

# Fault Tolerant Control (FTC) Scheme for Unmanned Aerial Vehicles (UAV)

M. Eslamdoust

P.G Scholar  
Department of Aerospace  
Engineering, Shahid Beheshti  
Univ. Iran

A.R. Toloee

Assistant Professor  
Department of Aerospace  
Engineering, Shahid Beheshti  
Univ. Iran

A.R. Vali

Advisor Professor

## ABSTRACT

The main focus of this paper is fault-tolerant control systems (FTCSs) for unmanned aerial vehicles (UAVs). The goals are to develop an Automatic-Flight Control System (AFCS)[8], based on fault detection and isolation (FDI) and a reconfiguration mechanism to accommodate them [7]. This paper describes a design approach of Model Predictive Control (MPC) with a linear internal model to achieve a level of reconfiguration in a generic Uninhabited Aerial Vehicle (UAV)[10][11]. This paper is based on fault existence in one of elevator surfaces and the implementation of two separate controllers to overcome these failures [9]. In the end the results will be compared with each other.

## General Terms

Aerospace, Fault tolerant control system

## Keywords

Fault Tolerant Control, Model Predictive Control, Reconfigurable Control, Fault Detection Identification.

## 1. INTRODUCTION

There is an increasing demand for safety and performance requirements in modern control systems. Faults occur and conventional control systems may not be able to adapt, resulting in reduced performance or even instabilities. Safety-critical applications such as aircrafts, spacecrafts, nuclear power plants, chemical plants, etc. must have capabilities beyond conventional control systems, where the cost and consequences of malfunctions are too great. New control systems must be able to preserve stable control when faults occur in the system. Technology improvements have enabled modern autopilots that decrease the pilots' workload and have increased the airline industries safer. In UAVs, The absence of the pilot and therefore no compensation by him, need to reconfigure the control and autopilot systems. In this paper, we examine the left elevator failure and compensate it by changing the angle of the deviation in the right elevator.

Fault tolerant control was introduced in 1980. An article named "Fault detection and isolation" was published in 1988 by Honeywell Company. Adaptive fault tolerant control was reported in 2007 by "Reding and co-workers" and fault tolerant control thesis using MPC was presented by Jose Duarte Pereira in 2007.

## 2. AIRCRAFT MODEL

This paper is based on the 1/4 scale Piper PA 18 Super Cub aircraft, this model has developed in [1]. The basic technical characteristics of this scale airplane are:

- Speed:min.70 Km/h, max.120Km/h
- Engine: Quadra 50 cc

- Power: 3.4 KW
- Wing span: 2.7 m
- Overall length: 1.72 m
- Weight: 10.5 Kg

The aircraft motion is expressed by defining the following two frames. The orientation of the navigation frame is North, East, Down ( $x_n y_n z_n$ ). This frame

Does not move and is attached to the earth's local tangent plane. When the plane is on the ground the navigation frame is initialized by the position of the airplane's centre of mass. In body frame, the positive x axis of the body frame points forward along the aircraft's longitudinal axis, the positive y axis is directed along the right wing, the positive z axis is normal to the x and y axes, pointing downward. The origin Ob is located at the aircraft's centre of mass. Throughout simulations, the aircraft trajectory has to be analysed in the earth-based coordinates. To obtain the transformation from the body frame, the earth-fixed frame is used. Model inputs can be seen in the following. However, in this article we are going to only control the elevator.

Table 1: Inputs Model

| Inputs   | Description         | Min    | Max    |
|----------|---------------------|--------|--------|
| $\eta_e$ | Elevator deflection | -0.305 | +0.305 |
| $\eta_a$ | Aileron deflection  | -0.349 | +0.349 |
| $\eta_r$ | Rudder deflection   | -0.523 | +0.523 |
| $\eta_f$ | Flaps deflection    | 0      | +0.785 |
| $T_H$    | Throttle            | 0      | 1      |

The simulation outputs can be seen in the following:

Table 2: Outputs Model

| Outputs  | Description                 | Min |
|----------|-----------------------------|-----|
| $\theta$ | Euler pitch angle           | rad |
| U        | Velocity in the x body axis | m/s |
| V        | Velocity in the y body axis | m/s |
| W        | Velocity in the z body axis | m/s |
| X        | Position x                  | m   |
| Y        | Position Y                  | m   |
| Z        | Position Z                  | m   |

In this aircraft model, all control surfaces have the same settling time ( $t_s = 0.05s$ ).

They have the same characteristics; the actuators are represented by the following first order system:

$$G(s) = \frac{60}{s+60}$$

### 3. FAULT

#### 3.1. Partial Power Loss [2]

The Partial Power Loss causes settling time increasing, up to the maximum of 4 seconds, and  $t_s = 4$  shows the full power loss. In order to introduce Partial Power Loss into the actuator state of spaces:

$$\dot{\eta}(t) = A_{act}\eta(t) + B_{act}\eta^*(t)$$

And also:

$$A_{act} = -B_{act} = \begin{cases} d_{i,j} = a_i(t) & \text{if } i = j \\ d_{i,j} = 0 & \text{if } i \neq j \\ \forall i, j \in \{0, \dots, 4\} \end{cases}$$

And  $a_i(t)$  is defined in the following:

$$a_i(t) = -60 + \sigma_i(t) \left( 60 + \frac{3}{t_s} \right)$$

$$\sigma(t) = [\sigma_1(t), \sigma_2(t), \sigma_3(t), \sigma_4(t)]$$

$$\sigma_i(t) = \begin{cases} 1 & \text{if the } i\text{th actuator fails at } t_j \\ 0 & \text{otherwise} \end{cases}$$

Where  $t_s$  is the fault actuator settling time, for simulation purposes  $\sigma_i$  is a step signal, which reaches to 1 at  $t_j$  actuator failure time.

#### 3.2. Total Power Loss [2][27]

While a UAV goes up continually, both of elevator surfaces fails and stick at a negative deflection (positive pitch moment), there are no other control strategies to recover these kinds of failures. That's why elevators and ailerons are made in Pairs. This was accomplished in Simulink by splitting the actuator signal and then recombining it again before sending it to the aerodynamic model. For simulation purposes only, the used value to define a specific fault deflection is related to one of the pair control surfaces in both aileron and elevator. In the case of full power loss, the control surfaces are regarded to be stuck at a fixed angle; the actuation command signals are then overridden by the desired deflection and  $\eta(t)$  can be expressed as:

$$\eta_i(t) = \eta_i^*(t) + \frac{1}{2}\sigma_i(t) \left( \eta_{fault} - \eta_i^*(t) \right) \quad i = 1,2$$

$$\eta_i(t) = \eta_i^*(t) + \sigma_i(t) \left( \eta_{fault} - \eta_i^*(t) \right) \quad i = 3,4$$

Where  $\sigma_i$  signal is already defined,  $\eta^*(t)$  is the command signal, and  $\eta_{fault}$  fault is the position or deflection angle, which the control surfaces remains when a total power loss is simulated. Finally  $i=1, 2$  stands for aileron and elevator command signals and  $i = 3, 4$  for rudder and flaps. Elevator failure causes great changes in longitudinal velocity  $U$  and in pitch angle  $\theta$  either. In this paper a fault is applied to the left elevator at 50 seconds and it is expected of the right one to

compensate it. The compensation by both controllers will be checked and the results will be compared.

### 4. FAULT-TOLERANT CONTROL SYSTEM [16]

A fault-tolerant control system is able to control the system with satisfactory Performance even if one or several faults, or more critically, one or several failures occur in this system. Fault-tolerant control systems may be regrouped into two main families: passive fault tolerant controllers and active fault-tolerant controllers. An active fault-tolerant controller usually contains a separate module: a fault election and isolation (FDI) system that monitors the health of the aircraft. The FDI system informs a supervision module of the seriousness of the fault/failure or damage. Based thereon, the supervision module may decide to reconfigure the flight controllers, the guidance system, and the navigation system [3].

We propose using the model-following the MPC controller in this application for several reasons. As stated above, MPC is a good framework for fault-tolerant control, as many kinds of aircraft failures can be handled online in an adaptive fashion via modifications to the internal model. The achievable performance of an aircraft will often be reduced after a failure. This can be managed by MPC through changes to the objective functions or through the use of a multi-objective formulation as discussed in, (for example Kerrigan and Maciejowski, 2002). Finally, the loss of some actuators will often require the remaining controls to be driven to their limits, requiring any fault tolerant scheme to deal with actuator constraints which can be handled naturally by MPC [4]. Although FDI system is not implemented here, the control strategy described in Control Strategy section depends on the existence of a fault detection and identification system. The main role of this system is to determine which surface suffers the failure, the deflection angle and then select the most adequate controller, updating at the same time its input and output bounds (constraints).[2]

### 5. MODEL PREDICTIVE CONTROL [10][11]

The MPC Toolbox is a Powerful toolbox for Model Predictive Control simulation that makes it possible to design, analyses and implements the desired control system. The MPC Toolbox design generates a discrete time controller, which takes action at regularly spaced. These sampling instants are the times at which the controller acts.

To calculate its next move  $u_k$  the controller operates in two phases:

**Estimation:** In order to make an intelligent move, the controller needs to know the current state. This includes the true value of the controlled variable,  $\bar{y}_k$ , and any internal variables that influence the future trend,  $\bar{y}_{k+1}, \dots, \bar{y}_{k+P}$ . To accomplish this, the controller uses all past and current measurements and the models  $u \rightarrow \bar{y}$ ,  $d \rightarrow \bar{y}$ ,  $w \rightarrow \bar{y}$ , and  $z \rightarrow y$ . For details, see "Prediction" and "State Estimation".

**Optimization:** Values of set points, measured disturbances, and constraints are specified over a finite horizon of future sampling instants,  $k+1, k+2, \dots, k+P$ , where  $P$  (a finite integer  $\geq 1$ ) is the prediction horizon. The controller computes  $M$  moves  $u_k, u_{k+1}, \dots, u_{k+m-1}$ , where  $M$  ( $\geq 1, \leq P$ ) is the control horizon. After that the controller sends move  $u_k$  to the plant. The plant operates with this constant input until the next sampling instant,  $\Delta t$  time units later. The controller obtains

new measurements, and totally revises its parameters, this cycle repeats indefinitely. As predictions are made during the optimization stage, periodic measurement feedback allows the controller to correct this error and for unexpected disturbances. Most designers choose P and M such that controller performance is insensitive to small adjustments in these horizons [2], [5].

Prediction Model and Optimization Problem are Solved According to The proposed model in reference [5].

The MPC Toolbox design is done in five steps:

- Launching the control and estimation tools manager
- Constructing specifications for the equilibrium operating point
- Finding operating point
- Linearizing the plant
- Building the MPC controller

After defining the horizons, constraints and weight tuning, the MPC Toolbox is ready to be used [5].

## 6. FAULT-TOLERANT CONTROL STRATEGY

Whenever the left elevator is broken down, and the right elevator can compensate it by more deflection, this control is a kind of Fault-Tolerant control. At this moment the aircraft have to keep the basic functionality, while no additional hardware or redundancy is available in the damaged aircraft. Our strategy is to implement the Fault-Tolerant control to development the MPC controllers. The implemented controller controls mainly the aircraft velocity and pitch angle  $\theta$ , during the elevator failure tests. To compare the results, a longitudinal classic controller (the PI controller) according to the reference [6] is used in each test. The MPC controller Linearizing is done around the nominal point:

- Throttle: 6.24%
- Aileron deflection  $\eta_a$ : 0.0 rad
- Elevator deflection  $\eta_e$ : -0.0285 rad
- Rudder deflection  $\eta_r$ : 0.0 rad
- Flaps deflection  $\eta_f$ : 0.0 rad
- Velocity: 21.156 m/s
- Pitch  $\theta = 5.6 * 10^{-5}$  rad

To Implement the Longitudinal the MPC controller, the input weights are applied as the following:

- Input weights  $w^u = [0]$
- Input Rate weights  $w^{\Delta u} = [0.1]$
- State variables weights  $w^y = [10]$
- Sample time (Interval) = 0.02 seconds
- Prediction Horizon P = 10 (Interval times)
- Control Horizon M=2 (Interval times)
- Rate  $\Delta u_{min} = -1$
- Rate  $\Delta u_{max} = 1$

## 7. FAULT SIMULATION RESULTS

In this article, the right elevator is supposed to be corrected and has been shown by  $n_e$ . The right elevator deflection angle is determined by the PI or the MPC controllers. The elevator fault is supposed to be happening on the left elevator and then it will be frozen in a constant angle and doesn't move any more.

In order to consider the PI and the MPC controller's abilities, a fault scenario is performed at 50th second. This fault is applied to the right elevator and the aim is to study the right elevator behaviors.

### 7.1. Reducing altitude maneuver

According to the following illustration, in a reducing altitude maneuver with the reference value of  $\theta = 0.1$  rad, when the fault is applied to the left elevator and it is stuck at 33% of its deflection limits, the MPC controller is capable to follow the reference value of pitch angle while the PI controller has 40% error.

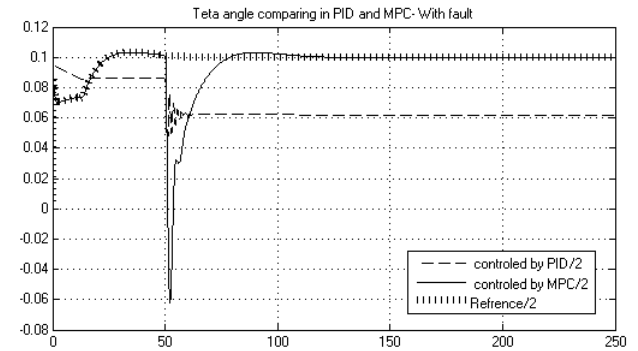


Fig 1: The MPC pitch angle compared with the PI controller

And the right and left elevator deflections in the PI and the MPC controllers are illustrated in the following:

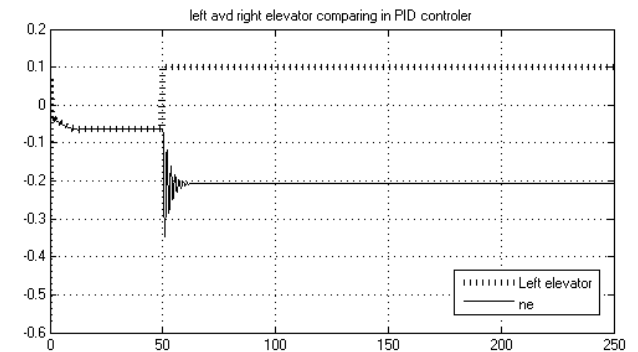


Fig 2: Right elevator deflection by PI controller

As shown, the right elevator deflection fault in the MPC controller is less than 1% error but it is 14% in the PI controller.

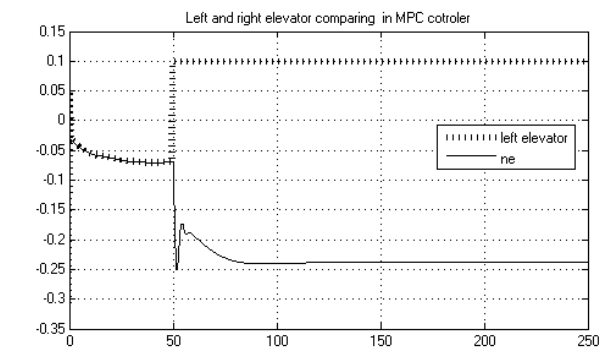
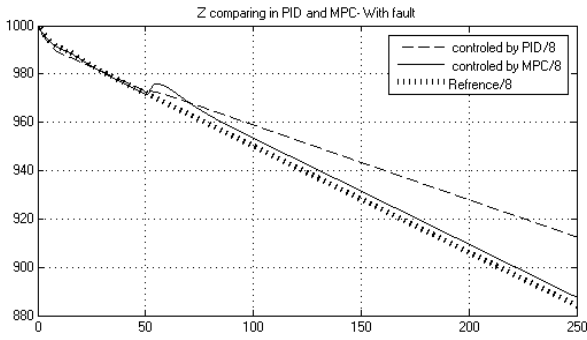


Fig 3: Right elevator deflection by MPC controller

The decreasing altitude diagram is illustrated in the following:

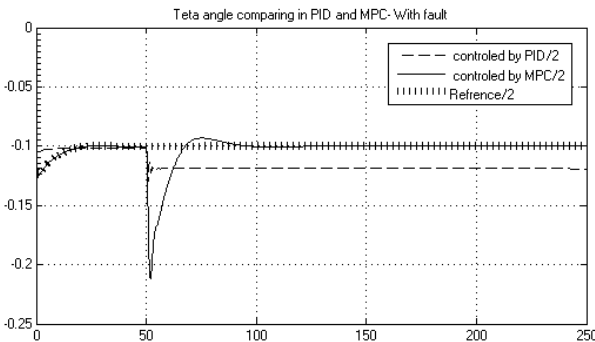


**Fig 4: The MPC decreasing altitude compared with the PI controller**

As shown, the MPC controller can pursue the altitude reference value with 0.35% error, but this error in the PI controller is 4.08%. By increasing the fault degree of the left elevator, the MPC controller performance to compensate it will become feebler. Whenever the left elevator is stuck at 65% of its deflection limit, the results of these two controllers are the same.

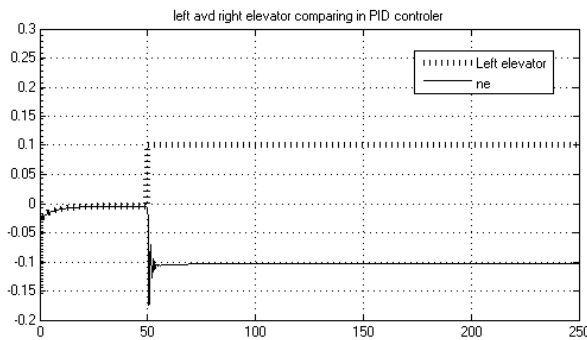
### 7.2. Increasing altitude maneuver

In a increasing altitude maneuver, with the reference value of  $\theta = -0.1$  rad when the left elevator is stuck at 33% of its deflection limit, the MPC controller is capable to follow the reference value of pitch angle while the PI controller has 19% error.



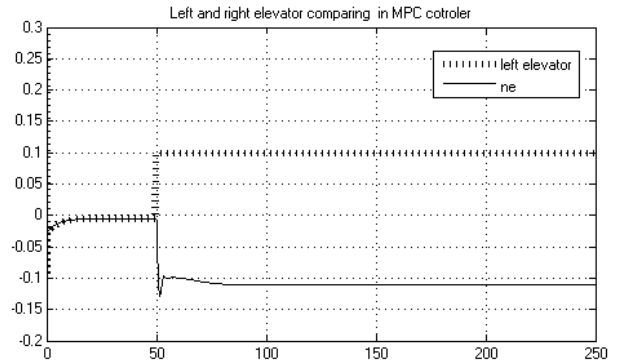
**Fig 5: The MPC pitch angle compared with the PI controller**

And also the right and left elevator deflections in the PI and the MPC controllers are illustrated in the following:



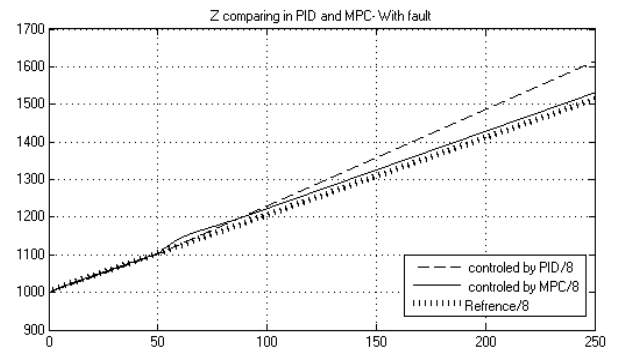
**Fig 6: Right elevator deflection by PI controller**

As you can see, the right elevator deflection fault in the MPC controller is approximately zero, but it has 4% error in the PI controller.



**Fig 7: Right elevator deflection by MPC controller**

As shown in the next diagram, the MPC controller can pursue the altitude reference value with 1% error, but this error in the PI controller is 7.7%. The increasing altitude diagram is illustrated in the following:



**Fig 8: The MPC increasing altitude compared with the PI controller**

By increasing the left elevator fault even if it has stuck at 100% of its deflection limit, the MPC controller performance is acceptable.

## 8. CONCLUSION

The following results are gained in 300 seconds time duration on the left elevator for decreasing altitude maneuver.

**Table 3: The output fault comparing**

| 33% left elevator fault, for reducing altitude maneuver |                              |                           |                                     |
|---|------------------------------|---------------------------|-------------------------------------|
| Controller  | Pitch angle fault percentage | altitude fault percentage | The right elevator fault percentage |
| MPC   | $\cong 0\%$                  | 0.35%                     | $\cong 1\%$                         |
| PI  | -40%                         | 4.06%                     | -14%                                |
| 50% left elevator fault, for reducing altitude maneuver |                              |                           |                                     |
| Controller  | Pitch angle fault percentage | altitude fault percentage | The right elevator fault percentage |

|  |                              |                           |                                     |
|--|------------------------------|---------------------------|-------------------------------------|
| MPC  | -6.6%                        | 1.16%                     | -1.04%                              |
| PI   | -45.5%                       | 5.27%                     | -12.7%                              |
| <b>65% left elevator fault, for reducing altitude maneuver</b> |                              |                           |                                     |
| Controller   | Pitch angle fault percentage | altitude fault percentage | The right elevator fault percentage |
| MPC  | -51.75%                      | 6.93%                     | 13%                                 |
| PI   | -62.75%                      | 8.06%                     | 16%                                 |

As shown, the MPC controllers are idealistic as fault tolerant controllers for elevator faults in case of decreasing maneuver up to 50% of their deflection limits and their performance between 50% up to 65% are acceptable.

The following results are gained in 300 seconds time duration on the left elevator for increasing altitude maneuver.

**Table 4: The output fault comparing**

|   |                              |                           |                                     |
|---|------------------------------|---------------------------|-------------------------------------|
| <b>33% left elevator fault, for increasing altitude maneuver</b>  |                              |                           |                                     |
| Controller  | Pitch angle fault percentage | altitude fault percentage | The right elevator fault percentage |
| MPC   | $\cong 0\%$                  | 1%                        | $\cong 0\%$                         |
| PI  | 19%                          | 7.7%                      | -4%                                 |
| <b>50% left elevator fault, for increasing altitude maneuver</b>  |                              |                           |                                     |
| Controller  | Pitch angle fault percentage | altitude fault percentage | The right elevator fault percentage |
| MPC   | $\cong 0\%$                  | 1.68%                     | -1%                                 |
| PI  | 27.7%                        | 11.7%                     | -5.6%                               |
| <b>65% left elevator fault, for increasing altitude maneuver</b>  |                              |                           |                                     |
| Controller  | Pitch angle fault percentage | altitude fault percentage | The right elevator fault percentage |
| MPC   | $\cong 0\%$                  | 2.26%                     | -2.3%                               |
| PI  | 35.4%                        | 15.5%                     | -7.7%                               |
| <b>100% left elevator fault, for increasing altitude maneuver</b> |                              |                           |                                     |

|                 |                              |                           |                                     |
|-----------------|------------------------------|---------------------------|-------------------------------------|
| <b>maneuver</b> |                              |                           |                                     |
| Controller      | Pitch angle fault percentage | altitude fault percentage | The right elevator fault percentage |
| MPC             | 32%                          | 20%                       | -6.4%                               |
| PI              | 52%                          | 24%                       | -10.2%                              |

As shown, the MPC controllers are idealistic as fault tolerant controllers for elevator faults in case of increasing altitude maneuver up to 65% of their deflection limits and their performance up to 100% are acceptable.

### 8.1 Future scope

In this article longitudinal fault has been inspected and inspecting the longitudinal fault is suggested for the feature. Because of coupling effects between Rader and aileron, fault inspection is so important and finding a precise set point is needed.

### 9. REFERENCES

- [1] R. Rato L.M. e Neves da Silva. Piper pa 18 super cub 1/4 escala - modelo nao linear. Master's thesis, INESC, 1993.
- [2] Reconfigurable Flight Control using Model Predictive Control. José Duarte Pereira Gonçalves.2009
- [3] Fault-Tolerant Flight Control and Guidance Systems for a Small Unmanned Aerial Vehicle. Guillaume Jacques Joseph Ducard. 2007
- [4] MPC Fault-Tolerant Flight Control Case Study: Flight 1862 . Jan M. Maciejowski Colin N. Jones
- [5] Model Predictive Control Toolbox for use with Matlab
- [6] Fiuza Jose e Silva Leonardo, 2005.Piloto Automatico de uma Aeronave,IST, Lisboa.
- [7] Colin N. Jones. Reconfigurable flight control first year report. Technical report, Pembroke College, 2005.
- [8] M.M. Kale and A.J. Chipperfield. Reconfigurable flight control strategies using model predictive control. Proceedings of 2002 IEEE, 2002.
- [9] Evangelos Papadopoulos and C. Chasparis Georgios. Analysis and model-based control of ser-vomechanism with friction. In International Conference on Intelligent Robots and Systems (IROS),2002.
- [10] Maciejowski J.M. Modelling and predictive control: Enabling technologies for reconfiguration.Elsevier Science, 1999.
- [11] Huzmezan M. Maciejowski J.M. Reconfigurable flight control of a high incidence research model using predictive control. In UKACC International Conference CONTROL'98, 1998.