IMAV2017-15
http://www.imavs.org/pdf/imav.2017.15

Flight Simulation of a MAKO UAV for Use in Data-Driven Fault Diagnosis

Elgiz Baskaya,* Murat Bronz, and Daniel Delahaye
ENGIE Ineo - Groupe ADP - SAFRAN RPAS Chair, ENAC,
UAS Lab, ENAC,
Laboratoire MAIAA, ENAC

ABSTRACT

Last decade witnessed the rapid increase in number of drones of various purposes. This pushes the regulators to rush for safe integration strategies in a way to properly share the utilization of airspace. Accommodating faults and failures is one of the key issues since they constitute the bigger chunk in the occurrence reports available. The hardware limitations for these small vehicles point the utilization of analytical redundancy rather than the usual practice of hardware redundancy in the conventional flights. In the course of this study, fault detection and diagnosis for aircraft is reviewed. Then a nonlinear model for MAKO aircraft is simulated to generate faulty and nominal flight data. This platform enables to generate data for various flight conditions and design machine learning implementations for fault detection and diagnosis.

1 INTRODUCTION

Unmanned Aircraft Systems (UAS) are becoming more efficient platforms everyday for scientific/commercial domains offering benefits in terms of cost, flexibility, endurance as well as realizing missions that would be impossible with a human onboard. Increasing usage of these vehicles for a variety of missions, such as defense, civilian tasks including transportation, communication, agriculture, disaster mitigation applications pushes demand on the airspace. Furthermore, this congestion is predicted to accelerate with the growing diversity of these vehicles[1].

Improvement of the reliability of the flight is considered to be one of the main goals for integrating UAVs into civil airspace according to Unmanned systems roadmap by US Office of the Secretary of Defense, DoD [2]. To achieve a safe flight is not an easy task considering the unknowns of the systems hardware, environment and possible system faults and failures to emerge. Also, increasing demand on cost effective systems, resulting in the smaller sensors and actuators with less accuracy, impose the software to achieve even more. The expectation that UAVs should be less expensive than their manned counterparts might have a hit on reliability of the system. Cost saving measures other than the need to support a pilot/crew onboard or decrement in size would probably lead to decrease in system reliability.

Systems are often susceptible to faults of different nature. Existing irregularities in sensors, actuators, or controller could be amplified due to the control system design and lead to failures. A fault could be hidden thanks to the control action [3].

Under the research and development programs and initiatives identified by DoD in order to develop technologies and capabilities for UAS, the biggest chunk in control technologies is the health management and adaptive control with a budget of 74.3 M dollars. Other safety features such as validation and verification of flight critical intelligent software is the second with 57.8 M dollars [2].

The widely used method to increase reliability is to use more reliable components and/or hardware redundancy. Both requires an increase in the cost of the UAS conflicting one of the main reasons of UAS design itself band consumer expectations [4]. To offer solutions for all different foreseen categories of airspace, a variety of approaches should be considered. While hardware redundancy could cope with the failure situations of UAVs in the certified airspace, it may not be suitable for UAVs in open or some subsets of specific categories due to budget constraints. Analytical redundancy is another solution, may be not as effective and simple as hardware redundancy, but relies on the design of intelligent methods to utilize every bit of information onboard aircraft wisely to deal with the instances.

There are three approaches to achieve safe FTC in standard flight conventions. First one is the fail operational systems which are made insensitive to any single point component failure. The second approach is the fail safe systems where a controlled shut down to a safe state is practiced whenever a critical fault is pointed out by a sensor. The level of degradation assures to switch to robust (alternate) or direct (minimal level of stability augmentation independent of the nature of the fault) mode. Switching from nominal mode to the robust and direct modes leads to a decrease in the available GNC functions. This causes a degradation in ease of piloting. And also some optimality conditions could have been compromised. The third approach is fault tolerant control systems in which redundancy in the plant and the

*Email addresses: elgiz.baskaya@enac.fr, murat.bronz@enac.fr, daniel.delahaye@enac.fr

automation system is employed to design software that monitors the components and takes in action whenever needed. The strategy is most probably to try to keep plant availability and accept reduced performance [5].

RECONFIGURE project of FP7 [6] aims to attack at this problem of piloting degradation and optimality compromisation by attacking Flight Parameter Estimation (FPE) which is the online estimation of aircraft parameters, FDD and FTC in case of off-nominal events [7] They utilize a black box nonlinear model of aircraft and The project uses some outputs of a previous FP 7 project ADDSAFE leaded by Deimos Space [8].

2 Methods for FTCS

Since fault tolerant control is comprised of a set of different disciplines and a relatively new topic, the terminology is not solid. FDI could be a proper example to this ambiguity. In some works, it stands for Fault Detection and Isolation while in some other Fault Detection and Identification, which could also named after Fault Detection and Diagnosis, meaning that identification is added to Fault Detection and Isolation [9].

One of the first attempts to unify the terminology is carried out by IFAC SAFEPROCESS technical committee in 1996 and published by [10]. Fault, failure, and the methodology to handle those such as fault detection, fault isolation, fault identification, fault diagnosis and supervision terms explained separately to avoid the ongoing ambiguity in this field. Although fault detection methods are clearer in the work, difference between the methods for two steps of fault diagnosis, namely the fault isolation and fault identification is not very obvious.

Among different categorizations for the fault tolerance, there are options to handle faults on-line or off-line. Employing fault diagnosis schemes on-line is a way to achieve fault tolerance. In this case, as soon as a fault detected, a supervisory agent is informed via a discrete event signal. Then accommodation of the faults are handled either with the selection of a predetermined controller for the specific fault case, or by designing the action online with real-time analysis and optimization [5].

Another common categorization of FTCS is passive and active FTCS. In passive FTCS, the flight controller is designed in such a way to accommodate not only the disturbances but also the faults. Most of the times it a robust controller and does not require a diagnosis scheme. Active FTCS first distinguishes the fault via fault detection and diagnosis module and then switch between the designed controllers specific to the fault case or design a new one online [4]. While active FTCS requires more tools to handle faults as seen in Fig. 1, for faults not predicted and not counted for during the design of the robust controller, robust controller most probably fails.

Even with a long list of available methods, aerospace industry has not implemented FTC widely, except some space

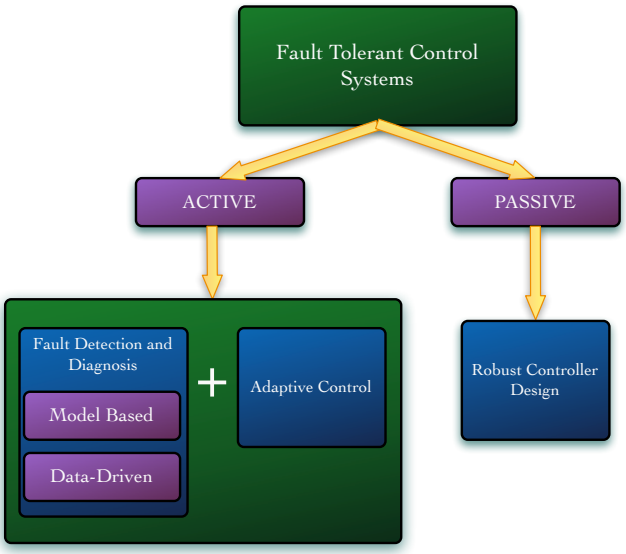

Figure 1: Variations of fault tolerant control systems

systems, due to the evolving nature of the methods, the tricks coming with the nonlinear nature of the problem, design complexity and high possibility of wrong alarms in case of large disturbances and/or modeling uncertainties. So the already carried reliability measures concerning the hardware redundancy is now the preferred way because of its ease and maturity being implemented on various critical missions with considering human lives.

3 Fault Detection and Diagnosis

FDD is handled in two main steps; fault detection and fault diagnosis. Fault diagnosis encapsulates fault isolation and fault identification. The methods for detection and diagnosis are investigated for their frequency of utilization separately for sensor, actuator, process and controller faults in [10]. FDD should not only be sensitive to the faults but also robust to the model uncertainties and external disturbances.

Two distinct options to proceed in analytical redundancy are the model based approaches and data-driven approaches. They form the two ends of a continuous solution set line, so utilizing them in a combination might end up with better solutions. Model based fault diagnosis highlights the components of a system and the connections in-between, and their corresponding fault modes. Data driven fault diagnosis rely on the observational data and prefers dense, redundant and with a frequency larger than the failure rate.

This work constitutes the basis for our research on fault detection. The idea to simulate the data using the MAKO model given here first, rather than utilizing flight data, is to start small in order to isolate some probable consequences

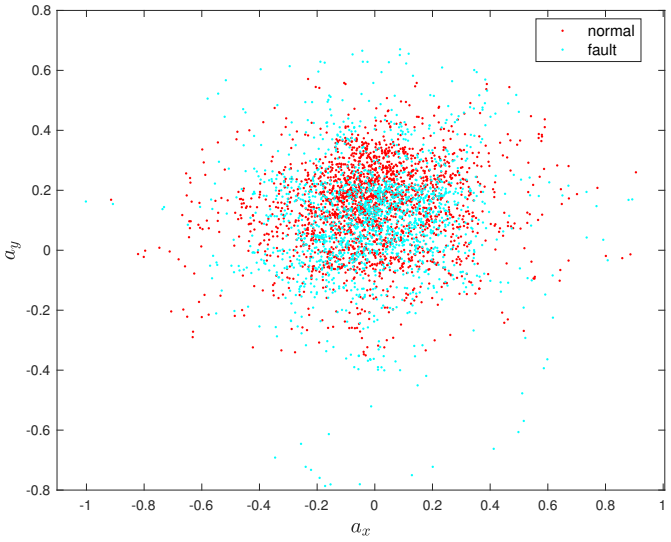

Figure 2: Accelerometer data readings a_x vs a_y

Figure 3: MAKO

such as the probable effect of the controller on the diagnosis. Detection of faults from real data is a challenging goal to start with as can be seen in Figure 2 real data accelerometer readings showing that it seems impossible to classify in case of a fault with one of the control surfaces is 50% less efficient.

Most of the FDI algorithms are implemented to open-loop systems, ignoring the probable influences of the controller might cause on the detection performance [11]. Here the system is open-loop as well. So we follow a step by step approach and hope to end with a more realistic case in the future, in which real flight data is utilized and diagnosis is achieved on-line aside a functioning controller. Here we present a literature survey for FTC of drones followed by efforts to deliver a full drone simulation which will serve as an environment to simulate measurements. A MAKO simulation is given in Matlab script which is freely available though the GIT [1] platform, using specifications MAKO, stability derivatives, aerodynamic force derivatives generated by AVL.

4 Methodology and Simulations

In this study, first, a model of an aircraft is simulated. This model, will not be used for the design of FDI algorithms, but instead will be utilized to test them. Nonlinear aircraft flight dynamics for translational and attitude motion can be given as a system of first order differential equations

$$\dot{\boldsymbol{x}}_n = \boldsymbol{C}_b^n \boldsymbol{v}^b \tag{1}$$

$$\dot{\boldsymbol{v}}^b = \frac{1}{m}\left[m\boldsymbol{g}^b + \boldsymbol{F}_t^b + \boldsymbol{F}_a^b\right] - {\boldsymbol{\omega}_{b/n}^b}^{\times}\boldsymbol{v}^b \tag{2}$$

$$\dot{q}_0 = -\frac{1}{2}\boldsymbol{q}_\nu^T \boldsymbol{\omega}_{b/n}^b \tag{3}$$

$$\dot{\boldsymbol{q}}_\nu = \frac{1}{2}\left(\boldsymbol{q}_\nu^{\times} + q_0 \boldsymbol{I}_3\right)\boldsymbol{\omega}_{b/n}^b \tag{4}$$

$$\boldsymbol{J}\dot{\boldsymbol{\omega}}_{b/n}^b = \boldsymbol{M} - {\boldsymbol{\omega}_{b/n}^b}^{\times}\boldsymbol{J}\boldsymbol{\omega}_{b/n}^b \tag{5}$$

where $\boldsymbol{x}_n \in \mathbb{R}^3$ is the position of the center of mass of UAV in navigation frame $\mathcal{N}$, $\boldsymbol{v}^b$ is the velocity of the center of mass of UAV in body frame $\mathcal{B}$, $\boldsymbol{q} = [q_0, \boldsymbol{q}_v^T]^T \in \mathbb{R}^3 \times \mathbb{R}$ is the unit quaternion representing the attitude of the body frame $\mathcal{B}$ with respect to navigation frame $\mathcal{N}$ expressed in the body frame $\mathcal{B}$, $\boldsymbol{\omega}_{b/n}^b$ is the angular velocity of the body frame $\mathcal{B}$ with respect to navigation frame $\mathcal{N}$ expressed in the body frame $\mathcal{B}$, $\boldsymbol{J} \in \mathbb{R}^{3\times3}$ is the positive definite inertia matrix of the drone, $\boldsymbol{M} \in \mathbb{R}^3$ represents the moments acting on the drone, $\boldsymbol{C}_b^n$ is the direction cosine matrix which transforms a vector expressed in the body frame to its equal expressed in the navigation frame, $\boldsymbol{I}_3 \in \mathbb{R}^{3\times3}$ is the identity matrix, $\boldsymbol{F}_t^b \in \mathbb{R}^3$ is the thrust force expressed in the body frame, $\boldsymbol{F}_a^b \in \mathbb{R}^3$ are the aerodynamic forces given in the body frame. The navigation frame is assumed to be a local inertial frame in which Newton's Laws apply. The notation $\boldsymbol{x}^{\times}$ for a vector $x = [x_1 \quad x_2 \quad x_3]^{\mathrm{T}}$ represents the skew-symmetric matrix

$$\boldsymbol{x}^{\times} = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix} \tag{6}$$

[1] https://github.com/benelgiz/curedRone/tree/MAKOmodel

The stability derivatives and aerodynamic force coefficients are generated by AVL and given in Appendix A. AVL is an open source program developed at MIT and uses vortex-lattice method for the aerodynamic and stability calculations. The output of the program is linearized at a selected condition, therefore all the coefficients are calculated around the equilibrium point at $14m/s$ cruise flight condition. The center of gravity is located at $X_{CG} = 0.295\,m$, which corresponds to a $8\,\%$ of positive static margin that has been flight tested.

As an addition to the aerodynamic coefficients and stability derivatives, it is useful to have the moments of inertia of the aircraft so that one can use the model in a simulator. For that purpose, the aircraft is hanged by two strings, at different orientations, as shown in Figure 4, and measurements performed by timing the oscillation period for each axis. The resultant moment of inertias are given Table 2.

Further, the equations for calculation of forces and moments are given in Appendix B to simulate translational and rotational motion of a MAKO UAV.

The input vector can be written as $\boldsymbol{u}\,(t) \in \mathbb{R}^3$

$$\boldsymbol{u}\,(t) = \left[\delta_a\ \delta_e\ n\right]^{\mathrm{T}} \tag{7}$$

Here δ_a aileron deflection angle in degrees, δ_e elevator deflection angle in degrees, n engine speed in rev/s.

To validate the written translational and attitude motion dynamics and kinematics, MATLAB Simulink *6DOF* block has been utilized. This block accepts inputs as the force and moment and outputs the states of aircraft motion Fig. 6. To compare the generated model and Simulink *6DOF* block, forces and moments have been calculated via equations and constants given in Appendix A and Appendix B. The simulated states from the model script have been saved in advance and called from Simulink by *From Workspace* blocks then compared with the *6DOF* outputs. The difference found to be negligible indicating the validity of the model.

When the actuators are healthy, actual control input signal will be equal to the given input signal. In case of a fault the actual signal can be modeled as

$$\boldsymbol{u}\,(t) = \boldsymbol{E}\boldsymbol{u}_c + \boldsymbol{u}_f \tag{8}$$

where $\boldsymbol{u}_c$ is the desired control signal, $\boldsymbol{E} = diag(e_1, e_2, e_3)$ is the effectiveness of the actuators where $0 \leq e_i \leq 1$ with $(i = 1, 2, 3)$ and $\boldsymbol{u_f}$ additive actuator fault. This model makes it possible to simulate all four types of actuator faults shown in Fig. 5.

The measurements are simulated using the statistics of the hardware in the house. The sensor suit simulated is the InvenSense MPU-9250 Nine-axis (Gyro + Accelerometer + Compass) MEMS MotionTracking Device.

$$\boldsymbol{z}_{gyro} = \boldsymbol{k}_{gyro}\boldsymbol{\omega}^b_{b/i} + \boldsymbol{\beta}_{gyro} + \boldsymbol{\eta}_{gyro} \tag{9}$$

$$\boldsymbol{z}_{acc} = \boldsymbol{k}_{acc}\boldsymbol{\omega}^b_{b/i} + \boldsymbol{\beta}_{acc} + \boldsymbol{\eta}_{acc} \tag{10}$$

Here $\boldsymbol{\beta}$ is the bias, and $\boldsymbol{\eta}$ is the zero mean Gaussian process with $\boldsymbol{\sigma}^2$ variance and given in Table 3. For faulty and normal measurement values, drone model simulation which outputs the measurements as well should be run twice with different control surface input values. As an example, a faulty situation can be that even the controller gives an desired output of 4 degrees to the control surface, the control surface might have stuck at 1 degrees. Generated set of measurements can be visualized in feature space one by one. Such an example is the normalized accelerometer measurement components plotted Fig. 7.

It is always important to visualize the features to have a grasp of data structure. For that reason, available observations forms the 6-dimensional pattern space, $\boldsymbol{z} \in \mathbb{R}^6$ can be visualized in pairs to observe. There are further methods to visualize multidimensional data such as Tours methods [12, 13, 14], and GGobi data visualization system [15].

In this study, dimensionality reduction technique called Principle Component Analysis (PCA) is used for visualization. In PCA, the idea in general is to map the feature vector, $\boldsymbol{x} \in \mathbb{R}^n$ to a lower dimensional space where the new feature set will be represented by $\boldsymbol{z} \in \mathbb{R}^k$. Fig. 8 shows the resulted most significant elements for a mapped feature space from six dimensional feature vector to two. The structure of the data gives insight for the selection of some parameters or kernels for the purpose of classification. Here, it seems that a linear kernel is satisfactory by discarding the outliers. Another point is that the classifier might need a nice tuning due the the presence of outliers. The learning phase utilizes data including outliers and preciseness to fit the model to each of the data might end up an overfitted model, resulting in worse performance to generalize to new data coming in the prediction phase.

5 Conclusion

In this work, first a review on fault tolerant control for UAVs is given by pointing out its importance on today's challenging task of safe integration of drones into airspace. Data-driven methods for fault diagnosis is aimed to avoid the burden of modeling each craft especially considering for small drones it is not very realistic for most of the applications to have an accurate model for a variety reasons such as cost. AVL program is used to generate the coefficients for MAKO and a full simulation is realized. Statistics of the sensor suite in house is used for simulation of accelerometer and gyro data. For a preliminary investigation on data, six dimensional feature space is mapped to two dimensions via PCA for visualization purposes. The data shows that a linear kernel might be satisfactory for the purpose of two class classification. Due

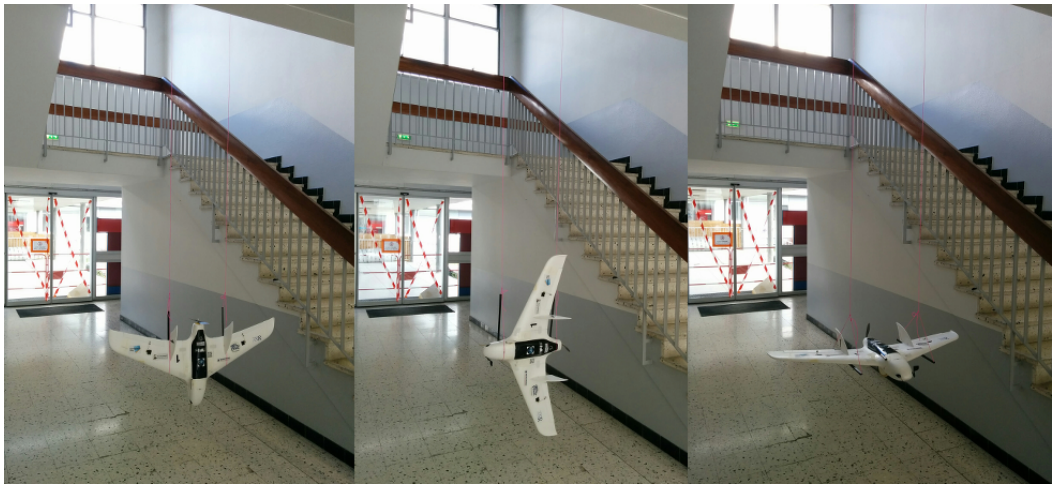

Figure 4: Moments of inertia measurements for each axis, I_{xx}, I_{yy}, I_{zz}.

to the presence of outliers, fine tuning or using optimization techniques could be needed to avoid overfitting or under fitting during the learning phase of the classification problem.

ACKNOWLEDGEMENTS

This work was supported by the ENGIE Ineo - Groupe ADP - SAFRAN RPAS Chair. Special thanks to Gautier Hattenberger and Torbjoern Cunis for code modifications to Paparazzi autopilot system, Xavier Paris, Michel Gorraz and Hector Garcia de Marina, and the rest of the ENAC Drone Lab for their help during test flights. Last but not least, we acknowledge Paparazzi community for their contributions to the autopilot system.

REFERENCES

[1] Elgiz Baskaya, Guido Manfredi, Murat Bronz, and Daniel Delahaye. Flexible open architecture for uass integration into the airspace: Paparazzi autopilot system. In *Digital Avionics Systems Conference (DASC), 2016 IEEE/AIAA 35th*, pages 1–7. IEEE, 2016.

[2] Unmanned Systems Roadmap 2005 - 2030, 2005.

[3] Guillaume JJ Ducard. *Fault-tolerant flight control and guidance systems: Practical methods for small unmanned aerial vehicles*. Springer Science & Business Media, 2009.

[4] Plamen Angelov. *Sense and avoid in UAS: research and applications*. John Wiley & Sons, 2012.

[5] Mogens Blanke, Christian W Frei, Franta Kraus, Ron J Patton, and Marcel Staroswiecki. What is fault-tolerant control. In *Preprints of 4th IFAC Symposium on Fault Detection Supervision ans Safety for Technical Processes, SAFEPROCESS*, pages 40–51, 2000.

[6] Philippe Goupil, Josep Boada-Bauxell, Andres Marcos, Paulo Rosa, Murray Kerr, and Laurent Dalbies. An overview of the fp7 reconfigure project: industrial, scientific and technological objectives. *IFAC-PapersOnLine*, 48(21):976–981, 2015.

[7] RECONFIGURE FP7 Project. reconfigure.deimos-space.com/, Accessed: 2016-07-19.

[8] ADDSAFE FP7 Project. http://addsafe.deimos-space.com/, Accessed: 2016-07-09.

[9] Youmin Zhang and Jin Jiang. Bibliographical review on reconfigurable fault-tolerant control systems. *Annual reviews in control*, 32(2):229–252, 2008.

[10] Rolf Isermann and Peter Ballé. Trends in the application of model-based fault detection and diagnosis of technical processes. *Control engineering practice*, 5(5):709–719, 1997.

[11] Rohit Pandita, József Bokor, and Gary Balas. Closed-loop performance metrics for fault detection and isolation filter and controller interaction. *International Journal of Robust and Nonlinear Control*, 23(4):419–438, 2013.

[12] Daniel Asimov. The grand tour: a tool for viewing multidimensional data. *SIAM journal on scientific and statistical computing*, 6(1):128–143, 1985.

[13] Dianne Cook and Andreas Buja. Manual controls for high-dimensional data projections. *Journal of computational and Graphical Statistics*, 6(4):464–480, 1997.

[14] Dianne Cook, Andreas Buja, Javier Cabrera, and Catherine Hurley. Grand tour and projection pursuit. *Journal of Computational and Graphical Statistics*, 4(3):155–172, 1995.

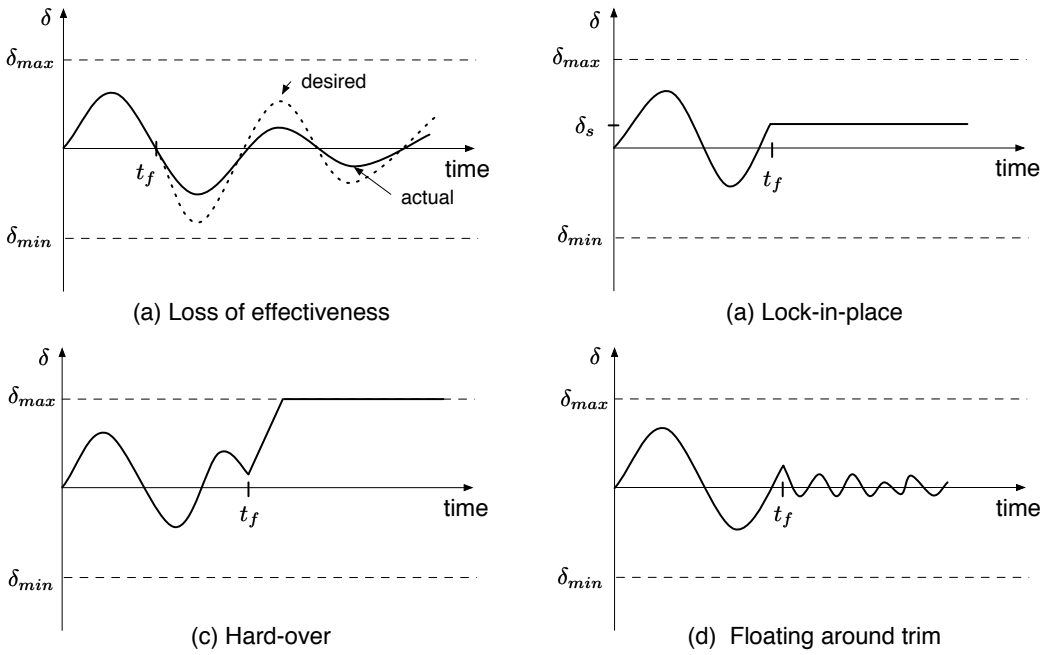

Figure 5: Common actuator faults [3]

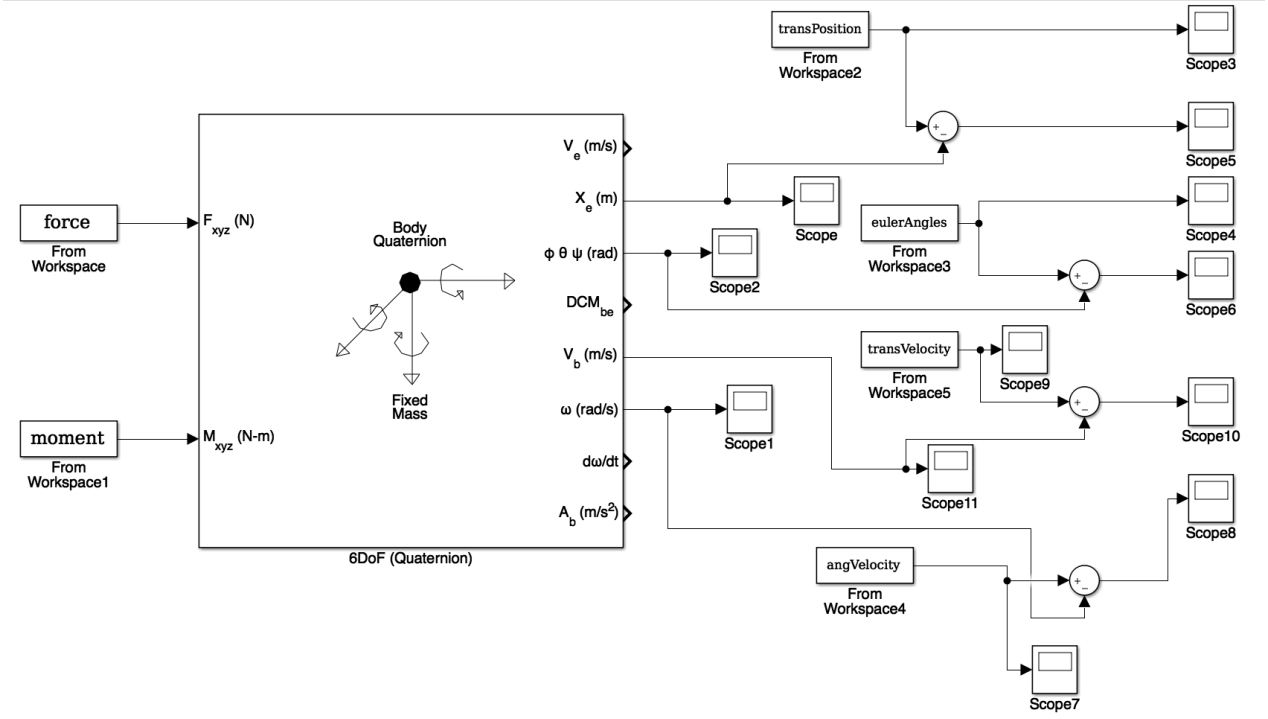

Figure 6: Validation with Simulink 6DOF aircraft model

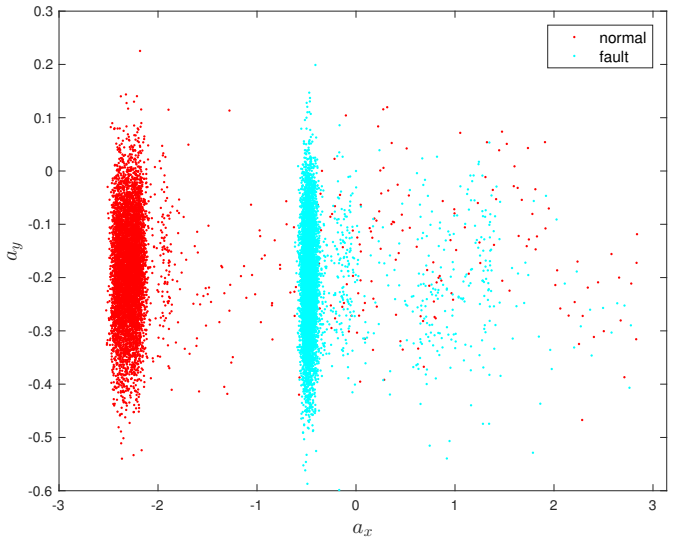

Figure 7: Accelerometer simulation a_x vs a_y

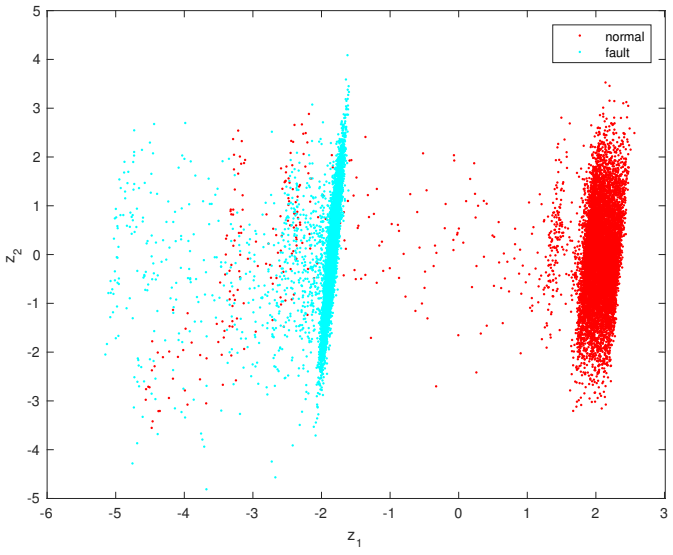

Figure 8: Reduced feature space z_1 vs z_2

[15] Dianne Cook and Deborah F Swayne. *Interactive and dynamic graphics for data analysis: with R and GGobi*. Springer Science & Business Media, 2007.

[16] Murat Bronz and Gautier Hattenberger. Aerodynamic characterization of an off-the-shelf aircraft via flight test and numerical simulation. In *AIAA Flight Testing Conference*, page 3979, 2016.

[17] Murat Bronz, Hector Garcia de Marina, and Gautier Hattenberger. In-flight thrust measurement using on-board force sensor. In *AIAA Atmospheric Flight Mechanics Conference*, page 0698, 2017.

[18] Jean-Philippe Condomines. *Développement d?un estimateur d?état non linéaire embarqué pour le pilotage-guidage robuste d?un micro-drone en milieu complexe*. PhD thesis, INSTITUT SUPERIEUR DE L'AERONAUTIQUE ET DE L'ESPACE (ISAE), 2015.

APPENDIX A: MAKO COEFFICIENTS

Table 1: General specifications of MAKO [16]

Parameter	**Value**	**Definition**
Wing span	1.288	$[m]$
Wing surface area	0.27	$[m^2]$
Mean aero chord	0.21	$[m]$
Take-off mass	$0.7 - 2.0$	$[kg]$
Flight velocity	$10 - 25$	$[m/s]$
I_{xx}	0.02471284	$[kg \cdot m^2]$
I_{yy}	0.015835159	$[kg \cdot m^2]$
I_{zz}	0.037424499	$[kg \cdot m^2]$

Table 2: Specifications of the sensor suit InvenSense MPU-9250 Nine-axis (Gyro + Accelerometer + Compass) MEMS MotionTracking Device[18]

Measurement	β	σ
z_{accx}	0.142	0.0319
z_{accy}	-0.3	0.0985
z_{accz}	0.19	0.049
z_{gyro_x}	-1.55	0.0825
z_{gyro_y}	-1.13	0.1673
z_{gyro_z}	-1.7	0.2214

Table 3: Stability derivatives for MAKO extracted from AVL program at $14m/s$ equilibrium cruise speed

Parameter	Value	Definition
C_{L_a}	-0.1956×10^{-2}	roll derivative
$C_{L_{\tilde{p}}}$	-4.095×10^{-1}	roll derivative
$C_{L_{\tilde{r}}}$	6.203×10^{-2}	roll derivative
C_{L_β}	3.319×10^{-2}	roll derivative
C_{M_0}	0	pitch derivative
C_{M_e}	-0.076×10^{-1}	pitch derivative
$C_{M_{\tilde{q}}}$	-1.6834	pitch derivative
C_{M_α}	-32.34×10^{-2}	pitch derivative
C_{N_a}	-0.0126×10^{-2}	yaw derivative
$C_{N_{\tilde{p}}}$	-4.139×10^{-2}	yaw derivative
$C_{N_{\tilde{r}}}$	-0.1002×10^{-1}	yaw derivative
C_{N_β}	2.28×10^{-2}	yaw derivative

Table 4: Aerodynamic force derivatives for MAKO extracted from AVL program at $14m/s$ equilibrium cruise speed

Parameter	Value	Definition
C_{Z_0}	-8.53×10^{-2}	lift derivative
C_{Z_α}	3.9444	lift derivative
C_{Z_q}	4.8198	lift derivative
C_{Z_e}	1.6558×10^{-2}	lift derivative
C_{X_0}	2.313×10^{-2}	drag derivative
C_{X_k}	1.897×10^{-1}	drag derivative
C_{Y_β}	-2.708×10^{-1}	side force derivative
$C_{Y_{\tilde{p}}}$	1.695×10^{-2}	side force derivative
$C_{Y_{\tilde{r}}}$	5.003×10^{-2}	side force derivative
C_{Y_a}	0.0254×10^{-2}	side force derivative

Table 5: Thrust force coefficients for propeller APC SF 9×6 from wind tunnel experiments [17]

Parameter	Value	Definition
$C_{F_{T1}}$	1.342×10^{-1}	thrust derivative
$C_{F_{T2}}$	-1.975×10^{-1}	thrust derivative
$C_{F_{T_{rpm}}}$	7.048×10^{-6}	thrust derivative
D	$0.228\,m$	propeller diameter

APPENDIX B: FORCE, MOMENT CALCULATIONS

Roll torque

$$L_B = \bar{q}\, S\, b\, C_L \tag{11}$$

$$C_L = C_{L_a}\, \delta_a + C_{L_{\tilde{p}}}\, \tilde{p} + C_{L_{\tilde{r}}}\, \tilde{r} + C_{L_\beta}\, \beta \tag{12}$$

Pitch torque

$$M_B = \bar{q}\, S\, \bar{c}\, C_M \tag{13}$$

$$C_M = C_{M_e}\, \delta_e + C_{M_{\tilde{q}}}\, \tilde{q} + C_{M_\alpha}\, \alpha \tag{14}$$

Yaw torque

$$N_B = \bar{q}\, S\, b\, C_N \tag{15}$$

$$C_N = C_{N_a}\, \delta_a + C_{N_{\tilde{p}}}\, \tilde{p} + C_{N_{\tilde{r}}}\, \tilde{r} + C_{N_\beta}\, \beta \tag{16}$$

$$\bar{q} = \frac{\rho V_T^2}{2} \tag{17}$$

Lift force

$$Z^w = \bar{q}\, S\, C_Z(\alpha) \tag{18}$$

$$C_Z(\alpha) = C_{Z_0} + C_{Z_\alpha}\alpha \tag{19}$$

Drag force

$$X^w = \bar{q}\, S\, C_Z(\alpha, \beta) \tag{20}$$

$$C_X(\alpha) = C_{X_1} + C_{X_k}\, C_Z^2 = C_{X_1} + C_{X_k}\, (C_{Z_1} + C_{Z_\alpha}\alpha)^2 \tag{21}$$

Lateral force

$$Y^w = \bar{q}\, S\, C_Y(\beta) \tag{22}$$

$$C_Y(\beta,\, \tilde{p},\, \tilde{r},\, \delta_a) = C_{Y_\beta}\beta + C_{Y_{\tilde{p}}}\, \tilde{p} + C_{Y_{\tilde{r}}}\, \tilde{r} + C_{Y_a}\, \delta_a \tag{23}$$

Thrust force model

$$F_T = \rho n^2 D^4 C_{F_T} \tag{24}$$

$$C_{F_T} = C_{F_{T1}} + C_{F_{T2}} J + C_{F_{T3}} J^2 \tag{25}$$

$$J = \frac{V_T}{n\pi D} \tag{26}$$