

11-2021

FLUTTER SUPPRESSION BY ACTIVE CONTROLLER OF A TWO-DIMENSIONAL WING WITH A FLAP

Abdallah Tarabulsi

Follow this and additional works at: https://scholarworks.uaeu.ac.ae/all_theses

 Part of the [Mechanical Engineering Commons](#)

United Arab Emirates University

College of Engineering

Department of Mechanical and Aerospace Engineering

FLUTTER SUPPRESSION BY ACTIVE CONTROLLER OF A TWO-
DIMENSIONAL WING WITH A FLAP

Abdallah Tarabulsi

This thesis is submitted in partial fulfilment of the requirements for the degree of
Master of Science in Mechanical Engineering

Under the Supervision of Professor Abdel-Hamid Ismail Mourad

November 2021

Declaration of Original Work

I, Abdallah Tarabulsi, the undersigned, a graduate student at the United Arab Emirates University (UAEU), and the author of this thesis entitled “*Flutter Suppression by Active Controller of a Two-Dimensional Wing with a Flap*”, hereby, solemnly declare that this thesis is my own original research work that has been done and prepared by me under the supervision of Prof. Abdel-Hamid Ismail Mourad, in the College of Engineering at UAEU. This work has not previously been presented or published or formed the basis for the award of any academic degree, diploma or a similar title at this or any other university. Any materials borrowed from other sources (whether published or unpublished) and relied upon or included in my thesis have been properly cited and acknowledged in accordance with appropriate academic conventions. I further declare that there is no potential conflict of interest with respect to the research, data collection, authorship, presentation and/or publication of this thesis.

Student's Signature: Abdalla Date: 25/11/2021

Copyright © 2021 Abdallah Tarabulsi
All Rights Reserved

Advisory Committee

1) Advisor: Prof. Abdel-Hamid Ismail Mourad

Title: Professor

Mechanical and Aerospace Engineering Department

College of Engineering

2) Co-advisor: Dr. Tariq Taha Darabseh

Title: Associate Professor

Aeronautical and Mechanical Engineering Departments

Jordan University of Science and Technology, Jordan

Approval of the Master Thesis


This Master Thesis is approved by the following Examining Committee Members:

- 1) Advisor (Committee Chair): Prof. Abdel-Hamid Ismail Mourad

Title: Professor

Mechanical and Aerospace Engineering Department

College of Engineering


Signature _____  _____ Date 14-12-2021

- 2) Member: Dr. Kassim Abdullah

Title: Associate Professor

Mechanical and Aerospace Engineering

College of Engineering

Signature _____  _____ Date 14-12-2021

- 3) Member (External Examiner): Dr. Daniil Iurchenko

Title: Associate Professor

School of Engineering and Physical Sciences

Institution: Heriot Watt University, UK

Signature Dr. E Elnajjar _____  _____ Date 14-12-2021

This Master Thesis is accepted by:

Dean of the College of Engineering: Professor James Klausner

Signature James F. Klausner Date 26/12/2021

Dean of the College of Graduate Studies: Professor Ali Al-Marzouqi

Signature Ali Hassan Date 26/12/2021

Copy ____ of ____

Abstract

Flutter is a divergent oscillation of an aeroelastic structure, and one of a family of aeroelastic instability phenomena, that results from the interaction of elastic and inertial forces of the structure with the surrounding aerodynamic forces. Airfoil Flutter is important due to its catastrophic effect on the durability and operational safety of the structure. Traditionally, flutter is prevented within an aircraft's flight envelope using passive approaches such as optimizing stiffness distribution, mass balancing, or modifying geometry during the design phase. Although these methods are effective but they led to heavier airfoil designs. On the other hand, active control methods allow for less weight and higher manoeuvring capabilities.

The main objective of this study is to investigate the potential effectiveness of using Model Predictive control MPC as an active control strategy to suppress flutter.

Lagrange's energy method and Theodorsen's unsteady aerodynamic theory were employed to derive the equations of motion of a typical 2D wing section with a flap. Using MATLAB[®], the air speed at which the flutter occurs for a specific wing's parameters were found to be 23.96 m/s, at a frequency of 6.12 Hz. A Linear Quadratic Gaussian compensator LQG was designed and simulated. MATLAB[®] was also used to design and simulate a discrete MPC using Laguerre orthonormal functions. The simulated results for states regulation and reference tracking tasks in the flutter airspeed region from both controllers were compared and discussed in terms of quantitative performance measures and performance indices.

The results showed that both LQG and MPC powerful in suppressing the flutter in addition to their effectiveness in tracking a reference input rapidly and accurately with zero steady state error. The superiority for the constrained MPC is manifested by results comparison. MPC were able to save more than 40% of the needed settling time for states regulation task. Furthermore, it performed the job with much less control energy indicated by the ISE and ISU indices. On top of that, the key advantage of MPC, which is the ability to perform real-time optimization with hard constraints on input variables, was confirmed.

Keywords: Flutter, Active control, AFS, Optimal control, Regulator, LQR, State observer, Kalman filter, LQG, MPC.

Title and Abstract (in Arabic)

تخميد الرفرفة باستخدام المتحكم النشط لجناح ثنائي الابعاد مع لوح تحكم

الملخص

تعرف ظاهرة الرفرفة (Flutter) على انها اهتزاز ذواتساع متزايد مع الزمن، تتأثر به الهياكل المرنة المعرضة لقوى الديناميكا الهوائية (Aerodynamic) وينتج عن التفاعل ما بين ثلاث قوى هي الديناميكا الهوائية، والمرونة (Elastic)، والقصور (Inertia). تعد الرفرفة من أكثر ظواهر عدم الاتزان (Instability) خطورة في مجال دراسة الهياكل المرنة (Aeroelasticity) المعرضة لقوى الديناميكا الهوائية مثل الاجنحة. وذلك بسبب التأثيرات الكارثية طويلة المدى على متانتها وقابليتها للاستخدام بشكل آمن. تقليدياً يتم تجنب الرفرفة عن طريق حل صناعة هياكل اكثر متانة، او اضافة اوزان لمناطق معينة في الهيكل. ولكن هذه الطرق ورغم فاعليتها تعد طرقاً غير مرغوبة لما يصاحبها من زيادة في وزن الهيكل. من ناحية اخرى تتميز طرق التحكم النشط (Active control methods) بخفة الوزن وزيادة قدرات المناورة.

إن الهدف الأساسي من هذه الدراسة هو دراسة واختبار إمكانية استخدام التحكم الاستشرافي (Model Predictive Control MPC) كمتحكم نشط لتخميد الرفرفة. لغرض الدراسة تم استخدام نموذج الجناح ثنائي الابعاد مع لوح التحكم، وتم اشتقاق المعادلات الديناميكية للنظام باستخدام طريقة لاغرانج (Lagrange's energy method)، واستخدمت نظرية ثيودورسين (Theodorsen unsteady aerodynamic theory) لتمثيل قوى الديناميكا الهوائية. ثم باستخدام برنامج MATLAB® تمت محاكاة النظام عند سرعات مختلفة، وحساب السرعة الحرجة للرفرفة (Flutter speed) لجناح اختبار محدد المواصفات، والتي وجدت عند 23.96 متر/ثانية مع تردد 6.12 هيرتز. تلا ذلك، وباستخدام MATLAB® أيضاً، تصميم ومحاكاة متحكم خطي تربيعي غاوسي (LQG)، ومتحكم استشرافي (MPC) لنفس الجناح، عند سرعات تقع ضمن نطاق الرفرفة. ولمقارنة النتائج استخدمت معايير ومؤشرات الأداء التحليلية مثل (ISE) و (ISU) و (Settling time).

أظهرت نتائج الدراسة ان كلا المتحكمين قادر على اخماد الرفرفة وتغيير زاوية لوح التحكم الى القيمة المطلوبة بسرعة ودقة. مع تفوق ملحوظ للمتحكم الاستشرافي (MPC)، حيث استطاع انجاز المهمة بوقت أقل بحوالي 40%، وباستهلاك اقل بشكل ملحوظ للطاقة، يظهر من

خلال مؤشرات (ISE) و (ISU). اضافة الى تميزه بإمكانية تحديد السرعة والزاوية القصوى لحركة لوح التحكم دون أن يؤثر ذلك على أداء النظام، وهي ميزة غير متوفرة لدى معظم المتحكمات النشطة.

مفاهيم البحث الرئيسية: الرفرفة، طرق التحكم النشط، نظرية التحكم الأمثل، المتحكم التريبيعي الخطي، مرشح كالمان، المتحكم التريبيعي الغاوسي، التحكم الاستشراقي.

Acknowledgements

My thanks are due to Prof. Abdel-Hamid Ismail Mourad, the main advisor for this work for his help, advice, and unlimited support. Special thanks are also due to Dr. Tariq Darabseh, who introduced me to the research subject, and whose encouragement, patience, and breadth knowledge made this thesis possible.

My Deepest gratitude and love are due to my beloved mother and father for their encouragement. I would also like to thank my wife Sarah, and daughters Noor, Huda and Leen for their patience and support during the period of this research.

Dedication

To my beloved parents and family

Table of Contents

Title	i
Declaration of Original Work	ii
Copyright	iii
Advisory Committee	iv
Approval of the Master Thesis	v
Abstract	vii
Title and Abstract (in Arabic)	viii
Acknowledgements	x
Dedication	xi
Table of Contents	xii
List of Tables	xiv
List of Figures	xv
List of Abbreviations	xvi
Chapter 1 : Introduction	1
1.1 Overview	1
1.2 Statement of the Problem	4
1.3 Relevant Literature.....	5
1.3.1 Active Flutter Suppression Methods	6
1.3.2 Aeroelastic Model Derivation	9
Chapter 2 : Theory and Methods.....	12
2.1 The Aeroelastic Model.....	12
2.2 The Aeroelastic System's Equations of Motion.....	13
2.2.1 Deriving the Equations of Motion Using Lagrange's Method.....	13
2.2.2 Representing the Unsteady Aerodynamic Forces	15
2.2.3 The Full System Equations in Dimensionless Form	24
2.2.4 Converting the System's Equations to State Space Representation.....	28
2.3 Open Loop Stability Analysis	30
2.4 Closed Loop System Design	31
2.4.1 System's Controllability and Observability	31
2.4.2 Linear Quadratic Regulator Optimal Controller	32
2.4.3 State Estimators and Linear Quadratic Gaussian Compensator	35
2.4.4 Model Predictive Control using Laguerre Functions	40
2.5 Discrete Time Kalman Filter.....	50
2.6 System's Performance Measures and Indices	50

Chapter 3 : Results and Discussion.....	52
3.1 Open Loop Analysis.....	53
3.2 Closed Loop Linear Quadratic Gaussian Compensator.....	56
3.3 Closed Loop Discrete Model Predictive Control using Laguerre Functions.....	62
3.4 Controllers Comparison.....	68
Chapter 4 : Conclusions and Future Work.....	70
4.1 Conclusions.....	70
4.2 Future Work.....	71
References.....	73
Appendices.....	79
Appendix A: Theodorsen's Functions.....	79
Appendix B: MATLAB® Codes.....	80

List of Tables

Table 1: System's numerical data for simulation.....	52
Table 2: The control surface (flap) actuator physical constraints	58
Table 3: Quantitative analysis for the regulator performance as the states weight Q change	58
Table 4: Quantitative analysis for the reference tracking performance as the control effort wight R changes.	59
Table 5: Quantitative analysis for MPC regulator performance with and without constraints	63
Table 6: Quantitative analysis for MPC reference tracking performance	66
Table 7: LQG and constrained MPC - Regulator - Quantitative comparison.....	69
Table 8: LQG and constrained MPC – Reference tracking - Quantitative comparison.....	69

List of Figures

Figure 1: A diagram of the forces interaction that causes flutter	1
Figure 2: NACA 0012 BACT wing	10
Figure 3: A typical two-dimensional airfoil with flab.	12
Figure 4: LQR block diagram	34
Figure 5: Closed loop state estimator (Observer) block diagram	36
Figure 6: LQG compensator block diagram.....	38
Figure 7: LQG compensator with integral action	39
Figure 8: MPC block diagram.....	40
Figure 9: How MPC works.	41
Figure 10: The calculation of the integral squared error ISE.....	51
Figure 11: Modal damping at air speed range 0 – 28 m/s.....	54
Figure 12: System's oscillation frequencies.	54
Figure 13: open loop system response to initial condition.....	55
Figure 14: Open loop step response at a speed that is less than the critical flutter speed	56
Figure 15: Simulink block diagram of the aeroelastic system with LQG controller and integrator	57
Figure 16: LQG closed loop response to initial disturbance – Displacement states	60
Figure 17: LQG closed loop response to initial disturbance – Rate states.....	60
Figure 18: LQG closed loop response to initial disturbance – Control signal	61
Figure 19: Step response results using LQG compensator with integral action	61
Figure 20: The control input and its rate of change - reference tracking case	62
Figure 21: MPC using Laguerre functions - Response to initial disturbance – Displacement States	64
Figure 22: MPC using Laguerre functions - Response to initial disturbance – Rate of change states.....	65
Figure 23: MPC – System's response to initial disturbance - Unconstrained control input and its rate of change.....	65
Figure 24: MPC – System's response to initial disturbance - Constrained control input and its rate of change.....	66
Figure 25: MPC– Reference tracking - Input and output simulated signals.....	67
Figure 26: MPC– Reference tracking - Unconstrained control input and its rate of change	67
Figure 27: MPC– Reference tracking - Constrained control input and its rate of change	68

List of Abbreviations

AFS	Active Flutter Suppression
LMIs	Linear Matrix Inequalities
LMPC	Model Predictive Control using Laguerre Functions
LQG	Linear Quadratic Gaussian
LQR	Linear Quadratic Control
LTI	Linear Time Invariant
MPC	Model Predictive Control

Chapter 1 : Introduction

1.1 Overview

When an elastic structure is exposed to an air stream, structural deformations may take place, these deformations lead to generating additional aerodynamic forces, which produce additional structural deformations, and so on. The interaction between the elastic, inertial, and aerodynamic forces is called aeroelasticity, and it is responsible of several undesirable phenomena that may reduce the structural fatigue life or lead to catastrophic failure.

Static divergence, flutter, limit cycle oscillations are examples of those phenomena. Static divergence is a phenomenon that results from the interaction between the aerodynamics and elasticity only (static aeroelasticity), while flutter and limit cycle oscillations are dynamic aeroelasticity phenomena that result from the interaction of all three forces as shown by the shaded area in Figure 1. Flutter is a dynamic instability that happens when three forces interact to generate a self-excited motion of a lifting surface (De Marqui et al., 2005; Hodges and Pierce, 2011).

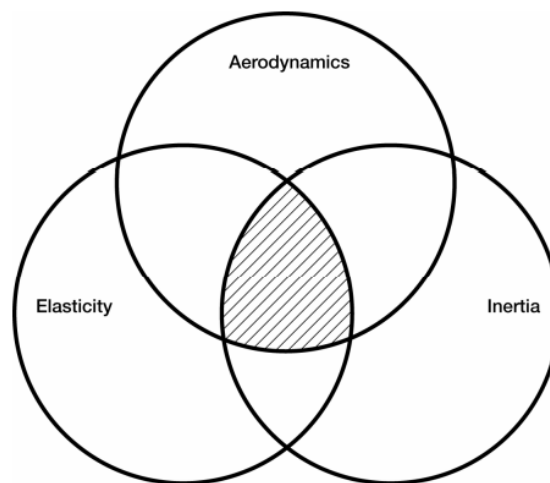


Figure 1: A diagram of the forces interaction that causes flutter (Ricketts, 1983)

Flutter is considered as one of the most important aeroelasticity phenomena. It is a dynamic aeroelastic instability, that happens when the structural damping changes from positive to negative due to the existence of aerodynamic forces at a specific speed known as the flutter speed. During the transition stage, two modes of vibration consolidate to the same frequency and achieve an aeroelastic resonance (De Marqui Junior et al., 2006; Kehoe, 1995).

There are several degrees of freedom that can occur in an airfoil during flutter, deemed 'flutter modes', two of which are dominant: first plunge mode (bending) and first pitch mode (torsion). Coupled damping occurs because of the airfoil structure and aerodynamic forces, and the damping ratio of the critical flutter mode may begin to decrease beyond some point with increasing airspeed. A flutter occurs when the damping ratio of the critical flutter mode reaches zero.

At the flutter speed the airframe structure undergoes a divergent sinusoidal oscillation which has the following characteristics (Akmeşe, 2006):

1. It is a self-excited oscillation; this means that once the vibration started, then no external action is required to maintain it. The system keeps getting more energy from the flow by itself.
2. Flutter starts at a certain airspeed and frequency called the flutter speed and flutter frequency.

At speeds above the flutter speed, the oscillation amplitude keeps increasing with time until a structural failure happens. The critical speed at which the flutter starts is a function of structural parameters like the structural damping, shape, stiffness, and mass distribution. It is also a function of flight parameters like the angle of attack, Mach number, the airspeed, and the altitude.

Airfoil flutter is typically prevented within an aircraft's flight envelope by optimizing stiffness distribution, mass balancing, or modifying geometry during the design phase. These are known as passive approaches for reducing airfoil flutter, which are effective but on the other hand led to heavier airfoil designs.

Passive approaches have included other techniques that can be found in literature such as the use of viscoelastic materials, these techniques have some advantages including inherent stability, and economical cost for fabrication and maintenance. Nevertheless a major drawback of these materials is the variation of its properties with temperature as presented in (Cunha-Filho et al., 2016).

The increase in flight speeds started to since World War II, flutter, among other aeroelastic phenomena, has become more significant. In addition, maximization of the modern aircraft performance which demands for extremely lightweight structures with less stiffness has increased the sensitivity to these aeroelastic problems. As a result, control technology has been brought into the field of aeroelasticity (De Marqui et al., 2005; Theis et al., 2016).

By applying appropriate control efforts to counteract flutter once it begins, one can further delay the onset of airfoil flutter to a higher airspeed without excessive structure modification and weight penalty. It is categorized as an active approach, which has been proven effective theoretically and experimentally in several leading research projects since the 1970s.

Active flutter suppression is employing a control surface the deflection of which is commanded by a calculated control law. The control law is the relationship between the motion of the main wing surface and the control surface deflection (York, 1980).

The field that studies the interactions of aeroelasticity with active control systems is called Aeroservoelasticity. Its primary goal is to adjust a system's aeroelastic behavior using calculated control forces (Tewari, 2015).

1.2 Statement of the Problem

Flutter is considered as one of the most important instability phenomena in aeroelasticity filed due to its catastrophic effect on the long-term durability and operational safety of the aircrafts wing. Traditionally passive solutions had been presented and used for many years, but they were not favorable due to the weight adding penalty which reduces the aircraft performance.

Active control methods allow for less weight and higher maneuvering capabilities. Although many active control methods have been studied and shown success in the past few decades, none of them has achieved operational status on any aircraft this is due to the fact that aircraft designers and operators are risk averse (Scott and Pado, 2000).

Nevertheless, with the recent improvements in the control systems hardware and software capabilities, and the increasing desire to increase flexibility and reduce the structural weight, AFS implementation could be closer than ever before (Livne, 2018; Marchetti et al., 2020).

This study is concerned with the active control methods, used to suppress wings flutter. Its main objective is to contribute to the efforts of finding the most capable and reliable active control strategy by investigating the potential effectiveness of using Model Predictive Control. This is achieved as following:

- Deriving a mathematical model of a typical two-dimensional wing section with three degrees of freedom, using Lagrange's energy method and Theodorsen's unsteady aerodynamic theory.
- Finding the flutter critical speed and the flutter frequency using by system's eigen value analysis for a specific experimental wing data.
- Validating the model by comparing the flutter speed and flutter frequency with an experimental work paper for the same wing.
- Designing a LQG compensator, then tuning and simulating it numerically using MATLAB[®] software to find the best performance parameters in for initial disturbance dissipation and reference input tracking, at a speed where the flutter is expected.
- Designing a discrete MPC controller using Laguerre functions with a Kalman filter, then tuning and simulating it to find the best performance.
- Comparing the simulated performance from both controllers analytically using performance indices.
- Discussing the results and concluding the research.

1.3 Relevant Literature

Formal beginning of active control methods can be traced back to the early 1970's, when the US Air Force launched the Load Alleviation and Mode Stabilization (LAMS) program. The improvements in this field during that period were facilitated by the advancements of optimal control theory.

In the 1980's and 1990's, the Aeroservoelasticity analysis and design efforts continued and boosted by the development of robust multivariable control theory (Glad and Ljung, 2018).

Active flutter suppression is based on using a calculated control law to command the deflection of a control surface (flap) to suppress the flutter. This control law is determined by applying methods from control systems theory (York, 1980).

1.3.1 Active Flutter Suppression Methods

Flutter suppression early works presented an approach that study the physics or mathematical structure of the flutter problem to find the mechanisms responsible for the flutter and try to suppress them. Among these physics approach is the “Aerodynamic Energy” that is the most presented in literature (Barker et al., 1999).

The aerodynamic energy method can be found in researches such as York (1980), where the fact that the stability is determined by the net work done per cycle by the forces acting on the system is employed. If the sign is positive, then the energy is being transferred from the surroundings to the system resulting in an unstable condition. The energy method is based upon increased energy dissipation near the flutter instability region by introducing the control flap to the system, where the added servo hinge torque is contributing to the rate of kinetic energy transferred to the system.

Although the physics approach has achieved major accomplishments, the development in general control systems theory has pushed it a side over time. Where a variety of active control law synthesis from classical and modern control theories have been developed and tested.

Classical control system design and analysis techniques using frequency domain is a trait in older works such as Horikawa and Dowell (1979), where the standard root locus technique had been used. The classical theory based on the Nyquist stability criterion is implemented in works such as Marretta and Marino (2007), Horikawa and Dowell (1979) to formulate a control law for SISO system. Although

the classic theory methods have successfully suppressed the flutter, Horikawa and Dowell (1979) had to neglect the inertia and damping effects of the trailing edge in addition to the structural and aerodynamic damping terms of the model, to keep the system at not more than fourth order.

In general, the classic design is working on varying the controller transfer function to achieve the desired closed-loop performance. This is indicated through the closed loop frequency response, or the location of the poles. For a large order system, by varying a limited number of constants in the controller transfer function, the location of a few of the closed loop poles could be varied, but not all of them. This is a major limitation of the classical design approach (Ashish, 2002).

Modern control theory techniques that are based on state-space modeling and analysis are more efficient in dealing with higher order, and multi-input, multi-output systems. These techniques have appeared in newer works. The Pole assignment with state feedback has been used in many papers such as De Marqui et al. (2005), and Karpel (1982). Linear quadratic regulator LQR optimal control method is another successful controller that has been used in Block et al. (1997), Garrard and Liebst (1985), Hopwood et al. (2019), and Olds (1997).

Uncertainty is inherent to the unsteady aerodynamic models. This fact must be taken in consideration while designing a successful controller. In general, when designing a linear system controller, robustness to modeling uncertainties can be achieved using high-gain feedback, but this deteriorates the response to high-frequency measurement noise. To reach a middle ground between robustness and noise rejection, linear feedback strategies such as LQG, where an optimal estimator is integrated with the LQR controller has been used in Bhoir and Singh (2004), Mahesh et al. (1981), and Sutherland (2010).

Controllers that are self-adaptive to changing flight conditions have also been developed and become of major interest as in Eversman and Roy (1997), Borglund and Kuttenukeuler (2002), in addition to gain-scheduling method presented in Barker et al. (1999).

Solving problems described by Linear Matrix Inequalities using convex optimization algorithms (LMIs) is one of the recent developments in active control field. This was investigated with the aim to design and simulate a robust control methodology as a flutter suppression control law by Silva and Lopes Júnior (2006).

Some Non-conventional techniques like fuzzy logic have also proofed success in suppressing flutter. Belo and Rocha JR (2001) simulated a rectangular wing aeroelastic structure and applied fuzzy logic using the method of Mamdani.

Kassem et al. (2020) designed an active dynamic vibration absorber by adding an active element that derives the mass to the classical mass-spring-damper system. Where a PI based hysteresis compensator controllers used feedback signal from the response of the aeroelastic system to generate the control law.

In the past few decades, the model predictive control has got much attention as an effective tool for the control of industrial systems. MPC is a real-time optimization strategy that computes an optimal control sequence every time step, based on the knowledge of the plant dynamics (a model) and the feedback information, in addition to a set of constraints (Boscariol et al., 2010).

Predictive control that has been around since early 1970's but due to its need of high computational power, it was limited to the industrial applications that are considered as slow dynamical systems such as chemical factories. However, with the recent massive technological improvements in controllers' and power electronics'

capabilities and speed, the model predictive control has received more attention as a useful tool for a wider range of application (Na, 2001; Pinheiro and Silveira, 2021).

MPC is a very powerful tool as it solves an optimization problem to find the optimal input trajectory every time step (real-time optimization), which gives it the ability to count for systems physical constraints in addition to any deviation in the measurements, which may happen due to unmeasured disturbances.

This study aims to investigate the potential effectiveness of discrete time MPC controller using Laguerre functions in flutter suppression application. Where MPC is expected to bring the advantages of real-time optimization to one on the most important Aeroservoelasticity applications.

1.3.2 Aeroelastic Model Derivation

Deriving an accurate aeroelastic model, based on modeling of the unsteady aerodynamic forces is considered as the main challenge in designing an active flutter suppression controller (Borglund and Kuttenukeuler, 2002).

Flutter can be formally defended as: a dynamic instability of a flight vehicle associated with the interaction of aerodynamic, elastic, and inertial forces. This definition implies that a quite good knowledge of the system's structural dynamic and aerodynamic properties is essential to investigate the subject (Hodges and Pierce, 2011).

For the structural model, Typical 2D section with three degrees of freedom model is the most common model in the literature related to flutter analysis, it has been used by NASA Langley Research Centre's in The Benchmark Active Control Technology (BACT) project (Waszak, 1996), in addition to many researches such as

York (1980), Hopwood et al. (2019), Conner et al. (1997), Edwards and Wieseman (2008), and Sutherland (2011).

The BACT is a rigid rectangular scaled wing with a NACA 0012 airfoil section, and a partial span trailing-edge control surface (Flab) shown in Figure 2. The model is equipped with linear accelerometers, which serve as the principal feedback control sensors. The wing is held by a pitch and plunge mechanism, which provides the necessary two degrees of freedom for flutter (Marretta and Marino, 2007).

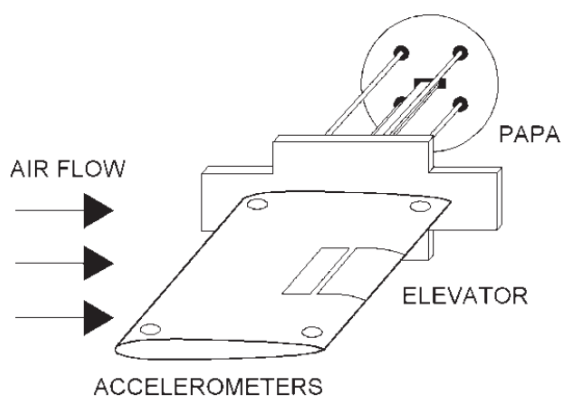


Figure 2: NACA 0012 BACT wing (Marretta and Marino, 2007)

Some other studies have used the elastic modelling of the wing and control surface, but this method results in a big number of states, so it is much more complicated, but gives more realistic results. Elastic modelling can be found where structural nonlinearities are involved such as Hoadley and Karpel (1991), Roger et al. (1975), Yehezky and Karpel (1996).

Moving to the aerodynamic forces, thin airfoil with steady incompressible aerodynamics had appeared in early studies such as Horikawa and Dowell (1979). Nevertheless, as reported in Abdelkefi et al. (2013), this model is unsatisfactory in its ability to accurately predict the flutter speed under incompressible flow compared to the unsteady aerodynamic model which can be found in most of the recent works such

as Hopwood et al. (2019), Borglund and Kuttenukeuler (2002), Conner et al. (1997), Sutherland (2011).

The unsteady aerodynamic model is obtained in most works using optimized rational function approximations, fitted to the frequency-domain aerodynamic data in the harmonic limit (Eversman and Tewari, 1991).

It can also be found using experimental model-based approach, where the flexible dynamics may be reliably predicted using system-identification techniques. The experimental aeroelastic model estimated using these techniques is in general of a low order, low complexity, and suitable for the model-based control design (Zeng et al., 2012).

The full system mathematical equation of motion of the typical 2D section can be obtained using either Newton's second law of motion, and the moment equation for a rigid body in planar motion as in York (1980) and Olds (1997). Or using Lagrange's equations and the principle of virtual work, which is used in (BACT) project, to formulate a model of the dynamic behavior of the Benchmark Active Controls Technology (wind-tunnel model) for application to design and analysis of flutter suppression controllers (Waszak, 1996). Lagrange's method has also been used in many other works such as De Marqui et al. (2005), and Sutherland (2011).

Chapter 2 : Theory and Methods

2.1 The Aeroelastic Model

In this study, the modal representation is used to set up a lifting surface (wing) flutter analysis as a linear set of ordinary differential equations that can be transformed into an eigenvalue problem to investigate the stability characteristics. To do so, a simple model is needed (Hodges and Pierce, 2011).

The typical two-dimensional airfoil with a control flab model is shown in Figure 3. This system was used by Theodorsen to develop his theory of unsteady aerodynamics, and it can be seen as section of a long and finite wing. The springs k_h, k_α represent the wing structural bending and torsional stiffness while dampers c_h, c_α represent the damping effect. The reference point is the elastic axis (Dimitriadis, 2017).

The distance b is the mid-cord distance, ab is the elastic access position measured from the mid-cord (positive aft of the mid-cord). bx_α is the distance to the center of gravity point from the elastic axis (positive aft the elastic axis).

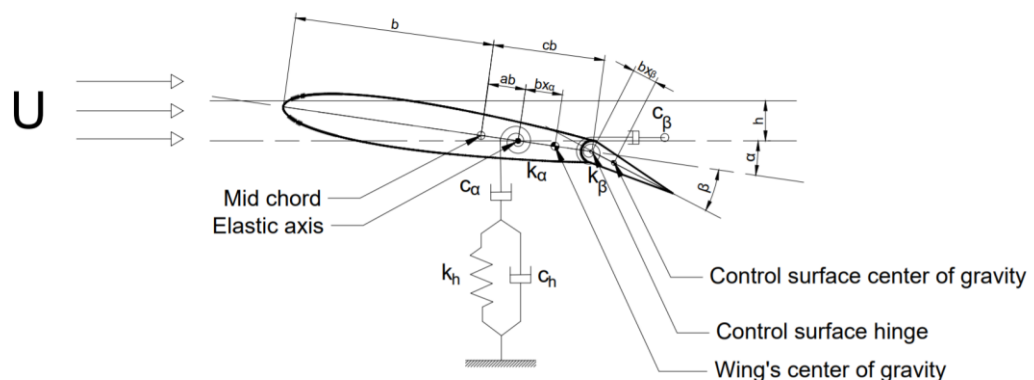


Figure 3: A typical two-dimensional airfoil with flab

The control flab hinge location is at a distance cb from the mid-cord (positive aft the mid-cord), and center of gravity for the flab is at a distance bx_β from the hinge line. The spring k_β and damper c_β represents the flab rotational stiffness and damping.

The setup is placed in a uniform, incompressible air flow of speed U and density ρ_∞ . The air flow generates aerodynamic lift and pitching moment about the elastic axis while the control flab is subjected to a torque about its hinge. In addition, the flab is equipped with an actuator that can apply a torque u about the control flap hinge line relative to the airfoil to control the system (Sutherland, 2008).

2.2 The Aeroelastic System's Equations of Motion

2.2.1 Deriving the Equations of Motion Using Lagrange's Method

For the development of the model, the generalized coordinates to represent the three degrees of freedom are the lateral position h (positive downward), the wing's pitch angle α (positive nose up) and the control flab deflection angle β (positive for a downward rotation).

The three degree of freedom equations of motion of the airfoil are derived using Lagrange's equation is generally expressed as follows:

$$\frac{d}{dt} \left(\frac{\partial KE}{\partial \dot{q}_i} \right) - \frac{\partial KE}{\partial q_i} + \frac{\partial D}{\partial \dot{q}_i} + \frac{\partial PE}{\partial q_i} = F_i \quad (2.1)$$

Where q_i are the generalized coordinates, and F_i is the generalized force associated with the generalized coordinates.

The total kinetic energy KE is the sum of the kinetic energies of the wing and the control surface, using the variables in Figure 3, it can be written as:

$$\begin{aligned}
KE = \frac{1}{2}m\dot{h}^2 + \frac{1}{2}I_\alpha\dot{\alpha}^2 + \frac{1}{2}I_\beta\dot{\beta}^2 + mx_\alpha b\dot{h}\dot{\alpha} + mx_\beta b\dot{h}\dot{\beta} \\
+ [(c-a)b^2mx_\beta + I_\beta]\dot{\alpha}\dot{\beta}
\end{aligned} \tag{2.2}$$

Where m is the total mass of the wing (with the control surface) per unit span, m_β is the mass of the control surface per unit span. I_α , I_β are respectively the mass moment of inertia of the wing (with the control surface) about the elastic axis per unit span, and the mass moment of inertia of the control surface about its hinge line.

Similarly, the total potential energy PE is:

$$PE = \frac{1}{2}k_h h^2 + \frac{1}{2}k_\alpha \alpha^2 + \frac{1}{2}k_\beta \beta^2 \tag{2.3}$$

The total energy dissipation D function (often characterized as a viscous force without friction) is:

$$D = \frac{1}{2}c_h \dot{h}^2 + \frac{1}{2}c_\alpha \dot{\alpha}^2 + \frac{1}{2}c_\beta \dot{\beta}^2 \tag{2.4}$$

Differentiating (2.2), (2.3), and (2.4) and substituting them in Lagrange's Equation (2.1), gives the following equations of motion:

$$m\ddot{h} + mx_\alpha b\ddot{\alpha} + mx_\beta b\ddot{\beta} + c_h\dot{h} + k_h h = L \tag{2.5}$$

$$I_\alpha\ddot{\alpha} + mx_\alpha b\dot{h} + [(c-a)b^2mx_\beta + I_\beta]\ddot{\beta} + c_\alpha\dot{\alpha} + k_\alpha\alpha = M_\alpha \tag{2.6}$$

$$mx_\beta b\dot{h} + [(c-a)b^2mx_\beta + I_\beta]\ddot{\alpha} + I_\beta\ddot{\beta} + c_\beta\dot{\beta} + k_\beta\beta = M_\beta \tag{2.7}$$

The three coupled equations of motion, in addition to the flap hinge torque (that is required to drive the flap and control the system) can be represented in matrix form as:

$$M_s \begin{Bmatrix} \ddot{h} \\ \ddot{\alpha} \\ \ddot{\beta} \end{Bmatrix} + D_s \begin{Bmatrix} \dot{h} \\ \dot{\alpha} \\ \dot{\beta} \end{Bmatrix} + K_s \begin{Bmatrix} h \\ \alpha \\ \beta \end{Bmatrix} = \begin{Bmatrix} L(t) \\ M_\alpha(t) \\ M_\beta(t) \end{Bmatrix} + L_c\{\beta_c\} \tag{2.8}$$

Where the structural inertia matrix M_s is:

$$M_s = \begin{bmatrix} m & S_\alpha & S_\beta \\ S_\alpha & I_\alpha & (c-a)bS_\beta + I_\beta \\ S_\beta & (c-a)bS_\beta + I_\beta & I_\beta \end{bmatrix} \quad (2.9)$$

$S_\alpha = mbx_\alpha$ and $S_\beta = mbx_\beta$ are the static mass moment of wing about the wing's elastic axis per unit span and the static mass moment of control surface about its hinge line per unit span.

The stiffness K_s , and damping D_s matrices are:

$$K_s = \begin{bmatrix} k_h & 0 & 0 \\ 0 & k_\alpha & 0 \\ 0 & 0 & k_\beta \end{bmatrix} \quad (2.10)$$

$$D_s = \begin{bmatrix} c_h & 0 & 0 \\ 0 & c_\alpha & 0 \\ 0 & 0 & c_\beta \end{bmatrix} \quad (2.11)$$

For simplicity, the flutter suppression system in this study does not include servo dynamics for the flap control, which means that the commanded flap angle β_c and the flap angle β are identical. The control load matrix L_c is which is required to represent the flap driving hinge torque is:

$$L_c = \begin{bmatrix} 0 \\ 0 \\ K_\beta \end{bmatrix} \quad (2.12)$$

2.2.2 Representing the Unsteady Aerodynamic Forces

The external forces acting on the airfoil are the unsteady lift L (assumed to act at the quarter chord position), the unsteady pitching moment M_α and the unsteady flap hinge moment M_β . These forces, are due to aerodynamics. They result from the distributed pressures applied to the surface of the wing (Sutherland, 2008; Waszak, 1996).

To find the full system dynamic equations, the aerodynamic forces must be represented in terms of the generalized coordinates (h , α , and β). The flow is unsteady due to two reasons, first is the unsteady motion of the wing with respect to the air. The second, is when the movement of the airfoil disrupts the flow, and shedding a vortex at the trailing edge that generates a downwash, which in turn changes the flow on the airfoil (Hodges and Pierce, 2011).

Assuming a thin section with infinite aspect ratio airfoil, and small angle of attack, under small oscillations in all vibration modes, and in incompressible flow, the unsteady aerodynamic forces can be approximated to depend linearly on the exciting structural motion, using Theodorsen's method (Fung, 2002; Tewari, 2015).

The air flow over the wing produces steady components of lift and pitching moment, as well as dynamic forces in reaction to the small fluctuations in the lifting surface motion (Hodges and Pierce, 2011).

For simplicity, it is assumed that the aerodynamic moments can be modeled as first-order lag (or circulatory) effects of the unsteady wake generated by the airfoil, as well as the non-circulatory contributions of the aerodynamics to inertia (known as the apparent mass effect), damping, and stiffness (Tewari, 2015).

Theodorsen unsteady aerodynamic theory shows that the aerodynamic forces acting on a typical section are due to two physical phenomena, the circulatory effect is due to the vorticity in the flow, and the non-circulatory effects, that does not depend on the frequency, it is generated when the wing motion has a non-zero acceleration, so a part of air surrounding the wing is carried with it, leading to inertial forces (due to the air finite mass) opposing its acceleration (Hodges and Pierce, 2011; Matter et al., 2018).

$$L(t) = L_c(t) + L_{nc}(t) \quad (2.13)$$

$$M_\alpha(t) = M_{\alpha,c}(t) + M_{\alpha,nc}(t) \quad (2.14)$$

$$M_\beta(t) = M_{\beta,c}(t) + M_{\beta,nc}(t) \quad (2.15)$$

The non-circulatory part is represented as:

$$L_{nc}(t) = -\pi\rho_\infty b^2 \left\{ \ddot{h} + U\dot{\alpha} - ba\ddot{\alpha} - \frac{b}{\pi} T_1 \ddot{\beta} - \frac{U}{\pi} T_4 \dot{\beta} \right\} \quad (2.16)$$

$$\begin{aligned} M_{\alpha,nc}(t) = \pi\rho_\infty b^2 \left\{ ba\ddot{h} - Ub \left(\frac{1}{2} - a \right) \dot{\alpha} - b^2 \left(\frac{1}{8} + a^2 \right) \ddot{\alpha} \right. \\ \left. + \frac{b^2}{\pi} (T_7 + (c-a)T_1) \ddot{\beta} \right. \\ \left. - \frac{bU}{\pi} \left(T_1 - T_8 - (c-a)T_4 + \frac{1}{2} T_{11} \right) \dot{\beta} \right. \\ \left. - \frac{U^2}{\pi} (T_4 + T_{10}) \beta \right\} \end{aligned} \quad (2.17)$$

$$\begin{aligned} M_{\beta,nc}(t) = \rho_\infty b^2 \left\{ bT_1 \ddot{h} - 2b^2 T_{13} \ddot{\alpha} + \frac{b^2}{\pi} T_3 \ddot{\beta} \right. \\ \left. + Ub \left[2T_9 + T_1 - T_4 \left(a - \frac{1}{2} \right) \right] \dot{\alpha} + \frac{bU}{2\pi} T_4 T_{11} \dot{\beta} \right. \\ \left. - \frac{U^2}{\pi} (T_5 - T_4 T_{10}) \beta \right\} \end{aligned} \quad (2.18)$$

While the circulatory is:

$$L_c(t) = -2\pi\rho_\infty Ub C(k) Q(t) \quad (2.19)$$

$$M_{\alpha,c}(t) = 2\pi\rho_\infty Ub^2 \left(a + \frac{1}{2} \right) C(k) Q(t) \quad (2.20)$$

$$M_{\beta,c}(t) = -\rho_\infty Ub^2 T_{12} C(k) Q(t) \quad (2.21)$$

Where $Q(t)$ is the total dynamic downwash (a measure of circulation) and is represented as:

$$Q(t) = U\alpha + \dot{h} + b\left(\frac{1}{2} - a\right)\dot{\alpha} + \frac{b}{2\pi}T_{11}\dot{\beta} + \frac{U}{\pi}T_{10}\beta \quad (2.22)$$

$C(k)$ is known as Theodorsen's function, it accounts for circulatory lift caused by vortex shedding of the fluttering airfoil or in other words, the lag between sinusoidal oscillation and lift development (Fung, 2002), it is a complex number function depends on the nondimensional reduced frequency $k = \frac{b\omega}{U}$ of the airfoil.

ρ_∞ is the air density and U is the air speed, and the geometric coefficients $T_i, i = 1, 2, \dots$ are called Theodorsen constants. They are functions of the nondimensional distances c and b as defined in the wing model, so they are specific for a typical section model and are listed in report no. 496 by Theodorsen (1935).

Assuming simple harmonic motion for all vibration modes, so we can model the position, rate, and acceleration of each degree of freedom in terms of the frequency ω as:

$$h(t) = h_0 e^{i\omega t}; \quad \dot{h}(t) = i\omega h; \quad \ddot{h}(t) = -\omega^2 h \quad (2.23)$$

$$\alpha(t) = \alpha_0 e^{i\omega t}; \quad \dot{\alpha}(t) = i\omega \alpha; \quad \ddot{\alpha}(t) = -\omega^2 \alpha \quad (2.24)$$

$$\beta(t) = \beta_0 e^{i\omega t}; \quad \dot{\beta}(t) = i\omega \beta; \quad \ddot{\beta}(t) = -\omega^2 \beta \quad (2.25)$$

Using Equations (2.23), (2.24), and (2.25) to temporarily convert the total dynamic downwash (2.22) to the frequency domain gives:

$$Q(\omega) = \frac{1}{i\omega} \left[i\omega U\alpha - \omega^2 h - \omega^2 b \left(\frac{1}{2} - a \right) \alpha - \frac{1}{2\pi} b T_{11} \omega^2 \beta + \frac{1}{\pi} T_{10} U i\omega \beta \right] \quad (2.26)$$

The first derivative of (2.22) with respect to time is:

$$\dot{Q}(t) = U\dot{\alpha} + \dot{h} + b \left(\frac{1}{2} - a \right) \ddot{\alpha} + \frac{1}{2\pi} b T_{11} \ddot{\beta} + \frac{1}{\pi} T_{10} U \dot{\beta} \quad (2.27)$$

Using Equations (2.23), (2.24), and (2.25) again to temporarily convert the first derivative of the total dynamic downwash (2.27) to the frequency domain gives:

$$\begin{aligned} Q'(\omega) = i\omega U\alpha - \omega^2 h - \omega^2 b \left(\frac{1}{2} - a \right) \alpha - \frac{1}{2\pi} b T_{11} \omega^2 \beta \\ + \frac{1}{\pi} T_{10} U i \omega \beta \end{aligned} \quad (2.28)$$

Comparing (2.26) and (2.28), $Q(\omega)$ can be expressed as:

$$Q(\omega) = \frac{1}{i\omega} Q'(\omega) \quad (2.29)$$

By substituting (2.29) in the circulatory part of aerodynamic forces Equations (2.19), (2.20), and (2.21) they are transferred to frequency domain and expressed as follows:

$$L_c(\omega) = -2\pi\rho_\infty U b \frac{C(k)}{i\omega} Q'(\omega) \quad (2.30)$$

$$M_{\alpha,c}(\omega) = 2\pi\rho_\infty U b^2 \left(a + \frac{1}{2} \right) \frac{C(k)}{i\omega} Q'(\omega) \quad (2.31)$$

$$M_{\beta,c}(\omega) = -\rho_\infty U b^2 T_{12} \frac{C(k)}{i\omega} Q'(\omega) \quad (2.32)$$

The *Fourier inverse* $\mathcal{F}^{-1} \left[\frac{C(k)}{i\omega} \right]$ is noun as Wagner's function $\Phi(t)$, which can be approximated using Jones' approximation as:

$$\Phi(t) = 1 - \delta_1 e^{-\lambda_1 \frac{U}{b} t} - \delta_2 e^{-\lambda_2 \frac{U}{b} t} \quad (2.33)$$

Where the constants $\delta_1 = 0.165$, $\delta_2 = 0.335$, $\lambda_1 = 0.041$, $\lambda_2 = 0.320$

Equations (2.30), (2.31), and (2.32) can now be transferred back to time domain using Fourier inversion and convolution theory. The convolution of two functions f and g is defined as:

$$f * g = \int_0^t f(t - \tau)g(\tau)d\tau \quad (2.34)$$

And the inverse Fourier transform of two multiplied functions $\mathcal{F}^{-1}\{F(\omega)G(\omega)\}$ is their convolution $f * g$, where f and g are the inverse Fourier transform of $F(\omega)$ and $G(\omega)$, this gives:

$$D(t) = \mathcal{F}^{-1}\left\{\frac{C(k)}{i\omega} Q'(\omega)\right\} = \int_0^t \Phi(t - \tau)\dot{Q}(\tau)d\tau \quad (2.35)$$

By substituting Jones' approximation (2.33) in (2.35), and rearranging we get

$$\begin{aligned} D(t) = & \int_0^t \dot{Q}(\tau)d\tau - \delta_1 \int_0^t e^{-\lambda_1 \frac{U}{b}(t-\tau)} \dot{Q}(\tau)d\tau \\ & - \delta_2 \int_0^t e^{-\lambda_2 \frac{U}{b}(t-\tau)} \dot{Q}(\tau)d\tau \end{aligned} \quad (2.36)$$

Equation (2.36) can be divided into two parts:

$$\int_0^t \dot{Q}(\tau)d\tau = Q(t) \quad (2.37)$$

and

$$\ell_n(t) = \int_0^t e^{-\lambda_n \frac{U}{b}(t-\tau)} \dot{Q}(\tau)d\tau = e^{-\lambda_n \frac{U}{b}t} \int_0^t e^{\lambda_n \frac{U}{b}\tau} \dot{Q}(\tau)d\tau \quad (2.38)$$

The term ℓ_n is called the n th aerodynamic lag state. It is associated with the unsteady aerodynamics and is a measure of the lag in the induced aerodynamic loads in following the motion of the airfoil.

Using (2.37) and (2.38), Equation (2.36) can be written as:

$$D(t) = Q(t) - \sum_{n=1}^2 \delta_n \ell_n \quad (2.39)$$

Substituting (2.39) in the circulatory part of aerodynamic forces Equations (2.30), (2.31), and (2.32) give:

$$L_c(t) = -2\pi\rho_\infty Ub D(t) = -2\pi\rho_\infty Ub [Q(t) - \delta_1 \ell_1 - \delta_2 \ell_2] \quad (2.40)$$

$$\begin{aligned} M_{\alpha,c}(t) &= 2\pi\rho_\infty Ub^2 \left(a + \frac{1}{2}\right) D(t) \\ &= 2\pi\rho_\infty Ub^2 \left(a + \frac{1}{2}\right) [Q(t) - \delta_1 \ell_1 - \delta_2 \ell_2] \end{aligned} \quad (2.41)$$

$$\begin{aligned} M_{\beta,c}(t) &= -\rho_\infty Ub^2 T_{12} D(t) \\ &= -\rho_\infty Ub^2 T_{12} [Q(t) - \delta_1 \ell_1 - \delta_2 \ell_2] \end{aligned} \quad (2.42)$$

A solution for ℓ_n is required to solve these equations. The first derivative of (2.38) with respect to time is:

$$\begin{aligned} \dot{\ell}_n(t) &= -\lambda_n \frac{U}{b} e^{-\lambda_n \frac{U}{b} t} \int_0^t e^{\lambda_n \frac{U}{b} \tau} \dot{Q}(\tau) d\tau \\ &\quad + e^{-\lambda_n \frac{U}{b} t} \frac{d}{dt} \left[\int_0^t e^{\lambda_n \frac{U}{b} \tau} \dot{Q}(\tau) d\tau \right] \end{aligned} \quad (2.43)$$

This is equivalent to:

$$\dot{\ell}_n(t) = -\lambda_n \frac{U}{b} \ell_n(t) + \dot{Q}(t) \quad (2.44)$$

or

$$\begin{aligned} \dot{\ell}_n(t) &= -\lambda_n \frac{U}{b} \ell_n(t) + U\dot{\alpha} + \ddot{h} + b \left(\frac{1}{2} - a\right) \ddot{\alpha} + \frac{1}{2\pi} b T_{11} \ddot{\beta} \\ &\quad + \frac{1}{\pi} T_{10} U \dot{\beta} \end{aligned} \quad (2.45)$$

This results in two simultaneous linear differential equations. where ℓ_1 and ℓ_2 are evaluated by solving them. In matrix form, they can be expressed as:

$$\begin{Bmatrix} \dot{\ell}_1(t) \\ \dot{\ell}_2(t) \end{Bmatrix} = Q_a \begin{Bmatrix} \ddot{h} \\ \ddot{\alpha} \\ \ddot{\beta} \end{Bmatrix} + Q_v \begin{Bmatrix} \dot{h} \\ \dot{\alpha} \\ \dot{\beta} \end{Bmatrix} + L_\lambda \begin{Bmatrix} \ell_1 \\ \ell_2 \end{Bmatrix} \quad (2.46)$$

where

$$Q_a = \begin{bmatrix} 1 & b\left(\frac{1}{2} - a\right) & \frac{bT_{11}}{2\pi} \\ 1 & b\left(\frac{1}{2} - a\right) & \frac{bT_{11}}{2\pi} \end{bmatrix} \quad (2.47)$$

$$Q_v = U \begin{bmatrix} 0 & 1 & \frac{T_{10}}{\pi} \\ 0 & 1 & \frac{T_{10}}{\pi} \end{bmatrix} \quad (2.48)$$

$$L_\lambda = \begin{bmatrix} -\lambda_1 \frac{U}{b} & 0 \\ 0 & -\lambda_2 \frac{U}{b} \end{bmatrix} \quad (2.49)$$

The total unsteady aerodynamic forces (circulatory and non-circulatory) can be expressed as:

$$\begin{aligned} L(t) = & -\pi\rho_\infty b^2 \ddot{h} + \pi\rho_\infty b^3 a \ddot{\alpha} + \rho_\infty b^3 T_1 \ddot{\beta} - 2\pi\rho_\infty b U \dot{h} \\ & - 2\pi\rho_\infty b^2 U (1-a) \dot{\alpha} + \rho_\infty b^2 U (T_4 - T_{11}) \dot{\beta} \\ & - 2\pi\rho_\infty b U^2 \alpha - 2\rho_\infty b U^2 T_{10} \beta + 2\pi\rho_\infty b U \delta_1 \ell_1 \\ & + 2\pi\rho_\infty b U \delta_2 \ell_2 \end{aligned} \quad (2.50)$$

$$\begin{aligned}
M_\alpha(t) = & \pi\rho_\infty b^3 a \ddot{h} - \pi\rho_\infty b^4 \left(\frac{1}{8} + a^2 \right) \ddot{\alpha} \\
& + \rho_\infty b^4 (T_7 + (c - a)T_1) \ddot{\beta} \\
& + 2\pi\rho_\infty b^2 U \left(a + \frac{1}{2} \right) \dot{h} - 2\pi\rho_\infty b^3 U a \left(\frac{1}{2} - a \right) \dot{\alpha} \\
& + \rho_\infty b^3 U (T_8 - T_1 + (c - a)T_4 + aT_{11}) \dot{\beta} \\
& + \rho_\infty b^2 U^2 (2aT_{10} - T_4) \beta \\
& + 2\pi\rho_\infty b^2 U^2 \left(a + \frac{1}{2} \right) \alpha \\
& - 2\pi\rho_\infty b^2 U \left(a + \frac{1}{2} \right) \delta_1 \ell_1 \\
& - 2\pi\rho_\infty U b^2 \left(a + \frac{1}{2} \right) \delta_2 \ell_2
\end{aligned} \tag{2.51}$$

$$\begin{aligned}
M_\beta(t) = & \rho_\infty b^3 T_1 \ddot{h} - 2\rho_\infty b^4 T_{13} \ddot{\alpha} + \rho_\infty b^4 \frac{1}{\pi} T_3 \ddot{\beta} - \rho_\infty b^2 U T_{12} \dot{h} \\
& + \rho_\infty b^3 U \left[2T_9 + T_1 + (T_4 - T_{12}) \left(\frac{1}{2} - a \right) \right] \dot{\alpha} \\
& + \rho_\infty b^3 U \frac{1}{2\pi} T_{11} (T_4 - T_{12}) \dot{\beta} - \rho_\infty b^2 U^2 T_{12} \alpha \\
& - \rho_\infty b^2 U^2 \frac{1}{\pi} [T_5 - T_{10} (T_4 - T_{12})] \beta \\
& + \rho_\infty U b^2 T_{12} \delta_1 \ell_1 + \rho_\infty U b^2 T_{12} \delta_2 \ell_2
\end{aligned} \tag{2.52}$$

In Matrix form, the three equations can be expressed as:

$$\begin{Bmatrix} L(t) \\ M_\alpha(t) \\ M_\beta(t) \end{Bmatrix} = M_a \begin{Bmatrix} \ddot{h} \\ \ddot{\alpha} \\ \ddot{\beta} \end{Bmatrix} + D_a \begin{Bmatrix} \dot{h} \\ \dot{\alpha} \\ \dot{\beta} \end{Bmatrix} + K_a \begin{Bmatrix} h \\ \alpha \\ \beta \end{Bmatrix} + L_\delta \begin{Bmatrix} \ell_1 \\ \ell_2 \end{Bmatrix} \tag{2.53}$$

Where the aerodynamic inertia matrix M_a is:

$$M_a = \pi\rho_\infty b^3 \begin{bmatrix} -\frac{1}{b} & a & \frac{T_1}{\pi} \\ a & -b\left(\frac{1}{8} + a^2\right) & -\frac{2bT_{13}}{\pi} \\ \frac{T_1}{\pi} & -\frac{2bT_{13}}{\pi} & \frac{bT_3}{\pi^2} \end{bmatrix} \quad (2.54)$$

The aerodynamic damping matrix D_a is:

$$D_a = \rho_\infty b^2 U \begin{bmatrix} -\frac{2\pi}{b} & -2\pi(1-a) & (T_4 - T_{11}) \\ \pi(2a+1) & -\pi ba(1-2a) & b(T_8 - T_1 + (c-a)T_4 + aT_{11}) \\ -T_{12} & b\left(2T_9 + T_1 + (T_{12} - T_4)\left(a - \frac{1}{2}\right)\right) & \frac{b}{2\pi}T_{11}(T_4 - T_{12}) \end{bmatrix} \quad (2.55)$$

The aerodynamic stiffness matrix K_a is:

$$K_a = \rho_\infty b^2 U^2 \begin{bmatrix} 0 & -\frac{2\pi}{b} & -\frac{2T_{10}}{b} \\ 0 & 2\pi\left(a + \frac{1}{2}\right) & 2aT_{10} - T_4 \\ 0 & -T_{12} & -\frac{1}{\pi}(T_5 - T_{10}(T_4 - T_{12})) \end{bmatrix} \quad (2.56)$$

And the aerodynamic lagging matrix L_δ is:

$$L_\delta = b^2 U \begin{bmatrix} \frac{2\pi\delta_1}{b} & \frac{2\pi\delta_2}{b} \\ -2\pi\left(a + \frac{1}{2}\right)\delta_1 & -2\pi\left(a + \frac{1}{2}\right)\delta_2 \\ T_{12}\delta_1 & T_{12}\delta_2 \end{bmatrix} \quad (2.57)$$

2.2.3 The Full System Equations in Dimensionless Form

To further simplify the problem, the equations are rewritten in dimensionless form, this can be achieved using the following dimensionless variables (Hodges and Pierce, 2011):

1. $\mu = \frac{m}{\pi\rho_{\infty}b^2}$; the ratio of the total wing's mass to the mass of the air affected by the wing
2. $r_{\alpha}^2 = \frac{I_{\alpha}}{mb^2}$; the dimensionless radius of gyration of the wing about the elastic axis
3. $r_{\beta}^2 = \frac{I_{\beta}}{m_{\beta}b^2}$; the dimensionless radius of gyration of the control surface about its hinge
4. $K_h = m\omega_h^2$; is the plunge structural stiffness, where ω_h is uncoupled plunge frequency
5. $K_{\alpha} = I_{\alpha}\omega_{\alpha}^2$; is the pitch structural stiffness, where ω_{α} is uncoupled pitch frequency
6. $K_{\beta} = I_{\beta}\omega_{\beta}^2$; is the control surface structural stiffness, where ω_{β} is uncoupled control surface frequency
7. $\sigma = \frac{\omega_h}{\omega_{\alpha}}$; is the ratio of uncoupled plunge and pitch natural frequencies.
8. $V = \frac{U}{b\omega_{\alpha}}$; is the reduced velocity, or the dimensionless free stream speed of air.
9. $c_h = 2m\omega_h\zeta_h$; is the plunge structural damping, where ζ_h is the plunge damping ratio.
10. $c_{\alpha} = 2I_{\alpha}\omega_{\alpha}\zeta_{\alpha}$; is the pitch structural damping, where ζ_{α} is the pitch damping ratio
11. $c_{\beta} = 2I_{\beta}\omega_{\beta}\zeta_{\beta}$; is the control surface structural damping, where ζ_{β} is the control surface damping ratio
12. $\tau = \omega_{\alpha}t$; is the dimensionless time

Substituting those parameters in the system matrices and simplifying, gives the following dimensionless matrices:

$$\bar{M}_s = \mu \begin{bmatrix} 1 & x_\alpha & \frac{m_\beta}{m} x_\beta \\ x_\alpha & r_\alpha^2 & \frac{m_\beta}{m} [(c-a)x_\beta + r_\beta^2] \\ \frac{m_\beta}{m} x_\beta & \frac{m_\beta}{m} [(c-a)x_\beta + r_\beta^2] & \frac{m_\beta}{m} r_\beta^2 \end{bmatrix} \quad (2.58)$$

$$\bar{D}_s = 2\mu \begin{bmatrix} \sigma \zeta_h & 0 & 0 \\ 0 & r_\alpha^2 \zeta_\alpha & 0 \\ 0 & 0 & \frac{m_\beta}{m} \frac{\omega_\beta}{\omega_\alpha} r_\beta^2 \zeta_\beta \end{bmatrix} \quad (2.59)$$

$$\bar{K}_s = \mu \begin{bmatrix} \sigma^2 & 0 & 0 \\ 0 & r_\alpha^2 & 0 \\ 0 & 0 & \frac{m_\beta}{m} \left(\frac{\omega_\beta}{\omega_\alpha}\right)^2 r_\beta^2 \end{bmatrix} \quad (2.60)$$

$$\bar{L}_c = \mu \begin{bmatrix} 0 \\ 0 \\ \frac{m_\beta}{m} \left(\frac{\omega_\beta}{\omega_\alpha}\right)^2 r_\beta^2 \end{bmatrix} \quad (2.61)$$

$$\bar{M}_a = \begin{bmatrix} -1 & a & \frac{T_1}{\pi} \\ a & -\left(\frac{1}{8} + a^2\right) & -\frac{2T_{13}}{\pi} \\ \frac{T_1}{\pi} & -\frac{2T_{13}}{\pi} & \frac{T_3}{\pi^2} \end{bmatrix} \quad (2.62)$$

$$\bar{D}_a = V \begin{bmatrix} -2 & -2(1-a) & \frac{T_4 - T_{11}}{\pi} \\ 1+2a & a(1-2a) & \frac{1}{\pi}(T_8 - T_1 + (c-a)T_4 + aT_{11}) \\ -\frac{T_{12}}{\pi} & \frac{1}{\pi}(2T_9 + T_1 + (T_{12} - T_4)(a - \frac{1}{2})) & \frac{T_{11}}{2\pi^2}(T_4 - T_{12}) \end{bmatrix} \quad (2.63)$$

$$\bar{K}_a = V^2 \begin{bmatrix} 0 & -2 & -\frac{2T_{10}}{\pi} \\ 0 & 1+2a & \frac{1}{\pi}(2aT_{10} - T_4) \\ 0 & -\frac{T_{12}}{\pi} & -\frac{1}{\pi^2}(T_5 - T_{10}(T_4 - T_{12})) \end{bmatrix} \quad (2.64)$$

$$\bar{L}_\delta = 2V \begin{bmatrix} \delta_1 & \delta_2 \\ -\left(\frac{1}{2} + a\right)\delta_1 & -\left(\frac{1}{2} + a\right)\delta_2 \\ \frac{T_{12}\delta_1}{2\pi} & \frac{T_{12}\delta_2}{2\pi} \end{bmatrix} \quad (2.65)$$

$$\bar{Q}_a = \begin{bmatrix} 1 & \frac{1}{2} - a & \frac{T_{11}}{2\pi} \\ 1 & \frac{1}{2} - a & \frac{T_{11}}{2\pi} \end{bmatrix} \quad (2.66)$$

$$\bar{Q}_v = U \begin{bmatrix} 0 & 1 & \frac{T_{10}}{\pi} \\ 0 & 1 & \frac{T_{10}}{\pi} \end{bmatrix} \quad (2.67)$$

$$\bar{L}_\lambda = V \begin{bmatrix} -\lambda_1 & 0 \\ 0 & -\lambda_2 \end{bmatrix} \quad (2.68)$$

With the dimensionless state vector for the modal displacements of *the three degrees of freedom* $x_s = \begin{bmatrix} h \\ b \\ \alpha \\ \beta \end{bmatrix}^T$, And the *two aerodynamic lag states* $x_a = [\ell_1 \ \ell_2]^T$.

Using the matrices (2.58) to (2.68), the full system can be represented as:

$$\begin{aligned}
& [\bar{M}_s - \bar{M}_a] \begin{Bmatrix} \ddot{h}/b \\ \ddot{\alpha} \\ \ddot{\beta} \end{Bmatrix} + [\bar{D}_s - \bar{D}_a] \begin{Bmatrix} \dot{h}/b \\ \dot{\alpha} \\ \dot{\beta} \end{Bmatrix} + [\bar{K}_s - \bar{K}_a] \begin{Bmatrix} h/b \\ \alpha \\ \beta \end{Bmatrix} \\
& = \bar{L}_\delta \begin{Bmatrix} \ell_1 \\ \ell_2 \end{Bmatrix} + \bar{L}_c \{\beta_c\}
\end{aligned} \tag{2.69}$$

2.2.4 Converting the System's Equations to State Space Representation

Using state space representation is a noticeable characteristic in modern control theory. Where the system is described by a set of first-order differential equations that relates the system input-output dynamics (Zhang, 2010).

State space representation has the advantage of being easily extended to multivariable case, which can then be easily used to analyze the system response and design controllers (Rossiter, 2003).

Equation (2.69) can now be converted to the standard state space form which is:

$$\dot{x} = Ax + Bu \tag{2.70}$$

$$y = Cx + Du \tag{2.71}$$

The complete state vector is defined by combining \dot{x}_s , x_s and x_a as following:

$$x = [\dot{h}/b \quad \dot{\alpha} \quad \dot{\beta} \quad h/b \quad \alpha \quad \beta \quad \ell_1 \quad \ell_2]^T \tag{2.72}$$

The system matrix A is:

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \tag{2.73}$$

where

$$A_{11_{3 \times 3}} = -[\bar{M}_s - \bar{M}_a]^{-1}[\bar{D}_s - \bar{D}_a] \tag{2.74}$$

$$A_{12_{3x3}} = -[\bar{M}_s - \bar{M}_a]^{-1}[\bar{K}_s - \bar{K}_a] \quad (2.75)$$

$$A_{13_{3x2}} = [\bar{M}_s - \bar{M}_a]^{-1}\bar{L}_\delta \quad (2.76)$$

$$A_{21_{3x3}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.77)$$

$$A_{22_{3x3}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (2.78)$$

$$A_{23_{3x2}} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (2.79)$$

$$A_{31_{2x3}} = \bar{Q}_a A_{11} + \bar{Q}_v \quad (2.80)$$

$$A_{32_{2x3}} = \bar{Q}_a A_{12} \quad (2.81)$$

$$A_{33_{2x2}} = \bar{Q}_a A_{13} + \bar{L}_\lambda \quad (2.82)$$

The input matrix B is:

$$B_{8x1} = \begin{bmatrix} B_{11} \\ B_{21} \\ B_{31} \end{bmatrix} \quad (2.83)$$

$$B_{11_{3x1}} = [\bar{M}_s - \bar{M}_a]^{-1}[\bar{L}_c] \quad (2.84)$$

$$B_{21_{3x1}} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.85)$$

$$B_{31_{2x1}} = \bar{Q}_a B_{11} \quad (2.86)$$

The output matrix C relates the state variables to the measured system variables. For this system, the two aerodynamic lag states are unmeasurable. In

addition, measuring all the remaining six states encounters unnecessary cost, so it's been decided to measure only one state (the control surface angle), while the remaining seven states are estimated using state estimator.

$$C = [0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0] \quad (2.87)$$

2.3 Open Loop Stability Analysis

When a lifting surface is disturbed while it is being in an airstream at a speed less than its flutter speed, the oscillations resulted from those disturbances will vanish out in time with exponentially decreasing amplitudes. This is because the air is providing damping for all these motions. While when the speed is above flutter speed, what happens can be described as negative damping provided by the air. Thus, the oscillation amplitude grows exponentially.

The open loop aeroelastic system is represented in state space form as:

$$\dot{x} = Ax \quad (2.88)$$

Where the system matrix A , is a function of the airspeed U , and air density ρ .

The system stability can be analyzed, and the flutter speed can be calculated using the eigen value analysis of the state matrix of A (the roots of the system's characteristic equation), over a range of air speeds where the flutter is expected (De Marqui et al., 2005).

For a given airspeed and density, the eigen values has a complex form $p = g + i\omega$, where the real part represents the damping frequency, and the imaginary is the *oscillation angular frequency* at that speed. The system is stable if the real part of each eigen value is negative. When the real part of any one eigenvalue becomes positive, the entire system becomes unstable (Conner et al., 1997).

To find the flutter speed, the modal damping at each air speed (real part of the eigen value) is calculated and then plotted against the air speeds range to find the zero-damping point (flutter speed), at which the sign of the damping frequency is changing from negative (stable system) to positive (unstable) (York, 1980).

2.4 Closed Loop System Design

The main goal of the controller is to suppress the flutter instability and keep the system stable over a wide range of velocities. In theory, closed loop control can be achieved in a few simple steps, the first step is the measurement (or estimation) of the motion components (or state variables) needed to construct the control law. Then the flap deflection command is formed by the control law and fed into the actuator which deflects the flap at the proper rate and phase. And hence it provides the aerodynamic forces and moments that oppose the motion of the wing. Generating the control law can be done in many different techniques such as classical control methods, modern control, optimal and other non-conventional methods (De Marqui et al., 2005; York, 1980).

In this study, two methods are used to find and simulate the control law. First is the LQG optimal controller. Then the MPC, which can implement a real-time optimization process on the system's states taking in consideration the physical system constraints.

2.4.1 System's Controllability and Observability

Controllability and observability are important characteristics of the plant, that are necessary to be checked prior to design a controller and an observer, respectively. A system is said to be controllable if there exist a control input $u(t)$ that can change

the system's states from any arbitrary initial value $\mathbf{x}(0)$, to any final value $\mathbf{x}(t_f)$, in a finite time t_f (Tewari, 2011).

It can be shown that an LTI (linear time invariant) system is controllable if and only if its controllability matrix has a full rank.

$$\text{rank}(CO) = \text{rank} [B \quad AB \quad A^2B \quad \dots \quad A^{n-1}B] = n \quad (2.89)$$

In practical cases, where not all system states are measurable, an observer is needed to estimate the states.

A continuous-time system is observable if for any initial state $x(0)$ can be uniquely determined by knowledge of the input $u(\tau)$ and output $y(\tau)$ for all $\tau \in [0, t]$ (Simon, 2006).

It can be shown that an LTI system is observable if and only if the observability matrix has a full rank.

$$\text{rank}(OB) = \text{rank} [C \quad CA \quad CA^2 \quad \dots \quad CA^{n-1}]^T = n \quad (2.90)$$

2.4.2 Linear Quadratic Regulator Optimal Controller

Optimal control is an important field in modern control theory. The main idea behind its concept is to design the best possible control system for a given performance requirements, which is typically the minimum control energy to satisfy the constraints of the maximum overshoot and settling time (Ashish, 2002).

If the state equations and the initial conditions of the system, in addition to the objectives set are given, optimal control methods can find a feasible control, such that the system that starts from the given initial conditions transfers its states to the objectives set, and in so doing minimizes a cost function (Zhang, 2010).

When the air speed U is greater than the flutter speed U_f , the airfoil become unstable, the objective of the closed loop system is to find a control function $u(t)$ on time period $[0, \infty]$ to stabilize the system. In other words, for a system in the state space form shown in Equations (2.70), and (2.71), if the required input is zero (regulator case), the LQR objective is to find a controller in the following form:

$$u(t) = -K_{LQR}x \quad (2.91)$$

The control energy can be expressed in a quadratic form as $u(t)^T R_{LQR} u(t)$, where R_{LQR} is a square, symmetric matrix called the control cost weighting matrix. And the transient energy in the form $x(t)^T Q_{LQR} x(t)$, where Q_{LQR} is also a square symmetric matrix called the state weighting matrix. The objective function can then be written as:

$$J = \int_0^{\infty} [x(t)^T Q_{LQR} x(t) + u(t)^T R_{LQR} u(t)] dt \quad (2.92)$$

The weighting matrices Q and R play critical roles in the LQR optimization process as the compositions of their components have a significant impact on system performance. The designer is free to choose the matrices Q and R , however the selection of matrices Q and R is often based on an iterative method that incorporates experience and practical understanding of the problem (Vinodh Kumar and Jerome, 2013).

The optimal control problem consists of solving for the feedback gain matrix K_{LQR} such that the performance measure (cost function) J is minimized subject to the constraint that $x(t)$ remains the solution of the system's state space equation (Ashish, 2002; Xie et al., 2020).

The value of K_{LQR} is obtained by solving the algebraic Riccati Equation (2.93) to find S which is a unique, symmetric, positive semidefinite solution.

$$SA + A^T S - SBR^{-1}B^T S + Q = 0 \quad (2.93)$$

Then K_{LQR} can be found as:

$$K_{LQR} = R^{-1}B^T S \quad (2.94)$$

The regulator closed loop system can then be represented as:

$$\dot{x} = (A - BK_{LQR})x \quad (2.95)$$

A block diagram of the linear quadratic regulator is shown in Figure 4.

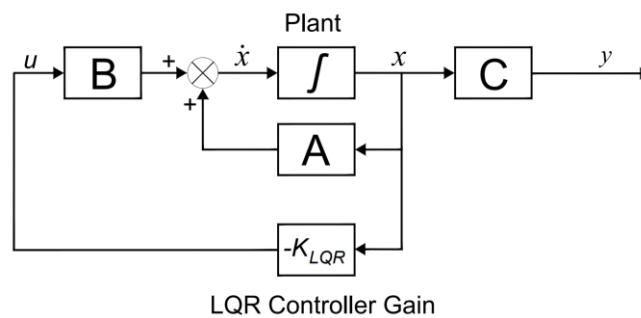


Figure 4: LQR block diagram

From the above, the LQR design procedure may be summarized in the following steps:

1. Checking the controllability of the system and control matrices A and B .
2. Selecting the design parameter matrices Q_{LQR} and R_{LQR} .
3. Solving the algebraic Riccati equation for S .
4. Finding the state variable feedback gain using $K_{LQR} = R^{-1}B^T S$.
5. Tuning the design parameter matrices Q_{LQR} and R_{LQR} to achieve the required performance.

2.4.3 State Estimators and Linear Quadratic Gaussian Compensator

As mentioned in the previous section, LQR optimal control assumes that all states values are available for the controller, but in most practical cases, this assumption is not true, as some states could be unmeasurable. Even in the cases of measurable states, accurate sensors can either be unavailable, or too expensive.

In addition, one of the LQR theory drawbacks is it's lake of robustness, so systems based on LQR fails to be robust to measurement noise, external disturbances and unmodeled dynamics (Naidu, 2002).

In such cases, a state estimator (observer) is required. The estimator concept has been wildly used in the engineering filed. In addition to estimating the full states, the state observer can work as a noise filter to reduce the impact of measurement noise (Ashish, 2002; Wang, 2009).

To design an observer for a plant, the plant should be observable, which means all the state variables must affect the output. If so, then based on the matrices A , B , C , and the input u , we may estimate the unmeasured x states by simply duplicating the original system as:

$$\dot{\hat{x}} = A\hat{x} + Bu \quad (2.96)$$

Equation (2.96) represents the open loop estimator. Using it is not recommended, as it requires an initial state estimate for each time step. In addition, if the real part of the eigenvalues of the system matrix A are positive, the error will keep increasing.

To overcome the disadvantages of open loop estimator, the error between the measured and estimated states $y - \hat{y}$ can be used to close the loop, as shown in Figure 5.

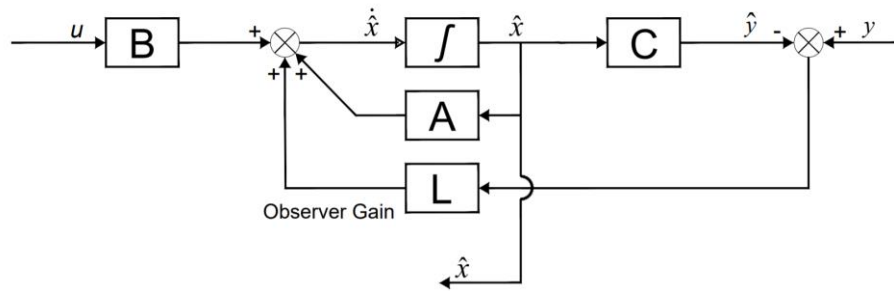


Figure 5: Closed loop state estimator (Observer) block diagram

The closed loop estimator equations are:

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - \hat{y}) \quad (2.97)$$

$$\hat{y} = C\hat{x} \quad (2.98)$$

For every new measurement y , the value is compared with the model value \hat{y} , then the error is corrected for the full state (Brunton, 2017).

Equation (2.97) can be rearranged as:

$$\dot{\hat{x}} = [A - LC]\hat{x} + Bu + Ly \quad (2.99)$$

Where $[A - LC]$ is the closed loop observer dynamics matrix, while u , and y are the inputs to the state estimator.

It is obvious from Equation (2.99) that the closed loop estimator's eigenvalues can be determined by selecting the observer gain L , if all eigenvalues have negative real parts, then all entries of the state's error vector $e_{obs}(t) = x - \hat{x}$ will approach zero. But as in the state feedback control systems, determining the gain that gives the best eigenvalues is not a simple problem.

In general, the estimator eigenvalues should be faster than the desired eigenvalues of the closed loop system, taking in consideration the saturation and noise problems (Chen, 2013).

The “optimal” way of evaluating the observer’s gain L is known as “Kalman filter”, which was invented in 1960 by R. E. Kalman. Kalman’s filter is an analogue of the LQR controller but for estimation, where it minimizes the mean square estimation error function J using the covariance of the measurement and process noise (disturbance) as weighting matrices.

The state space system equations, with measurement and process noise are:

$$\dot{x} = Ax + Bu + Gw \quad (2.100)$$

$$y = Cx + Du + v \quad (2.101)$$

Where w is the process noise (disturbance) vector, it is a function of time, and may arise due to modeling errors.

v is also a function of time, it represents the errors in measured signals. It is assumed that both w and v are white noise vectors (Gaussian random) with covariances Q_{Kalman} and R_{Kalman} , respectively (Hovland, 2004).

The mean square estimation error function J is:

$$J = \mathbb{E}[(x(t) - \hat{x}(t))^T (x(t) - \hat{x}(t))] \quad (2.102)$$

For steady state Kalman filter, the Gain L is:

$$L = S_e C^T R^{-1} \quad (2.103)$$

Where S_e is the solution of the estimator’s Riccati equation, which is:

$$S_e A^T + A S_e - S_e C^T R^{-1} C S_e + Q = 0 \quad (2.104)$$

As with LQR design, the Optimal estimator (Kalman filter) design procedure is summarized as following:

1. Checking the observability of the system and output matrices A and C

2. Using the system model to generate an estimate of the state vector x called \hat{x} .
3. Selecting the design parameter matrices which are the covariance matrices of the measurement and modeling noises Q_{Kalman} , and R_{Kalman} . These settings enable the designer to optimally balance the speed of state reconstruction with measurement noise protection (Block et al., 1997).

The system that combines a controller designed using LQR method, and an optimal observer (Kalman filter), is referred to as LQG compensator.

Figure 6 presents the block diagram of LQG.

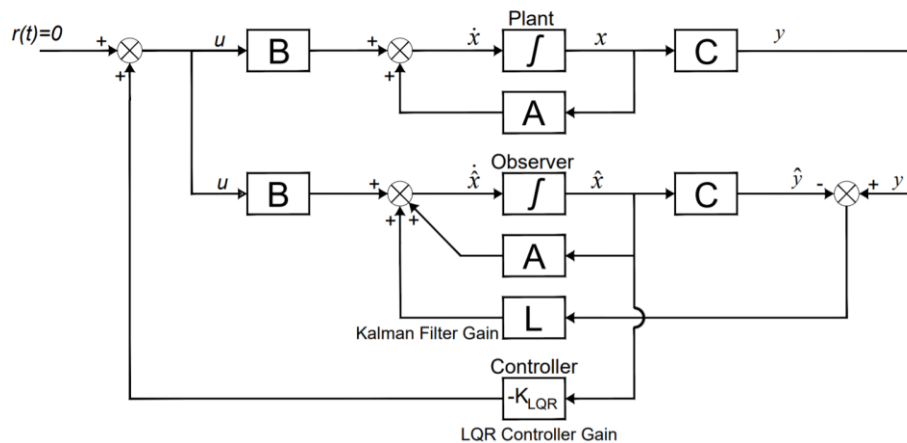


Figure 6: LQG compensator block diagram

LQG is represented as:

$$\dot{\hat{x}} = [A - BK_{\text{LQR}} - LC]\hat{x} + Ly \quad (2.105)$$

$$u = -K_{\text{LQR}}\hat{x} \quad (2.106)$$

For reference tracking case, where one or more of the desired states are non-zero, and the objective of the controller is to reach zero steady state error $e(t) = r(t) - y(t)$ while canceling out the effect of the noise (Ashish, 2002). one way to

achieve this goal is to use feedforward gain N to scale the reference input so that $u(t) = Nr(t) - y(t)$.

Another more robust and precise way is to add an integrator for each reference input as shown in the block diagram of Figure 7, where the integrator error equation is

$$\dot{x}_I(t) = e(t) = r(t) - y(t) = r(t) - Cx(t) \quad (2.107)$$

In general, the integral state can be augmented into the system dynamics state space model as

$$\begin{bmatrix} \dot{x} \\ \dot{x}_I \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} x \\ x_I \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ I \end{bmatrix} r(t) \quad (2.108)$$

where the control law $u(t)$ is

$$u(t) = -K \begin{bmatrix} x \\ x_I \end{bmatrix} = -[K_x \quad K_I] \begin{bmatrix} x \\ x_I \end{bmatrix} \quad (2.109)$$

The state feedback K which corresponds to K_x and K_I may be generated using LQR method as before

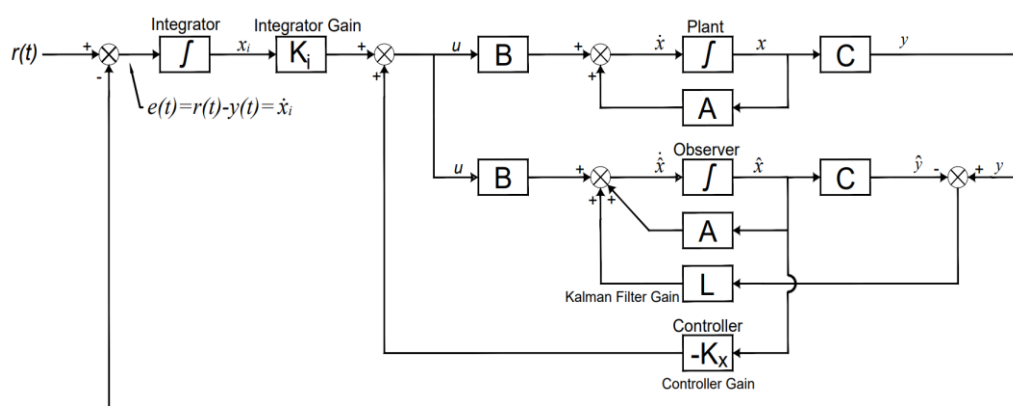


Figure 7: LQG compensator with integral action

2.4.4 Model Predictive Control using Laguerre Functions

The principle of prediction control comes basically from the idea that the future values of the system outputs can be predicted accurately if an accurate process model and current measurements are available. MPC are multivariable control algorithms that rely on generating process input values as a solution of a real-time optimization problem, constructed based on the process model and measurements, where the real-time optimization problem takes account of system dynamics, constraints, and control objectives, which are not handled explicitly by other control algorithms. Compared to optimal control methods, MPC algorithms are known for handling constraints, as they permit limitations on allowable control action. This has made it widely and successfully applied in many areas (Holkar and Waghmare, 2010; Nikolaou, 2001; Seborg et al., 2010).

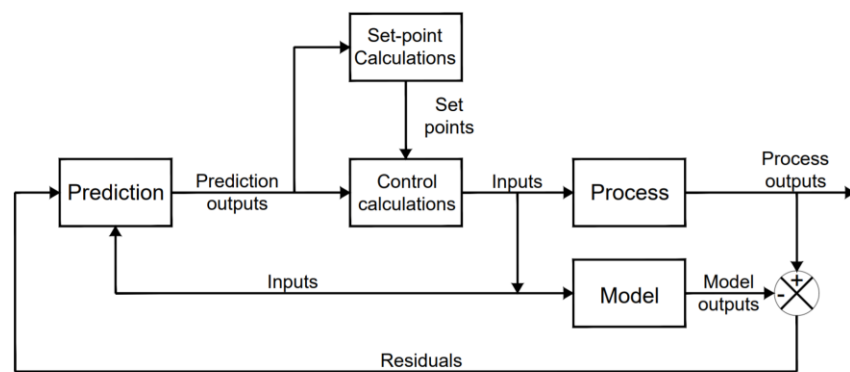


Figure 8: MPC block diagram (Seborg et al., 2010)

As shown in Figure 8, the current values of the output variables are predicted using a process model and compared to the actual process outputs. The difference value (residuals) is then fed back to the prediction block. The prediction block output is required at each sampling instant as an input for set-point calculations and control calculations. The setpoints for control calculations (targets) are calculated from an

economic optimization based on a steady state model of the process (Seborg et al., 2010).

Model predictive control requires minimizing a cost function J of the state x and the control input u over a finite time horizon called the prediction horizon N_p subject to a set of state and input constraints. Based on the current measurements and predictions of the future values of the output, a sequence of control moves is computed and then applied to the system over a time horizon called the control horizon N_c , where $N_c \leq N_p$, after which a new optimal control sequence is computed. The control input is computed and applied over a sliding window in time (Prazenica, 2014).

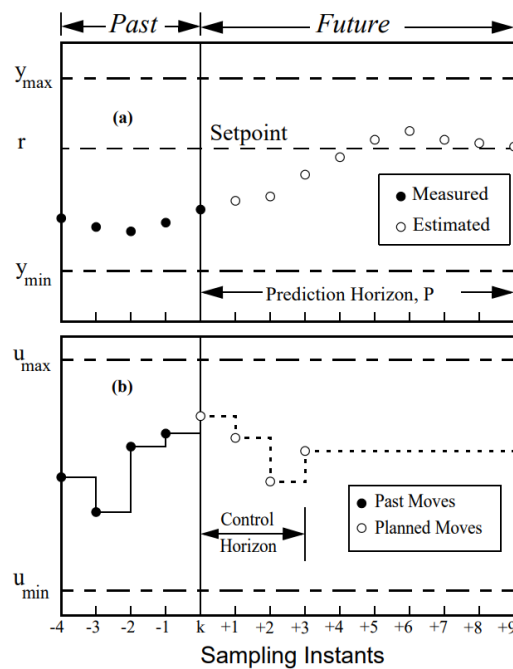


Figure 9: How MPC works (Bemporad et al., 2014).

The measured output, estimated outputs, and manipulated input u (past and planned moves) are shown in Figure 9, at the current sampling instant k , the controller calculates a set of values for the input that includes the current instant k input and the $M - 1$ future inputs to cover the control horizon length. Although M inputs are

available, only the first computed input is implemented. The procedure is then repeated at each subsequent instant (Seborg et al., 2010).

In MPC the process outputs are called controlled variables (CV's), while the process inputs are the manipulated variables (MV's). The main components of MPC are (Rossiter, 2003):

1. Actions depend on predictions: unlike most control laws, MPC explicitly compute the predicted behavior over some horizon. This allows for restricting the choice of the proposed input trajectories to those that do not lead to undesired future outputs.
2. Predictions are model based: to predict the future behavior of a process, there must be a model that shows the dependence of the output on the current measured variables and the current inputs. The model does not have to be linear, and because the decisions are updated regularly, model uncertainty can be dealt with rapidly.
3. Selecting the current input: criteria to decide which control action is the best is required to select the current input. Like in optimal control, this is done by selecting inputs that minimize a cost function.
4. Receding horizon: predictive control works by considering the predicted behavior over some horizon into the future, and therefore at each following sampling instant, it predicts one further sample into the future. When new information becomes available the input trajectory is automatically modified accordingly.
5. Optimal performance: MPC is a model-based algorithm, the more precise the model, the more accurate is the control.

6. Tuning: in MPC, stability and tuning are related to the cost function. They will take care of themselves if the cost function is right. Tuning is usually straightforward if the relative importance of performance in different loops is defined.
7. Handling of constraints: a major advantage of MPC over the other control syntheses is its ability to do systematic on-line constraint handling. This is done by optimizing the predicted performance subject to constraint satisfaction.
8. Feed forward is integrated with the constraint handling in MPC, this allows the controller to take account of future changes in the desired trajectory and include it in the overall control design.
9. Multivariable systems: this is one more major advantage for MPC, as it can deal systematically with multi variable MIMO systems.

Using MPC in the fields of relatively fast dynamic systems, such as aerospace, or fast sampling frequency control-loops like power electronics, systems require long prediction horizons to describe the complete transitory behavior of the system. The classic MPC approach is not computationally efficient for long prediction horizons, as it can lead to poorly numerically conditioned solutions, and heavy computational effort when implemented in real-time (Pineiro and Silveira, 2021; Wang, 2004).

Researchers have studied and presented several ways to reduce the computational effort in MPC. Among these methods is using Laguerre functions, which is implemented in this study, as it can be used to capture the future MPC control trajectory with small number of parameters that reduces the computational effort and yet gives good performance (Rossiter et al., 2010).

MPC is usually implemented in discrete time. The general plant discrete-time state space model, which can be generated easily by MATLAB[®] is described by:

$$\begin{aligned}x_m(k+1) &= A_m x_m(k) + B_m u(k) + B_d w(k) \\ y(k) &= C_m x_m(k)\end{aligned}\quad (2.110)$$

With n_1 states, m inputs and q outputs. $u(k)$ is the vector of manipulated variables (inputs), $x_m(k)$ is the state vector, and $w(k)$ is the input disturbance which is assumed to be a sequence of integrated white noise.

$$w(k) - w(k-1) = \epsilon(k) \quad (2.111)$$

To eliminate steady state errors, and in the presence of uncertainty or disturbances, it is necessary to embed integrators into the model (Rossiter, 2003).

By defining:

$$\begin{aligned}\Delta x_m(k) &= x_m(k) - x_m(k-1) \\ \Delta u(k) &= u(k) - u(k-1)\end{aligned}\quad (2.112)$$

Then $\Delta x_m(k+1)$ will be:

$$\Delta x_m(k+1) = A_m \Delta x_m(k) + B_m \Delta u(k) + B_d \epsilon(k) \quad (2.113)$$

The output $y(k)$ can be related to the state variable $\Delta x_m(k)$ by:

$$\begin{aligned}\Delta y(k+1) &= C_m \Delta x_m(k+1) \\ &= C_m A_m \Delta x_m(k) + C_m B_m \Delta u(k) + C_m B_d \epsilon(k)\end{aligned}\quad (2.114)$$

By defining a new state variable vector $x(k) = [\Delta x_m(k)^T \quad y(k)^T]^T$, the original plant model can be augmented with integrators as following:

$$\begin{aligned}\begin{bmatrix} \Delta x_m(k+1) \\ y(k+1) \end{bmatrix} &= \begin{bmatrix} A_m & 0_m^T \\ C_m A_m & I_{q \times q} \end{bmatrix} \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix} + \begin{bmatrix} B_m \\ C_m B_m \end{bmatrix} \Delta u(k) \\ &+ \begin{bmatrix} B_d \\ C_m B_d \end{bmatrix} \epsilon(k)\end{aligned}\quad (2.115)$$

$$y(k) = [o_m \quad I_{q \times q}] \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix}$$

Where $I_{q \times q}$ is identity matrix with $q \times q$ elements, and o_m is a $q \times n_1$ zero matrix.

For simplicity, the augmented model is represented as:

$$x(k+1) = Ax(k) + B\Delta u(k) + B_\epsilon \epsilon(k) \quad (2.116)$$

$$y(k) = Cx(k)$$

where

$$A = \begin{bmatrix} A_m & o_m^T \\ C_m A_m & I_{q \times q} \end{bmatrix}; B = \begin{bmatrix} B_m \\ C_m B_m \end{bmatrix}; \quad (2.117)$$

$$B_\epsilon = \begin{bmatrix} B_d \\ C_m B_d \end{bmatrix}; C = [o_m \quad I_{q \times q}]$$

The dimensionality of the augmented state-space equation is n which equals to $n_1 + q$.

Defining k_i as the sampling instant, then the current plant states is denoted by $x(k_i)$, and the future control trajectory is:

$$\Delta U = [\Delta u(k_i)^T \quad \Delta u(k_i + 1)^T \quad \dots \quad \Delta u(k_i + N_c - 1)^T]^T \quad (2.118)$$

Where N_c is called the control horizon, which represents the number of parameters used for the future control trajectory.

The future state variables are:

$$x(k_i + 1 | k_i) \quad x(k_i + 2 | k_i) \quad \dots \quad x(k_i + m | k_i) \quad x(k_i + N_p | k_i) \quad (2.119)$$

$x(k_i + m | k_i)$ means the predicted state variable at instant m given the current plant information. N_p is called the prediction horizon and represents the optimization window such that $N_c \leq N_p$.

To reduce the computational effort, the designing method selected for this study is Laguerre orthonormal functions based MPC. This method is proposed for applications where rapid system dynamics are required (Chipofya et al., 2015; Wang, 2001).

In this method, the control trajectory ΔU is expressed using a set of orthonormal functions called Laguerre functions. The Laguerre networks are known for their orthonormality. The z-transforms of the discrete-time Laguerre networks are written as:

$$\begin{aligned}\Gamma_1(z) &= \frac{\sqrt{1-a^2}}{1-az^{-1}} \\ \Gamma_2(z) &= \frac{\sqrt{1-a^2}}{1-az^{-1}} \frac{z^{-1}-a}{1-az^{-1}} \\ &\vdots \\ \Gamma_N(z) &= \frac{\sqrt{1-a^2}}{1-az^{-1}} \left(\frac{z^{-1}-a}{1-az^{-1}}\right)^{N-1}\end{aligned}\quad (2.120)$$

Where a is the pole of the discrete-time Laguerre network, it is also called the scaling factor, and is required to be selected by the user. For stability of the network, it should be $0 \leq a \leq 1$.

N is the order of the Laguerre network, and it is used to capture the control signal, it has a role similar to the control horizon in classical MPC.

The inverse z-transform of $\Gamma_N(z)$ is denoted by $l_N(k)$, and so the set of discrete time Laguerre functions for $i = 1, \dots, m$ inputs are represented in vector form as:

$$L_i(k) = [l_1(k) \quad l_2(k) \quad \dots \quad l_N(k)]^T \quad (2.121)$$

$L_i(k)$ can be solved as:

$$L_i(k+1) = A_{l_i} L_i(k) \quad (2.122)$$

Where A_{l_i} is $(N \times N)$ matrix and a function of a and $\beta = (1 - a^2)$

The initial condition of (2.121) is:

$$L_i(0) = \sqrt{\beta} [1 \quad -a \quad a^2 \quad \dots \quad (-1)^{N-1} a^{N-1}]^T \quad (2.123)$$

The Laguerre functions orthonormality is used in the design of discrete time MPC, and is represented by:

$$\sum_{k=0}^{\infty} l_i(k) l_j(k) = 0 \quad \text{for } i \neq j \quad (2.124)$$

$$\sum_{k=0}^{\infty} l_i(k) l_j(k) = 1 \quad \text{for } i = j \quad (2.125)$$

At time instant k_i , the control trajectory ΔU (2.118) is regarded as the impulse response of a stable dynamic system. Thus, a set of Laguerre functions, $l_1(k), l_2(k), \dots, l_N(k)$ can be used to capture this response with a set of Laguerre coefficients c_j that is determined during the design. At an arbitrary future time instant k , $\Delta u(k_i + k)$ is represented as:

$$\Delta u(k_i + k) = \sum_{j=1}^N c_j(k_i) l_j(k) = L_i(k)^T \eta_i \quad (2.126)$$

Where c_j are functions of the initial time instant of the moving horizon window k_i , and $\eta_i = [c_1 \quad c_2 \quad \dots \quad c_N]^T$.

The control horizon N_c from the classical MPC vanishes here, instead, the number of terms N and the parameter a are used to describe the control trajectory complexity.

The goal is now to find the optimal coefficient vector η that minimizes the cost function J , where J is written as:

$$J = \sum_{m=1}^{N_p} x(k_i + m | k_i)^T Q x(k_i + m | k_i) + \eta^T R \eta \quad (2.127)$$

The matrices Q and R are symmetric positive definite weighting matrices similar to those used in LQR optimal design.

Once the optimal coefficient vector η is found, the receding horizon control law is obtained using (2.126), which can be written in form of linear state feedback control as (Wang, 2009).

$$\Delta u(k) = -K_{mpc} x(k) \quad (2.128)$$

Where

$$K_{mpc} = L_i(0)^T \left(\left(\sum_{m=1}^{N_p} \phi(m) Q \phi(m)^T + R \right)^{-1} \sum_{m=1}^{N_p} \phi(m) Q A^m \right)$$

or

$$K_{mpc} = L_i(0)^T \Omega^{-1} \Psi \quad (2.129)$$

Where $\phi(m)^T = \sum_{i=0}^{m-1} A^{m-i-1} B L(i)^T$ is the convolution sum to compute the prediction of the system shown in (2.15), $\Omega = \sum_{m=1}^{N_p} \phi(m) Q \phi(m)^T + R$ and $\Psi = \sum_{m=1}^{N_p} \phi(m) Q A^m$.

Because the prediction of future states is based on the current information on $x(k_i)$, the set-point information is contained in $x(k_i)$ as follows:

$$x(k_i) = \begin{bmatrix} \Delta x_m(k_i) \\ e(k_i) \end{bmatrix} \quad (2.130)$$

$$e(k_i) = y(k_i) - r(k_i)$$

And the K_{mpc} gain can be rewritten and divided into two parts as:

$$K_{mpc} = [K_x \quad K_e] \quad (2.131)$$

Then the closed loop of discrete time MPC using Laguerre function is:

$$\begin{bmatrix} \Delta x_m(k+1) \\ y(k+1) \end{bmatrix} = (A - BK_{mpc}) \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix} + BK_e y(k) \quad (2.132)$$

$$y(k) = C \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix}$$

The operational constraints are known as a reason for performance deterioration of the control system when the control signals from the original design meet them. The ability to handle hard constraints is one of the main features of MPC. In this study the constraint on the difference of control variable and on the amplitudes of the control signal are studied. This investigation is motivated by the hard constrained mentioned for the experimental work of Block et al. (1997), where the dynamics of the motor are neglected, but the maximum possible flap angle is reported to be $\pm 32^\circ$ and the minimum motor increment is 0.016° . And it is assumed that the motor reacts exactly as specified, as long as the maximum velocity does not exceed 4.75 rad/s.

Using the Laguerre functions in the design, the incremental control signal is represented by:

$$\Delta u(k_i + m) = L(m)^T \eta \quad (2.133)$$

In principle, all the constraints are defined within the prediction horizon and processed as linear inequalities combined with the cost function. In other words, the optimization procedure is now to minimize the cost function J , while ensuring that:

$$\Delta u^{min} \leq \Delta u(k_i + m) \leq \Delta u^{max} \quad (2.134)$$

and

$$u^{min} \leq u(k_i + m) \leq u^{max} \quad (2.135)$$

2.5 Discrete Time Kalman Filter

As with the LQG, an estimator (observer) is required to estimate the state variables from the available output. If the pair A_m and C_m are observable, then:

$$\hat{x}(k_i + 1) = A\hat{x}(k_i) + B\Delta u(k_i) + K_{obs}(y(k_i) - C\hat{x}(k_i)) \quad (2.136)$$

Where K_{obs} is the gain matrix for Kalman filter, and $\hat{x}(k_i)$ is the current observer state.

As in the continues time Kalman filter, the gain K_{obs} is found by solving the discrete time Riccati equation (Outanoute et al., 2018; Wang, 2009).

$$P(k + 1) = APA^T - APC^T(R + CPC^T)^{-1}CPA^T + Q \quad (2.137)$$

Then $K_{obs}(k)$ can be calculated as:

$$K_{obs}(k) = APC^T(R + CPC^T)^{-1} \quad (2.138)$$

Where Q and R are the cost (weighting) matrices and are chosen by the designer.

2.6 System's Performance Measures and Indices

The controller's performance can be quantitatively evaluated by a set of performance measures and indices.

Three performance measures are used in this study to compare the different controllers step responses. The 10-90% rise time T_r to give an indication for the swiftness of the response. The percent overshoot $P.O.$ that gives information about the similarity or closeness with which the response matches the step input. And the settling time T_s which indicates how fast the system settle within 2% of the final value.

In addition to those measures, the integral of the square of the error, ISE, performance indicator is used. It is defined as:

$$ISE = \int_0^T e^2(t) dt \quad (2.139)$$

Where the error e is the difference between the input and output signals, as shown in Figure 10, and the integral upper limit T is the settling time (Dorf and Bishop, 2008).

In a similar way, the integral of the control action (ISU) associated with the input u is:

$$ISU = \int_0^T u^2(t) dt \quad (2.140)$$

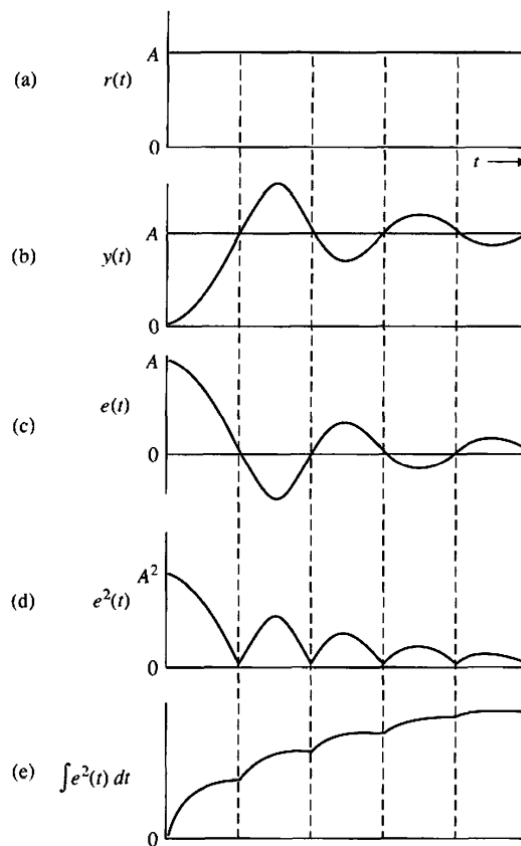


Figure 10: The calculation of the integral squared error ISE (Dorf and Bishop, 2008)

Chapter 3 : Results and Discussion

This Chapter will present the MATLAB® simulation results. Initially, the stability analysis of the open loop system, and flutter speed determination has been done using the mathematical model developed in Section 2.1, the system parameters used are taken from Conner et al. (1997) and shown in Table 1.

Table 1: System's numerical data for simulation (Conner et al., 1997).

Geometric parameters	
Chord	0.254 m
Span	0.52 m
Semi-chord, b	0.127 m
Elastic axis, a with respect to b	-0.5
Hinge line, c with respect to b	0.5
Mass parameters	
Mass of the wing	0.62868 kg
Mass of the aileron	0.18597 kg
Mass/length of the wing-aileron	0.1558 kg/m
Mass of support blocks	0.47485×2 kg
Inertial parameters	
S_α (per span)	0.08587 kg m
S_β (per span)	0.00395 kg m
x_α	0.434
x_β	0.01996
I_α (per span)	0.01347 kg m ²
I_β (per span)	0.0003264 kg m ²
r_α	0.7321
r_β	0.11397
κ	0.03984
Stiffness parameters	
K_α (per span)	14861 1/s ²
K_β (per span)	155 1/s ²
K_h (per span)	1809 1/s ²
Damping parameters	
ζ_α (log-dec)	0.01626
ζ_β (log-dec)	0.0115
ζ_h (log-dec)	0.0113

3.1 Open Loop Analysis

Using the eigenvalue analysis on the state space model, the modal damping (real part), and the oscillation frequencies (imaginary part) has been simulated for a constant air density (1.225 kg/m^3), in the velocity range at which the flutter is expected. The plot of the real part of eigen values (modal damping) with the air speed is shown in Figure 11. The instability (flutter speed) was found at the point where the modal damping passed through zero which was 23.96 m/s at a frequency of 6.12 Hz as presented in Figure 12.

To validate the mathematical model, the open loop analysis results can be compared to the numerical and experimental work of Conner et al. (1997). Where the numerical results are very close at 23.9 m/s and 6.112 Hz , while the experimental results are within 15% difference at 20.6 m/s and 5.47 Hz . According to Conner et al. (1997), this error is most likely due to the aerodynamic effects that are not modeled by Theodorsen model, and due to the three-dimensional aerodynamic effects in the wind tunnel.

Figure 11 shows that the damping of both modes initially increases (in negative direction), but at some point, the pitching mode damping continues to increase, while the plunge mode damping starts to decrease and becomes zero at the flutter speed of 23.96 m/s . after this critical speed, the plunge damping becomes positive (Prazenica, 2014).

From Figure 12, it is noticed that as the airspeed increases, the pitch and plunge frequencies of the system begin to approach each other near the flutter speed without

coalesce, but rather move close enough in frequency for the two modes to couple (Wright and Cooper, 2008).

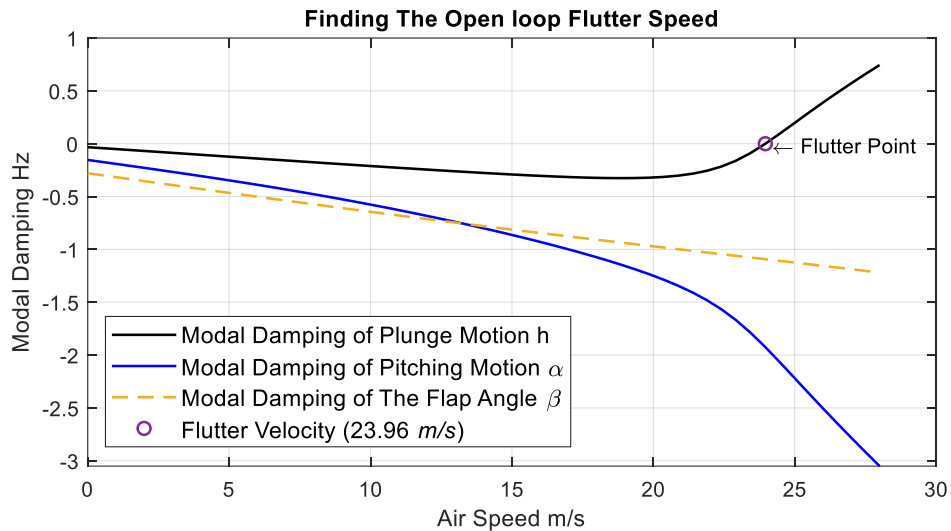


Figure 11: Modal damping at air speed range 0 – 28 m/s.

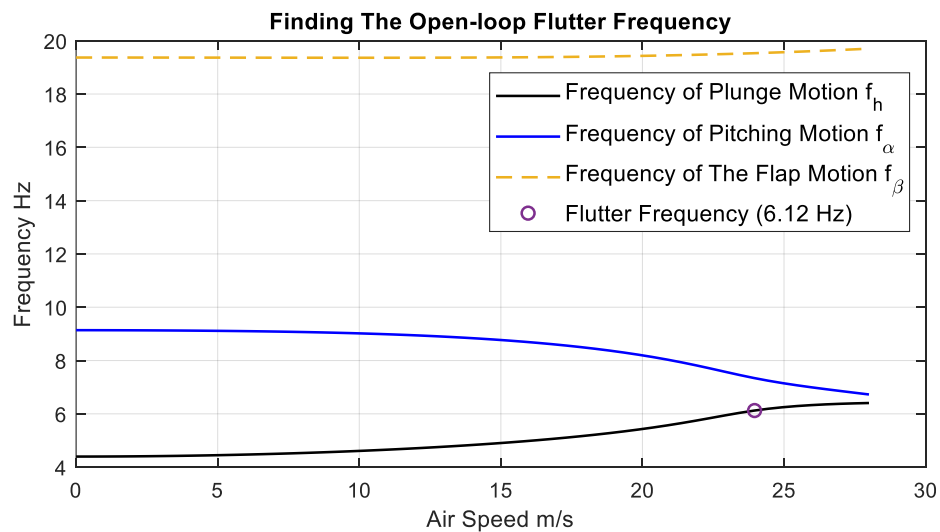


Figure 12: System's oscillation frequencies

To confirm the previous result, plunge, pitch, and flap angle system's response to initial condition (2° pitch angle disturbance) has been simulated in the time domain before flutter speed at 23.72 m/s and after flutter speed at 24.20 m/s. The results plotted in Figure 13. The plot shows that at less than the critical flutter speed, the oscillation

of the plunge, pitch and flap angle asymptotically approaches zero within few seconds. Hence, the aeroelastic system is stable at this free stream velocity. While at a speed of flow that is slightly higher than the critical flutter speed, the oscillations of the three degrees of freedom continue to grow without bound as time increase. In less than 6 seconds the plunge amplitude reached around 12 mm while the pitch angle crossed 4° . On other words a small accidental disturbance of the airfoil can act as a trigger to initiate an oscillation of great violence. It is obvious that applying such oscillation to a real physical system will eventually lead wing separation or damage (Fung, 2002).

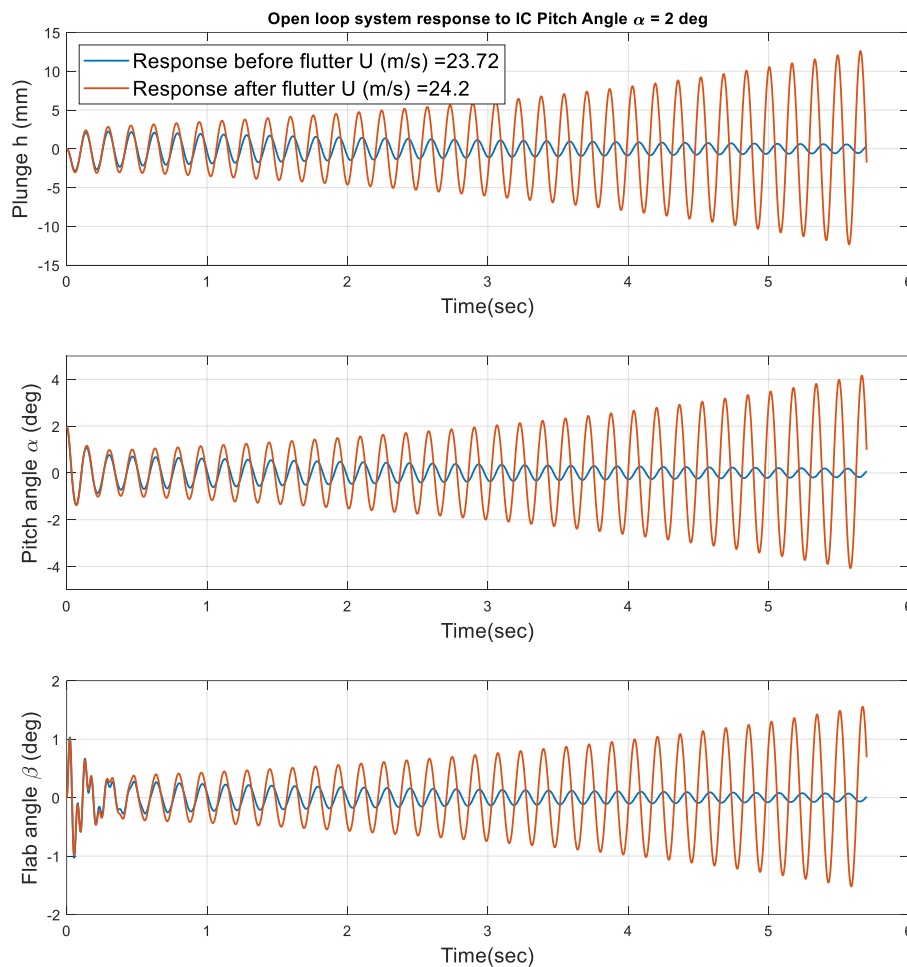


Figure 13: open loop system response to initial condition

Next, a reference input of 5° has been applied to the open loop system at a speed before that is less than the flutter critical speed. The open loop step response is plotted in Figure 14 below. The plot shows that the system failed to track the reference input. The output is stable but with oscillations that starts aggressively with high amplitudes and frequency then decay with time. In addition, the steady state error is quite high and unsatisfactory.

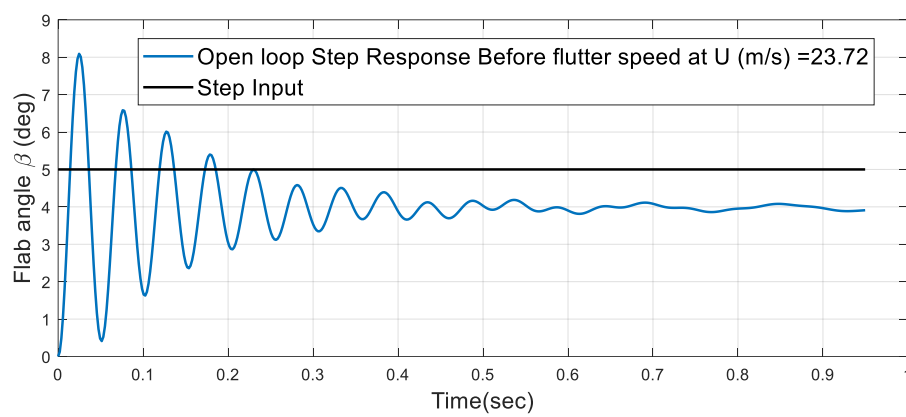


Figure 14: Open loop step response at a speed that is less than the critical flutter speed

3.2 Closed Loop Linear Quadratic Gaussian Compensator

The system that combines an optimal linear quadratic regulator, and an optimal observer (Kalman filter), is referred to as Linear Quadratic Gaussian compensator. In this section, The LQG closed system response to initial disturbance, and the response to step input were tuned and simulated at 25.52 m/s air speed, that is 10% higher than the critical flutter speed where the open loop system is unstable. The same system parameters used in the open loop analysis are also applied here.

The Simulink diagram for the aeroelastic system with LQG and integral action is shown in Figure 15.

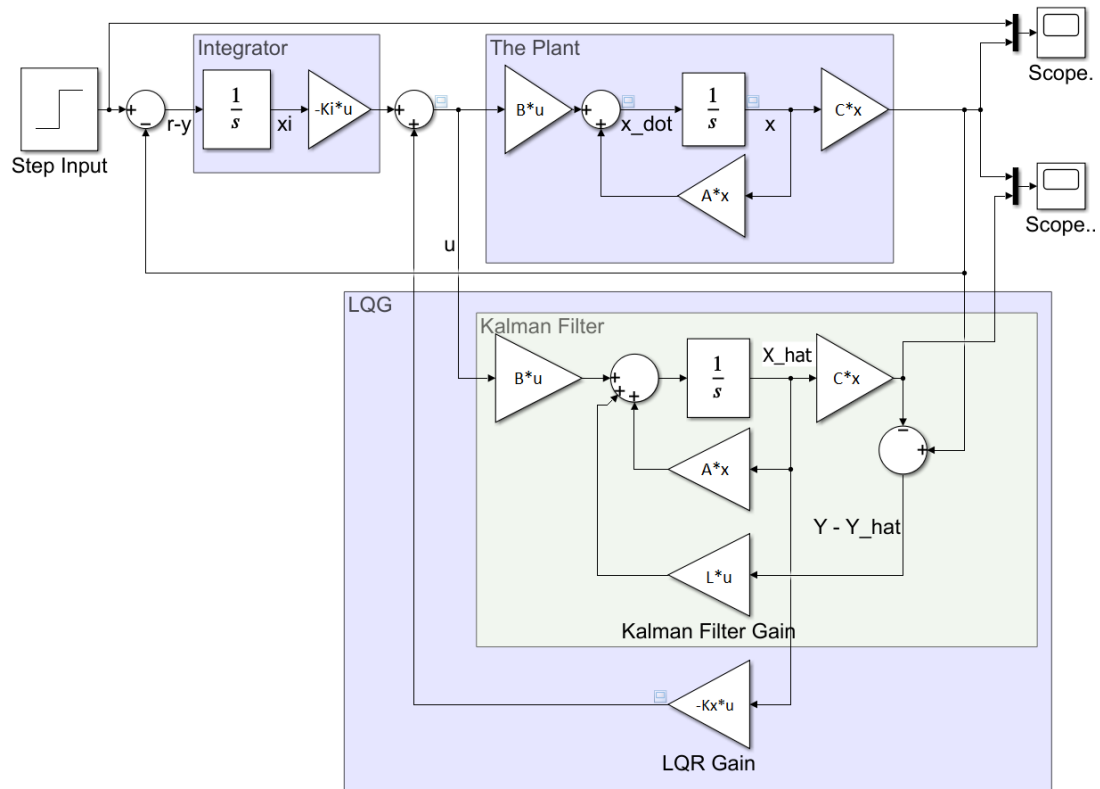


Figure 15: Simulink block diagram of the aeroelastic system with LQG controller and integrator

The controllability and observability were checked through MATLAB[®]. Where both of them were found to be full rank matrices, this means that a successful controller and observer can be designed.

Next, the LQR tuning process was performed through trial and error of different combinations for Q and R weighting matrices, where a quantitative analysis for each combination has been done, and the results are shown in Table 3 and Table 4 for the regulator and reference tracking cases respectively.

To set a target for the required performance, the actuator limitations presented in the experimental work of Block et al. (1997) were selected as limitations (physical constraints) for this study. These constraints are summarized in Table 2 below. So, the required performance is the fastest possible within these limitations.

Table 2: The control surface (flap) actuator physical constraints (Block et al., 1997)

Constraint description	Value
The maximum control surface deflection (\mathbf{u})	$\pm 32^\circ$
The minimum possible motor increment is	0.016°
The maximum velocity of control surface deflection $\frac{du}{dt}$	270 deg/s

For the optimal observer (Kalman filter), the process noise covariance Q_{kalman} was chosen to be 0.001 for all states, and the covariance R_{kalman} of the measurement noise was 0.01. Kalman filter estimates the process states by trading off between the measured and estimated data.

The effect of varying the Q value for each of the displacement states h , α , and β , appears from the results of trials 1 to 3 in Table 3. The (ISE) and settling time T_s for all state, in addition to the control effort (ISU) achieve the lowest value (best performance) when the weight Q_h of plunge displacement state is the highest among the other states. While results 3 to 5 show that although the value the control cost R has increased by five times, the impact on the indices value, and the maximum of control input are negligible.

Table 3: Quantitative analysis for the regulator performance as the states weight Q change

Trial No.	Q_h	Q_α	Q_β	R	T_s for h	ISE for h	T_s for α	ISE for α	T_s for β	ISE for β	ISU	$u(deg)$ max.	du/dt max
1	50	50	250	50	1.6	1.41	1.5	0.49	1.6	3.7	4.52	2.9	117
2	50	250	50	50	1.9	1.83	1.6	0.47	1.9	4.6	5.06	2.8	118
3	250	50	50	50	1.4	1.02	1.4	0.47	1.5	3.3	3.82	3.0	126
4	250	50	50	100	1.5	1.11	1.4	0.46	1.5	3.3	3.80	3.0	125
5	250	50	50	250	1.5	1.19	1.4	0.46	1.6	3.6	3.99	2.9	123

In summary, the regulator tuning trial 3 seems to be the best as it gives the fastest performance, and the lowest ISE.

Similar procedure has been used to tune and analyze the reference tracking case, where the results of the trials presented in Table 4 are all showing good performance with less than 0.7 seconds for settling time, and maximum overshoot of 12%. Furthermore, the required control inputs are within the systems physical limitations of Table 2.

Table 4: Quantitative analysis for the reference tracking performance as the control effort wight R changes.

Trial No.	Q_h	Q_a	Q_β	R	ISE for β	T_r	P.O. %	T_s	ISU	u max.	du/dt max.
1	250	50	50	50	0.78	0.09	11.2	0.64	2.78	7.1	260
2	250	50	50	100	0.85	0.10	7.65	0.63	2.22	6.9	185
3	250	50	50	250	0.92	0.11	4.11	0.56	2.20	6.7	117

From Table 4, it can be noticed that the effect of the control cost value R , on the performance of reference tracking is more significant than that on the regulator performance. The cheaper the control cost, the faster is the Rise time, but this comes with higher overshoot, and higher control ISU.

The tuning combination of trial 4 from Table 3 is used to plot the regulator response to initial disturbance and the results are presented in Figure 16 and Figure 17, it is clear that the closed loop system is asymptotically stable. As the values of the three position states and the three rate states have settled in around 1.5 seconds. In addition, Figure 16 shows that the Kalman filter has successfully converged to the actual state values, where state estimates matched the actual states perfectly in less than 0.5 seconds.

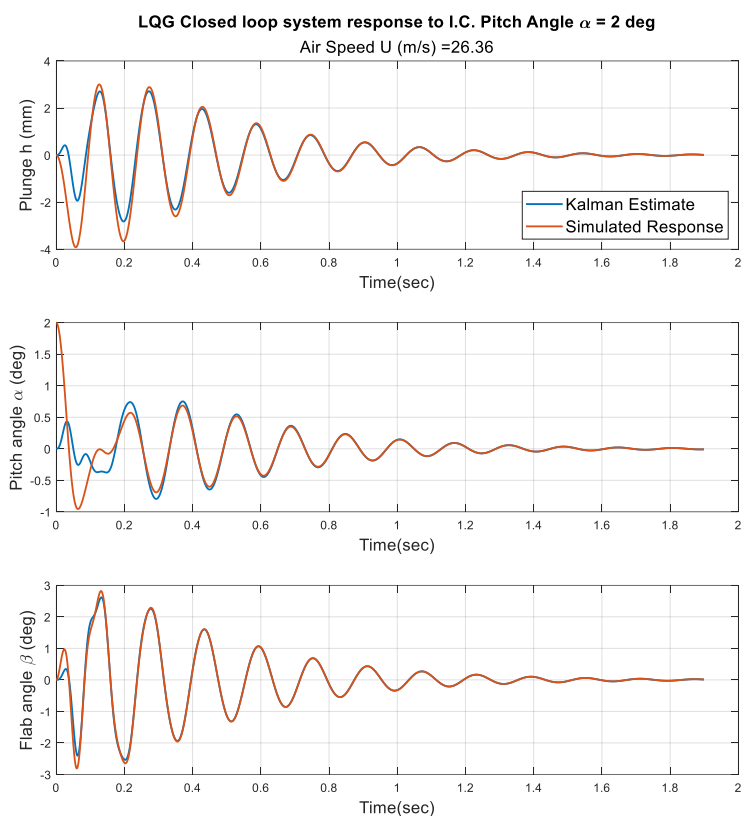


Figure 16: LQG closed loop response to initial disturbance – Displacement states

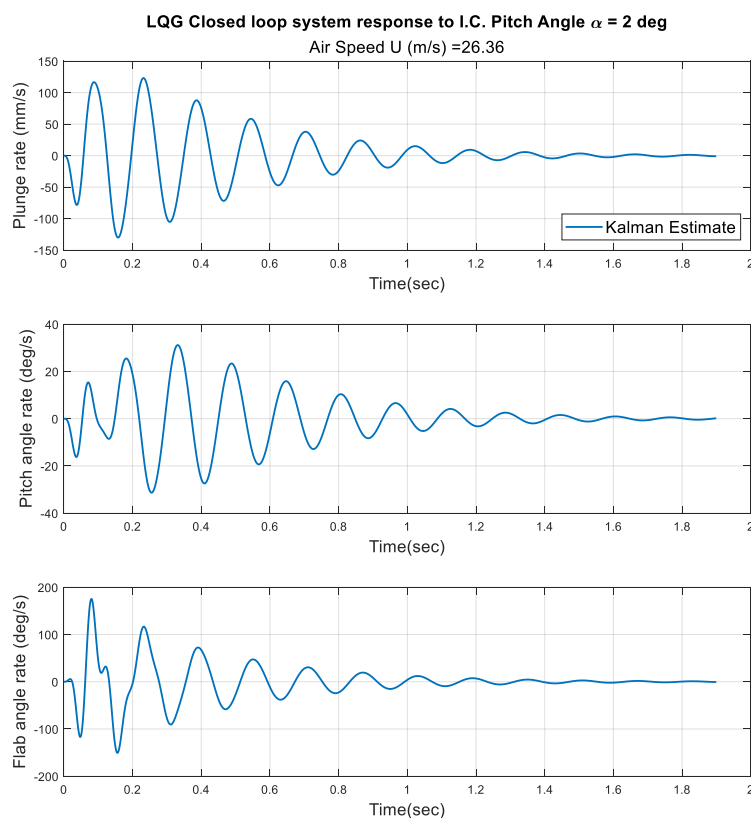


Figure 17: LQG closed loop response to initial disturbance – Rate states

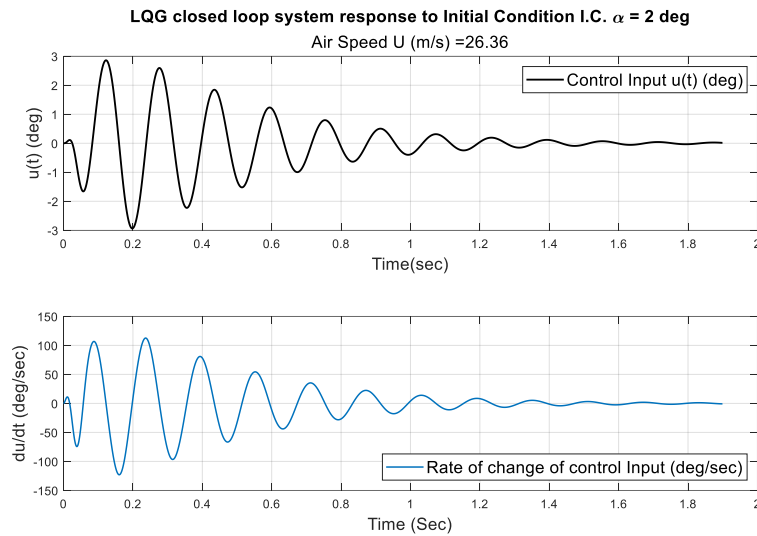


Figure 18: LQG closed loop response to initial disturbance – Control signal

Next, the flap angle tracking case has been simulated using LQG compensator with integral action to obtain a constant output with zero steady state error.

Using the same stream speed of 25.52 m/s, and the tuning combination of trial 2 in Table 4, the response to a 5° step input is plotted in Figure 19, and the control input u with its time rate of change $\frac{du}{dt}$ are in Figure 20.

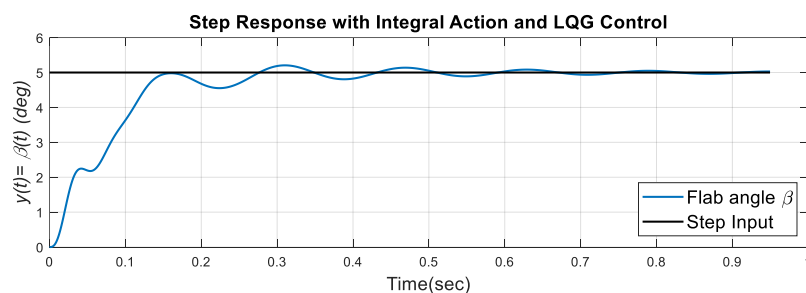


Figure 19: Step response results using LQG compensator with integral action

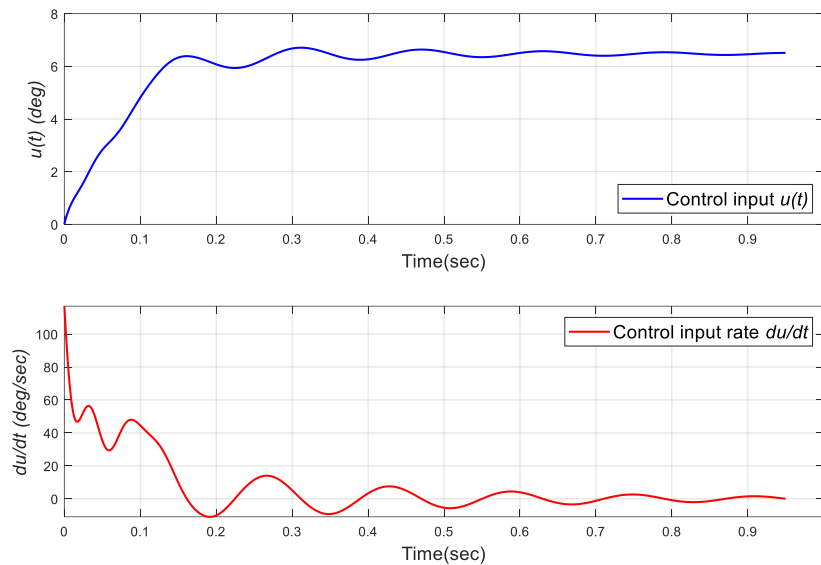


Figure 20: The control input and its rate of change - reference tracking case

From the plots, it is obvious that the compensator simulation has successfully derived the flap to the desired position in a very short time (0.6 sec), with zero steady state error while maintaining the stability of the system.

3.3 Closed Loop Discrete Model Predictive Control using Laguerre Functions

As with LQG, the closed system response to initial disturbance, and the reference tracking cases were simulated at 26.36 m/s air speed. The system parameters used here are similar to those used in the open loop and LQG analysis.

MPC tuning was performed through varying the control cost R value and setting the input constraints. Then deciding the best performance by the quantitative analysis as shown in Table 5 and Table 6 for the regulator, and reference tracking cases.

Table 5: Quantitative analysis for MPC regulator performance with and without constraints

Trial No.	R	$u(deg)$ Input Const.	du/dt Input Const.	T_s for h	ISE for h	T_s for α	ISE for α	T_s for β	ISE for β	ISU	Actual $u(deg)$ max.	Actual du/dt max.
1	25	No	No	0.6	0.02	0.7	0.02	0.5	0.16	0.14	11.8	2,000
2	50	No	No	0.8	0.03	0.8	0.02	0.6	0.17	0.15	7.9	1,194
3	25	10	105	0.8	0.03	0.8	0.02	0.7	0.13	0.16	3.9	105
4	50	10	105	0.8	0.03	0.8	0.02	0.8	0.14	0.16	3.7	105

From Table 5, the performance is in general higher when control cost R value is less, this result is expected as it comes with higher values for the maximum control input and control input rate. The control input rate in trials 1 and 2 is much higher than the control input limits (constraints) set in Table 2.

Moving to trials 3 and 4, it's clear that in the presence of the controller imbedded input constraints, the significance of the control cost R tuning is much less, as there is almost no difference in performance between these two trials.

The plots of Figure 21 and Figure 22 show the simulated states of the three degree of freedom displacements and rates of change. Using the parameters of trials 2 and 4 from Table 5. The system states in both trials were successfully and very quickly driven to zero in around 0.8 seconds only after a disturbance of 2° at the pitch angle value. Nevertheless, with the unconstrained MPC the state's amplitudes are higher at the beginning.

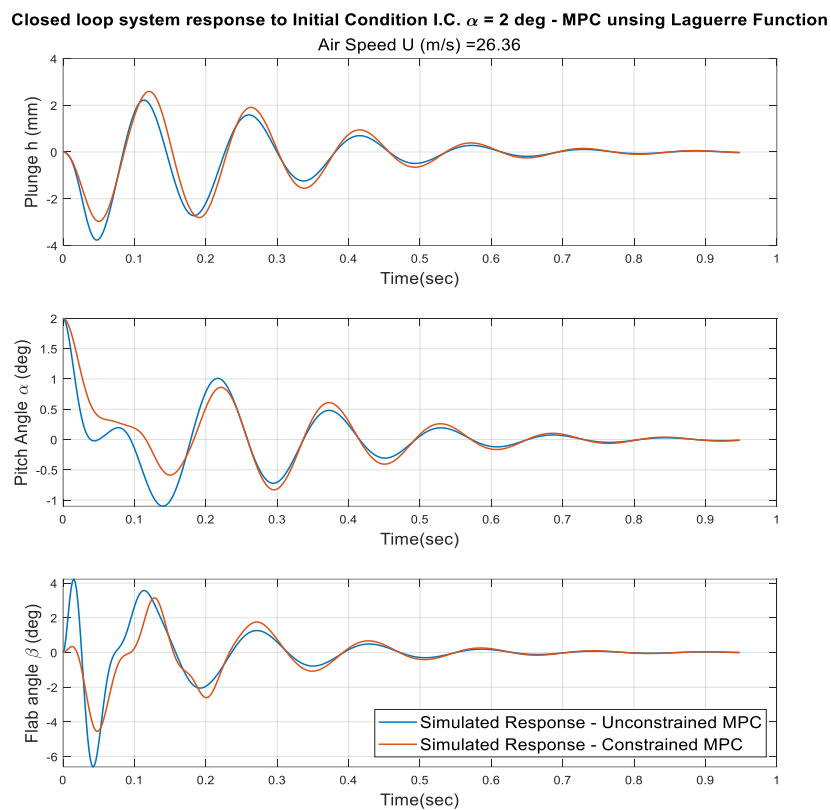


Figure 21: MPC using Laguerre functions - Response to initial disturbance – Displacement States

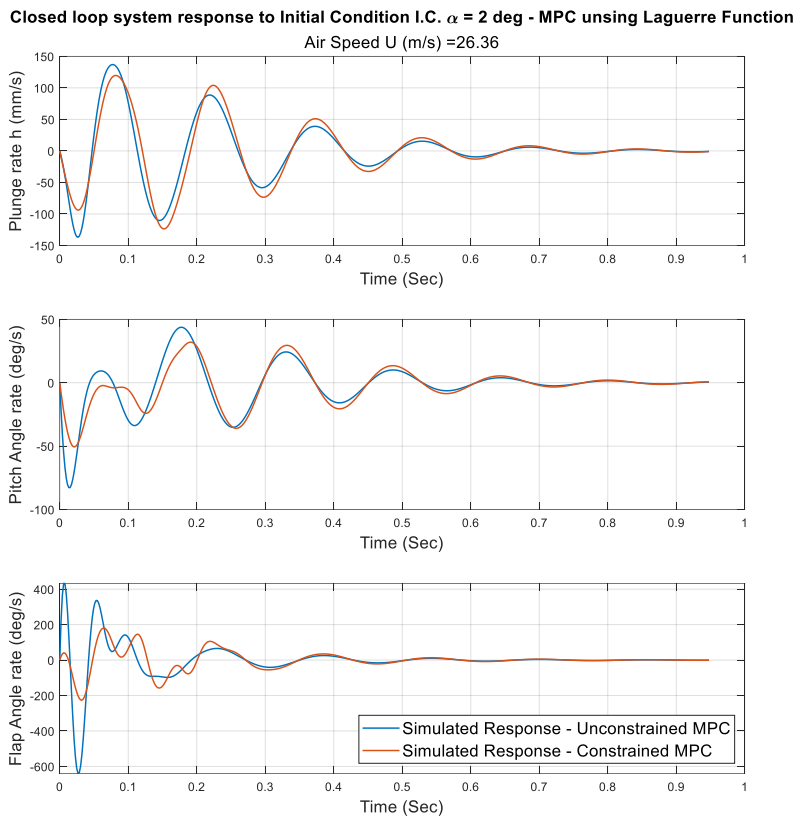


Figure 22: MPC using Laguerre functions - Response to initial disturbance – Rate of change states

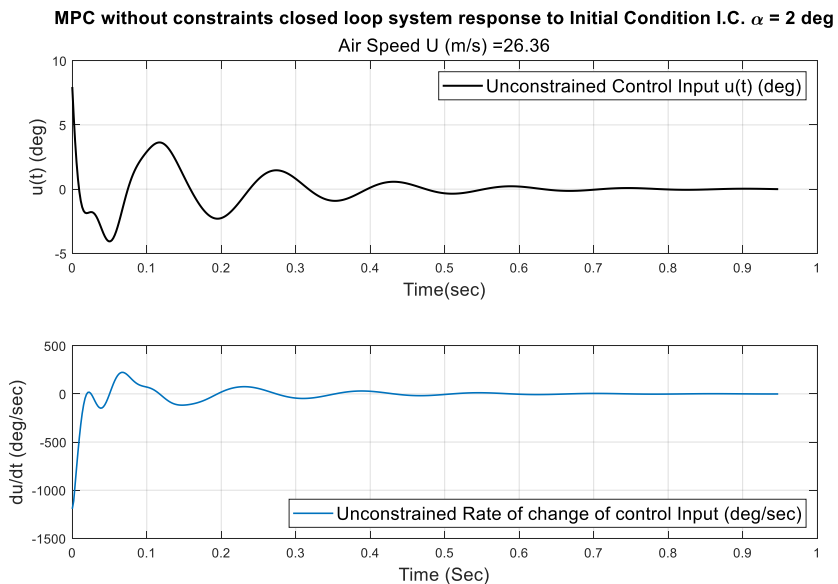


Figure 23: MPC – System’s response to initial disturbance - Unconstrained control input and its rate of change

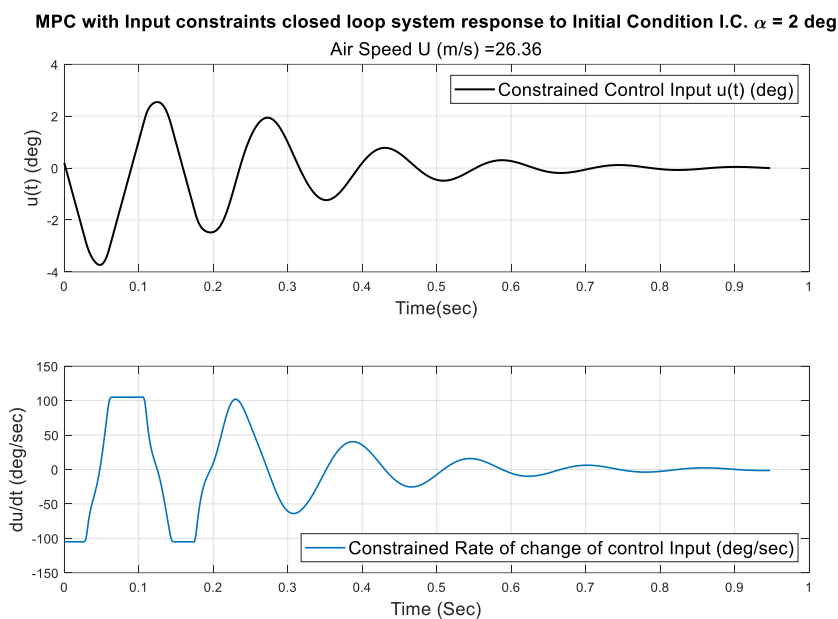


Figure 24: MPC – System’s response to initial disturbance - Constrained control input and its rate of change

Figure 23 and Figure 24 present the unconstrained and constrained control input and input rate respectively. Where the unconstrained input rate reaches 1,194 degrees per second, which exceeds the physical limitations of the actuator as mentioned earlier. On the other hand, the constrained input rate is way less at 105 degrees per second only without negative impact on the controller performance.

Table 6: Quantitative analysis for MPC reference tracking performance

Trial No.	R	$u(deg)$ Input Const.	du/dt Input Const.	ISE for β	T_r	P.O. %	T_s	ISU	u max.	du/dt max.
1	25	No	No	0.04	0.08	21	0.46	0.17	7.6	332
2	50	No	No	0.04	0.08	19	0.47	0.17	7.5	263
3	25	10	105	0.05	0.09	20	0.47	0.18	7.6	105
4	50	10	105	0.06	0.09	18	0.48	0.18	7.5	105

The MPC controller with the same choice of parameters and tuning procedure shown in Table 6 is next applied to the reference tracking case. Table 6 and Figure 25

show that the response is very fast with a settling time of less than 0.5 seconds, and overshoot that does not exceed 20%. These results are very satisfactory.

Similar to the regulator case, Figure 26 and Figure 27 show that the control input rate exceeds the physical limitations when there is no constrained applied at the controller. While by using the MPC feature of systematically handling physical constraints, the input signal remains within the allowable limit without effecting the controller's performance.

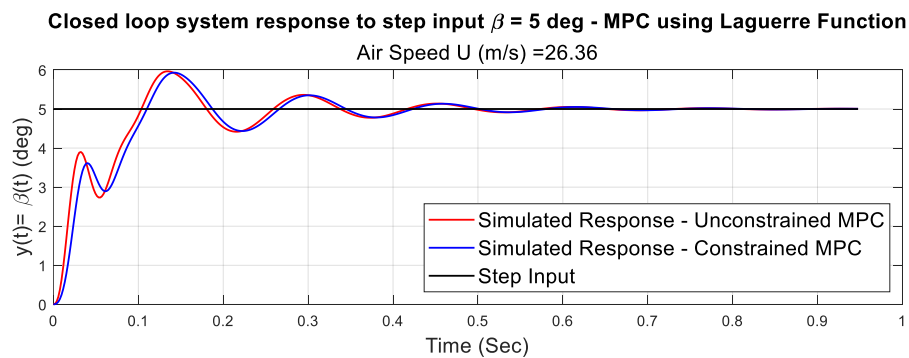


Figure 25: MPC– Reference tracking - Input and output simulated signals

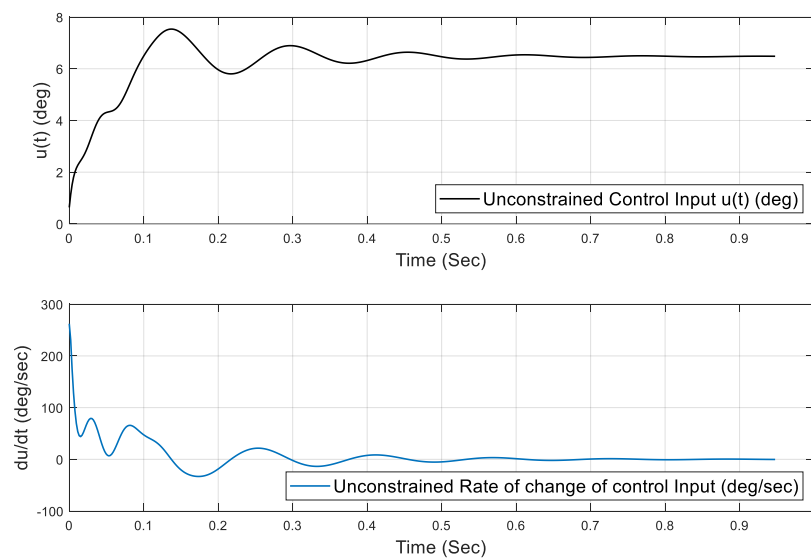


Figure 26: MPC– Reference tracking - Unconstrained control input and its rate of change

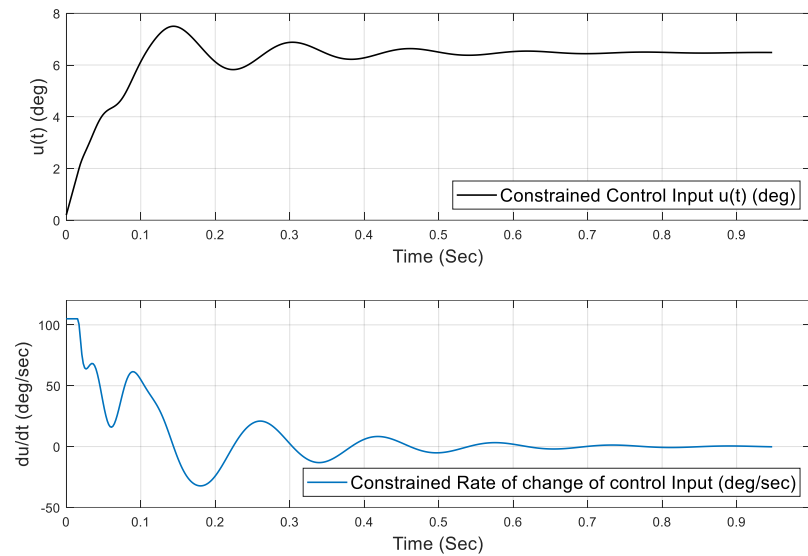


Figure 27: MPC– Reference tracking - Constrained control input and its rate of change

3.4 Controllers Comparison

The results presented in the previous sections has proofed that both the optimal controller LQG and constrained MPC using Laguerre functions are both capable of suppressing the wing flutter with satisfactory performance. The designs that showed the best performance from the results of both controllers are compared in Table 7 and Table 8.

It is obvious that the constrained MPC using Laguerre functions outperforms the LQG in both regulation and reference tracking cases. MPC regulator has achieved more than 40% less settling time with much less control energy indicated by the ISU value.

Although the difference in settling time is not significant when comparing the reference tracking case, it can be observed from Table 8 that the power consumption, associated to the ISU index, and the tracking performance, associated to the ISE index

were much smaller for the constrained MPC. This means that MPC is solving the same problem more efficiently and using less power to do so.

Table 7: LQG and constrained MPC - Regulator - Quantitative comparison

Trial No.	Controller	T_s for h	ISE for h	T_s for α	ISE for α	T_s for β	ISE for β	ISU	u max.	du/dt max.
1	LQG	1.4	1.02	1.4	0.47	1.5	3.3	3.82	3.0	126
2	Constrained MPC	0.8	0.03	0.8	0.02	0.7	0.13	0.16	3.9	105

Table 8: LQG and constrained MPC – Reference tracking - Quantitative comparison

Trial No.	Controller	ISE for β	T_r	P.O. %	T_s	ISU	u max.	du/dt max.
1	LQG	0.92	0.11	4.11	0.56	2.20	6.7	117
2	Constrained MPC	0.05	0.09	20	0.47	0.18	7.6	105

Chapter 4 : Conclusions and Future Work

4.1 Conclusions

This study aimed to investigate the effectiveness of using discrete MPC in suppressing the flutter of a two-dimensional wing with a control surface (flap). It started with the derivation of the full eight state $(\dot{h}, \dot{\alpha}, \dot{\beta}, h, \alpha, \beta, \ell_1, \ell_2)$ aeroelastic system equations of motion using the Lagrange's energy method and Theodorsen's method for the unsteady aerodynamic forces. The system has then been converted to the state space representation. The open loop flutter speed and frequency were evaluated using the eigenvalue analysis, which was found to be 23.96 m/s at a frequency of 6.12 Hz for the selected system parameters. These results were compared in section 3.1 to the previous simulation and experimental work done by Conner et al. (1997) for model verification, where a good match was confirmed.

The open loop responses to initial disturbance, before and after the flutter critical speed, were examined through MATLAB[®] numerical simulation, where the disturbance after flutter speed generated an unstable oscillation with diverging amplitude that would cause separation of wing or damage if applied to a real physical structure. Furthermore, even though the oscillation generated before flutter speed was convergent, it dissipated very slowly to the level that may also lead to structural harm with repetition. In addition, trying to derive the flap angle of the open loop system to a desired position (step response) has also failed, where it started with aggressive oscillation and ended with a respectively high steady state error.

Next, a LQG compensator was designed and simulated to stabilize the system in the flutter region and derive the control surface (flap) to any desired angle accurately

with zero steady state error. After a tuning process, the numerical simulation results demonstrated a very satisfactory results under the selected physical limitations.

Later, a constrained MPC controller using Laguerre function was designed and simulated for the same system. From to the regulator case results, it can be noticed that the MPC outperformed the LQG by saving more than 40% of the states settling time while driving them to zero after an initial disturbance. In addition, MPC achieved that with much less control energy indicated by the ISU index.

Although settling time required by MPC for tracking the step input was only slightly less than the that of LQG, but the tracking goal was accomplished with much less power consumption, associated to the ISU index, and better tracking performance, associated to the ISE index. In addition, the ability to systematically deal with constrains was also investigated, this feature alone gives a strong preference reason to predictive control over other approaches. As the designer does not have to worry about the performance deterioration due control signal saturation when it run into system physical limits.

4.2 Future Work

Due to the precautionary measures associated with the spread of COVID-19 during the study period, experimental work to verify the results could not be performed. So future work has to include the experimental verification, with further fine-tuning of the Laguerre functions, and MPC parameters to try to find the best possible performance. Furthermore, the analysis and comparison between the two studied controllers in terms of robustness and adding gain scheduling to cover different operational speeds could be done.

Although this study focused on a three degree of freedom linear flutter model, it revealed the benefits of using active control, and the advantages of the model predictive control with an aeroelastic system. To move some more steps in the direction of applying active flutter control in aerospace industry, the research efforts have to be directed towards more complex mathematical model and controller. There are many undiscussed other factors that may make affect the system performance. In this study, the mathematical model derivation has been done assuming incompressible flow, this could be valid at low speeds only. The flutter analysis and the predictive controller performance at the supersonic and hypersonic regimes can make a good subject for future work. In addition to the system nonlinearities which was not discussed. Considering these factors in future studies result in more practical Aeroservoelasticity system and take the research a few steps forward.

Furthermore, redundancy must be taken into consideration while designing the active flutter suppression system to avoid the catastrophic results in case of system failure. Other factors like the effect of the active flutter suppression on the aerodynamic forces and the interaction with other systems have also to be considered.

References

- Abdelkefi, A., Vasconcellos, R., Nayfeh, A. H., and Hajj, M. R. (2013). An analytical and experimental investigation into limit-cycle oscillations of an aeroelastic system. *Nonlinear Dynamics*, 71(1), 159–173.
<https://doi.org/10.1007/s11071-012-0648-z>
- Akmeşe, A. (2006). *Aeroservoelastic analysis and robust controller synthesis for flutter suppression of air vehicle control actuation systems*. The Middle East Technical University, Turkey.
- Ashish, T. (2002). *Modern control design with Matlab and simulink*. Indian Institute of Technology, Kanpur, India, John Wiley and Sons.
- Barker, J. M., Balas, G. J., and Blue, P. A. (1999). Gain-scheduled linear fractional control for active flutter suppression. *Journal of Guidance, Control, and Dynamics*, 22(4), 507–512.
- Belo, E. M., and Rocha JR, J. E. D. M. (2001). C. “A Fuzzy controller for active flutter suppression”. *Proceedings of 9th International Symposium on Dynamic Problems of Mechanics, Florianópolis/SC, Brazil*, 9–14.
- Bemporad, A., Morari, M., and Ricker, N. L. (2014). *Model predictive control toolbox*. The MathWorks, Inc.
- Bhoir, N., and Singh, S. N. (2004). Output feedback nonlinear control of an aeroelastic system with unsteady aerodynamics. *Aerospace Science and Technology*, 8(3), 195–205.
- Block, J., Gilliatt, H., Block, J., and Gilliatt, H. (1997). Active control of an aeroelastic structure. *35th Aerospace Sciences Meeting and Exhibit*, 16.
- Borglund, D., and Kuttenukeuler, J. (2002). Active flutter suppression using a trailing edge flap. *Journal of Fluids and Structures*, 16(3), 271–294.
<https://doi.org/10.1006/jfls.2001.0426>
- Boscariol, P., Gasparetto, A., and Zanotto, V. (2010). Active position and vibration control of a flexible links mechanism using model-based predictive control. *Journal of Dynamic Systems, Measurement, and Control*, 132(1).
- Brunton, S. (2017). *Control Bootcamp: Full-State Estimation*.
https://www.youtube.com/watch?v=MZJMi-6_4UU&list=PLMrJAKhIeNNR20Mz-VpZgfQs5zrYi085mandindex=17
- Chen, C.-T. (2013). *Linear System Theory and Design*. Oxford University Press.

- Chipofya, M., Lee, D. J., and Chong, K. T. (2015). Trajectory tracking and stabilization of a quadrotor using model predictive control of Laguerre functions. *Hindawi Publishing Corporation, Abstract and Applied Analysis*, 2015, 12. <http://dx.doi.org/10.1155/2015/916864>
- Conner, M. D., Tang, D. M., Dowell, E. H., and Virgin, L. N. (1997). Nonlinear behavior of a typical airfoil Section with control surface freeplay: a numerical and experimental study. *Journal of Fluids and Structures*, 11(1), 89–109. <https://doi.org/10.1006/jfls.1996.0068>
- Cunha-Filho, A. G., de Lima, A. M. G., Donadon, M. V., and Leão, L. S. (2016). Flutter suppression of plates using passive constrained viscoelastic layers. *Mechanical Systems and Signal Processing*, 79, 99–111. <https://doi.org/10.1016/j.ymsp.2016.02.025>
- De Marqui, C., Belo, E. M., and Marques, F. D. (2005). A flutter suppression active controller. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 219(1), 19–33. <https://doi.org/10.1243/095441005X9102>
- De Marqui Junior, C., Rebolho, D. C., Belo, E. M., and Marques, F. D. (2006). Identification of flutter parameters for a wing model. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 28(3). <https://doi.org/10.1590/S1678-58782006000300012>
- Dimitriadis, G. (2017). *Introduction to nonlinear aeroelasticity*. John Wiley and Sons.
- Dorf, R. C., and Bishop, R. H. (2008). *Modern control systems*. Pearson Prentice Hall.
- Edwards, J. W., and Wieseman, C. D. (2008). Flutter and divergence analysis using the generalized aeroelastic analysis method. *Journal of Aircraft*, 45(3), 906–915.
- Eversman, W., and Roy, I. D. (1997). Active flutter suppression using multi-input/multi-output adaptive least mean square control. *Journal of Aircraft*, 34(2), 244–250.
- Eversman, W., and Tewari, A. (1991). Modified exponential series approximation for the Theodorsen function. *Journal of Aircraft*, 28(9), 553–557.
- Fung, Y. C. (2002). *An Introduction to the Theory of Aeroelasticity*. Dover Publications.
- Garrard, W. L., and Liebst, B. S. (1985). Active flutter suppression using eigenspace and linear quadratic design techniques. *Journal of Guidance, Control, and Dynamics*, 8(3), 304–311.

- Glad, T., and Ljung, L. (2018). *Control Theory*. CRC Press.
- Hoadley, S. T., and Karpel, M. (1991). Application of aeroservoelastic modeling using minimum-state unsteady aerodynamic approximations. *Journal of Guidance, Control, and Dynamics*, 14(6), 1267–1276.
<https://doi.org/10.2514/3.20783>
- Hodges, D. H., and Pierce, G. A. (2011). *Introduction to Structural Dynamics and Aeroelasticity* (2nd ed.). Cambridge University Press.
<https://doi.org/10.1017/CBO9780511997112>
- Holkar, K. S., and Waghmare, L. M. (2010). Discrete model predictive control for DC drive using orthonormal basis function. *UKACC International Conference on CONTROL 2010*, 435–440.
<https://doi.org/10.1049/ic.2010.0322>
- Hopwood, J. W., Ruskin, B., Broderick, D. J., and Wei, F.-S. (2019, January 7). Aeroservoelastic design and wind tunnel testing using parameter-varying optimal control and inertial-based sensing. *AIAA Scitech 2019 Forum*. AIAA Scitech 2019 Forum, San Diego, California. <https://doi.org/10.2514/6.2019-0608>
- Horikawa, H., and Dowell, E. H. (1979). An Elementary explanation of the flutter mechanism with active feedback controls. *Journal of Aircraft*, 16(4), 225–232. <https://doi.org/10.2514/3.58509>
- Hovland, G. (2004). *Introduction to Kalman filtering*. METR4200 – Advanced Control. <https://www.scribd.com/document/500629483/METR4200-8>
- Karpel, M. (1982). Design for active flutter suppression and gust alleviation using state-space aeroelastic modeling. *Journal of Aircraft*, 19(3), 221–227.
- Kassem, M., Yang, Z., Gu, Y., Wang, W., and Safwat, E. (2020). Active dynamic vibration absorber for flutter suppression. *Journal of Sound and Vibration*, 469, 115110. <https://doi.org/10.1016/j.jsv.2019.115110>
- Kehoe, M. W. (1995). *A historical overview of flight flutter testing*. NASA Hugh L. Dryden Flight Research Center Edwards, CA, United States.
<https://ntrs.nasa.gov/api/citations/19960004074/downloads/19960004074.pdf>
- Livne, E. (2018). Aircraft active flutter suppression: State of the art and technology maturation needs. *Journal of Aircraft*, 55(1), 410–452.
- Mahesh, J. K., Stone, C. R., Garrard, W. L., and Dunns, H. J. (1981). Control law synthesis for flutter suppression using linear quadratic gaussian theory. *Journal of Guidance and Control*, 4(4), 415–422.

- Marchetti, L., De Gaspari, A., Riccobene, L., Toffol, F., Fonte, F., Ricci, S., Mantegazza, P., Livne, E., and Hinson, K. A. (2020). Active flutter suppression analysis and wind tunnel studies of a commercial transport configuration. 1677. <https://doi.org/10.2514/6.2020-1677>
- Marretta, R. A., and Marino, F. (2007). Wing flutter suppression enhancement using a well-suited active control model. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 221(3), 441–452.
- Matter, Y. S., Darabseh, T. T., and Mourad, A.-H. I. (2018). Effect of engine location on flutter speed and frequency of a tapered viscoelastic wing. *IOP Conference Series: Materials Science and Engineering*, 370, 012014. <https://doi.org/10.1088/1757-899X/370/1/012014>
- Na, M.-G. (2001). A model predictive controller for the water level of nuclear steam generators. *Nuclear Engineering and Technology*, 33(1), 102–110.
- Naidu, D. S. (2002). *Optimal Control Systems*. CRC Press.
- Nikolaou, M. (2001). Model predictive controllers: A critical synthesis of theory and industrial needs. [https://doi.org/10.1016/S0065-2377\(01\)26003-7](https://doi.org/10.1016/S0065-2377(01)26003-7)
- Olds, S. D. (1997). Modeling and LQR control of a two-dimensional airfoil [PhD Thesis]. Virginia Tech, Virginia, USA. <http://hdl.handle.net/10919/36668>
- Outanoute, M., Selmani, A., Guerbaoui, M., Ed-dahhak, A., Lachhab, A., and Bouchikhi, B. (2018). Predictive control algorithm using Laguerre functions for greenhouse temperature control. *International Journal of Control and Automation*, 11(10), 11–20.
- Pinheiro, T. C. F., and Silveira, A. S. (2021). Constrained discrete model predictive control of an arm-manipulator using Laguerre function. *Optimal Control Applications and Methods*, 42(1), 160–179.
- Prazenica, R. J. (2014). Model predictive control of a nonlinear aeroelastic system using reduced-order volterra models. *Aiaa Atmospheric Flight Mechanics Conference*, 2188. <https://doi.org/10.2514/6.2014-2188>
- Ricketts, R. H. (1983). Structural testing for static failure, flutter and other scary things (Vol. 84606). Langley Research Center.
- Roger, K. L., Hodges, G. E., and Felt, L. (1975). Active Flutter Suppression-A Flight Test Demonstration. *Journal of Aircraft*, 12(6), 551–556. <https://doi.org/10.2514/3.59833>

- Rossiter, J. A. (2003). *Model-based predictive control: A practical approach*. CRC press.
- Rossiter, J. A., Wang, L., and Valencia-Palomo, G. (2010). Efficient algorithms for trading off feasibility and performance in predictive control. *International Journal of Control*, 83(4), 789–797.
- Scott, R. C., and Pado, L. E. (2000). Active control of wind-tunnel model aeroelastic response using neural networks. *Journal of Guidance, Control, and Dynamics*, 23(6), 1100–1108.
- Seborg, D. E., Mellichamp, D. A., Edgar, T. F., and Doyle III, F. J. (2010). *Process dynamics and control*. John Wiley and Sons.
- Silva, S. da, and Lopes Júnior, V. (2006). Active flutter suppression in a 2-D airfoil using linear matrix inequalities techniques. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 28(1), 84–93.
- Simon, D. (2006). *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley and Sons.
- Sutherland, A. N. (2008). A Small scale pitch-plunge flutter model for active flutter control research. 26th International Congress of the Aeronautical Science. http://www.icas.org/ICAS_ARCHIVE/ICAS2008/ABSTRACTS/385.HTM
- Sutherland, A. N. (2010). A demonstration of pitch-plunge flutter suppression using LQG control. International Congress of the Aeronautical Sciences, ICAS. https://www.icas.org/ICAS_ARCHIVE/ICAS2010/PAPERS/459.PDF
- Sutherland, A. N. (2011). *Aeroservoelastic Analysis, Design and Wind Tunnel Testing of a Three Degree-of-Freedom Binary Flutter Model*. University of the Witwatersrand, South Africa.
- Tewari, A. (2011). *Automatic control of atmospheric and space flight vehicles: design and analysis with MATLAB® and Simulink®*. Springer Science and Business Media.
- Tewari, A. (2015). *Aeroservoelasticity: modeling and control*. Springer-Verlag. <https://doi.org/10.1007/978-1-4939-2368-7>
- Theis, J., Pfifer, H., and Seiler, P. J. (2016). Robust control design for active flutter suppression. *AIAA Atmospheric Flight Mechanics Conference*, 1751.
- Theodorsen, Theodore (1935). Report No. 496, general theory of aerodynamic instability and the mechanism of flutter. *Journal of the Franklin Institute*, 219(6), 766–767. [https://doi.org/10.1016/S0016-0032\(35\)92022-1](https://doi.org/10.1016/S0016-0032(35)92022-1)

- Vinodh Kumar, E., and Jerome, J. (2013). Robust LQR controller design for stabilizing and trajectory tracking of inverted pendulum. *Procedia Engineering*, 64, 169–178. <https://doi.org/10.1016/j.proeng.2013.09.088>
- Wang, L. (2004). Discrete model predictive controller design using Laguerre functions. *Journal of Process Control*, 14(2), 131–142.
- Wang, L. (2009). *Model predictive control System Design and Implementation Using MATLAB®*. Springer-Verlag. <https://doi.org/10.1007/978-1-84882-331-0>
- Wang, L. (2001). Discrete time model predictive control design using Laguerre functions. *Proceedings of the 2001 American Control Conference*. (Cat. No. 01CH37148), 3, 2430–2435.
- Waszak, M. (1996). Modeling the benchmark active control technology wind-tunnel model for application to flutter suppression. *21st Atmospheric Flight Mechanics Conference*, 3437.
- Wright, J. R., and Cooper, J. E. (2008). *Introduction to aircraft aeroelasticity and loads (Vol. 20)*. John Wiley and Sons.
- Xie, B., Mao, M., Zhou, L., Wan, Y., and Hao, G. (2020). Systematic design of linear quadratic regulator for digitally controlled grid-connected inverters. *IET Power Electronics*, 13(3), 557–567. <https://doi.org/10.1049/iet-pel.2019.0514>
- Yehezkeley, E., and Karpel, M. (1996). Nonlinear flutter analysis of missiles with pneumatic fin actuators. *Journal of Guidance, Control, and Dynamics*, 19(3), 664–670.
- York, D. L. (1980). *Analysis of flutter and flutter suppression via an energy method [Thesis, Virginia Tech, USA]*. <https://vtechworks.lib.vt.edu/handle/10919/43300>
- Zeng, J., Kukreja, S. L., and Moulin, B. (2012). Experimental model-based aeroelastic control for flutter suppression and gust-load alleviation. *Journal of Guidance, Control, and Dynamics*, 35(5), 1377–1390.
- Zhang, P. (2010). Chapter 2—Industrial control engineering. In P. Zhang (Ed.), *Advanced Industrial Control Technology* (pp. 41–70). William Andrew Publishing. <https://doi.org/10.1016/B978-1-4377-7807-6.10002-6>

Appendices

Appendix A: Theodorsen's Functions

These functions are required to include the effect of a control surface on the aerodynamics and so the flutter dynamics of the model (Theodorsen, 1935).

$$T_1 = -\frac{1}{3}\sqrt{1-c^2}(2+c^2) + c \cos^{-1}(c) \quad (\text{A.01})$$

$$T_2 = c(1-c^2) - \sqrt{1-c^2}(1+c^2) \cos^{-1}(c) + c[\cos^{-1}(c)]^2 \quad (\text{A.02})$$

$$T_3 = -\left(\frac{1}{8} + c^2\right) [\cos^{-1}(c)]^2 + \frac{1}{4}c\sqrt{1-c^2} \cos^{-1}(c)(7+2c^2) - \frac{1}{8}(1-c^2)(5c^2+4) \quad (\text{A.03})$$

$$T_4 = -\cos^{-1}(c) + c\sqrt{1-c^2} \quad (\text{A.04})$$

$$T_5 = -(1-c^2) - [\cos^{-1}(c)]^2 + 2c\sqrt{1-c^2} \cos^{-1}(c) \quad (\text{A.05})$$

$$T_6 = T_2 \quad (\text{A.06})$$

$$T_7 = -\left(\frac{1}{8} + c^2\right) \cos^{-1}(c) + \frac{1}{8}c\sqrt{1-c^2} (7+2c^2) \quad (\text{A.07})$$

$$T_8 = -\frac{1}{3}\sqrt{1-c^2}(2c^2+1) + c \cos^{-1}(c) \quad (\text{A.08})$$

$$T_9 = \frac{1}{2} \left[\frac{1}{3}(1-c^2)^{\frac{3}{2}} + aT_4 \right] \quad (\text{A.09})$$

$$T_{10} = \sqrt{1-c^2} + \cos^{-1}(c) \quad (\text{A.10})$$

$$T_{11} = \cos^{-1}(c)(1-2c) + \sqrt{1-c^2} (2-c) \quad (\text{A.11})$$

$$T_{12} = \sqrt{1-c^2}(2+c) - \cos^{-1}(c)(1+2c) \quad (\text{A.12})$$

$$T_{13} = \frac{1}{2} [-T_7 - (c-a)T_1] \quad (\text{A.13})$$

$$T_{14} = \frac{1}{16} + \frac{1}{2}ac \quad (\text{A.14})$$

Appendix B: MATLAB® Codes

The MATLAB codes used in the analysis and simulation of this study results are listed in this appendix, a summary and brief description about each code is in the table below

No.	File Name	Description
01	<i>Main_LQG.m</i>	The main code for the open loop system analysis in addition to the LQG closed loop controller design and analysis
02	<i>Main_UCLMPC.m</i>	The main code for discrete MPC using Laguerre functions <i>without constraints</i> controller design and simulation
03	<i>Main_CLMPC.m</i>	The main code for discrete MPC using Laguerre functions <i>with input constraints</i> controller design and simulation
04	<i>Airfoil.m</i>	The airfoil parameters used in the simulation are saved here
05	<i>Modal_Data.m</i>	A function that calculates and sorts the open loop eigenvalues then finds the open loop flutter frequency and speed
06	<i>SS_Matrices.m</i>	A function to find the state space matrices for the system at a specific air speed and density
07	<i>Flutter_LQI.m</i>	A function to find the gain matrices for the LQR controller, the Kalman filter, and the integral action.
08	<i>dmpc.m</i>	A function for generating the data matrices used in the design of the discrete MPC using Laguerre functions controller
09	<i>lagd.m</i>	A function to generates the initial condition of the Laguerre function, and the state space system matrix A1
10	<i>simuucob.m</i>	A function for MIMO closed loop MPC simulation, using Kalman filter, without constraints.
11	<i>simucob.m</i>	A function for MIMO closed loop MPC simulation, using Kalman filter, with constraints on the control input and its rate of change.
12	<i>ISEPerformance.m</i>	A function to calculate The Integral of Square of the Error ISE Performance index

```

%-----%
%----- Main_LQG.m -----%
%-----%
% Open Loop system analysis and LQG system design
close all
clear
clc
%----- Input Data -----%

rho = 1.225; % Ambient air density [kg/m^3]
U = linspace(0, 28, 56), % Airspeed range to analyses [m/s]

%----- Find Open loop flutter speed -----%

Airfoil;
load('Airfoil'),

[U_f, f_f, f_h, f_a, f_B, g_h, g_a, g_B,1] = Modal_Data(U, rho),

disp('Open loop flutter speed is (m/s):')
disp(U_f)
disp('Open loop flutter frequency is (Hz):')
disp(f_f)

%----- Plots of Flutter speed -----%

% Plot the open loop Modal damping (real part) against the Air speed range

figure (1)
plot(U,g_h,'k',U,g_a,'b',U,g_B,'--',U_f,0,'o','LineWidth',1.2)
txt = ' \leftarrow Flutter Point';
text(U_f,0,txt)
grid on
title('Finding The Open loop Flutter Speed', 'FontSize', 11)
xlabel('Air Speed m/s')
ylabel('Modal Damping Hz')
legend('Modal Damping of Plunge Motion h','Modal Damping of Pitching Motion \alpha','Modal Damping of The Flap Angle \beta', ['Flutter Velocity (' , num2str(U_f, '%.2f'), ' \it{m/s}\rm)'], 'FontSize', 11,'Location','southwest')

% Plot the open loop oscillation frequencies (imaginary part) against the Air speed range

figure (2)
plot(U,f_h,'k',U,f_a,'b',U,f_B,'--',U_f,f_f,'o','LineWidth',1.2),
grid on
title('Finding The Open-loop Flutter Frequency', 'FontSize', 11)
xlabel('Air Speed m/s')
ylabel('Frequency Hz')

legend('Frequency of Plunge Motion f_{h}','Frequency of Pitching Motion f_{\alpha}','Frequency of The Flap Motion f_{\beta}', ['Flutter Frequency (' , num2str(f_f, '%.2f'), ' Hz)'], 'FontSize', 11)
%----- Open loop system Analysis -----%

% Initial conditions for open loop system response

X_0 = [0; % Initial plunge rate
0; % Initial pitch rate
0; % Initial control surface rate
0; % Initial plunge displacement
2*pi/180; % Initial pitch displacement [rad.]
0; % Initial control surface angle [rad.]
0; % Initial aerodynamic lag (1st state)
0]; % Initial aerodynamic lag (2nd state)

states = {'h_dot' 'Alpha_dot' 'Beta_dot' 'h' 'Alpha' 'Beta' 'l1' 'l2'};

```

```

inputs = {'Aileron'};
outputs = {'\beta'};

tau = 0:0.1:300; % set the dimensionless time range
t=tau./w_na; % transform the dimensionless time to real time to use in plots
r =ones(size(tau))*0; % define the input matrix

% the state space matrices before flutter speed
BF_U = U_f*0.99; % The air speed before flutter is set at 99% of the flutter
speed
[A_b,B_b,C_b,D_b]= SS_Matrices (BF_U,rho),
BF_sys = ss(A_b,B_b,C_b,D_b, 'statename', states, ...
    'inputname', inputs, 'outputname', outputs),

% I.C. open loop system response before flutter
[y_BF,tau,x_BF]=lsim(BF_sys,r,tau,X_0),

% The state space matrices after flutter speed

AF_U=U_f*1.01; % The air speed after flutter is set at10% higher than
flutter speed
[A_f,B_f,C_f,D_f]= SS_Matrices (AF_U,rho),
AF_sys = ss(A_f,B_f,C_f,D_f, 'statename', states, ...
    'inputname', inputs, 'outputname', outputs),

[y_AF,tau,x_AF]=lsim(AF_sys,r,tau,X_0),

%----- Open loop I.C response Beta = 2 deg -----%
figure (3)
subplot (311)
plot(t,x_BF(:,4)*b*1000, t,x_AF(:,4)*b*1000,'LineWidth',1.2)
grid
xlabel('Time(sec)', 'FontSize', 14)
ylabel('Plunge h (mm)', 'FontSize', 14)
legend(strcat('Response before flutter U (m/s) = ',num2str(round
(BF_U,2))),strcat('Response after flutter U (m/s) = ',num2str(round
(AF_U,2))), 'FontSize', 14 , 'Location' , 'northwest')
title('Open loop system response to IC Pitch Angle \alpha = 2 deg',
'FontSize', 11)
subplot (312)
plot(t,x_BF(:,5)*180/pi, t,x_AF(:,5)*180/pi,'LineWidth',1.2)
grid
xlabel('Time(sec)', 'FontSize', 14)
ylabel('Pitch angle \alpha (deg)', 'FontSize', 14)
subplot (313)
plot(t,x_BF(:,6)*180/pi, t,x_AF(:,6)*180/pi,'LineWidth',1.2)
grid
xlabel('Time(sec)', 'FontSize', 14)
ylabel('Flab angle \beta (deg)', 'FontSize', 14)

%-----Open loop Step-response Beta = 2 deg -----%

X_00 = [0; % Initial plunge rate
0; % Initial pitch rate
0; % Initial control surface rate
0; % Initial plunge displacement [m]
0; % Initial pitch displacement [rad.]
0; % Initial control surface angle [rad.]
0; % Initial aerodynamic lag (1st state)
0]; % Initial aerodynamic lag (2nd state)

tau = 0:0.1:200; % the dimensionless time
t=tau/w_na;
r =ones(size(tau))*(5*pi/180),

% Open loop step system response before flutter
[y_BF,tau,x_BF]=lsim(BF_sys,r,tau,X_00),

```

```

figure (4)
plot(t,x_BF(:,6)*180/pi,t,r*180/pi, 'k','LineWidth',1.5)
grid
xlabel('Time(sec)', 'FontSize', 14)
ylabel('Flap angle \beta (deg)', 'FontSize', 14)
legend(strcat('Open loop Step Response Before flutter speed at U (m/s) = ',num2str(round (BF_U,2))), 'Step Input' , 'Location' , 'southeast', 'FontSize',14)

% Create a pole-zero plot of the Flutter system
figure (5)
pzmap(AF_sys),
title('Pole-Zero Map after flutter speed (Open-loop system)')

%----- Closed Loop - LQG Controller Design -----%

%controllability and Observability

disp('Checking Controllability:')
unco=length (A_f)-rank(ctrb(AF_sys)),
if (unco==0)
    disp('System is controllable!')
else
    disp(['Number of uncontrollable states are:',unco])
end

disp('Checking Observability:')
unobsv=length (A_f)-rank(obsv(AF_sys)),
if (unobsv==0)
    disp('System is Observable!')
else
    disp(['Number of unobservable states are:',unobsv])
end

%----- LQG closed loop -----%

U = linspace(0, 50, 200), % Airspeed range for closed loop analysis [m/s]

% the state space matrices after flutter speed

% prompt = 'Enter the Air Speed (m/s): ';
% AF_U = input(prompt),

AF_U=U_f*1.1; % The air speed after flutter is set at10% higher than flutter speed

[A,B,C,D]= SS_Matrices (AF_U,rho),
AF_sys = ss(A,B,C,D, 'statename', states, ...
    'inputname', inputs, 'outputname', outputs),
% choose Q and R for LQR controller

    Q = zeros (9,9),
    Q (4,4)=250; %weight for h state (plunging)
    Q (5,5)=50; %weight for alpha state (pitching)
    Q (6,6)=50; %weight for beta state(flap angle)

    Q (9,9)=50; %weight for the input integral action gain

    R =250; %weight for control effort

[Ki, Kx, L]= Flutter_LQI (AF_sys, A,C,Q,R),% finding the closed loop gain matrices

% The LQG with integrator closed loop matrices

```

```

Ac = [A -B*Kx -B*Ki; L*C A-B*Kx-L*C -B*Ki; -C 0 0 0 0 0 0 0 0 0];
Bc = [0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 1];
Cc = [C 0 0 0 0 0 0 0 0 0];
Dc = [D];
sys_cl = ss(Ac,Bc,Cc,Dc),

% ----- Stability (Eigen value Map) -----%
figure (6)
pzmap(sys_cl),
title('Pole-Zero Map after flutter speed (closed-loop system)', 'FontSize',
11)

%----- Response to initial condition (Regulator case) -----%

% set the I.C with pitch displacement disturbance

X_0_cl = [0; % Initial plunge rate
0; % Initial pitch rate
0; % Initial control surface rate
0; % Initial plunge displacement
2/(180/pi), % Initial pitch displacement [rad.]
0; % Initial control surface angle [rad.]
0; % Initial aerodynamic lag (1st state)
0; % Initial aerodynamic lag (2nd state)
0;0;0;0;0;0;0;0;0]; % zero initials for Kalman filter and integrator states

tau = 0:0.01:150; % Setting the dimensionless time range
t=tau/w_na; % Transforming the dimensionless time to real time (sec)
r =ones(size(tau))*0;% Define the input matrix (zeros) for regulator case

%simulate the closed loop system
[y,tau,x]=lsim(sys_cl,r,tau, X_0_cl),

% plot the actual vs observer states for regulator case

figure (7)% Plots of The Displacement states h, Alpha, Beta
subplot (3,1,1), plot(t,x(:,12)*b*1000, t,x(:,4)*b*1000,'LineWidth',1.5)
grid
xlabel('Time(sec)', 'FontSize', 14)
ylabel('Plunge h (mm)', 'FontSize', 14)
legend('Kalman Estimate', 'Simulated Response', 'Location', 'southeast',
'FontSize',14)
title('LQG Closed loop system response to I.C. Pitch Angle \alpha = 2
deg',strcat('Air Speed U (m/s) = ',num2str(round (AF_U,2))), 'FontSize',14)
subplot (3,1,2), plot(t,x(:,13)*180/pi, t,x(:,5)*180/pi,'LineWidth',1.5)
grid
xlabel('Time(sec)', 'FontSize', 14)
ylabel('Pitch angle \alpha (deg)', 'FontSize', 14)
subplot (3,1,3), plot(t,x(:,14)*180/pi, t,x(:,6)*180/pi,'LineWidth',1.5)
grid
xlabel('Time(sec)', 'FontSize', 14)
ylabel('Flab angle \beta (deg)', 'FontSize', 14)

%Finding ISE Performance indices for the displacement states
StepData_h=stepinfo(x(:,4)*b*1000, t, 0),
T_h=StepData_h.SettlingTime;
ISE_h = ISEPerformance(x(:,4),r,t,T_h),% plunge state regulation ISE
performance index

StepData_alpha=stepinfo(x(:,5)*180/pi, t, 0),
T_alpha=StepData_alpha.SettlingTime
ISE_alpha = ISEPerformance(x(:,5),r,t,T_alpha),% pitch state regulation ISE
performance index

StepData_beta=stepinfo(x(:,6)*180/pi, t, 0),
T_beta=StepData_beta.SettlingTime;

```

```

ISE_beta = ISEPerformance(x(:,6),r,t,T_beta),% Flap angle state regulation
ISE performance index

figure (8) % Plots of The Rate states h_dot, Alpha_dot and Beta_dot
subplot (3,1,1), plot(t,x(:,9)*b*1000*w_na, 'LineWidth',1.5)
grid
xlabel('Time(sec)', 'FontSize', 14)
ylabel('Plunge rate (mm/s)', 'FontSize', 14)
legend('Kalman Estimate', 'Location', 'southeast', 'FontSize',14)
title('LQG Closed loop system response to I.C. Pitch Angle \alpha = 2
deg',strcat('Air Speed U (m/s) = ',num2str(round (AF_U,2))), 'FontSize',14)
subplot (3,1,2), plot(t,x(:,10)*(180/pi)*w_na, 'LineWidth',1.5)
grid
xlabel('Time(sec)', 'FontSize', 14)
ylabel('Pitch angle rate (deg/s)', 'FontSize', 14)
subplot (3,1,3), plot(t,x(:,11)*(180/pi)*w_na, 'LineWidth',1.5)
grid
xlabel('Time(sec)', 'FontSize', 14)
ylabel('Flap angle rate (deg/s)', 'FontSize', 14)

% Finding the control signal u and control signal rate of change
u=-Kx(1)*(x(:,9))-Kx(2)*x(:,10)-Kx(3)*x(:,11)-Kx(4)*(x(:,12))-Kx(5)*x(:,13)-
Kx(6)*x(:,14)-Kx(7)*x(:,15)-Kx(8)*x(:,16)-Ki*x(:,17),

du=gradient(u(:))./gradient(tau(:)), % Finding the rate of change of control
signal

u_max_reg=max(abs(u))*180/pi;
du_max_reg=max(abs(du))*(180/pi)*w_na;

figure (9)% Plot of the control signal u and control signal rate of change
subplot (211)
plot(t,u*180/pi, 'k', 'LineWidth',1.5)
xlabel('Time(sec)', 'FontSize', 14)
ylabel('u(t) (deg)', 'FontSize', 14)
title('LQG closed loop system response to Initial Condition I.C. \alpha = 2
deg',strcat('Air Speed U (m/s) = ',num2str(round (AF_U,2))), 'FontSize',14)
legend('Control Input u(t) (deg)', 'FontSize', 14)
grid
subplot(212)
plot(t,du*(180/pi)*w_na, 'LineWidth',1.2)
xlabel('Time (Sec)', 'FontSize', 14)
ylabel('du/dt (deg/sec)', 'FontSize', 14)
legend('Rate of change of control Input (deg/sec)', 'Location', 'southeast',
'FontSize',14)
grid

StepData_u_reg=stepinfo(u*180/pi, t, 0),
T_u_reg=StepData_u_reg.SettlingTime
ISU_Reg = ISEPerformance(u,r,t,T_u_reg)% regulation input ISU performance
index

%-----Step Response-----%

% Define zero initial condition matrix for reference tracking case
X_0_cl = [0; % Initial plunge rate
0; % Initial pitch rate
0; % Initial control surface rate
0; % Initial plunge displacement
0; % Initial pitch displacement [rad.]
0; % Initial control surface angle [rad.]
0; % Initial aerodynamic lag (1st state)
0; % Initial aerodynamic lag (2nd state)
0;0;0;0;0;0;0;0;0];

```

```

tau = 0:0.01:50; % the dimensionless time (50 is 1 sec)
r =ones(size(tau))*(5*pi/180),% define the step input matrix for Tracking
case

% Simulate the closed loop tracking case
[y_t,tau,x]=lsim(sys_cl,r,tau, X_0_cl),
t=tau./w_na; % transform the dimensionless time to real time to use in plots

% Finding the control signal u and control signal rate of change
u_t=-Kx(1)*(x(:,9))-Kx(2)*x(:,10)-Kx(3)*x(:,11)-Kx(4)*(x(:,12))-
Kx(5)*x(:,13)-Kx(6)*x(:,14)-Kx(7)*x(:,15)-Kx(8)*x(:,16)-Ki*x(:,17),
du_t=gradient(u_t(:))./gradient(tau(:)),

u_max_t=max(abs(u_t))*180/pi;
du_max_t=max(abs(du_t))*(180/pi)*w_na;

figure (10)% Plots of Step response output, input signal, and input rate
signal
subplot (3,1,1), plot(t,y_t*180/pi,t,r*180/pi, 'k','LineWidth',1.5)
xlabel('Time(sec)', 'FontSize', 14)
ylabel('\ity(t)= \beta(t) (deg)', 'FontSize', 14)
title('Step Response with Integral Action and LQG Control', 'FontSize', 14)
legend('Flab angle \beta ', 'Step Input', 'FontSize', 14, 'Location'
, 'southeast')
grid
subplot (3,1,2), plot(t,u_t*180/pi, 'b','LineWidth',1.5)
xlabel('Time(sec)', 'FontSize', 14)
ylabel('\itu(t) (deg)', 'FontSize', 14)
legend('Control input \itu(t)', 'FontSize', 14, 'Location' , 'southeast')
grid
subplot (3,1,3), plot(t,du_t*(180/pi)*w_na, 'r','LineWidth',1.5)
xlabel('Time(sec)', 'FontSize', 14)
ylabel('\itdu/dt (deg/sec)', 'FontSize', 14)
legend('Control input rate \itdu/dt', 'FontSize', 14, 'Location'
, 'northeast')
grid

StepData_t=stepinfo(y_t*180/pi, t, 5),
T_t=StepData_t.SettlingTime;
ISE_t = ISEPerformance(y_t,r,t,T_t)% tracking ISE performance index

StepData_u_t=stepinfo(u_t*180/pi, t),
T_u_t=StepData_u_t.SettlingTime;
ISU_t = ISEPerformance(u_t,r,t,T_u_t),% Tracking input ISU performance index

%===== The End of the Code =====%

```

```

%-----%
%----- Main_UCLMPC.m -----%
%-----%
% 2D wing with control Flap Flutter suppression using Discreate MPC with
% Laguerre functions and Kalman filter - No constraints
close all
clear
clc
%----- Input Data -----%

rho = 1.225; % Ambient air density [kg/m^3]
U = linspace(0, 25, 50), % Airspeed range to analyses [m/s]

%----- Find Open loop flutter speed -----%

Airfoil;
load('Airfoil'),

[U_f, f_f, f_h, f_a, f_B, g_h, g_a, g_B,l] = Modal_Data(U, rho),

disp('Open loop flutter speed is (m/s):')
disp(U_f)
disp('Open loop flutter frequency is (Hz):')
disp(f_f)

%----- Set the Air speed and the state space Matrices -----%

% prompt = 'Enter the Air Speed (m/s): ';
% AF_U = input(prompt),

AF_U =U_f*1.1;% 10% higher than flutter speed

states = {'h_dot' 'Alpha_dot' 'Beta_dot' 'h' 'Alpha' 'Beta' 'l1' 'l2'};
inputs = {'Aileron'};
outputs = {'\beta'};

[Ac,Bc,Cc,Dc]= SS_Matrices (AF_U,rho),

%----- Discretize the system -----%

Delta_t = 0.1;
[Ad, Bd, Cd, Dd]=c2dm(Ac,Bc,Cc,Dc,Delta_t),

%find the integrator augmented matrices
[m1,n1]=size(Cd), %m1 is number of outputs and n1 is number of states
[n1,n_in]=size(Bd), % n_in is number of inputs

A=eye(n1+m1, n1+m1),
A(1:n1,1:n1)=Ad;
A(n1+1:n1+m1,1:n1)=Cd*Ad;
B=zeros(n1+m1,n_in),
B(1:n1,:)=Bd;
B(n1+1:n1+m1,:)=Cd*Bd;
C=zeros(m1,n1+m1),
C(:,n1+1:n1+m1)=eye(m1,m1),

[n, n_in]=size(B),

%----- Set the MPC parameters -----%

Np=500; %prediction horizon was 200
a=0.3;
N=16;
Q=C'*C;
R=50*eye(n_in,n_in),%tunning matrix for control effort

N_sim=500;%number of simulation points

```



```

%----- Find Omega and Psi -----%

[Omega, Psi]=dmpc(A,B,a,N,Np,Q,R),

%Laguerre matrix
[Al,L0]=lagd(a,N),
Lzerot=L0';

K_mpc=L0'*(Omega\Psi),

% ----- Stability (Eigen value comparison) -----%

Acl=A-B*K_mpc;
[X,Y,Z]=dlqr(A,B,Q,R),

figure (1)
viscircles([0 0],1, 'Color','k', 'LineWidth', 0.5 ),% unit circle
hold on
plot(eig(A),'k+', 'LineWidth', 1.0)% open loop Eigenvalues
plot(Z,'ro','LineWidth', 1.0)% LQR Eigenvalues A-B*K_lqr
plot(eig(Acl),'b*', 'LineWidth', 1.0)% MPC Eigenvalues
grid
xlabel('Real axis', 'FontSize', 14)
ylabel('Imaginry axis', 'FontSize', 14)
legend ('Open-loop Eigenvalues','LQR Eigenvalues','MPC using Laguerre
Functions Eigenvalues', 'FontSize',14)
title('Eigenvalues',strcat('Air speed U (m/s) = ',num2str(round (AF_U,2))),
'FontSize',14),

%----- Response to initial condition (Regulator case) -----%
up=0.0;
y=zeros(m1,1),
u=zeros(n_in,1),

% Initial conditions
xm = [0; % Initial plunge rate
0; % Initial pitch rate
0; % Initial control surface rate
0; % Initial plunge displacement
2*pi/180; % Initial pitch displacement [rad.]
0; % Initial control surface angle [rad.]
0; % Initial aerodynamic lag (1st state)
0]; % Initial aerodynamic lag (2nd state)

r1=zeros(1,N_sim+10), % zero input (regulator)

% Closed loop MPC using Laguerre functions simulation without constraints

% Select the covariance Q for the process noise w, and the covariance R for
% The measurement noise v for Kalman filter

Q_obs=0.001*eye(n1+m1, n1+m1), % Process noise covariance
R_obs=0.01*eye(m1,m1), % measurement noise covariance

[u1, y1,xm1, k]=simuucob(xm, up,u, y, r1, Ad, Bd,Cd, A, B, C, N_sim, Omega,
Psi, Lzerot, Q_obs, R_obs),

tau=k*Delta_t;
t=tau/w_na; % Transforming the discrete sample instant to time in sec to use
in plots

% MPC with Laguerre functions closed loop system response to Initial
Condition I.C. alpha = 2 deg

figure (2) % Plots of The Displacement states h, Alpha, Beta
subplot (311)

```

```

plot(t,xml(4,:)*b*1000,'LineWidth',1.2)
grid
xlabel('Time(sec)', 'FontSize', 14)
ylabel('Plunge h (mm)', 'FontSize', 14)
title('MPC without constraints closed loop system response to Initial
Condition I.C. \alpha = 2 deg',strcat('Air Speed U (m/s) = ',num2str(round
(AF_U,2))), 'FontSize',14)
subplot (312)
plot(t,xml(5,:)*180/pi,'LineWidth',1.2)
grid
xlabel('Time(sec)', 'FontSize', 14)
ylabel('Pitch Angle \alpha (deg)', 'FontSize', 14)
subplot (313)
plot(t,y1*180/pi,'LineWidth',1.2)
grid
xlabel('Time(sec)', 'FontSize', 14)
ylabel('Flap angle \beta (deg)', 'FontSize', 14)
legend('Simulated Response', 'Location', 'southeast', 'FontSize',14)

%Finding ISE Performance index for the displacement states
StepData_h=stepinfo(xml(4,:)*b*1000, t, 0),
T_h=StepData_h.SettlingTime;
ISE_h = ISEPerformance(xml(4,:),r1,t,T_h),% plunge state regulation ISE
performance index

StepData_alpha=stepinfo(xml(5,:)*180/pi, t, 0),
T_alpha=StepData_alpha.SettlingTime;
ISE_alpha = ISEPerformance(xml(5,:),r1,t,T_alpha),% pitch state regulation
ISE performance index

StepData_beta=stepinfo(y1*180/pi, t, 0),
T_beta=StepData_beta.SettlingTime;
ISE_beta = ISEPerformance(y1,r1,t,T_beta),% Flap angle state regulation ISE
performance index

%MPC with Laguerre functions closed loop system response to Initial
Condition I.C. alpha = 2 deg

figure (3) % Plots of The Rate states h_dot, Alpha_dot and Beta_dot
subplot(311)
plot(t,xml(1,:)*b*1000*w_na,'LineWidth',1.2)
grid
xlabel('Time (Sec)', 'FontSize', 14)
ylabel('Plunge rate h (mm/s)', 'FontSize', 14)
title('MPC witout constraints closed loop system response to Initial
Condition I.C. \alpha = 2 deg',strcat('Air Speed U (m/s) = ',num2str(round
(AF_U,2))), 'FontSize',14)
subplot(312)
plot(t,xml(2,:)*(180/pi)*w_na,'LineWidth',1.2)
grid
xlabel('Time (Sec)', 'FontSize', 14)
ylabel('Pitch Angle rate (deg/s)', 'FontSize', 14)
subplot(313)
plot(t,xml(3,:)*(180/pi)*w_na,'LineWidth',1.2)
grid
xlabel('Time (Sec)', 'FontSize', 14)
ylabel('Flap Angle rate (deg/s)', 'FontSize', 14)
legend('Simulated Response', 'Location', 'southeast', 'FontSize',14)

du=gradient(u1(:))./gradient(tau(:)), % Finding the rate of change of input
signal

u_max_reg=max(abs(u1))*180/pi;
du_max_reg=max(abs(du))*(180/pi)*w_na;

figure (4) % Plot of the control signal u and control signal rate of change
subplot (211)

```

```

plot(t,u1*180/pi, 'k','LineWidth',1.5)
xlabel('Time(sec)', 'FontSize', 14)
ylabel('u(t) (deg)', 'FontSize', 14)
title('MPC without constraints closed loop system response to Initial
Condition I.C. \alpha = 2 deg',strcat('Air Speed U (m/s) = ',num2str(round
(AF_U,2))), 'FontSize',14)
legend('Unconstrained Control Input u(t) (deg)', 'FontSize', 14)
grid
subplot(212)
plot(t,du*(180/pi)*w_na,'LineWidth',1.2)
xlabel('Time (Sec)', 'FontSize', 14)
ylabel('du/dt (deg/sec)', 'FontSize', 14)
legend('Unconstrained Rate of change of control Input (deg/sec)', 'Location'
,'southeast', 'FontSize',14)
grid

StepData_u_reg=stepinfo(u1*180/pi, t, 0),
T_u_reg=StepData_u_reg.SettlingTime;
ISU_Reg = ISEPerformance(u1,r1,t,T_u_reg),% regulation input ISU performance
index
%----- Step Response -----%
y=zeros(m1,1),
u=zeros(n_in,1),

r1=ones(1,N_sim+10)*5*pi/180; % Step input beta = 5 deg
xm=zeros(n1,1),% zero initial conditions

%Closed loop MPC with Laguerre functions simulation without constraints
[u_t,y_t,xm1,k]=simuucob(xm, up,u, y, r1, Ad, Bd,Cd, A, B, C, N_sim, Omega,
Psi, Lzerot, Q_obs, R_obs),

t=k.*(Delta_t/w_na), % Transforming the discrete sample instant to time in
sec
du_t=gradient(u_t(:))./gradient(tau(:)), % Finding the rate of change of
input signal

u_max_t=max(abs(u_t))*180/pi;
du_max_t=max(abs(du_t))*(180/pi)*w_na;

figure (5) % Plots of Step response output, input signal, and input rate
signal
subplot(311)
plot(t,y_t*180/pi,'r', t,r1 (1,1:N_sim)*180/pi, 'k','LineWidth',1.2)
grid
xlabel('Time (Sec)', 'FontSize', 14)
ylabel('y(t)= \beta(t) (deg)', 'FontSize', 14)
legend ('Output: Flab angle \beta (deg)', 'Step Input ', 'Location'
,'southeast', 'FontSize',14),
title('MPC without constraints system response to step input \beta = 5
deg',strcat('Air Speed U (m/s) = ',num2str(round (AF_U,2))), 'FontSize',14)
subplot(312)
plot(t,u_t*180/pi,'k','LineWidth',1.2)
grid
xlabel('Time (Sec)', 'FontSize', 14)
ylabel('u(t) (deg)', 'FontSize', 14)
legend('Unconstrained Control Input u(t) (deg)', 'Location' ,'southeast',
'FontSize',14)
subplot(313)
plot(t,du_t*(180/pi)*w_na,'LineWidth',1.2)
grid
xlabel('Time (Sec)', 'FontSize', 14)
ylabel('du/dt (deg/sec)', 'FontSize', 14)
legend('Unconstrained Rate of change of control Input (deg/sec)', 'Location'
,'southeast', 'FontSize',14)

StepData_t=stepinfo(y_t*180/pi, t, 5),

```

```
T_t=StepData_t.SettlingTime;
ISE_t = ISEPerformance(y_t,r1,t,T_t),% tracking ISE performance index

StepData_u_t=stepinfo(u_t*180/pi, t),
T_u_t=StepData_u_t.SettlingTime;
ISU_t = ISEPerformance(u_t,r1,t,T_u_t),% tracking input ISU performance
index

%===== The End of the Code =====%
```

```

%-----%
%----- Main_CLMPC.m -----%
%-----%

% This is the main code 2D wing with control Flap Active Flutter
% Suppression by discrete MPC using Laguerre functions with constraint/
% With Kalman filter controller design and simulation
close all
clear
clc
%----- Input Data -----%

rho = 1.225; % Ambient air density [kg/m^3]
U = linspace(0, 25, 50), % Airspeed range to analyses [m/s]

%----- Find Open loop flutter speed -----%

Airfoil;
load('Airfoil'),

[U_f, f_f, f_h, f_a, f_B, g_h, g_a, g_B, l] = Modal_Data(U, rho),

disp('Open loop flutter speed is (m/s):')
disp(U_f)
disp('Open loop flutter frequency is (Hz):')
disp(f_f)

%----- Set the Air speed and the state space Matrices -----%

% prompt = 'Enter the Air Speed (m/s): ';
% AF_U = input(prompt),

AF_U = U_f*1.1; % 10% higher than flutter speed

states = {'h_dot' 'Alpha_dot' 'Beta_dot' 'h' 'Alpha' 'Beta' 'l1' 'l2'};
inputs = {'Aileron'};
outputs = {'\beta'};

[Ac, Bc, Cc, Dc] = SS_Matrices (AF_U, rho),

%----- Discretize the system -----%

Delta_t = 0.1;
[Ad, Bd, Cd, Dd] = c2dm(Ac, Bc, Cc, Dc, Delta_t),

% Find the integrator augmented matrices
[m1, n1] = size(Cd), % m1 is number of outputs and n1 is number of states
[n1, n_in] = size(Bd), % n_in is number of inputs

A = eye(n1+m1, n1+m1),
A(1:n1, 1:n1) = Ad;
A(n1+1:n1+m1, 1:n1) = Cd*Ad;
B = zeros(n1+m1, n_in),
B(1:n1, :) = Bd;
B(n1+1:n1+m1, :) = Cd*Bd;
C = zeros(m1, n1+m1),
C(:, n1+1:n1+m1) = eye(m1, m1),

[n, n_in] = size(B),

%----- Set the MPC parameters -----%

Np = 500; % prediction horizon
a = 0.3;
N = 16;
Q = C'*C;
R = 50*eye(n_in, n_in), % tuning matrix for control effort

```

```

N_sim=500; %number of simulation points

% ----- Set the input Constraints -----%

% The control input constraints
u_min=-10*pi/180; %by radian
u_max=10*pi/180; %by radian

% The control input rate constraint
du_dt_min=-105*pi/180; % by radian/sec
du_dt_max= 105*pi/180; % by radian/sec

%----- Find Omega and Psi -----%

[Omega, Psi]=dmqc(A,B,a,N,Np,Q,R),

%Laguerre matrix
[Al,L0]=lagd(a,N),
Lzerot=L0';

K_mpc=L0'*(Omega\Psi),

% ----- Stability (Eigen value comparison) -----%

Acl=A-B*K_mpc;
[X,Y,Z]=dlqr(A,B,Q,R),

figure (1)
viscircles([0 0],1, 'Color','k', 'LineWidth', 0.5 ),% unit circle
hold on
plot(eig(A),'k+','LineWidth', 1.0)% open loop Eigenvalues
plot(Z,'ro','LineWidth', 1.0)% LQR Eigenvalues A-B*K_lqr
plot(eig(Acl),'b*','LineWidth', 1.0)% MPC Eigenvalues
grid
xlabel('Real axis', 'FontSize', 14)
ylabel('Imaginary axis', 'FontSize', 14)
legend ('Open-loop Eigenvalues','LQR Eigenvalues','MPC using Laguerre
Functions Eigenvalues', 'FontSize',14)
title('Eigenvalues',strcat('Air speed U (m/s) = ',num2str(round (AF_U,2))),
'FontSize',14),

%----- Response to initial condition (Regulator case) -----%
up=0.0;
y=zeros(m1,1),
u=zeros(n_in,1),

% Initial conditions
xm = [0; % Initial plunge rate
0; % Initial pitch rate
0; % Initial control surface rate
0; % Initial plunge displacement
2*pi/180; % Initial pitch displacement [rad.]
0; % Initial control surface angle [rad.]
0; % Initial aerodynamic lag (1st state)
0]; % Initial aerodynamic lag (2nd state)

r1=zeros(1,N_sim+10), % zero input (regulator)

% Closed loop MPC using Laguerre functions simulation with constraints on u
and delta u.

deltau_min=(du_dt_min/w_na)*Delta_t;
deltau_max=(du_dt_max/w_na)*Delta_t;

% Select the covariance Q for the process noise w, and the covariance R for
% the Measurement noise v for Kalman filter

```

```

Q_obs=0.001*eye(n1+m1, n1+m1), % Process noise covariance
R_obs=0.01*eye(m1,m1), % measured noise covariance

[u1, y1,xm1, k]=simucob(xm, up,u, y, r1, Ad, Bd,Cd, A, B, C, N_sim, Omega,
Psi, Lzerot, deltau_min, deltau_max, u_min, u_max, Q_obs, R_obs),

tau=k*Delta_t;
t=tau/w_na; % Transforming the discrete sample instant to time in sec to use
in plots

% MPC with Laguerre functions closed loop system response to Initial
Condition I.C. alpha = 2 deg

figure (2) % Plots of The Displacement states h, Alpha, Beta
subplot (311)
plot(t,xm1(4,:)*b*1000,'LineWidth',1.2)
grid
xlabel('Time(sec)', 'FontSize', 14)
ylabel('Plunge h (mm)', 'FontSize', 14)
title('MPC with Input constraints closed loop system response to Initial
Condition I.C. \alpha = 2 deg',strcat('Air Speed U (m/s) = ',num2str(round
(AF_U,2))), 'FontSize',14)
subplot (312)
plot(t,xm1(5,:)*180/pi,'LineWidth',1.2)
grid
xlabel('Time(sec)', 'FontSize', 14)
ylabel('Pitch Angle \alpha (deg)', 'FontSize', 14)
subplot (313)
plot(t,y1*180/pi,'LineWidth',1.2)
grid
xlabel('Time(sec)', 'FontSize', 14)
ylabel('Flap angle \beta (deg)', 'FontSize', 14)
legend('Simulated Response', 'Location' , 'southeast', 'FontSize',14)

%Finding ISE Performance index for the displacement states
StepData_h=stepinfo(xm1(4,:)*b*1000, t, 0),
T_h=StepData_h.SettlingTime
ISE_h = ISEPerformance(xm1(4,:),r1,t,T_h)% plunge state regulation ISE
performance index

StepData_alpha=stepinfo(xm1(5,:)*180/pi, t, 0),
T_alpha=StepData_alpha.SettlingTime
ISE_alpha = ISEPerformance(xm1(5,:),r1,t,T_alpha)% pitch state regulation
ISE performance index

StepData_beta=stepinfo(y1*180/pi, t, 0),
T_beta=StepData_beta.SettlingTime
ISE_beta = ISEPerformance(y1,r1,t,T_beta)% Flap angle state regulation ISE
performance index

%MPC with Laguerre functions closed loop system response to Initial
Condition I.C. alpha = 2 deg

figure (3) % Plots of The Rate states h_dot, Alpha_dot and Beta_dot
subplot(311)
plot(t,xm1(1,:)*b*1000*w_na,'LineWidth',1.2)
grid
xlabel('Time (Sec)', 'FontSize', 14)
ylabel('Plunge rate h (mm/s)', 'FontSize', 14)
title('MPC with Input constraints closed loop system response to Initial
Condition I.C. \alpha = 2 deg',strcat('Air Speed U (m/s) = ',num2str(round
(AF_U,2))), 'FontSize',14)
subplot(312)
plot(t,xm1(2,:)*(180/pi)*w_na,'LineWidth',1.2)
grid
xlabel('Time (Sec)', 'FontSize', 14)
ylabel('Pitch Angle rate (deg/s)', 'FontSize', 14)

```

```

subplot(313)
plot(t,xm1(3,:)*(180/pi)*w_na,'LineWidth',1.2)
grid
xlabel('Time (Sec)', 'FontSize', 14)
ylabel('Flap Angle rate (deg/s)', 'FontSize', 14)
legend('Simulated Response', 'Location', 'southeast', 'FontSize',14)

du=gradient(u1(:))./gradient(tau(:)), % Finding the rate of change of input
signal

u_max_reg=max(abs(u1))*180/pi;
du_max_reg=max(abs(du))*(180/pi)*w_na;

figure (4) % Plot of the control signal u and control signal rate of change
subplot (211)
plot(t,u1*180/pi, 'k','LineWidth',1.5)
xlabel('Time(sec)', 'FontSize', 14)
ylabel('u(t) (deg)', 'FontSize', 14)
title('MPC with Input constraints closed loop system response to Initial
Condition I.C. \alpha = 2 deg',strcat('Air Speed U (m/s) = ',num2str(round
(AF_U,2))), 'FontSize',14)
legend('Constrained Control Input u(t) (deg)', 'FontSize', 14)
grid
subplot(212)
plot(t,du*(180/pi)*w_na,'LineWidth',1.2)
xlabel('Time (Sec)', 'FontSize', 14)
ylabel('du/dt (deg/sec)', 'FontSize', 14)
legend('Constrained Rate of change of control Input (deg/sec)', 'Location'
,'southeast', 'FontSize',14)
grid

StepData_u_reg=stepinfo(u1*180/pi, t, 0),
T_u_reg=StepData_u_reg.SettlingTime
ISU_Reg = ISEPerformance(u1,r1,t,T_u_reg)% regulation input ISU performance
index
%-----Step Response-----%
y=zeros(m1,1),
u=zeros(n_in,1),

r1=ones(1,N_sim+10)*5*pi/180; % Step input beta = 5 deg

xm=zeros(n1,1),% zero initial conditions

%Closed loop MPC with Laguerre functions simulation with constraints on u
and delta u.
[u_t,y_t,xm1,k]=simucob(xm, up,u, y, r1, Ad, Bd,Cd, A, B, C, N_sim, Omega,
Psi, Lzerot, deltau_min, deltau_max, u_min, u_max, Q_obs, R_obs),

t=k.*(Delta_t/w_na), % Transforming the discrete sample instant to time in
sec

du_t=gradient(u_t(:))./gradient(tau(:)), % Finding the rate of change of
input signal

u_max_t=max(abs(u_t))*180/pi;
du_max_t=max(abs(du_t))*(180/pi)*w_na;

figure (5) % Plots of Step response output, input signal, and input rate
signal
subplot(311)
plot(t,y_t*180/pi,'r', t,r1 (1,1:N_sim)*180/pi, 'k','LineWidth',1.2)
grid
xlabel('Time (Sec)', 'FontSize', 14)
ylabel('y(t)= \beta(t) (deg)', 'FontSize', 14)
legend ('Output: Flap angle \beta (deg)', 'Step Input ', 'Location'
,'southeast', 'FontSize',14),

```



```

title('MPC with input constraints system response to step input \beta = 5
deg',strcat('Air Speed U (m/s) = ',num2str(round (AF_U,2))), 'FontSize',14)
subplot(312)
plot(t,u_t*180/pi,'k','LineWidth',1.2)
grid
xlabel('Time (Sec)', 'FontSize', 14)
ylabel('u(t) (deg)', 'FontSize', 14)
legend('Constrained Control Input u(t) (deg)', 'Location' , 'southeast',
'FontSize',14)
subplot(313)
plot(t,du_t*(180/pi)*w_na,'LineWidth',1.2)
grid
xlabel('Time (Sec)', 'FontSize', 14)
ylabel('du/dt (deg/sec)', 'FontSize', 14)
legend('Constrained Rate of change of control Input (deg/sec)', 'Location'
,'southeast', 'FontSize',14)

StepData_t=stepinfo(y_t*180/pi, t, 5)
T_t=StepData_t.SettlingTime;
ISE_t = ISEPerformance(y_t,r1,t,T_t)% tracking ISE performance index

StepData_u_t=stepinfo(u_t*180/pi, t),
T_u_t=StepData_u_t.SettlingTime;
ISU_t = ISEPerformance(u_t,r1,t,T_u_t)% tracking input ISU performance index

%===== The End of the Code =====%

```

```

%-----%
%----- Airfoil.m -----%
%-----%

function Airfoil
span = 0.520;           % Wingspan [m]
a = -0.5;              % Distance between mid-chord and ea in
semi-chords
b = 0.127;            % Reference semi-chord [m]
c = 0.5;              % Flap position in semichords
m_B = 0.18597;        % mass of the aileron [kg]
m = (0.62868+ m_B)/span; % Mass/length of wing-aileron [kg/m]
x_a = 0.434;          % Distance between airfoil ea and cg in
% semi-chords
x_B = 0.01996;        % Distance between control surface hinge
axis and
% cg in semi-chords
I_a = 13.47e-3;        % Mass inertia of wing about ea [kgm^2]
I_B = 326.4e-6;        % Mass inertia of flap about hinge axis [kgm^2]

%-----

k_h=2818.8;           %the plunge structural stiffness (per
span)
k_a=37.34;           %the pitch structural stiffness (per span)
k_B=3.9;             %the control surface structural stiffness
(per span)

w_nh = sqrt(k_h/m) ; % Uncoupled plunge frequency [rad/s]
w_na = sqrt(k_a/I_a), % Uncoupled pitch frequency [rad/s]
w_nB = sqrt(k_B/I_B), % Uncoupled control surface frequency
[rad/s]

% c_h = 0.025;        % Plunge damping coefficient [Ns/m]
% c_a = 0.05;         % Pitch damping coefficient [Ns/m]
% c_B = 0.05;         % Control surface damping coefficient [Ns/m]

%-----

S_a = 0.08587;        %the static mass moment of wing-aileron
about wing elastic axis [kg m]
S_B = 0.00395;        %the static mass moment of aileron about
about aileron hinge [kg m]

r_a = 0.7321;         %the dimensionless radius of gyration of
the wing about the elastic axis
r_B = 0.11397;        %the dimensionless radius of gyration of
the control surface about the hinge point
k = 0.03984;          %kapa ,

Z_h=0.0113;          %the plunge damping ratio
Z_a=0.01626;          %the pitch damping ratio
Z_B=0.0115;          %the control surface damping ratio

save ('Airfoil')

%===== The End of the Code =====%

```

```

%-----%
%----- Modal_Data.m -----%
%-----%

function [U_f, f_f, f_h, f_a, f_B, g_h, g_a, g_B] = Modal_Data(U, rho)
%----- Calculating open loop eigenvalues -----%

load('Airfoil'),
lambda = zeros(8,length(U)), % To store each speed-related eigenvalues in
a column
sorted_lambda=lambda;
sorted_real=lambda;

% A loop to calculate the eigenvalues at each step of the speed analysis
range
for i=1:length(U)
    A= SS_Matrices (U(i),rho),
    lambda(:,i) = eig(A),
end
%----- Sorting of the open loop eigenvalues -----%

[sorted_imag,idx]=sort(imag(lambda)), % ascending order sorting of the
eigenvalues based on imaginary part
for j=1:8
    for i=1:length(U)
sorted_lambda(j,i)=lambda(idx(j,i),i), % a new matrix contains the sorted
eigenvalues
sorted_real(j,i)=real(sorted_lambda(j,i)), % the real part of the sorted
eigenvalues (dimensionless damping frequency)
end
end

w_h=sorted_imag(6,:), % the imaginary part of eigenvalue (oscillation
dimensionless angular frequency) of h
w_a=sorted_imag(7,:), % the imaginary part of eigenvalue (oscillation
dimensionless angular frequency) of alpha
w_B=sorted_imag(8,:), % the imaginary part of eigenvalue (oscillation
dimensionless angular frequency) of beta
f_h=(w_h*w_na)/(2*pi), % multiplied by w_na to get frequency by Hz
f_a=(w_a*w_na)/(2*pi),
f_B=(w_B*w_na)/(2*pi),
g_h=(sorted_real(6,:)*w_na)/(2*pi), % the real part of eigenvalue (damping
frequency) related to h
g_a=(sorted_real(7,:)*w_na)/(2*pi), % the real part of eigenvalue (damping
frequency) related to alpha
g_B=(sorted_real(8,:)*w_na)/(2*pi), % the real part of eigenvalue (damping
frequency) related to beta

%----- finding open loop flutter speed and frequency -----%

% Interpolation loop to find the flutter frequencies and speed
for r=1:length(U)-1
    if g_h(r) <=0 andand g_h(r+1) >0 % when the eigenvalues real part
(related to h) moves from LH to RH plane
        g_i= [g_h(r) g_h(r+1)];
        f_i= [f_h(r) f_h(r+1)];
        U_i=[U(r) U(r+1)];
        U_f=interp1(g_i,U_i,0), % interpolate to find the flutter speed
        f_f=interp1(U_i,f_i,U_f), % interpolate to find the flutter
oscillation frequency
    end
end
%===== The End of the Code =====%

```

```

%-----%
%----- SS_Matrices.m -----%
%-----%

function [A,B,C,D] = SS_Matrices(U,rho)
% Calculates the State Space Matrices A B C and D for a specific Air speed U
and density

load ('Airfoil'),

%----- Theodorsen's Coefficients -----%
p = -(1/3)*(sqrt(1 - c^2)^3),
T_1 = -(1/3)*sqrt(1 - c^2)*(2 + c^2)+c*acos(c),
T_2 = c*(1 - c^2) - sqrt(1 - c^2)*(1 + c^2)*acos(c) + c*(acos(c))^2;
T_3 = -((1/8) + c^2)*(acos(c))^2 + (1/4)*c*sqrt(1 - c^2)*acos(c)*(7 + 2*c^2)
- (1/8)*(1 - c^2)*(5*c^2 + 4),
T_4 = -acos(c) + c*sqrt(1 - c^2),
T_5 = -(1 - c^2) - (acos(c))^2 + 2*c*sqrt(1 - c^2)*acos(c),
T_6 = T_2;
T_7 = -((1/8) + c^2)*acos(c) + (1/8)*c*sqrt(1 - c^2)*(7 + 2*c^2),
T_8 = -(1/3)*sqrt(1 - c^2)*(2*c^2 + 1) + c*acos(c),
T_9 = (1/2)*(-p + a*T_4),
T_10 = sqrt(1 - c^2) + acos(c),
T_11 = acos(c)*(1 - 2*c) + sqrt(1 - c^2)*(2 - c),
T_12 = sqrt(1 - c^2)*(2 + c) - acos(c)*(1+2*c),
T_13 = (1/2)*(-T_7 - (c - a)*T_1),
T_14 = (1/16) + (1/2)*a*c;
T = [T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8 T_9 T_10 T_11 T_12 T_13 T_14];

%----- W.P. Jones' Approximation -----%
delta_1 = 0.165;
delta_2 = 0.335;
lambda_1 = 0.041;
lambda_2 = 0.320;

%----- Dimensionless parameters -----%
meu=m/(pi*rho*b^2), %the mass ratio of the wing mass to the mass of
the air affected by the wing
segma=w_nh/w_na; %the ratio of uncoupled plunge and pitch frequencies
V=U./(b*w_na), %the dimensionless freestream speed of the air (reduced
velocity)

%----- Structural Matrices -----%
M_s=meu*[2.1658 x_a x_B;
x_a (r_a)^2 (c-a)*x_B+(r_B)^2,
x_B ((c-a)*x_B+(r_B)^2) (r_B)^2];

D_s=2*meu*[segma*Z_h 0 0;
0 (r_a)^2*Z_a 0;
0 0 (w_nB/w_na)*(r_B)^2*Z_B];

K_s=meu*[segma^2 0 0
0 (r_a)^2 0
0 0 (w_nB/w_na)^2*(r_B)^2];

%----- Aerodynamic Matrices -----%
i_end = length(V),

for i = 1: i_end

M_a=[-1 a T(1)/pi;
a -((1/8)+a^2) -2*T(13)/pi;
T(1)/pi -2*T(13)/pi T(3)/pi^2];

D_a=V(i)*[-2 -2*(1-a) (T(4)-
T(11))/pi];

```

```

    1+2*a          a*(1-2*a)
(1/pi)*(T(8)-T(1)+(c-a)*T(4)+a*T(11)),
    -T(12)/pi      (1/pi)*(2*T(9)+T(1)+(T(12)-T(4))*(a-1/2))
(T(11)/(2*pi^2))*(T(4)-T(12))];

K_a=V(i)^2*[0      -2      -2*T(10)/pi;
             0      1+2*a    (1/pi)*(2*a*T(10)-T(4)),
             0      -T(12)/pi  (-1/pi^2)*(T(5)-T(10)*(T(4)-T(12)))]];

L_delta=2*V(i)*[delta_1      delta_2;
                -((1/2)+a)*delta_1      -((1/2)+a)*delta_2;
                (T(12)*delta_1)/(2*pi)      (T(12)*delta_2)/(2*pi)];

%----- Aerodynamic Lag Matrices -----%
L_lambda = V(i)*[-lambda_1      0;
                 0      -lambda_2];

Q_a = [1      (1/2)-a      T(11)/(2*pi)];
Q_v = V(i)*[0      1      T(10)/pi];

%----- System Matrix -----%
A_11 = -inv(M_s - M_a)*(D_s - D_a),
A_12 = -inv(M_s - M_a)*(K_s - K_a),
A_13 = inv(M_s - M_a)*L_delta;
A_21 = eye(3, 3),
A_22 = zeros(3, 3),
A_23 = zeros(3, 2),
A_31 = [Q_a*A_11 + Q_v; Q_a*A_11 + Q_v];
A_32 = [Q_a*A_12; Q_a*A_12];
A_33 = [Q_a*A_13; Q_a*A_13] + L_lambda;
A = [A_11 A_12 A_13; A_21 A_22 A_23; A_31 A_32 A_33];

%----- Input Matrix -----%
B_11= inv(M_s - M_a)*[0;0;meu*(w_nB/w_na)^2*(r_B)^2];
B_21= [0;0;0];
B_31= [Q_a*B_11; Q_a*B_11];

B=[B_11;B_21;B_31];

%----- Output Matrix -----%

C = [0 0 0 0 0 1 0 0];
D = [0];

end

%===== The End of the Code =====%

```

```

%-----%
%----- Flutter_LQI.m -----%
%-----%

function [Ki, Kx, L]= Flutter_LQI (sys,A, C,Q,R)

%find the Tracking controller gain K using lqi function

K = lqi(sys, Q, R),

% extract integrator gain for Beata tracking
Ki=K(1,9),

% extract LQR Gain

Kx= zeros(1,8),
for m=1:8

Kx (1,m)= K (1,m),

end

%----- Kalman Filter -----%

% the process noise matrix G in  $x = Ax + Bu + Gw$  {State equation} is
% selected as

G_Kalman = eye(8),

% select the covariance Q for the process noise w, and the covariance R for
the
% measurement noise v

Q_Kalman = [0.001 0 0 0 0 0 0 0;
0 0.001 0 0 0 0 0 0;
0 0 0.001 0 0 0 0 0;
0 0 0 0.001 0 0 0 0;
0 0 0 0 0.001 0 0 0;
0 0 0 0 0 0.001 0 0;
0 0 0 0 0 0 0.001 0;
0 0 0 0 0 0 0 0.001];

R_Kalman = (C*C')*0.01;

%the observer gain matrix L

L = lqe(A, G_Kalman, C, Q_Kalman, R_Kalman),

%===== The End of the Code =====%

```

```

%-----%
%----- mpcgain.m -----%
%-----%

function [Phi_Phi, Phi_F, Phi_R, A_e, B_e, C_e]= mpcgain (Ad, Bd, Cd, Nc,
Np)

%A function to calculate phiT_phi, phiT_F, PhiT_R
%Create the augmented model for MPC
%Determine the dimensions of the system matrices
[m1,n1]=size(Cd), %m1 is number of outputs and n1 is number of states
[n1,n_in]=size(Bd), % n_in is number of inputs

%Produce the augmented state variable model for the design of predictive
control
A_e=eye(n1+m1, n1+m1),
A_e(1:n1,1:n1)=Ad;
A_e(n1+1:n1+m1,1:n1)=Cd*Ad;
B_e=zeros(n1+m1,n_in),
B_e(1:n1,:)=Bd;
B_e(n1+1:n1+m1,:)=Cd*Bd;
C_e=zeros(m1,n1+m1),
C_e(:,n1+1:n1+m1)=eye(m1,m1),

h(1,:)=C_e;
F(1,:)=C_e*A_e;
for kk=2:Np
    h(kk,:)=h(kk-1,:)*A_e;
    F(kk,:)=F(kk-1,:)*A_e;
end
v=h*B_e;
Phi=zeros(Np, Nc), %declare the dimension of Phi
Phi(:,1)=v; %first column of Phi
for i=2:Nc
    Phi(:,i)=[zeros(i-1,1),v(1:Np-i+1,1)]; %Toeplitz matrix
end
BarRs=ones(Np,1),
Phi_Phi=Phi'*Phi;
Phi_F=Phi'*F;
Phi_R=Phi'*BarRs;

end

%===== The End of the Code =====%

```

```
%-----%
%----- lagd.m -----%
%-----%

%Generates the initial condition of the Laguerre function L(0) and the state
space system matrix A1

function [A,L0]=lagd(a,N)
v(1,1)=a;
L0(1,1)=1;
for k=2:N
    v(k,1)=(-a)^(k-2)*(1-a*a),
    L0(k,1)=(-a)^(k-1),
end
L0=sqrt((1-a*a))*L0;
A(:,1)=v;
for i=2:N
    A(:,i)=[zeros(i-1,1),v(1:N-i+1,1)];
end
end

%===== The End of the Code =====%
```



```

%-----%
%----- simuucob.m -----%
%-----%

% MIMO closed loop MPC simulation without constraints. With observer
% sp is the setpoint signal
function [u1, y1,xm1, k]=simuucob(xm, up,u, y, sp, Ap, Bp,Cp, A, B, C,
N_sim, Omega, Psi, Lzerot, Q_obs, R_obs)

[m1,n1]=size(Cp),
[n1, n_in]=size (Bp),
xm_obs=zeros(n1,1),

K_obs=dlqr( A', C', Q_obs, R_obs)';

[ny,n]=size(C),
[n,nu]=size(B),

X_hat=zeros(n,1),

for kk=1:N_sim
    Xsp=[zeros(n-ny,1), sp(:,kk)];
    eta=-(Omega\Psi)*(X_hat-Xsp),
    deltau=Lzerot*eta;

    u=u+deltau; %update u

    deltau1(:,kk)=deltau;
    u1(1:nu, kk)=u;
    y1(1:ny, kk)=y;
    xm1(1:n1, kk)=xm;

    %u and y to generate X_hat(k+1)

    xm=Ap*xm+Bp*u; %find xm(k+1)

    X_hat=A*X_hat+K_obs*(y-C*X_hat)+B*deltau;

    y=Cp*xm; %find y(k=1)
    up=u;

end

k=0:(N_sim-1),

end
%===== The End of the Code =====%

```

```

%-----%
%----- simucob.m -----%
%-----%

%MIMO closed loop MPC simulation with constraints on u and delta u. with
observer
%sp is the setpoint signal
function [u1, y1,xm1, k]=simucob(xm, up,u, y, sp, Ap, Bp,Cp, A, B, C, N_sim,
Omega, Psi, Lzerot, deltau_min, deltau_max, u_min, u_max, Q_obs, R_obs)

[m1,n1]=size(Cp),
[n1, n_in]=size (Bp),
xm_obs=zeros(n1,1),

K_obs=dlqr( A', C', Q_obs, R_obs)';

[ny,n]=size(C),
[n,nu]=size(B),

X_hat=zeros(n,1),

for kk=1:N_sim
    Xsp=[zeros(n-ny,1), sp(:,kk)];
    eta=-(Omega\Psi)*(X_hat-Xsp),
    deltau=Lzerot*eta;
    if (deltau>deltau_max)
        deltau=deltau_max;
    end
    if (deltau<deltau_min)
        deltau=deltau_min;
    end
    u=u+deltau; %update u

    if (u>u_max)
        deltau=u_max-up;
        u=u_max;
    end
    if (u<u_min)
        deltau=u_min-up;
        u=u_min;
    end

    deltau1(:,kk)=deltau;
    u1(1:nu, kk)=u;
    y1(1:ny, kk)=y;
    xm1(1:n1, kk)=xm;

    %u and y to generate X_hat(k+1)

    xm=Ap*xm+Bp*u; %find xm(k+1)

    X_hat=A*X_hat+K_obs*(y-C*X_hat)+B*deltau;

    y=Cp*xm; %find y(k=1)
    up=u;

end

k=0:(N_sim-1),

end

%===== The End of the Code =====%

```

```
%-----%
%----- ISEPerformance.m -----%
%-----%

function [ISE] = ISEPerformance(y,r,t)
%A function to calculate The Integral of Square of the Error ISE Performance
index
%y is the output vector
%r is the input vector
%t is the time vector

error=zeros(length (t),1),
error_sq=error;

for ii=1:length(t)
    error(ii)=y(ii)-r(ii),
    error_sq(ii)=error(ii).^2;
end

ISE=trapz(t,error_sq),

end

%===== The End of the Code =====%
```