Quadcopter Control Algorithms in the event of Loss of One of the Actuators: A Review

Jackson O. Oloo and Stanley I. Kamau

***Abstract*—**There are increasing needs to develop reliable control systems that can be used to automatically stabilize a Quadcopter Unmanned Aerial Vehicle (UAV) when one of the four rotors is under fault. A fault tolerant mechanism that extends the capabilities of the quadcopter system to operate under the presence of faults is of interest to the research community. The quadcopter UAV is a great platform for control systems research as its nonlinear nature and under-actuated configuration makes it ideal to analyze control algorithms. The nonlinearity is even further pronounced when there is an actuator fault. This review gives an overview of Quadcopter UAV dynamic system. Considering a single actuator fault scenario, applicable algorithms are analyzed highlighting their advantages and disadvantages. The algorithms include PID control, Gain-Scheduling PID, Linear Quadratic control, Backstepping, Feedback linearization, Sliding Mode control, Model Predictive control and Model Reference Adaptive Controller. The conclusion of this work is a proposal of combination of positive attributes of each algorithm to compensate for the limitations of the other.

***Keywords*—** Control Algorithms, Nonlinear Control, Quadcopter

I. INTRODUCTION

A quadcopter is an aircraft that is lifted and propelled by four rotors in a cross configuration and its basic motions are generated by varying the speeds of all the four rotors. It is a 6 Degree of Freedom (DOF) device with only four actuators, which makes it an under actuated vehicle with unstable dynamics. For small aerial vehicles, due to hardware redundancy limitations, design of a reliable control system plays an important role in ensuring acceptable and efficient performance.

In recent years, there has been a surge of interest in the use of small Unmanned Aerial Vehicles (UAVs) for various civilian and military applications [1]. These applications include package delivery, aerial imagery, surveillance, and structural inspection; a common aspect is that these tasks are either in remotely inaccessible locations and require dangerous maneuverability or are in unfriendly environments in case of military operations. Several different UAV platforms exist that have the potential to solve these problems such as fixed-wing airplanes, lighter-than-air blimps, and multirotor aircrafts. A quadcopter has advantages over the fixed wing UAVs in that it has Vertical Take-off and Landing (VTOL) capabilities and can perform maneuvers. Its advantage over other rotary UAVs, such as a helicopter, is that it is mechanically simple; a quadcopter does not need a complex set of mechanical linkages to alter rotor blade angles. Quadcopter helicopters do not require a tail rotor and this allows it to devote all vehicle power o producing lift. This allows for significant payload capacity in relation to vehicle weight.

However, a quadcopter is a six degrees of freedom system with only four actuators, making it underactuated as well as being a highly nonlinear and unstable system [2]. With such a configuration, the entire vehicle must tip in one direction or another in order to direct the rotor thrusts to actuate lateral or longitudinal motion. This could be seen as a potential disadvantage as it does constrain the dynamics of the vehicle in that it cannot cause acceleration forward or back or from side to side while maintaining a given orientation.

It is therefore a bigger challenge to maintain full control of all the attitude states and all the translational states when one of the rotors has failed and the system becomes even further under actuated. This makes the quadcopter highly non-linear and several uncertainties are encountered during its missions. This has led to several control algorithms proposed in the literature. In this work, a review of the prominent controllers applied to the quadrotor is reviewed.

II. QUADCOPTER MODEL

A. MOTION OVERVIEW

Quadcopters consist of four rotors attached to a rigid cross airframe as shown in Fig.1, with two opposing rotors rotating clockwise (1,3) and the other two rotating counterclockwise (2,4). Control of quadcopter is achieved by differential control of the thrust generated by each rotor.

Attitude motion is accomplished by simultaneously increasing or decreasing the speed of all four rotors. Pitch angle, θ, is controlled by speeding up motor 3 while slowing down motor 1 or vice versa while roll angle, φ, is controlled by slowing down motor 4 while speeding up motor 2 or vice versa. Then yaw angle, ψ, is controlled by speeding up motors 1 and 3 while slowing down motors 2 and 4 or vice versa.

J. O. Oloo, Department of Electrical and Electronic Engineering, JKUAT (phone: +254727422602; e-mail: joloo@jkuat.ac.ke).

S. I. Kamau, Department of Electrical and Electronic Engineering, JKUAT (e-mail: skamau@eng. jkuat.ac.ke).

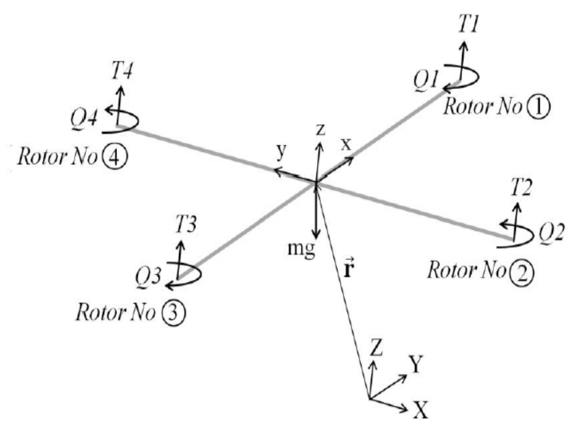


Fig 1 Quadrotor free body diagram.

B. EQUATIONS OF MOTION

The equations of motion that describe the dynamics of the vehicle are developed using a Newton-Euler formalism. The system in state space form is given as:

$$\dot{X} = f(x, U) \tag{1}$$

Where U is the input vector and x the state vector.

The derivations of the nonlinear dynamics of the system is summarised by the following equations [3] [4].

$$\dot{X} = \begin{bmatrix} \dot{\phi} \\ \ddot{\phi} \\ \dot{\theta} \\ \ddot{\theta} \\ \dot{\varphi} \\ \ddot{\varphi} \\ \dot{z} \\ \ddot{z} \\ \dot{x} \\ \ddot{x} \\ \dot{y} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta}\dot{\varphi}\frac{I_{YY}-I_{ZZ}}{I_{XX}} + \dot{\theta}\,\frac{J_r}{I_{XX}}\Omega_r + \frac{l_a}{I_{XX}}U_2 \\ \dot{\theta} \\ \dot{\theta}\dot{\varphi}\frac{I_{ZZ}-I_{XX}}{I_{YY}} - \dot{\phi}\,\frac{J_r}{I_{YY}}\Omega_r + \frac{l_a}{I_{YY}}U_3 \\ \dot{\varphi} \\ \dot{\theta}\dot{\phi}\frac{I_{XX}-I_{YY}}{I_{ZZ}} + \frac{1}{I_{ZZ}}U_4 \\ \dot{z} \\ -g + (\cos\phi\cos\theta)\frac{U_1}{m_s} \\ \dot{x} \\ u_x\frac{U_1}{m_s} \\ \dot{y} \\ u_y\frac{U_1}{m_s} \end{bmatrix} \tag{2}$$

Where,

I_{XX}, I_{YY}, I_{ZZ} are the Moment of Inertia of the Quadcopter.

l_a Quadcopter arm length

m_s total mass of the Quadcopter

g acceleration of gravity

ϕ, θ and φ are the roll, pitch and yaw angles respectively

J_r moment of inertia of the rotor about its axis of rotation and

$$\begin{bmatrix} u_x \\ u_y \end{bmatrix} = \begin{bmatrix} \cos\phi\sin\theta\cos\varphi + \sin\phi\sin\varphi \\ \cos\phi\sin\theta\sin\varphi - \sin\phi\cos\varphi \end{bmatrix}$$

U is the input vector consisting of U_1 (total thrust), and U_2, U_3, U_4 which are related to the rotation of the Quadcopter. The inputs are mapped by:

$$U = \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ b(-\Omega_2^2 + \Omega_4^2) \\ b(\Omega_1^2 - \Omega_3^2) \\ d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \end{bmatrix} \tag{3}$$

b, d thrust, drag coefficient and Ω is the angular velocity of the rotor.

A fault is generated in an actuator by multiplying its control input by a gain smaller than one, thus simulating a loss in the control effectiveness.

[4] provides a detailed quadrotor model that is derived from the Newton-Euler formulation.

III. QUADROTOR CONTROL ALGORITHMS

This section discusses both linear and nonlinear approaches towards the control of a quadcopter under a faulty rotor and compares their performance in relation to their advantages and disadvantages.

A. Proportional Integral and Derivative (PID) Technique

PID controllers are control loop feedback mechanisms that directly adjust control values with a closed-form formula based on derivative, integral, and proportional gains [5].

PID formulation is employed in [5] as shown below:

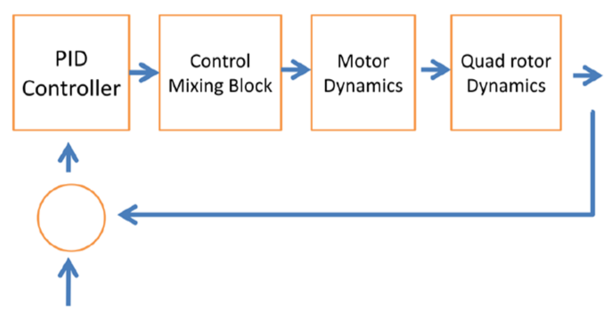


Fig 2 PID Control Strategy for a Quadrotor

Beginning with an open loop system to show the advantages of PID control, the results are as shown in Fig. 3 and 4.

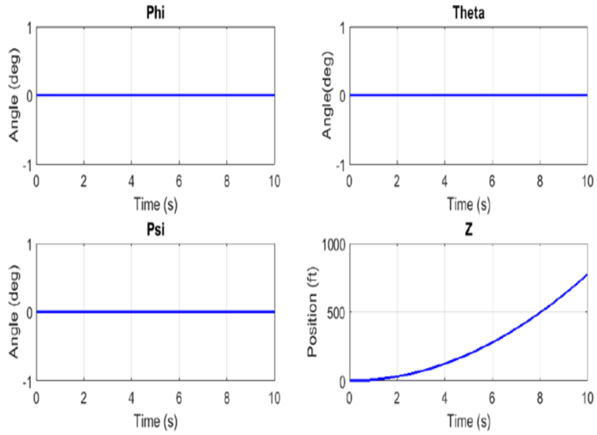


Fig. 3 Open loop step responses for motor speeds $w_1 = w_2 = w_3 = w_4 = 5000\ RPM$

Fig. 4 shows output of quadrotor response when $w_1 = w_2 = w_3 = 5000\ RPM\ \ and\ w_4 = 4950\ RPM$

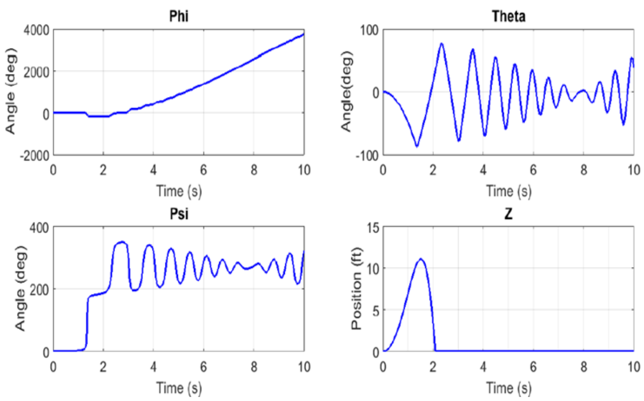


Fig. 4 Open loop step responses for motor speeds $w_1 = w_2 = w_3 = 5000\ RPM\ \ and\ w_4 = 4950\ RPM$

In [5], the scheme then describes the implementation of a PID controller without rotor failure. The gain values are set as $K_p = 10, K_i = 0$ and $K_d = 11$. The closed loop response is as shown in Fig. 6 with overshoot of 0% and settling time of 3.6043 seconds.

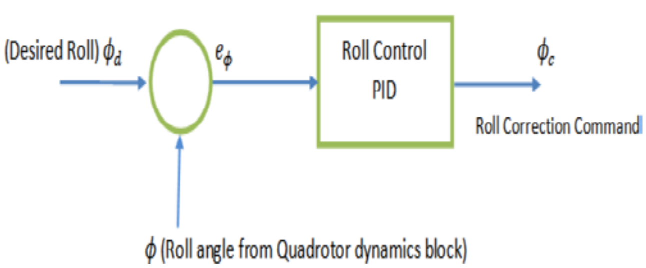


Fig. 5 Roll Control PID Block

Similar arrangement applies to PID Pitch control

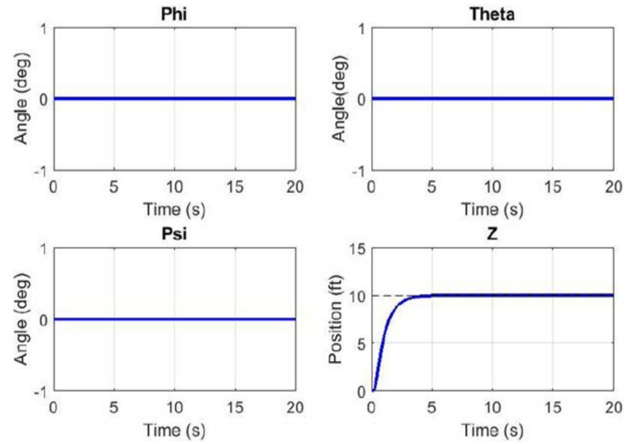


Fig. 6 Altitude Closed Loop Step Response

PID control was implemented in design of Quadrotor Controller for stabilization after failure of one of the rotors [6]. Motor 2 is switched off after 37 seconds and the response of the system is as shown in Fig. 7.

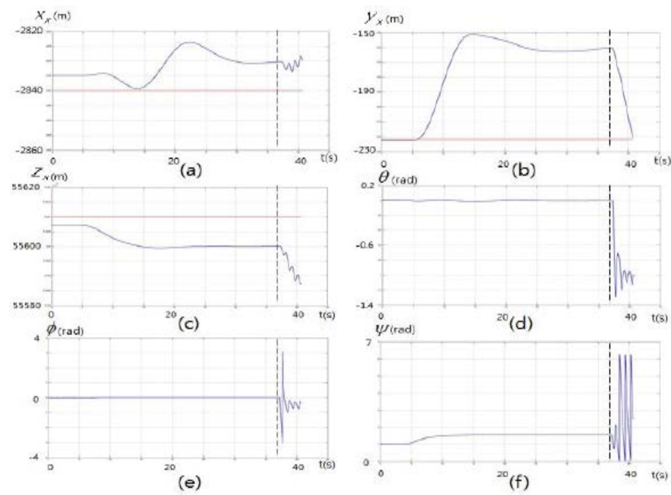


Fig. 7 Response for PID fault tolerant control

In conclusion, the classical PID controller has the advantage that parameter gains are easy to tune, simple to design and is robust. However, the quadrotor being underactuated has a nonlinear mathematical model. PID linear controller has been found to struggle with aggressive maneuvers [7] especially when one of the rotors is faulty.

B. Gain Scheduled Proportional Integral Derivative controller (GS-PID)

PID controllers are designed and tuned in both fault-free and faulty situations to control the quadcopter under normal and faulty flight conditions. In GS-PID controller, several sets of pre-tuned gains are applied to the controllers in different flight conditions under both fault-free and faulty cases. In the next step, attempts to obtain the best stability and performance of Quadcopter in attitude and altitude tracking control under both cases and to switch the controller gains from one set of pre-tuned PID controller to another set of the gains in the presence of different levels of actuator faults are carried out. One of the main parameters to be considered in GS-PID is the switching time between the time of fault occurrence and the time of switching to new set of gains. If this transient (switching) time is held long (more than one second) it can cause the Quadcopter to hit the ground and cause a crash.

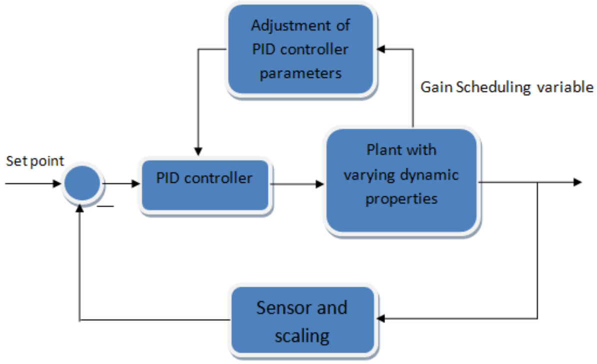


Fig. 8 GS-PID controller structures

GS-PID is applied in [8] for fault tolerant control of a quadcopter for an 18% of overall loss in power of all motors. Acceptable tracking deviation from the desired square trajectory after the fault occurrence was obtained with the fault injected at 20s.

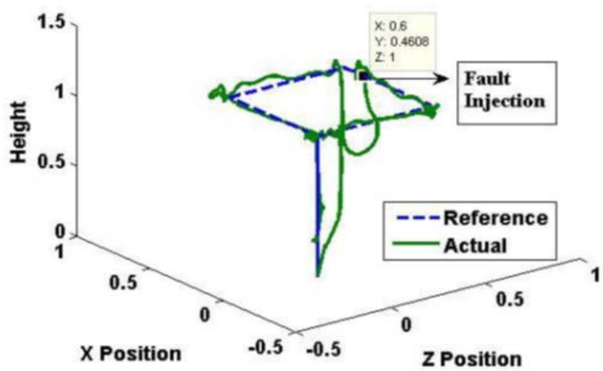


Fig. 9 GS-PID without time delay for controller switching in faulty condition

As much as GS-PID is easier to design and implement in MATLAB/ Simulink, it was observed that a good tuning for the GS-PID controller gains was very time consuming and gains could change from one flight depending on the flight environment. Any change in lab environment during flight forced the need for retuning of the gains.

C. *Model reference adaptive controller (MRAC)*

Model Reference Adaptive Control (MRAC) is concerned with forcing the dynamic response of the controlled system to asymptotically approach that of a reference system, despite parametric uncertainties (faults) in the system [9]. Two major subcategories of MRAC are those of indirect methods, in which the uncertain plant parameters are estimated and the controller redesigned online based on the estimated parameters, and direct methods, in which the tracking error is forced to zero without regard to parameter estimation accuracy.

Direct method poses an advantage over indirect method since there is no need of estimation of unknown parameters for implementing the adaptive controller. Direct method is selected in [9] for fault-tolerant control of a quadcopter.

In [9] the flight was tested for both hovering control and square trajectory tracking controls with fault injection. The experimental flight testing results are shown in Fig. 10 and 11. In Fig. 10, a fault-free condition is applied to the Quadcopter and the MRAC was able to track the trajectory close to real one. In Figure 10, a fault is injected to the left and back motors at 20 sec with a loss of 18% of power during the flight. From Fig. 11, the Quadcopter could still track the desired trajectory with a safe landing.

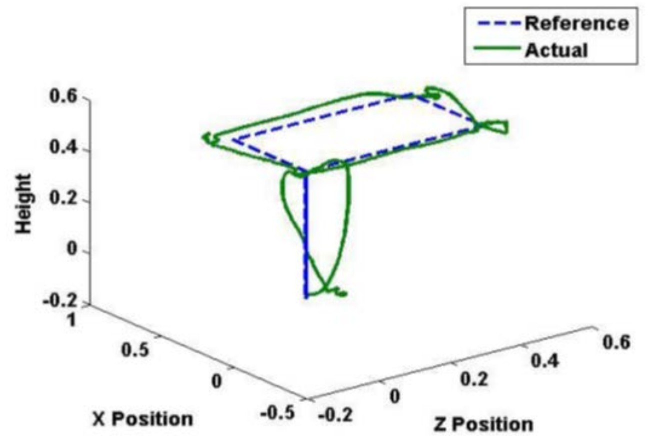


Fig. 10 Square trajectory in fault-free condition with MRAC

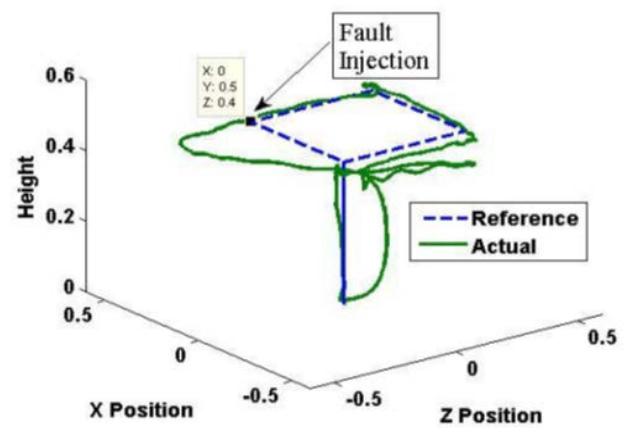


Fig. 11 Square trajectory in faulty condition (left and back motors) with MRAC

The MRAC used in [9] proved to be more reliable and robust to the lab noises and environment changes. However, involving mathematical model derivations are needed to design and implement the controller.

D. *LQR controller*

This method derives the feedback gain for a system. Applying the Linear Quadratic (LQ) control requires the system described in (1) to be linearized to $\dot{X} = AX + Bu$ i.e.

$$\dot{X}^T = [\dot{\phi}\ \ddot{\phi}\ \ \dot{\theta}\ \ddot{\theta}\ \dot{\varphi}\ \ddot{\varphi}]^T \tag{4}$$

A state feedback control $u = -Kx$ is designed to stabilize the system as:

$$\dot{X} = (A - BK)x \tag{5}$$

Where

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{I_{YY} - I_{ZZ}}{2I_{XX}}\dot{\varphi} & 0 & \frac{I_{YY} - I_{ZZ}}{2I_{XX}}\dot{\theta} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & \frac{I_{ZZ} - I_{XX}}{2I_{YY}}\dot{\varphi} & 0 & 0 & 0 & \frac{I_{YY} - I_{XX}}{2I_{YY}}\dot{\phi} \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & \frac{I_{XX} - I_{YY}}{2I_{ZZ}}\dot{\theta} & 0 & \frac{I_{XX} - I_{YY}}{2I_{ZZ}}\dot{\phi} & 0 & 0 \end{bmatrix}$$

And

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{l}{I_{XX}} & 0 & 0 & \frac{Jr\dot{\theta}}{I_{XX}} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{l}{I_{YY}} & 0 & 0 & \frac{Jr\dot{\theta}}{I_{YY}} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{I_{ZZ}} & 0 \end{bmatrix}$$

The state variable feedback gain, K is calculated as

$$K = R^{-1}B^TP \tag{6}$$

Where $R\ and\ P$ are weighted matrix and constant matrix respectively.

In [10], disturbance by means of a step signal is injected into the system and the model responds as shown in Fig. 12.

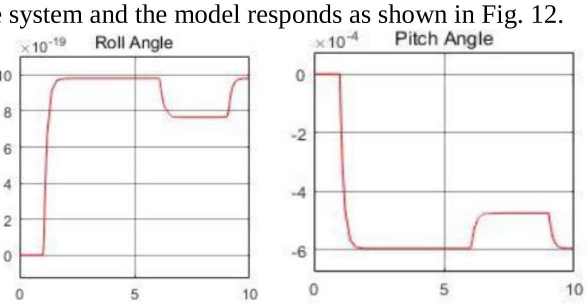


Fig. 12 Nonlinear model response for step signal disturbance

From the results, the controller tries to bring the model back to zero as shown in Fig. 12 but at the end the model does not go to zero because the given disturbance is a huge one compared to the values obtained from the model.

Castillo et al. [11] have implemented this kind of controller. During simulation, the controller has performed satisfactory. When strong perturbation was introduced, the controller due to its linearity was not able to stabilize the system. On the physical model, this controller was not able to stabilize the system at all.

However, simulation results have shown that an LQR architecture in fault free applications has superior performance compared to a classical PD architecture in terms of transient response [12]. In [13] LQR algorithm is applied to a quadrotor and its performance compared to that of the PID controller. However, the PID is applied on the quadrotor simplified dynamics and the LQR on the complete model. Both approaches provided average results but it implicitly was clear that the LQR approach had better performance considering the fact that it was applied to a more complete dynamic model.

E. Feedback Linearization (FBL)

FBL transforms a nonlinear system into an equivalent linear system through a suitable control law. A control law is designed such that the nonlinear term is cancelled to result in a controllable linear system. Some of the disadvantages of using FBL is that it needs a Fault Diagnosis and Identification system (FDIs), the loss of precision due to linearization and requiring an exact model for implementation [14].

In [15], a Fault Tolerant Controller (FTC) was developed having a double control loop architecture with an inner and outer controller. The inner and faster controller has the task to regulate the attitude angles and the altitude of the vehicle, while an outer and slower controller has the task to supply a proper input to the fault-free couple of rotors in order to make the quadrotor reach a desired position in space.

The inner control law ensures that when one of the rotor fails, the velocity of the rotor laying on the same axis of the faulty rotor is modulated until the value of the angle controlled by the faulty couple of rotors is zero. In this configuration, the quadrotor is parallel to the ground without the outer controller, spinning around the vertical axis and varying simultaneously the rotational velocity of the two rotors of the fault-free couple. This makes it possible to set a desired altitude for the vehicle. The outer control law ensures the quadcopter reaches a desired position in space by supplying proper input to the fault free rotors.

The inner control loop controls roll, pitch and altitude, while the outer control loop sets the desired values of the $\emptyset$ and θ angles in order to control position in the $xy - plane$. Since roll, pitch and altitude have the highest priority during flight, the inner controller works much faster than the outer controller.

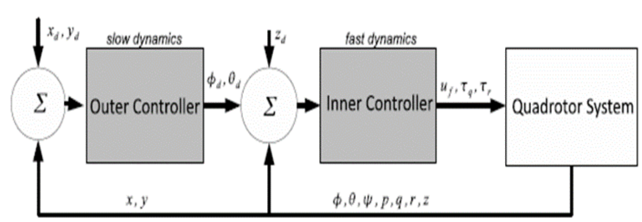


Fig. 13 Block diagram for FBL control strategy

FTC using FBL in case of failure on the actuator 2 is described in [15]. In the first simulation, only the inner controller is incorporated while the outer control is deactivated. The desired values for the roll and pitch are all set to 0 except for the altitude which is chosen to be$10m$. The simulation was run for $20s$ which is a sufficient time to reach hover.

As it can be seen from Fig. 14 the attitude angles $\emptyset$ and θ go to zero in less than $15s$ and with a smooth profile while the altitude z is quickly regulated to the desired value of $10m$.

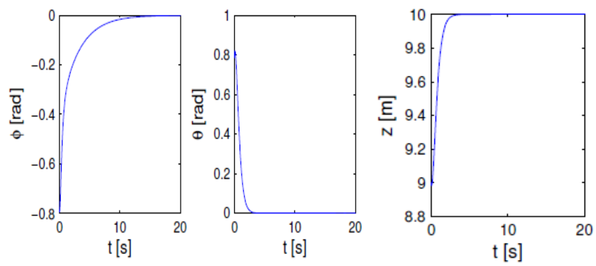


Fig. 14 System response with only inner controller applied

In the second simulation, both the inner controller and the outer controller are incorporated. The desired values for the roll and pitch are all set to 0 except for the altitude which is chosen to be10m. If the quadrotor does not reach the desired lateral and longitudinal position, then it is set to a ramp with negative slope until it becomes zero (landing procedure).

Figure 15 shows that the attitude angles $\emptyset$ and θ are stabilized, but this time there are oscillations due to the presence of the outer controller.

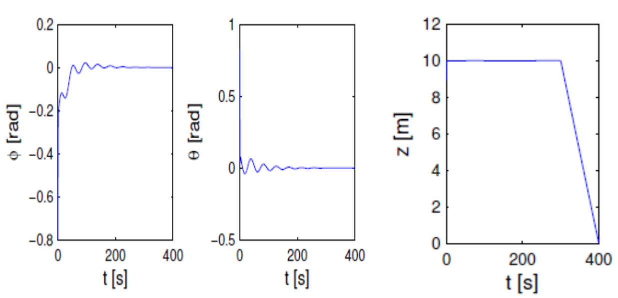


Fig. 15 Orientation angle stabilization with both inner and outer controller applied.

F. *Sliding Mode Controller (SMC)*

In SMC, the common task is to design a state feedback control law that maps the current state $X(t)$to the input u to stabilize the system around the origin $X = [0\ \ 0,..0]^T$. Whenever the system is started away from origin, it will return to it. SMC forces the system trajectories into a constrained subspace then holds them there so that they slide along with it.

In order to have a system track $\emptyset(t) \equiv \emptyset d(t)$, a sliding surface is defined as follows [16]:

$$s_\phi = \dot{z}_\phi + \lambda_\phi z_\phi \tag{7}$$

Where $z_\emptyset = \emptyset_d - \emptyset$ Then we have:

$$\dot{s}_\phi = \ddot{\phi} - \ddot{\phi}_d + \lambda_\phi \dot{z}_\phi \tag{8}$$

$$= \dot{\theta}\dot{\varphi}\frac{I_Y - I_Z}{I_X} + f_1 + \frac{l}{I_X}U_1 - K_4\dot{\phi} + g_1 - \ddot{\phi}_d + \lambda_\phi \dot{z}_\phi$$

The best approximation of U_1 for a continuous control law is obtained by setting $\dot{s}_\phi = 0$. The same steps for achieving the best U_1 approximation are used to extract U_2, U_3 and U_4

Main drawback of sliding mode control is the chattering due to the switching from the normal operation mode and the faulty mode.

Farid and co-researchers [16] describes the fault tolerance property of SMC and uses it in an FTC. The objective of their work is to land the quadrotor horizontally ($\emptyset = 0$ *and* $\theta = 0$) when an actuator fault occurs. During fault free operation, signal U_2 that controls the pitch angle is equal to the difference of thrust of the motor1 and 2. When there is a fault in the motor 1, it is possible that it cannot provide sufficient thrust for control signal U_2 . To resolve this problem, the effect of motor1 in control signal U_2 is omitted and the control signal is only provided with the motor2.

A faulty condition is introduced after 7 seconds to rotor 1 and the performance of the quadcopter is observed without and with the SMC.

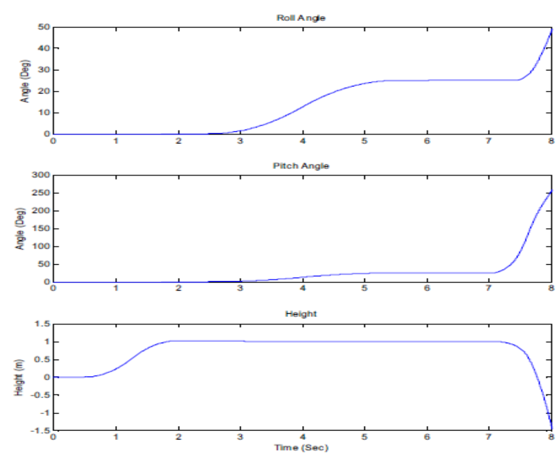


Fig. 17 Quadcopter performance in faulty situation with SMC

Li et al. [17] investigates both passive and active the fault-tolerance property of sliding mode controllers. Both simulation and experimental results using the Qball-X4 system model were performed. It was observed that the active FTC is more robust and shows better tracking performance in the presence of faults. A detailed analysis emphasizing the advantages and disadvantages of each type is discussed.

Studies presented in literature are believed to present the maximum fault value a passive and an active fault tolerant control SMC can handle. The SMC has not been tested for higher faults and severe actuator failures.

G. *Model predictive control (MPC)*

The MPC recalculates the control signal at each sampling time. Any change in the process dynamics is reflected into the control signal calculation [18]. The drawback of MPC is that it needs an almost explicit model of the system to calculate a stabilizing control signal. In addition, the abrupt changes in the model parameters, due to failure, requires a Fault Detection and Diagnosis (FDD) to provide information about the occurring faults to allow MPC to consider faults.

The system dynamics in presence of an actuator fault can be rewritten as:

$$x(k+1) = f(x(k), \alpha(k)u(k)) \tag{9}$$

Where $\propto$ is the fault parameter matrix capturing the fault severity. In derivations scalar $\propto_i$ denotes the amplitude of the fault in the i^{th} actuator. $\propto_i = 1$ denotes fault free actuator, $\propto_i = 0$ denotes a complete loss of actuator effectiveness and $0 < \propto_i < 1$ denotes a partial loss.

For a given time step k, MPC generates the input and state trajectories, by solving an optimization problem Q_k . The MPC controller applies only the first computed control input, $u_k(0)$ to the system and the time step is then incremented. The process is recursive as shown:

Given $x(0)$ and r^i , where x is the actual state vector and r^i is the reference state, then;

- $K = 0$
- Measurement (or estimate) $x(k)$
- Solve Q_k and generate u_k and x_k
- Apply u_k (0) to the system
- $K = K + 1$ and GOTO step2

An MPC strategy is proposed in [19], sacrificing the control on yaw. According to the simulations, the MPC method is able to get the quadrotor UAV in hover position. However, the controller cannot be implemented on hardware for experimental results since the angular velocities are very high and may cause problems therefore there is need to physically validate these simulation results. From Fig. 18 the yaw angle goes to infinity due to the aerodynamic drag of the vehicle, which was not taken into account in this work.

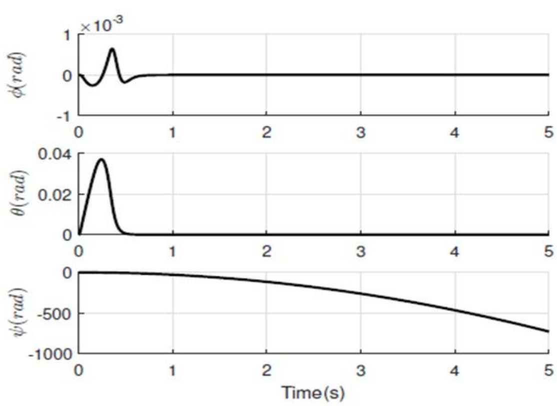


Fig. 18 Euler angles of the quadrotor UAV when rotor 1 is shut down.

The presented simulations showed that the roll and pitch angles were stabilized at the desired angles. However, MPC needs a relatively accurate model of the post-failure system to calculate a stabilizing control signal. The problem becomes more critical where the system dynamics is described by a nonlinear model [19]. This therefore calls for nonlinear parameter estimators such as Moving Horizon Estimation (MHE) and/or Unscented Kalman Filter (UKF) for reasonable online computation time as presented in [20].

H. *Backstepping control*

This is a Lyapunov stability based algorithm. The higher order nonlinear system is disintegrated systematically into a number of lower order subsystems. The lower order subsystems are arranged in cascaded form with each other to form overall closed loop system. Each lower order subsystem is stabilized by another lower order subsystem which generates a control input. This recursive loop keeps going until feedback control is achieved. This feedback control law is converted, through parameter estimation, into dynamic control law to accommodate the dynamic perturbations in parameters.

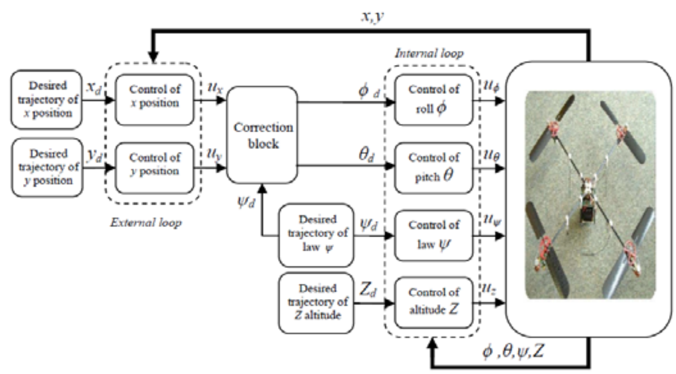


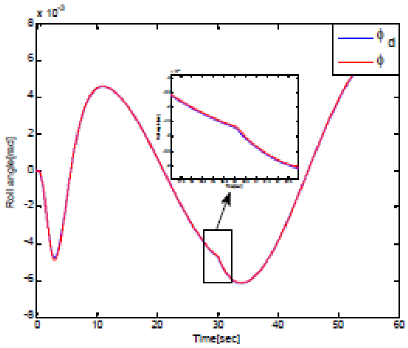
Fig. 19 Synoptic Scheme of the proposed control strategy [21]

The adopted control strategy in [21] describes a control strategy based on an internal loop and external loop. The internal loop controls the roll, pitch, yaw and altitude. The external loop includes control laws of positions x and y. The external control loop generates a desired roll (ϕ_d) and pitch (θ_d) through the correction block as follows:

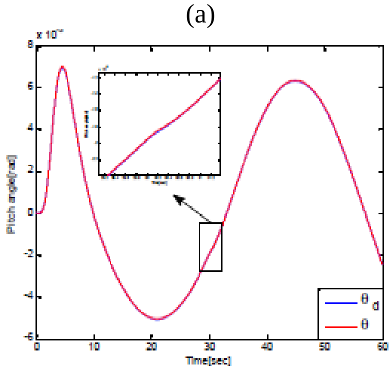
$$\phi_d = \arcsin(U_x Sin(\varphi_d) - U_y \cos(\varphi\_d)) \tag{10}$$

$$\theta_d = \arcsin(\frac{U_x \cos(\varphi_d) + U_y Sin(\varphi_d)}{\cos(\phi_d)}) \tag{11}$$

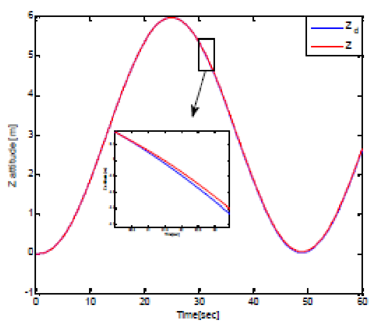
In Backstepping approach, a recursive algorithm is used to synthesize the control laws forcing the system to follow the desired trajectory in presence of actuator faults.



(a)



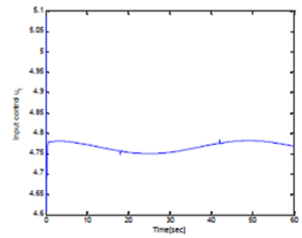
(b)



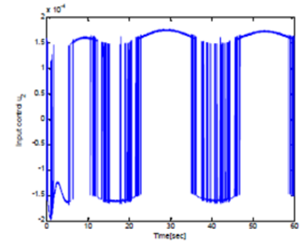
(c)

Fig. 20 (a), (b) and (c) shows tracking simulation results of trajectories along roll (ϕ), pitch (θ) and Z axis

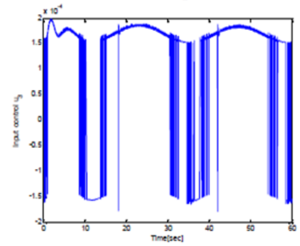
The simulation results in [21] shows robustness towards stability and tracking even after the occurrence of actuator faults explaining the efficiency of the control strategy developed. However, Backstepping approach is affected by chattering effects on the inputs as shown in Fig. 21. There is therefore need to develop other control strategies in order to eliminate the chattering phenomenon in inputs control u_1, u_2, u_3, u_4, while maintaining the stability and the performances of this system, with implementation them on a real system.



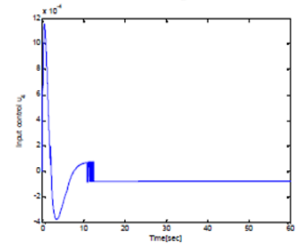
(a) evolution of input control u_1



(b) evolution of input control u_2



(c) evolution of input control u_3



(d) evolution of input control u_4

Fig. 21 chattering effects on inputs

IV. CONCLUSION

It can be concluded that the discussed algorithms have advantages over each other. Therefore, it can be pointed out that the limitations discussed can be compensated by the advantages of the other algorithm. . It is therefore necessary to also look at Knowledge based and Hybrid Algorithms in future for comparison with most of the classical algorithms discussed in this review. This review acts as a stepping-stone into further work that will be conducted by the authors on design of Quadcopter Fault tolerant controller.

V. REFERENCES

[1] A. Girard, A. Howell, and J. Hedrick, “Border patrol and Surveillance missions using multiple Unmanned Air Vehicles,” in in 43rd IEEE Conference on Decision and Control (CDC), vol. 1, pp. 620 - 625, Dec 2004.

[2] S. Lee, S. H. Kang, and Y. Kim, “Trajectory tracking control of quadrotor UAV,” in in 11th International Conference on Control, Automation and Systems (ICCAS), pp. 281 – 285, Oct. 2011.

[3] S.Bouabdallah, R.Siegwart, “Full control of a Quadrotor” ,Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on San Diego, CA, USA .

[4] S. Bouabdallah, “Design and Control of Quadrotors With Application to Autonomous Flying”, PhD thesis, Ecole Polythechnique Federale De Lausanne, 2007.

[5] M. Numan ,“Controller Design for Attitude and Position Control of Quadrotor,” Master’s thesis, Capital University of Science and Technology, Pakistan, 2017.

[6] K. Cho, J.Shin and T.Kuc , “Design of Quadrotor Controller for Emergency Situation using Xplane” The 12th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI 2015) October 28~30, 2015 / KINTEX, Goyang city, Korea

[7] C. Balas, “Modelling and Linear Control of Quadcopter,” M.S. Thesis, Cranfield University, Cranfield, 2007

[8] Y.Zhang, “Concordia University Fault-Tolerant Trajectory Tracking Control of a Quadrotor Helicopter Using Gain-Scheduled PID and Model Reference Adaptive”,Annual Conference of the Prognostics and Health Management Society, 2011

[9] I.Sadeghzadeh, A.Mehta, Y.Zhang, “Fault-Tolerant Trajectory Tracking Control of a Quadrotor Helicopter Using Gain-Scheduled PID and Model Reference Adaptive Control”, Annual Conference of the Prognostics and Health Management Society, 2011

[10] E. Gopalakrishnan, “Quadcopter Flight Mechanics Model and Control Algorithms”, Master’s Thesis, Czech Technical University Prague, May 2017.

[11] R.Castillo, R.Lozano & A.Dzul, “Stabilization of a mini rotorcraft with four rotors. Control Systems”, IEEE, 25(6), p.45- 55. Available at: [Accessed March 3, 2018.

[12] T.Lan, Q. Fei , Q.Geng, “Control Algorithm Research of Quadrotor system based on 3-DOF Simulation Platform”, IEEE International Conference on Automation Quality and Testing Robots (AQTR). Beijing Institute of Technology. 2010

[13] S.Bouabdallah, S.Noth, R.Siegwart, “PID vs LQ Control Techniques Applied to an Indoor Micro Quadrotor”, Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004), Vol. 3, 28 September-2 October 2004, 2451-2456.

[14] A. Zulu, S.John, “A Review of Control Algorithms for Autonomous Quadrotors, Open Journal of Applied Sciences”, 2014, 4, 547-556.

[15] A.Freddi, “A Feedback Linearization Approach to Fault Tolerance in Quadrotor Vehicles”, Proceedings of the 18th World Congress The International Federation of Automatic Control Milano (Italy) August 28 - September 2, 2011.

[16] F.Sharifi, M.Mirzaei, W. Gordon, and Y.Zhang, “Fault Tolerant Control of a Quadrotor UAV using Sliding Mode Control”,Conference on Control and Fault Tolerant Systems Nice, France, October 6-8, 2010.

[17] Li, T., Y Zhang, B. Gordon, “Passive and active nonlinear fault-tolerant control of a quadrotor unmanned aerial vehicle based on the sliding mode control technique”, Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering 227(1): 12–23.

[18] J. M. Maciejowski, & C. N. Jones, “MPC fault-tolerant flight control case study: flight 1862”, IFAC Symposium on Safeprocess,Washington D.C., USA.

[19] L. Wang, T.Beumer, “Fault Tolerant Control of a Quadrotor Unmanned Aerial Vehicles”, Masters thesis, RMIT University, Tutor DBL (OGO) at Eindhoven,2016.

[20] A. Hojjat, H. Izadi, “ Fault Tolerant Model Predictive Control of Quad-Rotor Helicopters with Actuator Fault Estimation”, Preprints of the 18th IFAC World Congress Milano (Italy) August 28 - September 2, 2011.

[21] H. Khebbache, “Robust Stabilization of a Quadrotor Aerial Vehicle in Presence of Actuator Faults”, International Journal of Information Technology, Control and Automation (IJITCA) Vol.2, No.2, April 2012