Fault-Tolerant Adaptive Control Algorithm with Aerospace Applications

Dr. KVR Murthy[1] , G Satyanarayana[2], K Durga3,

Professor[1],Assistant, Professor[2,3]
[1,2,3] Dept. of Mechanical Engineering,
[1,2,3] Global Institute of Engineering & Technology, Moinabad, Rangareddy Dist., Telangana State

Abstract—

Many unmanned aerial vehicles (UAVs) and drones employ control systems such as commonly used proportional integral-derivative (PID) controllers. These control systems take 3 constant PID gains and multiply them by the error, rate of error, and accumulated error. While incredibly effective at ensuring stability, these systems do not account for the presence of faults in the actuating surfaces. The optimal usefulness of something like a PID controller depends on the aircraft not encountering any anomalies or faults in flight. If a fault were to occur, the PID controller is no longer optimal and could potentially cause a loss of control of the aircraft. Hence, the Multi-Input Multi-Output (MIMO) Model Reference Adaptive Controller (MRAC) will be applied in the context of a UAV with a possible control surface fault. This makes use of Lyapunov stability theory; we can guarantee the global asymptotic stability within an input/output interval, without knowing the exact path of motion. This theory will then be implemented on MATLAB, with a nominal linear quadratic regulator (LQR) gain matrix to initialize the PID control system. This will then be uploaded to a Pixhawk-controlled UAV for verification. Tuning the adaptive controller is crucial to ensuring that all actuators remain within saturation limits, while still delivering effective control of the aircraft during this loss in efficiency. Tuning of the controller involves analytical and empirical means. For example, the LQR gain matrix is one way to mathematically estimate the ideal values of the nominal gains. However, variables within the equations themselves must still be estimated and varied based on the response of the dynamic system

Keywords: control, robust, adaptive, UAV, fault-tolerant, Lyapunov, allocation.

INTRODUCTION

Control systems are a common way to control the output of a system by comparing it to a reference model. One of the more well-known control systems is cruise control. Based off a reference speed, the ECU is able to actuate the throttle such that the vehicle reaches that desired speed. Applied to aerospace vehicles, control systems are responsible for not only maintaining speed, but reaching the desired Euler angles, which dictate the orientation of the aircraft. One of the more common control systems is the PID controller. This uses proportional, integral, and derivative gains (PID) to modify the control input such that the system state tracks the reference state. However, this control system may not robustly account for the presence of faults or other anomalies. To account for these anomalies, two options available are of interest to us. We may include an on-board fault detection system, or modify an existing control system to adapt to sudden changes in the effectiveness of the control surfaces. On-board systems will likely add weight and require additional sensors, which are themselves prone to errors. Thus, modifying the control system, while more mathematically complex, allows for the system to adapt to changes in the performance of the aircraft. The method presented in this paper is the Multi-InputMulti-Output (MIMO) Model Reference

Adaptive Controller (MRAC). The nominal PID controller gains will be obtained via the Linear Quadratic Regulator (LQR) method. After these initial gains are obtained, the gains will be adapted if there is a fault or other unexpected disturbance present. Currently, this theory has been successfully applied to the pitch axis of a hypothetical aircraft in MATLAB. Going forward, the theory will be applied to the roll and yaw axes as well. Further research will explore the over-actuation of an axis, as well as coupling between various actuators in the system.

METHODOLOGY

Nominally, a standard PID controller could be used. Finding the optimal gains for the PID controller can be done via the Linear Quadratic Regulator described in Hespanha [1]. This will serve as a baseline for the design of the control system. Additionally, when designing around an over-actuated system, similar equations must be used as described in Liu and Crespo [3]. Before diving into the advanced control architecture of the fault-tolerant controller, it is necessary to describe the PID control architecture. For a PID controller, the state variable x is governed by an LTI system. The LTI matrices are A and B, whilst the control input u is governed by an autonomous input

$$\dot{x} = Ax + Bu \quad (1)$$

To autonomize the control input, we must multiply the PID gains by error functions.

$$e = x - x_{desired} \quad (2)$$

$$u = K_p e + K_I \int e dt + K_D \dot{e} \quad (3)$$

When finely adjusted, this produces the desired effect. However, when errors occur, the control mechanism is subpar rather than ideal.

Conforming to Industry Standard MIMO MRAC

Following is the control system architecture presented by Lavretsky and Wise [1]. The Model Reference Adaptive Controller's built-in feedback loops will be used. The components of the control system, as described by the accompanying equations, are as follows. the state vector x, the tuning diagonal matrix, the input control vector u, and

$f(x) = \Theta^T \breve{\Phi}(x)$ is the input uncertainty. The variable y is the system regulated output.

Firstly, the open-loop plant is:

$$\dot{x}_p = A_p x_p + B_p \Lambda [u + \Theta^T \Phi(x_p)] \quad (4)$$

$$y = C_p x_p \quad (5)$$

We then introduce the integrated output tracking error and the extended state vector. The extension of the plant allows for integral feedback connections, and follows similar architecture as a PID controller.

$$\dot{e}_{Iy} = y - y_{cmd} \qquad x = \begin{pmatrix} e_{yI}^T \\ x_p^T \end{pmatrix} \quad (6)$$

By incorporating this into the open-loop plant and including an external disturbance ξ(t), we may rewrite the plant state equation (4) as such.

$$\dot{x} = Ax + B\Lambda[u + \Theta^T \Phi(x_p)] + B_{ref} y_{cmd} + \xi(t) \quad (7)$$

Next, the ideal reference model is shown as:

$$\dot{x}_{ref} = A_{ref} x_{ref} + B_{ref} y_{cmd} \qquad y_{ref} = C x_{ref} \quad (8)$$

The standard error vector is described by the difference between the actual x vector and the reference x_{ref} vector:

$$e = x - x_{ref}$$

Before defining the control input, the LQR gain matrix will need to be found to initialize the gain matrix. LQR calculates the optimal gain matrix such that it minimizes the quadratic cost function given by:

$$J_{LQR} = \int_0^\infty [x(t)^T Q x(t) + u(t)^T R u(t)] dt \quad (10)$$

Here, Q and R are tuning parameters, which may be selected. A way to select these matrices is using Bryson's rule [1], which assumes that Q and R are diagonal matrices, with the diagonal elements represented by:

$$Q_{ii} = \frac{1}{x^{i,max}} \qquad R_{jj} = \frac{1}{u_{j,max}^2} \quad (11)$$

In which the maximum value of the state variable and the control input, respectively, make up the denominator term. The greatest control input is restricted in this project because to the limited movement range of the control surface actuators. This suggests that the MATLAB command l qr might be used to determine the best possible gain matrix. Following the determination of the optimum gain matrix, the following equations are used to characterise the control input. From now on, the symbol will be used to denote an approximated variable.

$$u = \hat{k}_x^T x - \hat{\Theta}^T \Phi(x_p) \quad (12)$$

The Algebraic Lyapunov Equation allows for calculation of the matrix P, and is given by:

$$P A_{ref} + A_{ref}^T P = -Q \quad (13)$$

Where Q is the tuning parameter matrix, and P is the solution to the Lyapunov Equation. Lastly, the robust control laws can be defined. Using the symmetric tuning matrices Γ, we can tune the adaptive rates to best fit our needs.

$$\dot{\hat{k}}_x = -\Gamma_x x e^T P B \quad (14)$$

$$\mu(\|e\|) = \max\left(0, \min\left(1, \frac{\|e\| - \delta e_0}{(1-\delta) e_0}\right)\right) \quad (15)$$

$$\dot{\hat{\Theta}} = \Gamma_\Theta \Phi(x) \mu(\|e\|) e^T P B \quad (16)$$

Thus, the adaptive rates (14), (16) govern the gains and adaptively change their values when deviations from the reference model occur.

Over-Actuated System- The Allocation Problem

Designing a robust adaptive controller for an over actuated system follows a similar procedure. Liu and Crespo [3] details a similar problem when dealing with an over actuated system in one axis for an aircraft in a steady turn. First, we modify the nominal control input up in terms of the defined allocation vector α.

$$u_j = \alpha_j u_0(t) \tag{17}$$

In this case, the control input u is redefined and accounts for the additional actuators. We define χ to be the extended gain vector, and also extend the state vector to include the reference r.

$$u_0 = k_1 x(t) + k_2 r(t) \triangleq \chi \omega(t) \qquad \chi = \begin{pmatrix} k_1 & k_2 \end{pmatrix} \qquad \omega(t) = \begin{pmatrix} x \\ r \end{pmatrix} \tag{18}$$

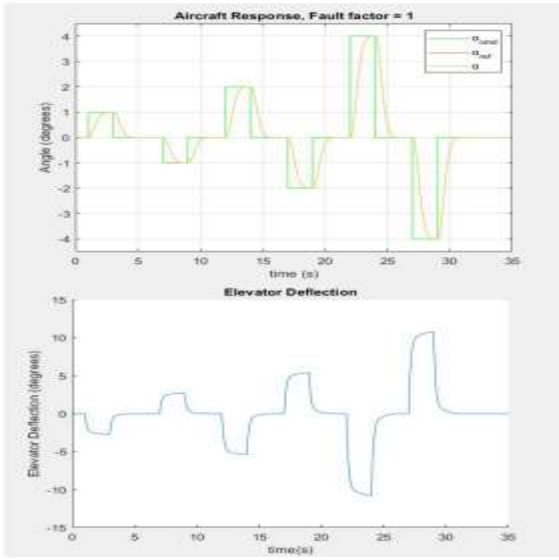
For this case, χ and α are the adaptive terms. χ is adaptive in a similar way to how the gains are adaptive for the MIMO MRAC, whilst α is adaptive due to the fact that the allocated amount of control input to each actuator can change depending on various circumstances such as faults or nominal operation. We can guarantee global asymptotic stability by defining their rates of change as a function of the tuning matrices Γ

$$\dot{\chi} = -\Gamma_\omega(t) e^T P b_0 \tag{19}$$

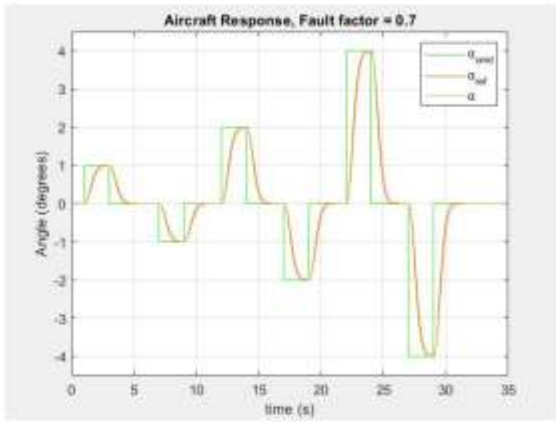
$$\dot{\alpha} = -\Gamma_\alpha B^T P e u_0(t) \tag{20}$$

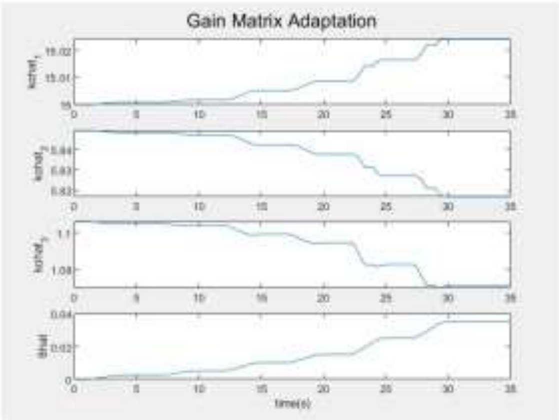
RESULTS

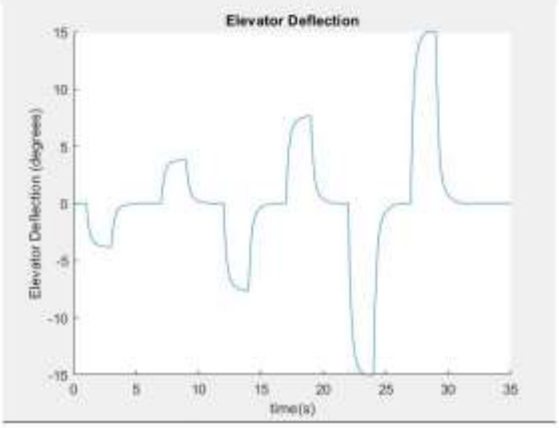
Indicated before, the system is now in charge of the hypothetical aircraft's pitch axis. This system accurately mimics the reference model in a fault-free state and works admirably in the face of input uncertainty and environmental disturbances. One set of images depicts normal operation.



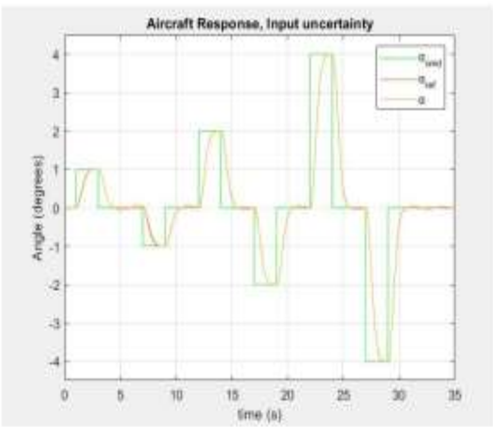
The adaptive gains also stay constant as a result of the perfect modelling of the ideal system. Additionally, the system is able to account for the presence of a fault. If the elevator efficiency is set to 0.7, the aircraft responds in the following manner.





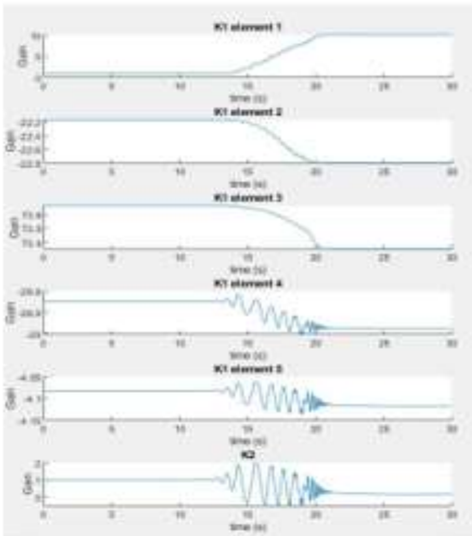


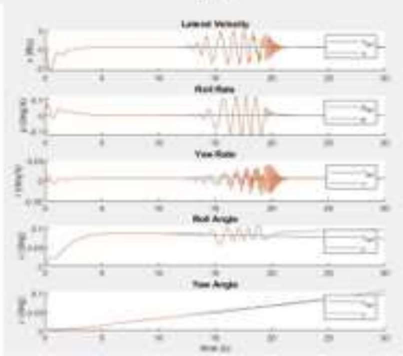
It's no surprise that the elevators' reduced efficiency necessitates more effort on the part of the user to obtain the same height. Indeed, mechanical failure of the actuators or control surfaces may be to blame for this efficiency drop. Since the techniques utilised cannot measure the true deflection, the graph of elevator deflection does not show the actual deflection. This, on the other hand, is the input that was ordered into the lifts. Similar behaviour is also seen when input uncertainty is present.



These simulations demonstrate the stability of the control system, and further efforts will centre on applying and refining the control equations to the UAV.

Using the numbers found in Liu and Crespo [3], we also looked at the issue of over-actuation. An aeroplane banking at a constant angle is modelled in this study. After 10 seconds, we intentionally introduce a malfunction to activate the adaptive pricing. Since the plane can't fly in a stable state with the problem, the gains are changed so that the aircraft's dynamics are better represented. The following were the outcomes of their MATLAB modelling of the data.





CONCLUSIONS

Now that the pitch axis control system is finished, scientists may investigate how best to incorporate it into a real aeroplane. The governing LTI equations in this research were tailored to the hypothetical aircraft employed in the investigation. Calculating approximations of LTI matrices requires determining dynamic equations of motion for a real aircraft.

Furthermore, coupling must be accounted for in the equations describing each axis. Inducing forces along the yaw and pitch axes may occur, for instance, when the ailerons are deflected along the roll axis. This is related to the allocation issue as a whole, and it may be utilised to limit the damage done by the failure of a control surface over one axis while it's used to stabilise another. Finally, the algorithm's viability is heavily dependent on its incorporation into an HIL simulation. This study will go toward hardware implementation with the help of a Pixhawk 4. It is expected that additional challenges will arise throughout the process of developing and testing this algorithm on the Pixhawk. Kalman Filtering is a useful tool for eliminating the controller's noise and preserving the integrity of the data being sent to and from the computer. In the future, this algorithm may be adapted for use on a Pixhawk aboard a UAV or quadcopter.

REFERENCES

[1] Lavretsky, E.; and Wise, K. A. Robust and Adaptive Control: with Aerospace Applications. Springer. (2013)

[2] Hespanha, J. P. Linear Systems Theory. Princeton University Press. (2018)

[3] Liu, Y.; and Crespo, L. G. Adaptive Control Allocation in the Presence of Actuator Failures. Hindawi Publishing Corporation. (2012)

[4] Nguyen, T. Model-Reference Adaptive Control. Springer. (2018)

[5] Nelson, R. C. Flight Stability and Automatic Control. WCB McGrawHill. (1998).