euler angles after recovery: PA vs. the direct SDRE method

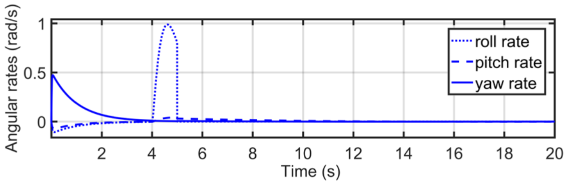

Figure 8. Angular velocities after recovery, using PA

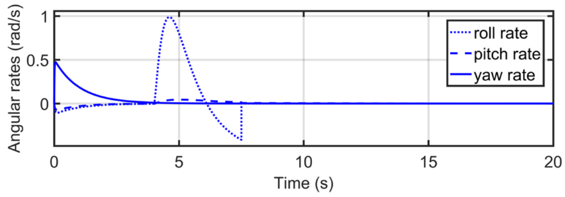

Figure 9. Angular velocities after recovery, using the direct SDRE method

Figure 10 illustrates the simulation results for the control inputs, which were obtained by using the PA strategy. One can notice the peaks that appear at time $t = 5s$ which correspond to the first application of the corrective control. After that, all control inputs converge to 0 and the speeds of the rotors are equalized (2).

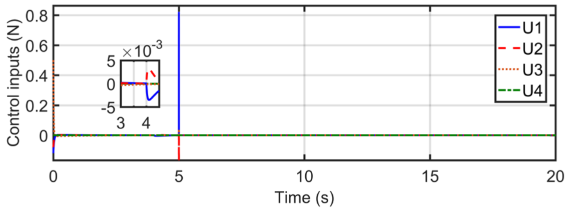

Figure 10. Control inputs using the PA

It should be reminded that the main objective of FTC is to avoid critical situations and to ensure that the faulty system can still achieve the control objectives of the healthy one.

Figure 7, Figure 8 and Figure 9 show that, for both aforementioned methods, the recovery process has led to a successful accommodation of the faulty system and it managed to bring the Euler angles back to the stable position.

However, it can be seen that the NL PA was advantageous in terms of convergence speed. When using the NL PA it took $5s$ to stabilize the system, as compared to $7.5s$ when employing the direct method.

Another important advantage of the PA is the minimization of the recovery delay. The lower this delay is, the smaller the deviation from the stable position.

Indeed, for the NPA case and in relation to Figure 6, the first corrective iteration (which was sufficient to make the system converge) was applied at $t_{i0} = t_f + t_{fdi} + t_e = 5s$. Thus, the faulty system was driven by nominal control u_n for only $\Delta = t_{i0} - t_f = 1s$. The iterative algorithm converged after 5 iterations i.e., the final FTC u_{fc} (16) was applied at $t_{fc} = t_{i0} + 5t_e = 7.5s$. However, in the case of the direct approach, the first corrective control u_{fc} was applied at $t_{fc} = 7.5s$ and the faulty system was driven by the nominal control u_n for the entire period of $\Delta = t_{fc} - t_f = 3.5s$.

As regards the practical aspect, several studies have proven the feasibility of a real-time implementation of the SDRE method.

(Bogdanov et al, 2003) presented results from initial flight tests of the SDRE control on an instrumented X-Cell-60 acrobatic helicopter. The real-time SDRE controller has successfully stabilized the autonomous vehicle. Even in the presence of wind (estimated at 3-6 km/hour), the SDRE controller has maintained a continuous control of the helicopter.

For the system presented in this paper, the implementation of the SDRE controller (using the direct method) on a small onboard computer is not easily feasible. Indeed, solving the equation (7) in real-time with a small sampling step (t_s) and for a complex system, is computationally demanding. However, the combination of the online iterative technique of PA (using the “Kleinman” algorithm) and a modern computing hardware can achieve a higher level of computational performance and enable a real-time implementation of the solution.

For experimental setup, a basic platform structure can be described as follows. The quadrotor is equipped with an IMU sensor that measures the orientation of the vehicle and provides data as a basis for feedback control. The computer running the SDRE controller receives the feedback data and generates the control inputs which are sent to the quadrotor through a radio link or a wireless modem. For safety measures, the quadrotor can also be controlled manually using the remote control system.

5. Conclusion

This paper presented a NL FTC for a quadrotor UAV. The aim of this study was to stabilize the vehicle attitude when partial failures affected its four rotors simultaneously.

Simulations were performed using the NL model of the quadrotor in fault-free and post-fault cases. By applying the iterative PA strategy and the direct SDRE method, the system has resumed its normal behaviour and the quadrotor attitude was successfully stabilized. The simulations allowed to compare the performance of the PA with that of the classical direct method. The efficiency of the PA was proven in terms of stability and convergence time namely by minimizing the recovery delay (needed for the computation of the FTC).

Although solving the SDRE in a recursive way has many advantages, some limitations can be encountered in this context: The only criterion of choosing the SDC matrices $\left(A(x), B(x)\right)$ is to guarantee the controllability of the system. This implies that each choice of the aforementioned matrices yields a different solution $T(x)$ of the SDRE. However, theoretically, there exists only one $T(x)$ that is considered an “optimal” solution. Since it is difficult to choose the SDC matrices that yield optimal control, the produced solution is only “suboptimal” which reveals to be a potential drawback of the SDRE method. Another limitation of this approach is its approximation nature i.e., it requires a very small sampling time in order to increase the accuracy of the control law which makes simulation time last longer. Finally, as regards the iterative technique, the main difficulty encountered is to find the initial stabilizing gain which is a hard-task when all matrices are state-dependent.

For future works, it could be interesting to establish a full scheme related to fault diagnosis and FTC such as the one described by (Saied et al., 2015) which would allow one to investigate the SMC for the FDI procedure. This work could also be extended with the purpose of attaining a full control of the UAV which would include controlling its altitude and position.

REFERENCES

Andrade, J. Campos, V., Potts, A. & Garcia, C. (2017). Damping Improvement of a F-16 Aircraft through Linear Matrix Inequalities. In *Proceedings of the 20th IFAC World Congress*, *50*(1), (pp. 3947-3952).

Bogdanov, A. Carlsson, M., Harvey, G., Hunt, J., Kieburtz, D., Merwe, R. & Wan, E. (2003). State-Dependent Riccati Equation Control Of A Small Unmanned Helicopter. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*. DOI: 10.2514/6.2003-5672

Bresciani, T. (2008). *Modelling, Identification and Control of a Quadrotor Helicopter.* MSc Theses, Lund University.

Ciubotaru, B. & Staroswiecki, M. (2006). Fault tolerant control of the B747 short-period mode using progressive accommodation. In *Proceedings of the 15th IEEE International Conference on Control Applications* (pp. 3288-3294).

Ciubotaru, B. & Staroswiecki, M. (2009). Anytime Algorithm for Parametric Faults Accommodation under Handling Quality Constraints. In *Proceedings of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, 42*(8), (pp. 887-892).

Emran, B. & Najjaran, H. (2018). A review of quadrotor: An underactuated mechanical system, *Annual Reviews in Control*, *46*, 165-180.

Ghachem, S., Benothman, K. & Benrejeb, M. (2012). Actuator fault accommodation with optimal control for a class of nonlinear systems, *International Review of Automatic Control*, *5*(2), 274-283.

Kleinman, D. (1968). On an iterative technique for Riccati equation computation, *IEEE Transactions on Automatic Control*, *13*(1), 114-115.

Korayem, M. H. & Nekoo, S. R. (2015). Finite-time state-dependent Riccati equation for time-varying nonaffine Systems: Rigid and flexible joint manipulator control, *ISA Transactions*, *54*, 125-144.

Menon, P., Lam, T., Crawford, L. & Cheng, V. (2002). Real-Time Computational Methods for SDRE Nonlinear Control of Missiles. In *Proceedings of the American Control Conference* (pp. 232-237).

Merheb, A., Noura, H. & Bateman, F. (2014). Active Fault Tolerant Control of Quadrotor UAV Using Sliding Mode Control. In *International Conference on Unmanned Aircraft Systems (ICUAS)*, (pp. 156-166).

Mu, C., Wang, D. & He, H. (2018). Data-Driven Finite-Horizon Approximate Optimal Control for Discrete-Time Nonlinear Systems Using Iterative HDP Approach, *IEEE Transactions on Cybernetics*, *48*(10), 2948-2961.

Pandey, V. K., Kar, I. & Mahanta, C. (2017). Controller design for a class of nonlinear MIMO coupled system using multiple models and second level adaptation, *ISA Transactions*, *69*, 256-272.

Raharijaona, T., Outbib, R., Ouladsine, M. & Ghachem, S. (2009). Comments on actuator fault accommodation. In *Proceedings of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes* (pp. 1611-1616).

Saied, M., Lussier, B., Fantoni, I., Francis, C., Shraim, H. & Sanahuja, G. (2015). Fault diagnosis and fault tolerant control strategy for rotor failure in an octorotor. In *IEEE International Conference on Robotics and Automation (ICRA)*, (pp. 5266-5271).

Staroswiecki, M., Yang, H. & Jiang, B. (2007). Progressive accommodation of parametric faults in linear quadratic control, *Automatica, 43*(12), 2070-2076.

Tchon, K. & Janiak, M. (2009). Repeatable approximation of the Jacobian pseudo-inverse, *Systems & Control Letters, 58*(12), 849-856.

Yen, G. G. & Ho, L. W. (2004). Intelligent on-line fault tolerant control for unanticipated catastrophic failures, *ISA Transactions*, *43*(4), 549-569.

Yu, Z., Zhang, Y., Liu, Z., Qu, Y. & Jiang, B. (2019). Decentralized finite-time adaptive fault-tolerant synchronization tracking control for multiple UAVs with prescribed performance, *Journal of the Franklin Institute*. DOI: 10.1016/j.jfranklin.2019.11.056

Zhang, Q. Wang, C., Su, X. & Xu, D. (2018). Observer-based terminal sliding mode control of non-affine nonlinear systems: Finite-time approach, *Journal of the Franklin Institute*, *355*(16), 7985-8004.

Zhi, J., Chen, Y., Dong, X., Liu, Z. & Shi, C. (2018). Robust adaptive FTC allocation for over-actuated systems with uncertainties and unknown actuator non-linearity, *IET Control Theory and Applications*, *12(*2), 273-281.