International Journal of Scientific & Engineering Research, Volume 5, Issue 4, April-2014 304
ISSN 2229-5518

Design and Simulation of Integrated FDD-RC for an UAV

N.Mahalakshmi,Dr.S.Lakshmi

Abstract— According to a reliability study conducted by the US Office of the Secretary of Defense, about 80 % of flight incidents concerning unmanned aerial vehicles (UAV) are due to faults affecting propulsion, flight control surfaces, or sensors. To allow autonomous aerial vehicles to continue their missions, there is an absolute necessity to identify unexpected changes (faults) in the system before they lead to a complete breakdown (failure).The aim of the paper is to present the concept of the integrated system dedicated for Fault detection and diagnosis and Reconfigurable controller for the unmanned aerial vehicle (UAV). This work combines Stateflow with Simulink to efficiently model hybrid systems. This type of modeling is particularly useful for systems that have numerous possible operational modes based on discrete events. Traditional signal flow is handled in Simulink while changes in control configuration are implemented in Stateflow. In this paper, representation of a fuel control system for a gasoline engine is alsoimplemented. The system is highly robust in that individual sensor failures are detected and the control system is dynamically reconfigured for uninterrupted operation.

Index Terms— AFTCS, Control logic, EGO sensor ,Fault, FTCS, , MAP sensor,Throttle Sensor.

1 Introduction

The term UAV is an abbreviation of Unmanned Aerial vehicle, meaning aerial vehicles which operate without a human pilot. UAVs are commonly used in both the military and police forces in situations where the risk of sending a human piloted aircraft is unacceptable, or the situation makes using a manned aircraft impractical. Only a handful of systems are capable of carrying weapons. One of the predecessors of today's fully autonomous UAVs were the "aerial torpedoes", designed and built during World War One. More advanced UAVs used radio technology for guidance, allowing them to fly missions and return. They were constantly controlled by a human pilot, and were not capable of flying themselves. This made them much like today RC model airplanes which many people fly as a hobby. After the invention of the in3tegrated circuit, engineers were able to build sophisticated UAVs, using electronic autopilots. It was at this stage of development that UAVs became widely used in military applications. UAVs could be deployed, fly themselves to a target location, and either attack the location with weapons, or survey it with cameras and other sensor equipment. Modern UAVs are controlled with both autopilots, and human controllers in ground stations. This allows them to fly long, uneventfully flights under their own control, and fly under the command of a human pilot during complicated phases of the mission.

- *N.Mahalakshmi is currently pursuing masters degree program inApplied Electronics in Tagore Engineering College, India,. E-mail: mahalakshmi4191@gmail.com*
- *Dr.s.Lakshmi,Professor,Electronics andCommunication Engineering , Tagore Engineering College,India, E-mail: drlakshmi79@gmail.com*

UAVs represent an area of rapid development in both military and civilian applications. UAVs unique capability of flying dangerous, long, or precision missions give it a unique advantage over conventional aircraft. In the next section an introduction is given to the Fault detection and diagnosis. Next to it describes Fault tolerant Fuel Control system and the results are given in the section that follows.

2 Fault Detection And Diagnosis

2.1 Fault Tolerant Control System

A fault is an unpermitted deviation of at least one characteristic property or parameter of the system from acceptable/usual/standard conditions. A fault may lead to a failure, which is a permanent interruption of the system ability to perform a required function under specified operating conditions. When a fault occurs in a system, the main problem to be addressed is to raise an alarm, ideally diagnose what fault has occurred, and then decide how to deal with it. The problem of detecting a fault, finding the source/location and then taking appropriate action is the basis of fault tolerant control. Three types of faults are generally distinguished according to the part of the system they affect.

- Sensor fault.
- Actuator fault.
- Process fault.

A sensor fault is an abnormal variation in measurements, e.g. a systematic error abruptly affecting the value provided by an accelerometer. An actuator fault is a malfunction on a device acting on the system dynamics, e.g. the locking in- place of a flight control surface. Process faults are changes in the Inner parameters of the system that modify its dynamics, such as an un-modeled change in aerodynamic coefficients.

ISSN 2229-5518

Fault tolerant control systems (FTCS) as control systems that possess the ability to accommodate system component failures automatically. They are capable of maintaining overall system stability and acceptable performance in the event of such failures. FTCS were also known as self-repairing, reconfigurable, re-structurable, or self designing control systems. FTCS can be classified into two categories. The classification of FTC can be well explained with figure 1. The major types are given below:

1) Passive FTCS
2) Active FTCS

2.2 Passive FTCS

In passive fault tolerant control systems, the controller is designed to be robust against faults and uncertainty. Therefore when a fault occurs, the controller should be able to maintain stability of the system with an acceptable degradation in performance. PFTCS does not require FDD and does not require controller reconfiguration or adaptation.

2.3 Active FTCS

AFTCS on the other hand responds to system component failures in an 'active' way by reconfiguration so that stability and acceptable performance of the entire system can be maintained. Therefore, most AFTCS require to provide the fault or failure information before reconfiguration can be undertaken.
In active FTC, FDI plays a vital role in providing information about faults/failures in the system to enable appropriate reconfiguration to take place.
The main function of FDI is to detect a fault or failure and to find its location so that corrective action can be made to eliminate or minimize the effect on the overall system performance. The classification of FDI in model- and non-model-based FDI. Model-based FDI schemes can be grouped into two major categories; FDI using residual schemes and FDI which has the capability to estimate the faults. Fault Identification and Reconstruction (uses kalman filters). Non-model based FDI methods—especially those utilizing artificial intelligence and 'soft computing' approaches such as neural networks, and fuzzy logic.

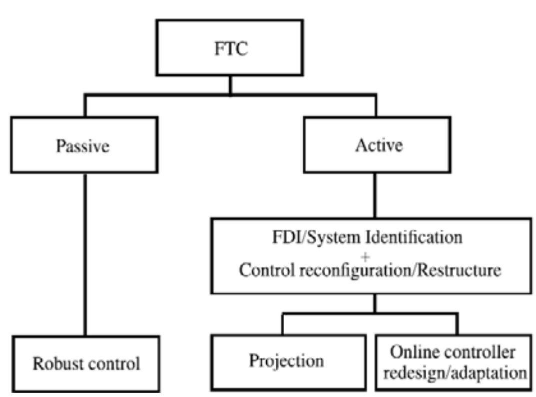

Fig. 1 . Classification of FTCS

2.4 Fault Tree Analysis

Fault Tree Analysis (FTA) is a mathematically simple Boolean tool for modeling a system's unreliability. It is clearly explained in Figure 2. The conventional analysis, proposed by Vesely, determines the probability of failure (*PoF*) of a system from a set of components that make up that system. The components are organized in a tree format with lower cells being individual components and the top level event is system failure. Boolean AND and OR gates are used to connect components with one another.

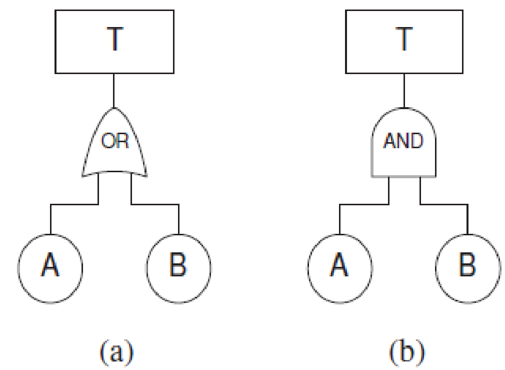

Fig. 2. Fault Tree Analysis

FMEA provides a description of the observable effects of each fault. These effects are consistent, providing all potential effects in qualitative terms. The FMEA report provides,

1) A consistent report that covers all component failures of specific types
2) Coverage provides a basis for generating diagnostics

3 Engine Model In Simulink

Simulink, an add-on product to matlab, provides an interactive, graphical environment for modeling, simulating, and analyzing of dynamic systems. It enables rapid construction of virtual prototypes to explore design concepts at any level of detail with minimal effort. For modeling, Simulink provides a graphical user interface (GUI) for building models as block diagrams. It includes a comprehensive library of pre-defined blocks to be used to construct graphical models of systems using drag-and-drop mouse operations. The user is able to produce an "up-and-running" model that would otherwise require hours to build in the laboratory environment. It supports linear and nonlinear systems, modeled in continuous-time, sampled time, or hybrid of the two. Simulink is integrated with matlab and data can be easily shared between the programs. This model, describes the simulation of a four cylinder spark ignition internal combustion engine. The key elements of the engine model are

- Throttle
- Intake manifold
- Mass flow rate
- Compression stroke
- Torque generation and acceleration

ISSN 2229-5518

3.1 Throttle

The first element of the simulation is the throttle body. Here, the control input is the angle of the throttle plate. The rate at which the model introduces air into the intake manifold can be expressed as the product of two functions—one, an empirical function of the throttle plate angle only; and the other, a function of the atmospheric and manifold pressures. This model accounts for this low pressure behavior with a switching condition in the compressibility equations shown in "(1),"

$$f(\theta) = 2.821 - 0.005231\theta + 0.10299\theta^2 - 0.00063\theta^3 \quad (1)$$

3.2 Intake Manifold

The simulation models the intake manifold as a differential equation for the manifold pressure. The difference in the incoming and outgoing mass flow rates represents the net rate of change of air mass with respect to time. This quantity, according to the ideal gas law, is proportional to the time derivative of the manifold pressure. This model doesn't incorporate exhaust gas recirculation (EGR), although this can easily be added.

3.3 Intake Massflow Rate

The mass flow rate of air that the model pumps into the cylinders from the manifold is described in "(2)," by an empirically derived equation. This mass rate is a function of the manifold pressure and the engine speed.

$$m_{a0} = -0.366 + 0.08979NP_m - 0..0337NP_m^2 + 0.0001N^2 p_m \quad (2)$$

To determine the total air charge pumped into the cylinders, the simulation integrates the mass flow rate from the intake manifold and samples it at the end of each intake stroke event. This determines the total air mass that is present in each cylinder after the intake stroke and before compression.

3.4 Compression Stroke

In an inline four-cylinder four-stroke engine, 180° of crankshaft revolution separate the ignition of each successive cylinder. This results in each cylinder firing on every other crank revolution. In this model, the intake, compression, combustion, and exhaust strokes occur simultaneously (at any given time, one cylinder is in each phase). To account for compression, the combustion of each intake charge is delayed by 180° of crank rotation from the end of the intake stroke.

3.5 Torque Generation and Acceleration

The final element of the simulation describes the torque developed by the engine. An empirical relationship dependent upon the mass of the air charge, the air/fuel mixture ratio, the spark advance, and the engine speed is used for the torque computation.

4 Fault Tolerant Fuel Control System

The model described below represents a fuel control system for a gasoline engine. The system is highly robust in that individual sensor failures are detected and the control system is dynamically reconfigured for uninterrupted operation. The fuel rate control uses signals from the system's sensors to determine the fuel rate which gives a stochiometric mixture. The fuel rate combines with the actual air flow in the engine gas dynamics model to determine the resulting mixture ratio as sensed at the exhaust. The fuel rate control block, shown in Figure 3 uses the sensor input and feedback signals to adjust the fuel rate to give a stoichiometric ratio. The model uses three subsystems to implement this strategy. They are Control logic, Airflow calculation, Fuel calculation. Under normal operation, the model estimates the airflow rate and multiplies the estimate by the reciprocal of the desired ratio to give the fuel rate. Feedback from the oxygen sensor provides a closed-loop adjustment of the rate estimation in order to maintain the ideal mixture ratio.

4.1 Control Logic

A single State flow chart, consisting of a set of six parallel states, implements the control logic in its entirety. The four parallel states correspond to the four individual sensors. The remaining two parallel states at the bottom consider the status of the four sensors simultaneously and determine the overall system operating mode. The model synchronously calls the entire State flow diagram at a regular sample time interval of 0.01 sec. Regardless of which sensor fails, the model always generates the directed event broadcast Fail.INC. The model also uses a corresponding sensor recovery event, Fail.DEC. The Fail state keeps track of the number of failed sensors. The counter increments on each Fail.INC event and decrements on each Fail.DEC event. The bottom parallel state represents the fueling mode of the engine. If a single sensor fails, operation continues but the air/fuel mixture is richer to allow smoother running at the cost of higher emissions. If more than one sensor has failed, the engine shuts down as a safety measure, since the air/fuel ratio cannot be controlled reliably. If a sensor failure occurs during the warm-up period, the Single Failure state is entered after the warm-up time elapses. Otherwise, the Normal state is activated at this time.

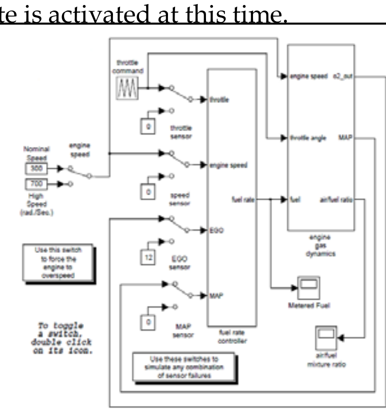

Fig 3 Fuel Rate Controller

ISSN 2229-5518

4.2 Airflow Calculation

The Airflow Calculation block as in figure 4 is the location for the central control laws. This block is found inside the fuel rate control sub system. The block estimates the intake air flow to determine the fuel rate which gives the appropriate air/fuel ratio. Closed-loop control adjusts the estimation according to the residual oxygen feedback in order to maintain the mixture ratio precisely. Even when a sensor failure mandates open-loop operation, the most recent closed-loop adjustment is retained to best meet the control objectives.

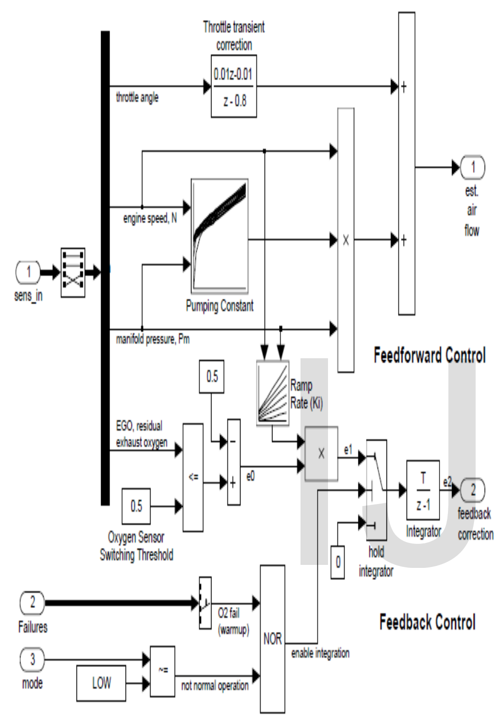

Fig 4 Airflow Calculation

4.3 Fuel Calculation

The fuel calculation subsystem sets the injector signal to match the given airflow calculation and fault status, figure 5. The first input is the computed airflow estimation. This is multiplied with the target fuel/air ratio to get the commanded fuel rate. Normally the target is stoichiometric, i.e. equals the optimal air to fuel ratio of 14.6. When a sensor fault occurs, the State flow control logic sets the mode input to a value of 2 or 3 (RICH or DISABLED) so that the mixture is either slightly rich of stoichiometric or is shut down completely. The fuel calculation subsystem employs adjustable in order to achieve different purposes in different modes. In normal operation, phase lead compensation of the feedback correction signal adds to the closed-loop stability margin. In RICH mode and during EGO sensor failure (open loop), however, the composite fuel signal is low-pass filtered to attenuate noise introduced in the estimation process. The end result is a signal representing the fuel flow rate which, in an actual system, would be translated to injector pulse times.

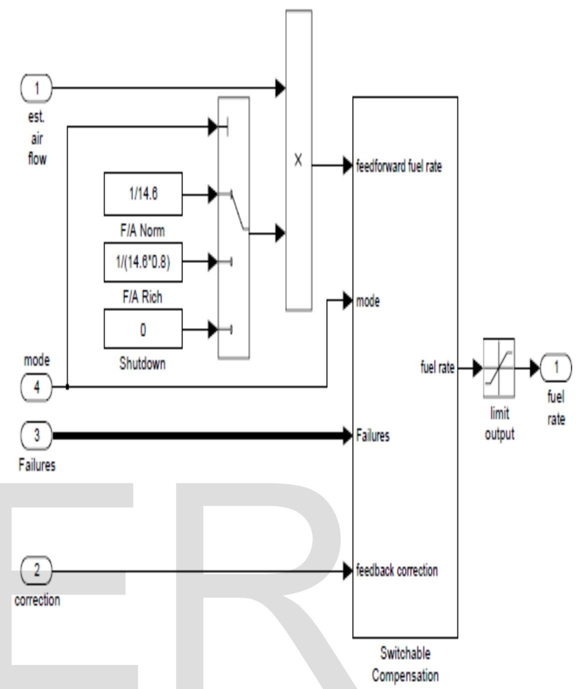

Fig 5 Fuel Calculation Subsystem

5 Results

The results of the simulation are given below in Figures 6, 7 . The simulation is run with a throttle input that ramps from 10 to 20 degrees over a period of two seconds, then back to 10 degrees over the next two seconds. This cycle repeats continuously while the engine is held at a constant speed with different fault conditions and failure modes.. Figure 6 compares the fuel flow rate under fault-free conditions (baseline) with the rate applied in the presence of a single failure in each sensor individually. The control strategy is proven effective in maintaining the correct fuel profile in the single-failure mode. In each of the fault conditions, the fuel rate is essentially 125% of the baseline flow, fulfilling the design objective of 80% rich. Figure 7 plots the corresponding air/fuel ratio for each case.

The baseline plot shows the effects of closed-loop operation. The mixture ratio is regulated very tightly to the stochiometric objective of 14.6. The rich mixture ratio is shown in the bottom four plots of Figure 8. Although they are not tightly regulated, as in the closed-loop case, they approximate the objective of air/fuel = 0.8(14.6) = 11.7. With a constant 12° throttle angle and the system in steady-state, a throttle failure is introduced at t = 2 and corrected at t= 5. At the onset of the failure, the fuel rate increases immediately. The effects are seen at the exhaust as the rich ratio propagates through the system. The steady-state condition is then quickly recovered when closed-loop operation is restored.

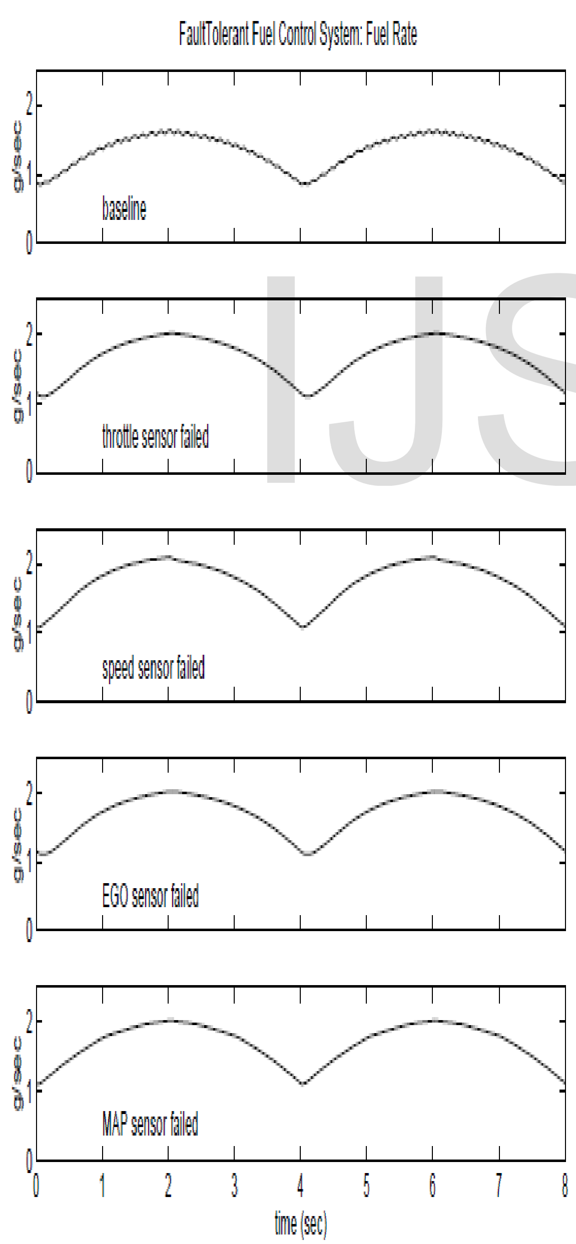

Fig 6 Comparative results for simulated fuel rate

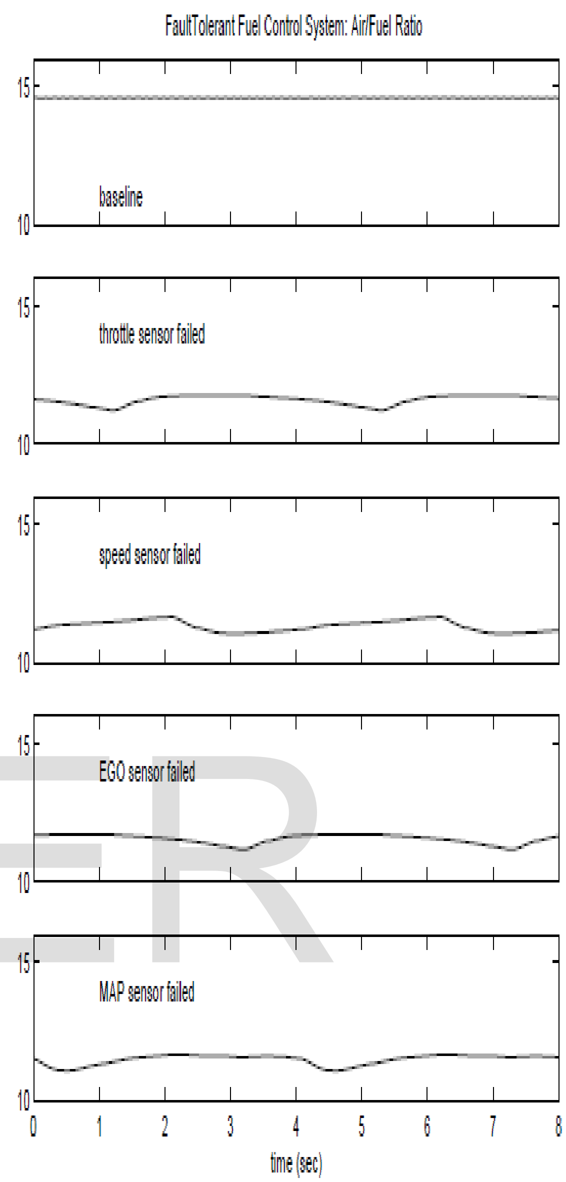

Fig 7 Comparative results for simulated air fuel ratio

6 Conclusion

If a single sensor fails, operation continues but the air/fuel mixture is richer to allow smoother running at the cost of higher emissions. If more than one sensor has failed, the engine shuts down as a safety measure, since the air/fuel ratio cannot be controlled reliably.

REFERENCES

[1] Angeli, C. and Chatzinikolaou, A (2004). On-line faultdetection techniques for technical system: a survey. Int. J. Comput. Sci. Appl, 1(1), pp. 12–30.

[2] De Persis, C. and Isidori, A (2001). A geometric approach to nonlinear fault detection and isolation. IEEE Trans. Autom. Control 46(6), pp. 853–865.

[3] Ding, S. X (2008). Model-based fault diagnosis techniques: design schemes, algorithms, and tools, (Springer-Verlag, Berlin/Heidelberg).

[4] Kalman, R. E. A new approach to linear filtering and prediction problems. J. Basic Engng, 1960,82(1), pp. 35–45

[5] Kim, S., Kim, Y., Park, C. G., and Jung, I (2006).Hybrid fault detection and isolation techniques for aircraft inertial measurement sensors. Int. J. Aeronaut. Space Sci., 7(1), pp. 73–83.

[6] Luenberger, D. G.(1996). Observers for multivariable systems. IEEE Trans. Autom. Control, 11(2), pp. 190–197.

[7] Napolitano, M. R., An, Y., and Seanor, B. A(2000). A fault tolerant flight control system for sensor and actuator failures using neural networks. Aircraft Des., 3(2), pp. 103–128.

[8] Sikandar samar (2006) .Embedded Estimation of Fault Parameters in an Unmanned Aerial Vehicle IEEE Conference on Control Applications, Munich, Germany.

[9] Youmin Z. (2001)'Integrated Active Fault Tolerant Control Using IMM approach', IEEE Transactions on Aerospace Electronic Systems, Vol. 37, No. 4, pp. 1221-1235

[10] Youmin Z. (2013), A Tutorial for 2013 International Conference on Unmanned Aircraft Systems (ICUAS'13)

[11] Youmin Z. and Jiang J. (2003), 'Bibliographical Review on Reconfigurable Fault-Tolerant Control Systems', Proc. Of the 5th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes, USA, pp. 265-276

[12] Zhang, N., Fernandez, J. L. L., Doncescu, A., and Mora-7.Camino (2009), F. Rotorcraft fault detection using difference flatness. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Chicago, USA, pp. 10–13.