ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

MODEL PREDICTIVE CONTROL FOR UNMANNED AERIAL VEHICLE

M.Sc. THESIS

Halit Fırat ERDOĞAN

Department of Mechanical Engineering

System Dynamics and Control Programme

SEPTEMBER 2014

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

MODEL PREDICTIVE CONTROL FOR UNMANNED AERIAL VEHICLE

M.Sc. THESIS

Halit Fırat ERDOĞAN
(503111607)

Department of Mechanical Engineering

System Dynamics and Control Programme

Thesis Advisor:Assist. Prof. Dr. Ayhan KURAL

SEPTEMBER 2014

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

İNSANSIZ HAVA ARACININ MODEL TABANLI ÖNGÖRÜLÜ KONTROLÜ

YÜKSEK LİSANS TEZİ

Halit Fırat ERDOĞAN
(503111607)

Makina Mühendisliği Anabilim Dalı

Sistem Dinamiği ve Kontrol Programı

Tez Danışmanı: Y. Doç. Ayhan KURAL

EYLÜL 2014

Halit Fırat ERDOĞAN, a **M.Sc.** student of ITU **Graduate School of Science Engineering and Technology** student ID 503111607, successfully defended the **thesis** entitled "**MODEL PREDICTIVE CONTROL FOR UNMANNED AERIAL VEHICLE**", which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Assist. Prof. Dr. Ayhan KURAL** İstanbul Technical University	.....
Jury Members :	**Assos. Prof Dr. Kenan Refah KUTLU** İstanbul Technical University	.....
	Assist. Prof. Dr. Ertan ÖZNERGİZ Yıldız Technical University	.....

Date of Submission : 02 September 2014
Date of Defense : 16 Semtember 2014

Thanks to Prof. Dr. Can ÖZSOY for his great support,

FOREWORD

Unmanned aerial vehicles are autonomous systems. The stability must be maintain in hard conditions. Model predictive control is a smart and human-like solution for the control of UAV. MPC is applied to nonlinear model of Aerosonde which is a small unmanned aircraft. The lateral and longitudinal dynamics are not seperated in this thesis. Single controller is used to control all UAV. Aerosim which is a very spesific blockset for UAV model is used for simulations. And FlightGear visiual interface is used to visiualize the simulations.

SEPTEMBER 2014 Halit Fırat ERDOĞAN

TABLE OF CONTENTS

Page

FOREWORD ..... **iix**
TABLE OF CONTENTS ..... **xi**
ABBREVIATIONS ..... **xiii**
LIST OF TABLES ..... **xv**
LIST OF FIGURES ..... **xvii**
SUMMARY ..... **xix**
ÖZET ..... **xxi**
1. INTRODUCTION **1**
1.1 Some Applications of UAV 1
1.2 The History of UAV 2
1.3 The Need of Autonomous 3
1.4 The Future of UAV 3
1.5 Command and Control Elements 4
1.6 Model Predictive Control 5
2. MODELLING UAV **7**
2.1 Coordinate Frames 7
2.1.1 Rotation of reference frame 7
2.1.2 Rotation of a vector 9
2.2 UAV Coordinate Frames 10
2.2.1 The inertial frame Fi 10
2.2.2 The inertial frame Fv 11
2.2.3 Euler Angles 11
2.2.3.1 ψ Heading (yaw) 11
2.2.3.2 θ Elevation (pitch) 12
2.2.3.3 $\emptyset$ Bank (roll) 12
2.2.4 Inertial frame to body frame transformation 13
2.3 Airspeed, Wind Speed and Ground Speed 13
3. EQUATIONS OF MOTION OF AIRCRAFT **15**
3.1 States 15
3.2 Kinematics 16
3.3 Equations of Motion 16
3.3.1 Translational motion 16
3.3.2 Rotational motion 16
3.4 Forces and Moments 20
3.4.1 Aerodynamic forces and moments 20
3.4.2 Control surfaces 21
3.4.2.1 Longitudinal aerodynamics 22
3.4.2.2 Lateral aerodynamics 23
3.4.3 Propulsion forces and moments 23
3.4.3.1 Propoller thrust 23

3.5 Linear Model of UAV 24
3.5.1 Trim conditions 25
3.5.2 Linear state space models 27
3.6 Aerosim 28
3.6.1 Library description 29
4. MODEL PREDICTIVE CONTROL 31
4.1 State Space Model for MPC 32
4.2 Performance Index (Cost Function) 34
4.2.1 Constraints 34
4.2.2 Weighting matrices 35
5. SIMULATIONS and RESULTS 37
5.1 Model Selection 37
5.2 Trim 38
5.3 Closed Loop Model 41
5.4 Tuning 42
5.5 Results 46
5.6 FlightGear Results 54
6. CONCLUSIONS AND RECOMMENDATIONS 57
6.1 Recommendations 58

ABBREVIATIONS

MPC	**:** Model Predictive Control
UAV	**:** Unmanned Aerial Vehicle
p	**:** Roll rate of the UAV
q	**:** Pitch rate of the UAV
r	**:** Yaw rate of the UAV
u	**:** Intertial velocity of the airframe on x-axis
v	**:** Intertial velocity of the airframe on y-axis
w	**:** Intertial velocity of the airframe on z-axis
$\boldsymbol{\phi}$	**:** Roll Angle
$\boldsymbol{\theta}$	**:** Pitch Angle
$\boldsymbol{\psi}$	**:** Yaw Angle
$\boldsymbol{C_L}$	**:** Aerodynamic lift coefficient
$\boldsymbol{m}$	**:** Aerodynamic Pitching moment Coefficient

LIST OF TABLES

Page

Table 3.1 : State Variable of UAV 15
Table 3.2 : Lateral State Space Model Coefficients 28
Table 3.3 : Longitudinal State Space Model Coefficients 28

LIST OF FIGURES

Page

Figure 2.1 : Rotation in 2D 7
Figure 2.2 : Rotation of p about the $k0$-axis. 9
Figure 2.3 : The inertial coordinate frame. 11
Figure 2.4 : Yaw angle 11
Figure 2.5 : Elevation (Pitch) 12
Figure 2.6 : Bank Angle 12
Figure 3.1 : Definition of axes motion. 15
Figure 3.2 : Pressure distribution around wing 20
Figure 3.3 : Control surfaces deflection 21
Figure 3.4 : Linear aerodynamic model 22
Figure 3.5 : Simulink library browser 30
Figure 4.1 : Basic structure of MPC 32
Figure 5.1 : Open loop model of Aerosonde 38
Figure 5.2 : Main structure of the model without mask 40
Figure 5.3 : Trim model used in trim program 41
Figure 5.4 : MPC structure overview 42
Figure 5.5 : Closed loop model of aircraft with MPC and FlightGear interface 43
Figure 5.6 : Prediction and control horizons of first simulation 44
Figure 5.8: Weights on inputs, input rates and outputs for first simulation 45
Figure 5.9: Weights on inputs, input rates and outputs for second simulation 45
Figure 5.10: The constraints on inputs, input rates and outputs for first simulation 46
Figure 5.11: The constraints on inputs, rates and outputs for second simulation 46
Figure 5.12 : Aileron Deflection of first simulation 47
Figure 5.13 : Elevator deflection of first simulation 47
Figure 5.14 : Rudder deflection of first simulation 47
Figure 5.15 : Throttle deflection of first simulation 48
Figure 5.16 : Aileron Deflection of second simulation 48
Figure 5.17 : Elevator deflection of second simulation 48
Figure 5.18 : Rudder deflection of second simulation 49
Figure 5.19 : Throttle deflection of second simulation 49
Figure 5.20 : Airspeed result for first simulation 49
Figure 5.21 : Sideslip result for first simulation 50
Figure 5.22 : Angle of attack result for first simulation 50
Figure 5.23 : Bank angle result for first simulation 50
Figure 5.24 : Pitch angle result for first simulation 51
Figure 5.25 : Yaw angle result for first simulation 51
Figure 5.26 : Altitude result for first simulation 51
Figure 5.27 : Airspeed 52
Figure 5.28 : Sideslip 52
Figure 5.29 : Angle of attack 52
Figure 5.30 : Bank angle 53

Figure 5.31 : Pitch angle 53
Figure 5.32 : Yaw angle 53
Figure 5.33 : Altitude 54
Figure 5.34 : Aircraft flight from back. 54
Figure 5.35 : Cockpit and aircraft together 55

MODEL PREDICTIVE CONTROL OF UAV

SUMMARY

Discrete time model predictive controller will be designed to control a UAV in this thesis. The UAV 'Aerosonde' was modeled by u-dynamics and named Aerosim with the MATLAB Simulink software is choosen as model. The reasons behind choosing Aerosim are having six degree of freedom aircraft model, very detailed modeling which even includes worlds magnetic model and ready to use Aerosonde model.
UAVs are designed to fly at low altitude like 1000 meters to observe ground objects, enemies etc. This low altitude flight situation brings highly crash risk to UAVs. A robust and accurate autopilot system is neccesary for small UAV's to successfully perform tasks like low-altitude surveillance. Thats why MPC is chosen. The controller calculates the system controls, needed to track the selected outputs of the Aerosonde aircraft model. Model predictive controller makes it possible to add hard constraints to the states, controls and outputs of the linear state space model. MPC has been applied to UAV control before, but in this thesis, seven output and four input controlled by MPC. This brings the ability to MPC controller to account the interactions between all output and input variables. On the other hand for all inputs and outputs just one controller algorithm will be calculated. This also brings less computing effort to the computer.
Because of the high nonlinearities of the UAV dynamics, lots of control techniques, like PID control, neural network (NN), fuzzy logic (FL), sliding mode control, and H_∞ control, have been used in UAVs autopilot systems to acquire a smooth desirable trajectory navigation. Nowadays, technological developments in wireless networks and micro electromechanical systems make it possible to use inexpensive micro controllers on small UAVs. The small UAVs can give developers a different view of the environment and a much easier way for remote sensing, especially in cases like environment characterization and natural habitats monitoring.

İNSANSIZ HAVA ARACININ MODEL TABANLI ÖNGÖRÜLÜ KONTROLÜ

ÖZET

Bu tezde bir insansız hava aracının model tabanlı öngörülü kontrolü yapılmıştır.
Aerosonde isimli küçük insansız hava aracı model olarak seçildi. Bu uçağın matematik modeli için, u-dynamics firması tarafından Matlab-Simulink için geliştirilen Aerosim isimli blok setten yararlanılmıştır. Bu blok setin tercih edilmesinin sebepleri şu şekildedir: Aerosim herhangi bir hava aracının simulasyonu için gereken her unsuru ve hava aracının 6 serbestlik dereceli doğrusal olmayan modelini barındırır. Diğer sıradan hava aracı modellerine ek olarak parametrelerin değiştirilerek dilenilen yapıda bir insansız hava aracı modelinin elde edilmesini sağlar. Bünyesinde Aerosonde insansız hava aracının hazır modelini barındırır. Ayrıca bu blok set simulasyon verisini FlightGear gibi görsel simulasyon programlarına aktarabilmesi sayesinde eş zamanlı olarak matlab ortamında simulasyon yaparken bir yandan da FlightGear'da uçağın nasıl bir uçuş yaptığı gözlemlenebilmektedir. Bu blok setin kütüphanesinden söz etmek gerekirse, bünyesinde yüzden fazla uçağın modelinin geliştirilmesinde kullanılan blok barındırır. Bu bloklar uçağın lineer olmayan hareket denklemleri, motor-piston gücü, uçak ataletparametreleri, atmosfer modeli, dünya modeli, sensör ve aktüatörler, çerçeve dönüşümleri, pilot arayüzleri (joystick) gibi blokları barındırır.
İnsansız hava araçlarını (İHA) aslında mürettebatın bulunmadığı uçaklar olarak tanımlanabilir. Aslında daha komplike olan bu durumu açıklamak gerekirse pilotajın radyo sinyali yada tamamen otomatik olarak yapıldığı hava araçlarıdır.
Genellikle küçük İHA'lar düşük yüksekliklerde (genellikle 1000mt) kullanılırlar. Kullanım amaçları sivil ve askeri olarak ikiye ayrılabilir. Sivil amaçlı kullanımına örnek olarak; meteorolojik gözlemleme, tarımsal serpme ve gözlemleme, video ve fotoğraflama, sahil kıyısı araştırma, ara ve kurtarma, vahşi doğa gözlemleme, afet kontrolü, petrol şirketlerinin alan gözlemlesi ve boru güvenliği, trafik görüntüleme ve kontrolü gibi alanlarda kullanılır. Askeri olarak ise; düşman izi arama, suni sinyallerle güdümlü füzeleri tuzağa düşürmekte, radyo sinyallerini yaymakta, kıyı taarruzlarını engellemekte, hafif bombardımanda, nükleer, biyolojik ve kimyasal açıdan tehlikeli bölgelerin incelenmesinde, uzun süre ve yüksek mesafelerdeki uçuşlarda, radar sistemini bozmak ve yok etmek amaçlı ve benzeri alanlarda kullanılır.
Günümüz tanımına uyan ilk insansız hava aracı olarak Almanya'nın 2. Dünya Savaşında kullandığı ve mekanik gaz kesiciler ve mekanik gayroskopların yardımıyla uçurulan uçak gösterilebilir. Günümüzde ise elektromekanik parçaların gelişmesi ve GPS teknolojisi sayesinde insansız bir hava aracının uçurulması çok daha kolaylaşmıştır.
İnsansız hava aracının kontrolü genel olarak 2 farklı kısıma ayrılır. Bunlardan ilki uçağın kontrolünün merkezden bir pilot kontrolüyle radyo sinyalleri ile yapılmasıdır.

Bir diğeri ise kontrolörün insansız hava aracının bünyesinde olduğu ve önceden belirlenen bir yörüngeyi izleyecek şekilde ayarlanmış olduğu kontrol yapısıdır.
İlk belirtilen kontrol teorisindeki en büyük handikap merkezden gelen pilotaj sinyallerinin doğal yollarla yada düşman tarafından kesintiye uğratılması yada taklit edilerek uçağın rotasının değiştirilmesidir. Bu yüzden tam otonom kontrol insansız hava araçlarında daha fazla tercih edilen bir yöntemdir.
Görüldüğü gibi insansız hava araçları genellikle alçakta ve zor şartlarda uçuş yapar. Alçaktan uçuş yaptığı için herhangi bir yere çarparak kaza yapma riski artar. Bu yükseklikte hava değişimleri ve rüzgarlar daha çok olduğu için uçuşu zorlaştırır. Ve yerine getirmesi gereken görevler genellikle zorlayıcı görevler olduğu için zaten oldukça otonom olan bu uçaklara bu durum lar da gözetilerek oldukça gelişmiş ve güçlü oto pilot tasarlamak gerekir.
Bu bahsedilen durumlar dikkate alınarak model tabanlı öngörülü kontrolörün (MPC) insansız hava aracı için iyi bir controlör olacağı düşünülmüştür. MPC, bir sistemi öngörü ve kontrol stratejilerinin combinasyonuyla kontrol altında tutar. Kontol stratejisi öngörülen sistem sinyallerini ele alarak giriş değerlerini manupile eder ve istenilen değerlerin elde edilmesini sağlar. Bu esnada sistemin kısıtlarını da dikkate alınır. Bu kısıtlar modelin yani uçağın fiziksel kısıtları olmakla birlikte dilenilen güveni sınırlar da olabilir. MPC anlık giriş çıkış sinyallerini uçağın doğrusal modeline uygular ve gelecek sinyalleri hesaplar. Bu gelecek sinyaller doğrultusunda sisteme o an uygulanması gereken en optimum kontrol giriş sinyallerini uygular. Her adımda bu prosedürü tekrar eder. Optimizasyon problemine kısıtlar dahil edilerek elde edilen kontrol sinyallerinin kısıtlara uygun olması sağlanır. Ayrıca optimizasyon probleminde giriş ve çıkışlar üzerindeki ağırlıklar değiştirilerek istenilen optimum çıkış değerleri elde edilebilir.
Bu bağlamda insansız hava araçlarında daha öncelerde de model tabanlı öngörülü kontrol uygulandı. Fakat bu tezde diğer çalışmalardan farklı olarak sistemin dört giriş ve yedi çıkışı tek bir kontrolör tarafından kontrol edildi.
Tek bir kontrolör kullanılmasının sisteme oldukça getirisi olacağı öngörüldü. Tek bir kontrolör sayesinde insansız hava aracının bütün giriş ve çıkışlarına ayrı kontrolör tasarlamak yerine tek bir kontrolör ile bütün giriş çıkışlar kontrol edilir. Bütün giriş ve çıkışlar tek bir kontrolör ile kontrol edildiği için sistemdeki etkileşimler de dikkate alınmış oldu. Her ne kadar uçağın enlemsel ve boylamsal hareketlerinin göz ardı edilecek kadar etkileşimde olduğu söylense de aslında bir etkileşim söz konusudur. Bu etkileşimler MPC tarafından dikkate alındığı için çok daha güvenli ve gürbüz bir kontrolör elde edilmiş olur. Ayrıca bütün giriş ve çıkışlar tek bir MPC ile kontrol edildiği için çok daha az hesaplama eforu harcanır.
İHA'ların çok fazla doğrusal olmayışından oransal-integral,türevsel denetleyici(PID), yapay sinir ağları (NN), bulanık mantık(FL) gibi birçok gelişmiş kontrol algoritması bu uçakların otopilotlarında denenmiştir.
Bunların arasında is en çok PID kontrolör tercih edilmektedir. PID kontrolörün diğer kontrolcülere göre en büyük avantajı basit yapısıdır. Basit yapısı sayesinde en karmaşık modeller dahi PID kontrolör ile kontrol edilebilir. Fakat PID kontrolörün negatif yanları yukarıda MPC'de bahsedildiği gibi her bir giriş çıkış sinyali için bir PID kontrolör tasarlanması gerektiği, etkileşimi göz önünde bulunduramaması ve kısıtları dikkate alamamasıdır.
Günümüzde teknolojinin çok gelişmesi ve mikro elektromekanik sistemlerin ucuzlamasıyla birçok gelişmiş kontrol algoritmasını bu uçaklarda kolayca kullanma imkanı yakalanmıştır.

Bu sayede MPC gibi gelişmiş dijital kontrol algoritmaları uygulanabilir hale gelmiştir. Bu tezde yedi çıkış dört girişe uygulanan MPC'ye sistemin giriş ve bazı çıkışları üzerindeki kısıtlar tanıtılmıştır. Daha sonra bir çok farklı ağırlık parametreleri denenerek en iyi sonucu veren ağırlık değerlerinin sonuçları verilmiştir.
MPC 0.02 saniyelik bir periyotta çalışmaktadır. 70 adım sonrasını öngörüp 15 adım sonrası içinde kontrol sinyalini hesaplamaktadır.
90 saniyelik simulasyon un 20. Saniyesinde 1000 metrede 23 m/s hızla seyreden uçağın hızı 25 m/s çıkarılmıştır. Giriş ve çıkış sinyalleri incelendiğinde sistemin bu adım girişine çok hızlı ve bir cevap verip referans değere eriştiği görülmüştür. Ayrıca hiçbir giriş sinyali verilen kısıtı aşmamıştır. Diğer çıkışların bu basamak girişi sonrası etkilendiği fakat kontrolörün kısa zamanda bütün çıkış değerlerini tekrar denge konumuna getirdiği gözlemlenmiştir.
Bu basamak girişinden en fazla etkilenen dikkat çekici çıkış ise yuvarlanma açısı olmuştur. Bu durum şu şekilde açıklanmaktadır: Tek motorlu uçaklarda rotorda elde edilen rotasyonel moment uçağın kendisini ters yönde yuvarlanmaya sebep olur. Kontrolör daha uçuşun en başında bu etkiyi sıfırlayacak kontrol sinyallerini üretir. Fakat referans hızdaki artış sonrası motor devrinin de ani bir şekilde artmış olmasından dolayı yuvarlanma açısındaki denge bozulur. Fakat MPC kontrolör kısa zamanda yuvarlanma açısındaki değişimi de tekrar denge konumuna getirir. Buradaki değişimler ayrıca FlightGear isimli görsel simulasyon programında da gözlemlenebilir.

1. INTRODUCTION

A simply view of an unmanned aircraft is that it is an aircraft and aircrew removed and controlled by a computer system and a radio-link. Actually it is more complicated than this, and the aircraft must be well designed, from the beginning, without aircrew and their accommodation, etc. The aircrafts autopilot is merely part, altough an important part, of a total system. The whole system profits from designing, from the start, as a complete system which, briefly comprises:

a) A control station (CS) which store the system operators, the interfaces between the operators and the rest of the system;

b) The UAV carrying the payload which may be of many types;

c) The system of communication between the control station which transmits control inputs to the UAV and returns payload and other data from the aircraft to the CS (this is usually achieved by radio transmission);

d) Support equipment which may include maintenance and transport items.

1.1 Some Applications of UAV

Before looking into UAV in more detail, it is better to list some of the uses. Civilian uses are: Aerial photography film, video, still, etc. Herd monitoring and driving. Agriculture monitoring and spraying. Conservation pollution and land monitoring. Fire services and forestry fire detection, incident control. Coastguard search and rescue, coastline and sea lane monitoring. Customs and excise surveillance for illegal imports. Energy companies powerline inspection. Fisheries protection. Info service news information and pictures, wildlife pictures and e.g. Lifeboat institutions incident investigation, guidance and control. Local authorities survey, disaster control. Gas and oil supply companies land survey and pipeline security. Meteorological services analysis of atmosphere for forecasting, etc. Traffic monitoring and control of traffic. Aerial photography for mapping. Search for missing persons, security and incident surveillance. Rivers authorities water course and level monitoring, flood and pollution control. Survey organisations geographical,

geological and archaeological survey. Water boards reservoir and pipeline monitoring. Military roles: Shadowing enemy fleets. Entraping missiles by the emission of artificial signatures. Relaying radio signals. Protection of ports from offshore attack. Monitoring of sonar buoys and possibly other forms of antisubmarine. Warfare. Reconnaissance. Surveillance of enemy activity. Monitoring of (nbc) nuclear, biological or chemical contamination. Target designation and monitoring. Location and destruction of land mines. Elimination of unexploded bombs. Long-range, high-altitude surveillance. Radar system jamming and destruction. Airfield base security. Airfield damage assessment.

1.2 The History of UAV

The history of UAV is the history of all aircraft. From centuries past when Chinese kites honored the skies to the first hot air balloon, unmanned flying craft came first before the risk of someone climbing on board occurred. One early user of unmanned aircraft was by the Chinese General (180–234 ad) who used paper balloons adapted with oil-burning lamps to heat the air; he then flew these over the enemies at nights to make them believe there was a divine force at work. In modern times, UAV means an autonomous or remotely piloted air vehicle which flies about imitating the maneuvers of piloted aerial vehicle. Even the name allocated to unoccupied aerial vehicle has changed over the years as viewed by aircraft manufacturers, civil aviation authorities, and the military. Aerial torpedoes, radio controlled, remotely piloted, remote control, autonomous control, pilotless vehicle, unmanned aerial vehicles (UAVs). In the early years of aviation, the idea of flying an aircraft with no one inside had the obvious advantage of removing the risk to life and limb of these highly experimental contraptions.

As a result, many mishaps are listed where advances were made without injury to an onboard pilot. Although such ideas to remove people from the UAV were used, the lack of a satisfactory method to affect control limited the use of these early unmanned aircraft. Early aviation developmental attempts quickly turned to the use of the first “test pilots” to fly these pioneering craft. Further advances beyond unmanned gliders proved painful as even pioneer Lilienthal was killed flying an experimental glider in 1896. As seen in the modern use of unmanned aircraft, historically unmanned aircraft often followed a consistent operational pattern,

described today as the three D's, which stand for dangerous, dirty, and dull. Dangerous being that someone is either trying to bring down the aircraft or where the life of the pilot may be at undue risk operationally. Dirty is where the environment may be contaminated by chemical, biological, or radiological hazards precluding human exposure. Finally, dull is where the task requires long hours in the air making manned flight fatiguing, stressful, and therefore not desirable.

1.3 The Need of Autonomous

From the first unmanned aircraft, designers attempt to gain as much independent flight operation from manned ground control as possible. Military needs called for maximum standoff distance, long endurance, and important data flow from sensors. The require for data further competed with bandwidth for flight control transmission further driving the need for self-flight or autonomous operation. Enemy jamming may delay sensor transmission but disrupting required flight control information may cause the loss of the aircraft. The German V-1 Flying Bomb of World War II intentionally employed a crude, fully autonomous flight control and navigation system based on mechanical gyros, timers, and some ancient preprogramming include fuel shutoff to begin the dive. It was not until the advent of small, lightweight digital computers, inertia navigation technology, and the global positioning system (GPS) satellite network that autonomous UAV operation gained flight autonomy. Lightweight computer technology developed in the 1970s. This led to the worldwide explosion in personal computers and the digitalization of everyday items but the most significant role in unmanned aircraft autonomy. With high development in computer technology, UAV earned bigger flexibility in addressing changes in winds and weather conditions besides new variables affecting the mission aquipment payloads. Mapping data could now be stored aboard the aircraft, which not only improved navigation but also enabled more accurate sensor camera imagery.

1.4 The Future of UAV

The UAV control runs from a completely autonomous flight control system independent of any outside signals to one that employs a constant data link enabling a pilot to remotely fly the aircraft. A fully autonomous aircraft is able to fly without the effects of enemy signal jamming and etc. The disadvantage of a fully

autonomous UAV is that a fully autonomous flight control system can be simulated in a computer, enabling the enemy to develop counters to the system. Once the program flaws are identified, it becomes a simple task to defeat the autonomous system. Also, fully autonomous systems will usually not be allowed to employ lethal force since the chain of responsibility is not nexistent. On th other hand, an UAV that depends on an outside signal, has the potential to be jammed or worse like directing by the enemy through a false coded message, no matter how it is well encrypted. Altough realistic artificial intelligence is built to make possible an UAV to act autonomously with the intuitiveness of a human being, the responsibility factor will prevent UAV from fully replacing manned aircraft. This will me more real with passenger travel for civil applications where at least one "conductor" on board will be required to be held accountable for the actions of the aircraft and to exercise authority over the passengers.

1.5 Command and Control Elements

The meaning of autonomy is being capable for an unmanned system to achive its mission following preprogrammed instructions without operator intervention. A fully autonomous UAV is able to fly without operator intervention from takeoff to touchdown. Some kind of aircrafts are operated completely by remote control with constant operator involvement like an external pilot. The UAV's flight characteristics are stabilized by its autopilot system; however if the pilot were to be removed from the controls the aircraft would crash. And the other kind of vehicle's onboard autopilot controls everything from takeoff to landing, requiring no pilot intervention. The pilot command can occur in case of emergencies, overriding the autopilot if necessary to change the flight path or to avoid a hazard. The autopilots for UAVs are used to guide the vehicle along a appointed path via forced waypoints. Autopilot systems for UAVs are programmed with redundant technology. For safety, the feature of most UAV autopilots use is that; the system can execute a lost link procedure if communication becomes cut off between the ground control station and the UAV. There are various ways that these systems execute this procedure. Many of these protocol recall creating a lost-link profile where the mission flight profiles (airspeed, altitudes, flight path, etc) are loaded into the memory of the system prior to aircraft launch. After the UAV is launched, the autopilot will fly the mission profile

as long as it recieves radio signals from the ground control station. The lost-link profile can be changed when necessary if connectivity remains during flight. If connection with the ground station is lost during flight, the autopilot will execute its preprogrammed lost link profile.

1.6 Model Predictive Control

A model predictive controller (MPC) automates a target system (the plant) by combining a prediction strategy and a control strategy. An approximate linear system model gives the prediction. The control strategy compares predicted plant signals to purposes, and then modify available actuators to achieve the objectives while respecting the systems constraints. The constraints can contain the physical limits of the actuator, boundaries of safe operation. MPC uses the current system measurements, the current dynamic state of the real system, the MPC models, and the process variable targets and limits to calculate future changes in the dependent variables. These changes are computed to hold the dependent variables close to target while respecting the constraints on both independent and dependent variables. The MPC usually sends out just the first change in each independent variable to be executed, and repeats the calculation when the next change is needed. While many actual processes are not linear, they can often be considered to be approximately linear over a small operating range. Linear MPC approaches are used in the majority of applications with the feedback of the MPC compensating for prediction errors due to structural mismatch between the model and the process. In MPCs that consist only of linear models, the superposition principle of linear algebra enables the effect of changes in multiple independent variables to be added together to predict the response of the dependent variables. This simplifies the control problem to a series of direct matrix algebra calculations that are fast and robust.

If linear models are not sufficiently accurate to represent the real system nonlinearities, then several approaches can be used. In some of the cases, the process variables can be converted before and/or after the linear MPC model to reduce the nonlinearity. The process can be controlled with nonlinear MPC that uses a nonlinear model directly in the control application. The nonlinear model can be in the form of an empirical data fit (e.g. artificial neural networks) or a high fidelity dynamic model

based on basic mass and energy balances. The nonlinear model may be linearized to derive a Kalman or specify a model for linear MPC.

2. MODELLING UAV

2.1 Coordinate Frames

Understanding to how different bodies are oriented relative to each other in UAVs is important. It should be known how an aircraft is oriented with respect to each other. For example, how a sensor is oriented relative to the aircraft or how an antenna is oriented relative to a signal source on the ground. For this reason the various coordinate systems used to describe the position and orientation of the aircraft and its sensors, and the transformation between these coordinate systems. It is necessary to use various different coordinate systems for the following reasons:

- Newton's EOMs are derived relative to a fixed, inertial reference frame. But, motion is most easily described in a body-fixed frame.
- Aerodynamic forces and torques act on the UAV's body and are most easily described in a body-fixed reference frame.
- On board sensors like accelerometers and rate gyros measure information with respect to the body frame. Alternatively, GPS measures position, ground speed and course angle with respect to the inertial frame.

2.1.1 Rotation of reference frame

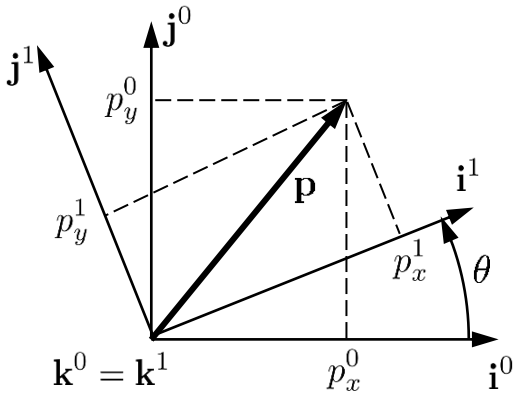


Figure 2.1 : Rotation in 2D.

The two coordinate frames shown in Figure 2.1. The vector **p** can be explained in both the F^0 frame (specified by (i^0, j^0, k^0)) and in the F^1 frame (specified by (i^1, j^1, k^1)). In the F^0 frame we got,

$$\boldsymbol{p} = p_x^0 \boldsymbol{i}^0 + p_y^0 \boldsymbol{j}^0 + p_z^0 \boldsymbol{k}^0 \tag{2.1}$$

Alternatively, in the F^1 frame we got,

$$\boldsymbol{p} = p_x^1 \boldsymbol{i}^1 + p_y^1 \boldsymbol{j}^1 + p_z^1 \boldsymbol{k}^1 \tag{2.2}$$

The vector sets $(\boldsymbol{i}^0, \boldsymbol{j}^0, \boldsymbol{k}^0)$ and $(\boldsymbol{i}^1, \boldsymbol{j}^1, \boldsymbol{k}^1)$ are each mutually vertical sets of unit basis vectors.

Setting these two expressions equal to each other gives

$$p_x^1 \boldsymbol{i}^1 + p_y^1 \boldsymbol{j}^1 + p_z^1 \boldsymbol{k}^1 = p_x^0 \boldsymbol{i}^0 + p_y^0 \boldsymbol{j}^0 + p_z^0 \boldsymbol{k}^0 \tag{2.3}$$

Taking the dot product of both sides with $\boldsymbol{i}^1, \boldsymbol{j}^1 \; and \; \boldsymbol{k}^1$ respectively and stacking

The result into matrix from gives

$$\boldsymbol{p}^1 \triangleq \begin{pmatrix} p_x^1 \\ p_y^1 \\ p_z^1 \end{pmatrix} = \begin{pmatrix} \boldsymbol{i}^1 . \boldsymbol{i}^0 & \boldsymbol{i}^1 . \boldsymbol{j}^0 & \boldsymbol{i}^1 . \boldsymbol{k}^0 \\ \boldsymbol{j}^1 . \boldsymbol{i}^0 & \boldsymbol{j}^1 . \boldsymbol{j}^0 & \boldsymbol{j}^1 . \boldsymbol{k}^0 \\ \boldsymbol{k}^1 . \boldsymbol{i}^0 & \boldsymbol{k}^1 . \boldsymbol{j}^0 & \boldsymbol{k}^1 . \boldsymbol{k}^0 \end{pmatrix} \begin{pmatrix} p_x^0 \\ p_y^0 \\ p_z^0 \end{pmatrix} \tag{2.4}$$

From the geometry of Figure 2.1 we get

$$\boldsymbol{p}^1 = R_0^1 \boldsymbol{p}^0 \tag{2.5}$$

Where

$$R_0^1 \triangleq \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.6}$$

The notation R_0^1 is used to denote a rotation from coordinate frame F^0 to coordinate frame F^1.

Proceeding in a similar way, a right-handed rotation of the coordinate system about the y-axis gives

$$R_0^1 \triangleq \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix}, \tag{2.7}$$

and a right-handed rotation of the coordinate system about the x-axis is,

$$R_0^1 \triangleq \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{pmatrix}. \quad \textbf{(2.8)}$$

The negative sign on the sine term is above the line and with just ones and zeros. The matrix R_0^1 in the above equations is example of a more general class of *orthonormal* rotation matrices that have the following properties:

$$(R_a^b)^{-1} = (R_a^b)^T = R_a^b \quad \textbf{(2.9)}$$

$$R_b^c R_a^b = R_a^c \quad \textbf{(2.10)}$$

$$det(R_a^b) = 1 \quad \textbf{(2.11)}$$

In the derivation of equation (2.5), the vector **p** remains constant and the new coordinate frame F^1 was obtained by rotating F^0 through a right-handed rotation of angle θ.

2.1.2 Rotation of a vector

As an another option, rotation matrices can be used rotate a vector through a preordained angle in a fixed reference frame. For example, left handed rotation of a vector **p** in frame F^0 about the $\boldsymbol{k}^0$-axis by the angle θ, as shown in Figure 2.2.

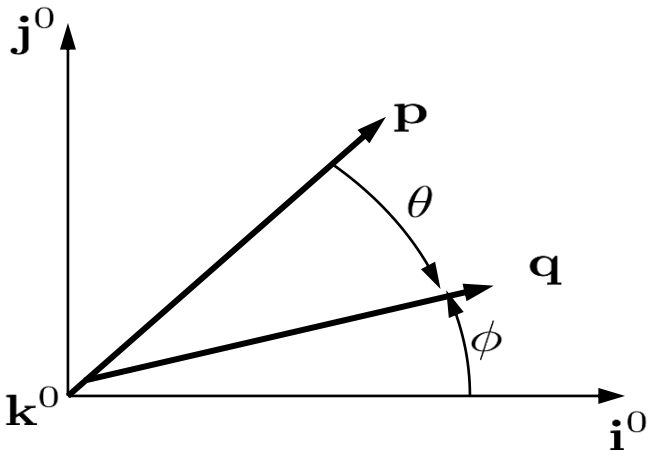


Figure 2.2 : Rotation of p about the k^0-axis.

Assuming **p** and **q** are confined to the $\boldsymbol{i}^0$-$\boldsymbol{j}^0$ plane, we can write the components of **p** and **q** as

$$\boldsymbol{p} = \begin{pmatrix} p\cos(\theta+\varphi) \\ p\sin(\theta+\varphi) \\ 0 \end{pmatrix} \quad \textbf{(2.12)}$$

$$\boldsymbol{p} = \begin{pmatrix} p\cos\theta\cos\theta - p\sin\theta\sin\varphi \\ p\sin\theta\cos\varphi + p\cos\theta\sin\varphi \\ 0 \end{pmatrix} \tag{2.13}$$

$$\boldsymbol{q} = \begin{pmatrix} q\cos\varphi \\ q\sin\varphi \\ 0 \end{pmatrix} \tag{2.14}$$

where $p \triangleq |\boldsymbol{p}| = q \triangleq |\boldsymbol{q}|$. Expressing equation (2.13) in terms of (2.14) gives

$$\boldsymbol{p} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \boldsymbol{q} \tag{2.15}$$

$$= (R_0^1)^T \boldsymbol{q} \tag{2.16}$$

and

$$\boldsymbol{q} = (R_0^1)^T \boldsymbol{p}. \tag{2.17}$$

In this situation, the rotation matrix (R_0^1) can be explained as a left-handed rotation of the vector **p** through the angle θ to a new vector **q** in the same reference frame. The right-handed rotation of a vector (in this case from **q** to **p**) can be obtained by using $(R_0^1)^T$. This interpretation contrasts with our original use of the rotatin matrix to transform a fixed vector **p** from an expression in frame F^0 to an expression in frame F^1 where F^1 has been obtained from F^0 by right-handed rotation.

2.2 UAV Coordinate Frames

2.2.1 The inertial frame F^i

The inertial coordinate system is an earth-fixed coordinate system with its origin at the specified base location. As shown in Figure 2.3, the unit vector i^i is directed north, j^i is directed east and k^i is directed toward the center of the earth, or down. This coordinate system is usually indicated as a north-east-down (NED) reference frame. It is common for north to be referred to as the inertial x direction, east to be referred to as the inertial y direction, and down to be referred to as the inertial z direction.

2.2.2 The inertial frame F^v

The origin of vehicle frame is at the center of mass of the UAV. But, the axes of F^v are aligned with axis of inertial frame F^i.

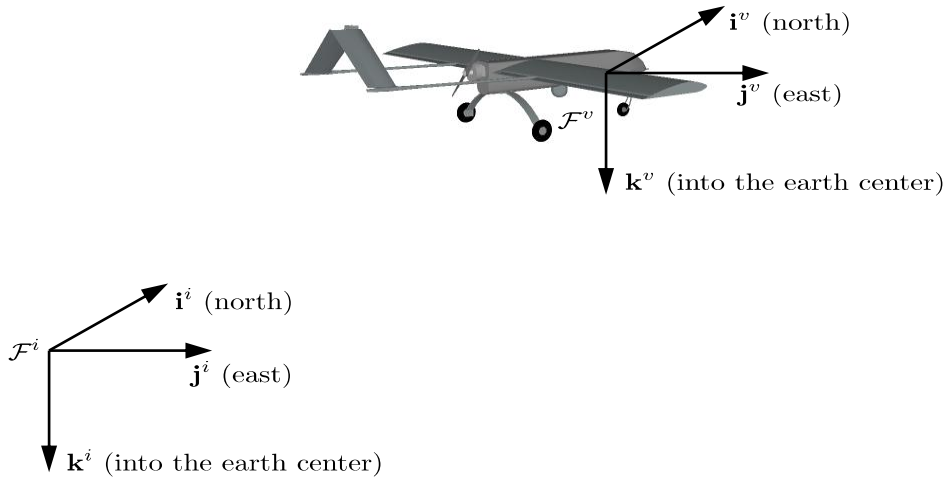


Figure 2.3: The inertial coordinate frame.

2.2.3 Euler angles

Euler angles describe the attitude of UAV. Euler angles are:

ψ: heading (yaw)

θ: elevation (pitch)

ϕ: bank (roll)

2.2.3.1 ψ Heading (yaw)

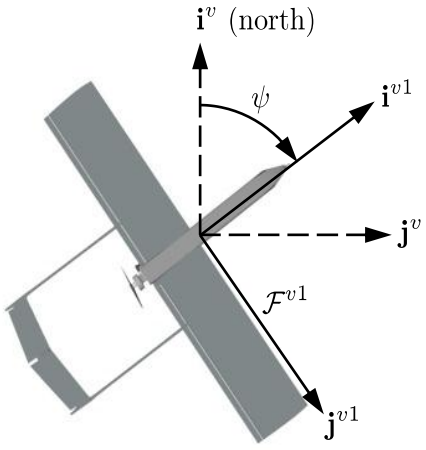


Figure 2.4: Yaw angle.

$$\mathcal{R}_v^{v1}(\psi) = \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.18}$$

$$\mathbf{p}^{v1} = \mathcal{R}_v^{v1}(\psi)\mathbf{p}^v \tag{2.19}$$

2.2.3.2 θ Elevation (pitch)

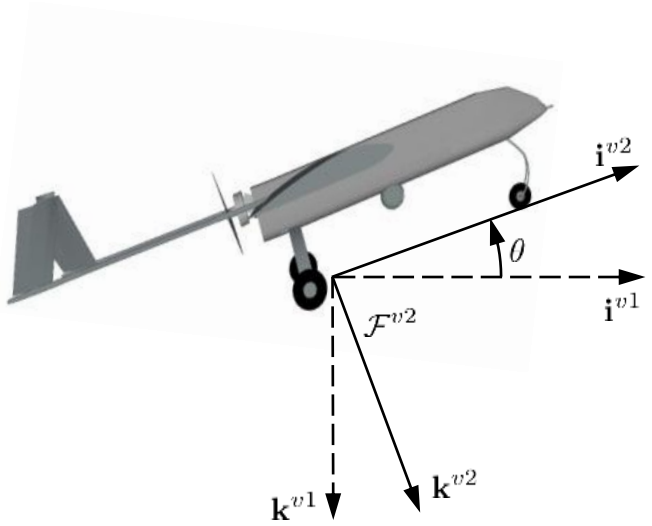


Figure 2.5: Elevation (Pitch).

$$\mathcal{R}_{v1}^{v2}(\theta) = \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \tag{2.20}$$

$$\mathbf{p}^{v2} = \mathcal{R}_{v1}^{v2}(\theta)\mathbf{p}^{v1} \tag{2.21}$$

2.2.3.3 ∅ Bank (roll)

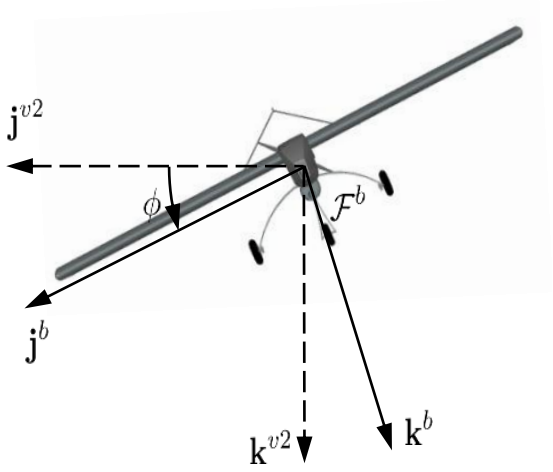


Figure 2.6: Bank Angle.

$$\mathcal{R}_{v2}^{b}(\phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{pmatrix} \tag{2.22}$$

$$\mathbf{p}^{b} = \mathcal{R}_{v2}^{b}(\phi)\mathbf{p}^{v2} \tag{2.23}$$

2.2.4 Inertial Frame to Body Frame Transformation

$$\begin{aligned} \mathcal{R}_{v}^{b}(\phi,\theta,\psi) &= \mathcal{R}_{v2}^{b}(\phi)\mathcal{R}_{v1}^{v2}(\theta)\mathcal{R}_{v}^{v1}(\psi) \\ &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{pmatrix} \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} \mathrm{c}_\theta\mathrm{c}_\psi & \mathrm{c}_\theta\mathrm{s}_\psi & -\mathrm{s}_\theta \\ \mathrm{s}_\phi\mathrm{s}_\theta\mathrm{c}_\psi - \mathrm{c}_\phi\mathrm{s}_\psi & \mathrm{s}_\phi\mathrm{s}_\theta\mathrm{s}_\psi + \mathrm{c}_\phi\mathrm{c}_\psi & \mathrm{s}_\phi\mathrm{c}_\theta \\ \mathrm{c}_\phi\mathrm{s}_\theta\mathrm{c}_\psi + \mathrm{s}_\phi\mathrm{s}_\psi & \mathrm{c}_\phi\mathrm{s}_\theta\mathrm{s}_\psi - \mathrm{s}_\phi\mathrm{c}_\psi & \mathrm{c}_\phi\mathrm{c}_\theta \end{pmatrix} \end{aligned} \tag{2.24}$$

$$\mathbf{p}^{b} = \mathcal{R}_{v}^{b}(\theta)\mathbf{p}^{v} \tag{2.25}$$

2.3 Airspeed, Wind Speed and Ground Speed

When developing the dynamic equations of motion for a UAV, the inertial forces experienced by the UAV are dependent on velocities and accelerations relative to fixed (inertial) reference frame. The aerodynamic forces, however, depend on the velocity of the airframe relative to the surrounding air. When wind is not present, these velocities are same. However, wind is almost always present with UAVs and it must be carefully distinguished between airspeed, represented by the velocity with respect to the surrounding air V_a and the ground speed, represented by the velocity with respect to the inertial frame V_g. These velocities are related by the expression

$$V_a = V_g - V_w, \tag{2.26}$$

where V_w is the wind velocity relative to the inertial frame.

$$\mathbf{V}_{g}^{b} = \begin{pmatrix} u \\ v \\ w \end{pmatrix} \tag{2.27}$$

where V_g^b is the velocity of UAV with respect to the inertial frame.

$$\mathbf{V}_a^b = \begin{pmatrix} u_r \\ v_r \\ w_r \end{pmatrix} = \mathcal{R}_w^b \begin{pmatrix} V_a \\ 0 \\ 0 \end{pmatrix}$$
$$= \begin{pmatrix} \cos\beta\cos\alpha & -\sin\beta\cos\alpha & -\sin\alpha \\ \sin\beta & \cos\beta & -\sin\beta\sin\alpha \\ \cos\beta\sin\alpha & 0 & \cos\alpha \end{pmatrix} \begin{pmatrix} V_a \\ 0 \\ 0 \end{pmatrix} \tag{2.28}$$

$$\begin{pmatrix} u_r \\ v_r \\ w_r \end{pmatrix} = V_a \begin{pmatrix} \cos\alpha\cos\beta \\ \sin\beta \\ \sin\alpha\cos\beta \end{pmatrix} \tag{2.29}$$

$$V_a = \sqrt{u_r^2 + v_r^2 + w_r^2} \tag{2.30}$$

$$\alpha = \tan^{-1}\left(\frac{w_r}{u_r}\right) \tag{2.31}$$

$$\beta = \sin^{-1}\left(\frac{v_r}{\sqrt{u_r^2 + v_r^2 + w_r^2}}\right) \tag{2.32}$$

Combining expressions, we can express the airspeed vector body-frame components in terms of the airspeed magnitude, angle of attack, and side slip angles.

3. EQUATIONS OF MOTION OF AIRCRAFT

To develop control strategies for UAV, the dynamic model of UAV must be developed. Firstly, the expressions for the kinematics and the dynamics of a rigid body will be derived.

3.1 States

To develop UAVs equations of motion, twelve state variables will be used. These states are; three position states, three velocity states, three angular position, three angular velocity.

Table 3.1: State Variables of UAV.

Name	Description
p_n	Inertial north position of MAV along $\mathbf{i}^i$ in $\mathcal{F}^i$.
p_e	Inertial east position of MAV along $\mathbf{j}^i$ in $\mathcal{F}^i$.
p_d	Inertial down position of MAV along $\mathbf{k}^i$ in $\mathcal{F}^i$.
u	Body frame velocity measured along $\mathbf{i}^b$ in $\mathcal{F}^b$.
v	Body frame velocity measured along $\mathbf{j}^b$ in $\mathcal{F}^b$.
w	Body frame velocity measured along $\mathbf{k}^b$ in $\mathcal{F}^b$.
ϕ	Roll angle defined with respect to $\mathcal{F}^{v2}$.
θ	Pitch angle defined with respect to $\mathcal{F}^{v1}$.
ψ	Heading (yaw) angle defined with respect to $\mathcal{F}^v$.
p	Roll rate measured along $\mathbf{i}^b$ in $\mathcal{F}^b$.
q	Pitch rate measured along $\mathbf{j}^b$ in $\mathcal{F}^b$.
r	Yaw rate measured along $\mathbf{k}^b$ in $\mathcal{F}^b$.

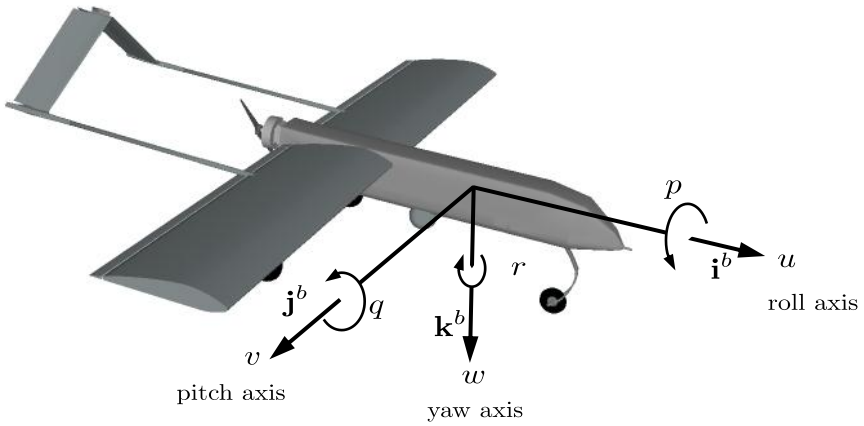


Figure 3.1: Definition of axes motion.

3.2 Kinematics

The translational velocity and position requires differentiation and rotational transformation.

$$\frac{d}{dt}\begin{pmatrix} p_n \\ p_e \\ p_d \end{pmatrix} = \mathcal{R}_b^v \begin{pmatrix} u \\ v \\ w \end{pmatrix} = (\mathcal{R}_v^b)^\top \begin{pmatrix} u \\ v \\ w \end{pmatrix} \tag{3.1}$$

The relationship between Euler angles and rates.

$$\begin{pmatrix} \dot{p}_n \\ \dot{p}_e \\ \dot{p}_d \end{pmatrix} = \begin{pmatrix} \mathrm{c}_\theta \mathrm{c}_\psi & \mathrm{s}_\phi \mathrm{s}_\theta \mathrm{c}_\psi - \mathrm{c}_\phi \mathrm{s}_\psi & \mathrm{c}_\phi \mathrm{s}_\theta \mathrm{c}_\psi + \mathrm{s}_\phi \mathrm{s}_\psi \\ \mathrm{c}_\theta \mathrm{s}_\psi & \mathrm{s}_\phi \mathrm{s}_\theta \mathrm{s}_\psi + \mathrm{c}_\phi \mathrm{c}_\psi & \mathrm{c}_\phi \mathrm{s}_\theta \mathrm{s}_\psi - \mathrm{s}_\phi \mathrm{c}_\psi \\ -\mathrm{s}_\theta & \mathrm{s}_\phi \mathrm{c}_\theta & \mathrm{c}_\phi \mathrm{c}_\theta \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} \tag{3.2}$$

3.3 Equations of Motion

Newton's second law is applied to translational degrees of freedom and rotational degrees of freedom.

3.3.1 Translational motion

$$m\frac{dV_g}{dt_i} = \mathbf{f} \tag{3.3}$$

m is the mass of UAV, time derivative in the inertial frame. **f** is the sum of all external forces acting on UAV. The derivative of velocity taken in the inertial frame can be written in terms of the derivative in the body frame and the angular velocity.

$$\begin{aligned} \begin{pmatrix} p \\ q \\ r \end{pmatrix} &= \begin{pmatrix} \dot{\phi} \\ 0 \\ 0 \end{pmatrix} + \mathcal{R}_{v2}^b(\phi) \begin{pmatrix} 0 \\ \dot{\theta} \\ 0 \end{pmatrix} + \mathcal{R}_{v2}^b(\phi)\mathcal{R}_{v1}^{v2}(\theta) \begin{pmatrix} 0 \\ 0 \\ \dot{\psi} \end{pmatrix} \\ &= \begin{pmatrix} \dot{\phi} \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{pmatrix} \begin{pmatrix} 0 \\ \dot{\theta} \\ 0 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{pmatrix} \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ \dot{\psi} \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{pmatrix} \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} \end{aligned} \tag{3.4}$$

$$\frac{d\mathbf{V}_g}{dt_i} = \frac{d\mathbf{V}_g}{dt_b} + \boldsymbol{\omega}_{b/i} \times \mathbf{V}_g \tag{3.5}$$

With combining results an alternative representation of Newton's second law with differentiation carried out in the body frame:

$$\mathsf{m}\left(\frac{d\mathbf{V}_g}{dt_b}+\boldsymbol{\omega}_{b/i}\times\mathbf{V}_g\right)=\mathbf{f} \tag{3.6}$$

In body frame :

$$\mathsf{m}\left(\frac{d\mathbf{V}_g^b}{dt_b}+\boldsymbol{\omega}_{b/i}^b\times\mathbf{V}_g^b\right)=\mathbf{f}^b \tag{3.7}$$

$$\frac{d\mathbf{V}_g^b}{dt_b}=\begin{pmatrix}\dot{u}\\ \dot{v}\\ \dot{w}\end{pmatrix} \tag{3.8}$$

$$\begin{pmatrix}\dot{u}\\ \dot{v}\\ \dot{w}\end{pmatrix}=\begin{pmatrix}rv-qw\\ pw-ru\\ qu-pv\end{pmatrix}+\frac{1}{\mathsf{m}}\begin{pmatrix}f_x\\ f_y\\ f_z\end{pmatrix} \tag{3.9}$$

3.3.2 Rotational motion

Here is the Newton's second law for the rotational motion;

$$\frac{d\mathbf{h}}{dt_i}=\mathbf{m} \tag{3.10}$$

$$\frac{d\mathbf{h}^b}{dt_b}+\boldsymbol{\omega}_{b/i}^b\times\mathbf{h}^b=\mathbf{m}^b \tag{3.11}$$

h is the angular momentum vector form and **m** is the sum of all externally applied moments. Time derivative taken with respect to inertial frame.

Expressed in the body frame,

$$\mathbf{h}^b\triangleq\mathbf{J}\boldsymbol{\omega}_{b/i}^b \tag{3.12}$$

$$\begin{aligned}\mathbf{J}&=\begin{pmatrix}\int(y^2+z^2)\,d\mathsf{m} & -\int xy\,d\mathsf{m} & -\int xz\,d\mathsf{m}\\ -\int xy\,d\mathsf{m} & \int(x^2+z^2)\,d\mathsf{m} & -\int yz\,d\mathsf{m}\\ -\int xz\,d\mathsf{m} & -\int yz\,d\mathsf{m} & \int(x^2+y^2)\,d\mathsf{m}\end{pmatrix}\\ &\triangleq\begin{pmatrix}J_x & -J_{xy} & -J_{xz}\\ -J_{xy} & J_y & -J_{yz}\\ -J_{xz} & -J_{yz} & J_z\end{pmatrix}\end{aligned} \tag{3.13}$$

For a rigid body, angular momentum is defined as the product of the inertia matrix and the angular velocity vector:

$$\frac{d\mathbf{h}^b}{dt_b} + \boldsymbol{\omega}_{b/i}^b \times \mathbf{h}^b = \mathbf{m}^b \tag{3.14}$$

Diagonal elements are called moments of inertia. Off-diagonal elements are called products of inertia. Determined from mass properties in CAD program or measured experimentally using a bifilar pendulum.

Because is unchanging in the body frame,

$$\frac{d\mathbf{J}}{dt_b} = 0 \tag{3.15}$$

$$\mathbf{J}\frac{d\boldsymbol{\omega}_{b/i}^b}{dt_b} + \boldsymbol{\omega}_{b/i}^b \times \left(\mathbf{J}\boldsymbol{\omega}_{b/i}^b\right) = \mathbf{m}^b \tag{3.16}$$

$$\dot{\boldsymbol{\omega}}_{b/i}^b = \mathbf{J}^{-1}\left[-\boldsymbol{\omega}_{b/i}^b \times \left(\mathbf{J}\boldsymbol{\omega}_{b/i}^b\right) + \mathbf{m}^b\right] \tag{3.17}$$

$$\dot{\boldsymbol{\omega}}_{b/i}^b = \frac{d\boldsymbol{\omega}_{b/i}^b}{dt_b} = \begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} \tag{3.18}$$

$$\frac{d\mathbf{h}}{dt_i} = \frac{d\mathbf{h}}{dt_b} + \boldsymbol{\omega}_{b/i} \times \mathbf{h} = \mathbf{m} \tag{3.19}$$

If the aircraft is symmetric about the $\mathbf{i}^b$-$\mathbf{k}^b$ plane, then $J_{xy} = J_{yz} = 0$ and

$$\mathbf{J} = \begin{pmatrix} J_x & 0 & -J_{xz} \\ 0 & J_y & 0 \\ -J_{xz} & 0 & J_z \end{pmatrix} \tag{3.20}$$

This symmetry assumption helps simplify the analysis. The inverse of becomes

$$\begin{aligned} \mathbf{J}^{-1} = \frac{\mathrm{adj}(\mathbf{J})}{\det(\mathbf{J})} &= \frac{\begin{pmatrix} J_y J_z & 0 & J_y J_{xz} \\ 0 & J_x J_z - J_{xz}^2 & 0 \\ J_{xz} J_y & 0 & J_x J_y \end{pmatrix}}{J_x J_y J_z - J_{xz}^2 J_y} \\ &= \begin{pmatrix} \frac{J_z}{\Gamma} & 0 & \frac{J_{xz}}{\Gamma} \\ 0 & \frac{1}{J_y} & 0 \\ \frac{J_{xz}}{\Gamma} & 0 & \frac{J_x}{\Gamma} \end{pmatrix} \end{aligned} \tag{3.21}$$

$$\Gamma \triangleq J_x J_z - J_{xz}^2 \tag{3.22}$$

The components of the externally applied moment about the $\boldsymbol{i}^b, \boldsymbol{j}^b, \boldsymbol{k}^b$ axes as $\boldsymbol{m}^b \triangleq (l, m, n)^T$, this equation $\dot{\boldsymbol{\omega}}^b_{b/i} = \mathbf{J}^{-1}\left[-\boldsymbol{\omega}^b_{b/i} \times \left(\mathbf{J}\boldsymbol{\omega}^b_{b/i}\right) + \mathbf{m}^b\right]$ can be written in component from as;

$$\mathbf{a} \times \mathbf{b} = \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \tag{3.23}$$

$$\begin{aligned} \begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} &= \begin{pmatrix} \frac{J_z}{\Gamma} & 0 & \frac{J_{xz}}{\Gamma} \\ 0 & \frac{1}{J_y} & 0 \\ \frac{J_{xz}}{\Gamma} & 0 & \frac{J_x}{\Gamma} \end{pmatrix} \left[\begin{pmatrix} 0 & r & -q \\ -r & 0 & p \\ q & -p & 0 \end{pmatrix} \begin{pmatrix} J_x & 0 & -J_{xz} \\ 0 & J_y & 0 \\ -J_{xz} & 0 & J_z \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} + \begin{pmatrix} l \\ m \\ n \end{pmatrix} \right] \\ &= \begin{pmatrix} \frac{J_z}{\Gamma} & 0 & \frac{J_{xz}}{\Gamma} \\ 0 & \frac{1}{J_y} & 0 \\ \frac{J_{xz}}{\Gamma} & 0 & \frac{J_x}{\Gamma} \end{pmatrix} \left[\begin{pmatrix} J_{xz}pq + (J_y - J_z)qr \\ J_{xz}(r^2 - p^2) + (J_z - J_x)pr \\ (J_x - J_y)pq - J_{xz}qr \end{pmatrix} + \begin{pmatrix} l \\ m \\ n \end{pmatrix} \right] \\ &= \begin{pmatrix} \Gamma_1 pq - \Gamma_2 qr + \Gamma_3 l + \Gamma_4 n \\ \Gamma_5 pr - \Gamma_6(p^2 - r^2) + \frac{1}{J_y} m \\ \Gamma_7 pq - \Gamma_1 qr + \Gamma_4 l + \Gamma_8 n \end{pmatrix} \end{aligned} \tag{3.24}$$

$$\begin{aligned} \Gamma_1 &= \frac{J_{xz}(J_x - J_y + J_z)}{\Gamma} & \Gamma_4 &= \frac{J_{xz}}{\Gamma} & \\ \Gamma_2 &= \frac{J_z(J_z - J_y) + J_{xz}^2}{\Gamma} & \Gamma_5 &= \frac{J_z - J_x}{J_y} & \Gamma_7 &= \frac{(J_x - J_y)J_x + J_{xz}^2}{\Gamma} \\ \Gamma_3 &= \frac{J_z}{\Gamma} & \Gamma_6 &= \frac{J_{xz}}{J_y} & \Gamma_8 &= \frac{J_x}{\Gamma} \end{aligned} \tag{3.25}$$

$$\Gamma \triangleq J_x J_z - J_{xz}^2 \tag{3.26}$$

$$\begin{pmatrix} \dot{p}_n \\ \dot{p}_e \\ \dot{p}_d \end{pmatrix} = \begin{pmatrix} \mathrm{c}_\theta \mathrm{c}_\psi & \mathrm{s}_\phi \mathrm{s}_\theta \mathrm{c}_\psi - \mathrm{c}_\phi \mathrm{s}_\psi & \mathrm{c}_\phi \mathrm{s}_\theta \mathrm{c}_\psi + \mathrm{s}_\phi \mathrm{s}_\psi \\ \mathrm{c}_\theta \mathrm{s}_\psi & \mathrm{s}_\phi \mathrm{s}_\theta \mathrm{s}_\psi + \mathrm{c}_\phi \mathrm{c}_\psi & \mathrm{c}_\phi \mathrm{s}_\theta \mathrm{s}_\psi - \mathrm{s}_\phi \mathrm{c}_\psi \\ -\mathrm{s}_\theta & \mathrm{s}_\phi \mathrm{c}_\theta & \mathrm{c}_\phi \mathrm{c}_\theta \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} \tag{3.27}$$

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} = \begin{pmatrix} rv - qw \\ pw - ru \\ qu - pv \end{pmatrix} + \frac{1}{\mathsf{m}} \begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix} \tag{3.28}$$

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \sin\phi \tan\theta & \cos\phi \tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} \tag{3.29}$$

$$\begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = \begin{pmatrix} \Gamma_1 pq - \Gamma_2 qr \\ \Gamma_5 pr - \Gamma_6(p^2 - r^2) \\ \Gamma_7 pq - \Gamma_1 qr \end{pmatrix} + \begin{pmatrix} \Gamma_3 l + \Gamma_4 n \\ \frac{1}{J_y} m \\ \Gamma_4 l + \Gamma_8 n \end{pmatrix} \tag{3.30}$$

These equations represent the dynamics of UAV.

3.4 Forces and Moments

Three sources of force and moment is assumed to act to UAV. These forces and moments are, gravity,aerodynamics and propulsion.

$\boldsymbol{f}_g$: force of gravity

$\boldsymbol{f}_a$, $\boldsymbol{m}_a$: aerodynamic forces and moments

$\boldsymbol{f}_p$, $\boldsymbol{m}_p$: moments and forces of propulsion

$$\boldsymbol{f} = \boldsymbol{f}_g + \boldsymbol{f}_a + \boldsymbol{f}_p \tag{3.31}$$

$$\boldsymbol{m} = \boldsymbol{m}_a + \boldsymbol{m}_p \tag{3.32}$$

$\boldsymbol{f}$: total force acting on the airframe

$\boldsymbol{m}$: total moment acting on the airframe

$$\begin{aligned} \mathrm{f}_g^b &= \mathcal{R}_v^b \begin{pmatrix} 0 \\ 0 \\ \mathrm{m}g \end{pmatrix} \\ &= \begin{pmatrix} -\mathrm{m}g\sin\theta \\ \mathrm{m}g\cos\theta\sin\phi \\ \mathrm{m}g\cos\theta\cos\phi \end{pmatrix} \end{aligned} \tag{3.33}$$

3.4.1 Aerodynamic forces and moments

While UAV passing through air, a pressure is affecting on airframe like in Figure 3.2 The pressure acting on UAV is function of airspeed, shape of UAV, attitude and the air denisty. The dynamic pressure is given by :

$$\tfrac{1}{2}\rho V_a^2 \tag{3.34}$$

Where ρ is the air denisty, V_a is the airspeed.

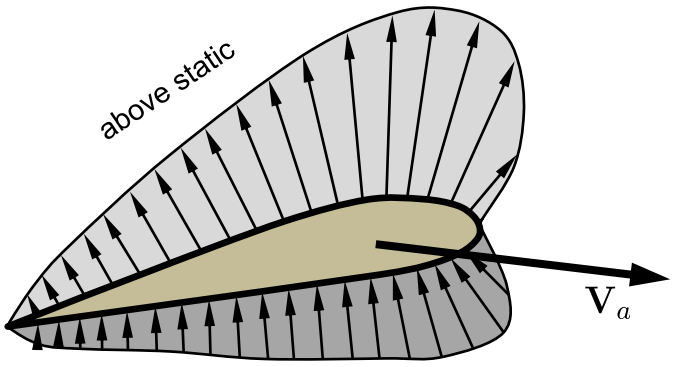


Figure 3.2: Pressure distribution around wing.

The lift, drag and moment are expressed as

$$F_{\text{lift}} = \frac{1}{2}\rho V_a^2 S C_L \tag{3.35}$$

$$F_{\text{drag}} = \frac{1}{2}\rho V_a^2 S C_D \tag{3.36}$$

$$m = \frac{1}{2}\rho V_a^2 S c C_m \tag{3.37}$$

C_L, C_D, C_m are aerodynamic coefficients. S is the planform area of UAV wing. c is the mean chord of UAV wing.

3.4.2 Control surfaces

The control surfaces are used to maneuver the UAV. These control surfaces are elevator, aileron and rudder.

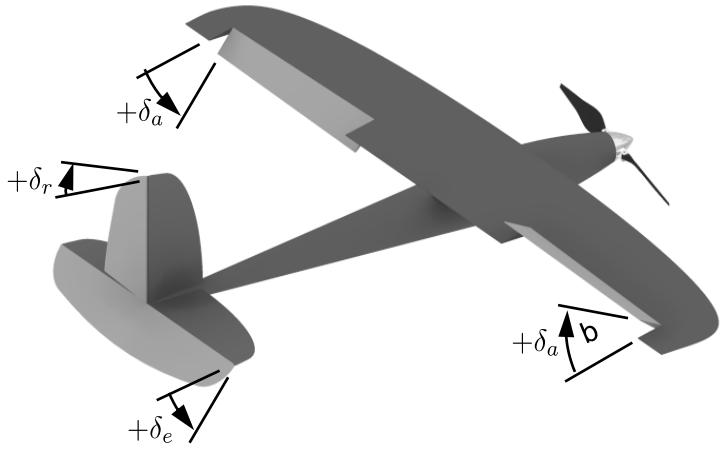


Figure 3.3: Control surfaces deflection.

δ_a : Aileron deflection. Ailerons are used to control the bank angle $\emptyset$.

δ_r : Rudder deflection. Rudder is used to control the yaw angle ψ.

δ_e : Elevator deflection. Elevator is used to control the pitch angle θ.

Aircraft dynamics and aerodynamics are commonly separated into two groups:

Longitudinal: Up-down, pitch plane, pitching motions

Lateral-directional: Side-to-side, turning motions (bank and yaw)

3.4.2.1 Longitudinal aerodynamics

The longitudinal aerodynamics are the acts of UAV in $\boldsymbol{i}^b - \boldsymbol{k}^b$ plane. It also called as pitch plane. Longitudinal dynamics are heavily influenced by angle of attack pitch rate and elevator deflection.

$$F_{\text{lift}} \approx \frac{1}{2}\rho V_a^2 S C_L(\alpha, q, \delta_e) \tag{3.38}$$

$$F_{\text{drag}} \approx \frac{1}{2}\rho V_a^2 S C_D(\alpha, q, \delta_e) \tag{3.39}$$

$$m \approx \frac{1}{2}\rho V_a^2 S c C_m(\alpha, q, \delta_e) \tag{3.40}$$

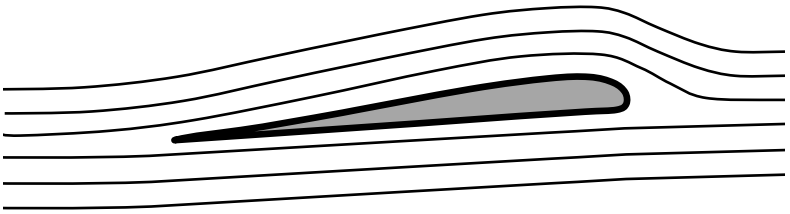


Figure 3.4: Linear aerodynamic model.

The force and moment equations are nonlinear. But for small angle of attack, the flow over the wing is laminar and attached. That's why forces and moments can be accepted as linear. First order Taylor series approximation of the forces and moment can be written as:

$$F_{\text{lift}} = \frac{1}{2}\rho V_a^2 S \left[C_{L_0} + C_{L_\alpha}\alpha + C_{L_q}\frac{c}{2V_a}q + C_{L_{\delta_e}}\delta_e\right] \tag{3.41}$$

$$F_{\text{drag}} = \frac{1}{2}\rho V_a^2 S \left[C_{D_0} + C_{D_\alpha}\alpha + C_{D_q}\frac{c}{2V_a}q + C_{D_{\delta_e}}\delta_e\right] \tag{3.42}$$

$$m = \frac{1}{2}\rho V_a^2 S c \left[C_{m_0} + C_{m_\alpha}\alpha + C_{m_q}\frac{c}{2V_a}q + C_{m_{\delta_e}}\delta_e\right] \tag{3.43}$$

To express longitudinal forces in the body frame. A rotation is required by angle of attack.

$$\begin{pmatrix} f_x \\ f_z \end{pmatrix} = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} -F_{\text{drag}} \\ -F_{\text{lift}} \end{pmatrix}$$

$$= \frac{1}{2}\rho V_a^2 S \begin{pmatrix} \begin{matrix}[-C_D(\alpha)\cos\alpha + C_L(\alpha)\sin\alpha] \\ + \left[-C_{D_q}\cos\alpha + C_{L_q}\sin\alpha\right]\frac{c}{2V_a}q + \left[-C_{D_{\delta_e}}\cos\alpha + C_{L_{\delta_e}}\sin\alpha\right]\delta_e \end{matrix} \\ --- \\ \begin{matrix}[-C_D(\alpha)\sin\alpha - C_L(\alpha)\cos\alpha] \\ + \left[-C_{D_q}\sin\alpha - C_{L_q}\cos\alpha\right]\frac{c}{2V_a}q + \left[-C_{D_{\delta_e}}\sin\alpha - C_{L_{\delta_e}}\cos\alpha\right]\delta_e \end{matrix} \end{pmatrix} \tag{3.44}$$

3.4.2.2 Lateral aerodynamics

Lateral aerodynamic force and moments cause translational motion in lateral direction. Motions in yaw and bank will cause changes in lateral dynamics.

f_y : The lateral force

l : The bank moment

n: The yaw moment

$$f_y = \frac{1}{2}\rho V_a^2 S C_Y(\beta, p, r, \delta_a, \delta_r) \tag{3.45}$$

$$l = \frac{1}{2}\rho V_a^2 S b C_l(\beta, p, r, \delta_a, \delta_r) \tag{3.46}$$

$$n = \frac{1}{2}\rho V_a^2 S b C_n(\beta, p, r, \delta_a, \delta_r) \tag{3.47}$$

$$f_y \approx \frac{1}{2}\rho V_a^2 S \left[C_{Y_0} + C_{Y_\beta}\beta + C_{Y_p}\frac{b}{2V_a}p + C_{Y_r}\frac{b}{2V_a}r + C_{Y_{\delta_a}}\delta_a + C_{Y_{\delta_r}}\delta_r\right] \tag{3.48}$$

$$l \approx \frac{1}{2}\rho V_a^2 S b \left[C_{l_0} + C_{l_\beta}\beta + C_{l_p}\frac{b}{2V_a}p + C_{l_r}\frac{b}{2V_a}r + C_{l_{\delta_a}}\delta_a + C_{l_{\delta_r}}\delta_r\right] \tag{3.49}$$

$$n \approx \frac{1}{2}\rho V_a^2 S b \left[C_{n_0} + C_{n_\beta}\beta + C_{n_p}\frac{b}{2V_a}p + C_{n_r}\frac{b}{2V_a}r + C_{n_{\delta_a}}\delta_a + C_{n_{\delta_r}}\delta_r\right] \tag{3.50}$$

$$\text{For symmetric aircraft, } C_{Y_0} = C_{l_0} = C_{n_0} = 0 \tag{3.51}$$

3.4.3 Propulsion forces and moments

3.4.3.1 Propeller thrust

Bernoulli's principle can be applied to calculate the pressure ahead of and behind the propeller. Then the difference of the pressure is applied to propeller area. This model is for ideal propeller.

The total pressure upstream of the propeller :

$$P_{\text{upstream}} = P_0 + \frac{1}{2}\rho V_a^2 \tag{3.52}$$

The total pressure upstream of the propeller :

$$P_{\text{downstream}} = P_0 + \frac{1}{2}\rho V_{\text{exit}}^2 \tag{3.53}$$

P_0 is the static pressure, ρ is the air denisty, V_{exit} is the speed of the air as it leaves the propeller, δ_t is PWM command.

$$V_{\text{exit}} = k_{\text{motor}}\delta_t \tag{3.54}$$

$$\begin{aligned} F_{x_p} &= S_{\text{prop}} C_{\text{prop}} (P_{\text{downstream}} - P_{\text{upstream}}) \\ &= \frac{1}{2}\rho S_{\text{prop}} C_{\text{prop}} \left[(k_{\text{motor}}\delta_t)^2 - V_a^2 \right] \end{aligned} \tag{3.55}$$

$$\mathbf{f}_p = \frac{1}{2}\rho S_{\text{prop}} C_{\text{prop}} \begin{pmatrix} (k_{\text{motor}}\delta_t)^2 - V_a^2 \\ 0 \\ 0 \end{pmatrix} \tag{3.56}$$

Propeller torque:

$$T_p = -k_{T_p}(k_\Omega \delta_t)^2 \tag{3.57}$$

$$\mathbf{m}_p = \begin{pmatrix} -k_{T_p}(k_\Omega \delta_t)^2 \\ 0 \\ 0 \end{pmatrix} \tag{3.58}$$

Wind effects :

$$\mathbf{V}_a^b = \begin{pmatrix} u_r \\ v_r \\ w_r \end{pmatrix} = \begin{pmatrix} u - u_w \\ v - v_w \\ w - w_w \end{pmatrix} \tag{3.59}$$

$$V_a = \sqrt{u_r^2 + v_r^2 + w_r^2} \tag{3.60}$$

$$\alpha = \tan^{-1}\left(\frac{w_r}{u_r}\right) \tag{3.61}$$

$$\beta = \sin^{-1}\left(\frac{v_r}{\sqrt{u_r^2 + v_r^2 + w_r^2}}\right) \tag{3.62}$$

3.5 Linear Model of UAV

As it seen the equations of motion of UAV are nonlinear. Beause of the complexity and difficulty in designing controller the nonlinear model of UAV will be linearize. The dynamics of UAV can be decomposed into two motion. These are longitudinal and lateral motions. Longitudinal motion includes airspeed, pitch angle and altitude. Lateral motion includes bank and yaw angles. In flight dynamics, force and moment equilibrium is called trim.

3.5.1 Trim conditions

Nonlinear System :

$$\dot{x} = f(x, u) \tag{3.63}$$

In equilibrium :

$$f(x^*, u^*) = 0 \tag{3.64}$$

Trim given by :

$$\dot{x}^* = f(x^*, u^*) \tag{3.65}$$

Trim states will be compute while satisfies these conditions:

- Constant Speed $\boldsymbol{V_a^*}$,
- Climbing at constant flight pat angle $\boldsymbol{\gamma}^*$,
- Constant orbit of radius $\boldsymbol{R}^*$,

States :

$$x \triangleq (p_n, p_e, p_d, u, v, w, \phi, \theta, \psi, p, q, r)^\top \tag{3.66}$$

Inputs:

$$u \triangleq (\delta_e, \delta_t, \delta_a, \delta_r)^\top \tag{3.67}$$

$$\dot{u}^* = \dot{v}^* = \dot{w}^* = 0 \tag{3.68}$$

$$\dot{\phi}^* = \dot{\theta}^* = \dot{p}^* = \dot{q}^* = 0 \tag{3.69}$$

Turn rate constant,

$$\dot{\psi}^* = \frac{V_a^*}{R^*} \cos\gamma^* \quad \rightarrow \quad \dot{r}^* = 0 \tag{3.70}$$

Climb rate constant,

$$\dot{h}^* = V_a^* \sin\gamma^* \tag{3.71}$$

Given parametersV_a^*,γ^*, and R^* can specify $\dot{x}^*$ as

$$\dot{x}^* = \begin{pmatrix} \dot{p}_n^* \\ \dot{p}_e^* \\ \dot{h}^* \\ \dot{u}^* \\ \dot{v}^* \\ \dot{w}^* \\ \dot{\phi}^* \\ \dot{\theta}^* \\ \dot{\psi}^* \\ \dot{p}^* \\ \dot{q}^* \\ \dot{r}^* \end{pmatrix} = \begin{pmatrix} [\text{don't care}] \\ [\text{don't care}] \\ V_a^* \sin\gamma^* \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{V_a^*}{R^*} \cos\gamma^* \\ 0 \\ 0 \\ 0 \end{pmatrix} = f(x^*, u^*) \tag{3.72}$$

3.5.2 Linear state space models

State equation:

$$\begin{aligned}\dot{\bar{x}} &= \dot{x} - \dot{x}^* \\ &= f(x,u) - f(x^*,u^*) \\ &= f(x + x^* - x^*, u + u^* - u^*) - f(x^*,u^*) \\ &= f(x^* + \bar{x}, u^* + \bar{u}) - f(x^*,u^*)\end{aligned} \tag{3.73}$$

Linearize around trim condition,

$$\begin{aligned}\dot{\bar{x}} &= f(x^*,u^*) + \frac{\partial f(x^*,u^*)}{\partial x}\bar{x} + \frac{\partial f(x^*,u^*)}{\partial u}\bar{u} + H.O.T - f(x^*,u^*) \\ &\approx \frac{\partial f(x^*,u^*)}{\partial x}\bar{x} + \frac{\partial f(x^*,u^*)}{\partial u}\bar{u}\end{aligned} \tag{3.74}$$

Lateral state space equation is,

$$\begin{pmatrix}\dot{\bar{v}}\\ \dot{\bar{p}}\\ \dot{\bar{r}}\\ \dot{\bar{\phi}}\\ \dot{\bar{\psi}}\end{pmatrix} = \begin{pmatrix} Y_v & Y_p & Y_r & g\cos\theta^*\cos\phi^* & 0 \\ L_v & L_p & L_r & 0 & 0 \\ N_v & N_p & N_r & 0 & 0 \\ 0 & 1 & \cos\phi^*\tan\theta^* & q^*\cos\phi^*\tan\theta^* - r^*\sin\phi^*\tan\theta^* & 0 \\ 0 & 0 & \cos\phi^*\sec\theta^* & p^*\cos\phi^*\sec\theta^* - r^*\sin\phi^*\sec\theta^* & 0 \end{pmatrix} \begin{pmatrix}\bar{v}\\ \bar{p}\\ \bar{r}\\ \bar{\phi}\\ \bar{\psi}\end{pmatrix} + \begin{pmatrix} Y_{\delta_a} & Y_{\delta_r} \\ L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix}\bar{\delta}_a \\ \bar{\delta}_r\end{pmatrix} \tag{3.75}$$

Longitudinal state space model is,

$$\begin{pmatrix}\dot{\bar{u}}\\ \dot{\bar{w}}\\ \dot{\bar{q}}\\ \dot{\bar{\theta}}\\ \dot{\bar{h}}\end{pmatrix} = \begin{pmatrix} X_u & X_w & X_q & -g\cos\theta^* & 0 \\ Z_u & Z_w & Z_q & -g\sin\theta^* & 0 \\ M_u & M_w & M_q & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \sin\theta^* & -\cos\theta^* & 0 & u^*\cos\theta^* + w^*\sin\theta^* & 0 \end{pmatrix} \begin{pmatrix}\bar{u}\\ \bar{w}\\ \bar{q}\\ \bar{\theta}\\ \bar{h}\end{pmatrix} + \begin{pmatrix} X_{\delta_e} & X_{\delta_t} \\ Z_{\delta_e} & 0 \\ M_{\delta_e} & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix}\bar{\delta}_e \\ \bar{\delta}_t\end{pmatrix} \tag{3.76}$$

Table 3.2 : Lateral State Space Model Coefficients.

Lateral	Formula
Y_v	$\frac{\rho S b v^*}{4mV_a^*}\left[C_{Y_p}p^* + C_{Y_r}r^*\right] + \frac{\rho S v^*}{\mathsf{m}}\left[C_{Y_0} + C_{Y_\beta}\beta^* + C_{Y_{\delta_a}}\delta_a^* + C_{Y_{\delta_r}}\delta_r^*\right] + \frac{\rho S C_{Y_\beta}}{2\mathsf{m}}\sqrt{u^{*2}+w^{*2}}$
Y_p	$w^* + \frac{\rho V_a^* Sb}{4m}C_{Y_p}$
Y_r	$-u^* + \frac{\rho V_a^* Sb}{4m}C_{Y_r}$
Y_{δ_a}	$\frac{\rho V_a^{*2} S}{2\mathsf{m}}C_{Y_{\delta_a}}$
Y_{δ_r}	$\frac{\rho V_a^{*2} S}{2\mathsf{m}}C_{Y_{\delta_r}}$
L_v	$\frac{\rho S b^2 v^*}{4V_a^*}\left[C_{p_p}p^* + C_{p_r}r^*\right] + \rho S b v^*\left[C_{p_0} + C_{p_\beta}\beta^* + C_{p_{\delta_a}}\delta_a^* + C_{p_{\delta_r}}\delta_r^*\right] + \frac{\rho S b C_{p_\beta}}{2}\sqrt{u^{*2}+w^{*2}}$
L_p	$\Gamma_1 q^* + \frac{\rho V_a^* Sb^2}{4}C_{p_p}$
L_r	$-\Gamma_2 q^* + \frac{\rho V_a^* Sb^2}{4}C_{p_r}$
L_{δ_a}	$\frac{\rho V_a^{*2} Sb}{2}C_{p_{\delta_a}}$
L_{δ_r}	$\frac{\rho V_a^{*2} Sb}{2}C_{p_{\delta_r}}$
N_v	$\frac{\rho S b^2 v^*}{4V_a^*}\left[C_{r_p}p^* + C_{r_r}r^*\right] + \rho S b v^*\left[C_{r_0} + C_{r_\beta}\beta^* + C_{r_{\delta_a}}\delta_a^* + C_{r_{\delta_r}}\delta_r^*\right] + \frac{\rho S b C_{r_\beta}}{2}\sqrt{u^{*2}+w^{*2}}$
N_p	$\Gamma_7 q^* + \frac{\rho V_a^* Sb^2}{4}C_{r_p}$
N_r	$-\Gamma_1 q^* + \frac{\rho V_a^* Sb^2}{4}C_{r_r}$
N_{δ_a}	$\frac{\rho V_a^{*2} Sb}{2}C_{r_{\delta_a}}$
N_{δ_r}	$\frac{\rho V_a^{*2} Sb}{2}C_{r_{\delta_r}}$

Table 3.3 : Longitudinal State Space Model Cofficients.

Longitudinal	Formula
X_u	$\frac{u^*\rho S}{\mathsf{m}}\left[C_{X_0} + C_{X_\alpha}\alpha^* + C_{X_{\delta_e}}\delta_e^*\right] - \frac{\rho S w^* C_{X_\alpha}}{2\mathsf{m}} + \frac{\rho S c C_{X_q} u^* q^*}{4mV_a^*} - \frac{\rho S_{\text{prop}} C_{\text{prop}} u^*}{\mathsf{m}}$
X_w	$-q^* + \frac{w^*\rho S}{\mathsf{m}}\left[C_{X_0} + C_{X_\alpha}\alpha^* + C_{X_{\delta_e}}\delta_e^*\right] + \frac{\rho S c C_{X_q} w^* q^*}{4mV_a^*} + \frac{\rho S C_{X_\alpha} u^*}{2\mathsf{m}} - \frac{\rho S_{\text{prop}} C_{\text{prop}} w^*}{\mathsf{m}}$
X_q	$-w^* + \frac{\rho V_a^* S C_{X_q} c}{4\mathsf{m}}$
X_{δ_e}	$\frac{\rho V_a^{*2} S C_{X_{\delta_e}}}{2\mathsf{m}}$
X_{δ_t}	$\frac{\rho S_{prop} C_{prop} k^2 \delta_t^*}{\mathsf{m}}$
Z_u	$q^* + \frac{u^*\rho S}{\mathsf{m}}\left[C_{Z_0} + C_{Z_\alpha}\alpha^* + C_{Z_{\delta_e}}\delta_e^*\right] - \frac{\rho S C_{Z_\alpha} w^*}{2\mathsf{m}} + \frac{u^*\rho S C_{Z_q} c q^*}{4mV_a^*}$
Z_w	$\frac{w^*\rho S}{\mathsf{m}}\left[C_{Z_0} + C_{Z_\alpha}\alpha^* + C_{Z_{\delta_e}}\delta_e^*\right] + \frac{\rho S C_{Z_\alpha} u^*}{2\mathsf{m}} + \frac{\rho w^* S c C_{Z_q} q^*}{4mV_a^*}$
Z_q	$u^* + \frac{\rho V_a^* S C_{Z_q} c}{4\mathsf{m}}$
Z_{δ_e}	$\frac{\rho V_a^{*2} S C_{Z_{\delta_e}}}{2\mathsf{m}}$
M_u	$\frac{u^*\rho S c}{J_y}\left[C_{m_0} + C_{m_\alpha}\alpha^* + C_{m_{\delta_e}}\delta_e^*\right] - \frac{\rho S c C_{m_\alpha} w^*}{2J_y} + \frac{\rho S c^2 C_{m_q} q^* u^*}{4J_y V_a^*}$
M_w	$\frac{w^*\rho S c}{J_y}\left[C_{m_0} + C_{m_\alpha}\alpha^* + C_{m_{\delta_e}}\delta_e^*\right] + \frac{\rho S c C_{m_\alpha} u^*}{2J_y} + \frac{\rho S c^2 C_{m_q} q^* w^*}{4J_y V_a^*}$
M_q	$\frac{\rho V_a^* S c^2 C_{m_q}}{4J_y}$
M_{δ_e}	$\frac{\rho V_a^{*2} S c C_{m_{\delta_e}}}{2J_y}$

3.6 Aerosim

The AeroSim aeronautical simulation blockset provides a complete set of tools for the rapid development of nonlinear 6-degree of freedom aircraft dynamic models. In addition to the basic aircraft dynamics blocks, the library also includes complete aircraft models which can be customized through parameter files. Since the AeroSim blockset components are built using only basic Simulink blocks and portable C/C++ code, you can use Real-TimeWorkshop to automatically generate source code from

the aircraft models. Aircraft model examples include the Aerosonde UAV and the Navion general-aviation airplane. The library allows importation of XML FlightGear aircraft configuration files (JSBSim format).

3.6.1 Library Description

The AeroSim library folders, presented in Figure 3.4, provide more than one-hundred blocks commonly used in the development of aircraft dynamic models. These include nonlinear equations of motion, linear aerodynamics, piston-engine propulsion, aircraft inertia parameters, atmosphere models, Earth models, sensors and actuators, frame transformations, and pilot interfaces such as joystick input and 3-D visual output. The library also provides complete aircraft models built using AeroSim blocks, and which can be customized to particular aircraft by editing an aircraft parameter file. In addition to the block library, a set of Simulink demos can be found in the samples directory of the AeroSim blockset. These provide dynamic models of actual aircraft such as the Aerosonde Unmanned Air Vehicle and the Navion general-aviation airplane. Aircraft model trim and linearization functions as well as trim examples for Aerosonde and Navion can be found in the trim directory. The fgutil directory contains XML parser scripts that can load JSBSim aircraft configuration files in Matlab aircraft structures. The util directory contains several useful engineering math

functions including Euler and Quaternion attitude representations, and eigenvalue analysis. The src directory provides the C/C++ source code for the Sfunctions that are implemented as C-MEX files. This allows you to use Real-Time Workshop with any aircraft models developed with AeroSim blocks and be able to build the resulting source code for other hardware platforms.

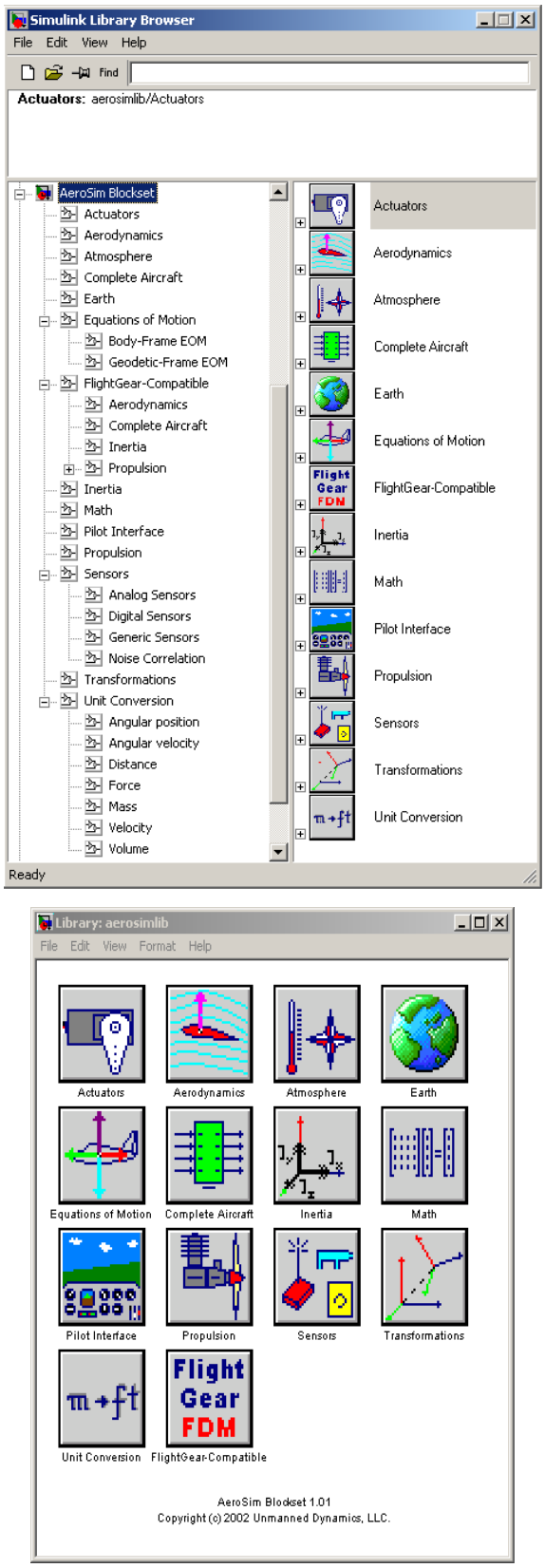


Figure 3.5 : Simulink library browser

4. MODEL PREDICTIVE CONTROL

As it is explained in introduction before. MPC has very important role in control. As a summary MPC :

- Predict the future behavior of the process state/output over the finite time horizon.
- Compute the future input signals on line at each step by minimizing a cost function under inequality constraints on the manipulated (control) and/or controlled variables.
- Apply on the controlled plant only the first of vector control variable and repeat the previous step with new measured input/state/output variables.

The success of MPC depends on the degree of precision of the plant model. In practice, modelling real plants inherently includes uncertainties that have to be considered in control design, that is control design procedure has to guarantee robustness properties such as stability and performance of closed-loop system in the whole uncertainty domain. Two typical description of uncertainty, state space polytope and bounded unstructured uncertainty are extensively considered in the field of robust model predictive control. Most of the existing techniques for robust MPC assume measurable state, and apply plant state feedback or when the state estimator is utilized, output feedback is applied.

The reason of choosing MPC as controller :

- It handles multivariable control problems naturally
- It can take account of actuator limitations
- It allows operation closer to constraints, hence increased profit
- It has plenty of time for on-line computations
- It can handle non-minimal phase and unstable processes
- It is an easy to tune method
- It handles structural changes.

A basic structure of MPC is shown in Figure 4.1.

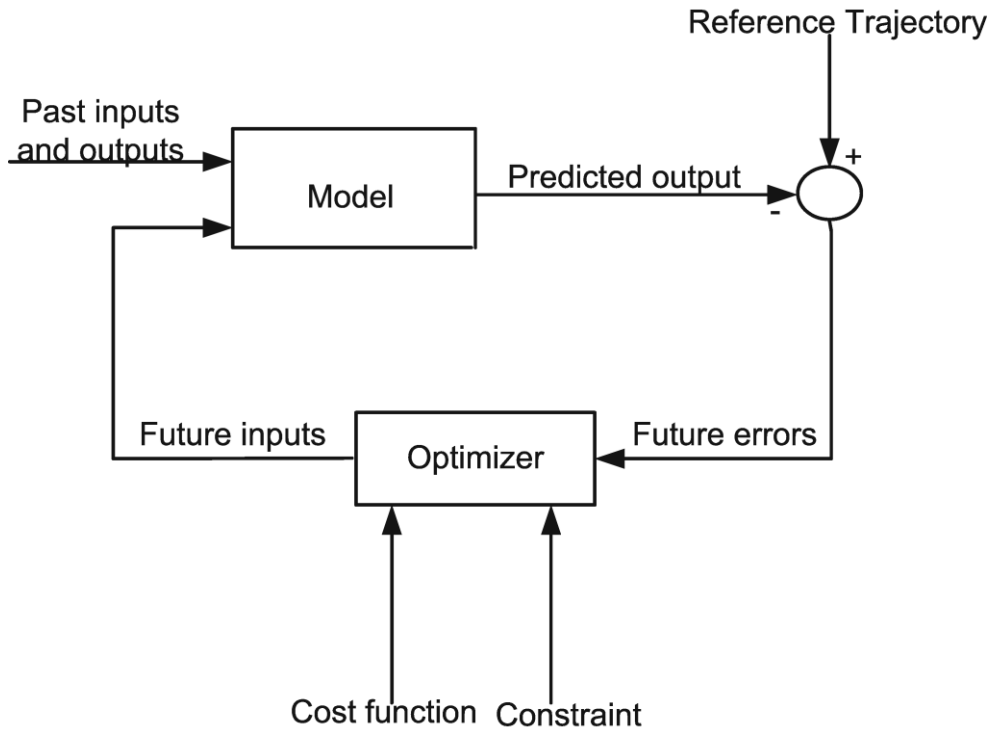


Figure 4.1 : Basic structure of MPC.

4.1 State Space Model for MPC

This part represents state-space models and typical modelling assumptions used in MPC.

$$\boldsymbol{x}(t+1) = A(\alpha)\boldsymbol{x}(t) + B(\alpha)u(t) \quad \textbf{(4.1)}$$

$$y(t) = Cx(t) \quad \textbf{(4.2)}$$

Where $x(t) \in R^n, u(t) \in R^m, y(t) \in R^l$ are state, control and output variables of the system, respectively; A(α), B(α) belong to the convex set

$$S = \{A(\alpha) \in R^{nxn}, B(\alpha) \in R^{nxm}\} \quad \textbf{(4.3)}$$

$$\{A(\alpha) = \sum_{j=1}^{N} A_j\alpha_j \; B(\alpha) = \sum_{j=1}^{N} B_j\alpha_j, \; \alpha_j \geq 0\}, j = 1,2 \dots N, \sum_{j=1}^{N} \alpha_j = 1 \quad \textbf{(4.4)}$$

Matrices A_{i} , B_{i} and C are known matrices with constant entries of corresponding dimensions. Simultaneously with (4.1) we consider the nominal model of system (4.1) in the form,

$$x(t+1) = A_o x(t) + B_o u(t) \qquad y(t) = Cx(t) \quad \textbf{(4.4)}$$

where A_0, B_o are any constant matrices from the convex bounded domain S (4.4). The nominal model (4.5) will be used for prediction, while (4.1) is considered as real plant description providing plant output. Therefore in the robust controller design we assume that for time t output y(t) is obtained from uncertain model (4.1), predicted outputs for time t + 1, ...t + N_2 will be obtained from model prediction, where the nominal model (4.5) is used. The predicted states and outputs of the system (4.1) for the instant t + k, k = 1, 2, ... N_2 are given by

• k=1

$$x(t+2) = A_0 x(t+1) + B_o u(t+1)$$
$$= A_0 A(\alpha) x(t) + A_0 B(\alpha) u(t) + B_o u(t+1) \quad \textbf{(4.5)}$$

• k=2

$$x(t+3) = A_0^2 A(\alpha) x(t) + A_0^2 B(\alpha) u(t) + A_0 B_o u(t+1) + B_o u(t+2) \quad \textbf{(4.6)}$$

• for k

$$x(t+k+1)$$
$$= A_0^k A(\alpha) x(t) + A_0^k B(\alpha) u(t) + \sum_{i=0}^{k-1} A_0^{k-i-1} B_o u(t+1+i) \quad \textbf{(4.7)}$$

and corresponding output is

$$y(t+k) = Cx(t+k) \quad \textbf{(4.8)}$$

Consider a set of k = 0, 1, 2, ..., N_2 state/output model predictions as follows

$$z(t+1) = A_f(\alpha) z(t) + B_f(\alpha) v(t), \quad y_f(t) = C_f z(t) \quad \textbf{(4.9)}$$

where

$$z(t)^T = [x(t)^T \dots z(t+N_2)^T], \quad v(t)^T = [u(t)^T \dots u(t+N_u)^T] \quad \textbf{(4.10)}$$

$$y_f(t)^T = [y(t)^T .. y(t+N_2)^T] \quad \textbf{(4.11)}$$

and

$$B_f(\alpha) = \begin{bmatrix} B(\alpha) & 0 & \dots & 0 \\ A_0 B(\alpha) & B_0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ A_0^{N_2} B(\alpha) & A_0^{N_2} B_o & \dots & A_0^{N_2 - N_u} B_o \end{bmatrix} \quad \textbf{(4.12)}$$

$$A_f(\alpha) = \begin{bmatrix} A(\alpha) & 0 & \dots & 0 \\ A_0 A(\alpha) & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ A_0^{N_2} A(\alpha) & 0 & \dots & 0 \end{bmatrix} \quad \textbf{(4.13)}$$

$$C_f = \begin{bmatrix} C & 0 & \dots & 0 \\ 0 & C & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & C \end{bmatrix} \quad \textbf{(4.14)}$$

where N_2, N_u are output and control prediction horizons of model predictive control, respectively. Note that for output/state prediction in (4.6) one needs to put A(α) = A_0, B(α) =B_0. Matrices dimensions are $A_f(\alpha) \in R^{n(N_2+1)xn(N_2+1)}$, $B_f(\alpha) \in R^{n(N_2+1)xm(N_u+1)}$ and $C_f \in R^{l(N_2+1)xn(N_2+1)}$.

4.2 Performance Index (Cost Function)

The control law is determined from the minimisation of a 2-norm measure of predicted performance. A typical performance index (or cost function) is

$$J = \sum_{i=n_w}^{n_y} \|\boldsymbol{r}_{k+i} - \boldsymbol{y}_{k+i}\|_2^2 + \lambda \sum_{i=0}^{n_u-1} \|\Delta \boldsymbol{u}_{k+i}\|_2^2 \quad \textbf{(4.15)}$$

$$= \sum_{i=n_w}^{n_y} \|\boldsymbol{e}_{k+i}\|_2^2 + \lambda \sum_{i=0}^{n_u-1} \|\Delta \boldsymbol{u}_{k+i}\|_2^2 \quad \textbf{(4.16)}$$

Sum the squares of the predicted tracking errors from an initial horizon n_w to an output horizon n_y. Sum the squares of the control changes over the horizon n_u. It is assumed that control increments are zero beyond the control horizon, that is

$$\Delta \boldsymbol{u}_{k+i|k} = 0, i \geq n_u \quad \textbf{(4.17)}$$

The degrees of freedom (d.o.f.) which can be used to minimise *J* are the future control moves, typically the first n_u control moves. Hence the on-line control law is determined from the minimisation of the cost *J* with respect to the n_u future control moves, that is $\Delta\mathbf{u}$. This minimisation is denoted as

$$\min_{\Delta \mathbf{u}} J = \|\boldsymbol{e}\|_2^2 + \lambda \|\Delta \boldsymbol{u}\|_2^2 \quad \textbf{(4.18)}$$

Of the optimising $\Delta\mathbf{u}$, only the first element, that is $\Delta\mathbf{u}k$, is implemented as the optimisation is repeated (updated) at each sampling instant.

4.2.1 Constraints

One of the most important ability of MPC is, takes account of constraints. These constraints can be easily be incorporated into the optimisation. The general structure of constraints are like ;

$$\underline{u_i} \leq u_i \leq \overline{u_i} \quad \textbf{(4.19)}$$

The constraints on input rates,

$$\underline{\Delta u_i} \leq \Delta u_i \leq \overline{\Delta u_i} \quad \textbf{(4.20)}$$

The constraints on outputs and states,

$$\underline{x_i} \leq x_i \leq \overline{x_i}; \quad \underline{y_i} \leq y_i \leq \overline{y_i} \tag{4.21}$$

The cost function becomes,

$$\min_{\Delta \mathbf{u}} J = \Delta \boldsymbol{u}^T S \Delta \boldsymbol{u} + 2\Delta \boldsymbol{u}^T \boldsymbol{f} \;\; s.t. \quad C\Delta \boldsymbol{u} - \boldsymbol{d}_k \leq 0 \tag{4.22}$$

$$\text{where } \; S = H^T H + \lambda I, \quad \boldsymbol{f} = [P\Delta \boldsymbol{u} + Q\boldsymbol{y} - \boldsymbol{r}] \tag{4.23}$$

4.2.2 Weighting Matrices

The weighting λ is scalar. For the multivariable systems, the weighting matrices are used. Cost function for the multivariable systems becomes,

$$J = \sum_{i=n_w}^{n_y} \left\| W_y(\boldsymbol{e}_{k+i}) \right\|_2^2 + \lambda \sum_{i=0}^{n_u-1} \| W_u(\Delta \boldsymbol{u}_{k+i}) \|_2^2 \tag{4.24}$$

where matrix weights *Wy, Wu* are positive definite and diagonal. These weights can even vary (usually increase) with the horizon *i* though such a complication would rarely be justified, as the increase in the number of design parameters to select often gives negligible benefits. Often the sensitivity of the resulting controller parameters to the weights can be quite small so one may need to change the weights by an order of magnitude to see a significant effect.

5. SIMULATIONS AND RESULTS

Matlab-Simulink is used to simulate the control theory on the model. A special blockset named Aerosim for Simulink is chosen its advantages as expressed before. Aerosonde is a small maritime UAV for intelligence, surveillance, and reconaissance firstly introduced in 2001. This aircraft is suitable to work on it.

5.1 Model Selection

Aerosim block set includes that autonomous unmanned aircraft model is six DOF and named Aerosonde.

This model has 7 inputs and 7 inputs. But as you can see it in Figure 5.1, three of inputs are fixed. System has 14 states;

$$\boldsymbol{x} = (u\ v\ w\ p\ q\ r\ \phi\ \theta\ \psi\ p_N\ \ p_E\ p_D\ F\ \omega)^T \qquad \textbf{(5.1)}$$

u, v, w are the velocities in x-y-z axes.

$\phi\ \theta\ \psi$ are Euler angles.

p, q, r are the angular rates.

$p_N\ \ p_E\ p_D$ are the positions with respect to NED frame.

$F\ \omega$ are Fuel and engine angular velocity. These two states are additional states for usual description.

The inputs are throttle, elevator, rudder, aileron, flaps, mixture and ignition. Mixture and ignition are fixed and there is no need to use them for controller.

The structure of the Aerosonde model is in Figure 5.2.

5.2 Trim

Trim is the definition of the aircraft flying at steady state. This means that all the aircraft velocity variables are fixed at a constant value. The aircraft is 'tuned' in a way that it conducts a steady flight and no torque is applied to the airplane's center of gravity. This implies that the aircraft acceleration components are all zero. This definition leads to:

$$\dot{u}, \dot{v}, \dot{w}, \dot{p}, \dot{q}, \dot{r} \equiv 0, \quad u > 0 \tag{5.2}$$

Aerosim has a trim program for Aerosonde. The program uses the Simulink model is showed in Figure 5.3. This trim program has structure as below.

The trim and linearization script will create the trim parameters structure with values provided by the user and with default values for several parameters.

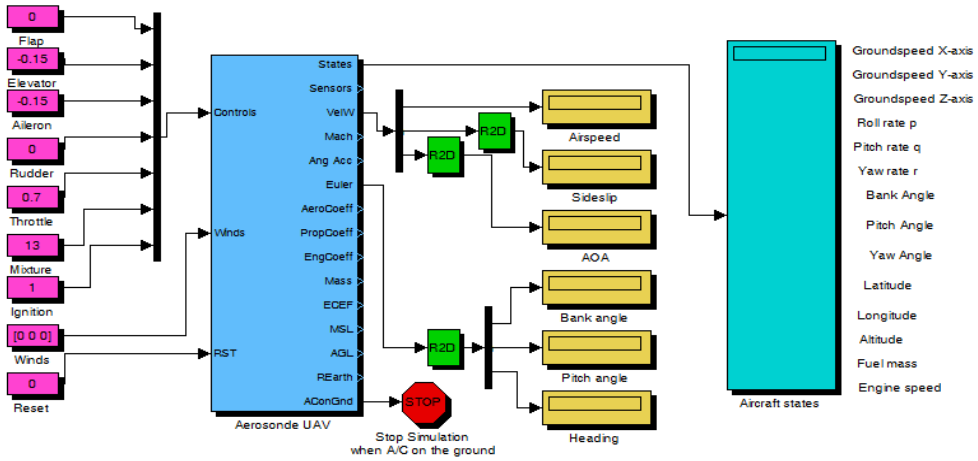


Figure 5.1 : Open loop model of Aerosonde.

Flight conditions are chosen as;

Airpseed :23 m/s

Altitude : 1000 m

Bank Angle : 0 rad

Fuel : 2 kg

Flap : 0 (Because Aerosonde doesn’t have any flaps)

The results of the trim is ;

Inputs :

Elevator = -0.1429

Aileron = -0.0086

Rudder = -0.0010

Throttle = 0.5697

States:

u = 22.93 m/s

v = 0.01 m/s

w = 1.73 m/s

p = 0.00 rad/s

q = -0.00 rad/s

r = 0.00 rad/s

ϕ = -0.00 rad

θ = 0.08 rad

ψ = 0.00 rad

p_N= 1.3664 rad

p_E = -2.0817 rad

p_D= 1000.00 m

F = 2.00 kg

ω = 4835 rot/min

Outputs:

Airspeed = 23.00 m/s

Sideslip = 0.00 rad

AOA = 0.08 rad

Bank = -0.00 rad

Pitch = 0.08 rad

Heading = 0.00 rad

Altitude = 1000.00 m

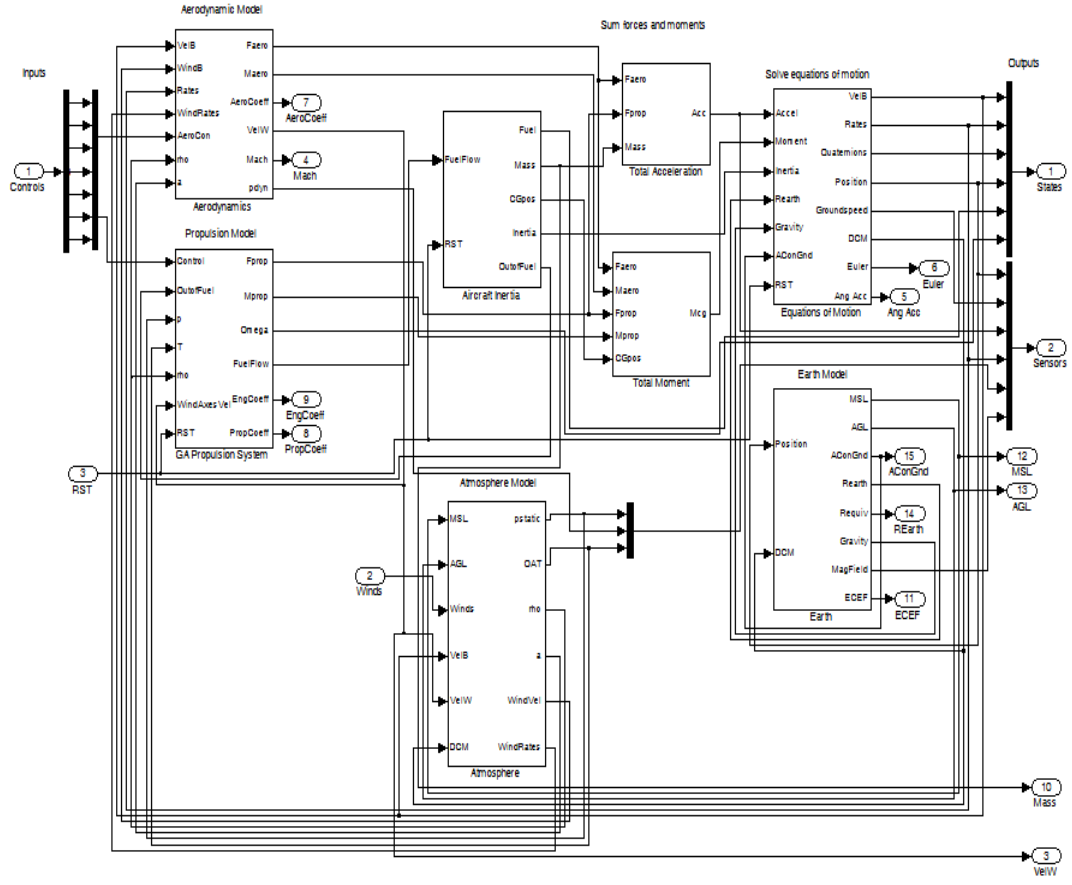


Figure 5.2 : Main structure of the model without mask.

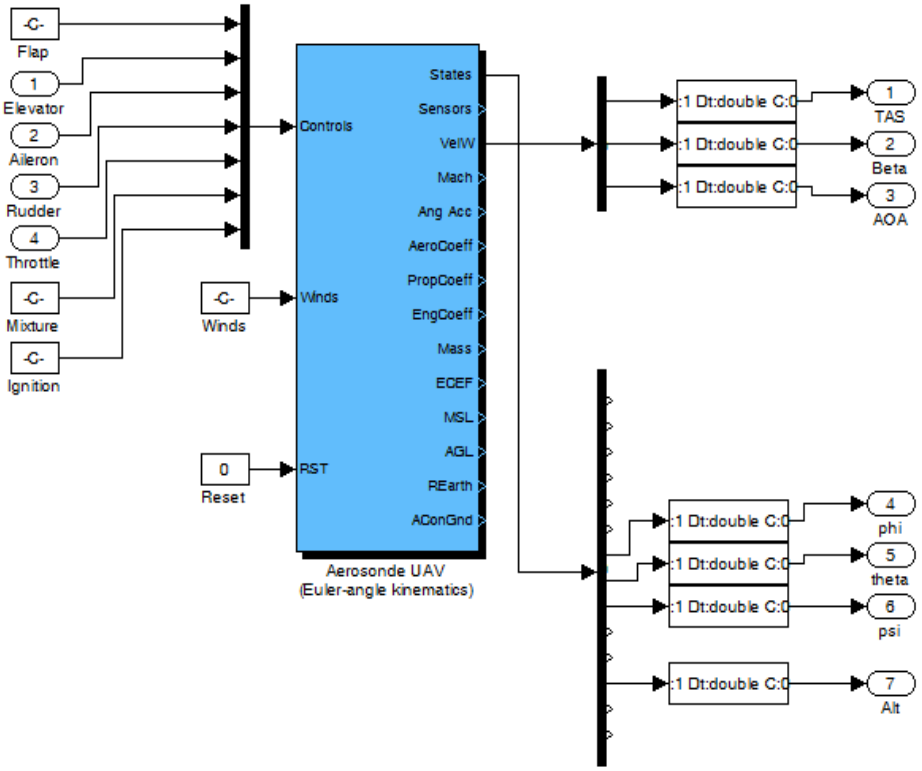


Figure 5.3 : Trim model used in trim program.

5.3 Closed Loop Model

The Closed loop model is used in simulation has just one MPC that control 7 output while manipulating 4 inputs. As it is seen in Figure 5.8. Simulink model has aircraft model at left, and at the bottom the MPC, at right scopes and FlightGear interface.

The control structure is different from the traditional control theories. In traditional control theories of an aircraft separates aircrafts dynamics to two parts. These are lateral dynamics and longitudinal dynamics.

The reason behind of separating them is nonlinearity of aircraft. In addition, these two dynamics can be accepted as decoupled.

In this thesis it's suggested to control all aircraft with one controller. As it was explained in earlier chapter, MPC has abilities to calculate the relation between all inputs and outputs and constraints on them.

The structure of the MPC is shown in Figure 5.4.

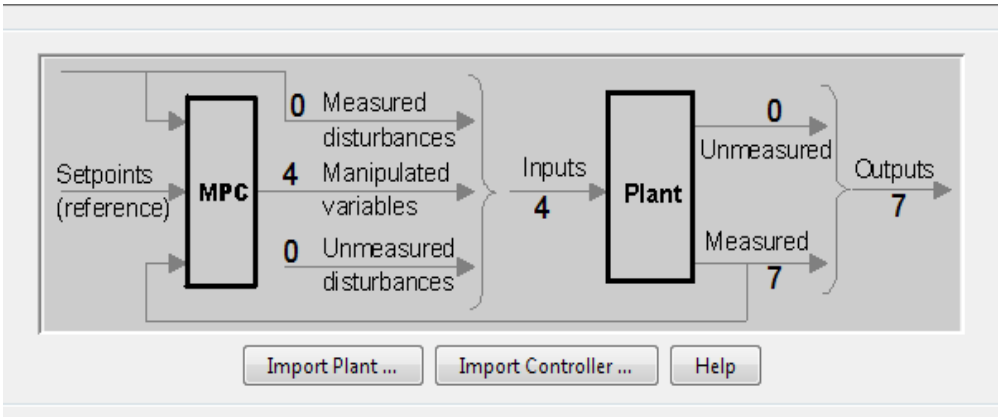


Figure 5.4 : MPC structure overview.

5.4 Tuning

The behavior of MPC can be change by changing parameters like weights, prediction horizon, control horizon etc. If the cost function is built right, stability and tuning will be succesful. When reasonable guidelines are used, MPC will provide stable control. The importance of tuning is to get stability between input and speed of response. By tunning the weighting matrices, it becomes easier to tune. Its hard to generalise it. Because every system has different properites.

Prediction horizon Tp and Control Horizon: For a stable system, the prediction horizon is can be chosen nearly equal to the open loop process settling time. There are two different settings for both parameters. Here is the control and prediction horizon values are used in simulations are shown in Figure 5.6-7.

Weighting matrix W: When there are constraints active on the control signal, these need to be scaled in such a was that the relative action of the controls is equally weighted.Weights are used in simulations are shown in Figure 5.8-9.

Constraints: One of the most important feature of the MPC is ability to handle constraints. Inputs,outputs and their rates have constraints in real. Our aerosim model also has all constraints about aircraft. As a reason of that we directly put the constraints to our cost function and do the simulations with constraints. The constraints used in simulations are shown in Figure 5.10-11.

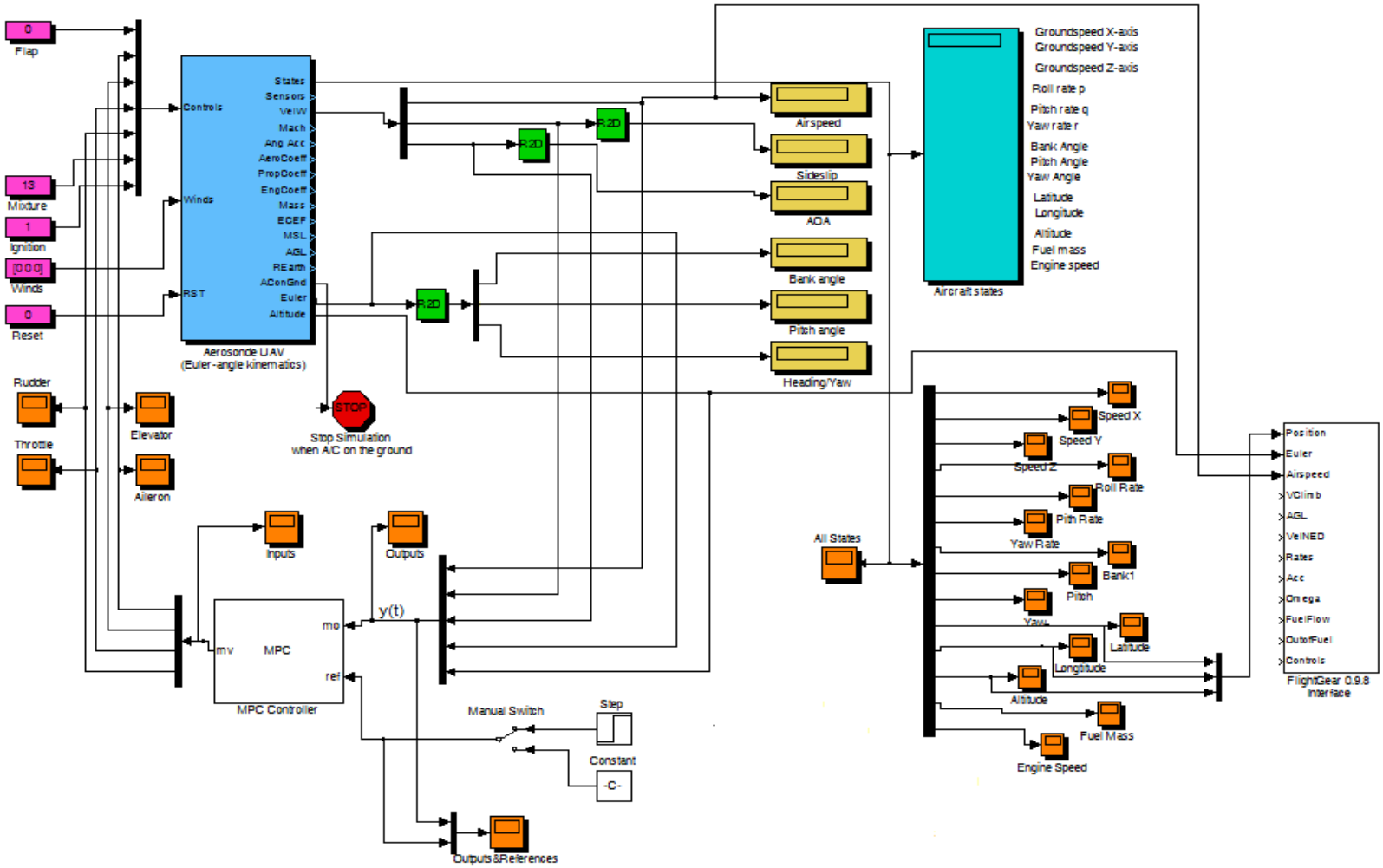


Figure 5.5 : Closed loop model of aircraft with MPC and FlightGear interface.

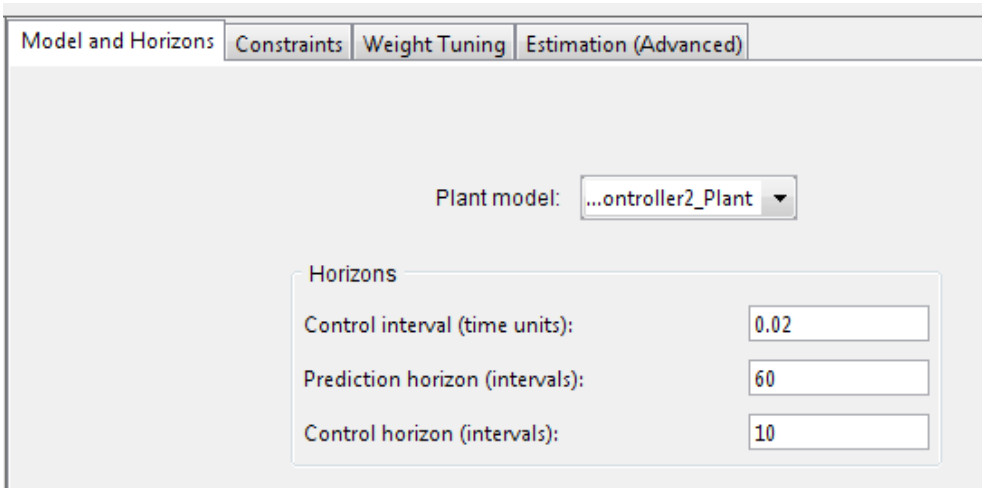


Figure 5.6 : Prediction and control horizons of first simulation.

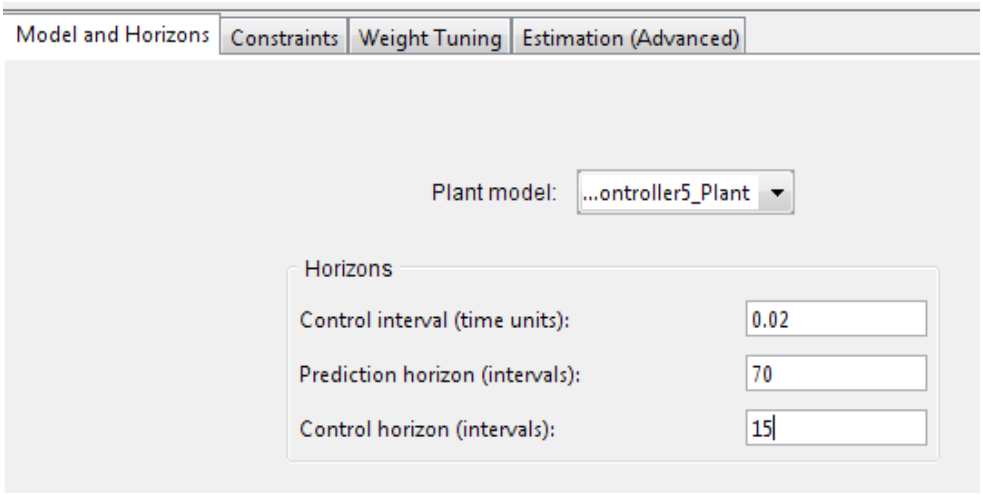


Figure 5.7: Prediction and control horizons of second simulation.

5.5 Results

The results of the simulations with the different tuning parameters shown above are like below. As it seen the first simulation results are not stable and as it expected. To obtain stable and desireable result so many different variations of paramaters have been tried. Both simulations times are 90 seconds and at 20^{th} second, there is a step at airspeed of aircraft. The initial value of airspeed is 23 m/s. The airspeed of aircraft becomes 24 m/s at 20^{th} second. The changes of input signals for first simulation are shown in Figure 5.12-15.

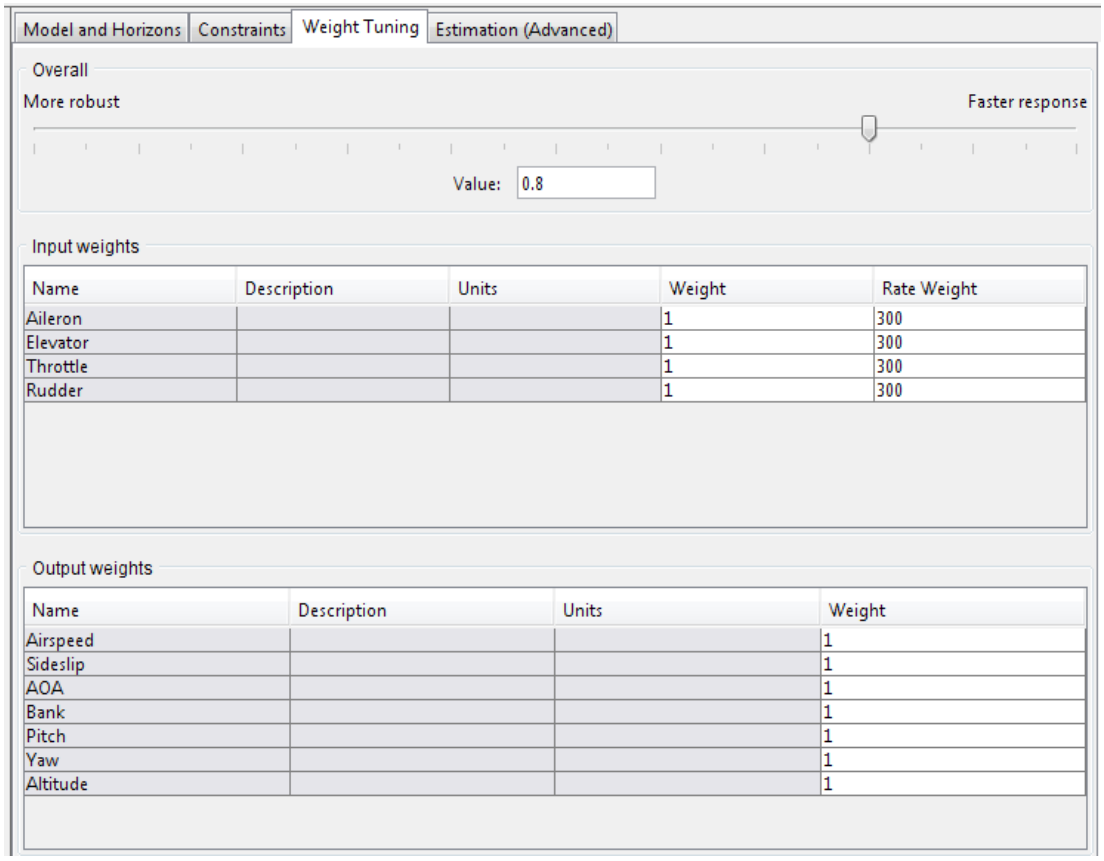


Figure 5.8: Weights on inputs, input rates and outputs for first simulation.

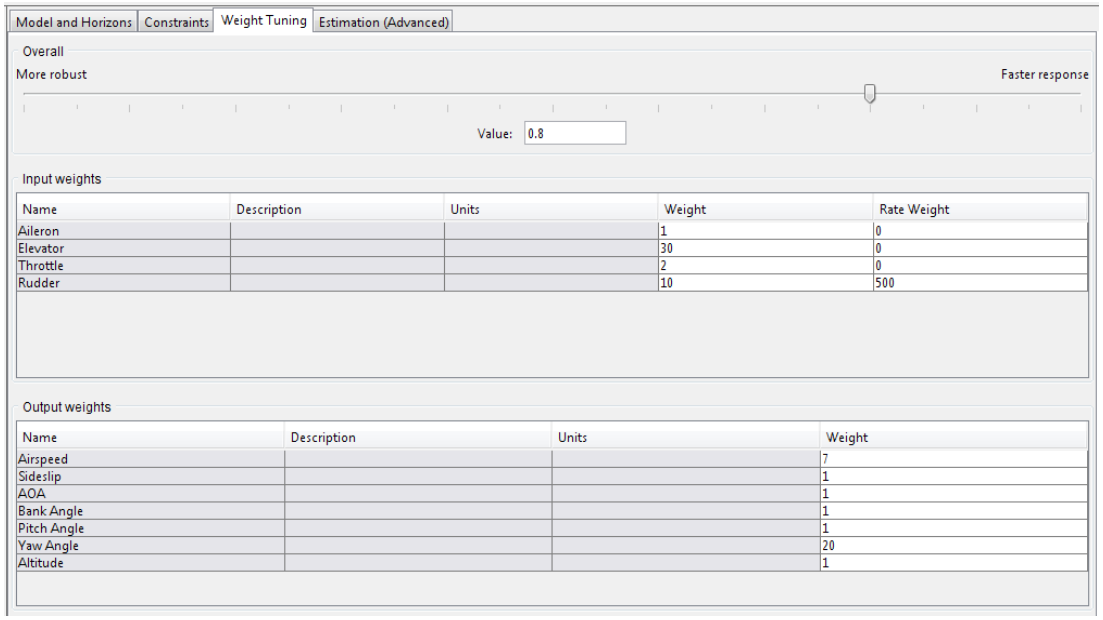
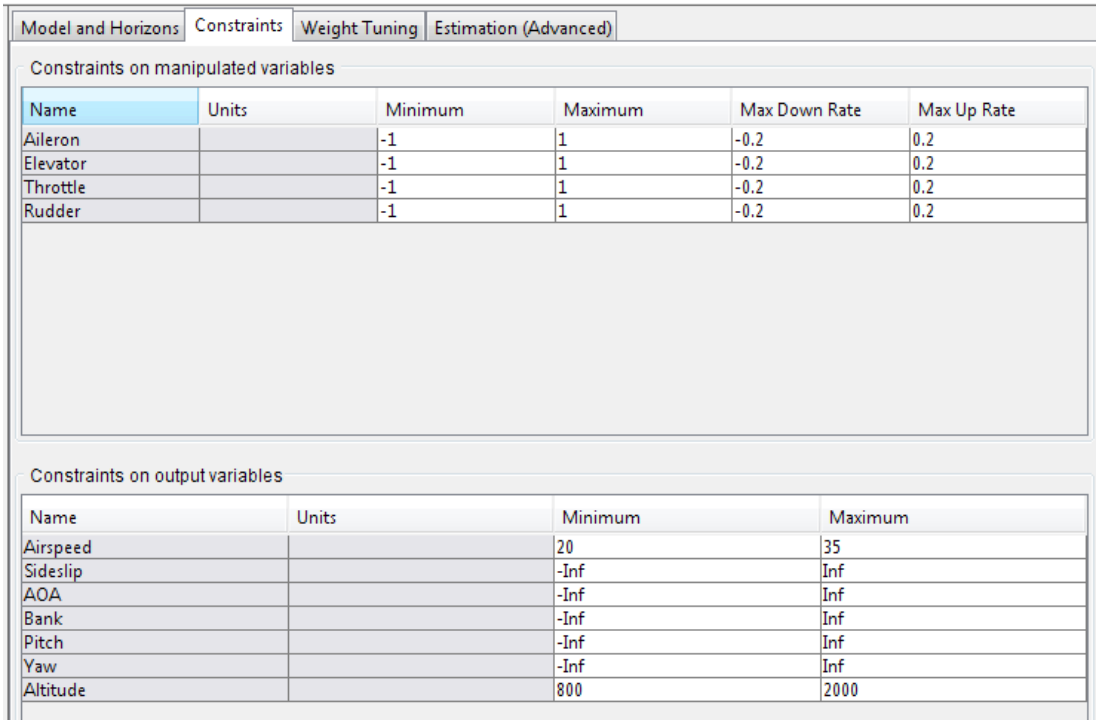


Figure 5.9: Weights on inputs, input rates and outputs for second simulation.



Model and Horizons | Constraints | Weight Tuning | Estimation (Advanced)

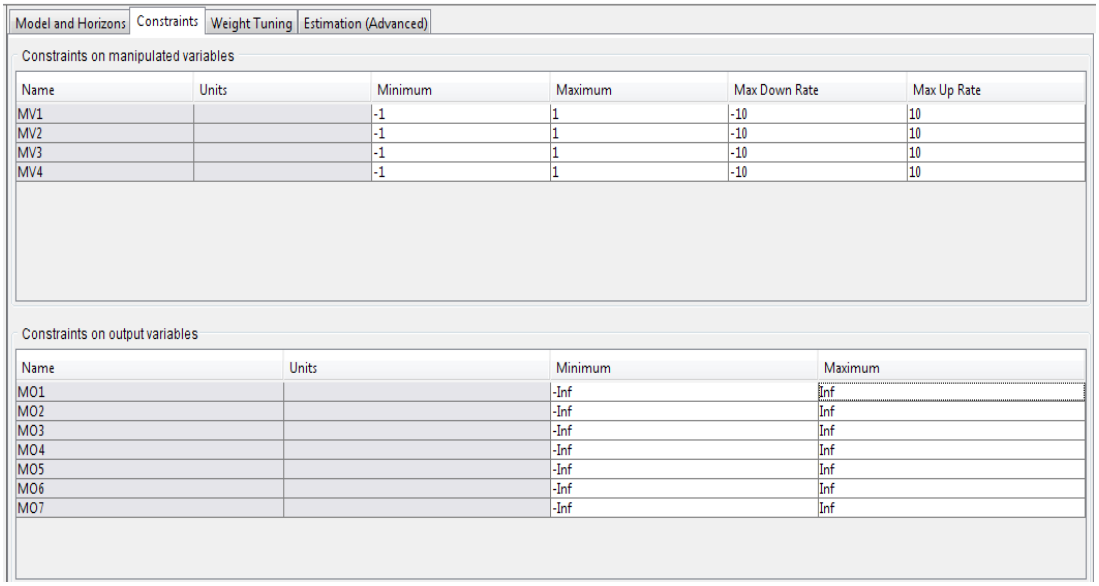
Constraints on manipulated variables

Name	Units	Minimum	Maximum	Max Down Rate	Max Up Rate
Aileron		-1	1	-0.2	0.2
Elevator		-1	1	-0.2	0.2
Throttle		-1	1	-0.2	0.2
Rudder		-1	1	-0.2	0.2

Constraints on output variables

Name	Units	Minimum	Maximum
Airspeed		20	35
Sideslip		-Inf	Inf
AOA		-Inf	Inf
Bank		-Inf	Inf
Pitch		-Inf	Inf
Yaw		-Inf	Inf
Altitude		800	2000

Figure 5.10: The constraints on inputs, input rates and outputs for first simulation.



Model and Horizons | Constraints | Weight Tuning | Estimation (Advanced)

Constraints on manipulated variables

Name	Units	Minimum	Maximum	Max Down Rate	Max Up Rate
MV1		-1	1	-10	10
MV2		-1	1	-10	10
MV3		-1	1	-10	10
MV4		-1	1	-10	10

Constraints on output variables

Name	Units	Minimum	Maximum
MO1		-Inf	Inf
MO2		-Inf	Inf
MO3		-Inf	Inf
MO4		-Inf	Inf
MO5		-Inf	Inf
MO6		-Inf	Inf
MO7		-Inf	Inf

Figure 5.11: The constraints on inputs, input rates and outputs for second simulation.

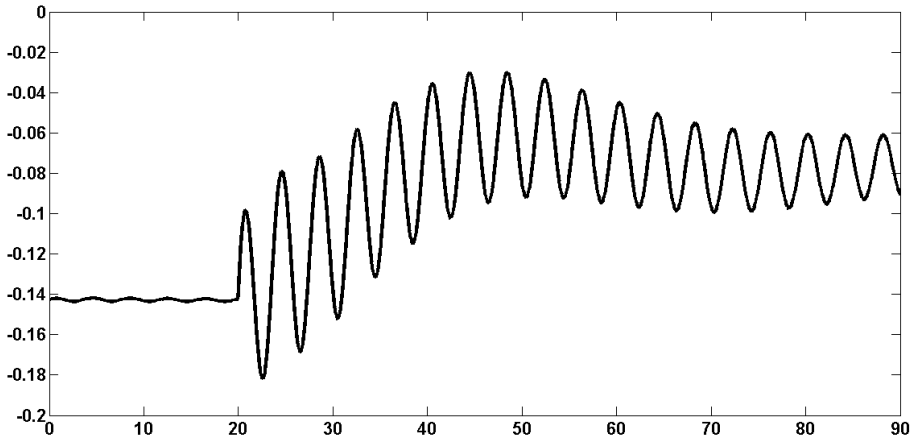


Figure 5.12 : Aileron Deflection of first simulation.

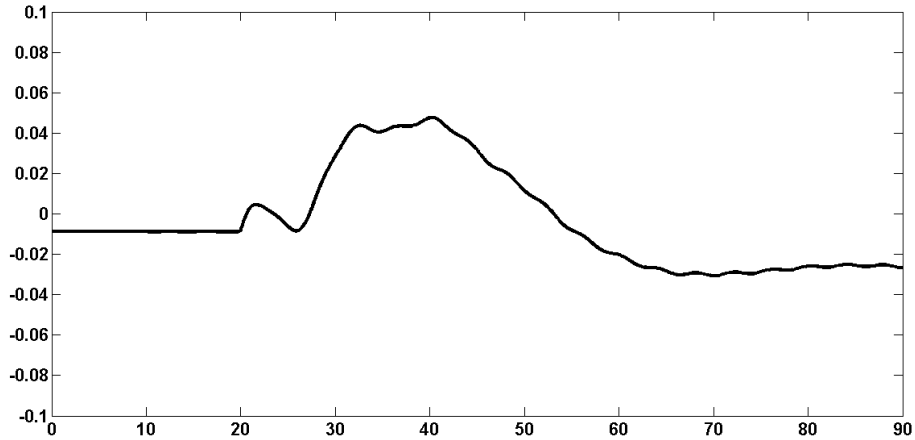


Figure 5.13 : Elevator deflection of first simulation.

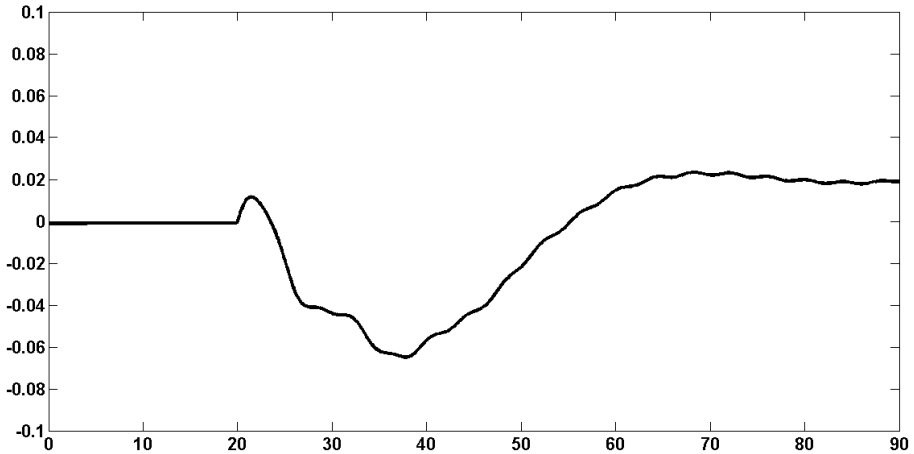


Figure 5.14 : Rudder deflection of first simulation.

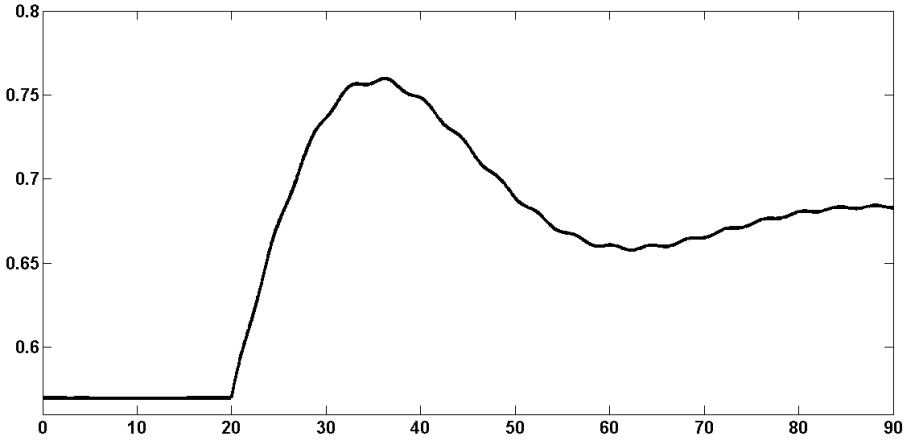


Figure 5.15 : Throttle deflection of first simulation.

The changes of input signals for second simulation are shown in Figure 5.16-19.

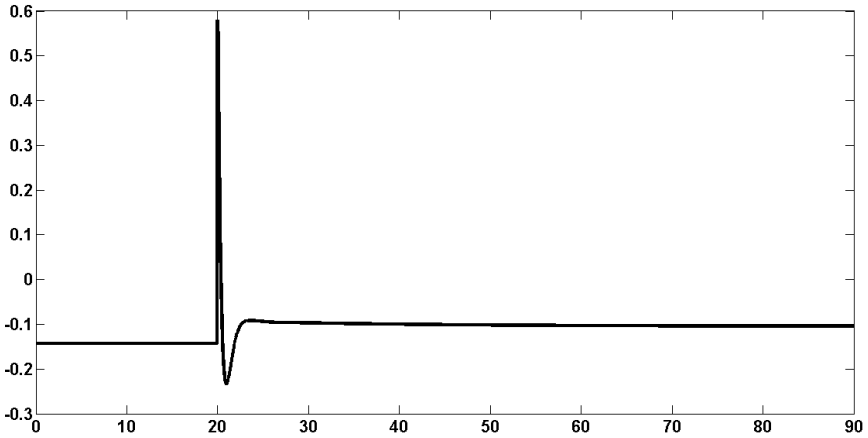


Figure 5.16 : Aileron Deflection of second simulation.

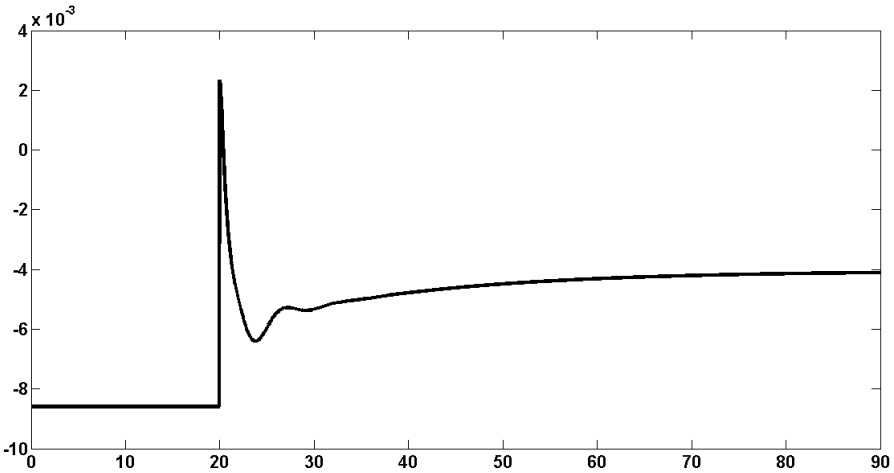


Figure 5.17 : Elevator deflection of second simulation.

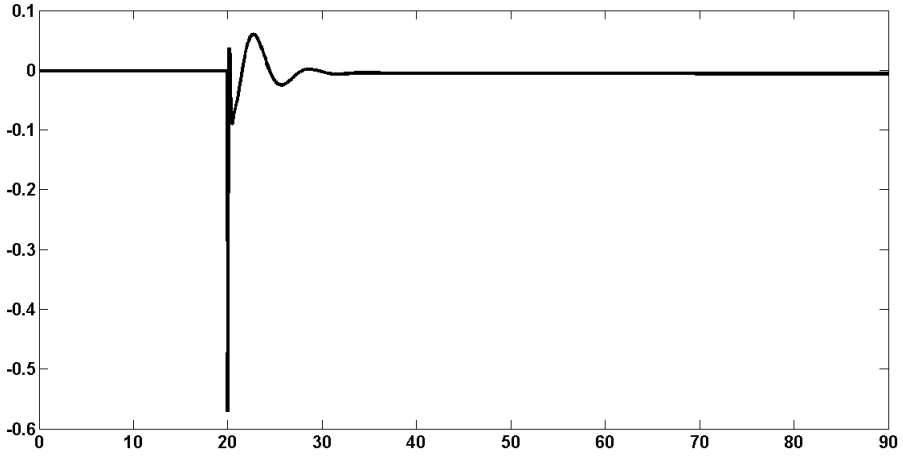


Figure 5.18 : Rudder deflection of second simulation.

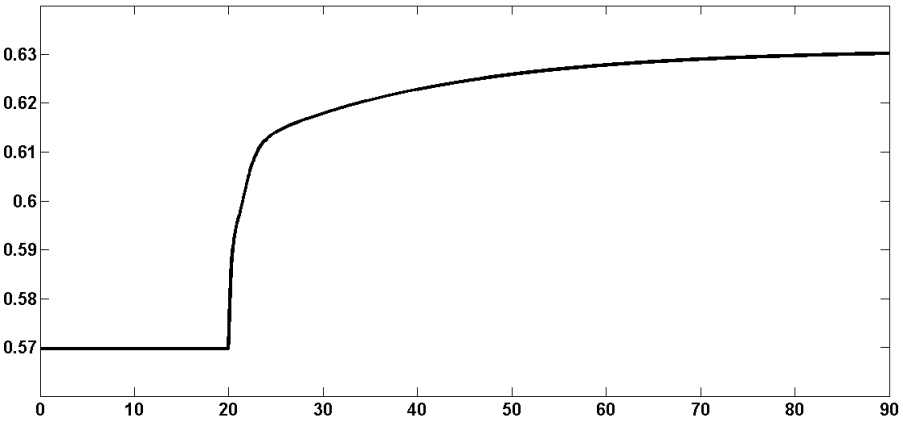


Figure 5.19 : Throttle deflection of second simulation.

The changes on outputs of first simulation are in Figure 5.20-5.26.

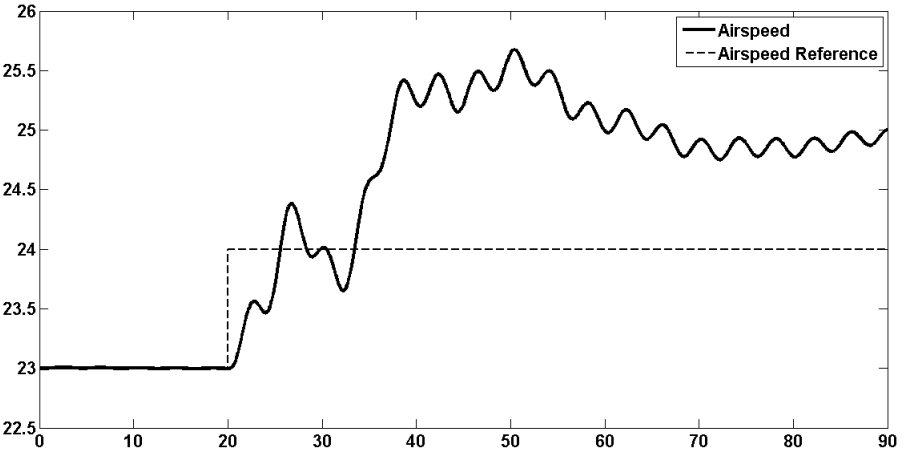


Figure 5.20 : Airspeed result for first simulation.

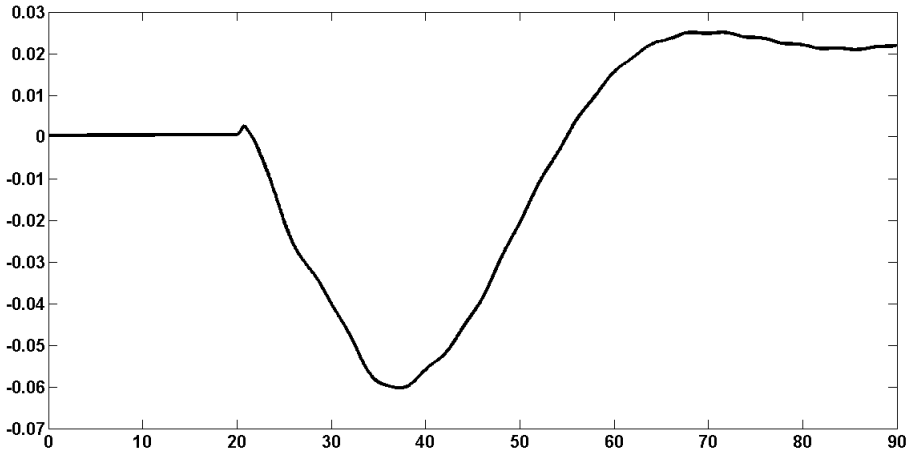


Figure 5.21 : Sideslip result for first simulation.

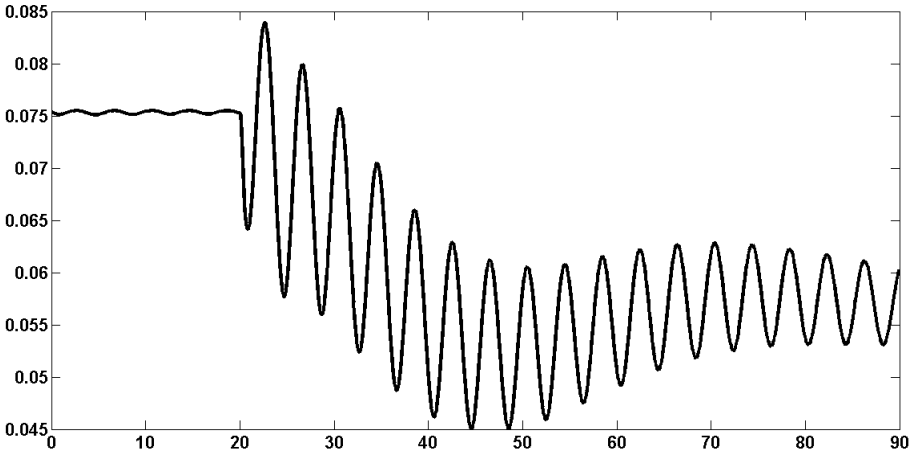


Figure 5.22 : Angle of attack result for first simulation.

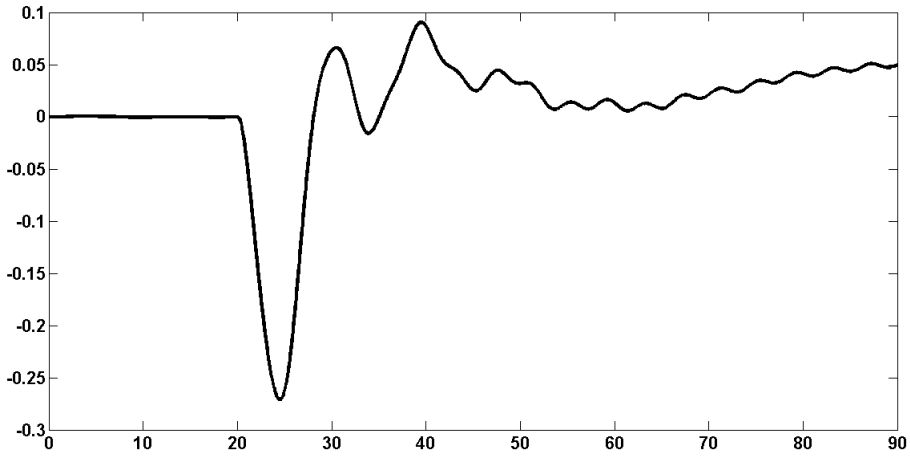


Figure 5.23 : Bank angle result for first simulation.

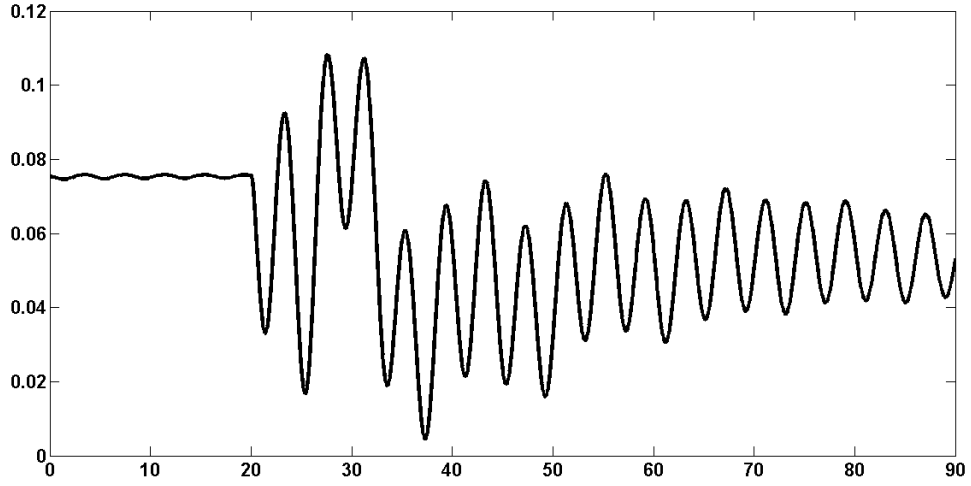


Figure 5.24 : Pitch angle result for first simulation.

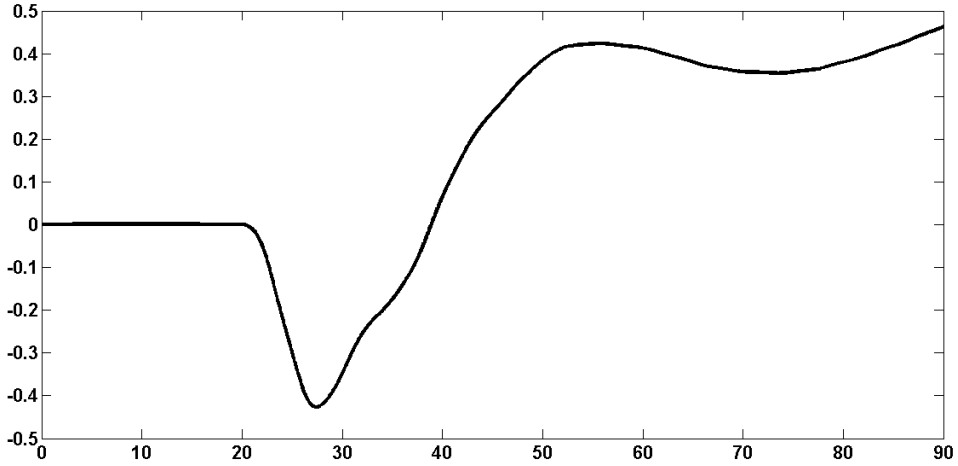


Figure 5.25 : Yaw angle result for first simulation.

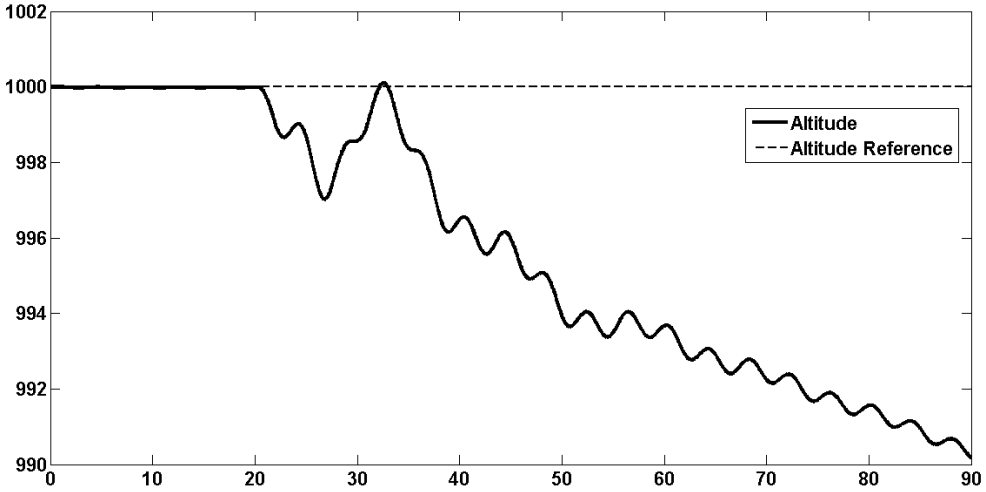


Figure 5.26 : Altitude result for first simulation.

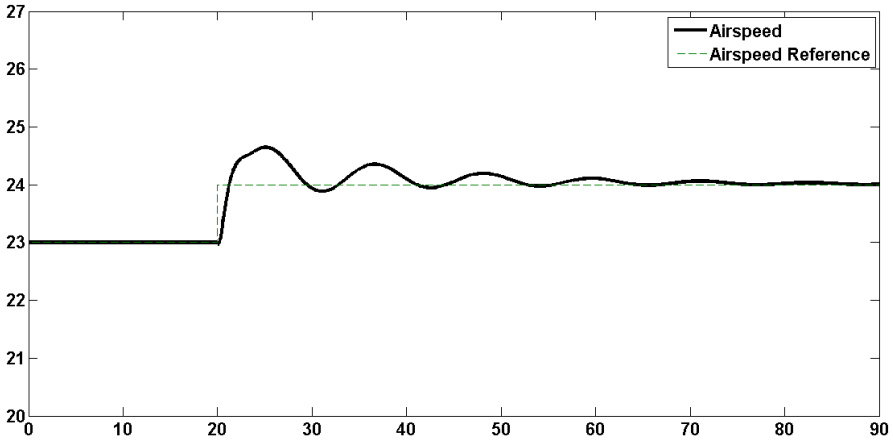


Figure 5.27 : Airspeed.

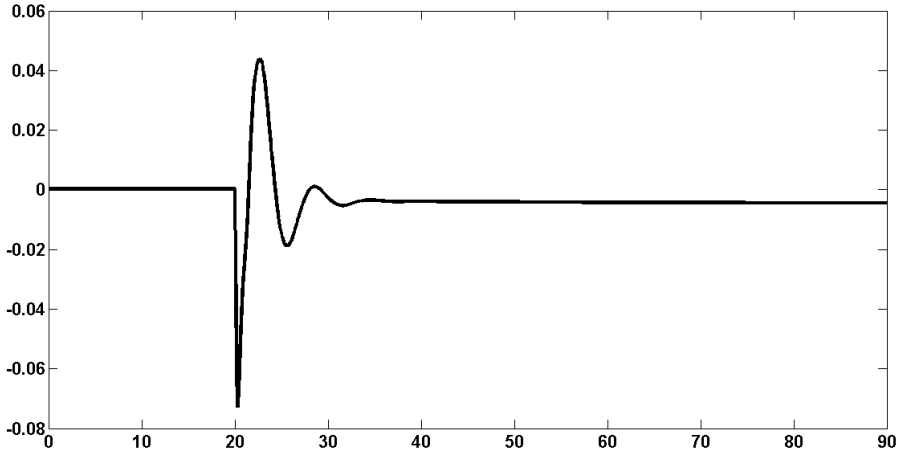


Figure 5.28 : Sideslip.

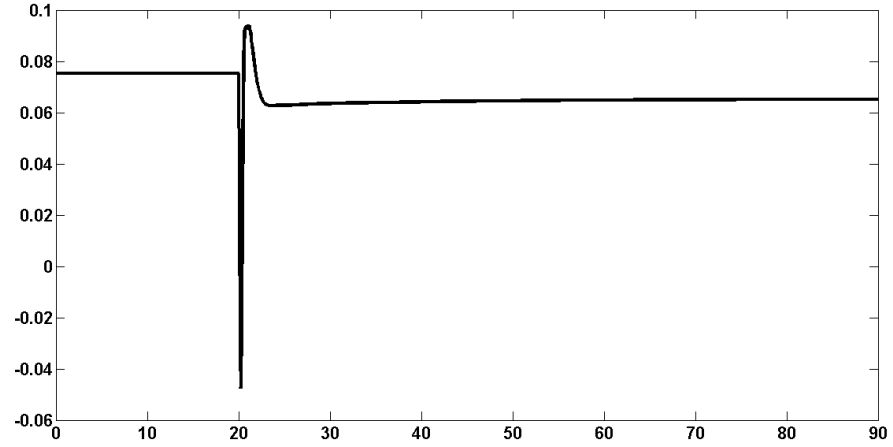


Figure 5.29 : Angle of attack.

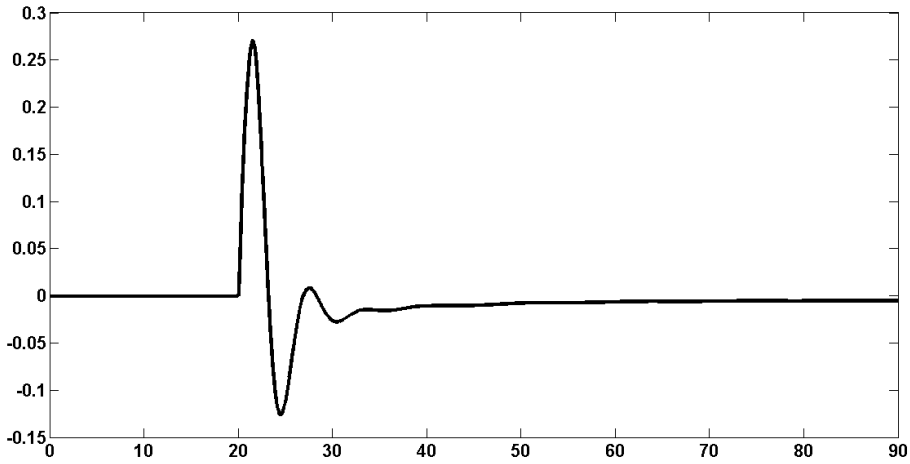


Figure 5.30 : Bank angle.

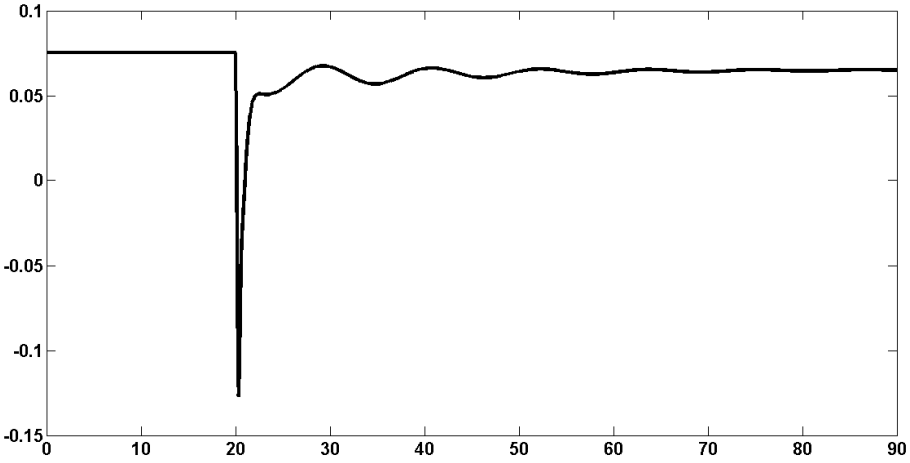


Figure 5.31 : Pitch angle.

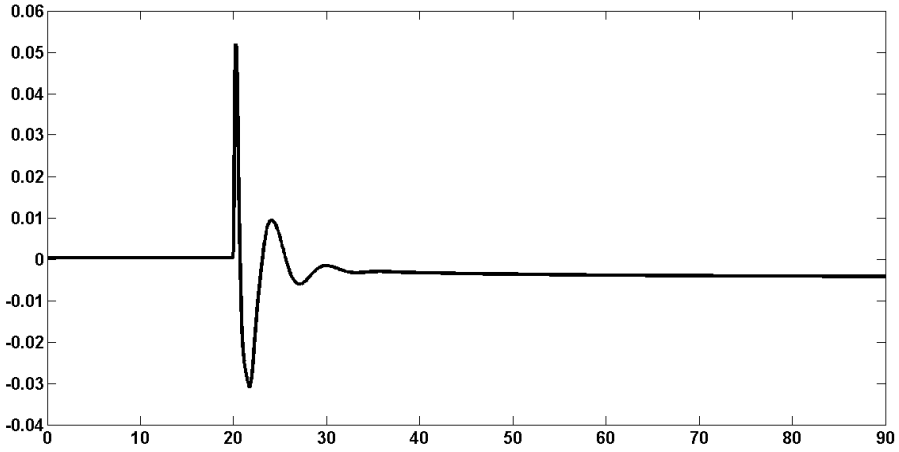


Figure 5.32 : Yaw angle.

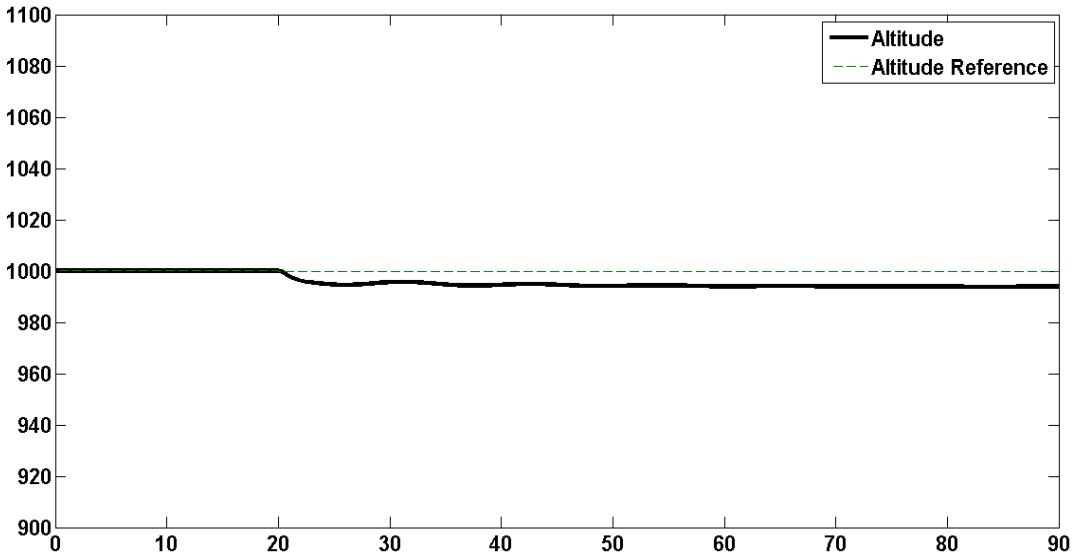


Figure 5.33 : Altitude.

5.6 FlightGear Results

The visual results from FlightGear interface are below in Figure 5.20 and 5.21.

Figure 5.34 : Aircraft flight from back.

Figure 5.35 : Cockpit and aircraft together.

6. CONCLUSIONS AND RECOMMENDATIONS

The UAV aerosonde has controlled with MPC. Model predictive controller handled with four inputs and seven outputs.

Before explain the performance of the controller and the criticals about the system that under control of MPC, it is better to explain the parameter changes between to simulations. In the first simulation the control and prediction horizon were not choosen big enough to obtain a desirable step response of the system. In the second simulation they have been choosen bigger. And also the constrains on the input rates were choosen much more bigger. That solution is also improved the response. The last changes were made on the weights. The weights were changed according to system characteristics and the response of the each output.

The first control problem MPC controller must be handle is bank angle. The reason of this can be explained like this: The plane Aerosonde has a single motor. Rotor of the motor causes a reverse moment on stator. This reverse moment makes plane rotate respect x-axes. That's mean is reverse moment changes equilibrium point of bank angle. Thus the plane needs bank control even its at equilibrium point. As it seen at Figure 5.16 the bank angle have no change until the step. This shows MPC can handle control problem of bank angle.

At 20. second, a step applied to the reference of airspeed. The airspeed reference changed from 23 m/s to 24 m/s. The change of airspeed actually effects all outputs.

If we look at the Figure 5.14 and Figure 5.18, we can see the throttle started to increase to compensate the step on airspeed. But the same control problem that explained about bank angle and reverse moment occured again. When the throttle input increased, the rotor turns faster. That's increases the reverse moment and according to that the change of bank angle. As we observe that change at bank angle with graphs and FlightGear, when step occurs bank angle changes but MPC controller handles is quickly.

The other important change caused by airspeed is picth angle and altitude. Increase of airspeed makes the pitch angle and altitude decrease. But after 20 seconds from step altitude and pitch gets to new equilibrium points (Figure 5.30, Figure 5.32).

As a result; the system is highly nonlinear and coupled. Change at airspeed reference effects all outputs of UAV. Most controllers used in UAV's are seperates the dynamics of the aircraft and uses a few controller to control these outputs. But in this study, just one MPC used to control seven outputs by manipulating four inputs. This approach gives MPC to calculate interaction between outputs and optimum input signals with minimum energy. MPC also takes account of UAVs constraints. That gives us to able to control UAV at safe limits of inputs and outputs.

6.1 Recommendations

In this study, the control case is trying to keep UAV at steady state. MPC got successful at that point. UAV kept flying in steady state after step on airspeed reference. However, UAV lost its path and started to fly in another path after the step. To avoid this situation, path following can be added to control algorithm. This will make it a harder control problem but the most of the control problem of the UAV will be solved.

REFERENCES

Beard, R. B. and McLain, T. W. (2012). Small Unmanned Aircraft Theory and Practice.

Austin, R. (2010). Unamanned Aircraft Systems, UAVs Design, Development and Deployment.

Vachtsevanos, G and Ludington, B.(2007). Modeling and Control of Unmanned Aerial Vehicles- Current Status and Future Directions. *Workshop on Modeling and Control of Complez Systems (MCCS).* Ayia Napa, Cyprus. July 2005.

Raemaekers, A. J. M. (2007). Design of a Model Predictive Controller to Control UAVs

Barnhart, R.K. and Hottman, S. B. And Marshall, D.M. and Shappee, E. (2011). Introduction to Unmanned Aircraft Systems.

Chao, H. and Cao, Y. Chen,Y. (2010). Autopilots for Small Unmanned Aerial Vehicles: A Survey. *International Journal of Control, Automation and Systems.*

Rossiter, J.A. (2005). Model Based Predictive Control, A Pratical Approach.

Wang, L. (2009). Model Predictive Control System Design and Implementation Using MATLAB.

Foelianto, E. and Sumarjono, E. M. (2011). Model Predictive Control for Autonomous Unmanned Helicopters

Cheng, Z. Necsulescu, D. and Kim, B. (2004). Model Predictive Control and Dynamic Inversion for Unmanned Aerial Vehicle. *5th IFAC/EURON Symposium on Intelligent Autonomus Vehicles. Instituto Superior Tecnico, Lisboa, Portugal.* July, 2004.

Url-1 < *http://en.wikipedia.org/wiki/Model_predictive_control*> , date retrieved 04.04.2014

CURRICULUM VITAE

Name Surname: Halit Fırat ERDOĞAN

Place and Date of Birth: İskenderun, 10.05.1989

E-Mail: ferdogan@outlook.com

B.Sc.: Kocaeli University, Mechanical Enginnering.

PUBLICATIONS/PRESENTATIONS ON THE THESIS

- Erdoğan H. F., Kural A., Özsoy C., 2014: İnsansız Bir Hava Aracının Kısıtlamalı Model Öngörülü Kontrolü ve PID Kontrolör ile Kıyaslanması. *National Congress – TOK*,September 11-13, 2014 Kocaeli, Turkey.