Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Control Engineering
Prague, Czech Republic

Luleå University of Technology
Department of Space Science
and Electrical Engineering
Kiruna, Sweden

May 2008

MASTER'S THESIS

Fault Detection on an Aircraft and Development of a Contingency Control Strategy

Martín Oscar Giacomelli

Master Diploma Thesis for
Erasmus Mundus programme SpaceMaster

Supervisor:
Doctor Martin Hromcík
Czech Technical University
Prague, Czech Republic

Co-supervisor:
Doctor Andreas Johansson
Luleå University of Technology
Luleå, Sweden

Abstract

In this work the possibility of developing a contingency control strategy for certain faults in a UAV without using additional hardware is studied. First, controllers for achieving basic flying regimens are determined, that combined with a nonlinear model of the UAV establish a realistic simulation platform. Then, the problem of detection and identification of faults in the ailerons and rudder of the aircraft is tackled. The unknown input observer is used in a proposed variation of the dedicated observer. Finally a contingency system that reconfigures the controller in the case of a fault is developed. Under computer simulation, the system proved to successfully detect and identify usual faults in aircraft actuators under nominal flight conditions. Moreover, the reconfiguration of the controllers allows the UAV to perform basic maneuvers like regaining a straight level flight and perform a turn even in the presence of severe faults.

Declaration of Independent Work

I, Martín Oscar Giacomelli, hereby declare that I have completed this master thesis independently and that I have listed all the literature and publications used.

I have no objection to usage of this work in compliance with the act 60 Zákona č. 121/2000Sb. (copyright law), and with the rights connected with the copyright act including the changes in the act.

27 June 2008

Table of contents

1. Introduction

During the last years there has been a growing interest in the development of unpiloted aircrafts commonly referred to as unmanned aerial vehicles (UAV). Currently, UAVs are mainly being used in a number of military roles, including reconnaissance and attack e.g. the MQ-1 Predator system of the United States[1]. However, there is also a growing interest in civil applications. UAVs present several advantages when compared with conventional manned aircrafts, but the most important are the absence of risk to human beings and the significantly lower price. Some examples of civilian applications are:

- *Scientific Research in dangerous areas*. For example the National Oceanic and Atmospheric Administration (NOAA) used UAVs which can fly into a hurricane and communicate near-real-time data to a base.
- *Transport* of goods to zones of difficult access (e.g. after natural disasters)
- *Remote Sensing*. UAVs equipped with electromagnetic sensors that typically include visual spectrum, infrared, or near infrared cameras as well as radar systems are used in commercial applications.
- *Surveillance operations* including inspecting and monitoring river boundaries, bridges and coastlines.

At the present time the Air Force of the Armed Forces of the Czech Republic is carrying out research in a UAV though the VTÚLaPVO institute. The Department of Control Engineering at the Czech Technical University (CTU or CVUT) is also involved in the project where several researches are being carried out in this system.

The name of the project is "Sojka" and is qualified as a tactical reconnaissance UAV system. According to the mission it is possible to equip the system with reconnaissance sensors for optical surveillance of terrain, objects and military vehicle, monitoring artillery fire, border control, natural disasters consequences (floods, fires, etc.), contaminated areas or as an aerial target for gunnery practice of shooting[2].

The nature of these missions implies that Sojka is going to operate in a much riskier environment than the one in which common aircraft do. This fact leads the Department of Control Engineering at the Czech Technical University (CTU or CVUT) to consider the possibility of implementing a system for online fault detection and identification on Sojka. The system of course should be completely autonomous, and as weight and space is of major concern on a UAV, no additional hardware or redundancy should be used. Once that a fault has been detected a contingency action should be taken. So it was desired to also study the possibility of developing an automatic contingency control strategy (without using additional hardware of redundancy) to be applied in the case of a fault.

There are not many statistics available of UAV common faults. However, there is basic information about incidents in commercial airplanes. The Transportation Safety Board of Canada Information Strategies and Analysis Directorate provide free official statistics about incident involving Canadian aircrafts[3]. Analyzing that data it can be seen that from the incidents related to hardware problems, engine failure, hydraulic failure and electrical failure are the most relevant. Concerning actuators, hydraulic and electrical faults can be treated as power problems.

For this work only the faults that are susceptible of being overcome without additional hardware or redundancy should be considered. In this way, faults in the engine are going to be excluded from the analysis. Power problems related with the ailerons and rudder are going to be studied. Sensor failures could also be addressed but are not going to be tackled in this work.

2. Thesis Definitions

2.1. Goal and Objectives

2.1.1. Goal

The goal of this thesis is to study the possibility of determining a contingency control strategy to be applied in the case that a fault occurred in an actuator of an unmanned aerial vehicle.

Due to the broad extent of the task, only faults in the aileron and rudder actuators will be tackled. The main motivation for centering the study on these actuators is the fact that their dynamics are highly coupled, making their effects difficult to differentiate for a fault detection system.

2.1.2. Objectives

1. Develop the controllers needed to make the UAV reach a nominal regimen and perform the basic maneuvers in which the fault detection system can be tested. The basic flight scenarios required are:
    a. Steady level flight.
    b. Coordinated turn.
2. Develop a fault detection system that successfully detects and identifies determinate faults in certain actuators of the UAV without using redundancy or additional hardware. The actuator of interest are:
    a. Ailerons.
    b. Rudder.

    The faults that must be detected are the ones that present a risk to the UAV mission and that are susceptible of being overcome with no additional hardware. The faults of interest are:
    a. Loss of power supply to the actuator.
    b. Actuator stuck at a fixed deviation angle during a maneuver.
3. Determine a contingency strategy to be applied when the fault scenario is detected. This strategy must include the development of an alternative controller that allows the UAV to perform basic maneuvers even at reduced control conditions. No additional hardware or redundancy is available.

2.2. Plant Description

2.2.1. Sojka: Unmanned Aerial Vehicle (UAV)

The plant in which this thesis work is based is the unpiloted aircraft developed by the VTÚLaPVO branch of the Air Force of the Armed Forces of the Czech Republic, called Sojka. The basic technical characteristics of Sojka are:

- Speed: max. 210 km/h, min: 120 km/h
- Endurance: 4 hr
- Ceiling: 4000 m
- Radius of operation: 200 km
- Navigation: Inertial with GPS correction

Dimensions and weights:

- Wing span: 4,5 m
- Overall length: 3,78 m
- GTOW: 145 kg
- Payload: 20 kg

Even though the UAV has significant differences with conventional airplanes, the basic physic equation that describes the dynamic of airplanes can be used to determine the model of this type of aircrafts. This fact guaranties a fairly wide amount of literature about the subject. In the following subsections a brief description of the airplane model is given.

2.2.2. General Aircraft Model

Considering the flat earth equations the conventional 6 degrees of freedom (6-DOF) aircraft model can be derived. The aircraft is supposed to be a rigid body moving with respect to an inertial frame. The orientation of the body coordinate axes is fixed in the shape of the airplane's body (see Figure 2-1).

- The x-axis points through the nose of the craft.
- The y-axis points to the right of the x-axis (facing in the pilot's direction of view), perpendicular to the x-axis.
- The z-axis points down through the bottom the craft, perpendicular to the xy plane and satisfying the right hand rule.

The variables that comprise the state vector of the model in this coordinate system are:

- The components of the velocity vector with respect to the wind: U, V and W in the x, y and z direction respectively.
- The Euler angles of rotations: ϕ (roll about the x-axis), θ (pitch about the y-axis) and ψ (yaw about the z-axis). A representation can be seen in Figure 2-1.
- The components of the angular velocity vector: P (about the x-axis), Q (about the y-axis) and R (about the z-axis).

- The position: p_N, p_E and h are, respectively, the north, east and vertical components of the aircraft position.

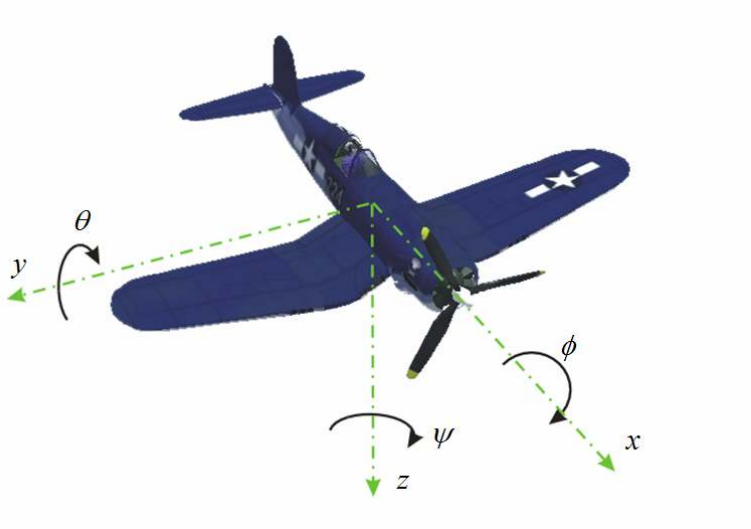

Figure 2-1: Body coordinate axis and Euler angles

The equations that determine the relation between the state variables for the 6-DOF model are:

Force equations:

$$\begin{aligned}\dot{U} &= RV - QW - g_0' \sin\theta + \frac{F_x}{m} \\ \dot{V} &= -RU - PW + g_0' \sin\theta\cos\theta + \frac{F_y}{m} \\ \dot{W} &= QU - PV + g_0' \cos\theta\cos\theta + \frac{F_z}{m}\end{aligned} \tag{2.1}$$

Kinematics Equations

$$\begin{aligned}\dot{\phi} &= P + \tan\theta\left(Q\sin\phi + R\cos\phi\right) \\ \dot{\theta} &= Q\cos\phi - R\sin\phi \\ \dot{\psi} &= \frac{Q\sin\phi + R\cos\phi}{\cos\theta}\end{aligned} \tag{2.2}$$

Moment Equations

$$\begin{aligned}\dot{P} &= \left(c_1 R + c_2 P\right)Q + c_3\bar{L} + c_4 N \\ \dot{Q} &= c_5 PR - c_6\left(P^2 - R^2\right) + c_7 M \\ \dot{R} &= \left(c_8 P - c_2 R\right)Q + c_4\bar{L} + c_9 N\end{aligned} \tag{2.3}$$

Navigation Equations

$$\begin{aligned}
\dot{p}_N &= U\cos\theta\cos\psi + V\left(-\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi\right) \\
&\quad + W\left(\sin\phi\sin\psi + \cos\psi\cos\phi\sin\theta\right) \\
\dot{p}_E &= U\cos\theta\sin\psi + V\left(\cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi\right) \\
&\quad + W\left(-\sin\phi\cos\psi + \sin\psi\cos\phi\sin\theta\right) \\
\dot{h} &= U\sin\theta - V\sin\phi\cos\theta - W\cos\phi\cos\theta
\end{aligned} \tag{2.4}$$

where N, M and $\bar{L}$ are the torque components acting on the center of gravity of the aircraft, in x, y and z respectively and F_x, F_y and F_z are the force components. The constants c_i are functions of the moment of inertia of the aircraft. $g_0' = 9.805 m/s^2$ is the magnitude of the gravity acceleration at the sea level and 45° latitude.

The force and torque components as well as the parameters c_i depend on the control inputs to the system. The inputs to the system are the throttle, and the deviation angles of the rudder, elevator and aileron. So the input vector is:

$$U_{input} = \begin{bmatrix} thl & rdr & el & ail \end{bmatrix}^T \tag{2.5}$$

As it can be seen from these equations, the model of the aircraft is nonlinear. It is common to decompose the model into two decoupled set of equations. One set that describes the longitudinal motion (pitch and transformation in the $x-z$ plane), and another set that describes the lateral-directional motion (rolling and sideslipping and yawing). The handling of the equations and the simulations are made easier by this decoupling, nevertheless it requires certain simplifications that reduce the accuracy of the model. The model that is used for simulation purposes in thesis work is not based on decoupled equations and uses the complete set of nonlinear equations, so the results obtained should represent the reality with a significant accuracy. This model is available as a Simulink file, so the results obtained throughout this thesis will be tested in this simulation environment. The exact model used for simulations is not presented here due to legal conditions.

2.2.3. Sensors

In order to determine its position, orientation and velocity, the UAV possesses a set of inertial measurement devices. Gyroscopes are used to measure the angular velocity of the system in the inertial reference frame. Linear accelerometers measure how the vehicle is moving in space; there is a linear accelerometer for each axis. With the information provided by these sensors, a computer can integrate and calculate the angular and linear velocities. Another sensor incorporated in the UAV is the altimeter, which measures the atmospheric pressure from a static port outside the aircraft. Even though it is not highly accurate, the error is acceptable. Finally the sideslip angle is available as an output of the system. This angle is not directly measured but derived by a calculation from the other measurements.

These sensors, thus, provide measurements of: angular velocities, attitude (roll, pitch and yaw), height and sideslip angle, and represent the output of the UAV model (their transfer functions are considered to be 1).

2.2.4. Actuators

As stated before the input to the system are throttle, and the deviation angles of the rudder, elevator and aileron. In Figure 2-2 the ailerons and elevators of an aircraft are shown and in Figure 2-3 the rudder is indicated. The throttle is, of course, provided by the engine and the

aerodynamics. The model of the engine of an aircraft is of considerable complexity and the throttle is not relevant to the objectives of this work, so this input will not be used for control purposes, and the throttle will be supposed constant.

The dynamic of the hydraulic actuators used to deflect the ailerons and rudder present nonlinear behaviors and dead zones. Nevertheless, for designing controllers, it is common to approximate them as[4]:

$$G_{rudder}(s) = \frac{20.2}{s + 20.2} \tag{2.6}$$

$$G_{aileron}(s) = \frac{20.2}{s + 20.2} \tag{2.7}$$

$$G_{elevator}(s) = \frac{10}{s + 10} \tag{2.8}$$

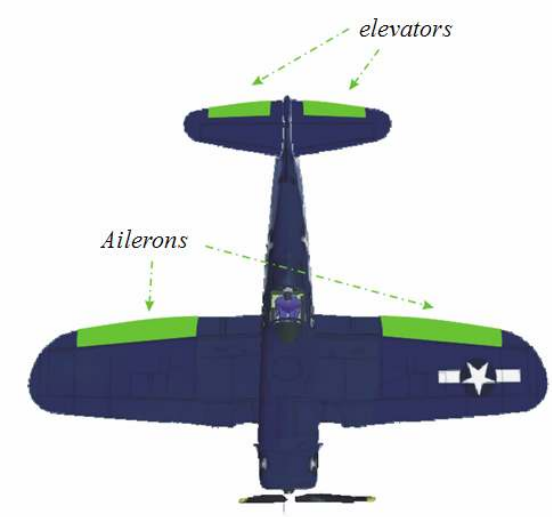

Figure 2-2: Ailerons and elevators of an aircraft

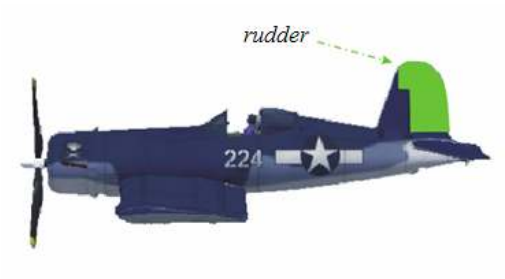

Figure 2-3 Rudder of an aircraft

3. UAV Controllers for Basic Maneuvers

In this section the objective 1 is tackled. It is desired to develop controllers that will successfully maintain the UAV in a straight level flight and perform a coordinated turn. The motivation to do it is to confer the UAV a realistic behavior so as to test the fault detection system in a simulated plant that recreates the actual situation with certain accuracy. Even though there are many controllers that a modern aircraft uses, here only the two necessary ones are going to be developed. They are the coordinated turn system and the altitude hold system.

3.1. UAV Lateral Control System: Coordinated Turn

When a fixed-wing aircraft is making a turn (changing its direction) the aircraft must roll to a banked position so that its wings are partly angled towards the desired direction of the turn. When the turn has been completed the aircraft must roll back to the wings-level position in order to resume straight flight.

In straight level flight, the lift acting on the aircraft acts vertically upwards to counteract the weight of the aircraft which acts downwards. During a balanced turn where the angle of bank is ϕ the lift acts at an angle ϕ away from the vertical. The lift vector can be decomposed into a vertical component and a horizontal component. The horizontal component is the centripetal force causing the aircraft to turn.

During the coordinated turn the rudder is used to maintain the nose of the aircraft point along the flight path, that is, to keep the sideslip angle β at zero degrees. If the rudder is not used, an adverse yaw could be encountered in which the drag on the outer wing pulls the aircraft nose away from the flight path.

3.1.1. Problem Formulation

The objective of a lateral control system is to provide coordinated turns by causing the bank angle ϕ to follow a desired command while maintaining the sideslip angle β at zero degrees. However due to stability issues it must also be guaranteed that the angular velocities about the x and z axis do not growing too much. Thus, the problem is a step command tracking exercise for the angles ϕ and β and a regulator for R and P. The problem scheme is shown on Figure 3-1, where δ_{rdr} and δ_{ail} are the angles of the rudder and the ailerons respectively. The plant represents the dynamic of the UAV from those inputs to the outputs shown.

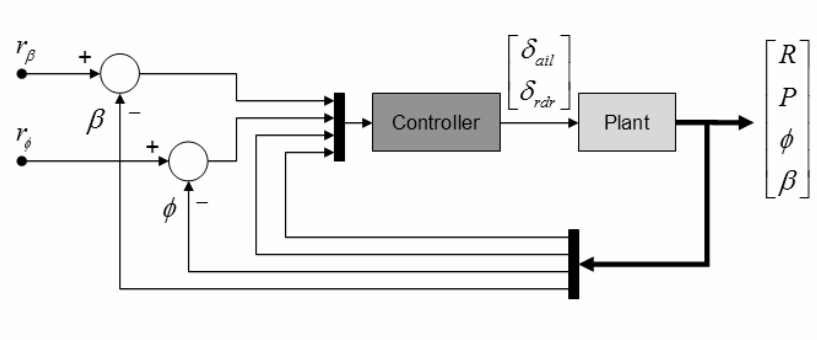

Figure 3-1: Coordinated turn system

3.1.2. Background Theory

Several strategies are proposed for solving this problem. In [4] the static output feedback matrix approach is proposed. The main advantage of the static output feedback is the ability it provides for designing controllers of a desired structure. In this way, engineers can take advantage of their knowledge about airplane controllers. For a system of the form

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{3.1}$$

The control law for the static output feedback controller becomes:

$$u = -Ky \tag{3.2}$$

In the case of a regulator, the close loop system equations are found to be

$$\dot{x} = \left(A - BKC\right)x \tag{3.3}$$

Necessary and sufficient conditions for solving this problem have been and presented in [5]. Several approaches have been applied to solve this problem, including the use of LMI techniques. Nevertheless, the solution in the general case is not trivial, see for example [6]. Another approach is to use direct search methods like the ones that have been implemented in the MATLAB toolbox "Matrix Computation Toolbox"[7].

Another control strategy that is commonly applied is the LQG controller, that is, the combination of a Kalman observer with a Linear-Quadratic Regulator (LQR) as shown in Figure 3-2.

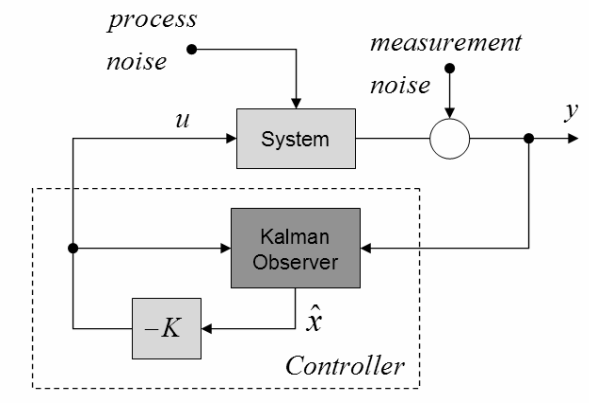

Figure 3-2: LQG controller

This technique is widely know and documented in any bibliography of control systems, so it is considered that the reader of this work is familiarized with it. A good description of the linear quadratic regulator can be found in [8] for continuous time and in [9] for the description in discrete time, where also the Kalman observer is presented. Following, for presenting the notation, a basic description for discrete time is given.

For a discrete time linear system of the form

$$x_{k+1} = Ax_k + Bu_k \tag{3.4}$$

given the control law

$$u_k = -Kx_k \tag{3.5}$$

the optimal state feedback matrix K that minimizes the cost function

$$J = \sum_{k=0}^{\infty} \left(x_k^T Q_{lq} x_k + u_k^T R_{lq} u_k \right) \tag{3.6}$$

is given by

$$K = \left(R_{lq} + B^T PB \right)^{-1} B^T PA \tag{3.7}$$

where P is the solution to the discrete time algebraic Riccati equation

$$P = Q + A^T \left(P - PB\left(R + B^T PB \right)^{-1} B^T P \right) \tag{3.8}$$

As most of the times the complete state vector x_k is not available, the separation theorem guarantees that a state observer can be used to estimate it under the condition of observability. The equations of the Kalman observer are presented in 4.2.3.1.

The LQG theory is going to be used for deriving the coordinated turn controller.

3.1.3. Augmented System for Step Command Reference Tracking

As it can be seen in Figure 3-2 so far only the regulator problem has been considered, however, in the coordinated turn it is desired to perform a tracking with a zero steady state error to a step command. The tracking error can be defined as $e = [e_\beta \quad e_\phi]^T$ with

$$\begin{aligned} e_\beta &= r_\beta - \beta \\ e_\phi &= r_\phi - \phi \end{aligned} \tag{3.9}$$

In order to achieve a zero steady state error to a step command input, integrators should be included, so an augmented system should be derived. This can be done as shown in Figure 3-3.

Figure 3-3: Augmented system for the turn coordination system

Another issue that should be considered is the dynamic of the actuators. Considering a plant of the form

$$\begin{aligned} \dot{x}_p &= A_p x_p + B_p u_p \\ y &= C_p x_p \end{aligned} \tag{3.10}$$

Here, the state x, the input u, and the output y are functions of time with values in $\Re^n$, $\Re^m$ and $\Re^o$ respectively. A_p, B_p, C_p are real matrices of size $n \times n$, $n \times m, o \times m$ respectively.

The actuators dynamic as:

$$\begin{aligned} \dot{w} &= A_{act} w + B_{act} u \\ y_{act} &= C_{act} w \end{aligned} \tag{3.11}$$

And the integrated tracking errors states:

$$\begin{bmatrix} \dot{\varepsilon}_\phi \\ \dot{\varepsilon}_\beta \end{bmatrix} = \begin{bmatrix} r_\phi - \phi \\ r_\beta - \beta \end{bmatrix} \tag{3.12}$$

The augmented system is:

$$\begin{aligned} \begin{bmatrix} \dot{x}_p \\ \dot{w} \\ \dot{\varepsilon}_\phi \\ \dot{\varepsilon}_\beta \end{bmatrix} &= \begin{bmatrix} A_p & B_p C_{act} & 0 & 0 \\ 0 & A_{act} & 0 & 0 \\ -C_p(i_\phi,:) & 0 & 0 & 0 \\ -C_p(i_\beta,:) & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_p \\ w \\ \varepsilon_\phi \\ \varepsilon_\beta \end{bmatrix} + \begin{bmatrix} 0 \\ B_{act} \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} \delta_{rdr} \\ \delta_{ail} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} r_\phi \\ r_\beta \end{bmatrix} \\ y_{aug} &= \begin{bmatrix} C_p(i,:) & 0 & 0 & 0 \\ -C_p(i_\phi,:) & 0 & 0 & 0 \\ -C_p(i_\beta,:) & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_p \\ w \\ \varepsilon_\phi \\ \varepsilon_\beta \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} r_\phi \\ r_\beta \end{bmatrix} \end{aligned} \tag{3.13}$$

Where $C_p(i_\phi,:)$ represents the row vector of the matrix C_p corresponding to ϕ and $C_p(i,:)$ for $1 \le i \le o$ and $i \ne i_\phi, i_\beta$ is the ith row of C_p, with o being the number of outputs of the plant.

Once that the augmented system is determined the equation (3.7) can be used to find the optimum feedback matrix for this augmented system. To be consistent with Figure 3-3 the matrix K from equation (3.7) has the form

$$K = \begin{bmatrix} K_n & K_{\varepsilon\phi} & K_{\varepsilon\beta} \end{bmatrix} \tag{3.14}$$

Even though it is correct to consider the dynamic of the actuators, it can be seen that their dynamic is considerably fast (poles at -20.2 and -10) so the transfer functions of the actuators can be approximated to 1 without loosing much accuracy.

3.1.4. Controller Development and Simulation

In order to derive the controller, the first step is to linearize the UAV model at a proper equilibrium point.

3.1.4.1. Linearization

According to the dynamic equations there are two equilibrium points that can be used as linearization points for the determination of this controller:

- *Steady turning flight*, defined for the following conditions

$$\begin{aligned} &\dot{P},\dot{Q},\dot{R},\dot{U},\dot{V},\dot{W} = 0 \\ &\dot{\theta},\dot{\phi} = 0 \\ &\dot{\psi} = turn\,rate \end{aligned} \tag{3.15}$$

- *Steady roll*, defined for the following conditions

$$\begin{aligned} &\dot{P},\dot{Q},\dot{R},\dot{U},\dot{V},\dot{W} = 0 \\ &\dot{\theta},\dot{\psi} = 0 \\ &\dot{\phi} = roll\,rate \end{aligned} \tag{3.16}$$

As in the coordinated bank turn the rolling movement of the aircraft is more important than the yaw movement, the steady roll equilibrium point will be selected for linearization.

As the nonlinear model of the UAV is available in Simulink, the linearization model is derived using the MATLAB linearization function "linmod". Due to legal conditions the model obtained will note be presented in this thesis. The deviation of each variable form the linearization point is going to be noted as the lower case of the variable, e.g. p is the deviation of the variable P from its linearization point.

3.1.4.2. Controller Determination

First the dynamic of the UAV from the aileron (δ_a) and rudder (δ_r) to the output vector $y = \begin{bmatrix} p & r & \phi & \beta \end{bmatrix}^T$ is determined, this can be easily achieved using MATLAB.

Once that the augmented system from equations (3.13) is derived, the LQG algorithms can be used to obtain the controller using equations (3.7) and (3.8). The equations of the Kalman observer are given in 4.2.3.1. MATLAB provides functions that calculate the Kalman observer steady state gain and the LQR optimum state feedback matrix. Using this application and iteratively tuning the weighting matrices R_{lq} and Q_{lq} of the LQR, a desired time domain performance can be achieved. A satisfactory result was achieved for the discretized system with $T_s = 0.1s$ using the state feedback matrix

$$K = \begin{bmatrix} -0.56 & 0.05 & 0.72 & 0.48 & 0.05 & -2.42 \\ -0.05 & -0.35 & 0.15 & 0.11 & -0.26 & -0.36 \end{bmatrix} \tag{3.17}$$

3.1.4.3. Simulation

A normal bank angle is usually kept below 30°, and for commercial airlines 15-20 degrees of bank is all it takes to accomplish a standard rate turn and follow a traffic pattern. In Figure 3-4, the step response for a target bank angle of 25° is shown. It is important to state that the simulations are always performed using the nonlinear plant. It can be seen that the command bank angle is correctly tracked and that the sideslip angle is kept very close to zero. Also the actuators deviation angles are shown in Figure 3-4; it can be seen that these angles are significantly small.

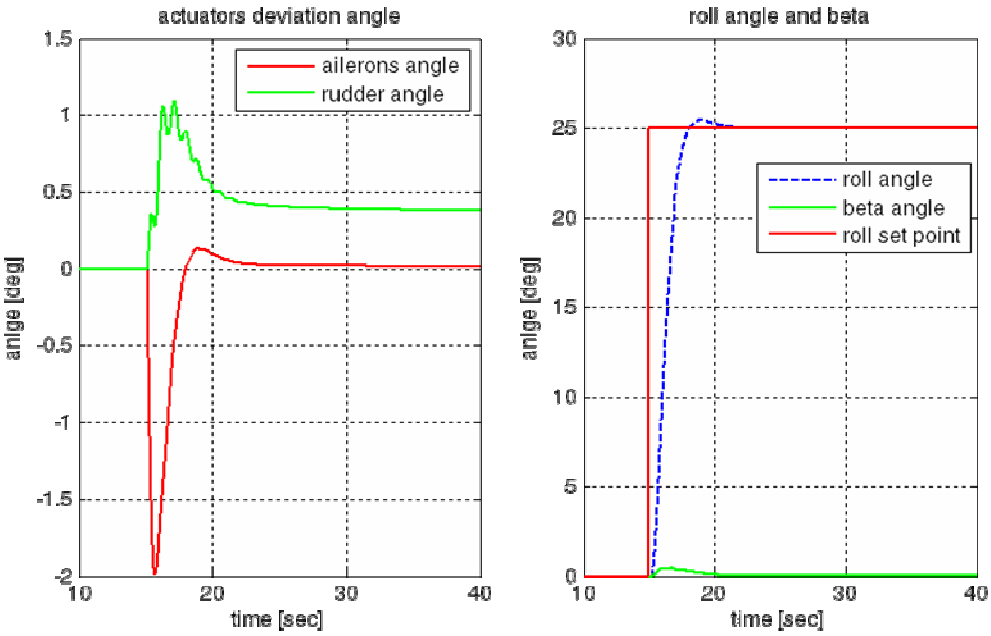

Figure 3-4: Step response for a target bank angle of 25°

3.2. Altitude Hold

The goal of the altitude hold system is to maintain the aircraft at a fixed altitude. Another objective of this system is to reject disturbances like air flows that could make the aircraft loose its desired height. Another important task of the altitude hold system is to provide additional lift when the aircraft is turning. As explained in 3.1, the vertical component of the lift vector is reduced during the turn and the horizontal is increased. However, the vertical component must continue to equal the weight of the aircraft, otherwise it would loose height[4]. The altitude hold system commands the elevators to increase the angle of attack and provide additional lift thus maintaining the altitude.

3.2.1. Problem Formulation

The altitude hold system must be able to maintain the UAV in a desired reference altitude with zero steady state error. However this must be done keeping the pitch angle θ and the angular velocity about the y axis q close to zero. The last requirements are imposed in order to provide certain stability properties to the aircraft. A typical altitude hold system is shown in Figure 3-5, where the actuator transference function is included. In this figure the Plant represents the dynamic of the UAV from the elevator input to the outputs shown.

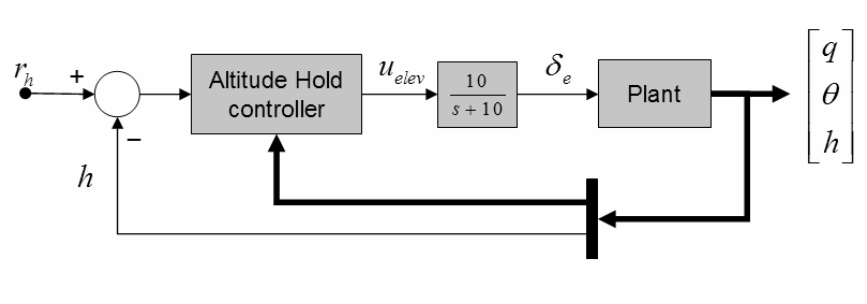

Figure 3-5: Altitude hold system

3.2.2. Controller Development and Simulation

The controller used for the altitude hold system is the LQG described in the previous section. As it is required to follow a step command with zero steady state error, an integrator has to be included in the controller.

3.2.2.1. Linearization

The equilibrium point at which the linearization is performed is the steady pull up flight, which is defined as the situation when the following equations are verified:

$$\begin{aligned} &\dot{P},\dot{Q},\dot{R},\dot{U},\dot{V},\dot{W} = 0 \\ &\phi,\dot{\phi},\dot{\psi} = 0 \\ &\dot{\theta} = pull\,up\,rate \end{aligned} \tag{3.18}$$

This working point was selected because at this regimen the modes relevant to the altitude hold system are properly excited.

3.2.2.2. Controller Determination

The first step is to determine the augmented system in order to follow a step command without error. This is achieved following the same procedure as in 3.1.3, but this time only one reference is required: the height set point. The configuration required is the one shown in Figure 3-6.

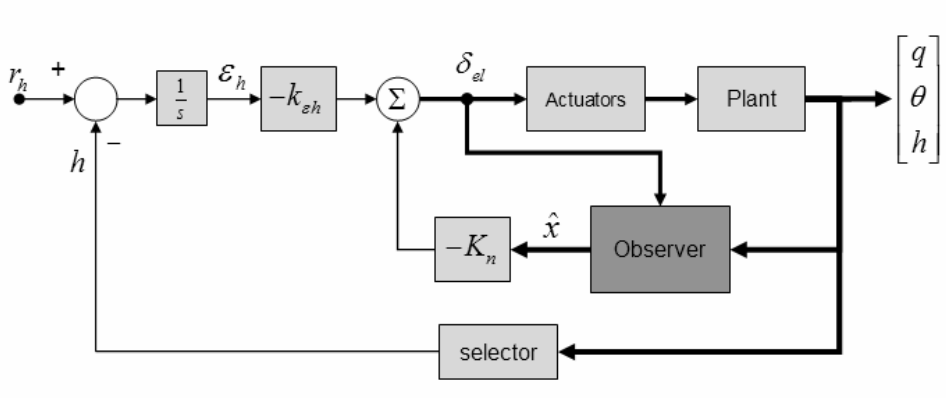

Figure 3-6: Augmented system for the altitude hold system

Once that the augmented system is derived it is discretized with $T_s = 0.1\text{s}$. Then the Kalman observer and the LQR feedback matrix $K = \begin{bmatrix} K_n & k_{\varepsilon h} \end{bmatrix}$ can be obtained using

MATLAB. The desired step response is achieved by tuning the weighting matrices Q_{lq} and R_{lq} (see equation (3.6)). A satisfactory response was obtained with

$$K = \begin{bmatrix} 0.68 & 0.43 & -8e-4 & 0.03 & -1.34 & 0.02 \end{bmatrix} \tag{3.19}$$

3.2.2.3. Simulation

The step response of the altitude hold system is shown in Figure 3-4. It can be seen that the set point is successfully reached after about 10sec.

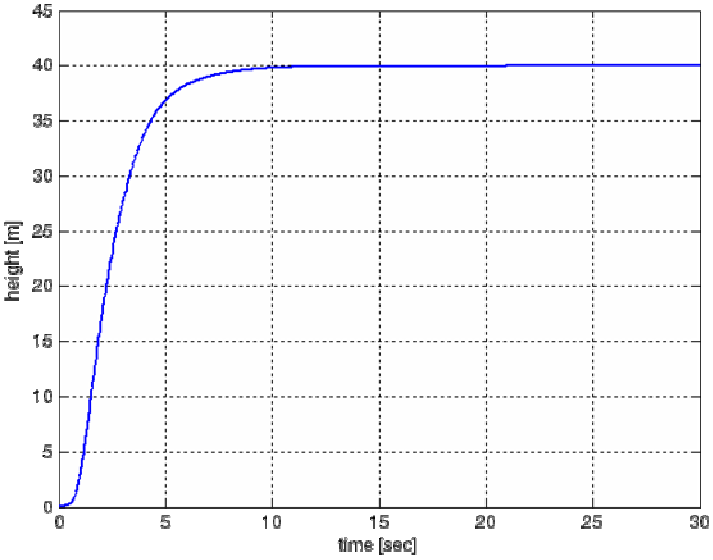

Figure 3-7: Step response of the altitude hold system

4. Model Based Fault Detection

In this section the objective 2 is tackled, that is, the detection of certain faults. A fault can be defined as an unexpected deviation of at least one characteristic property or parameter of the system from the acceptable, usual or standard condition[11]. Three types of faults can be encountered in a system given by the three parts in which a system can be split[10]:

- *Actuators faults*, which can be viewed as any malfunction of the equipment that actuates the system, e.g. a malfunction in a solenoid valve.
- *system dynamics faults* (or component faults), which occur when some changes in the system make the dynamic relation invalid, e.g. leak in a tank in a two-tank system.
- *Sensors faults*, which can be viewed as serious measurements variations.

At present time, one of the most widely used techniques in fault detection is the model based approach. This technique makes use of the a priori knowledge available about the model of the system, which is usually developed based on some fundamental understanding of the physics of the process. There are basically two model based categories of fault detection methods that use this information. The first one is the quantitative, in which this information is expressed in terms of mathematical functional relationships between the inputs and outputs of the system in the form of system descriptions (e.g. difference or differential equations, state-space models, transfer functions, neural networks, etc.) The main advantage that this approach presents is that it makes use of the results from widely-understood control theory, i.e. state observers or filters, parameter estimation, parity relation concepts, etc.[10]. The second model based category is the qualitative. In this approach, relationships are expressed in terms of qualitative functions between different parts of the system. This technique usually makes use of the knowledge from experts of the system in both the fault free case and the faulty case. In this way, qualitative models are used to estimate the system's behavior under the normal and faulty operating conditions.

The model based techniques require two basic stages as shown in Figure 4-1. The first one is to generate indicators of inconsistencies between the actual and expected behavior of the system. Such indicators are called residuals. The residuals should be close to zero when no fault occurs but show 'significant' values when the underlying system changes. Another characteristic that the residuals should have is orthogonality, so that each fault presents an impact on only one residuum, while the others remain unchanged.

The second stage comprises a decision making, where the residuals are analyzed by means of a decision rule or algorithm and it is decided if a fault has occurred. Many techniques have been applied to the decision making, some of them are the simple norm measurements, Bayesian tests, neural networks, etc.

In order to generate the residuals some form of redundancy should be used. There are two types of redundancies, hardware redundancy and analytical redundancy. The traditional

approach to fault detection was based on hardware or physical redundancy methods which use multiple sensors, actuators, components to measure and control a particular variable[11]. The major problems encountered with hardware redundancy are the extra equipment and maintenance cost, as well as the additional space required to accommodate the equipment[12]. These aspects are of major concern in the case of a UAV, where the space and cost must be maximized, so no hardware redundancy will be used in this work. On the other hand, analytical redundancy relies on a priori knowledge about the system, where inherent redundancy contained in the static and dynamic relationship among the system input and measured outputs is exploited.

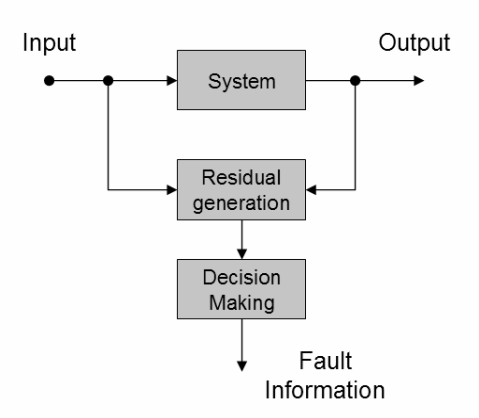

Figure 4-1: Fault detection basic stages

Some of the most popular analytical redundancy residual generation techniques are[10]:

- Parameter estimation
- parity relation
- observer-based

In the section 4.2 a basic review of these techniques is presented. First, a description of a system that faces faults is stated.

4.1. Fault Detection Problem Formulation

Throughout the literature the most commonly used description of a fault in a system is to assume linearity and include an additive term that corresponds to the fault. This can be represented as shown in Figure 4-2.

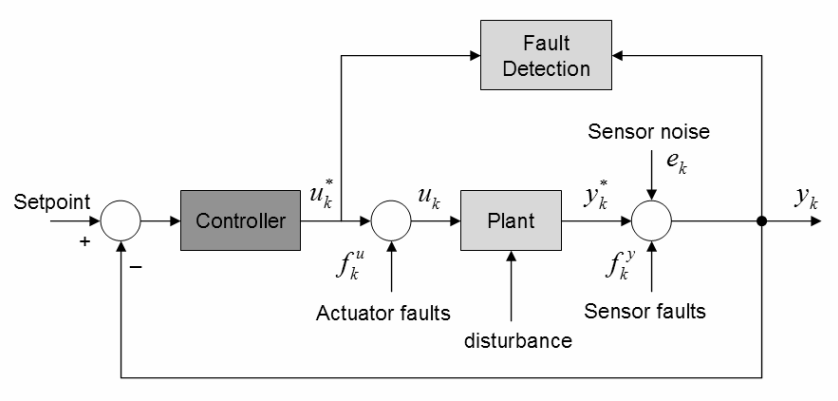

Figure 4-2: Problem formulation scheme

The normal behavior of the dynamic system can be the following discrete time linear system:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + Ed_k \\ y_k^* &= Cx_k + Du_k \end{aligned} \tag{4.1}$$

Where $u_k \in \Re^l$ is the process input; $y_k^* \in \Re^m$ is the fault free process output; $x_k \in \Re^n$ is the process state vector; $d_k \in \Re^q$ represents the unmeasured deterministic process; $E \in \Re^{n \times q}$ is a gain matrix of disturbances. A, B, C are the process matrices with appropriate dimensions.

The presence of a fault in the sensors and actuators and measurement noise can be represented by

$$\begin{aligned} u_k &= u_k^* + f_k^u \\ y_k &= y_k^* + f_k^y + e_k \end{aligned} \tag{4.2}$$

where $y_k \in \Re^m$ is the measured output vector; $u_k^* \in \Re^l$ is the fault free input vector process. $e_k \in \Re^m$ is the output measurement noise. $f_k^u \in \Re^l$ is the actuator fault, and $f_k^y \in \Re^m$ is the sensor fault. e_k is the measurement noise that is assumed to be white with Gaussian distribution.

In the case that a fault occur in an actuator or sensor the corresponding element in the vector f_k^u or f_k^y will acquire a determinate value, in fault free case both vectors are zero. Some times a component of the system fails this would lead to a change in, for example, the *A* matrix.

4.2. Review of Most Popular Model Based Residual Generation Techniques

4.2.1. Parameter Estimation

This approach of residual generation has its basis on model identification. A certain parameter of the system is estimated and then compared with the one provided by a model. This parameter should have a physical meaning like stiffness, resistance, etc. The basic

scheme for this technique is shown in Figure 4-3, where $r_k = p_k - \hat{p}_k$ is the generated residuum.

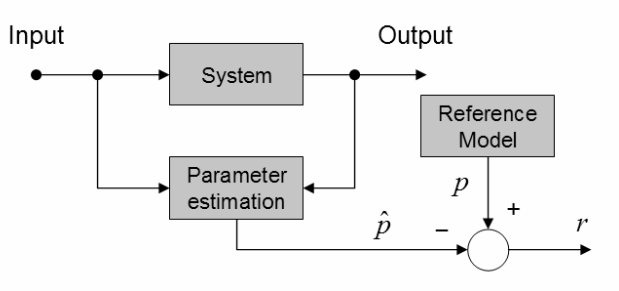

Figure 4-3: principle of parameter estimation-based residual generation

If the model is accurate and the estimation can be correctly performed, the residuum should be close to zero. When a fault occurs that has an impact in the parameter p the residuum will acquire a significant value. Generally a least square technique is used to estimate the parameter.

4.2.2. Parity Relations

One of the most popular parity relations approach is the one presented by Chow and Willsky[13]. Its formulation is based on a discrete time system like (4.1) considering the faults in the actuators and sensors as in (4.2). For simplicity the disturbance will not be included. The system then is represented as

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + L_1 f_k^u \\ y_k &= Cx_k + Du_k + L_2 f_k^y \end{aligned} \tag{4.3}$$

After a series of recursions of these equations starting from the step $k-s$ up k to, the following system is obtained

$$\begin{bmatrix} y_{k-s} \\ y_{k-s+1} \\ \vdots \\ y_k \end{bmatrix} - H \begin{bmatrix} u_{k-s} \\ u_{k-s+1} \\ \vdots \\ u_k \end{bmatrix} = Wx_{k-s} + M^u \begin{bmatrix} f_{k-s}^u \\ f_{k-s+1}^u \\ \vdots \\ f_k^u \end{bmatrix} + M^y \begin{bmatrix} f_{k-s}^y \\ f_{k-s+1}^y \\ \vdots \\ f_k^y \end{bmatrix} \tag{4.4}$$

where

$$H = \begin{bmatrix} D & 0 & 0 & \cdots \\ CB & D & 0 & \cdots \\ CAB & CB & D & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \quad W = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \end{bmatrix} \tag{4.5}$$

$$M^u = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ CL_1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{s-1}L_1 & CA^{s-2}L_1 & \vdots & 0 \end{bmatrix}, \quad M^y = \begin{bmatrix} L_2 & 0 & \cdots & 0 \\ 0 & L_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \vdots & L_2 \end{bmatrix} \tag{4.6}$$

$$Y_k - HU_k = Wx_{k-s} + M^u F_k^u + M^y F_k^y \tag{4.7}$$

Following the Chow-Willsky approach, introducing the matrix V the residual signal can be defined as

$$r_k = VY_k - VHU_k = VWx_{k-s} + V\left(M^u F_k^u + M^y F_k^y\right) \tag{4.8}$$

From this equation it is straight forward to see that in order to make the residual only sensitive to the faults it is necessary that

$$\begin{aligned} &VW = 0 \\ &VM^u \neq 0, VM^y \neq 0 \end{aligned} \tag{4.9}$$

It can be shown that there always exists and s such that $VW = 0$ [14]. Once that the matrix V is selected, each the residual can be evaluated and the decision making phase can be started, for example it can be tested if the residual doesn't exceed a threshold.

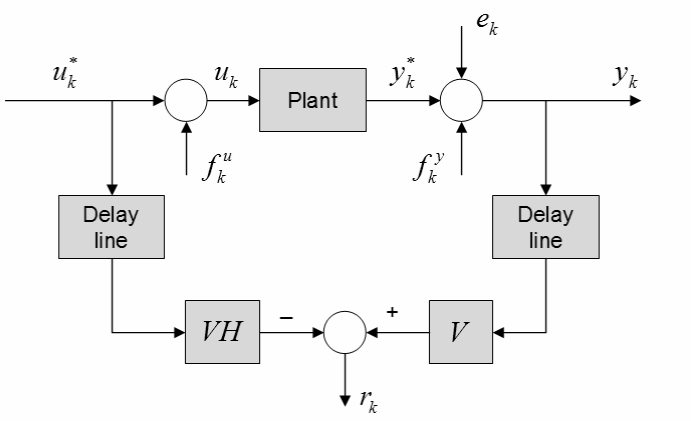

Figure 4-4: Parity relations method scheme

An important statement that must be mentioned is that it can be proved that the parity relation scheme is equivalent to the observer method (see 4.2.3) when the observer has been designed as a dead beat[15]. The inconvenience of using dead beat observers in the presence of noise is well known. However, if for example a Kalman observer is used, the effect of noise in the estimation error can be minimized. This leads to believe that the resulting residual generated by an observer method could be superior to the one obtained with parity relations under certain conditions.

4.2.3. Observer Based

The basic idea of the observer based fault detection is to compare the actual measurements with the output provided by an observer, so the residual could simply be $r_k = y_k - \hat{y}_k$, see Figure 4-5. As the comparison is made between the output of the system and the output of the observer, it is generally not necessary to design a full state observer, an output observer is enough.

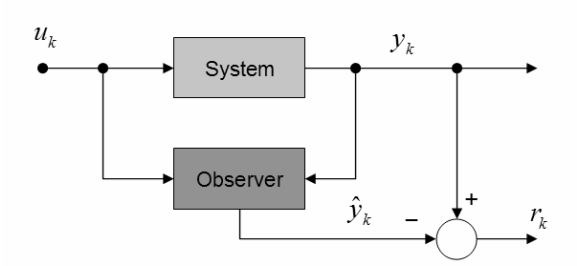

Figure 4-5: Basic scheme of the observer based residual generator

There are many observers that can be implemented, for example the simple Luenberger observer. It is also possible to implement a Kalman observer, which will provide a better performance in the case of noisy measurements. However, a commonly used observer is the Unknown Input Observer (UIO), which provides advantages.

4.2.3.1. Luenberger Observer and Kalman Predictor

Even though this observer is widely known it will be addressed here only to see how the residual are affected by a fault. Let's consider the system described in (4.3) and Figure 4-5. The system is observable, and for simplicity $D = 0$. The residual is then given by

$$r_k = y_k - \hat{y}_k \tag{4.10}$$

where $\hat{y}_k$ is the output of the Luenberger observer, whose equation is

$$\begin{aligned} \hat{x}_{k+1} &= A\hat{x}_k + Bu_k + K\left(y_k - \hat{y}_k\right) \\ \hat{y}_k &= C\hat{x}_k \end{aligned} \tag{4.11}$$

where K is the observer gain. Then, the estimation error $\tilde{x}_k = \left(x_k - \hat{x}_k\right)$ is

$$\tilde{x}_{k+1} = \left(A - KC\right)\tilde{x}_k + L_1 f_k^u - KL_2 f_k^y \tag{4.12}$$

Then, selecting K so that the matrix $\left(A - KC\right)$ is stable, the estimation error for the fault free case will converge to zero. However, for the faulty case, the estimation error will converge to a value that is a function of the faults.

From (4.10) it follows that

$$r_k = C\tilde{x}_k + L_2 f_k^y \tag{4.13}$$

Then, the residual also converge to zero for the fault free case and to a value that depends on the faults in the faulty case.

Pole placement is a very common technique for determining K. However, if the measurements are noisy or some stochastic disturbance is acting on the system the Kalman filter would provide a gain matrix K that minimizes the variance of the estimation error. Consider the system in (4.14)

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + v_k \\ y_k &= Cx_k + Du_k + e_k \end{aligned} \tag{4.14}$$

where v_k and e_k are white noise with covariance matrices R_1 and R_2 respectively. The Kalman gain K_k is given by

$$K_k = AP_kC^T\left(R_2 + CP_kC^T\right)^{-1} \tag{4.15}$$

where P_k is the variance of the estimation error is given by

$$P_{k+1} = AP_kA^T + R_1 - AP_kC^T\left(R_2 + CP_kC^T\right)^{-1}CP_kA^T \tag{4.16}$$

So the estimation error when the fault terms are considered is

$$\tilde{x}_{k+1} = \left(A - K_kC\right)\tilde{x}_k + \left(L_1f_k^u + v_k\right) + K_k\left(L_2f_k^y + e_k\right) \tag{4.17}$$

It can be seen that if $\left(A - K_kC\right)$ is stable the mean value of the estimation error converges to zero for the fault free case and to a nonzero value that depends on the fault for the faulty case. So the mean value of the residual will also converge to zero or to a value that depends on the fault case because in this stochastic scheme the residual takes the form

$$r_k = C\tilde{x}_k + L_2f_k^y + e_k \tag{4.18}$$

4.2.3.2. Unknown Input Observer

The Luenberger and Kalman observer are good options when the system model is precise and there is no disturbance acting. However, in the case of a more significant model mismatch or disturbance, the residual provided by these approaches would be affected and thus would not provide a good failure indicator.

Another observer that can be used is the Unknown Input Observer (UIO) which takes into consideration a disturbance input. Consider the system shown in (4.19)

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + L_1f_k^u + Ed_k \\ y_k &= Cx_k + L_2f_k^y \end{aligned} \tag{4.19}$$

where d_k is an unknown disturbance input.

The UIO provides an estimation that converges to the true states even in the presence of the unknown input d_k. The equations for this observer are given by

$$\begin{aligned} z_{k+1} &= Fz_k + TBu_k + Ky_k \\ \hat{x}_k &= z_k + Hy_k \end{aligned} \tag{4.20}$$

z_k is the state vector of the observer, shown in Figure 4-6.

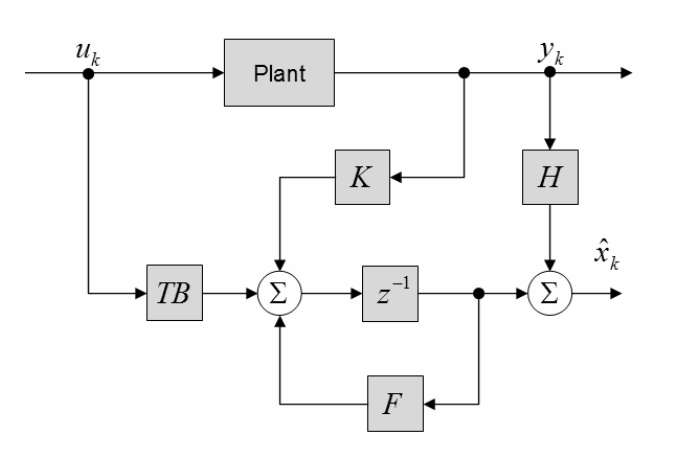

Figure 4-6: Unknown Input Observer scheme

By simple substitution, it can be seen that the reconstruction error is[11]

$$\begin{aligned}\tilde{x}_{k+1} &= \left(A - HCA - K_1C\right)\tilde{x}_k + \left(F - \left(A - HCA - K_1C\right)\right)z_k \\ &+ \left(K_2 - \left(A - HCA - K_1C\right)\right)y_k + \left(T - \left(I - HC\right)\right)Bu_k + \\ &+ \left(HC - I\right)Ed_k\end{aligned} \tag{4.21}$$

where $K = K_1 + K_2$

Then, if the matrices F, T, K and H verify the equations:

$$\begin{aligned}&\left(HC - I\right)E = 0 \\ &I - HC = T \\ &A - HCA - K_1C = F \\ &FH = K_2 \\ &K = K_1 + K_2\end{aligned} \tag{4.22}$$

The estimation error would be:

$$\tilde{x}_{k+1} = F\tilde{x}_k \tag{4.23}$$

It can be seen that designing F to be stable, the estimation error will converge to zero. So K_1 can be selected by simple pole placement for an equivalent system $\left(A_1, C\right)$ with state matrix $A_1 = A - HCA$.

A special solution[14] for H is

$$H = E\left(CE\right)^+ \tag{4.24}$$

where $\left(CE\right)^+$ is the pseudoinverse of CE.

The design procedure can be summarized as

1. Define the matrix E.
2. Calculate H according to (4.24).
3. Find the matrix $A_1 = A - HCA$ and determine an observer gain matrix for the system $\left(A_1, C\right)$.

4. Use the set of equations (4.22) to calculate the remaining matrices.

An important property of the UIO is that the additional term Ed_k provides certain robustness properties to the observer, as it can be interpreted an additive disturbance, model uncertainty, time varying dynamics, etc. For example, in the case of a model mismatch, the true values $\{A,B\}$ are not exactly known. However, an estimation $\{A_0,B_0\}$ of the true parameters is available. A common representation of model mismatch is the inclusion of an additive uncertainty such that

$$\begin{aligned} A &= A_0 + \delta A \\ B &= B_0 + \delta B \end{aligned} \tag{4.25}$$

It can be considered that the models of the sensors are accurate so the matrix C is known almost exactly[16]. Then by replacing (4.25) in (4.19)

$$x_{k+1} = \left(A_0 + \delta A\right)x_k + \left(B_0 + \delta B\right)u_k + L_1 f_k^u + Ed_k \tag{4.26}$$

Then, for certain cases of δB the unknown input d_k and the term $\delta B u_k$ could be considered as disturbance acting on the system $E'd_k' = Ed_k + \delta B u_k$, and the UIO would partly overcome this mismatch. Of course, this solution can only be applied for the cases where E that stabilize A_1 can be obtained.

4.2.3.3. Dedicated Observer Scheme

When it is desired to detect more than one fault using the observer based residual generation a possible solution is to use the dedicated observer scheme. In this case, another problem must be tackled: fault identification, because it is not sufficient to detect a fault, it is necessary to identify where the failure have occurred. The dedicated observer scheme was introduced in 1975[17], and since it has been many times upgraded. In this work a simple approach is shown.

If it is desired to monitor the m inputs to the system then m observers must be designed. Each observer should be fed with all but one input, and all the outputs, like shown in Figure 4-7. From each observer a residual vector $r_i = y_i - \hat{y}_i$ is generated. Then, in the presence of a fault in the i-th actuator, the residual from all observers will be affected, except from the residual of the observer which has not been fed by the i-th input. In this way by a simple logic the faulty actuator can be identified.

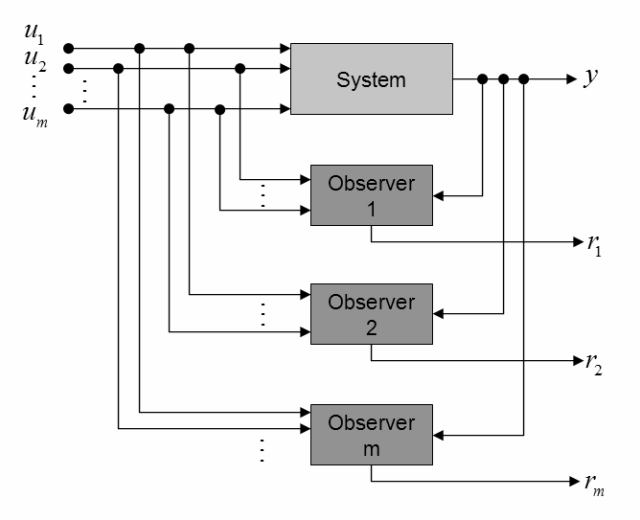

Figure 4-7: Dedicated observer scheme for actuator fault detection

To detect faults in the sensors the procedure is similar: as many observers as sensors should be designed and all of them should be fed by all inputs and only one output. When a fault occur in the i-th sensor, only the residuum from the i-th observer will be affected.

It can be seen that more than one faulty sensor can be identified simultaneously by this technique; however, only one actuator fault at a time can be acknowledged.

A common strategy for designing the observers for this configuration is given in [11] and transcribed here; however, other observers can be selected according to the application. This is a variation of the unknown input observer presented in 4.2.3.2 and has the particularity of being insensitive to one of the command inputs. In this way, one observer is designed for each of the command inputs and the dedicated observer scheme can be determined. The equations of the i-th observer are

$$\begin{aligned} z^i{}_{k+1} &= \left(T^i A - K^i C\right) z^i{}_k + J^i u_k + S^i y_k \\ r^i{}_k &= L^i_1 z^i{}_k + L^i_2 y_k \end{aligned} \tag{4.27}$$

T^i is a linear transformation of the state vector. Under the hypothesis of no fault in the inputs and low amount of measurement and process noise, the estimation error for the i-th observer is $\tilde{x}_k = z^i{}_k - T^i x_k$, so it can be proven[14] that

$$\tilde{x}^i_{k+1} = F^i \tilde{x}^i_k + \left(F^i T^i - T^i A + S^i C\right) x_k + \left(J^i - T^i B\right) - T^i f^u_k \tag{4.28}$$

And the residual is

$$r^i{}_k = L^i_1 \tilde{x}^i_k + \left(L^i_1 T^i + L^i_2 C\right) x_k \tag{4.29}$$

So, by selecting the matrices

$$
\begin{aligned}
T^i &= I - B_i \left(CB_i\right)^+ C \\
F^i &= T^i A - K^i C \\
S^i &= K^i - F^i B_i \left(CB_i\right)^+ \\
J^i &= T^i B \\
L_1^i &= -C \\
L_2^i &= \mathrm{I} - \left(CB_i\right)\left(CB_i\right)^+
\end{aligned}
\tag{4.30}
$$

the estimation error becomes

$$
\begin{aligned}
\tilde{x}_{k+1}^i &= F^i \tilde{x}_k^i + T^i B f_k^u \\
r^i{}_k &= L_1^i \tilde{x}_k^i
\end{aligned}
\tag{4.31}
$$

In equation (4.30) B_i is the i-th column of B. Then, by designing K^i so that F^i is stable, the estimation error converges to a value that only depends on the actuator fault f_k^u. It is important to note that as a result of this set of equations the i-th column of the matrix J^i is zero, thus making the observer insensitive to the i-th input. The scheme for this observer is shown in Figure 4-8.

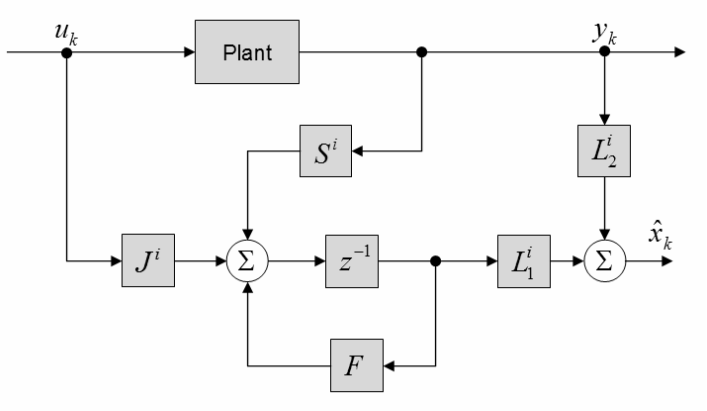

Figure 4-8: Observer insensitive to the *i*-th input.

4.3. Design of the Fault Detection and Identification System

The residual generation techniques presented in the previous section do not include nonlinear cases like the one from the UAV. In the literature different approaches have been stated to extend them to nonlinear systems. For example extended Kalman filters and extended Luenberger observers have been proposed and successfully used[18]. However, the convergence of these observers is most times not guaranteed and they tend to be sensitive to model mismatch errors. The nonlinear UIO is also an alternative, but the complexity of the design procedure is considerably high, thus limiting the applications in which it can be used.

In this thesis the nonlinear case is not tackled. Instead, the system is linearized at an equilibrium point and the fault detection strategy is determined for this linearized system.

Finally, its performance is analyzed by simulations on the nonlinear model at a variety of working points. If the performance achieved is not acceptable, the nonlinear case should be considered. As in any linearization, at other working points, the difference between the linearized and the actual model can be significant. In this situation, the residual obtained would grow due to the model mismatch even when no fault has occurred. It is possible that at certain flight situations the residual for the non faulty condition reach a level high enough to be considered as the occurrence of a fault. To attenuate the possibility of a false alarm, a variation of the dedicated observer scheme is proposed in 4.3.2.

4.3.1. Linearization

The linearization must be done in an equilibrium point. According to the dynamic equations the equilibrium points of interest for any aircraft are:

- Steady wings level flight
- Steady turning flight
- Steady pull up
- Steady roll

The equilibrium point of interest for this part of the thesis is one in which the ailerons and rudder have a deviation angle different than zero, so as to be able to obtain information about their incidence on the system. This is verified is the steady roll, which is defined as the flight situation when the following conditions are verified:

$$\begin{aligned} &\dot{P},\dot{Q},\dot{R},\dot{U},\dot{V},\dot{W} = 0 \\ &\dot{\theta},\dot{\psi} = 0 \\ &\dot{\phi} = roll\,rate \end{aligned} \tag{4.32}$$

As the nonlinear model of the UAV is available in Simulink, the linearization model is derived using the MATLAB linearization function "linmod". Again, due to legal reasons, the system obtained will not be transcribed in this work. The roll rate settled for the linearization is .5°/sec at a speed of 50m/sec.

4.3.2. Variation of the Dedicated Observer Scheme and Decision Making

Generally, when using a dedicated observer scheme, the decision making phase of the fault detection is based on checking if the norm of the residual has exceeded a threshold. If the threshold is crossed, it is concluded that a fault has occurred. It is important to note that the threshold does not need to be constant. There are several techniques for selecting this threshold; some of them are summarized by Michael Bask in [19], that presents an adaptive threshold method.

In this work a simple variation of the dedicated observer scheme is employed for detecting faults in actuators. Apart from the m observers shown in Figure 4-7, another observer is implemented. This additional observer is fed by all the command inputs and all the measurements, as any regular state observer. This is shown in Figure 4-9, where the additional observer is called *reference observer*. The estimation error of this observer is used as a reference residual r_{ref}. In the fault free scenario, if the model used in the observer is exact, r_{ref} would converge zero, but any fault in an actuator would increase its value, so the reference observer provides an indication of the fault free scenario.

As stated in 4.2.3.3, each of the other m observers is sensitive to faults in all but one actuator, so if a fault occurred in the i-th input, only r_i would remain at a low value. In this way, the norm of the residual is a measurement of the accuracy in which each observer is estimating each fault scenario. Then, the observer that provides the minimum residual is the one that best matches the actual situation of the system. So, instead of checking whether the norm of any residual crosses a threshold or not, the norms of $r_1,\cdots,r_m$ and r_{ref} are measured and the smallest residuum identifies the situation. If the reference observer has the smallest estimation error it can be concluded that no fault has occurred. On the other hand if r_i reaches the smallest value, then a fault has occurred in the i-th input thus identifying the faulty actuator.

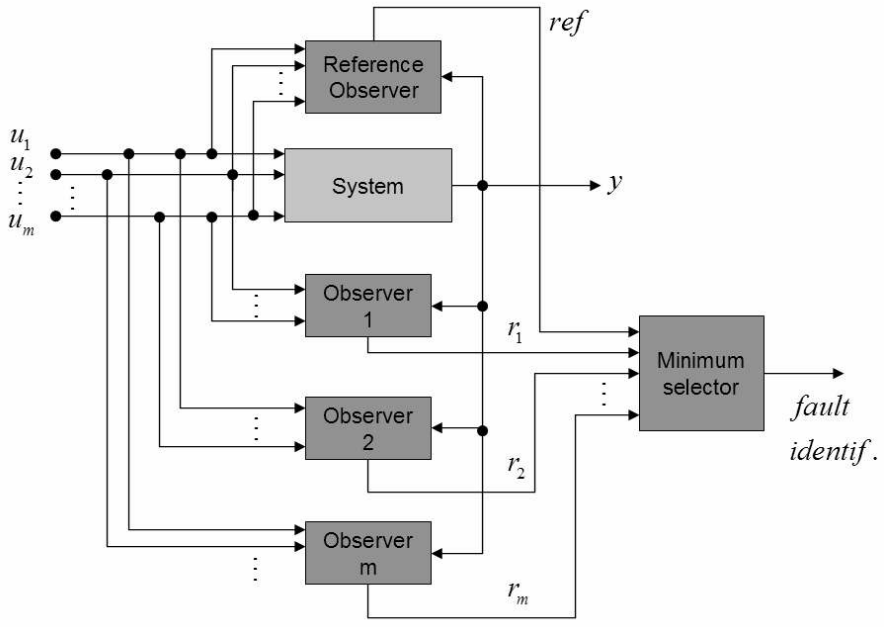

Figure 4-9: Variation of the dedicated observer scheme

4.3.3. Dedicated Observer Scheme Development

In principle, under the hypothesis of observability, any type of observer can be used for this scheme and the results will depend on the characteristics of the plant and the tuning of the observers. Several configurations of observers were tested for the residual generation. An acceptable performance achieved with the UIO implemented in the scheme of Figure 4-9 is presented in this work. The sample time used is 0.1sec.

It is not necessary to design observers that estimate the complete output vector in order to generate the residuals vectors. It is sufficient to design observers that estimate the outputs that provide information about the variable of interest (sensor, actuator or parameter).

From the physics of the problem it is well known that the effects of the ailerons and the rudder are not relevant on certain output variables for example θ (pitch) and Q (angular velocity about the z angle). On the other hand it is certain that the influence of the ailerons and the rudder in the angular velocities about the x and y angles is significant. Another fact that should be considered is that depending on the design of the observer some state variables will not be estimated as accurately as others, thus making the residual more or less valid. In this work, the results obtained when measuring only the angular

velocity about the x axis, and the sideslip angle β are presented. In this way, each of the residual vector $r = y - \hat{y}$ have $\Re^2$ dimension.

The decision making phase used is mainly composed by averaging filters and a selection logic. A representation can be seen in Figure 4-10, where r_{ref} is the residual generated by the observer that measures all the command signals, r_{ail} is the residual produced by the observer that does not measures the aileron command signal and r_{rdr} is the residual generated by the observer that does not measures the rudder command signal.

The first action performed is the calculation of the norm 1 of each residual obtaining an $\Re^1$ dimensional signal.

The averaging filter is a basic auto regressive moving average. Its transfer function is

$$G(z) = \frac{k}{z^2 - z + k} \tag{4.33}$$

The factor k is the "forgetting factor" which provides a tuning parameter for the filter so as to make the average more or less extent in time. The average is a much more reliable signal that prevents the residual to reach low values for small time slots. The forgetting factor was set to 0.0015.

The selection logic simply finds the minimum of the residuals and provides an output number that characterizes it, thus identifying the fault free scenario or the faulty actuator.

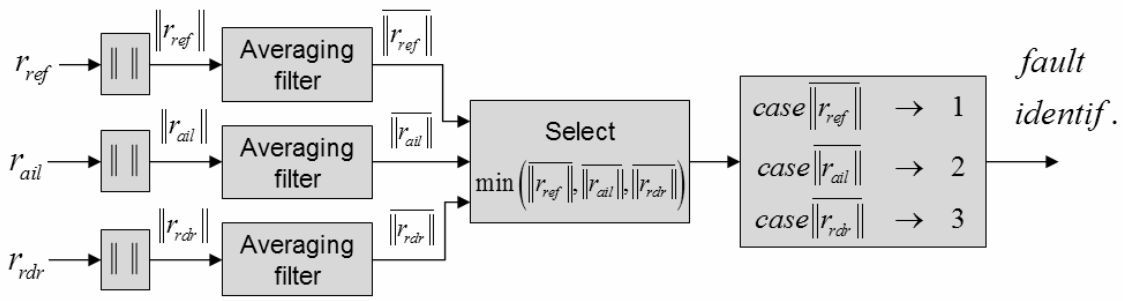

Figure 4-10: Decision making phase

In the fault free scenario, all the residual must present a low value, however, it is know that a fault occurrence is quite improbable and this fact must be considered. One way of doing it is to add a weighting factor to the averaging filters that represent the likelihood of each scenario to occur. To include this consideration in the design, a weighting factor was added to the average filter that reduces the residual value $\overline{\|r_{ref}\|}$ and tuned empirically to a value of 0.5.

The selection logic simply provides as an output a number that identifies the lowest of the residuals.

4.4. Simulations

The simulations are performed at a coordinated turn maneuver, where both, the ailerons and rudder are acting. First the results obtained for the fault free case are shown. Following the simulations when the aileron fails are presented and finally the results in the case of rudder failure are given.

4.4.1. Fault Free Case

This simulation begins with the UAV in a straight level flight. After the steady state has been reached at about 500sec a coordinated turn maneuver is performed at a target bank angle of 10° (double than the linearization point) and at a speed of 50m/sec (equal to the linearization point). The remaining time of the simulation the UAV keeps turning at the same target angle. The results are shown in Figure 4-11. On the left the norms of the residuals are plotted. This is the signal present at the input of the averaging filters (see Figure 4-10). The notation used in the graphs is:

- $\|r_{ref}\|$: norm of the residual of the observer that estimates the fault free scenario (reference).
- $\|r_{ail}\|$: norm of the residual of the observer that estimates the "failure in the aileron" scenario.
- $\|r_{rdr}\|$: norm of the residual of the observer that estimates the "failure in the rudder" scenario.
- $\overline{\|r_{ref}\|}$: "average" of $\|r_{ref}\|$ generated by the filtering $\|r_{ref}\|$ with the filer (4.33). The same holds for $\overline{\|r_{ail}\|}$ and $\overline{\|r_{rdr}\|}$.

On the graph at the right of Figure 4-11 the "averaged" residuals are plotted. It can be seen that before and after the turn the average reference residual $\overline{\|r_{ref}\|}$ is smaller than the others. In this way, the selection logic selects this as the minimum and correctly identifies the situation as normal.

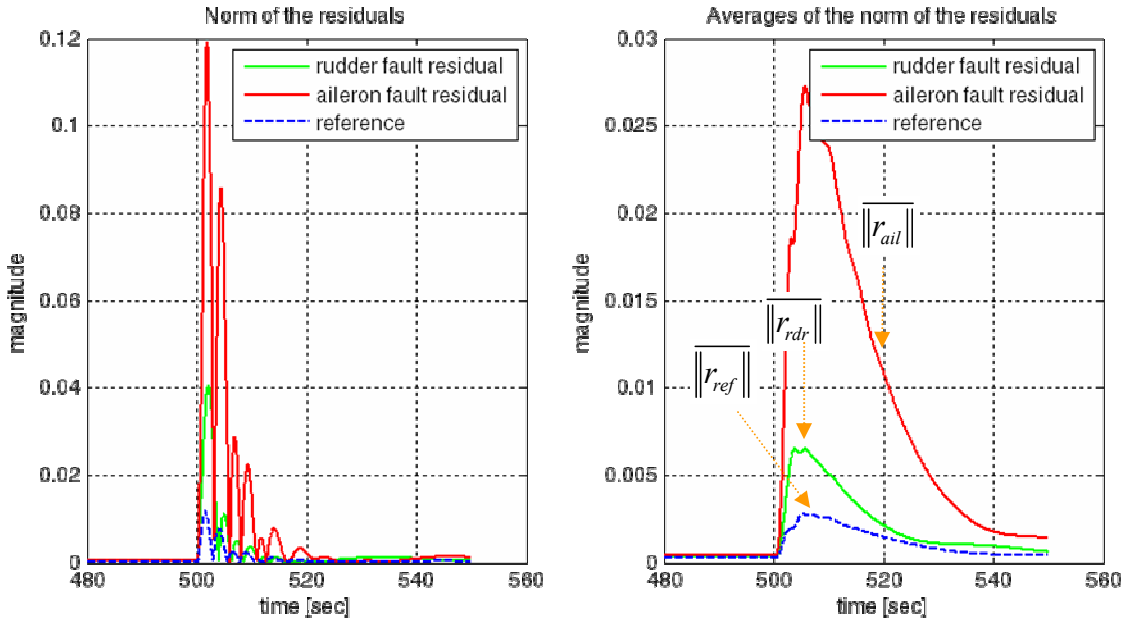

Figure 4-11: (left) Norms of the residuals and (right) averages of the norm of the residuals for the fault free case when a coordinated turn maneuver is performed.

A fact that deserves special attention is that many times $\|r_{ail}\|$ reaches a smaller value than $\|r_{ref}\|$ for a few seconds; this can be seen in the figure of the left. At this point the estimation of the observer that does not measures the aileron command signal crosses the true state values, thus generating a minimum in its residuals. It is most important that the fault detection system does not mistake this situation with the occurrence of a failure in the ailerons. To prevent this from happening the averaging filters are included. On the right graph it can be seen that $\overline{\|r_{ref}\|}$ is always smaller than the other signals.

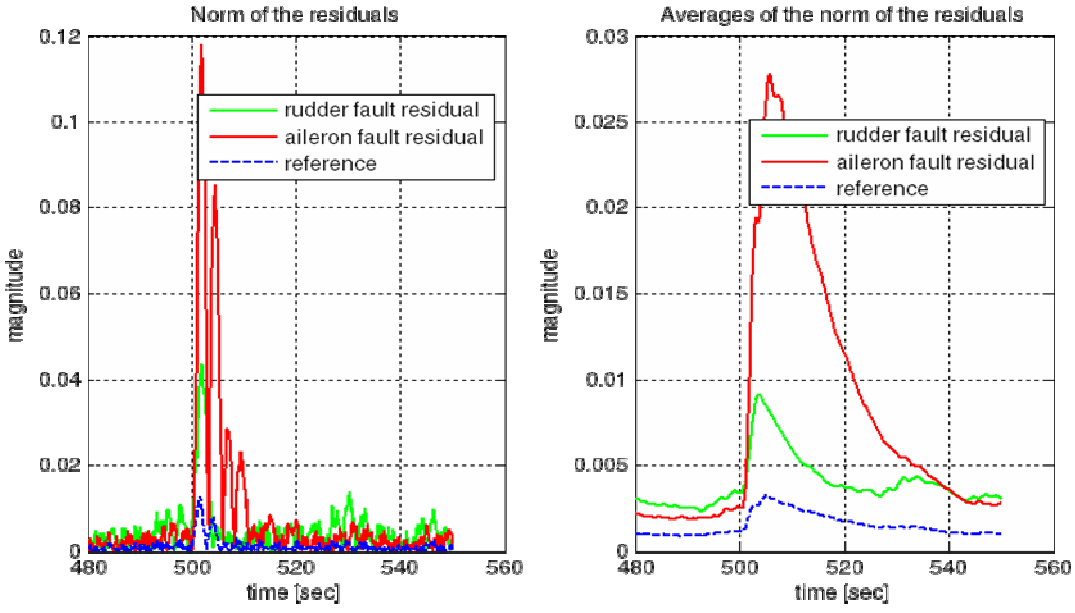

Figure 4-12: Results for the fault free case when a coordinated turn maneuver is performed and measurement noise is present.

In Figure 4-12 the same situation is repeated but this time measurement noise is included. As the measurement noise affects all of the residuals in the same proportion the noise does not present a major inconvenience. It can be noted that before the turn maneuver is performed the steady state residuals for this operation point have increased their values with respect to the previous simulation due to the presence of the measurement noise.

4.4.2. Faults in the Aileron

4.4.2.1. Loss of Power Supply

In this subsection the same experiment is repeated but as a result of the loss of power, there is no deflection in the ailerons, so even though there is a command signal being sent to the ailerons, they remain at 0°. The results of the simulations are shown in Figure 4-13. The detection is done in less than 1 sec after the turning maneuver command is sent the failure is detected and identified.

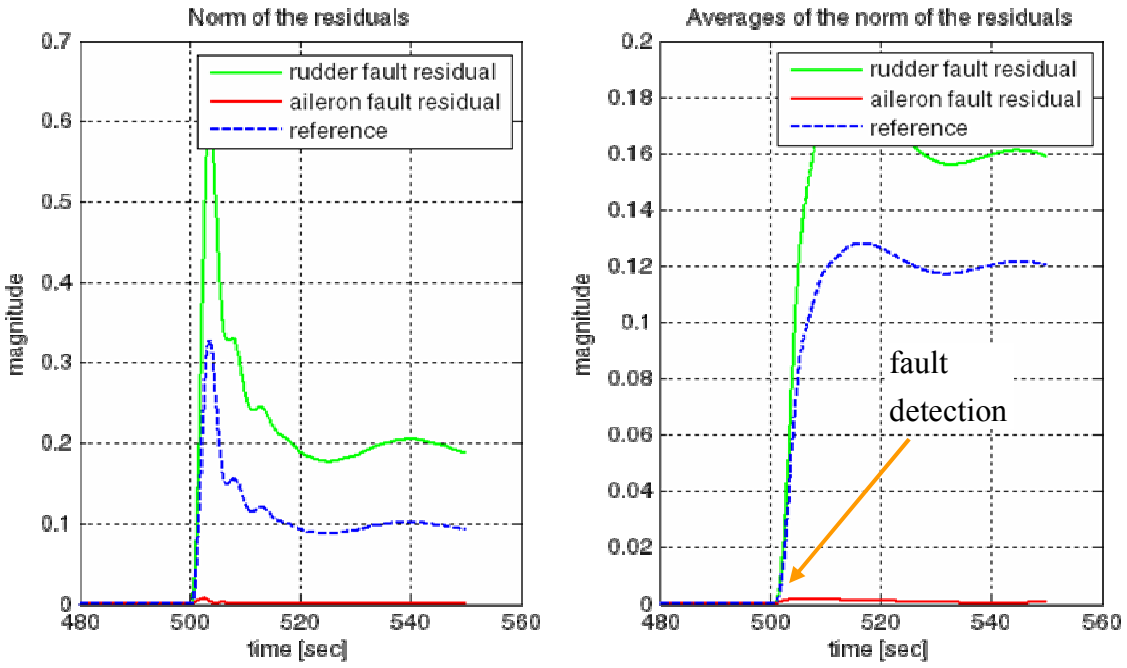

Figure 4-13: Results for the case when a loss of power prevents the ailerons from deflecting. The failure is detected and identified in less than 1 sec.

The detection of this fault is also possible at working points considerably different than the linearization point.

4.4.2.2. Actuator Stuck At a Fixed Deviation Angle

The same experiment is repeated in this subsection but the actuator gets stuck at a certain deviation angle. This could be the outcome of a mechanical problem.

Again the simulation begins with the UAV at a steady flight. At 500sec a command to start a turn is given. The ailerons start to deflect but at an angle of 0.3° they get stuck (500.5sec). The results are shown in Figure 4-14.

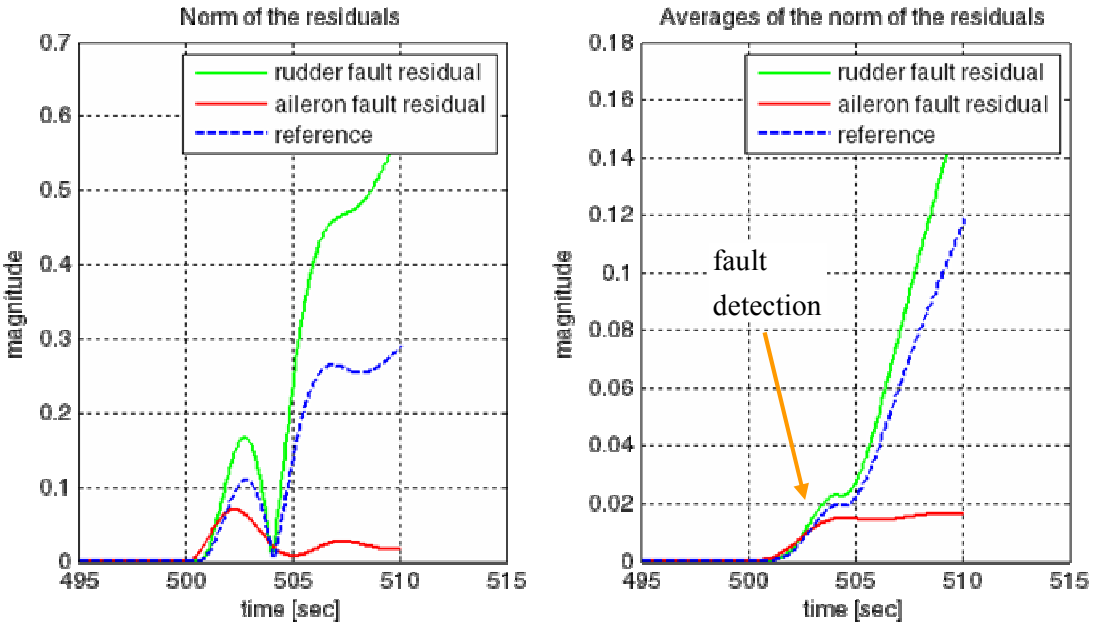

Figure 4-14: Results for the case in which the ailerons get stuck at 0.3° when performing a coordinated turn maneuver. The fault is detected after 3.5sec

It can be seen that the fault is successfully detected and identified after 3.5sec. Nevertheless in this case it took more time for the fault detection system to notice that a fault has occurred. This result is not surprising because there was an actual command signal being sent to the ailerons to deflect them, so the fault is not evident for some seconds.

The detection takes more time if the target bank angle is small and there is more noise in the measurements. However, increasing the speed of the UAV at many times the speed at which the linearization was made does not have a big influence in the amount of time that it takes to detect the fault. In fact, the detection of this fault is achieved faster if the UAV speed is increased.

4.4.3. Faults in the Rudder

In this subsection the experiment faults in the rudder are simulated. The experiment is the same as in the previous subsections.

4.4.3.1. Loss of Power

Again, a loss of power in an actuator is simulated, this time on the rudder. As a result the rudder does not deflect when the coordinated turn maneuver starts and remains at 0°. The outcome is shown in Figure 4-15.

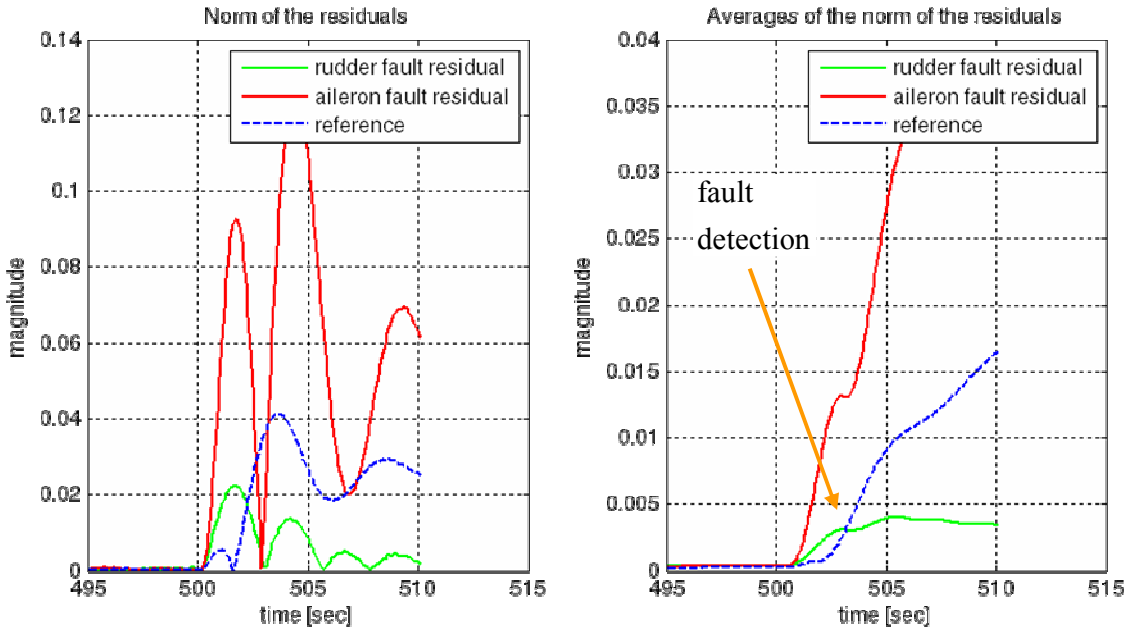

Figure 4-15: Results for the case when a loss of power prevents the rudder from deflecting. The failure is detected and identified after 3 sec.

It can be see in this figure that the fault is detected after 3 seconds, a considerably higher amount of time than for the ailerons. The main reason for this is that the rudder is not performing a predominant task in the coordinated turn, so its effects are not as noticeable from the outputs as the ones of the ailerons. As a result, it takes more time for the residuals to grow to the required level for fault detection.

4.4.3.2. Actuator Stuck At a Fixed Deviation Angle

Finally, the result of the simulation when the rudder is stuck at 0.3° is shown in Figure 4-16. It can be seen that it took about 20 sec to detect the fault. For this case the efficiency of the fault detection system is considerably lower than for the case of a fault in the ailerons. However, it is important to consider that the main task of the rudder is to maintain the sideslip angle at a low level, so the delay in the detection of the fault is not so critical; more about this will be said later.

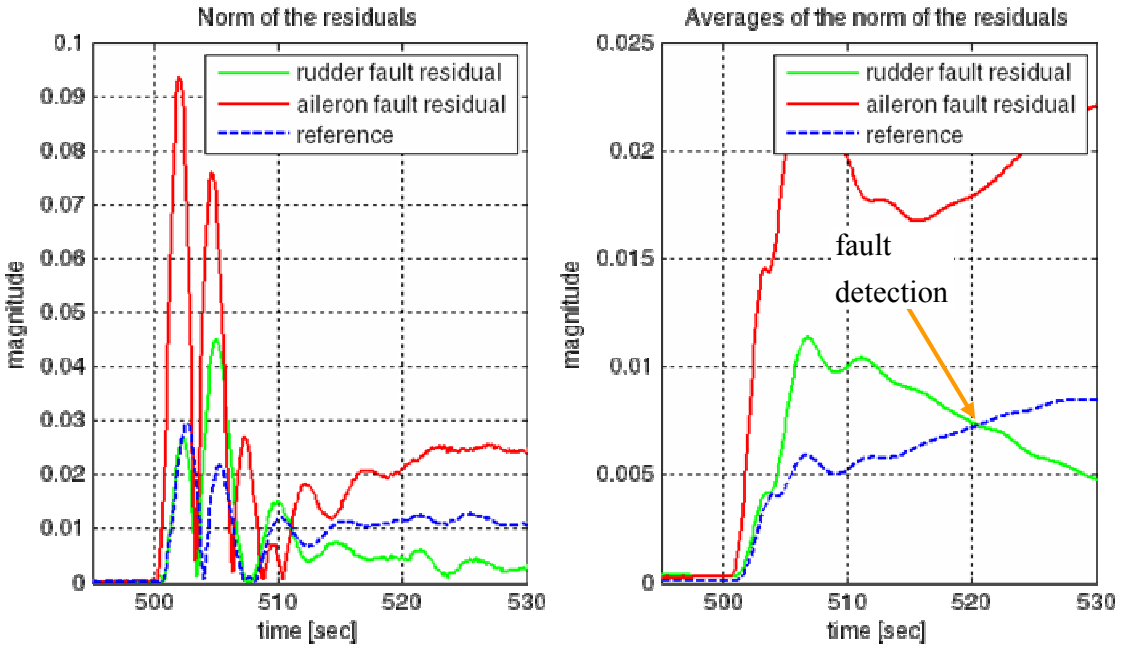

Figure 4-16: Results for the case in which the ailerons get stuck at 0.3° when performing a coordinated turn maneuver. The fault is detected after 20sec

5. Contingency Strategy

In this section, the possibility of determining a contingency control strategy to be applied when a fault is detected is studied; this represents the objective 3 of this work. The goal of this strategy is to allow the UAV to maintain basic functionality after a fault occurred. For this work, the basic functionality is considered to be maintained if the UAV is able to reach a straight level flight and can perform a turn maneuver at a desired roll set point. As stated in Objective 3, no additional hardware or redundancy can be used, so the strategy will be limited to the development of controllers.

Two additional controllers have to be design. One to control the rudder when there is a failure in the ailerons, which is called *rudder failure contingency controller* in this work. The second is the controller that has to command the rudder when a failure in the ailerons has been detected, this one is called *aileron failure contingency controller*.

The strategy implemented is a switch of controllers. If a failure in the ailerons is detected, the contingency controller for this fault is used to command the rudder. On the other hand if a failure in the rudder is detected, the rudder failure contingency controller is used to command the ailerons. This is shown in Figure 5-1.

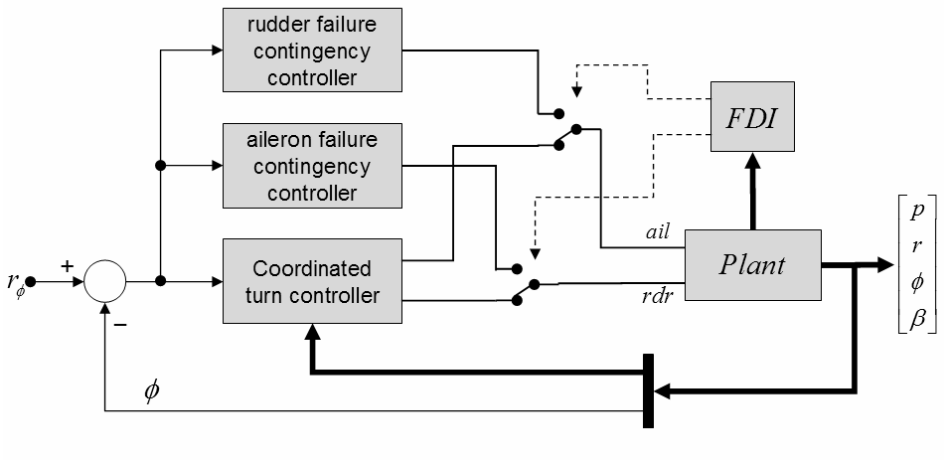

Figure 5-1: Contingency scheme

5.1. Controller for the Aileron Failure Contingency

The goal of this controller is to successfully control the rudder of the UAV when a fault in the ailerons has occurred. The faults that are considered are the ones stated in 4.4.2.

5.1.1. Development

In the case of a failure in the ailerons as severe as the ones considered in this work, the rudder constitutes the only actuator that can be used to govern the UAV turns. Then, the rudder should no longer be used to keep the sideslip angle β close to zero, but it has to be used to reach the desired roll set point.

The controller developed to perform this task is the LQG. As it is required to reach a set point, an integrator is included, thus creating an augmented system that can track a step command with zero steady state error. The configuration is shown in Figure 5-2, where $G_{rdr,\phi}(s)$ is the transfer function from the rudder to the roll angle.

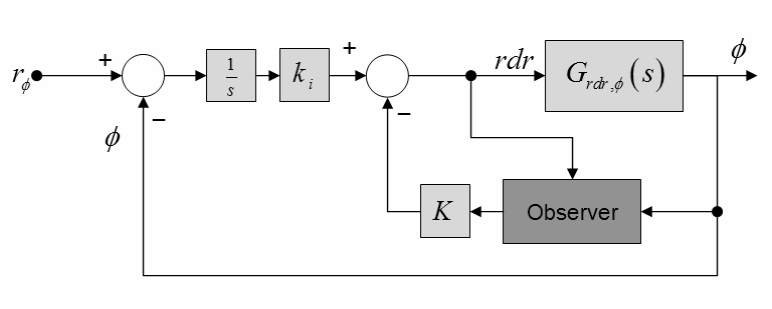

Figure 5-2 Aileron failure contingency controller

A set of equations equivalent to (3.13) can be used to derive this augmented system and calculate the state feedback matrix using the LQR technique. A discretized version of the augmented system with sample time 0.1sec is given in (5.1) in state space form. The same linearization point used for deriving the coordinated turn is used here.

$$\begin{aligned}\dot{x} &= \begin{bmatrix} 0.35 & 0.06 & -0.01 & 0.02 & 0 \\ -1.57 & 0.01 & -0.85 & 1{,}71 & 0 \\ -1.27 & -0.8 & 0.31 & 1.37 & 0 \\ -2.09 & -0.85 & -0.77 & 2.53 & 0 \\ 0.02 & 0.07 & -0.07 & -0.01 & 1 \end{bmatrix} x + \begin{bmatrix} -0.23 \\ -0.65 \\ -0.72 \\ 0.10 \\ 0 \end{bmatrix} u \\ y &= \begin{bmatrix} -0.23 & -0.65 & 0.716 & 0.1 & 0 \end{bmatrix} x \end{aligned} \tag{5.1}$$

Using the LQR command in MATLAB the linear quadratic regulator can be calculated. The feedback matrix obtained with this command has the form:

$$K_{fb} = \begin{bmatrix} K & k_i \end{bmatrix} \tag{5.2}$$

After recursively tuning the weighting matrices a satisfactory performance is obtained with

$$K_{fb} = \begin{bmatrix} -0.02 & 0.09 & -0.13 & 0.02 & 0.07 \end{bmatrix} \tag{5.3}$$

The Kalman observer has been used to estimate the state vector.

5.1.2. Simulations

In order to test the performance of the contingency controller under the two failures cases studied (see 4.4.2) two simulations scenarios very similar to the ones performed in 4.4 are presented. The UAV is at a steady level flight and a turning maneuver command is given to reach a desired roll set point but this time, after 100 seconds, it is desired to reach a steady level flight again. The tasks have to be achieved even in the presence of a failure in the ailerons.

5.1.2.1. Loss of Power

This simulation is almost the same as the one presented in 4.4.2.1. It also begins with the UAV in a steady level flight. At about 500sec a turn maneuver is performed at a target bank angle of 15° but this time at 600sec the roll angle set point is set to zero. As a result from the loss of power in the actuator, the ailerons do not deflect and remains at 0°.

The results are presented in Figure 5-3. The fault is detected in less than 1 second (see 4.4.2.1). It can be seen that the contingency controller successfully deflects the rudder so as to achieve the target roll angle.

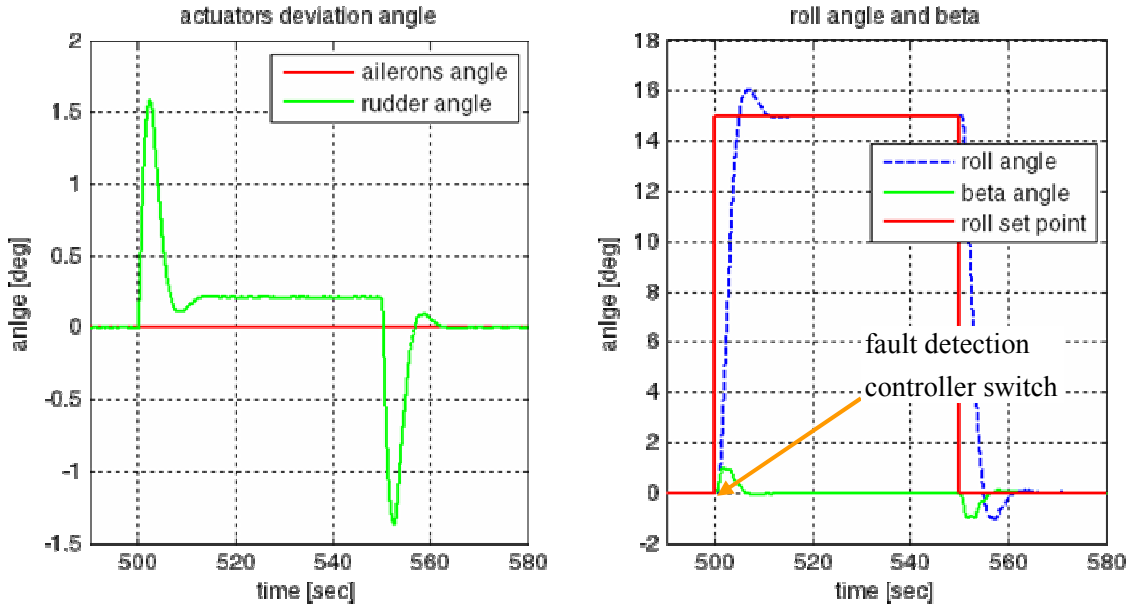

Figure 5-3: (left) actuators deviation and (right) roll angle for the case of a power loss in the ailerons. The contingency controller successfully reaches the desired bank angle.

5.1.2.2. Actuator Stuck At a Fixed Deviation Angle

In this subsection the simulation is repeated to test the performance of the contingency controller when the ailerons are stuck at a fixed deviation angle. The results are shown in Figure 5-4, where it can be seen that the ailerons are stuck at 0.5°.

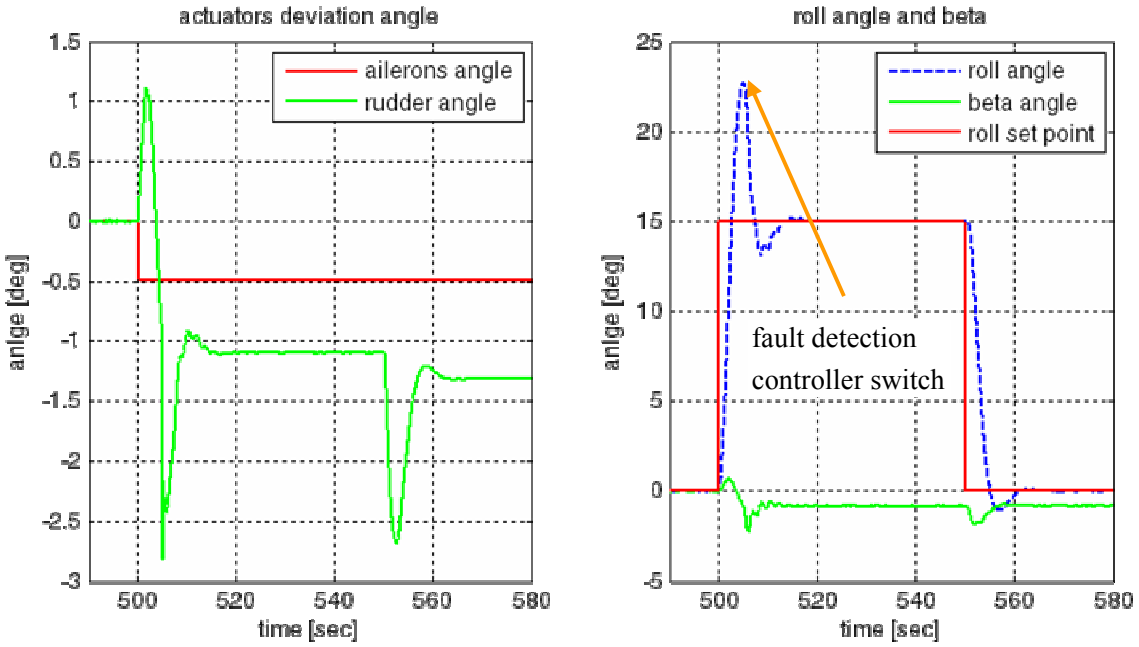

Figure 5-4: Results for the case when the ailerons are stuck at a fixed angle due to a mechanical problem. The contingency controller successfully reaches the desired bank angle.

Just as described in 4.4.2.2 after about 5 sec the fault is detected, but this time the contingency controller is activated and using only the rudder the desired roll angle is successfully reached.

It is important to notice that the sideslip angle has not been taken into consideration for control purposes in this maneuver, and of course it does not remain close to zero. As the sideslip angle is not zero, the weight vector of the payload inside the UAV would not be directed to the bottom of the aircraft but directed to the sides. This effect is unwanted, but it is of small concern compared with the loss of control of the aircraft.

It can be seen that after flying 5 sec with the ailerons fixed at 0.5° the UAV reached a 24° bank angle. If the bank angle increases too much, the airplane becomes highly unstable, so it would be desired that the fault detection system acted faster to avoid a more dangerous situation.

The worst case scenario for this kind of fault is that the ailerons got stuck at the maximum deflection angle of the ailerons. For this UAV, in normal conditions (fault free case) the maximum deviation angle of the ailerons needed to reach a 35° bank angle is of only about 3° and for a small fraction of time. It should be considered that for commercial airplanes the bank angle generally does not exceed the 25° bank angle, so this is a very demanding command. This worst case scenario was simulated and the fault detection showed to be not fast enough and the UAV became unstable before the fault could be detected. It was determined that if the ailerons got stuck at an angle higher than 1.5° (about ½ of the maximum) the fault detection system is not fast enough to act before UAV reaches a bank angle of 55°, which is highly dangerous for stability. This result is shown in Figure 5-5.

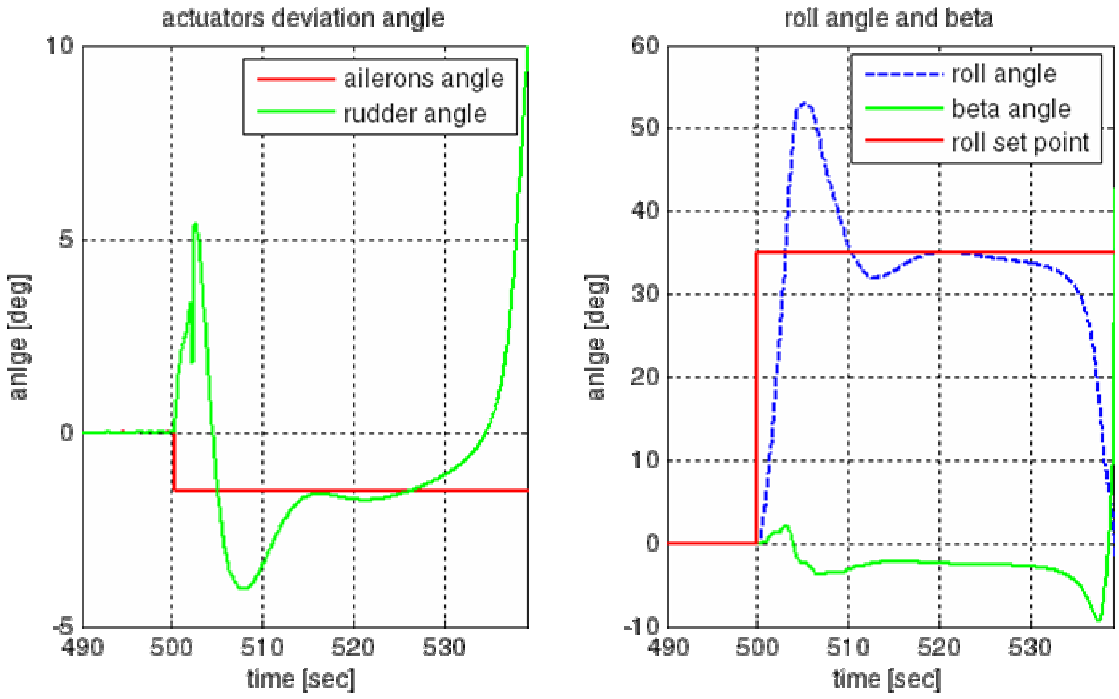

Figure 5-5: Results when the ailerons are fixed at 1.4°. It can be seen that the fault detection is not achieved fast enough to prevent the UAV from reaching a dangerous 55° bank angle.

Even if the fault detection was immediate, another limiting factor of this contingency strategy is the maximum angle at which the ailerons can be stuck and it is still possible to control the aircraft with the rudder. By simulations it was found that when the ailerons are stuck at angle higher than about 2° the contingency controller derived can not take the UAV to the desired bank angle using only the rudder. The results are presented in Figure 5-6.

These results are not at all surprising. It is highly improbable that any contingency strategy could perform successfully under an extreme situation like the one analyzed here, that is an airplane with the ailerons stuck at almost the maximum deviation for a normal maneuver.

Another expected result that could also be verified by simulations is that at higher speeds the UAV can reach higher roll angles without loosing stability. In this way, the highest angle at which the ailerons can get stuck and the contingency system still performs successfully is also higher.

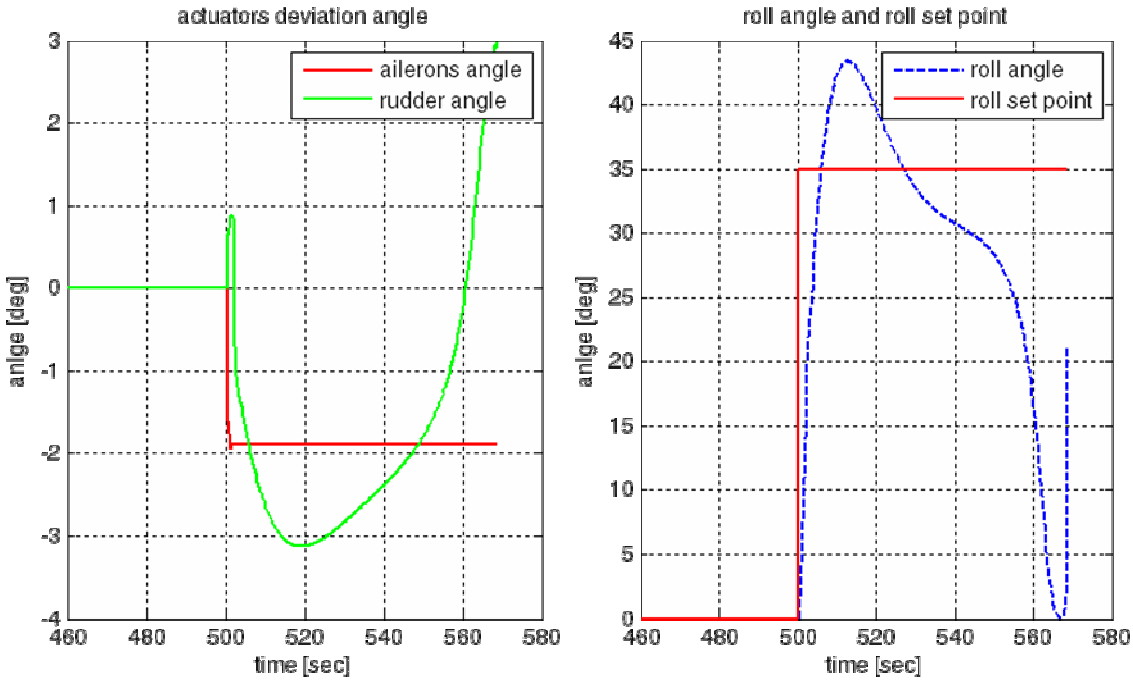

Figure 5-6: Results when the ailerons are stuck at 2°. Even if the fault detection is immediate, the contingency controller is not able to control the UAV using only the rudder.

5.2. Controller for the Rudder Failure Contingency

The goal of the "rudder failure contingency controller" is to successfully command the ailerons of the UAV when a fault in the rudder has occurred. The faults that are considered are the ones stated in 4.4.2.

5.2.1. Development

The development of the contingency controller to be used in the case of a failure in the rudder is totally equivalent to the development of the "aileron failure contingency controller" presented in the previous subsection. Its objective is to provide a zero steady state error to a step command in the roll set point input by commanding the ailerons. The scheme used is the one shown in Figure 5-7.

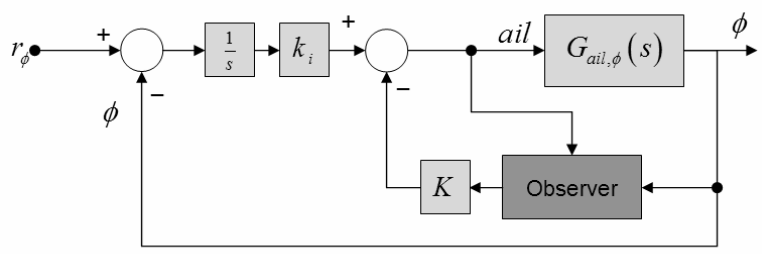

Figure 5-7: Rudder failure contingency controller scheme.

An LQG controller is used to control the augmented plant obtained after the integrator is included. The augmented plant discretized at a sample time of 0.1 sec.

$$\dot{x} = \begin{bmatrix} 0.94 & 0.15 & 0.02 & 0 \\ -0.11 & 1.34 & 0.05 & 0 \\ 1.37 & -4.07 & 0.57 & 0 \\ 0.08 & 0.05 & 0.01 & 1 \end{bmatrix} x + \begin{bmatrix} -0.17 \\ 0.46 \\ -1.08 \\ 6e-5 \end{bmatrix} u \qquad (5.4)$$
$$y = \begin{bmatrix} -0.85 & -0.49 & -0.07 & 0 \end{bmatrix} x$$

The state feedback matrix used is

$$\begin{bmatrix} K & K_i \end{bmatrix} = \begin{bmatrix} -0.11 & 0.01 & -3e-3 & -0.01 \end{bmatrix} \qquad (5.5)$$

The state vector is obtained by means of a Kalman state observer.

5.2.2. Simulations

In order to evaluate the performance of the rudder failure contingency controller, the same simulations performed in the previous subsection are repeated here, however, this time the fault occurs in the rudder.

5.2.2.1. Loss of Power

The results for the case when the rudder remains at 0° due to a loss of power are shown in Figure 5-8. It can be seen that after the fault has been detected (about 3sec) the contingency controller takes the command of the ailerons, and the desired roll angle is achieved.

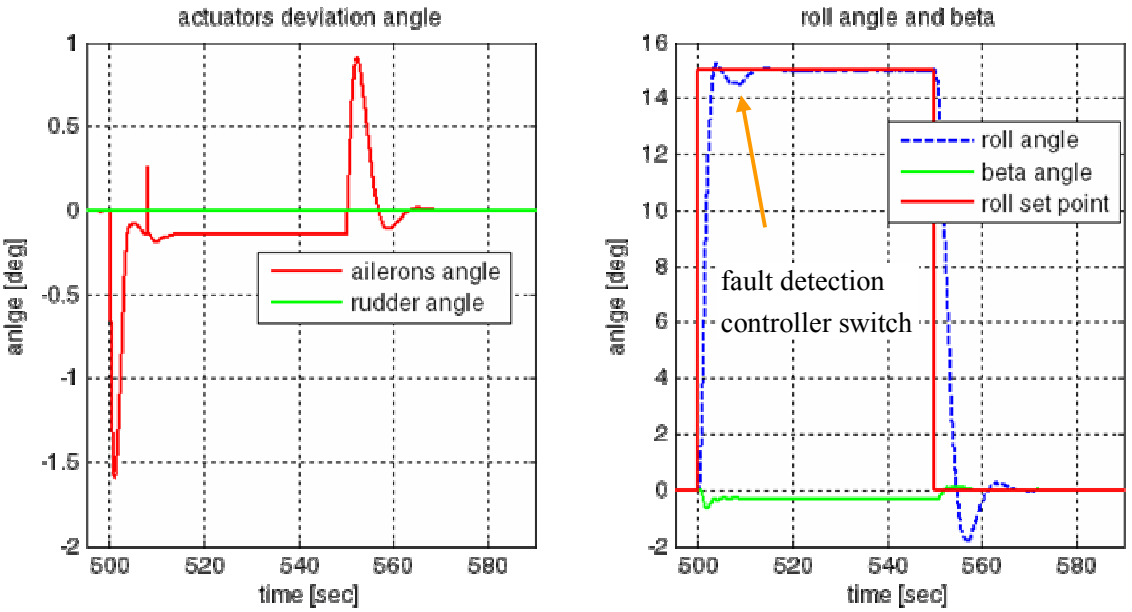

Figure 5-8: Results for the case of a power loss in the rudder. The contingency controller successfully reaches the desired roll angle.

The performance of the controller is satisfactory.

5.2.2.2. Actuator Stuck At a Fixed Deviation Angle

The last simulation presented is the "fixed rudder" scenario. The simulation is the same as the one shown in 4.4.3.2 in which the rudder actuator gets stuck at a fixed deviation angle while performing a turn maneuver. It should be remembered that the amount of time it takes for the fault detection system to identify the fault in this scenario is higher: 20sec. The results are shown in Figure 5-9. It can be seen that after the failure is detected the contingency controller successfully reaches the zero degrees roll angle. It can be also noticed in this graph that the roll angle achieved by the aircraft before the fault is detected is closely similar to the set point. This fact makes it difficult for the fault detection system to notice the failure, thus increasing the detection time.

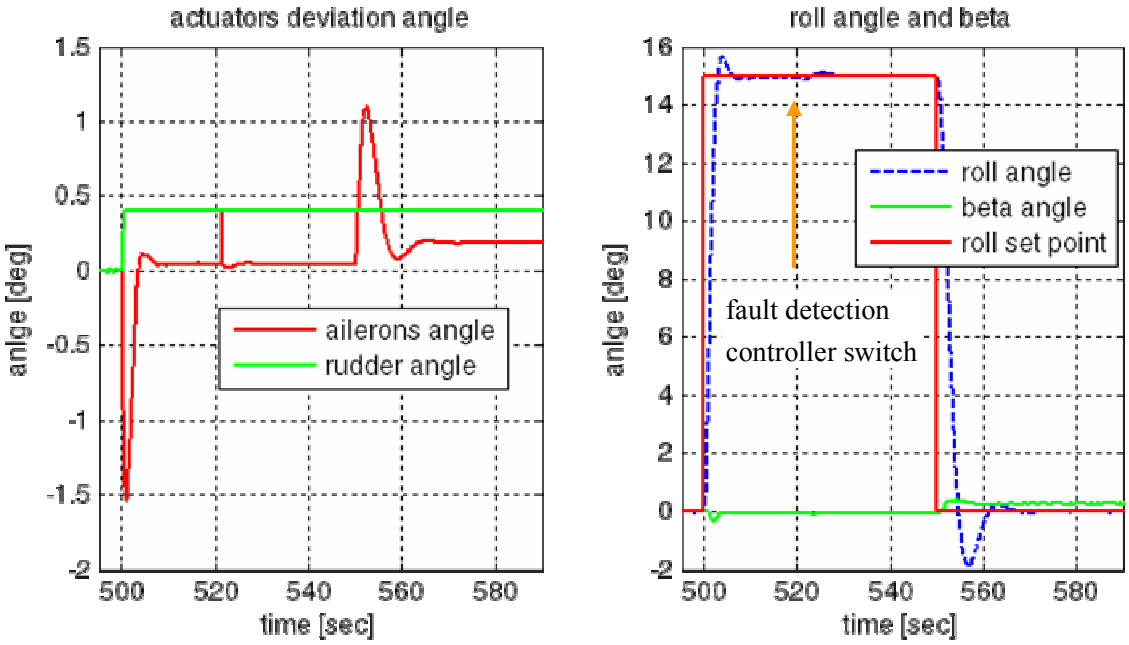

Figure 5-9: Results for the case when the rudder is stuck at a fixed angle due to a mechanical problem. The contingency controller successfully reaches the desired bank angle.

6. Conclusions

In this work the possibility of developing a contingency control strategy for certain faults in a UAV was studied, with the requirement that no additional hardware could be used. In section 3 the controller for keeping the aircraft in a straight level flight and the controller to perform a coordinated turn were developed and tested in an accurate Simulink model of a UAV that was available for this work. Both controllers were designed using the LQG theory, proving a suitable performance for each task. This section established the basic simulation platform for the following two.

In section 4 the fault detection and identification problem was tackled. First the basic background knowledge was presented and then applied to derive a possible solution. The unknown input observer was used in a dedicated observer scheme to detect and identify the faults in the ailerons and rudder of the plant.

Finally, in section 5 two controllers were derived to be used in each of the fault scenarios studied. Combining these controllers with the fault detection and identification system a contingency control system was proposed. The aim of this strategy is to provide the UAV with the ability of regaining a straight level flight and perform a turn even in the presence of a fault. Two fault scenarios for the ailerons and rudder were simulated. The first one comprises each actuator stuck at 0° deviation. In the second the actuators are stuck but at a deviation angle different than 0°.

The contingency system determined proved to fulfill the objectives when any of the two actuators remains stuck at 0°. First, the system was able to detect the fault and identify which of the actuators was failing. It took the system less than 1 sec to recognize this fault in the ailerons after a turning maneuver started, and about 3 sec when the fault occurred in the rudder. Then, for both cases, each contingency controller allowed the UAV to successfully perform the basic maneuvers even in the presence of the fault, taking the aircraft to a bank angle desired or regain a straight level flight.

For the second fault scenario, that is, the actuators stuck at a certain deviation angle, the results were acceptable but as it could obviously be predicted, not as favorable. When the fault occurred in the ailerons, it was detected and identified after about 3.5 sec. If the ailerons are stuck at an average deviation angle (or lower), the contingency controller proved to be efficient and again a turning maneuver and straight level flights could be achieved. However in the case of an important deviation angle, this is not fast enough to avoid the UAV from reaching a dangerous high bank angle. At this point the contingency controller is not capable of performing its task.

It is important to note that the fault here considered is so severe that it is highly probable that no controller could provide stability if the deviation angle of the ailerons is high, even when the fault was immediately detected. In the case of the rudder stuck at a certain deviation angle the performance of the fault detection and identification system was

considerably high: about 20 seconds. Nevertheless once the fault was detected, the contingency controller could successfully achieve its task.

It can be concluded that a contingency strategy is feasible and that this contingency system can provide a much higher degree of security to the aircraft. Moreover, with a more exhaustive tuning of the observers and controllers an even better performance than the one presented here should be achieved.

7. References

[1] U.S. Air Force Fact Sheet: MQ-1 PREDATOR.. U.S. airforce web page. (May 2008)

http://www.af.mil/library/

[2] Sojka. Uav Tactical Reconnaissance System. VTÚLaPVO web page. (2005)

http://www.vtul.cz/index.php?page=sojka&lang=en

[3] Transportation Safety Board of Canada Information Strategies and Analysis Directorate. Minister of Public Works and Government Services Canada 2001.

http://www.bst.gc.ca/en/stats/air/2000/statssummaryair00.asp?print_view=1

[4] Brian L. Stevens, Frank L. Lewis. Aircraft Control and Simulation. Wiley-Interscience Publication (1992). ISBN 0-471-61397-5

[5] Yong-Yan Cao, You-Xian Sun, and Wei-Jie Mao (1996). A New Necessary and Sufficient Condition for Static Output Feedback Stabilizability and Comments on "Stabilization via Static Output Feedback". IEEE Transactions On Automatic Control, Vol. 43, No. 8

[6] A. Astolfi and P. Colaneri. Static output feedback stabilization of linear and nonlinear systems. Proceedings of the 39th IEEE Conference on Decision-and Control Sydney, Australia December, 2000.

[7] N. J. Higham. The Matrix Computation Toolbox. Version 1.2 released in September 2002.

http://www.ma.man.ac.uk/~higham/mctoolbox.

[8] Katsuhiko Ogata. Modern Control Engineering. Pearson US Imports & PHIPEs; 3rd Ed edition (27 Aug 1996). ISBN-10: 0132273071

[9] Karl J Åström, Bjorn Wittenmark. Computer Controlled Systems: Theory and Design. Prentice-Hall (Jun 1984). ISBN-13: 978-0131643192.

[10] Marcin Witczak. Identification and fault detection of non-linear dynamic systems. Institute of Control and Computation Engineering (2003). ISBN 83-88317-65-2

[11] Jie Chen, R.J. Patton (1998). Robust Model-Based Fault Diagnosis for Dynamic Systems. ISBN-10: 0792384113. 1 edition (December 31, 1998)

[12] Isermann, R. and Ballé, P. (1997). Trends in the application of model-based fault detection and diagnosis of technical processes. Control Engineering Practice, 5(5):709-719.

[13] Chow EY, Willsky AS (1984). Analytical redundancy and the design of robust failure detection systems. IEEE Transactionson Automatic Control 1984; 29:603–614.

[14] Chen J. and Patton R. J. (1999): Robust Model-based Fault Diagnosis for Dynamic Systems. London: Kluwer Academic Publishers.

[15] Patton and Chen, (1994) A review of parity space approaches to fault diagnosis for aerospace systems. AIAA J. of Guidance, Contr. & Dynamics, 17(2):278-285.

[16] Zhong, M., Ding, S., Lam, J., & Wang, H. (2003). An LMI approach to design robust fault detection filter for uncertain LTI systems.Automatica, 39, 543–550.

[17] R. N. Clark, D. C. Fosth, and V. M. Walton (1975). Detecting instrument malfunctions in control systems. IEEE Trans. on Aeroesp. and Elec. Sys., AES-11(4):465-473, July 1975.

[18] Kowalczuk Z. and Gunawickrama K. (2000): Leak detection and isolation for tranmission pipelines via non-linear state estimation. – Proc. IFAC Symp.: Fault Detection, Supervision and Safety of Technical Processes: SAFEPROCESS 2000, Budapest, Hungary.

[19] Michael Bask (2005): Dynamic Threshold Generators for Robust Fault Detection. Luleå University of Technology, Department of Computer Science and Electrical Engineering, Control Engineering Group. ISSN: 1402-1544.