ROBUST FLIGHT AND LANDING AUTOPILOT

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY
OZAN DURMAZ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

FEBRUARY 2015

Approval of the thesis:

ROBUST FLIGHT AND LANDING AUTOPILOT

submitted by **OZAN DURMAZ** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver
_____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Gönül Turhan Sayan _____
Head of Department, **Electrical and Electronics Engineering**

Prof. Dr. Kemal Leblebicioğlu _____
Supervisor, **Electrical and Electronics Engineering Dept., METU**

Examining Committee Members:

Prof. Dr. Mübeccel Demirekler _____
Electrical and Electronics Engineering Dept., METU

Prof. Dr. Kemal Leblebicioğlu _____
Electrical and Electronics Engineering Dept., METU

Prof. Dr. Gülbin Dural Ünver _____
Electrical and Electronics Engineering Dept., METU

Prof. Dr. Aydan Erkmen _____
Electrical and Electronics Engineering Dept., METU

Prof. Dr. Mehmet Önder Efe _____
Computer Engineering Dept., Hacettepe University

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Ozan DURMAZ

Signature :

ABSTRACT

ROBUST FLIGHT AND LANDING AUTOPILOT

Durmaz, Ozan
M.S., Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. Kemal Leblebicioğlu

February 2015, 144 Pages

In this thesis, a mathematical model of a small unmanned aircraft is implemented where the static and dynamic stability derivative coefficients are found by Digital DATCOM software. Several control methods are applied such as PID control, LQT, SMC. Two types of Sliding mode controllers are designed using different sliding surfaces. Linearized aircraft models which are trimmed at two different airspeeds are used to design controllers.

A guidance block is implemented to guide aircraft with waypoints. Different guidance methods are used to suppress wind disturbance effects. Model Predictive Control is implemented to track the desired tracks. These desired tracks are generated by steepest decent algorithm. The tracking performances of mentioned controllers are tested in MATLAB/Simulink environment.

Keywords: UAV, Mathematical Modelling of Aircraft, Autonomous Landing, Autopilot, Guidance, PID Control, Sliding Mode Control, Nonlinear Model Predictive Control, Linear Quadratic Tracker, Steepest Descent.

ÖZ

GÜRBÜZ UÇUŞ VE İNİŞ OTOPİLOTU

Durmaz, Ozan
Yüksek Lisans., Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Kemal Leblebicioğlu

Şubat 2015, 144 Sayfa

Bu tez çalışmasında, statik ve dinamik kararlılık türevleri Digital DATCOM programı ile bulunmuş olan, küçük bir insansız hava aracının matematiksel modeli kurulmuştur. Uçağın kontrolü OIT, KKD ve DIDT kontrolcüleri ile yapılmıştır. Farklı kayma yüzeyleri kullanan, İki tip KKD kontrolcü tasarlanmıştır, Söz konusu kontrolcüler, uçağın iki farklı nirengi noktasında hesaplanan, doğrusal modelleri kullanılarak tasarlanmışlardır.

Uçağın yol güzergahları noktalarını takip edebilmesi için bir güdüm bloğu oluşturulmuştur. Rüzgar etkisini bastırabilmek için farklı güdüm teknikleri denenmiştir. Ayrıca sistemin belirli bir rotayı takip etmesi için model öngörümlü denetleyici tasarlanmıştır. Bu rotalar, "en hızlı iniş" optimizasyon tekniği kullanılarak bulunmuştur. Bahsi geçen kontrol algoritmalarının takip performansları MATLAB/Simulink ortamında test edilmiştir.

Anahtar Kelimeler: İHA, Uçağın Matematiksel Modellenmesi, Otomatik İniş, Otopilot, Güdüm, Orantılı İntegral Türetme, Kayan Kipli Denetim, Model Öngörümlü Denetleyici, Doğrusal İkinci Dereceden Takipçi, En Hızlı İniş Yöntemi.

To My Family

ACKNOWLEDGMENTS

First, I would like to thank my supervisor Prof. Dr. Kemal Leblebicioğlu, for his advices, and guidance throughout this research work.

And, I would like to thank all members of my family and my best friends for their support and understanding. Without their support, this work would have been much harder.

TABLE OF CONTENTS

APPENDICES

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

DOF	Degree of Freedom
UAV	Unmanned Aerial Vehicle
LQR	Linear Quadratic Regulator
LQT	Linear Quadratic Tracker
PID	Proportional-Integral-Derivative
AOA	Angle of Attack
AOS	Angle of Sideslip
SMC	Sliding Mode Control
NMPC	Nonlinear Model Predictive Control
NACA	National Advisory Committee for Aeronautics
M	Mach number
USAF	United States Air Force Stability

CHAPTER 1

INTRODUCTION

The small scale unmanned aerial vehicles (UAVs) are becoming more common in today's aviation. They are being used both for military and civil purposes. Small aircrafts are more vulnerable to environmental disturbances due to their small weights [1]. In the present study we planned to introduce control and guidance of the aircrafts in air autopilot systems. The main purpose of autopilots is to make an unstable system stable and to cope with disturbances.

Since the use of an unmanned aerial vehicle on August 22, 1849 for a warfighting, the effectiveness and importance of UAVs are well understood. The most striking example is the victory of Israeli Air Force's over Syrian Air Force in 1982. At war Israel used these drones to gain Intel, jamming and most importantly they used UAVs as decoys to manned aircrafts with coordination which gave the opportunity to destroy Syrian Air Forces with minimal losses. Nowadays, very sophisticated UAVs are developed and being used for civil and military purposes. Anyone can make an unmanned aerial vehicle on his own with little cost. As we give greatly expanded role to UAVs, challenging problems also arises, since a UAV needs to be recovered after a mission, a safe landing must be provided by sophisticated algorithms.

The statistics show that, 47% of aircraft accidents happen during the landing or final approach of flights. Since the landing phase is the most crucial path of flight, landing autopilots are becoming more essential due to their more smoothness than manual landing. Tail and cross winds may cause serious problems while landing. This kind of situations requires high skilled pilots to overcome. Since the required time and money to raise a pilot is demanding. For this reason autopilots are becoming more and more

important. To land an aircraft, path following control methods are widely used. Some of them are way point guidance, generating commands using vector fields and virtual vehicle techniques to drive the aircraft to the desired trajectory. Way point guidance is generally the simplest solution since it's easy to implement, but it leads to simple paths. Vector field technique is also simple but generated paths are often restricted to be planar. With some computational cost, virtual vehicle guidance can generate more flexible paths due to other to guidance methods.

Path following is a main problem for a control system of a vehicle. Generally it is a boundary of some physical entity or generated by a mission planner. In aviation, landing corridors are very popular. In the case it is generated by a physical entity, it is required to develop a sensing strategy to evaluate distance to the desired track and generate the necessary inputs to the system to track it. When the trajectory is generated by a mission planner, it is unclear if it should be updated in real time, since the disturbances or deviations effecting the vehicle or the initial trajectory should kept same and generate a feedback command to track it [2]. A vision based landing algorithm for a helipad used in [3]. The navigation (Global Positioning System) and vision data are combined. The landing target parameters are updated via vision and an onboard controller is used to track a landing path.

It is a challenging task to land with adverse weather effects with pilot or an autopilot, especially tail and cross wind effects. When it comes to small aircrafts, it is far more important to cope with wind and turbulence disturbances while in landing phase. Since some relatively strong deviations may cause a crash. To deal with wind many studies are presented in literature. Vector field approaches used to generate desired course inputs to altitude control law in [4]. Common way to design a control system is to use linearized model of the system at different trim or equilibrium conditions and scheduling the gains of controller in order to achieve good performance. However these methods will not be enough under severe uncertainties and fault conditions. Adaptive nonlinear flight control schemes may be used to overcome to these difficulties [5].

Since the 6-DOF system consists of high nonlinearities, many advanced control methods are used such as PID control, fuzzy logic, neural network, sliding mode control and H_∞ control. HaiYang Chao, YongCan Cao, and YangQuan Chen compared several commercial and research autopilot systems [6]. Fuzzy logic controller was developed by Royer. He uses Mamdami fuzzy system and centroid area in de-fuzification, Malaek investigated the performance of PID (Proportional-Integral-Derivative) controller, neuro controller and neuro-PID controller for the landing mode [7].

In order to more precisely control aircraft, some intelligent techniques are presented like neural networks, adaptive systems and optimal control. Sonny Adiansyah used PI (Proportional-Integral) and PID (Proportional – Integral -Derivative) controllers where the gains of these controllers were gathered from fuzzy logic. He also controlled Telemaster UAV in his M.Sc. Thesis [8]. Shashank Sunil established a simulation model of UAV and studied obstacle avoidance and detection [9]. John Osborne investigated the wind effects on small UAVs and proposed a guidance algorithm including an observer based wind estimator [10]. J. D. Brigido-Gonzalez and H. Rodriguez-Cortes has also used digital DATCOM software to perform an aerodynamic analysis on a radio controlled fixed wing aircraft [11]. Moreover, Volkan Kargın, has designed automatic flight control systems for the autonomous landing of fixed wing unmanned aircraft and simulated the performance of three different controller under different kinds of wind at landing phase [12].

In this thesis, Telemaster UAV is selected as aircraft since it is present at the avionic laboratory in Electrical and Electronics Engineering department in METU. Aircraft's static and dynamic stability derivatives have obtained using United States Air Force Stability (USAF) and Control Digital DATCOM [13]. Several control methods were implemented such as Proportional- Integral-Derivative (PID), linear quadratic tracker (LQT), first order sliding mode controller and nonlinear model predictive control algorithms. For landing, optimal landing paths were generated for reference altitude values using steepest descent algorithm (gradient descent). These generated paths were interpolated in order to be available for, any initial position, located in landing cone,

in MATLAB. Guidance block has designed, the aircraft can move along waypoints. In order to beat the constant and gust wind disturbances lateral track guidance and cross track guidance methods are developed. The performances of developed controllers are observed in predefined scenarios, at landing phase of flight.

CHAPTER 2

MATHEMATICAL MODELING

2.1 Definitions and notation

The body fixed reference frame, represented as $\{X_b, Y_b, Z_b\}$, is used to calculate all states except for $x, y \; and \; z$. Body fixed reference frame is a right handed orthogonal reference system. The center of gravity of the aircraft is the origin of this frame and the X_bZ_b plane divides the aircraft equally if it is symmetric. X_b-Axis is oriented towards the nose of the aircraft, Y_b-axis points to the right wing and the Z_b-axis is directed towards the bottom of the aircraft.

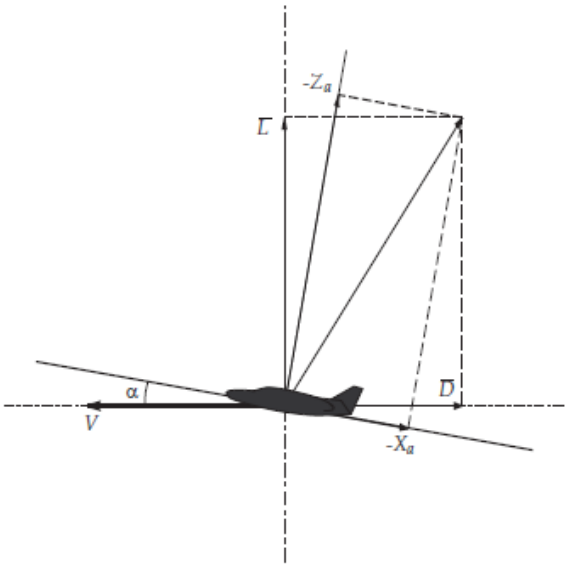

Figure 1- Relationship between aerodynamic forces in flight-path and body-axes

The stability axis reference frame is also a body axis system and donated by $\{X_s, Y_s, Z_s\}$. If there is no sideslip ($\beta = 0$), then OX_s axis points the flight direction.

If there is sideslip $(\beta \neq 0)$ OX_s axis coincides with the projection of the flight velocity vector. This frame is also a right handed reference frame. To compute the 3-D location (x, y, z) of the aircraft the earth fixed reference frame $\{X_e,\ Y_e,\ Z_e\}$ is used. Earth fixed reference frame is also a right handed orthogonal system [14]. Its origin can be placed anywhere, but it must coincide with center of mass of the aircraft when the start of a flight manoeuver takes place.

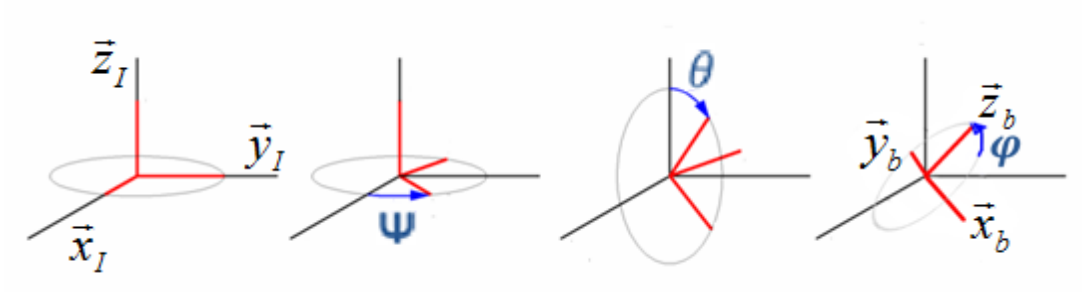

Figure 2 – Euler angles of aircraft

The velocities along body axes $\{X_b,\ Y_b,\ Z_b\}$ are denoted by $\{u, v, w\}$ respectively. The angular velocities are $\{p, q, r\}$ denoted as angular velocities about the body axes and Euler angles $\{\emptyset, \theta, \psi\}$ are bank angle, pitch angle and heading angle, respectively. $\{M_x,\ M_y,\ M_z\}$ and $\{F_x,\ F_y,\ F_z\}$ are total moments and forces acting on the aircraft in the body frame axis [15].

2.2 UAV planform

For the simulation purposes Telemaster UAV was implemented on USAF Stability and Control Digital DATCOM software. DATCOM simply takes the geometric data of the aircraft and returns its static and dynamic aerodynamic coefficients [16]. The software considers control surface deflections as inputs and the coefficient outputs are dependent on the aerodynamic angles α and β. Basic geometry and inertia parameters are illustrated in Table 1.

Table 1 – Mass and basic geometry of Telemaster UAV

Mass	4.1 kg
I_x	0.22 kg-m^2
I_y	0.31 kg-m^2
I_z	0.45 kg-m^2
Wing area	1.56 m^2
Wing span	1.83 m
Wing mean aerodynamic chord	0.30 m

2.2.1 Airfoil Designations

Since we explained the common geometry of airfoils, we had to describe their camber of the mean line as well as the stations thickness distribution along airfoils. DATCOM uses NACA airfoil series to describe airfoils. NACA four-digit series have been used to implement the wing, vertical tail and horizontal tail, since it's the simplest way to describe an airfoil. The first digit designates the maximum camber (m) in percentage of the chord, the second indicates the position of the maximum camber (p) in tenths of chord, and the last two digits implies the maximum thickness(t) of the airfoil in percentage of the chord. These parameters are described in Figure 3.

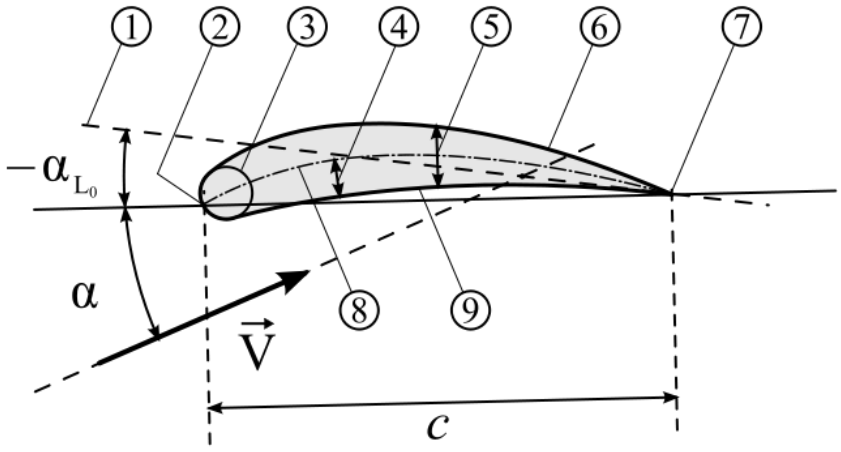

Figure 3 - Profile geometry – 1: Zero lift line; 2: Leading edge; 3: Nose circle; 4: Camber; 5: Max. Thickness; 6: Upper surface; 7: Trailing edge; 8: Camber mean-line; 9: Lower surface

The mean camber line in four-digit series is found by the formula:

$$y_c = \begin{cases} m\frac{x}{p^2}\left(2p - \frac{x}{c}\right), & 0 \le x \le pc \\ m\frac{c-x}{(1-p)^2}\left(1 + \frac{x}{c} - 2p\right), & pc \le x \le c \end{cases} \quad (2.1)$$

Where m is the camber in percentage of the chord, c is the length of the chord and p is the location of maximum camber.

And the coordinates of upper and lower surfaces of airfoil is calculated as:

$$x_U = x - y_t cos(\theta) \qquad y_U = y_c - y_t cos(\theta) \quad (2.2)$$

$$x_L = x - y_t cos(\theta) \qquad y_L = y_c - y_t cos(\theta) \quad (2.3)$$

Where

$$\theta = arctan\left(\frac{dy_c}{dx}\right) \quad (2.3)$$

$$\frac{dy_c}{dx} = \begin{cases} \frac{2m}{p^2}\left(p - \frac{x}{c}\right), & 0 \le x \le pc \\ \frac{2m}{(1-p)^2}\left(p - \frac{x}{c}\right), & pc \le x \le c \end{cases} \quad (2.4)$$

For Telemaster NACA airfoils are selected as;

NACA-W-4-4415

NACA-V-4-0018

NACA-H-4-4415

In this notation NACA implies that it's a NACA airfoil, W-H-V stands for wing, horizontal tail and vertical tail respectively. ''4'' indicates that it is a 4 digit airfoil. After that is the airfoil designation number. The final aircraft looks like Figure 4.

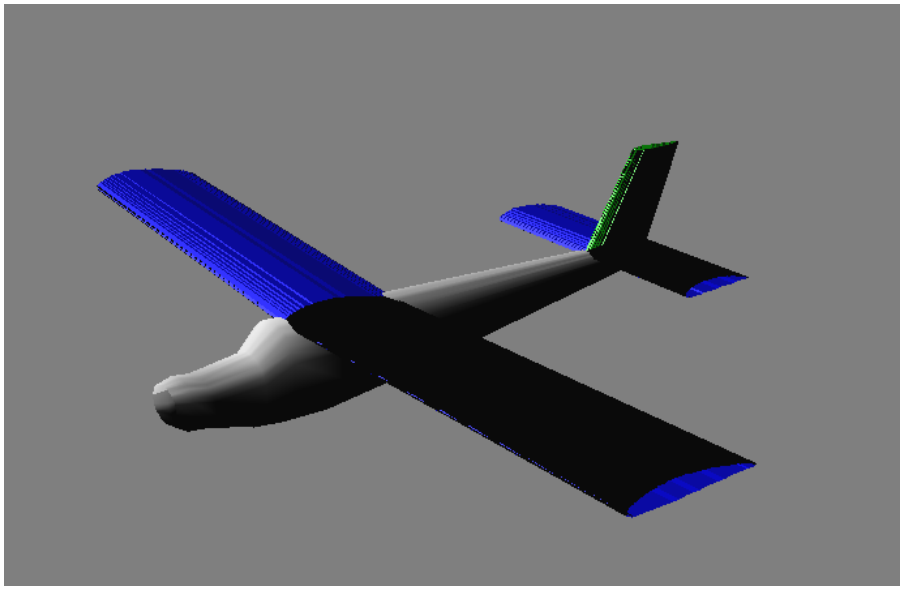

Figure 4 – Telemaster UAV in AC3D file format.

Complete aircraft and airfoils implemented with DATCOM are visualized by MATLAB in figure 5:

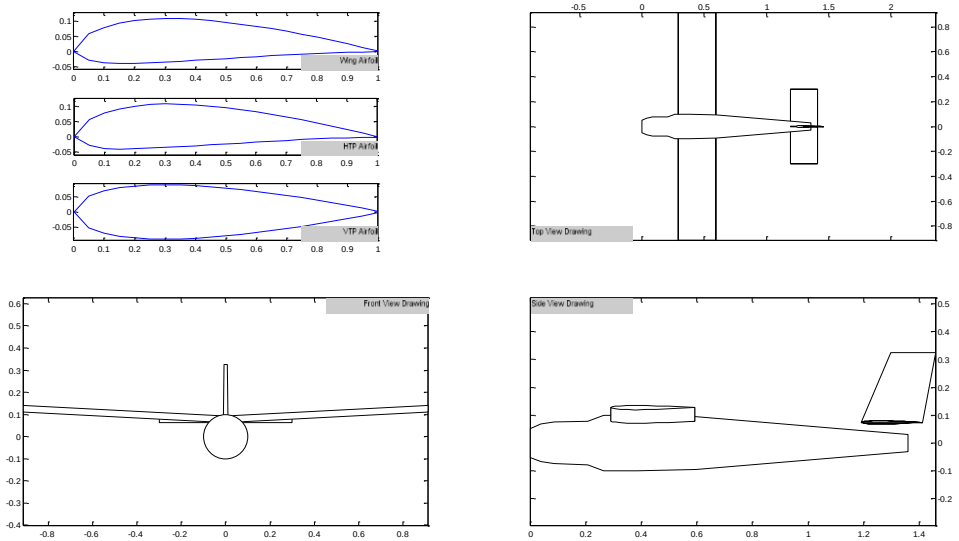

Figure 5 – From left top to right bottom: wing, horizontal tail, vertical tail airfoils, top view, front view and left side view.

In order to obtain aerodynamic forces and moments which are greatly depend on flight condition parameters, we need to calculate and: angle of attack (α), angle of sideslip (β), Mach number (M) and dynamic pressure (ρ).

- Angle of attack (AOA) is the angle between chord line of the wing and the relative motion vector of aircraft in atmosphere. It is denoted by α.

- Angle of sideslip (AOS) which is denoted by β, is the rotation of the aircraft center line from the wind.

- Mach number (M) is the ratio of speed of an aircraft and the local speed of sound which is given by:

$$M = \frac{V}{C} \tag{2.6}$$

Where $C = \gamma RT$, here γ is the ratio of specific heat of a gas at a constant pressure to heat at a constant volume and it is equal to 1.4. R is the universal gas constant which is $287 J/kgK$. T is the temperature.

- Dynamic pressure is kinetic energy per unit volume of a fluid particle. It is given by:

$$q = \frac{1}{2}\rho V^2 \tag{2.7}$$

Where q is the dynamic pressure in pascals, ρ is the air density in kg/m^3 and V is the velocity of relative air (m/s).

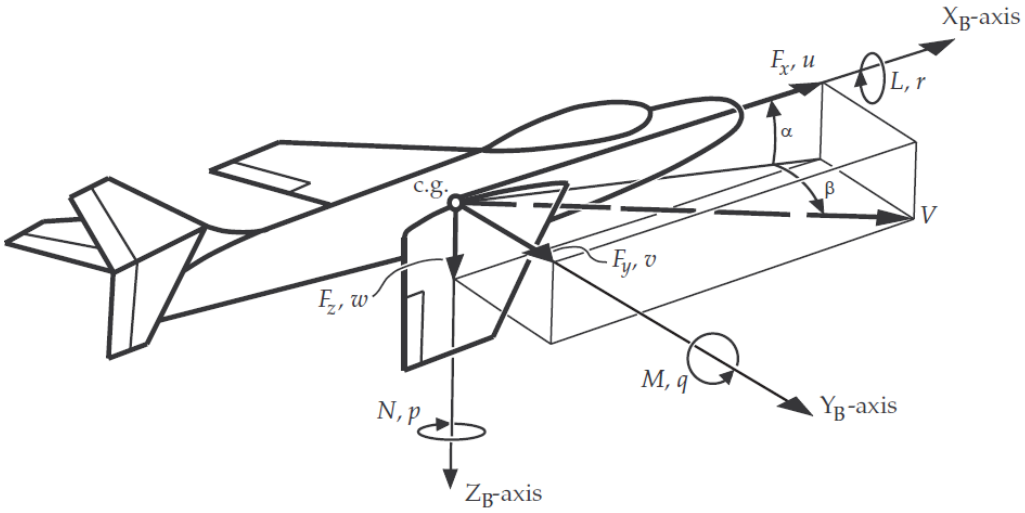

Figure 6 – Acting forces and moments on an aircraft's body axes

DATCOM software is used to calculate dynamic and static stability derivatives where the program takes number of mach numbers, angle of attacks, and angle of sideslips to be calculated at, as inputs. In our case, it is decided to have one mach number which is 0.05 approximately 18 m/s speed and eighteen angle of attack values which can be seen in Figures 7 to 17.

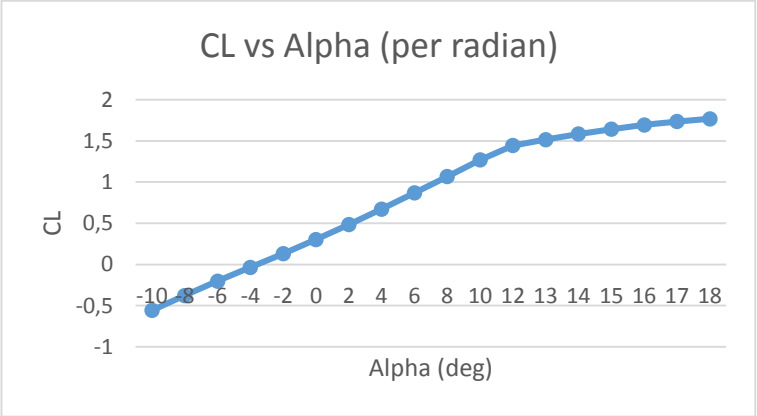

Figure 7 – Lifting force coefficient change with alpha ($\boldsymbol{C_Z}$)

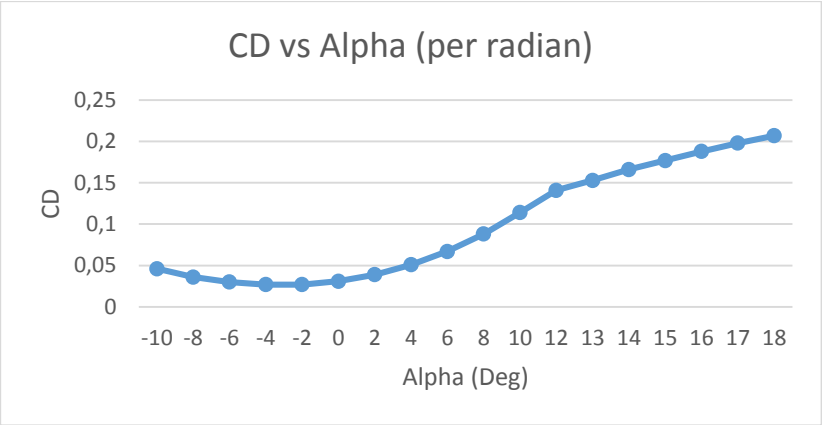

Figure 8 – Drag force coefficient change with alpha ($\boldsymbol{C_X}$)

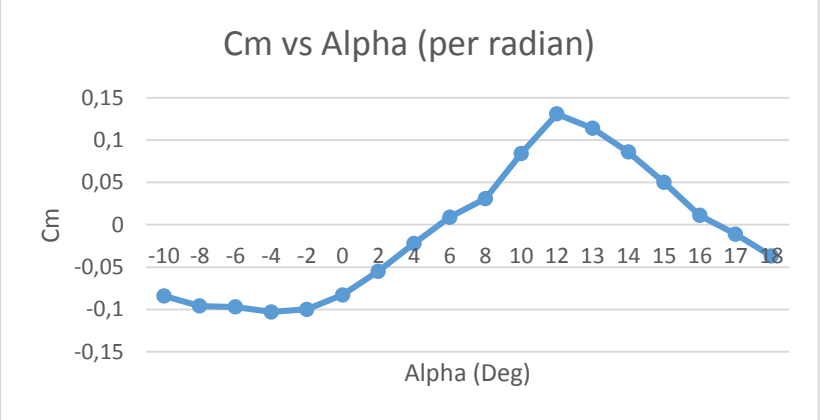

Figure 9 – Pitching moment coefficient change with alpha ($\boldsymbol{C_m}$)

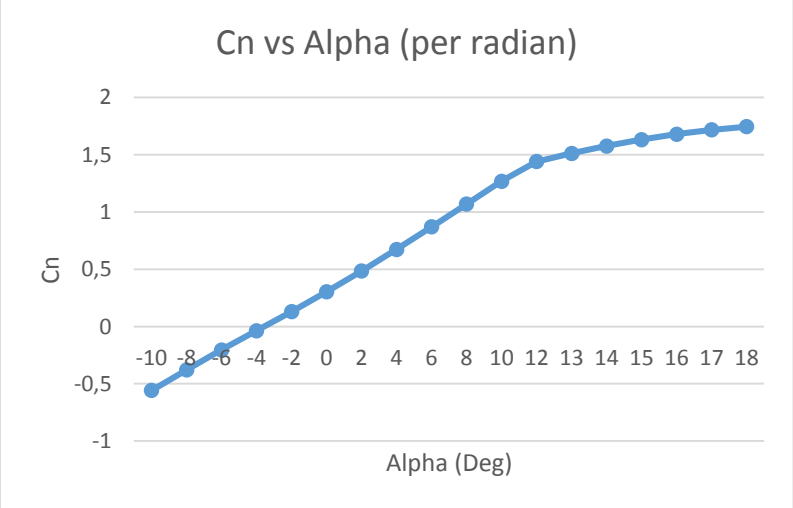

Figure 10 – Yawing moment coefficient change with alpha ($\boldsymbol{C_n}$)

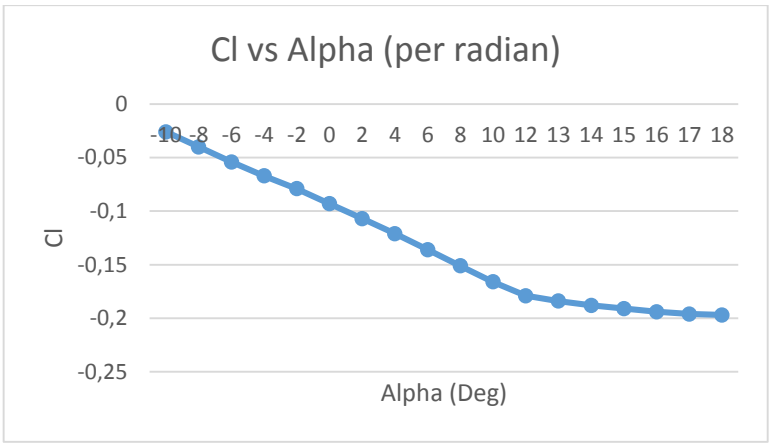

Figure 11 – Rolling moment coefficient change with alpha ($\boldsymbol{C_l}$)

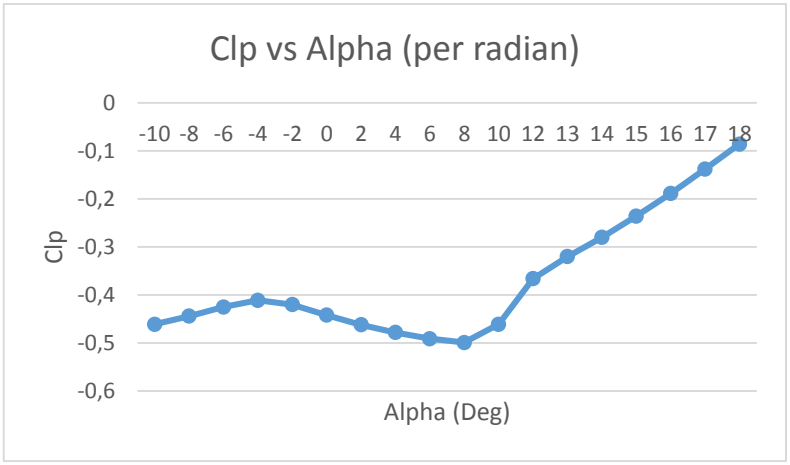

Figure 12 – Rolling moment dynamic derivative change with alpha ($\boldsymbol{C_{l_p}}$)

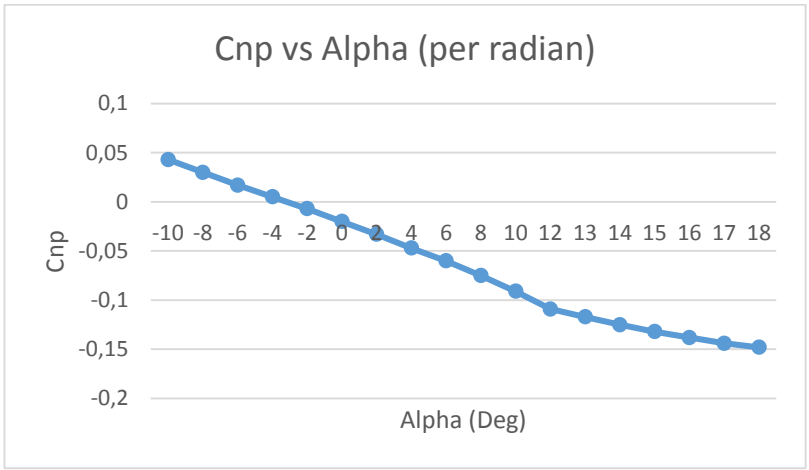

Figure 13 – Yawing moment dynamic derivative change with alpha ($\boldsymbol{C_{n_p}}$)

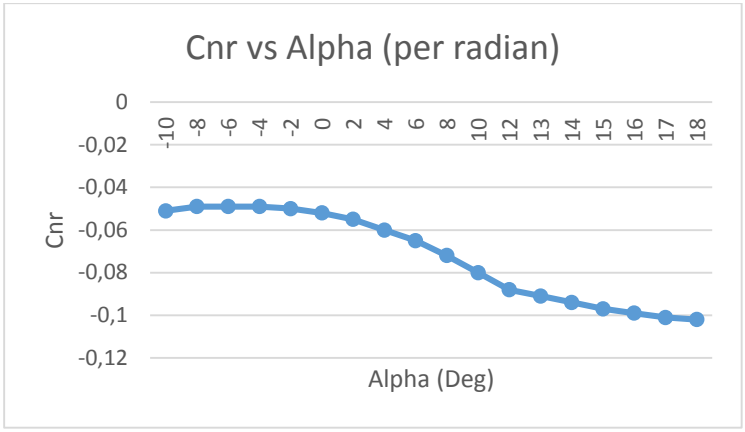

Figure 14 – Yawing moment dynamic derivative change with alpha ($\boldsymbol{C_{n_r}}$)

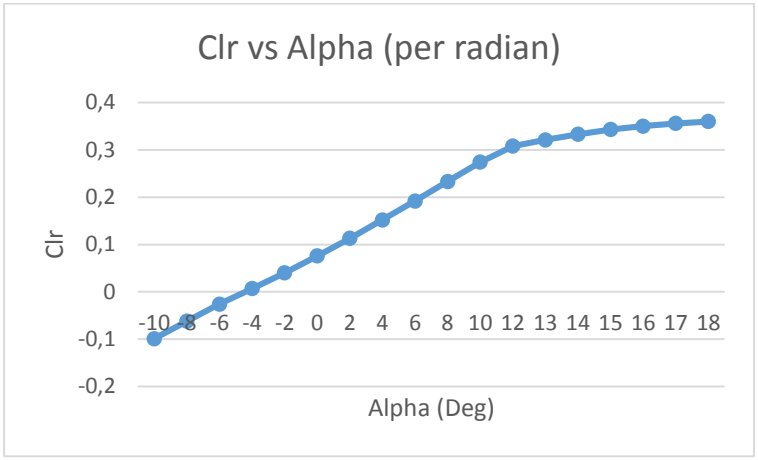

Figure 15 – Rolling moment dynamic derivative change with alpha ($\boldsymbol{C_{l_r}}$)

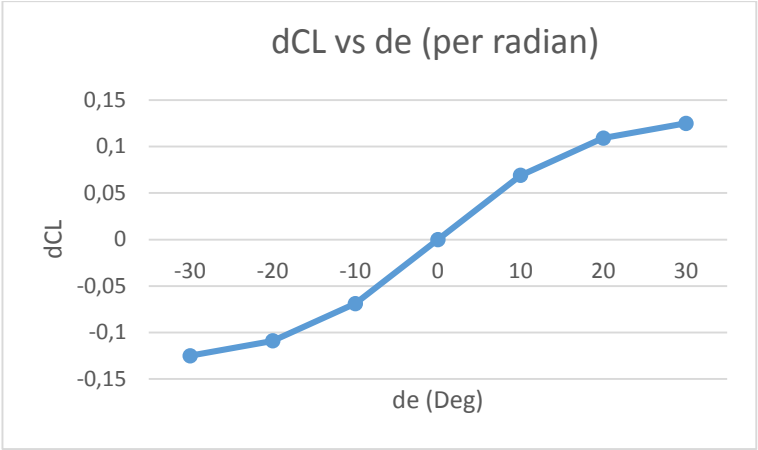

Figure 16 – Lifting force derivative change with elevator deflection ($C_{Z_{\delta_e}}$)

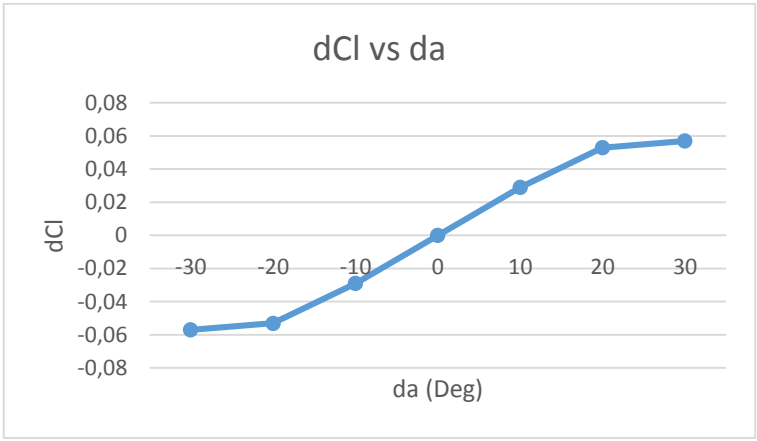

Figure 17 – Rolling moment derivative change with aileron deflection ($C_{l_{\delta_a}}$)

2.3 Equations of motion

Newton's second law for rigid-body dynamics is used to derive non-linear aircraft equations [17]. The general force and moment equations are given by:

$$F = m(\frac{\partial V}{\partial t} + \Omega x V) \tag{2.8}$$

$$M = \frac{\partial (I.\Omega)}{\partial t} + \Omega \text{ x } (\text{I} \cdot \Omega) \tag{2.9}$$

V = $[u\ v\ w]^T$ is the velocity vector at center of gravity, $\Omega = [p\ q\ r]^T$ is the angular velocity vector about the center of gravity. $F = [F_x\ F_y\ F_z]^T$ is the total force vector which consists of engine thrust, aerodynamic forces, wind forces and gravitational forces. $M = [L\ \ M\ \ N]^T$ consists of moments acting to the aircraft. I is the inertia tensor of the rigid body and it's given by:

$$I = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix} \tag{2.10}$$

The aerodynamic forces acting on body of aircraft are computed from the stability derivatives which are obtained from DATCOM. Where

$$C_{X_a} = C_{X_0} + C_{X_\alpha}\alpha + C_{X_q}\frac{qc}{V} + C_{X_{\delta_e}}\delta_e \tag{2.11}$$

$$C_{Y_a} = C_{Y_0} + C_{Y_\beta}\beta + C_{Y_p}\frac{pb}{2V} + C_{Y_r}\frac{rb}{2V} + C_{X_{\delta_a}}\delta_a \tag{2.12}$$

$$C_{Z_a} = C_{Z_0} + C_{Z_\alpha}\alpha + C_{Z_q}\frac{qc}{V} + C_{Z_{\delta_e}}\delta_e \tag{2.13}$$

$$C_{l_a} = C_{l_0} + C_{l_\beta}\beta + C_{l_p}\frac{pb}{2V} + C_{l_r}\frac{rb}{2V} + C_{l_{\delta_a}}\delta_a \quad (2.14)$$

$$C_{m_a} = C_{m_0} + C_{m_\alpha}\alpha + C_{m_q}\frac{qc}{V} + C_{m_{\delta_e}}\delta_e \quad (2.15)$$

$$C_{n_a} = C_{n_0} + C_{n_\beta}\beta + C_{n_p}\frac{pb}{2V} + C_{n_r}\frac{rb}{2V} + C_{n_{\delta_a}}\delta_a \quad (2.16)$$

Subscripts X, Y, Z and l, m, n stand for

Using these stability derivatives the dimensional aerodynamic forces can be calculated as, $[N]$:

$$X_a = C_{X_A} q_{dyn} S \quad (2.17)$$

$$Y_a = C_{Y_A} q_{dyn} S \quad (2.18)$$

$$Z_a = C_{Z_A} q_{dyn} S \quad (2.19)$$

Dimensional aerodynamic moments are, $[Nm]$:

$$L_a = C_{l_A} q_{dyn} S b \quad (2.20)$$

$$M_a = C_{m_A} q_{dyn} S \mathrm{c} \quad (2.21)$$

$$N_a = C_{N_A} q_{dyn} S b \quad (2.22)$$

And the total forces acting on aircrafts body along body axes, are defined as follows, $[N]$:

$$F_X = X_a + X_g + X_p + X_w \tag{2.23}$$

$$F_Y = Y_a + Y_g + Y_p + Y_w \tag{2.24}$$

$$F_Z = Z_a + Z_g + Z_p + Z_w \tag{2.25}$$

Resulting moments about body axes, $[Nm]$:

$$L = L_a + L_p \tag{2.26}$$

$$M = M_a + M_p \tag{2.27}$$

$$N = N_a + N_p \tag{2.28}$$

Using (2.8) and (2.9) the translational and rotational dynamics of aircraft can be found as:

$$\dot{u} = \frac{F_X}{m} - qw + rv \tag{2.29}$$

$$\dot{v} = \frac{F_Y}{m} - ru + pw \tag{2.30}$$

$$\dot{w} = \frac{F_Z}{m} - pv + qu \tag{2.31}$$

$$\dot{p} = P_{pp}p^2 + P_{pq}pq + P_{pr}pr + P_{qq}q^2 + P_{qr}qr + P_{rr}r^2 + P_lL + P_mM + P_nN \tag{2.32}$$

$$\dot{q} = Q_{pp}p^2 + Q_{pq}pq + Q_{pr}pr + Q_{qq}q^2 + Q_{qr}qr + Q_{rr}r^2 + Q_lL + Q_mM + Q_nN \tag{2.33}$$

$$\dot{r} = R_{pp}p^2 + R_{pq}pq + R_{pr}pr + R_{qq}q^2 + R_{qr}qr + R_{rr}r^2 + R_lL + R_mM + R_nN \tag{2.34}$$

2.4 Trimming and linearization of aircraft

In order to linearize an aircraft, we need to know its equilibrium point where all dynamical equations are set to zero. Since the speed value has a high impact on dynamics of aircraft, different trim points for different speeds will be obtained. The equilibrium condition is provided when the force/ moment equations are balanced [18].

$$X_0 - mg\sin\theta = 0 \tag{2.35}$$

$$Z_0 + mg\cos\theta = 0 \tag{2.36}$$

$$Y_0 = L_0 = M_0 = N_0 = 0 \tag{2.37}$$

Trimmed condition of aircraft requires the correct simultaneous adjustment of dynamic equations in six degrees of freedom. It is depended on airspeed, flight path angle, airframe configuration, weight and center of gravity position. When any of these parameters is changed than the whole trimming operation needs to be revised [19].

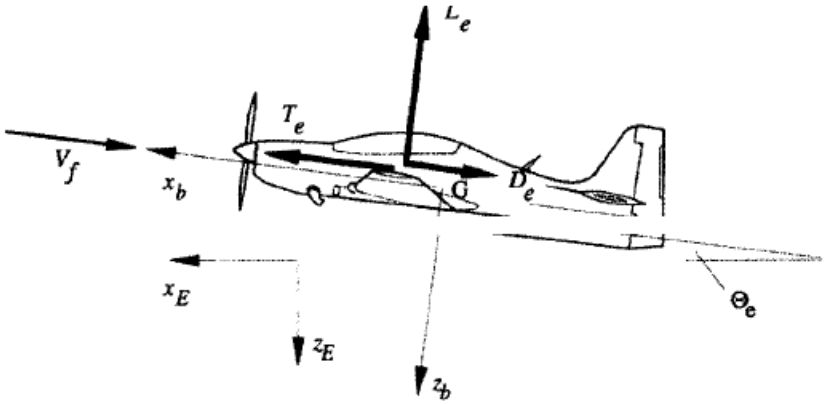

Figure 18 – Trimmed flight

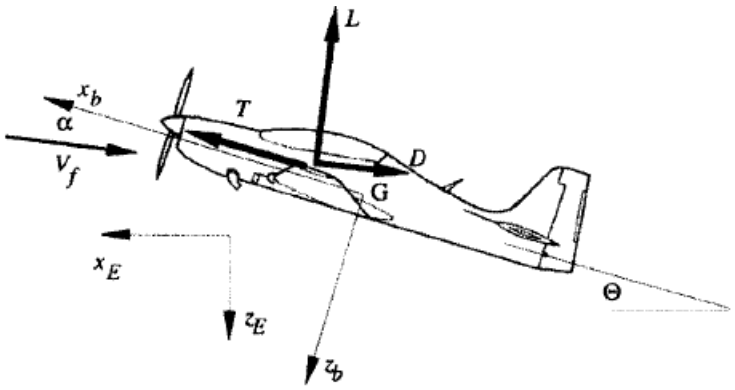

Figure 19 – Deflected flight from trimmed condition

As mentioned before the effect of airspeed to the equilibrium point, the aircraft is trimmed and linearized for two different speed values, 12m/s and 18m/s which are landing and level flight speed values respectively. Then the aircraft is linearized at these trim conditions. MATLAB's control analysis toolbox is used in linearizing and trimming of the aircraft. The Simulink block diagram used in this purpose is illustrated in Figure 20.

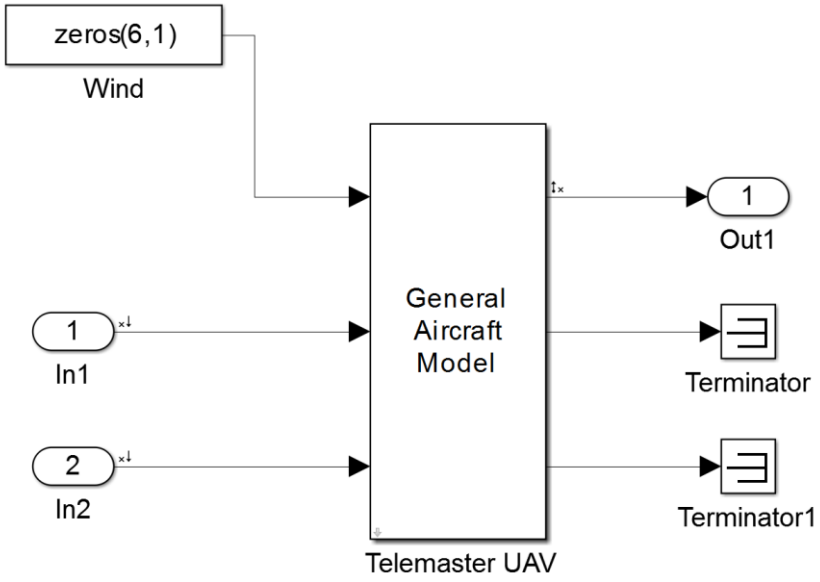

Figure 20 – Simulink block used in trimming and linearizing of Telemaster UAV

CHAPTER 3

AUTOPILOT DESIGN

After linearization of the system at two trim points, the resultant state space models are used to design controllers. The control block separates into two blocks containing longitudinal and lateral autopilots, where longitudinal block uses elevator and throttle and lateral block uses aileron and rudder as control channels. Longitudinal autopilot controls the speed and pitch angle of the aircraft, and lateral autopilot controls the roll and yaw angle of the aircraft. The block diagram of closed loop system is provided in Figure 21.

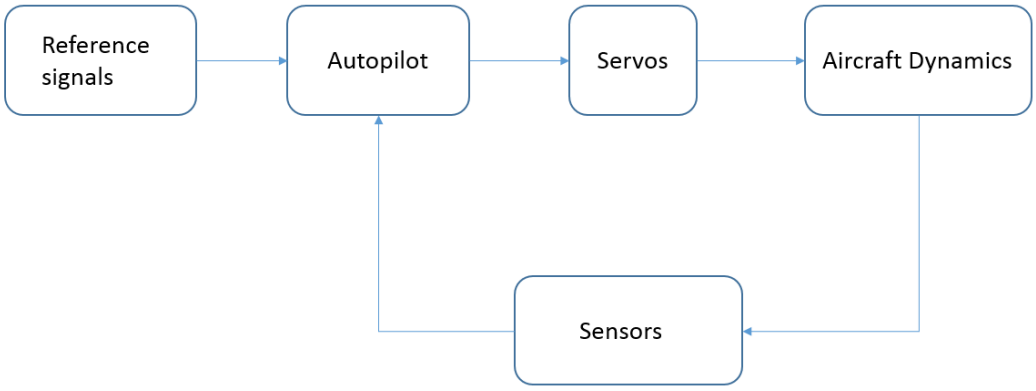

Figure 21 – Block diagram of closed loop system

It’s required to design an autopilot that both considers level flight, approach and landing phases of Telemaster UAV. First a Proportional-Integral-derivative (PID) controller designed to control aircraft. An optimal control approach, linear quadratic tracker, is implemented after PID controller. After that, a first order sliding mode

control (SMC) and nonlinear model predictive control (MPC) methods are implemented and later, these methods are compared to each other in the sense of, level flight and landing, performances of these controllers in some predefined scenarios.

3.1 Proportional-Integral-derivative (PID) controller

A Proportional-Integral-derivative (PID) controller and its modified versions are very common feedback loop element in today in industry [20]. In order to set the measured value to a desired reference point, PID controller looks for the difference of measured and desired values (or 'error' signal) and produces some input to the process.

A PID can adjust the output of the system based on the history and rate of change of the error signal. Different from more complex control algorithms such an optimal control theory, PID design does not require advanced mathematics [21].

- Proportional action

$$u\,(t) = K_p e(t) + u_b \tag{3.1}$$

According to this part of the controller, the input to the process is proportional to the control error. Where u_b is a bias and K_p is the proportional gain.

- Integral action

$$u(t) = K_i \int_0^t e(\tau) d\tau \tag{3.2}$$

The integral action provides that the steady state error goes to zero. Its contribution is proportional to both magnitude of error and its duration. This will increase the speed of convergence to the desired reference but since it responds to accumulated errors from past this will cause some overshoot the reference value. Here K_i is the integral gain.

- Derivative action

$$u(t) = K_d \frac{de(t)}{dt} \tag{3.3}$$

Derivative action produces a control signal proportional to an estimate of error at the next time instant. It calculates the slope of error over time and multiplies this with the gain K_d. This action improves the settling time and overshoot of the system, increasing stability [22, 23].

3.1.1 Lateral control

Heading Controller

The heading controller takes yaw angle, roll angle and roll rate as inputs and generates an aileron command to reach a desired state heading. It consists of one inner loop and an outer loop, where the outer loop produces a roll command from initial heading error processed by a PI controller. Summing up the error and derivative of Ψ_c speeds up the response of the controller. Saturation block bounds the roll command to $\pm 30^o$ [24].

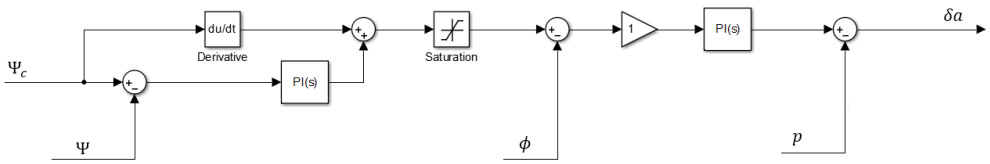

Figure 22 – Block diagram of heading controller

After generating roll command in the outer loop, inner loop looks for roll angle error and generates an aileron command with a PI controller.

Table 2 – Heading controller PI gains

	P	I
Outer loop PI controller	0.9	0.0001
Inner loop PI controller	1.2	0.02

The response of aircraft to the given psi angle command is given in Figure 23.

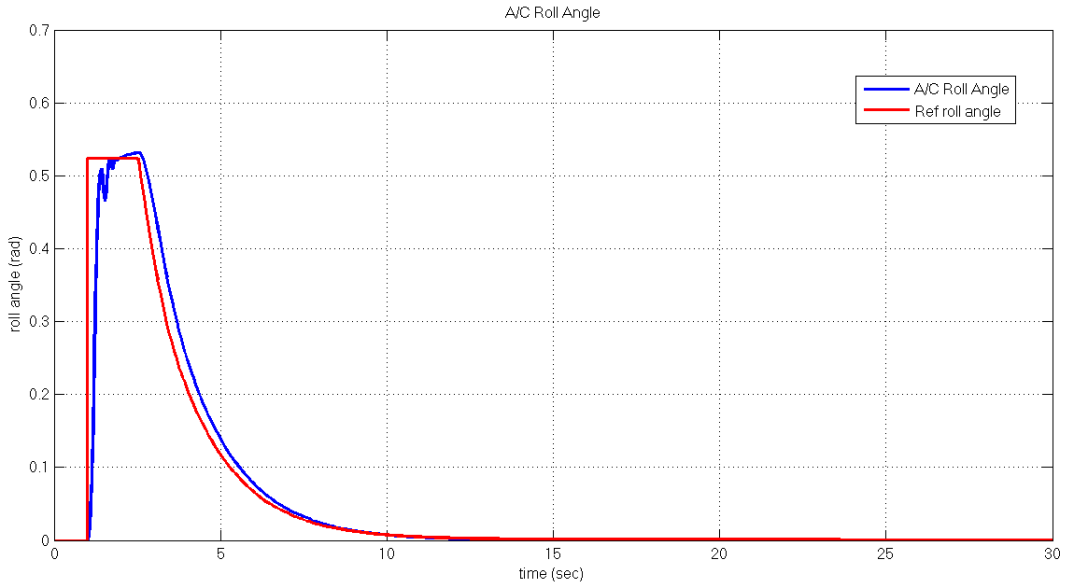

Figure 23 – Roll angle response of aircraft wrt step input from aileron channel

Yaw rate controller

The main purpose of yaw damper is to counteract the Dutch-roll aerodynamic mode. The main reason for Dutch-roll is that the lateral stability is relatively stronger than the directional stability of the aircraft. It is artificially corrected in this thesis. To overcome this situation a yaw damper has implemented. This type of controller tends to reject any change in yaw rate. To overcome this situation, a wash-out block is used. The purpose of wash-out block is to differentiate the signal from the yaw rate gyroscope [25].

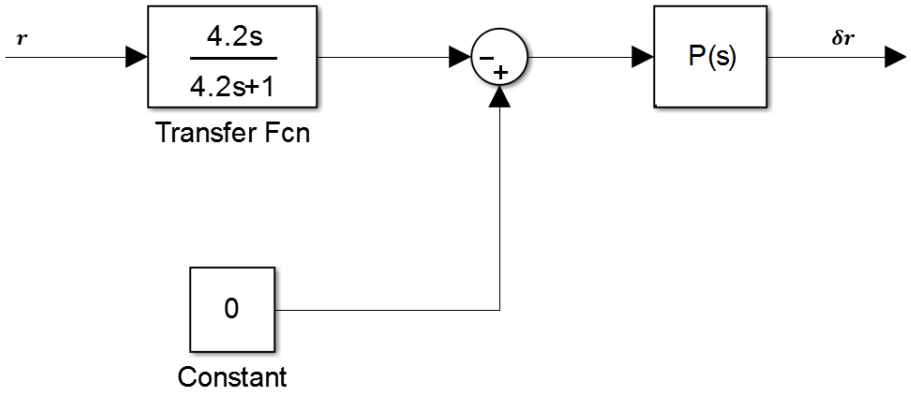

Figure 24 – Block diagram of yaw damper

The wash-out block has the following transfer function:

$$\frac{s}{\left(1+{}^{1}/_{T_{wo}}\right)}=\frac{sT_{wo}}{1+sT_{wo}} \tag{3.4}$$

It is important to select T_{wo} (time constant). If it is too short, the yaw damper will have less time to act which leads to less effectiveness. If it is too long, then the stability problems arise. By trial and error methodology 4.2 is selected as time constant in simulations. The yaw rate response of the aircraft after giving a step input from aileron channel is provided in Fig. 25.

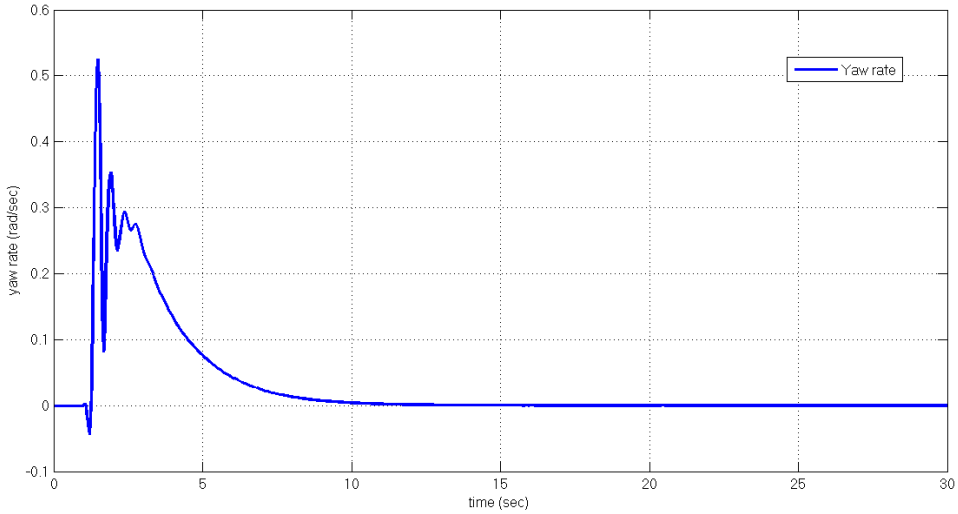

Figure 25 – Yaw rate ($\boldsymbol{r}$) response of the aircraft wrt step input given from aileron channel

3.1.2 Longitudinal control

Longitudinal control includes speed and altitude controllers. These states are very correlated with each other, for level flight separating these states maybe logical but while landing since its required to be in specific states at a certain time, it would be wise to design a controller which includes both of these coupled states $[v, h]$ and process them at the same time.

Speed controller

For level flight a simple speed controller has designed. A throttle command is generated using a PID block, fed by the error between commanded speed and the feedback value of speed.

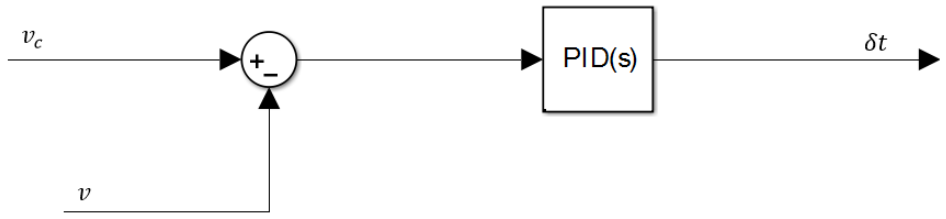

Figure 26 – Block diagram of speed controller (PID)

Table 3 – Speed controller PID gains

	P	I	D
Speed controller	35	1.6	-2.7

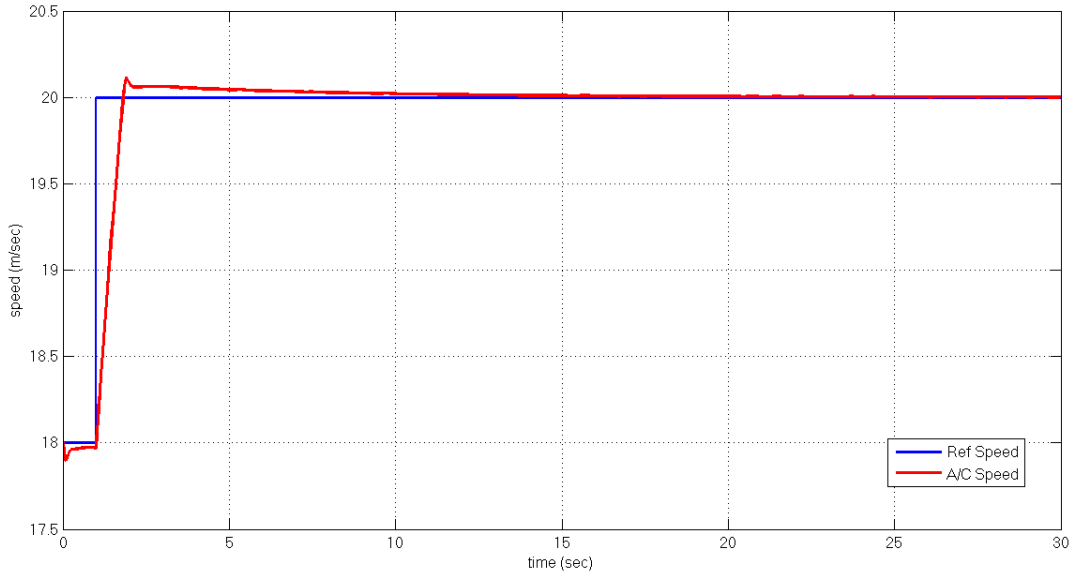

Figure 27 – Speed response of aircraft wrt given reference command

Altitude controller

Altitude controller consists of two loops, inner and outer. Inner loop generates an elevator command using pitch angle, pitch rate where the outer loop looks for the error of altitude with commanded altitude. The block diagram of controller is illustrated in Fig. 28.

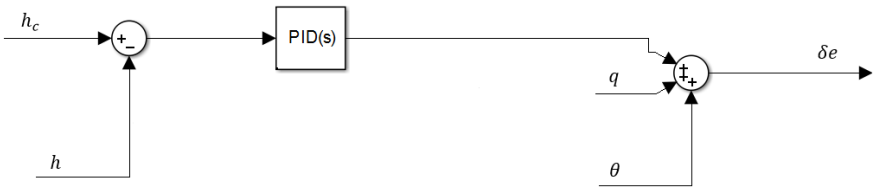

Figure 28 – Block diagram of altitude controller (PID)

Here the altitude error is used to generate elevator command and states q (pitch rate) and θ (pitch angle) are added with the output of PID block to increase the speed of controller. Obtained gain values for PID can be seen in table 5. The proportional gain

is negative since the elevator deflection angle and the resultant movement happens to be in opposite directions.

Table 4 – Altitude controller PID gains

	P	I	D
Altitude controller	-0.1	0.01	0.8

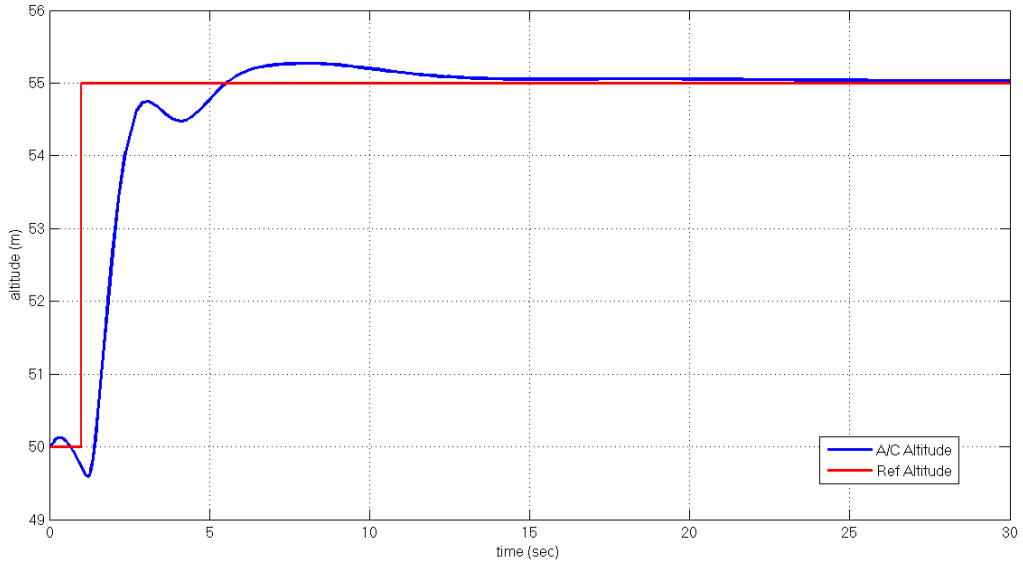

Figure 29 – Altitude response of aircraft wrt given reference command

3.2 Linear Quadratic Tracker (LQT)

Linear quadratic tracker (LQT) is an optimal control approach which simply tries to find best controller that has the best performance according to a performance index. This method has very similar philosophy with linear quadratic regulator (LQR) uses. Assume a linear time-invariant systems is defined as:

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{3.5}$$

$$y(t) = Cx(t) \tag{3.6}$$

Where x(t) $\epsilon\ R^n$ and u(t) $\epsilon\ R^m$ and y(t) is the output of the system ($Q \geq 0$, $R > 0$ and $Q_f \geq 0$). A control law that makes this linear system stable and robust is guaranteed if (A, B) are stabilizable, (A, C) are observable.

Since it's desired that the designed controller drives the states to their reference values, the performance index is defined as containing not only the states but the error of the states with their desired values. The performance index for our work is given in equation (3.7):

$$J = \frac{1}{2}(y(T) - r(T))^T Q_f (y(T) - r(T)) + \frac{1}{2}\int_0^T [(y(t) - r(t))^T Q(y(t) - r(t)) + u^T Ru]dt \tag{3.7}$$

Here Q and R, are weighting matrices that Q corresponds to weight of energy of the controlled output and R stands for the weight of input signal energy. We desire to find a control input u(t), for the given matrices Q and R that makes the performance index minimum. Selecting a large R will tend to employ a small control signal to the system, at the expense of a large controlled output. When we choose a large Q, we will obtain a very small controlled output but in the expense of a large input command [25]. The optimal control for given performance index is given by [27]:

$$u(t) = -K(t)x(t) + R^{-1}B^{T}v(t) \tag{3.8}$$

where

$$K(t) = R^{-1}B^{T}S(t) \tag{3.9}$$

and S(t) is the solution to Riccati equation:

$$-\dot{S} = A^{T}S + SA - SBR^{-1}B^{T}S + Q \tag{3.10}$$

$$S(t) = Q_f \tag{3.11}$$

with

$$-\dot{v} = (A - BK)^{T}v - Qr(t) \tag{3.12}$$

$$v(t) = Q_f r(t) \tag{3.13}$$

If we define $t_0 = 0$ and $Q_f = 0$, Also assuming the pair (A, B) is stabilizable and (A, C) is observable, the Algebraic Ricatti equation goes to its steady state solution, so

$$A^{T}S_{ss} + S_{ss}A - S_{ss}BR^{-1}B^{T}S_{ss} + Q = 0 \tag{3.14}$$

The solution of the Algebraic Ricatti Equation (ARE) is becomes:

$$u(t) = -K_{ss}x(t) + Fr \tag{3.15}$$

where

$$F = -R^{-1}B^{T}[(A - BK)^{T}]^{-1}Q \tag{3.16}$$

Since we desire the Telemaster aircraft to follow desired waypoints, it's required to control speed, heading and altitude of the aircraft. For that purpose three controllers designed for roll, heading and pitch attitude using LQT method. Linearized aircraft dynamics for V = 18 m/s and V = 12 m/s are used to design LQT. Since the system has 12 states coupled with each other, these states are separated and three single-input single-output systems are created.

$$[K\ S] = \ lqr(A, B, Q, R) \tag{3.17}$$

The MATLAB's 'lqr' command is used to calculate the necessary gain matrix 'K' and solution to the Ricatti equation, the gain matrix F is obtained using the previously described formula (3.17). The main block diagram of LQT controller is illustrated in Fig. 30.

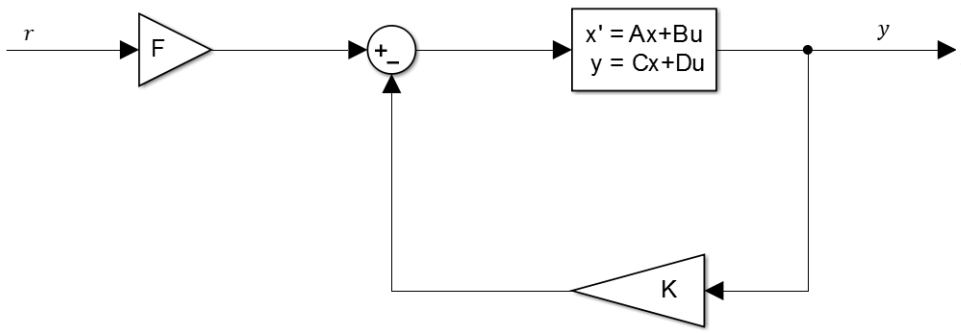

Figure 30 – LQT controller block diagram

3.2.1 Roll controller (LQT)

The states $[\beta; p; r; \emptyset]$ are selected for roll attitude controller. First the linearized aircraft is analyzed. The block diagram for roll controller is given in Figure 31.

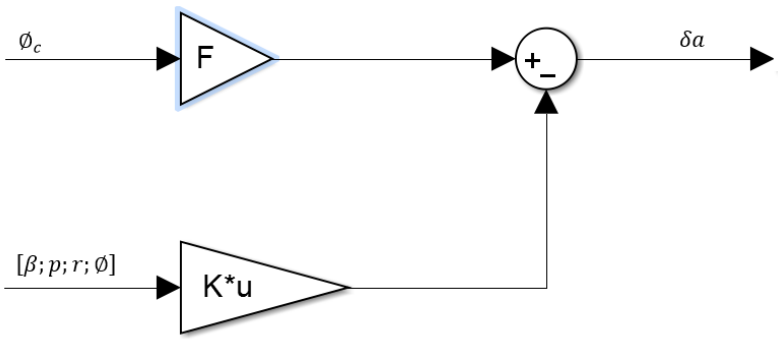

Figure 31 – LQT roll controller block diagram

Step response of the LQT for an input given in aileron channel is given in Figure 32.

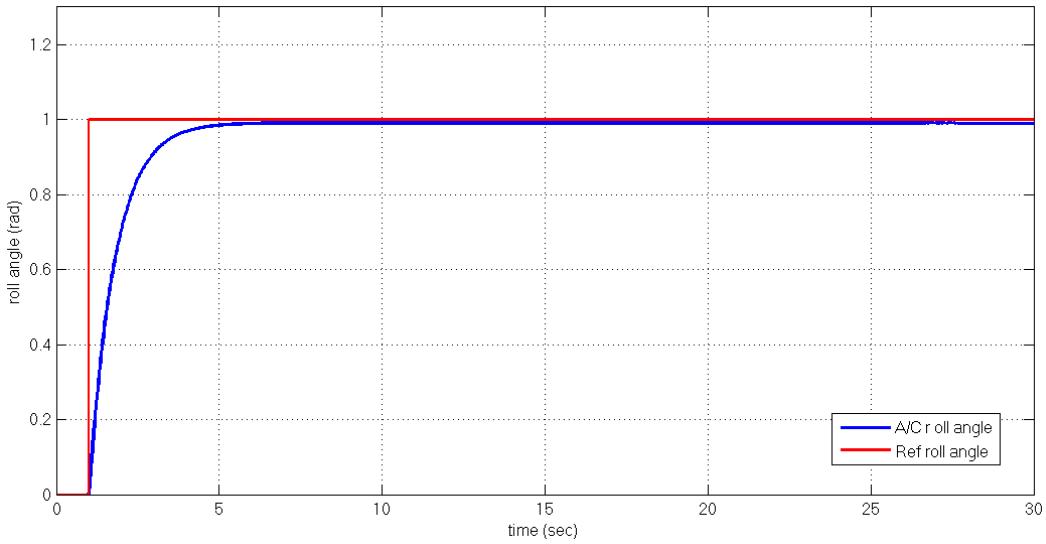

Figure 32 – Roll angle response of aircraft wrt step input from aileron channel

3.2.2 Altitude controller (LQT)

The states $[\alpha; q; \theta; h]$ are selected for altitude attitude controller. First the linearized aircraft is analyzed. Step response of the LQT for an input given in elevator channel is given in Fig. 33.

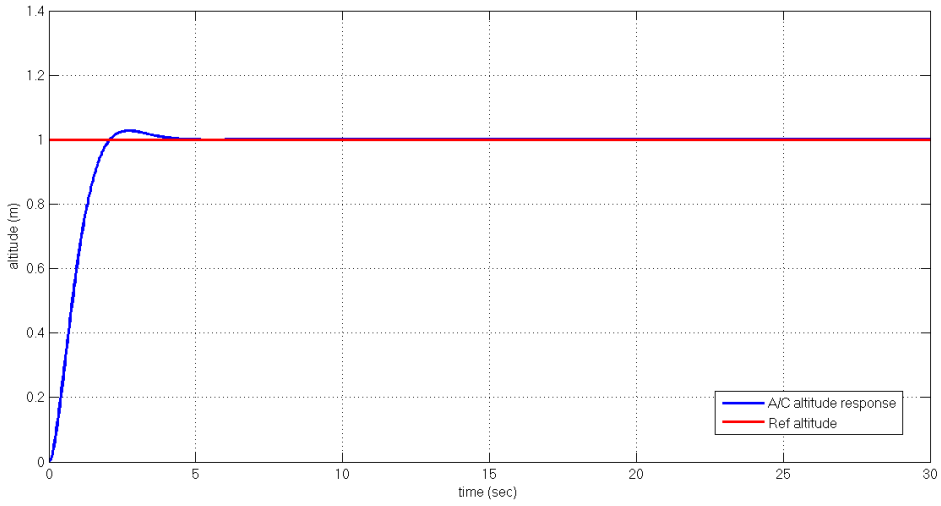

Figure 33 –Altitude response of aircraft wrt step input from elevator channel

3.2.3 Speed controller (LQT)

The states $[v; \alpha; q; \theta]$ are selected for altitude speed controller. First the linearized aircraft is analyzed. Step response of the LQT for an input given in throttle channel is given in Fig. 34.

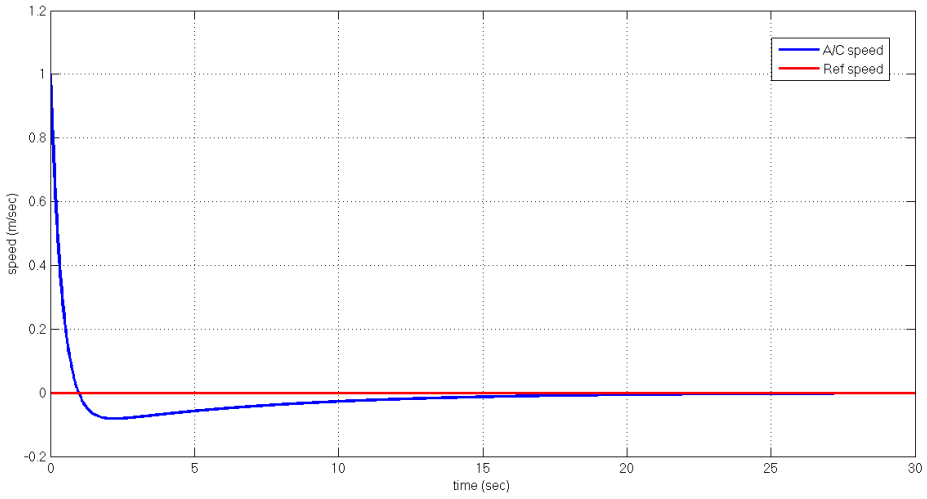

Figure 34 –Speed response of aircraft wrt step input from throttle channel

3.2.4 Nonlinear simulations

After testing LQT with linear model, the performance of controller with nonlinear aircraft model is tested. Since the performance of LQT in linear simulations is very good, similar performance is expected at the trim points of aircraft.

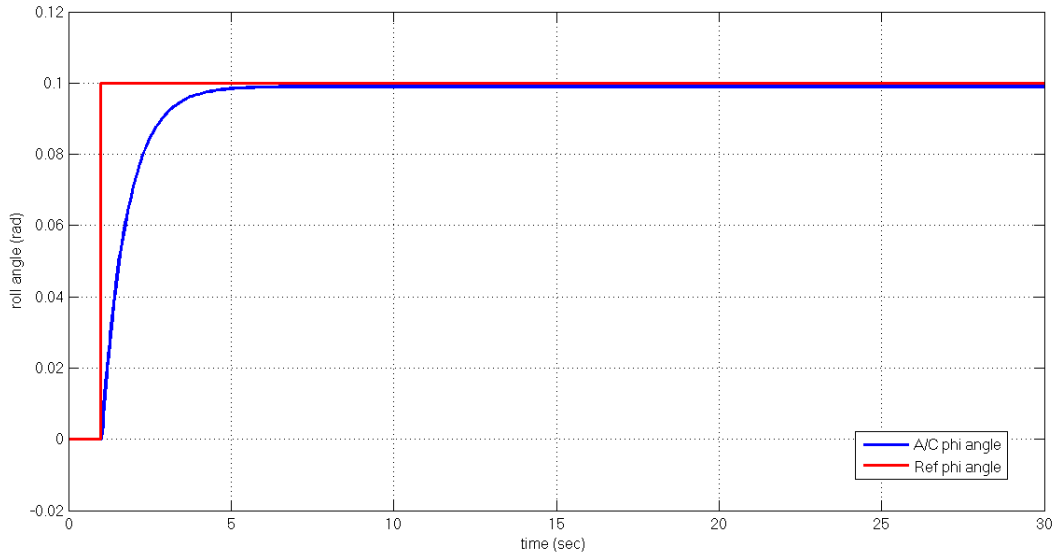

Figure 35 –Roll response of aircraft wrt step input from aileron channel

Given an initial altitude, it is requested from the aircraft to increase altitude to 45 meters, the response can be seen below, there is a small offset observed which may be a cause of small errors of initial value of states assumed.

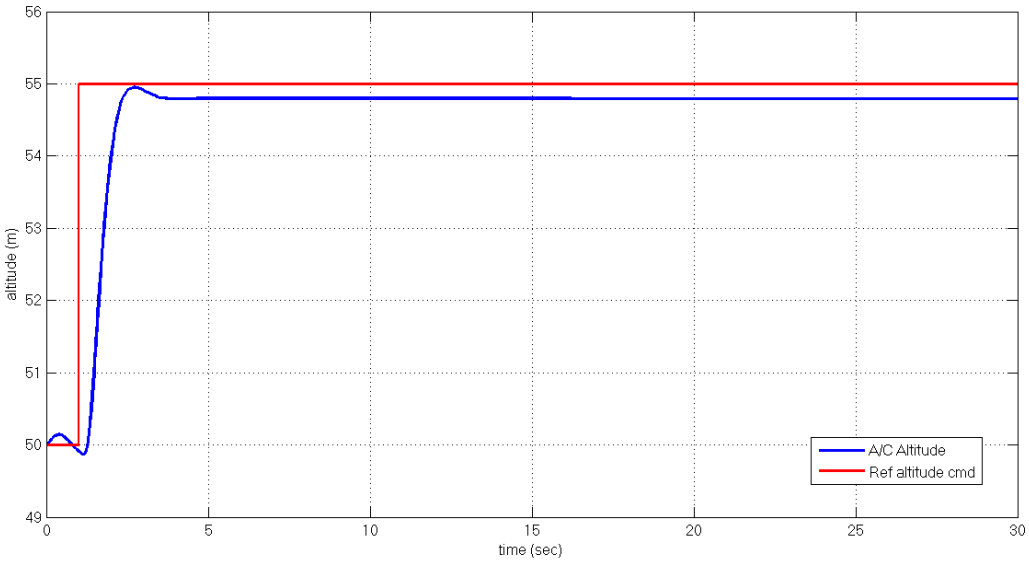

Figure 36 –Altitude response of aircraft wrt step input from elevator channel

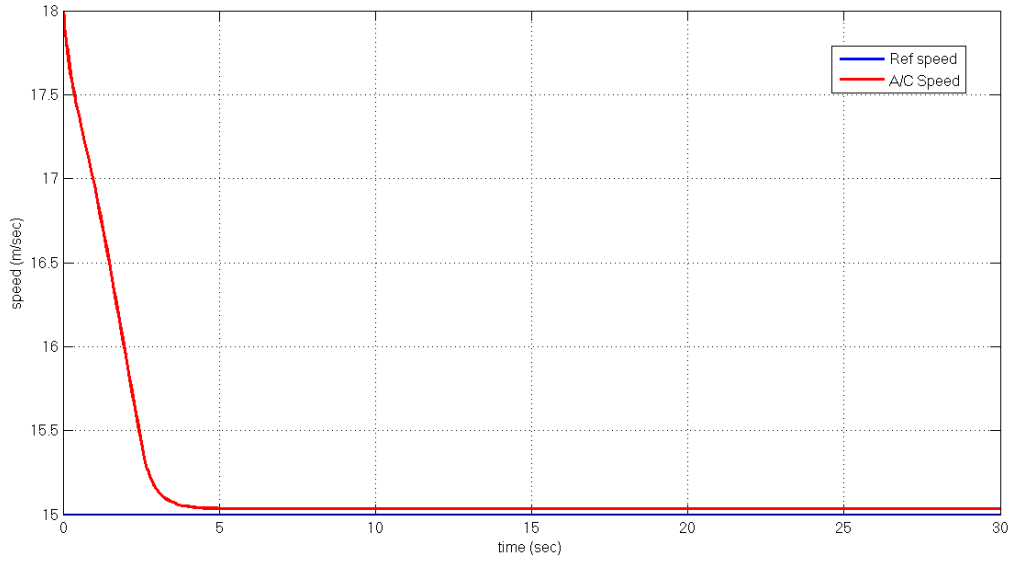

Figure 37 –Speed response of aircraft wrt step input from thrust channel

3.3 Sliding mode controller (SMC)

Sliding mode control is a unique technique obtained from variable structure control (VSC). It uses high speed switched feedback control which results in sliding mode [28]. Depending on the value of state at each time instant, the gains in feedback path are switched according to a user-defined rule. Purpose of this rule is to drive the trajectory of state onto a prespecified surface in the state space. This surface is called 'switching surface' and the resulting motion of state trajectory is 'sliding mode' [29].

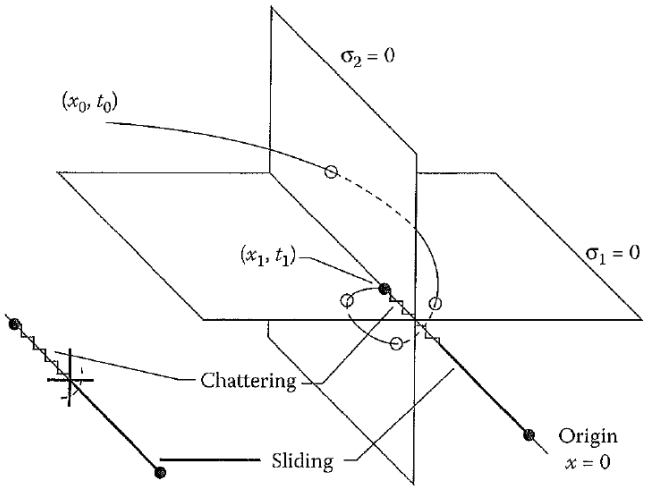

Figure 38 – Sliding mode on the intersection of two surfaces

Ideal sliding mode control is insensitive to parameter variations and disturbances, regardless of nonlinearity and uncertainty [30]. Although the method possesses good robustness characteristics, pure sliding mode control comes with, large control requirements and chattering. When the trajectory reaches the sliding surface, instead of having a sliding mode motion, it begins to drift away due to lack of synchronization between the actual switching time and controller's time to react [31]. Such motion can be seen in Fig. 39.

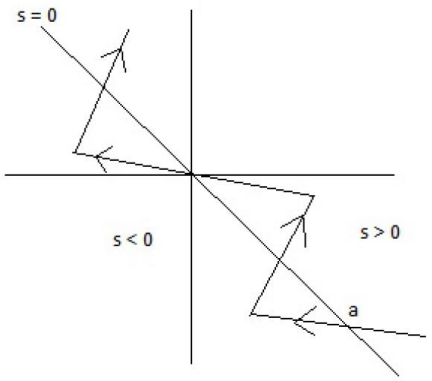

Figure 39 - Sliding mode with chattering due to delay in controller

Suppose the system in form given:

$$\dot{x}(t) = F(x,t,u) = f(x,t) + B(x,t)u(x,t) \tag{3.18}$$

Where $x(t) \in R^n$, $u(t) \in R^m$, $B(x,t) \in R^{n\times m}$ and each entry in $f(x,t)$ and $B(x,t)$ are continuous. We can separate this problem into two phases. First objective is to design a switching surface, so the plant state trajectory is bounded to the surface which has desired system dynamics. The second one is to design a switching control which will lead the state trajectory to switching surface and maintain it on the surface.

3.3.1 Regular form Approach

For a large class of systems, linear sliding modes are more convenient to design since its simple and clear. So the sliding mode controller design will be based on the nominal linear model of the system which is given by [32, 33]:

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{3.19}$$

$rank(B) = m$ and (A, B) is a controllable pair. We will define a linear sliding surface:

$$s(x) = Sx(t) \tag{3.20}$$

In order to maintain sliding motion on this surface, equivalent control is applied. In this approach the time derivative of vector $s(x)$ is set to zero and the resultant algebraic equations are solved to find control vector. To adapt our system of equations to regular form which is governed by:

$$\dot{z_1}(t) = A_{11}z_1(t) + A_{12}z_2(t) \tag{3.21}$$

$$\dot{z_2}(t) = A_{21}z_1(t) + A_{22}z_2(t) + B_2u(t) \tag{3.22}$$

And

$$s(t) = S_1z_1(t) + S_2z_2(t) \tag{3.23}$$

Define the transformation matrix T_r:

$$z(t) = T_rx(t) \tag{3.24}$$

Where the equations (3.21) and (3.22) can be derived from original pair (A, B) from

$$T_r A T_r^T = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \tag{3.25}$$

$$T_r B = \begin{bmatrix} 0 \\ B_2 \end{bmatrix} \tag{3.26}$$

And

$$S T_r^T = [S_1 \quad S_2] \tag{3.27}$$

$$S_1 z_1(t) + S_2 z_2(t) = 0 \tag{3.28}$$

During sliding motion $S(t)$ will be identically zero, so S will be selected that matrix product SB is nonsingular. This assumption implies that S_2 is nonsingular so the condition for sliding mode given by (3.28) can be solved for z_2 as

$$\begin{aligned} z_2(t) &= -S_2^{-1} S_1 z_1(t) \\ &= -M z_1(t) \end{aligned} \tag{3.29}$$

where $M \in \mathbb{R}^{m \times (n-m)}$ and shown as

$$M = S_2^{-1} S_1 \tag{3.30}$$

Using equations (3.21) and (3.29), the sliding mode is now governed by this formula

$$\dot{z}_1(t) = A_{11} z_1(t) + A_{12} z_2(t) \tag{3.31}$$

$$z_2(t) = -M z_1(t) \tag{3.32}$$

$$\dot{z}_1 = (A_{11} - A_{12}M)z_1(t) \tag{3.33}$$

So the hyperpalne matrix S can be determined from M by letting $S_2 = I_m$, leads to

$$ST_r^T = [M \quad I_m] \tag{3.34}$$

By this way, the possibility of numerical errors are minimized by calculation of S from M.

3.3.2 Quadratic minimization

In order to design switching hyper plane, a method proposed by Utkin & Young will be discussed. For a model following control system, this method enables to establish desirable weights to the states of the system. Consider the problem of minimizing the following performance index:

$$J = \frac{1}{2}\int_{t_s}^{\infty} x(t)^T Q x(t) dt \tag{3.35}$$

Where Q is a symmetric and positive definite matrix. t_s is the time the state trajectory intercepts sliding surface. So the aim is to minimize this cost function in order to obtain switching function matrix. For compatibly to our problem, the matrix Q is transformed into z coordinates.

$$T_r Q T_r^T = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \tag{3.36}$$

Where, $Q_{21} = Q_{21}^T$, so the equation (3.35) can be expressed as in z coordinates

$$J = \frac{1}{2}\int_{t_s}^{\infty} z_1^T Q_{11} z_1 + 2z_1^T Q_{12} z_2 + z_2^T Q_{22} z_2 dt \tag{3.37}$$

In order to solve this problem the equation (3.37) is restated as a LQR problem. Where $z_1$determines the the system dynamics and z_2 serves as an effective control input to the system. After eliminating cross terms as $2z_1^T Q_{12} z_2$ the cost function becomes:

$$J = \frac{1}{2}\int_{t_s}^{\infty} z_1^T (Q_{11} - Q_{12}Q_{22}^{-1}Q_{21}) z_1 + (z_2 + Q_{22}^{-1}Q_{21}z_1)^T Q_{22}(z_2 + Q_{22}^{-1}Q_{21}z_1) dt \tag{3.38}$$

Define

$$\hat{Q} = Q_{11} - Q_{12}Q_{22}^{-1}Q_{21} \tag{3.39}$$

$$v = z_2 + Q_{22}^{-1}Q_{21}z_1 \tag{3.40}$$

Equation (3.38) can be written as

$$J = \frac{1}{2}\int_{t_s}^{\infty} z_1^T \hat{Q} z_1 + v^T Q_{22} v dt \tag{3.41}$$

The original constraint equation was

$$\dot{z}_1(t) = A_{11}z_1(t) + A_{12}z_2(t) \tag{3.42}$$

After elimination of z_2 contribution using equation (3.40), new constraint equation becomes

$$\dot{z}_1(t) = \hat{A}z_1(t) + A_{12}v(t) \tag{3.43}$$

Where

$$\hat{A} = A_{11} - A_{12}Q_{22}^{-1}Q_{21} \tag{3.44}$$

After these modifications, the problem becomes to minimize equation (3.41) with respect to equation (3.41) which is a standard LQR problem. If Q is positive definite, $Q_{22} > 0$, so Q_{22}^{-1} exists and so that $\hat{Q} > 0$. And the controllability of (A,B) pair ensures that the pair ($A_{11}A_{12}$) is controllable as well as ($\hat{A}, A_{12}$). So the Algebraic Riccati equation defined by equations (3.41) and (3.43), has a guaranteed positive definite solution P_1 which is given by

$$P_1\hat{A} + \hat{A}^T P_1 - P_1 A_{12} Q_{22}^{-1} A_{12}^T P_1 + \hat{Q} = 0 \tag{3.45}$$

Since the optimal v minimizing the equation (3.41) is

$$v = -Q_{22}^{-1}A_{12}^T P_1 z_1 \tag{3.46}$$

Applying this expression to equation (3.40) brings

$$z_2 = -Q_{22}^{-1}(A_{12}^T P_1 + Q_{21})z_1 \tag{3.47}$$

After comparing this term with equation (3.29) M can be found as

$$M = Q_{22}^{-1}(A_{12}^{T}P_1 + Q_{21}) \tag{3.48}$$

From M, the switching function matrix can be easily found as inserting equation (3.48) into equation (3.34).

The sliding mode controller is designed for linearized aircraft model which is trimmed at 50m height and 18ms speed values. Roll, altitude and speed controllers are designed, where all controllers consists of four states. The equivalent control for system is given by

$$u_{eq}(t) = -(SB)^{-1}\big(SAx(t)\big) \tag{3.49}$$

This signal is not applied to the system alone since $u_{eq}(t)$ will keep the system in sliding mode if there is no disturbances and un-modelled dynamics. So a discontinuous control function is also added to equivalent control. The control function becomes

$$u(x,t) = u_{eq} + \rho sgn(S) \tag{3.50}$$

3.3.3 Roll controller (SMC)

The states $[\beta; p; r; \emptyset]$ are selected for roll attitude controller. First the linearized aircraft is analyzed. The aileron command is input and the output is selected as phi angle of the aircraft. The sliding surface is selected as

$$S = S_1\beta + S_2p + S_3r + S_4(\emptyset - \emptyset_d) \tag{3.51}$$

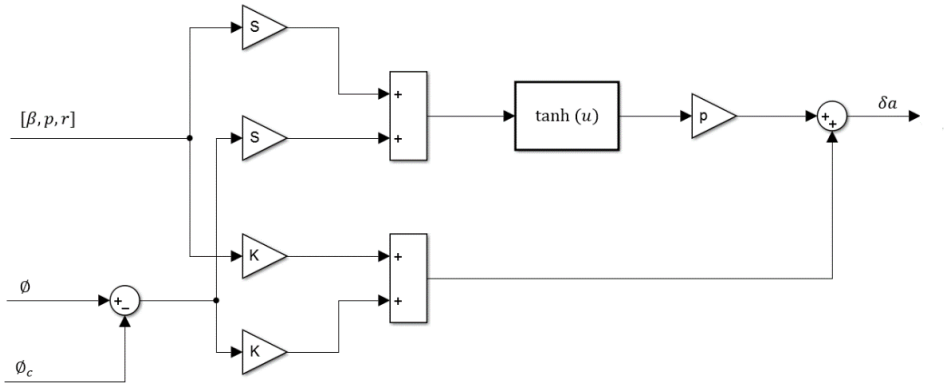

Figure 40 – Block diagram of SMC roll controller

For simulation, step input is given to the linear model of aircraft and response of system has observed.

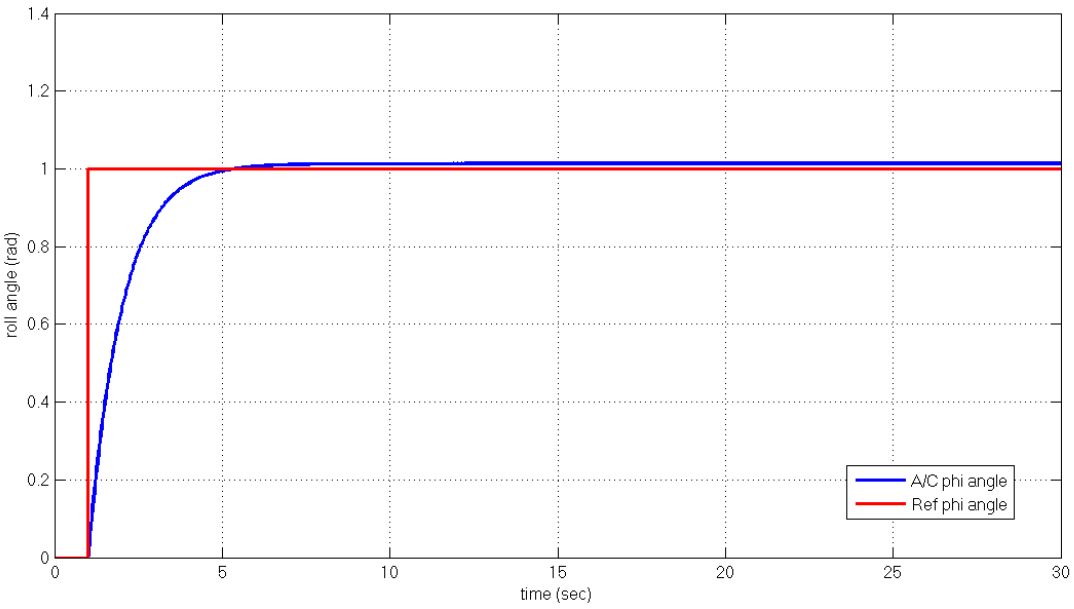

Figure 41 – Roll response of aircraft wrt step input from aileron (linear model)

3.3.4 Altitude controller (SMC)

The states $[\alpha; q; \theta; h]$ are selected for altitude controller. First the linearized aircraft is analyzed. The elevator command is input and the output is selected as height of the aircraft. The sliding surface is selected as

$$S = S_1\alpha + S_2 q + S_3\theta + S_4(h - h_d) \tag{3.52}$$

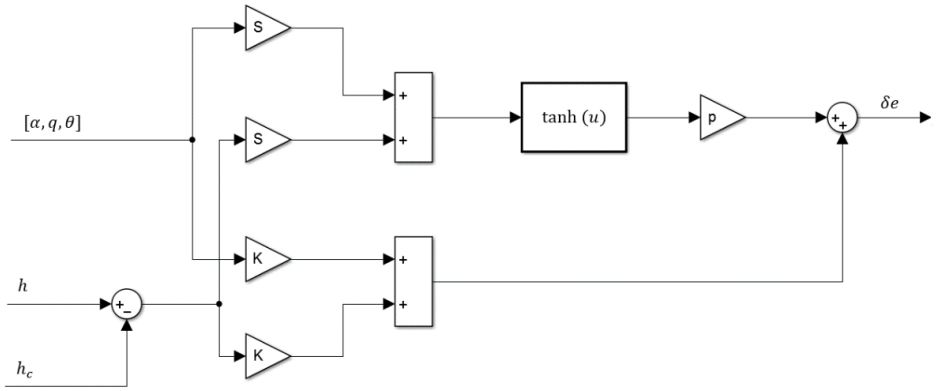

Figure 42 – Block diagram of SMC altitude controller

For simulation, step input is given to the linear model of aircraft and response of system is observed.

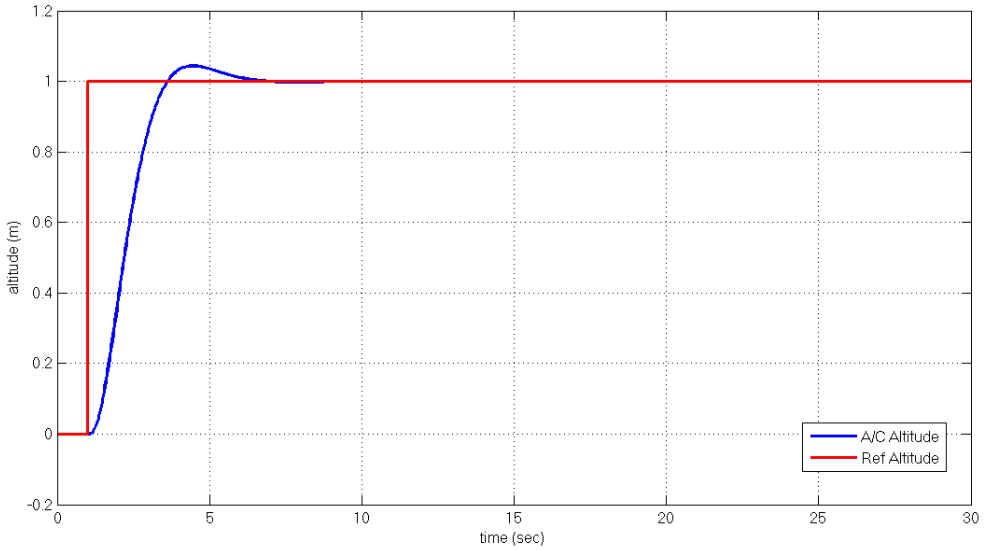

Figure 43 – Altitude response of aircraft wrt step input from elevator channel (linear model)

3.3.5 Speed controller (SMC)

The states $[v; \alpha; q; \theta]$ are selected for speed controller. First the linearized aircraft is analyzed. The elevator command is input and the output is selected as height of the aircraft. The sliding surface is selected as

$$S = S_1(v - v_d) + S_2\alpha + S_3 q + S_4\theta \tag{3.53}$$

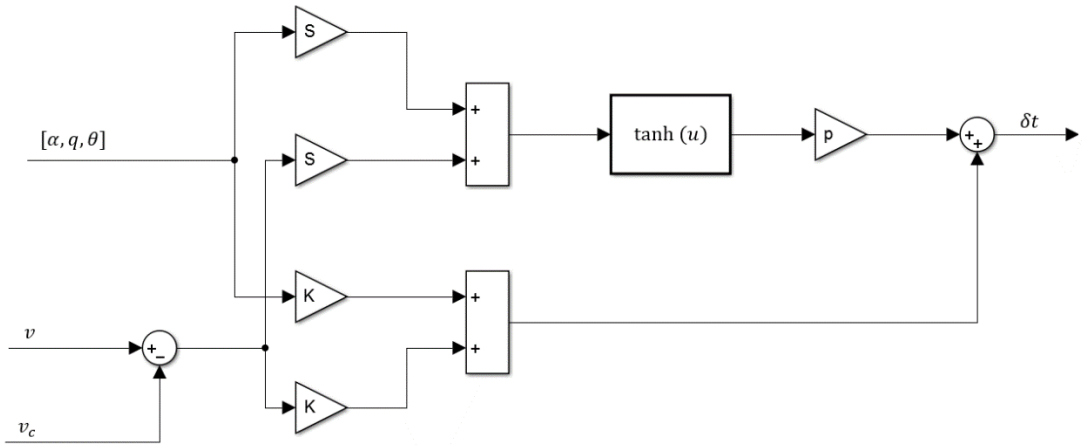

Figure 44 – Block diagram of SMC speed controller

To drive the S to zero which is defined in equation 3.53, the speed error should be zero. The defined sliding surface will track the desired speed value by producing throttle command. The response of linear model with respect to step input is given in figure 45.

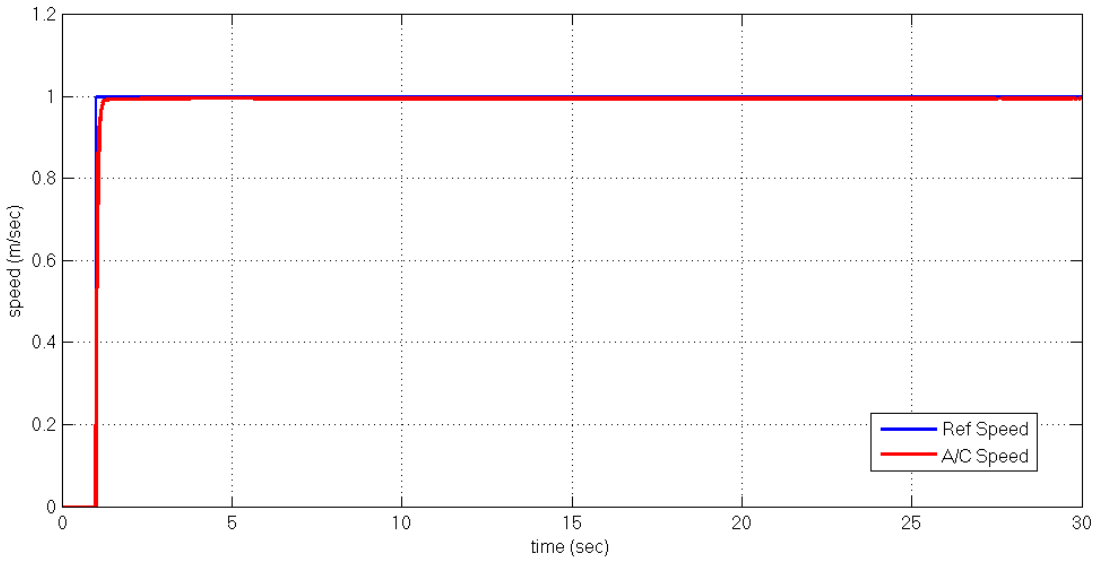

Figure 45 – Speed response of aircraft wrt step input from throttle (linear model)

3.3.6 Nonlinear Simulations

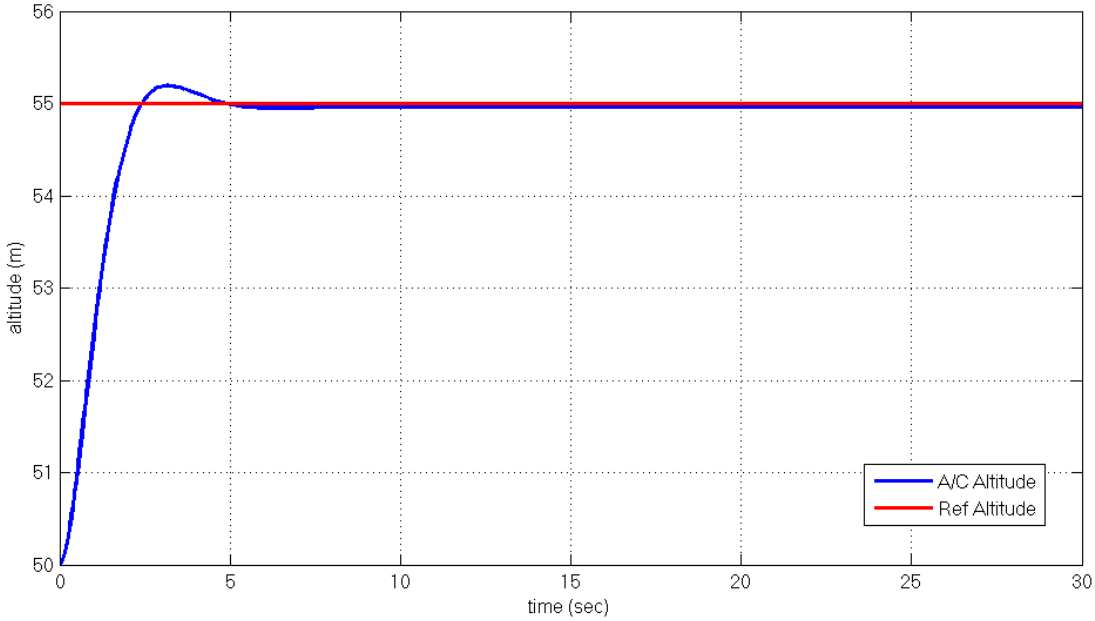

Figure 46 – Altitude response of aircraft wrt step input from elevator channel (nonlinear model)

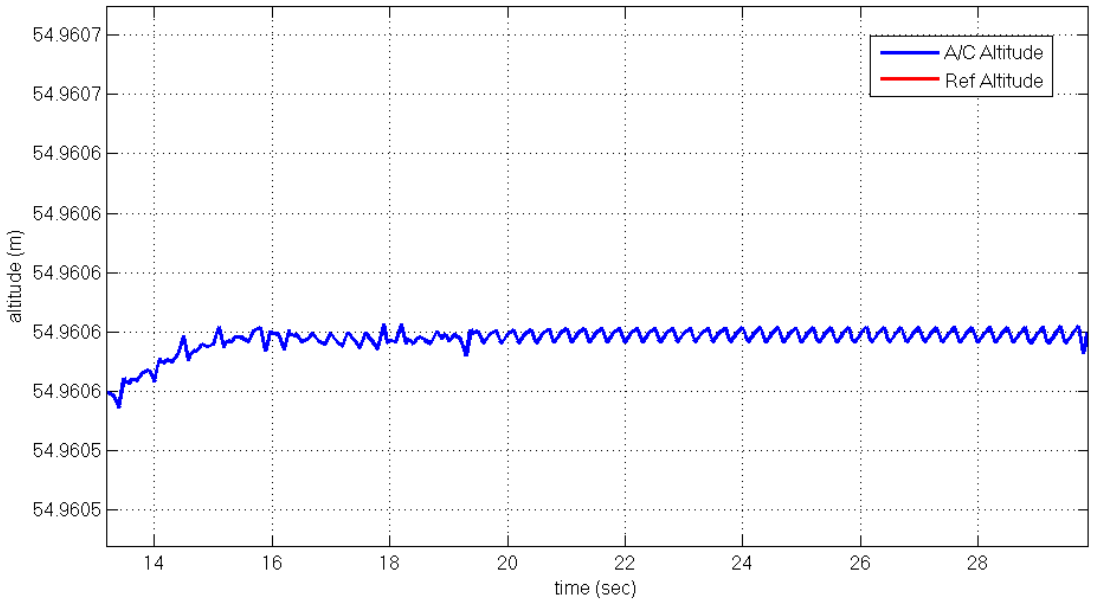

Figure 47 – Chattering behavior of altitude controller wrt step input from elevator channel (nonlinear model)

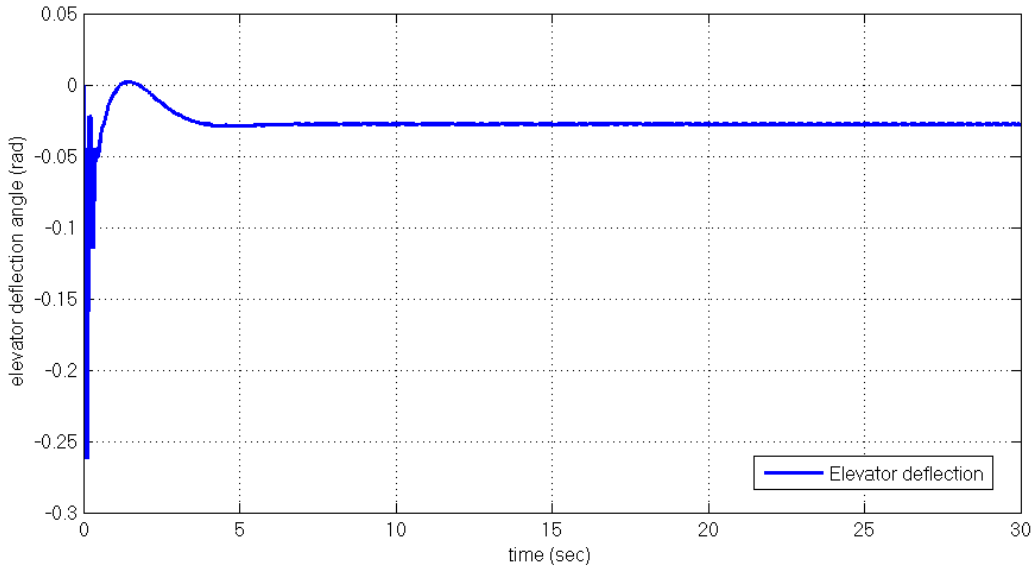

Figure 48 – Elevator deflection (nonlinear model)

While performing a level flight at 50m altitude with cruise speed 18m/s, the reference altitude command is updated as 55m. Generated elevator command and the altitude response plots are illustrated in figures 48 and 46. After the reaching phase the system slides through the sliding surface which is given in equation (3.52). The chattering behavior can be seen in figure 47.

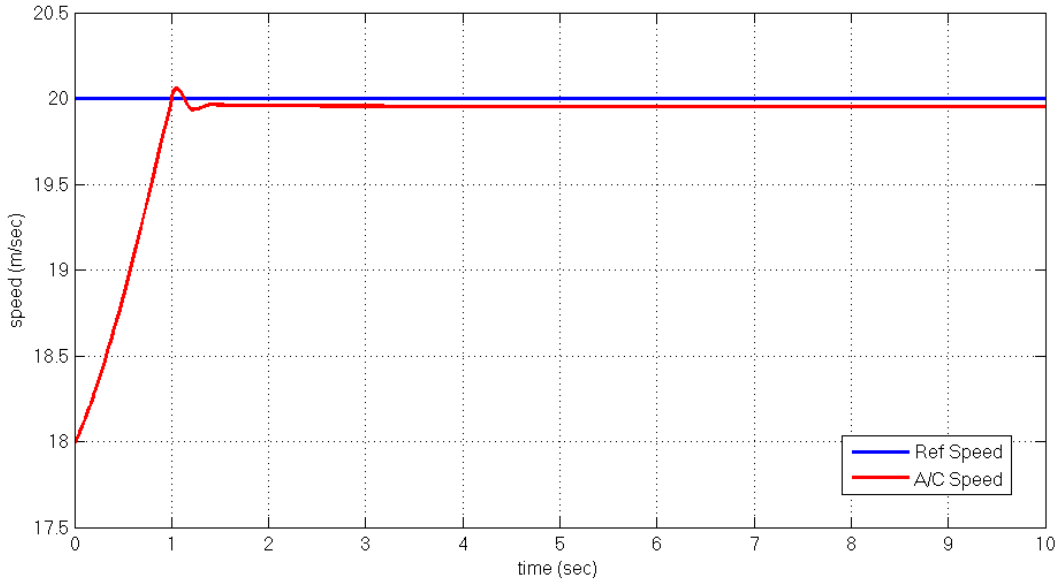

Figure 49 – Speed response of aircraft wrt step input from throttle (nonlinear model)

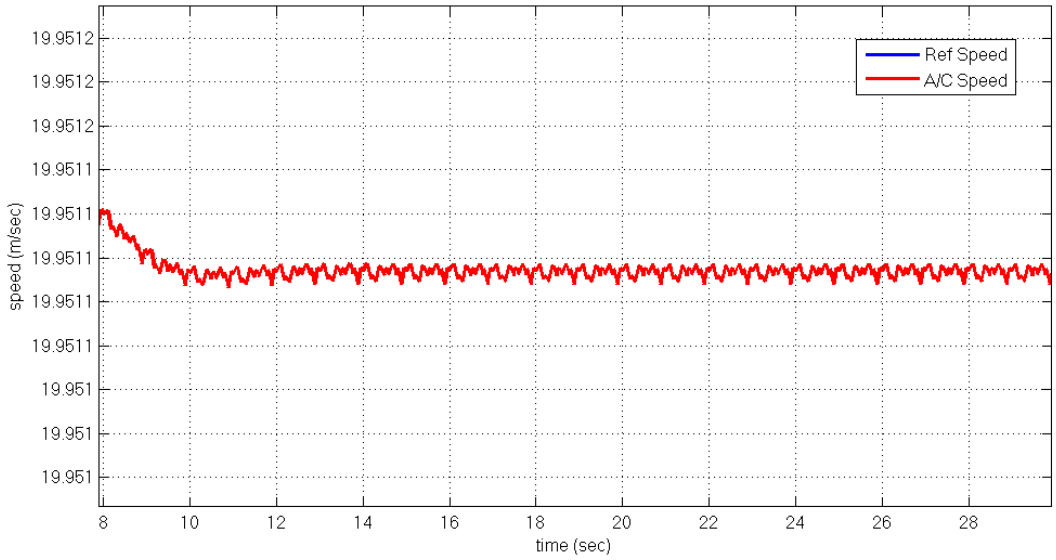

Figure 50 - Chattering behavior of speed controller wrt step input from elevator channel (nonlinear model)

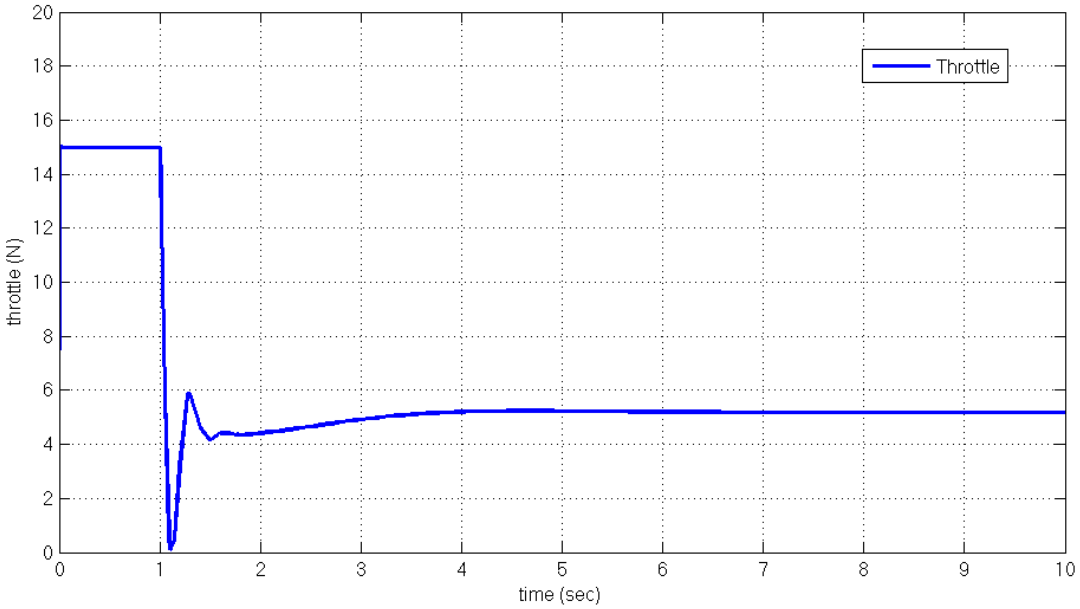

Figure 51 – Throttle command (nonlinear model)

While performing a level flight at 50m altitude with cruise speed 18m/s, the reference speed command is updated as 20m/s. Throttle command and the speed response plots are illustrated in figures 51 and 49. After the reaching phase the system slides through the sliding surface which is given in equation (3.53). The chattering behavior can be seen in figure 50.

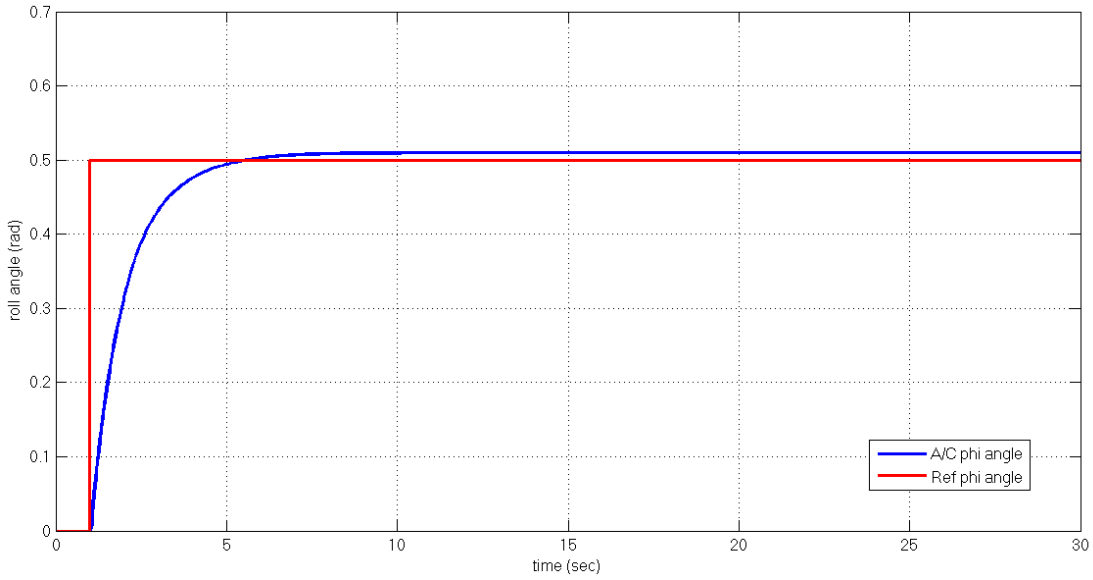

Figure 52 – Roll response of aircraft wrt step input from aileron channel (nonlinear model)

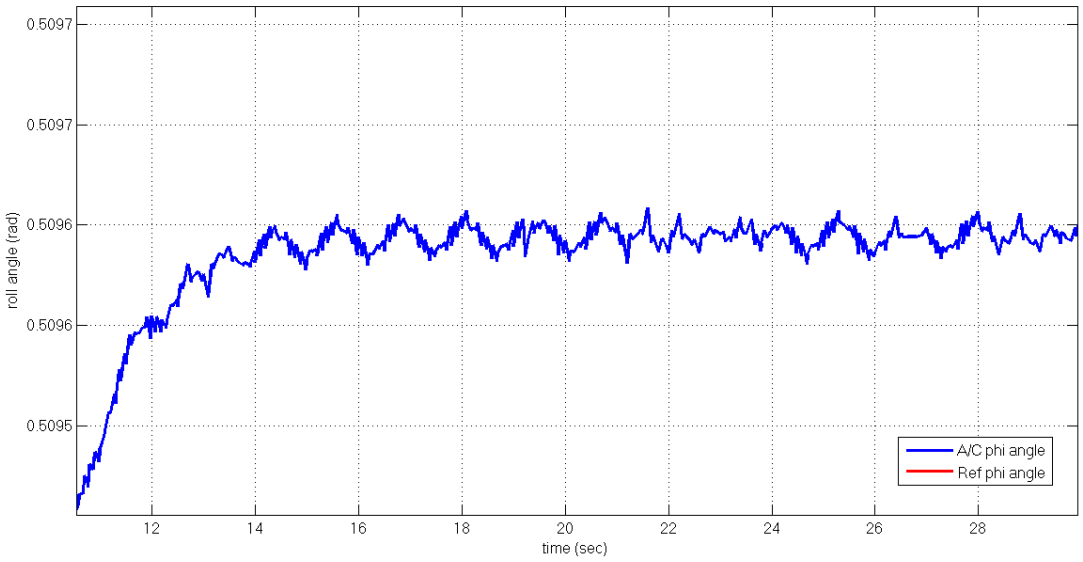

Figure 53 - Chattering behavior of roll angle controller wrt step input from elevator channel (nonlinear model)

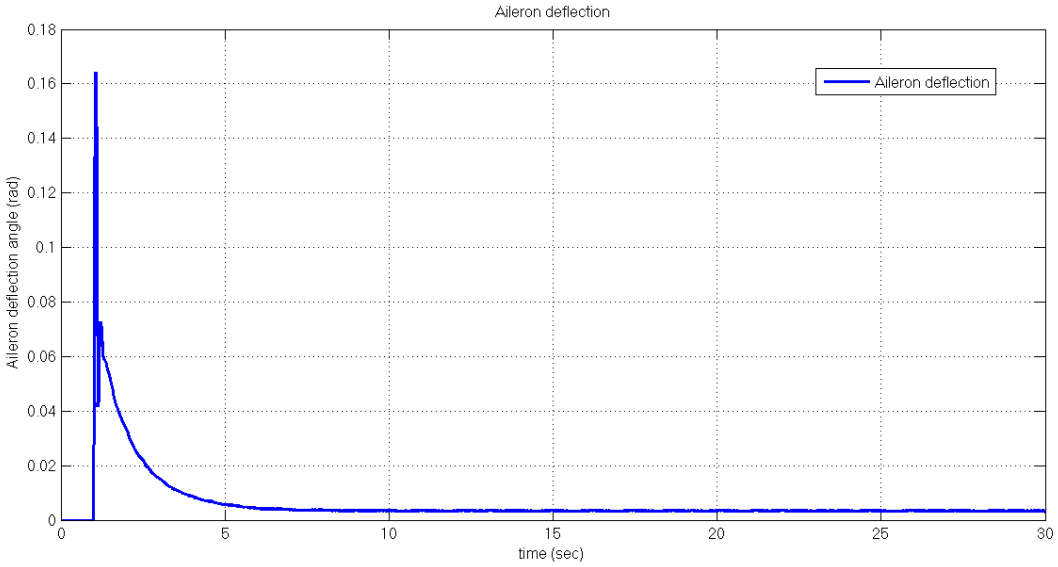

Figure 54 - Aileron deflection (nonlinear model)

At level flight at 50m altitude with cruise speed 18m/s, the reference roll command is updated as 0.5 radian. Aileron command and the roll attitude plots are illustrated in figures 55 and 52. After the reaching phase the system slides through the sliding surface which is given in equation (3.51). The chattering behavior can be seen in figure 53.

3.4 Sliding mode guidance

For landing phase, a second type of sliding mode altitude controller has been designed. Which contains states $[v, \alpha, q, \theta, h]$ and controls height of the aircraft for a given trajectory. Controller uses elevator channel as output. The block diagram of such controller is given in figure 55:

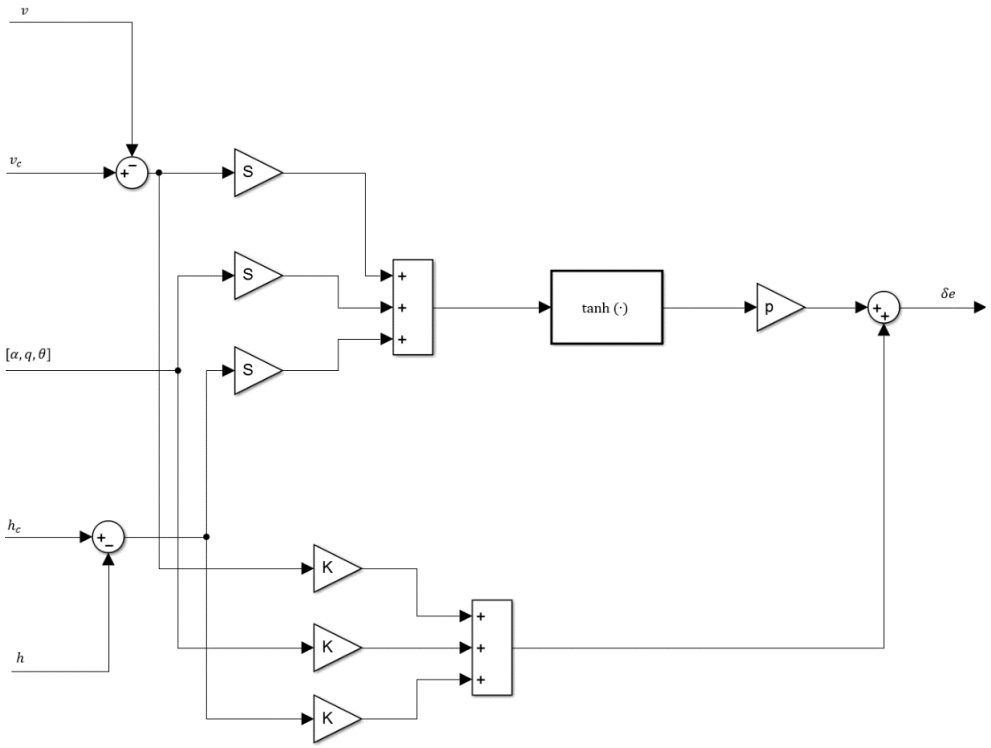

Figure 55 – 2nd Sliding Mode Controller block diagram

Where the sliding surface is:

$$S = S_1(v - v_d) + S_2\alpha + S_3 q + S_4\theta + S_5(h - h_d) \tag{3.54}$$

Some landing results, obtained from optimal landing trajectory which will be discussed a at guidance design, are gathered. Taking initial altitude of 50m we are interested in landing on a runway which is 780 m away. The speed of the aircraft is 12ms. The simulation results are given as:

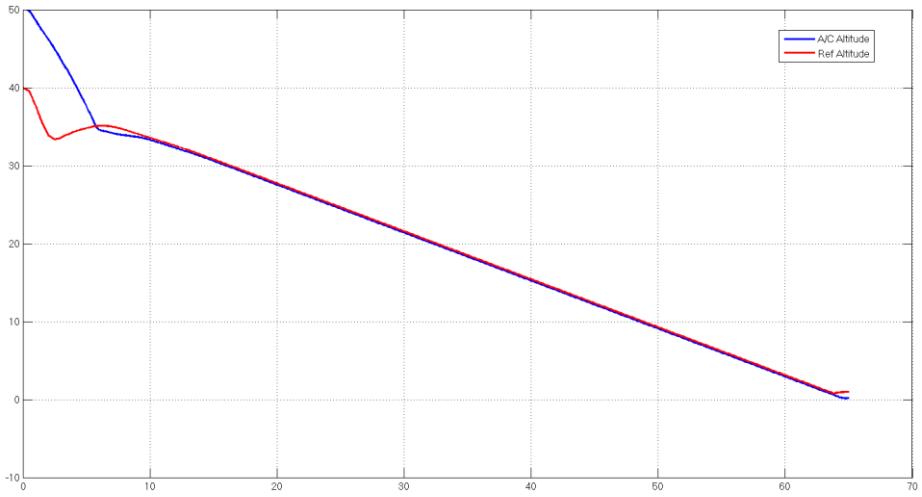

Figure 56 – Altitude plot of SMC guidance controller

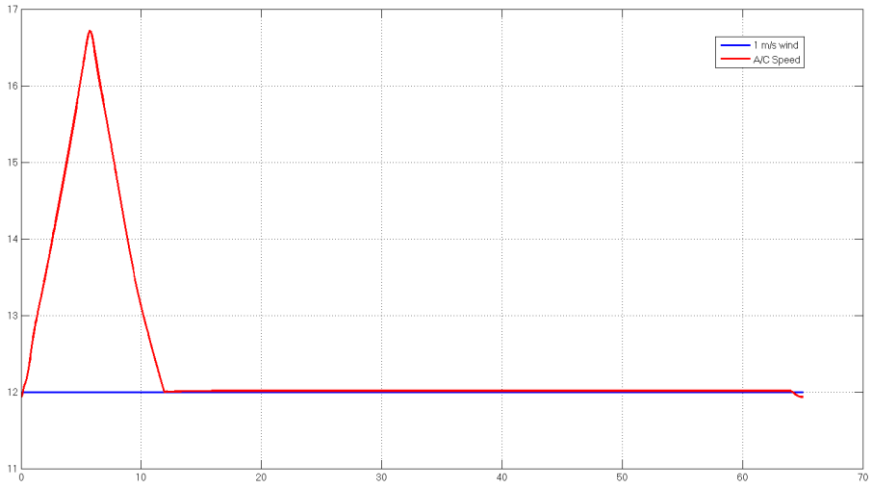

Figure 57 – Speed plot of SMC guidance controller

Using five states in SMC, has very robust results. Since we can decide the importance of states, by selecting values for weighting matrix ‘Q’, the behavior of the aircraft can be adjusted easily. For example, quick altitude drops results in a speed gain, by selecting right weighting parameters the trade of between altitude error and speed loss can be set.

CHAPTER 4

GUIDANCE DESIGN

It is demanded that the Telemaster UAV to fly and land autonomously. For this purpose, way points, which generates the necessary reference altitude, speed and heading commands, are implemented. MATLAB's state flow diagrams were used to construct these waypoints. For landing phase of aircraft, depending on the properties of runway, there are some predefined parameters and limitations such as glide path angle, altitude constraints, speed constraints and lateral distance tolerances. In our case, glide path angle (GPA) was selected as -3^{o} , which normally can vary between -2.5^{o} and -3.5^{o} . The maximum landing altitude was selected as approximately 50m. The landing speed was approximately %20-%30 higher than the stall speed, which was selected as 12m/s.

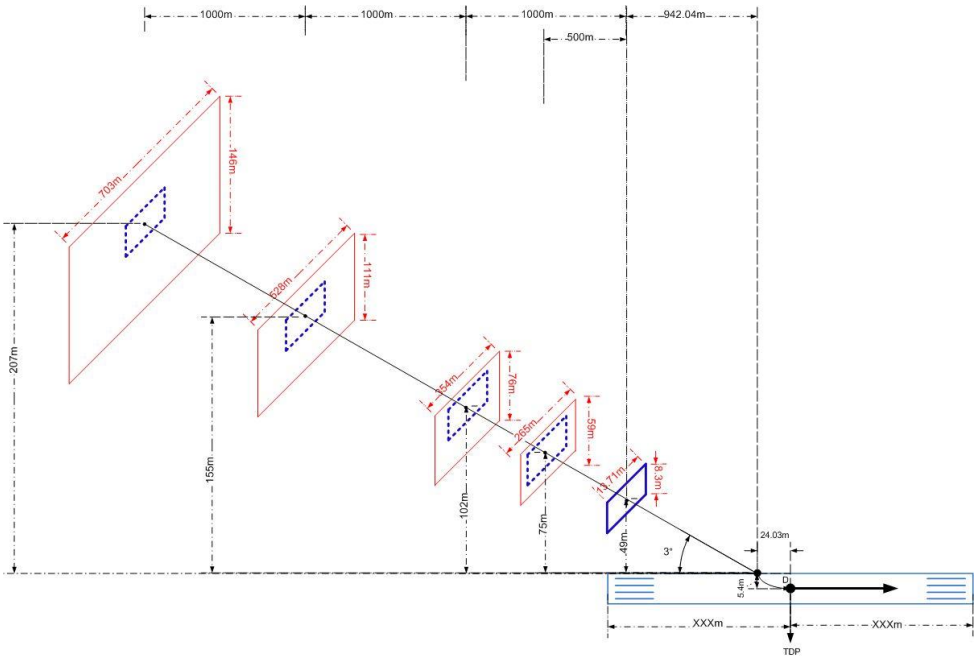

Figure 58 – Ideal landing path

For autonomous landing, some reference altitude and corresponding distance to touchdown points were selected. Designed autopilots in chapter 3 are used to control the aircraft to stay at desired states. Since Telemaster UAV is a small aerial vehicle, these altitude points are 40m, 20m, and 10m. Using these points, ideal landing paths are generated with an optimization algorithm, taking into account of their tolerances at these references. For the landing corridor tolerances, standard ILS area coverage is selected which is lateral tolerance angle value as 5.0 degrees and the low lateral tolerance angle value is 0.41 degrees. The chosen high longitudinal tolerance angles are +0.5 degrees for upward and -1.5 degrees for downward given in figures 59 and 60.

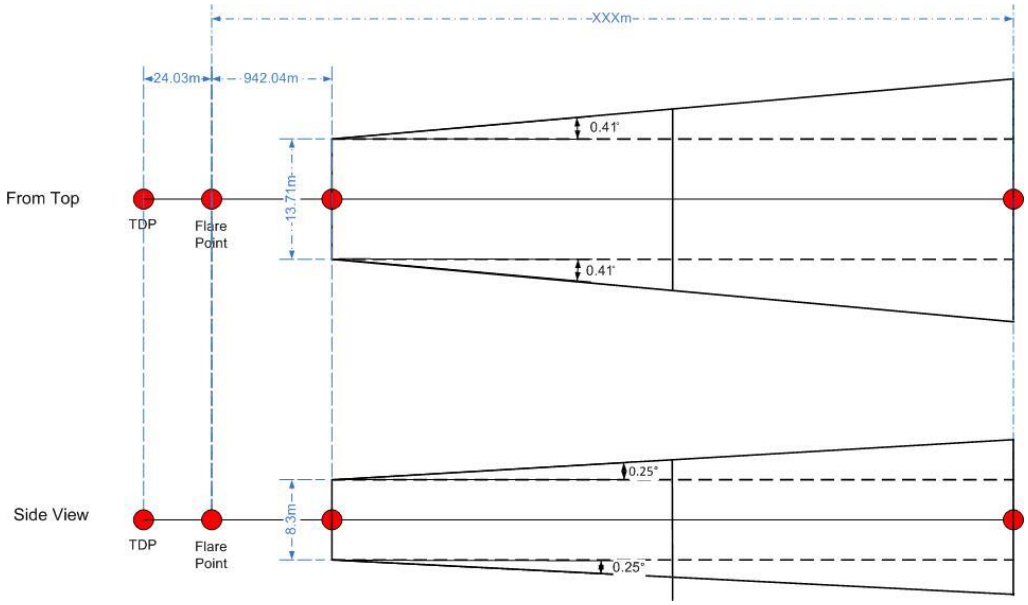

Figure 59 –Minimum tolerances of landing path

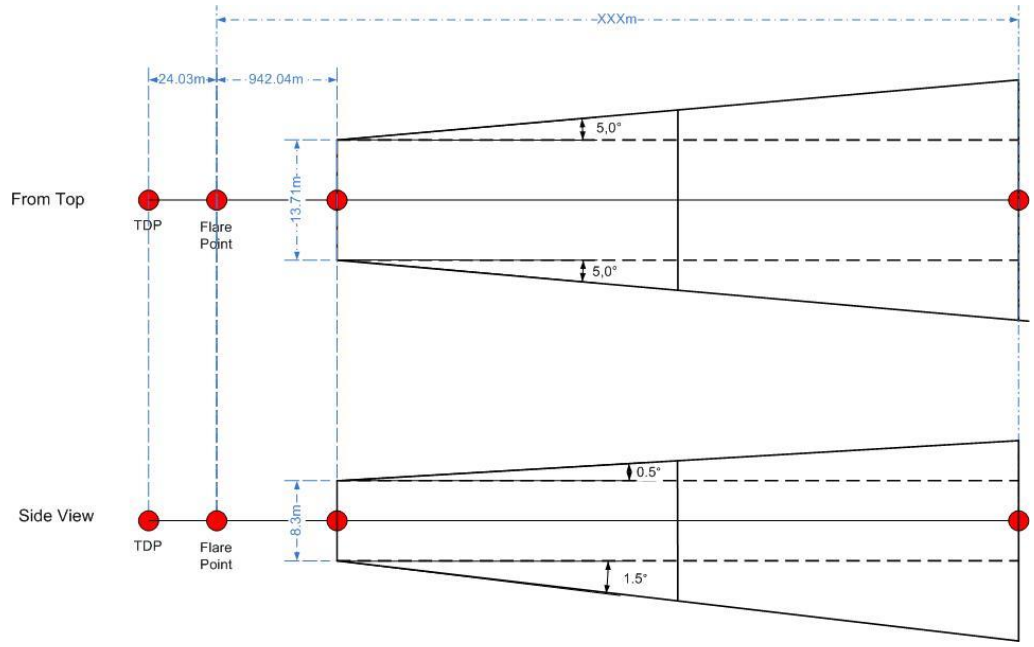

Figure 60 –Maximum tolerances of landing path

While the aircraft is descending on glide path, getting closer to the touchdown point, it has to perform a flare maneuver in order to have a safe landing. This means, the aircraft needs to raise its nose upwards, slowing the descent rate of aircraft. Resulting an exponential path which can be seen in Fig. 61.

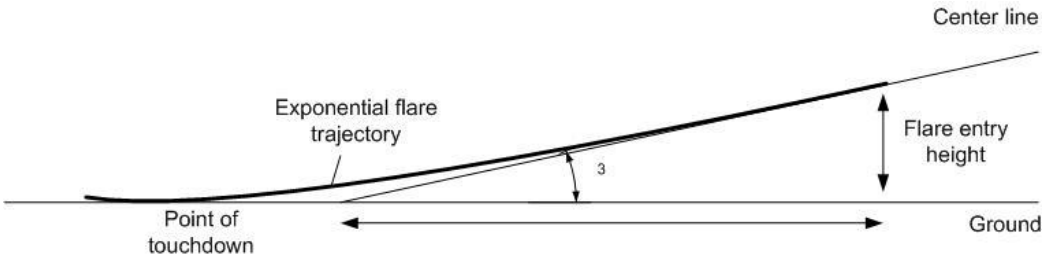

Figure 61 – Flare maneuver (side view)

A guidance block which contains reference altitude, speed and heading values for all waypoints is implemented for level and landing phases of flight. The block diagram can be seen in Fig. 62.

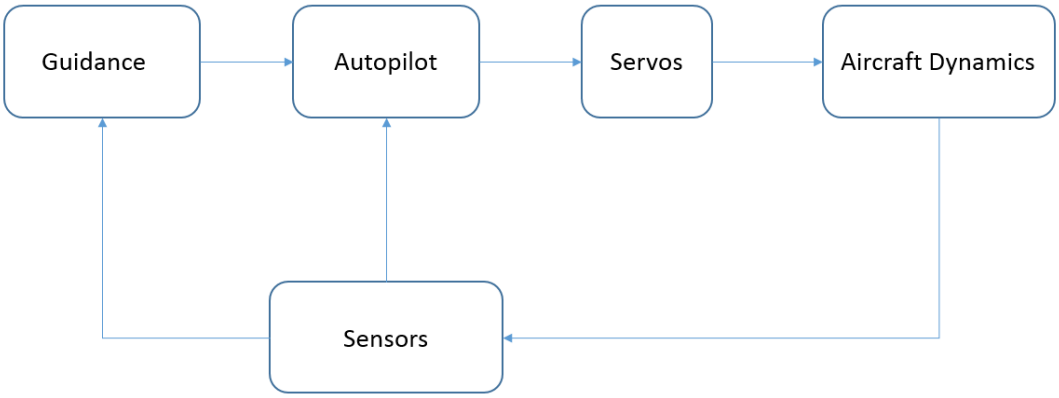

Figure 62 – Autonomous flight block diagram

4.1 Optimal landing part design

Using the selected reference altitudes, some optimal landing paths are generated using steepest decent algorithm. A cost function, containing error of these reference states as well as control signal values has constructed.

$$J = \sum_{i=0}^{t_f} \big(x(i) - x_d(i)\big)^T Q\big(x(i) - x_d(i)\big) + \sum_{0}^{t_f} u^T(i) R\, u(i) \tag{4.1}$$

Where Q and R are, symmetric weighting matrices. x_d contains the predefined states which are speed, altitude and heading, at landing phase of aircraft and input vector u includes throttle, elevator and aileron signals respectively. Optimization algorithm runs on a m-file and the required gradients are obtained by 'sim' function command of MATLAB. This algorithm will give us some reference commands for predefined states, to land the aircraft using an ideal glide path. After generating different flight paths for several starting positions of landing, the generated paths will be interpolated for the use of any initial position of aircraft.

4.1.1 Steepest decent algorithm

Steepest decent is a very well-known method for minimizing a function of several variables. Let f have continuous first partial derivatives, the gradient $\nabla f(x)$ will frequently needed. The method is defined by the iterative algorithm:

$$x_{k+1} = x_k - \alpha_k g_k \tag{4.2}$$

Where $g_k = \nabla f(x_k)^T$ and α_k is a nonnegative scalar minimizing $f(x_k - \alpha g_k)$. Choice of α is important, since too large α can cause to algorithm diverge and a very small one will be very slow. There are many ways to find optimal $\boldsymbol{\alpha}$ like, polynomial fit methods and region elimination methods. Since we are dealing with numerical optimization, dichotomous search, a simple one dimensional search method have used [34].

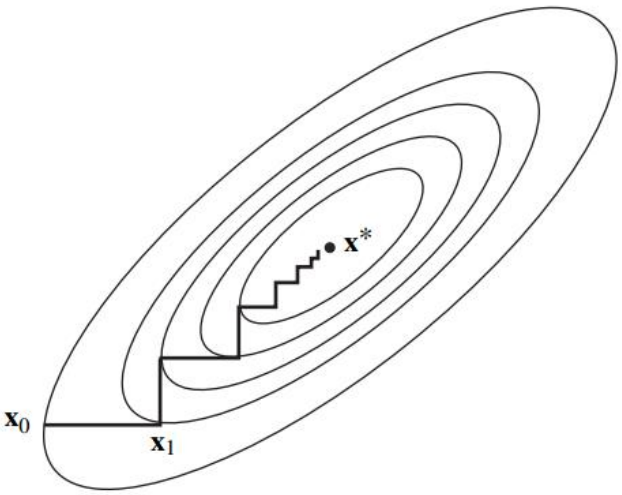

Figure 63 – Steepest decent converging algorithm

The flowchart of the steepest decent algorithm can be given as:

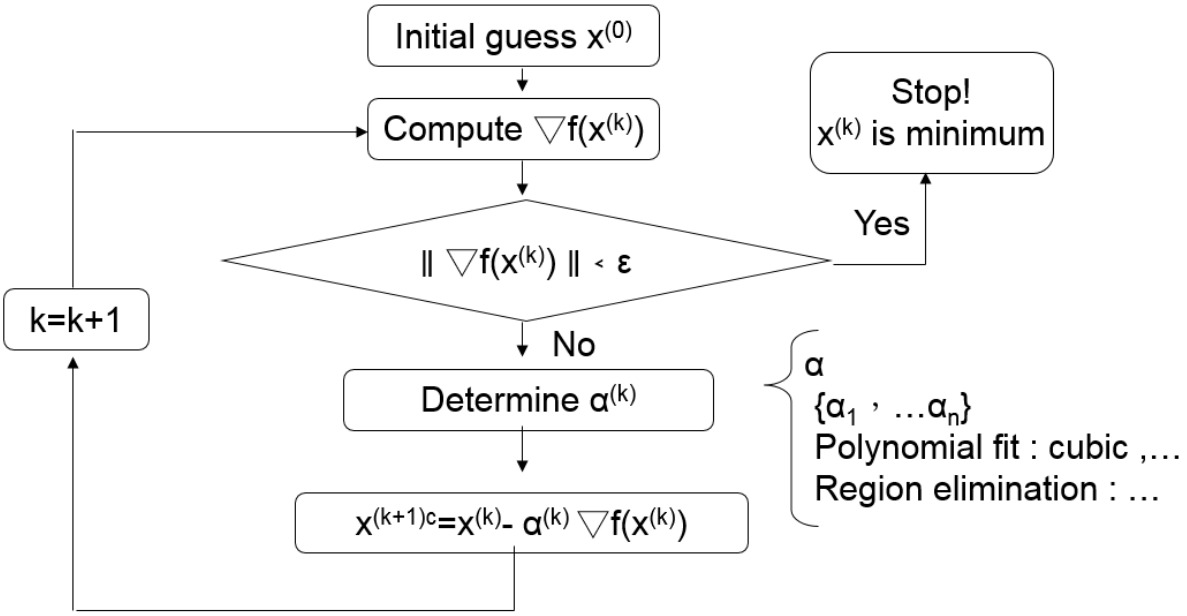

Figure 64 – Flowchart of steepest decent algorithm

In this thesis, steepest decent algorithm tries to find optimal command vectors, which make the aircraft to follow predefined state values. These predefined states are altitude, speed and heading.

Dichotomous search

It maybe algebraically impossible to solve for minimum using calculus. In nonlinear optimization problems with a single independent variable, we can approximate the solution. In our case, consider, α is in the region [a, b], divide this region into two overlapping intervals [a,x_1] and [x_2, b]. Then determine the subinterval where the optimal solution lies and us that subinterval to continue the search. Which can be shown as:

1. Calculate the following evaluation points:

$$b = \frac{\alpha_{min} + \alpha_{max}}{2} - \epsilon \qquad c = \frac{\alpha_{min} + \alpha_{max}}{2} + \epsilon \tag{4.3}$$

2. If $f(b) < f(c)$, set $\alpha_{min} = c$
   Otherwise, set $\alpha_{min} = b$
3. Repeat until convergence

Using optimization algorithm created in a MATLAB m-file, some optimal landing paths for reference altitudes which are 40m, 20m and 10m with perturbations are found. The distances to the runways are 780m, 340m and 170m respectively. Optimal landing path found for 40m altitude and 780m distance is obtained as follows:

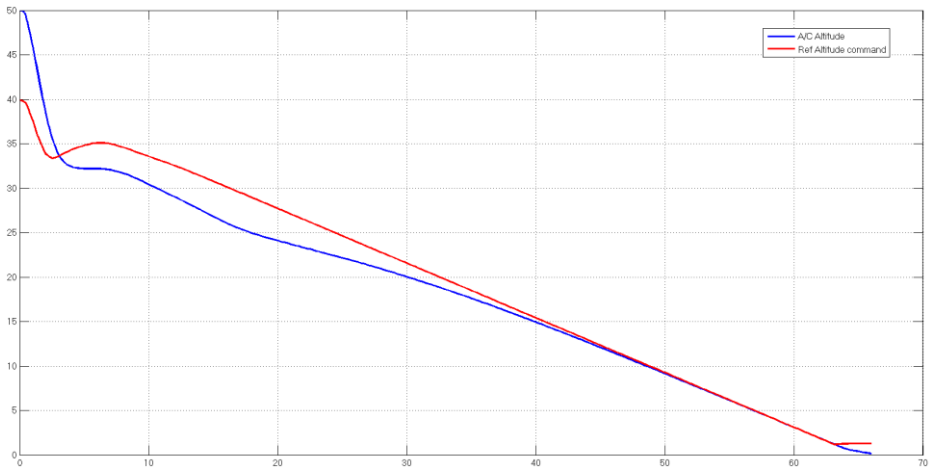

Figure 65 – Optimum landing path for initial altitude of 50m

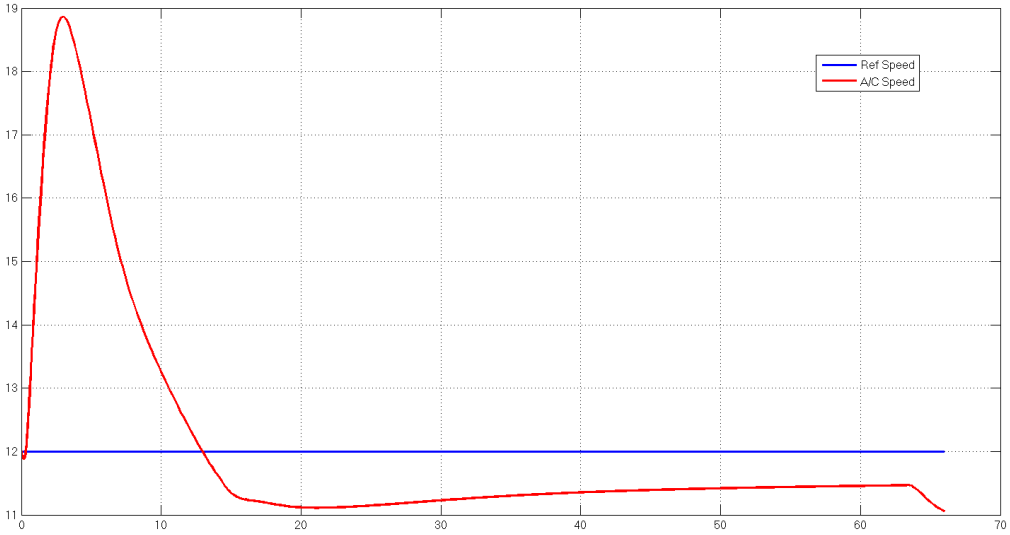

Figure 66 – Speed response of aircraft (initial altitude 50m)

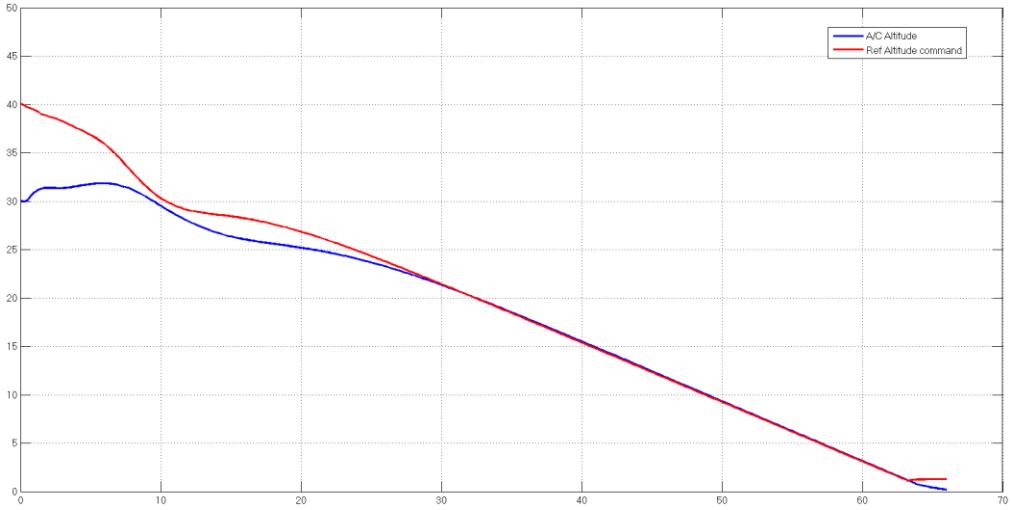

Figure 67 – Optimum landing path for initial altitude of 30m

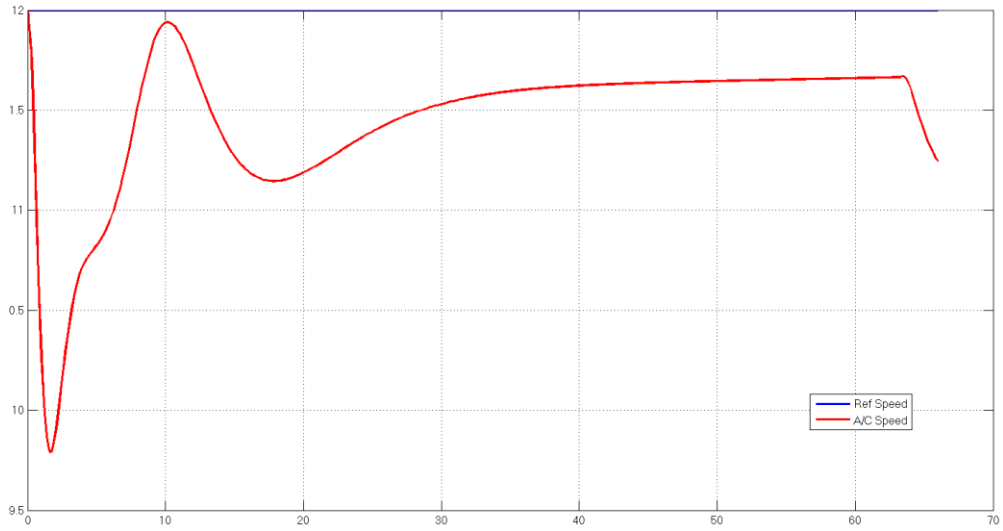

Figure 68 – Speed response of aircraft (initial altitude 30m)

Since it's required that the aircraft to find an optimal path at some arbitrary initial position, the gathered landing paths from optimization algorithm, will be interpolated. At interpolation phase, the distance to the runway is calculated and decision of which reference altitude landing path is going to use, will be given. For this purpose, lookup tables are used. They basically take the initial altitude and distance to the runway and decides the optimal landing path when the landing command arises. Block diagram of interpolation with lookup tables is given in Fig. 69 [32].

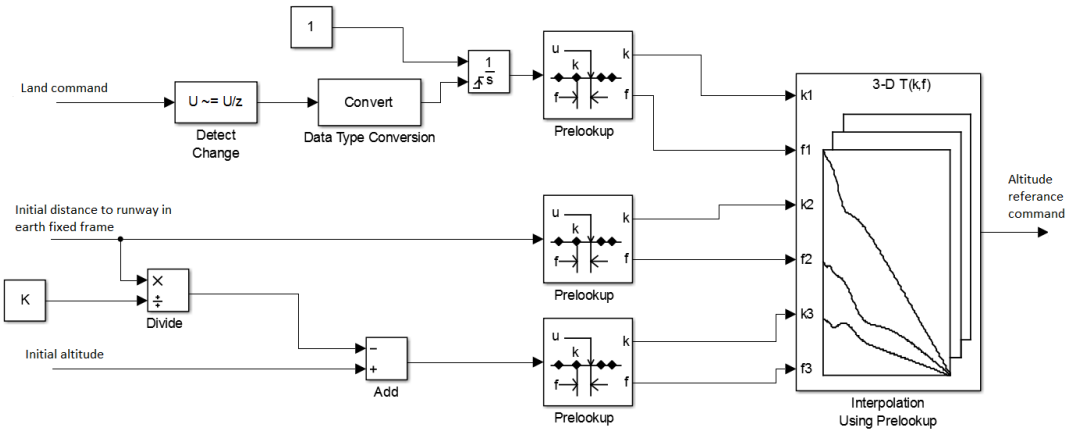

Figure 69 – Block diagram of interpolation block

When landing command is given the count begins to give the desired reference signal at that specific time instant. By using the information of runway distance and initial altitude, optimal commands are determined by interpolation block.

4.2 Nonlinear model predictive control (NMPC)

Predictive controller was first used in industries in 1970s. The aim of NMPC is to solve on-line finite horizon optimal control problem with respect to system dynamics and constraint involving states and input signals.

For tracking problems, like following a landing trajectory, nonlinear model predictive control (NMPC) is an applicable choice, which is an optimization based method. Consider that we have a controlled process whose state $x(t)$ can be measured at discrete time instants. Our aim was to determine the control inputs $u(t)$ such that $x(t)$ follows a given reference $x_{ref}(t)$ as good as possible. Consider the function

$$x^+ = f(x, u) \tag{4.4}$$

Where x^+ is the state value at next time instant. Starting from current state and an initial control sequence with $prediction\ horizon\ lenght > 2$, we were interested in constructing a prediction horizon x_u by iterating equation (71). Proceeding this way we obtained predictions $x_u(k)$ at time t_{n+k} in the future. Now by running an optimization algorithm, an optimal sequence of $u(k), \dots, u(N-1)$ was determined where N is the length of prediction horizon. This optimization algorithm was minimizing a cost function which is given by:

$$\gamma\big(x_u(k)\,, u(k)\big) = \|x_u(k) - x_d(k)\|^2 + \lambda\|u(k)\|^2 \tag{4.5}$$

Where $\|\cdot\|$ is Euclidean norm and $\lambda \geq 0$ which is a weighting parameter between control signal and state error. When the optimal input signal is achieved, only the first element of this signal is applied to the system dynamics and the process is iterates to next time instant and so on [35].

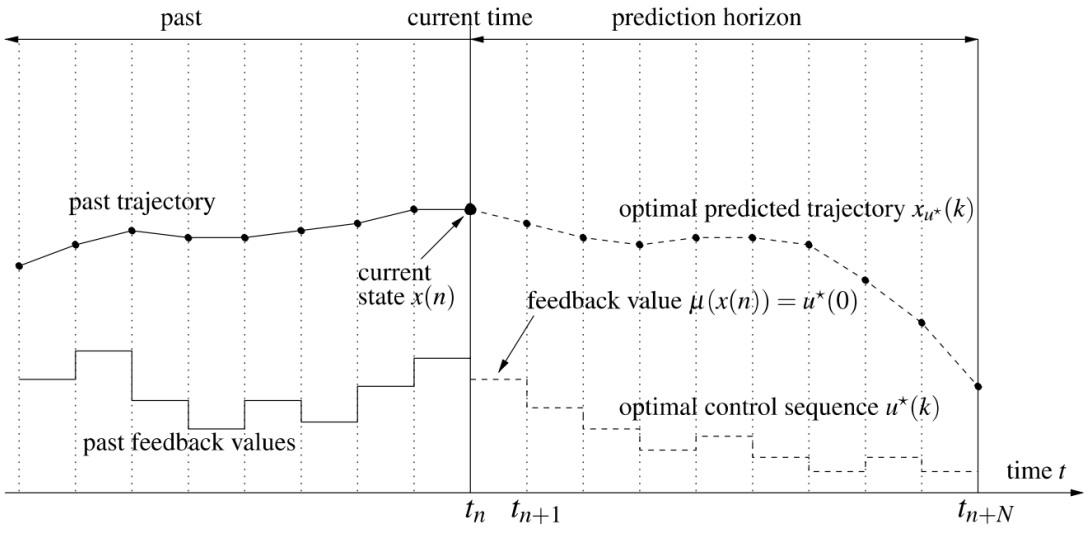

Figure 70 – Model predictive control architecture

4.2.1 Discretization of continuous Time system

In order to apply NMPC to nonlinear system dynamics, equations of motion need to be discretized. There are many ways to discretize a system like Eulers forward d differentiation method, Euler's backward differentiation method, Tustin's method, bilinear transformation and Runge Kutta method [36]. Higher order methods are more accurate than others but it is important for NPMC to have a simple and fast solutions, so Euler's forward method used in this thesis. Continuous dynamics of the aircraft is given by:

$$\dot{x}(t) = f_{cont}\big(x(t), u(t)\big) \tag{4.6}$$

This function is discretized using Euler's forward discretization method for interval T_s and time step t_k.

$$\dot{x}(t) = \frac{x(t_{k+1}) - x(t_k)}{T_s} \tag{4.7}$$

So it becomes;

$$x(t_{k+1}) \approx x(t_k) + T_s f_{cont}\big(x(t), u(t)\big) \tag{4.8}$$

In our case, we were interested in controlling the aircraft's speed, altitude and heading with control signals of throttle, elevator and aileron respectively which implied that we have a control horizon length of three. Sampling interval of 100 Hz was used in simulations ($T_s = 100$). Euclidian norms of states errors were used in order to setup a cost function which contained altitude, speed and heading errors. For simulation purposes step reference inputs given to the system and relevant states were observed. In figure 54, the aircraft is in level flight at altitude of 50m and 12ms speed and it is demanded to climb 5 meters. Since Speed and altitude states were closely bounded with each other, both of them are sketched and observed. The result is illustrated in Fig. 71.

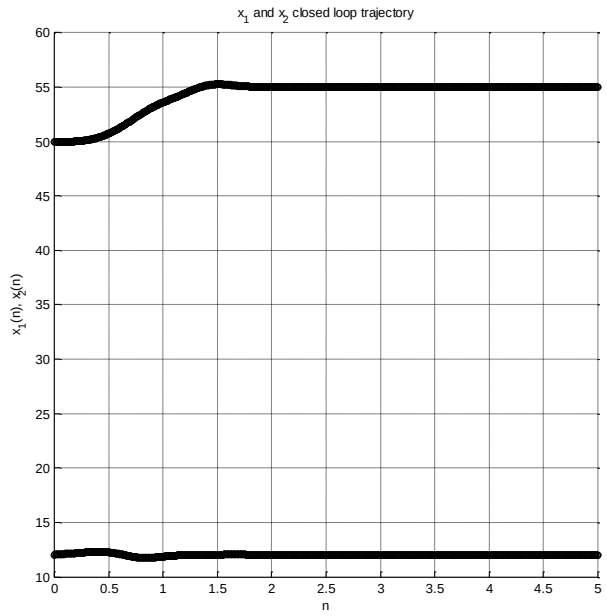

Figure 71 – Aircraft altitude response wrt given step reference input (NMPC)

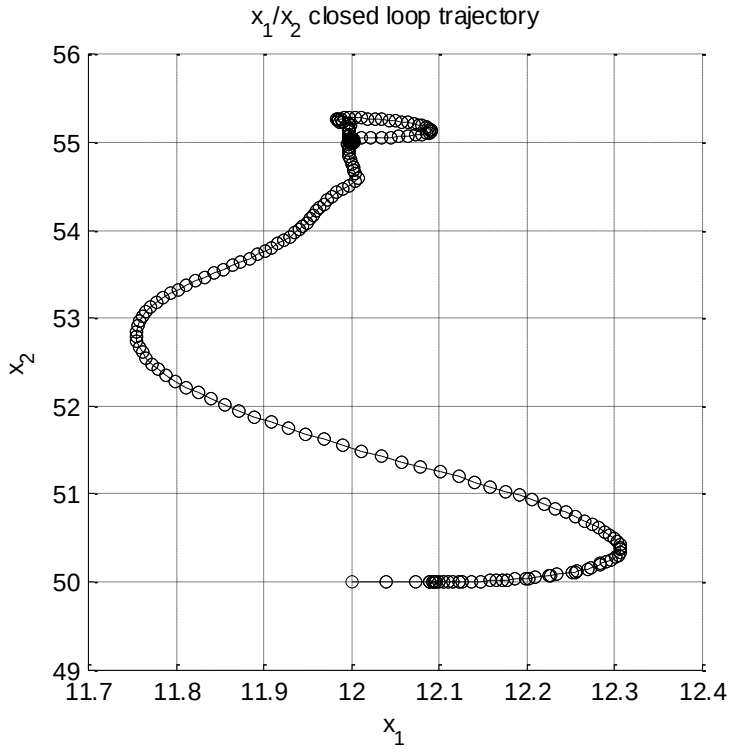

Figure 72 – State graph where x1 is speed and x2 is altitude

Step reference input given as 15ms speed and the states, speed and altitude are observed:

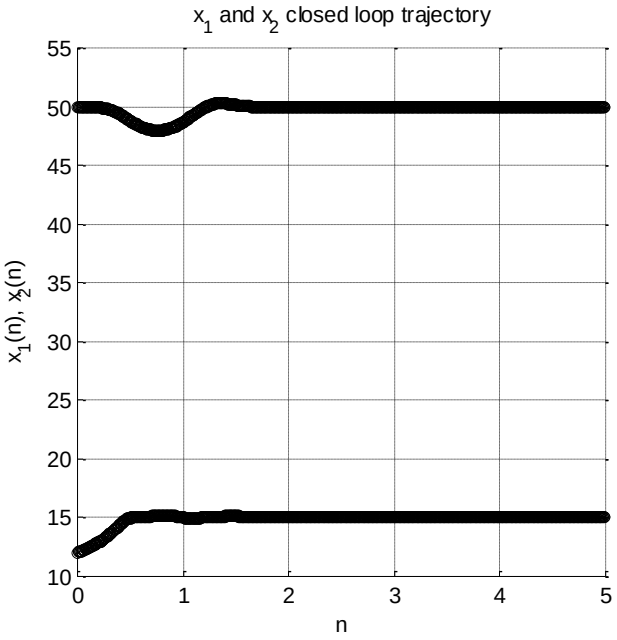

Figure 73 – State graph where x1 is speed and x2 is altitude

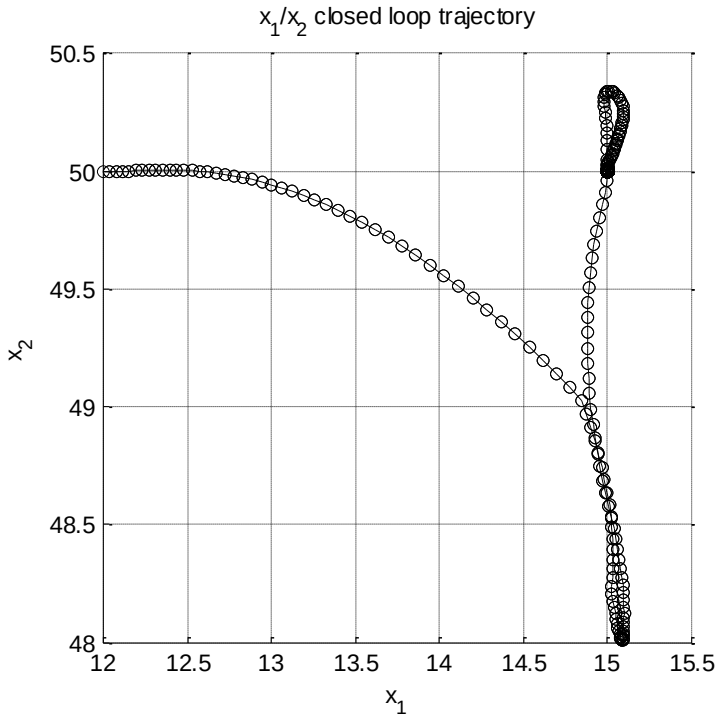

Figure 74 – State graph where x1 is speed and x2 is altitude

Optimizing the nonlinear model dynamics requires very high computational work. So linearized model of Telemaster UAV was used to predict aircraft movements. The Linearized model which was trimmed at 12ms speed and 50m altitude was used. The simulations were tested on nonlinear model of Telemaster UAV. By trial and error methodology the prediction horizon was selected as 20 and control horizon was 3.

4.3 Cross Track Controller

In order to control cross track error, a PI controller with wind correction is implemented. A heading command is generated using the equation (4.9).

$$\psi_{cmd} = K_p(xt_d - xt_{a/c}) + K_i \int (xt_d - xt_{a/c}) + \psi_{WCA} \tag{4.9}$$

Wind speed, wind direction, airspeed and desired course angle are used for calculation of ψ_{cmd}.

Wind to track angle:

$$WTAngle = DesiredCourse - windDir$$

Wind correction angle:

$$SinWCA = windspeed * \frac{\sin(WTAngle)}{Airspeed}$$

$$WCA = sin^{-1}(sinWCA)$$

Here WCA is the angle that must be added to the desired heading to correct the effect of crosswinds [37].

4.4 Lateral track control law

A nonlinear track guidance algorithm is implemented in to our Telemaster UAV model. This algorithm enables the aircraft to track a smother path while flying along these waypoints and while landing phase of flight, this guidance method will reject the wind disturbances to keep the aircrafts y-position aligned with the runway. This method simply computes the initial cross track error between the desired track and the aircrafts position, than produces aileron commands to eliminate the cross track error. In high wind situations, by directly using the wind information, this algorithm is able to keep lateral tracking and stability of aircraft. It is required to calibrate our reference frame with respect to some constant [38]. So the following notations are needed: $\psi = \angle(\bar{U}, y_{North})$, $\psi_W = \angle(\bar{W}, y_{North})$ and $\psi_{12} = \angle(< wp_1, wp_2 >, y_{North})$. Since we have the velocity and position components in the North-East frame, we need to find velocity and position components in track reference $(< x_t, y_t >)$ frame. A transformation is applied as a rotation by an angle $(\psi_{12} - {}^{\pi}/_{2})$ and the resulting rotation matrix is given by

$$T_{\psi} = \begin{bmatrix} \cos(\psi_{12} - {}^{\pi}/_{2}) & -\sin(\psi_{12} - {}^{\pi}/_{2}) \\ sin(\psi_{12} - {}^{\pi}/_{2}) & \cos(\psi_{12} - {}^{\pi}/_{2}) \end{bmatrix} \tag{4.10}$$

When we apply the transformation matrix to windspeed and airspeed vectors, we obtain

$$\vec{U}_{track} = T_{\psi}\vec{U} \tag{4.11}$$

And

$$\vec{W}_{track} = T_{\psi}\vec{W} \tag{4.12}$$

Cross-track and along-track velocities can be written as

$$\begin{cases} \dot{X}_{track} = U_{track_x} + W_{track_x} \\ \dot{Y}_{track} = U_{track_y} + W_{track_y} \end{cases} \tag{4.13}$$

And the psi-rate command is obtained as:

$$\dot{\varphi}_{com} = K_R(kx_e\dot{y}_t - y_e\dot{x}_t) \tag{4.14}$$

Where K_R is some constant determined by simulations iteratively and $K_R = -0.00001$ is found satisfactory in our simulations. The yaw rate command is limited by $\pm 0.2 rad/s$. The k parameter in the equation implies the point that aircraft crosses the desired track. k=1 means that aircraft goes to the desired waypoint directly.

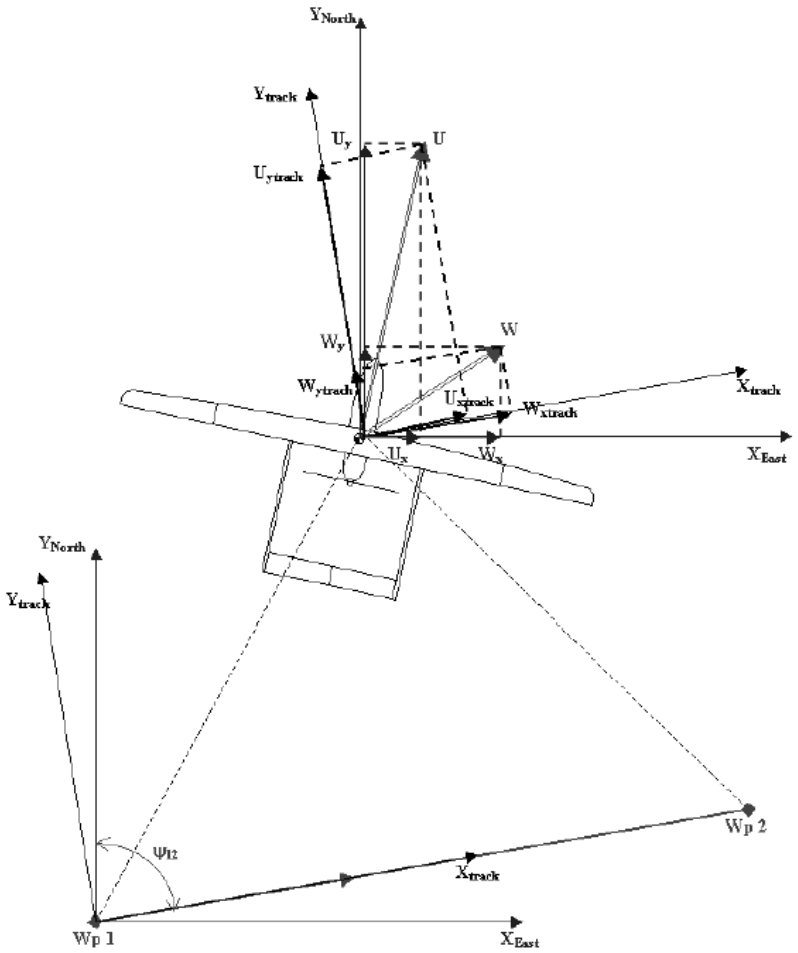

Figure 75 – Lateral track controller wind and airspeed axis

For different k values and several wind speeds the simulation results are illustrated in Figures 76-79.

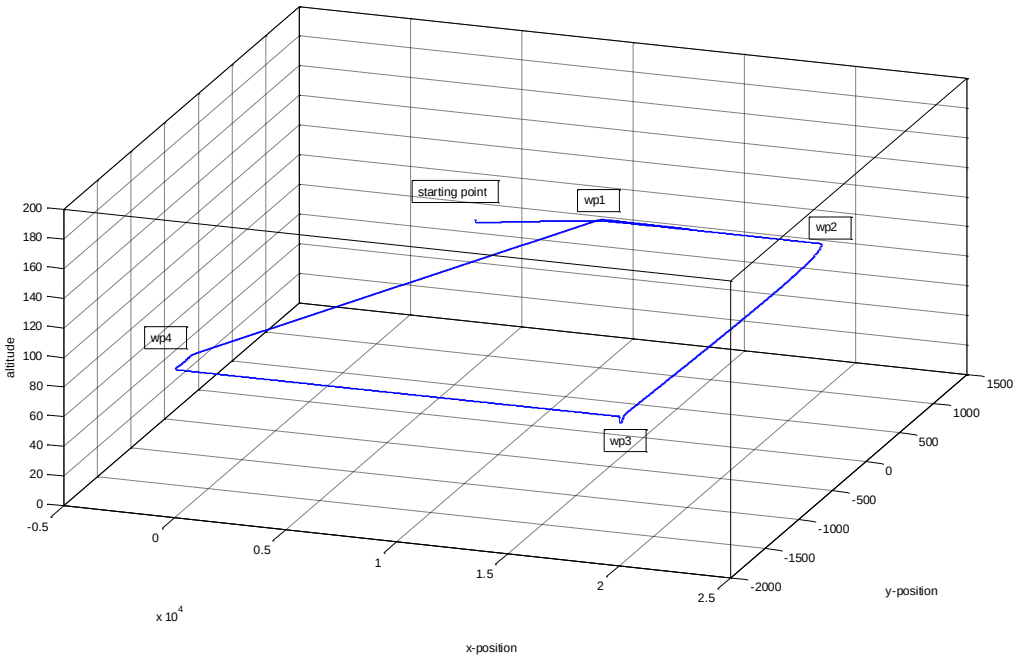

Figure 76 – k = 1

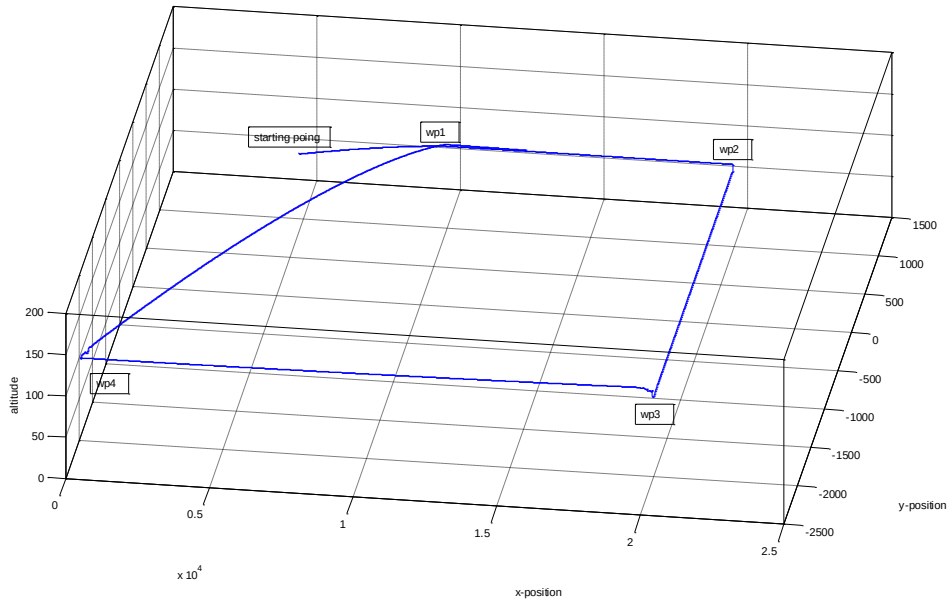

Figure 77 – k = 0.7

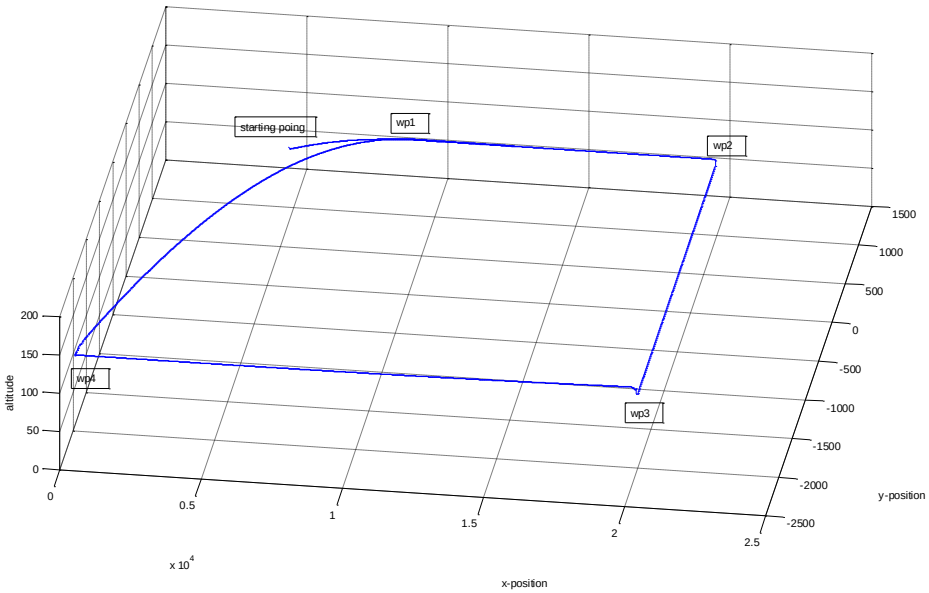

Figure 78 – k = 0.4

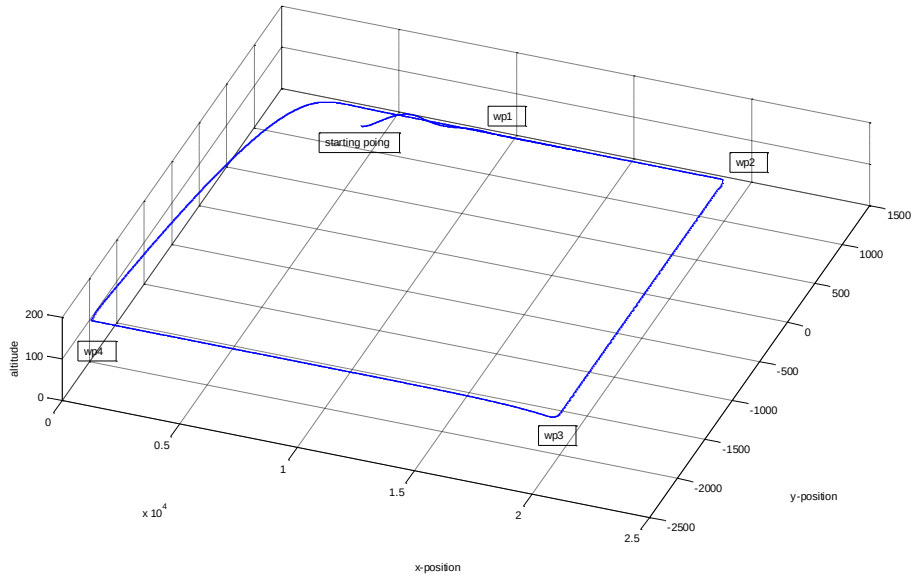

Figure 79 – k = 0.1

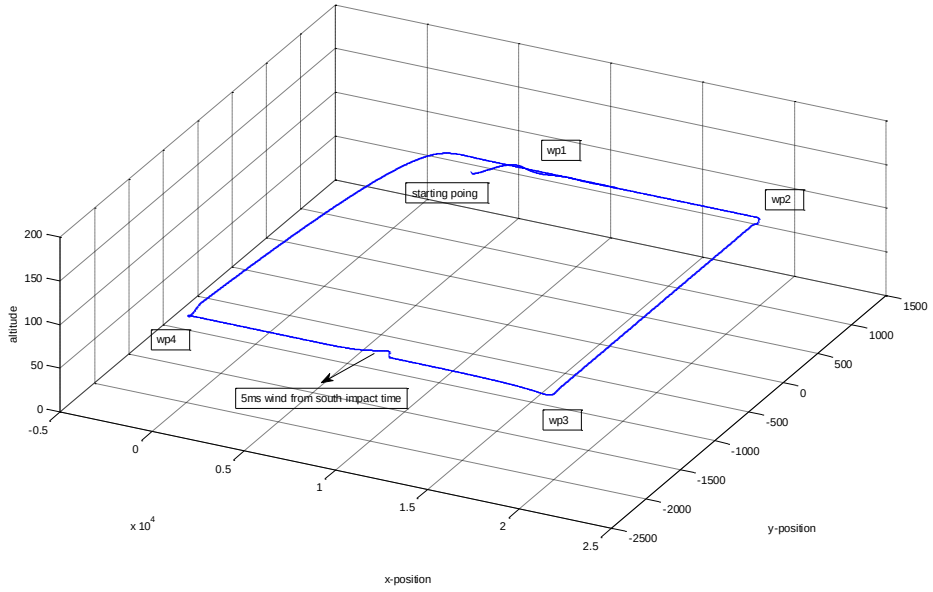

Figure 80 – k = 0.1, 5 m/s wind from south at 1500 sec. in simulation time

Lateral track guidance enables us to determine the flight behavior of the aircraft by changing the parameter k. Figures 76-79 shows the effect of change of parameter values which implies us, as the k gets close to one, the aircraft tends to go to the next waypoint directly. In figure 80 a 5 m/s continuous wind is applied in a specific time. The performance of this guidance block has observed.

CHAPTER 5

SIMULATION RESULTS

Different scenarios are tested, in order to compare the performances of mentioned control algorithms. All simulations are made with aircrafts nonlinear 6DOF model created in MATLAB/Simulink. The static and dynamic stability derivatives are taken from DATCOM software. The adverse weather conditions applied as disturbances while landing phase and the robustness of controller and guidance algorithms are tested. These methods are:

- PID with Lateral track guidance
- PID with Cross track guidance
- LQT with Cross track guidance
- SMC with Cross track guidance
- NMPC

The starting parameters of aircraft are 780m away from runway, 50m altitude, and 12m/s airspeed. The optimal path found by steepest decent algorithm is used as reference altitude command for all simulations. It is desired that to enter flare maneuver at 12 m/s speed. Cross, tail and head winds applied at different times with different durations to the aircraft. We expect to see the designed controllers and guidance algorithms to level the aircraft and track the desired trajectory created for landing. The aircraft must be in specific states at a given time.

It is expected from controllers that to handle instant and continuous wind disturbances while keeping the altitude in safe levels. Since lateral movements cause some altitude

drops which may cause at slow speeds to crash. Cross-winds, tail-winds and head-winds are applied with different durations.

It is observed that, lateral and longitudinal dynamics require different prediction horizons lengths while designing model predictive controller. Since the lateral movements are coupled with longitudinal movements, the lateral controller needs to have greater prediction horizon. Considering this, two model predictive controller are designed where one controls longitudinal maneuvers and the other controls lateral movements. The prediction horizon is selected as 20 for the longitudinal controller while the control interval is 0.05 seconds. Prediction horizon for lateral controller is 100 and the control interval is the same as longitudinal one. In simulations the landing performances of designed controllers are plotted and discussed. Except scenario 9, no disturbances are applied to the simulations.

5.1 Scenario 1

For the first scenario, a continuous wind is applied from the start of landing phase. The wind speeds are: 1m/s, 3m/s and 5m/s. The direction of wind is north (coming from south). It is expected that the lateral controllers to beat the wind effect and with some sideslip, maintain the aircrafts position in y-axis which is aligned with the runway. Performances of all five methods are analyzed.

PID with LTG

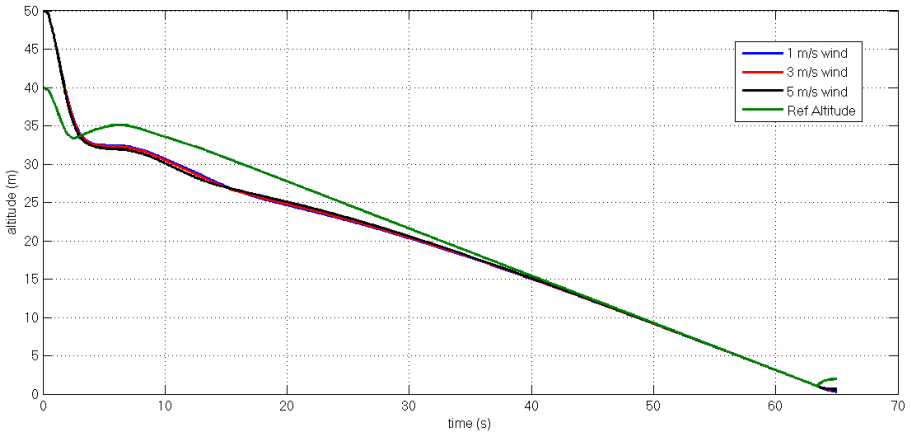

Figure 81 – Altitude response of PID with LTG

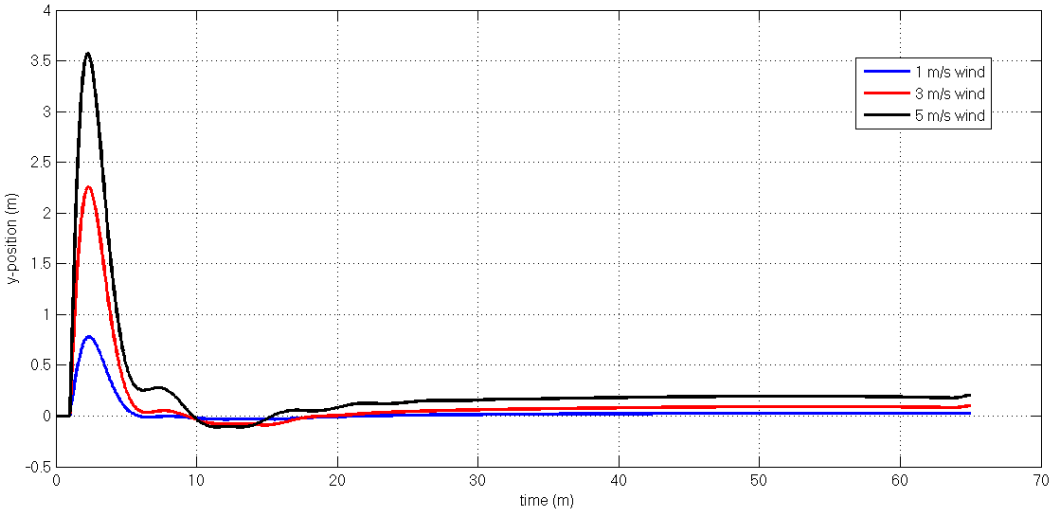

Figure 82 - Y-position plot for PID with LTG

1ms and 3ms wind speeds did slightly effect the y-position of aircraft but the 5ms wind speed caused some oscillations. Also here is a steady state error. The altitude response of aircraft didn't effected by wind forces.

PID with CTG

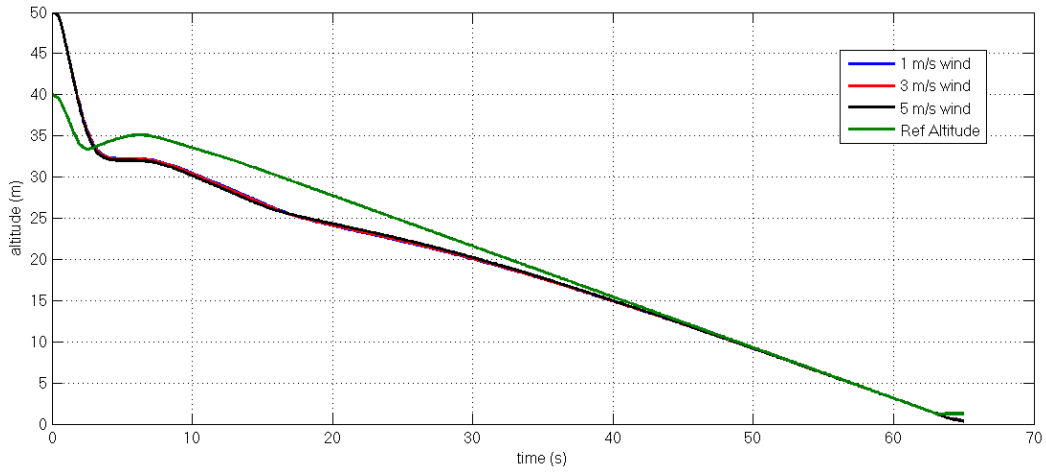

Figure 83 - Altitude response of PID with CTG

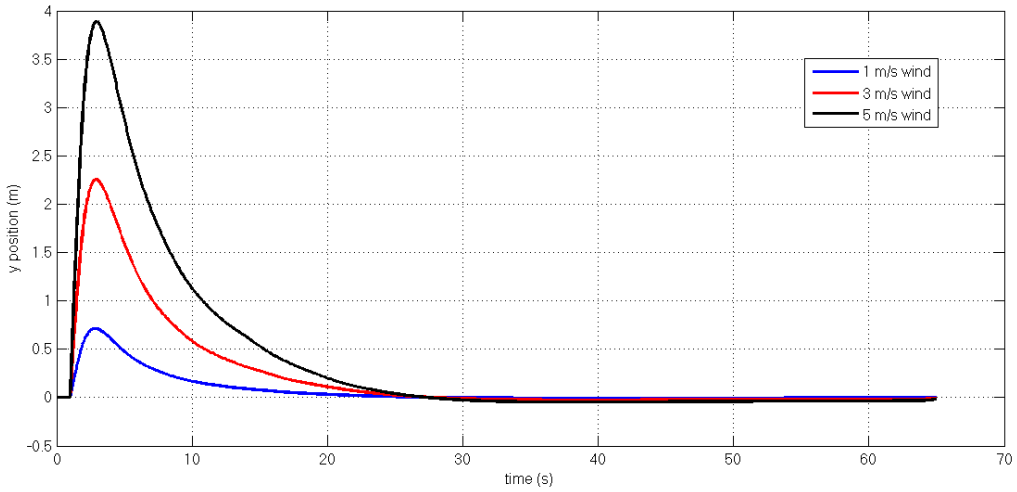

Figure 84 - Y-position Plot for PID with CTG

The response of PID controller with CTG, has a slow response, but there is no steady state error like in PID with LTG. The altitude of aircraft didn't effected by side winds.

LQT with CTG

Then, the LQT with CTG is simulated using scenario 1

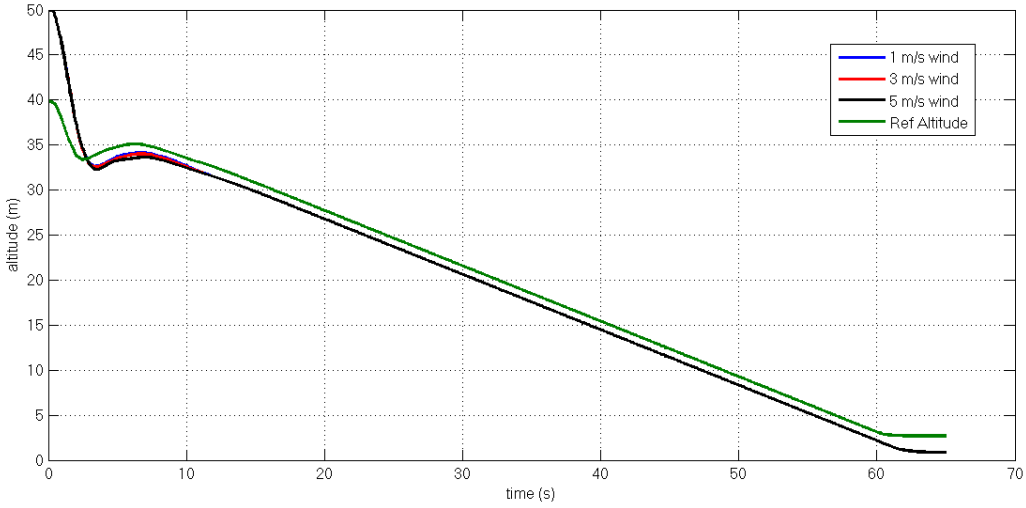

Figure 85 – Altitude response of LQT with CTG

Although there is a small offset, LQT controller didn't effected by winds (1ms, 3ms and 5ms).

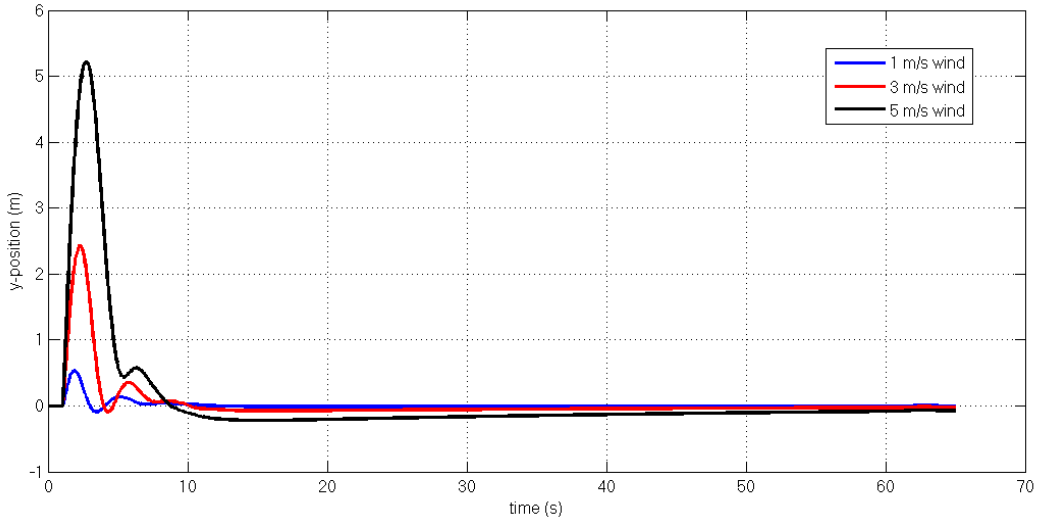

Figure 86 - Y-position plot for LQT with CTG

The response of LQT is faster and more accurate than PID controllers. Hoverer oscillations are still present and there is a small steady state error.

1st SMC with CTG

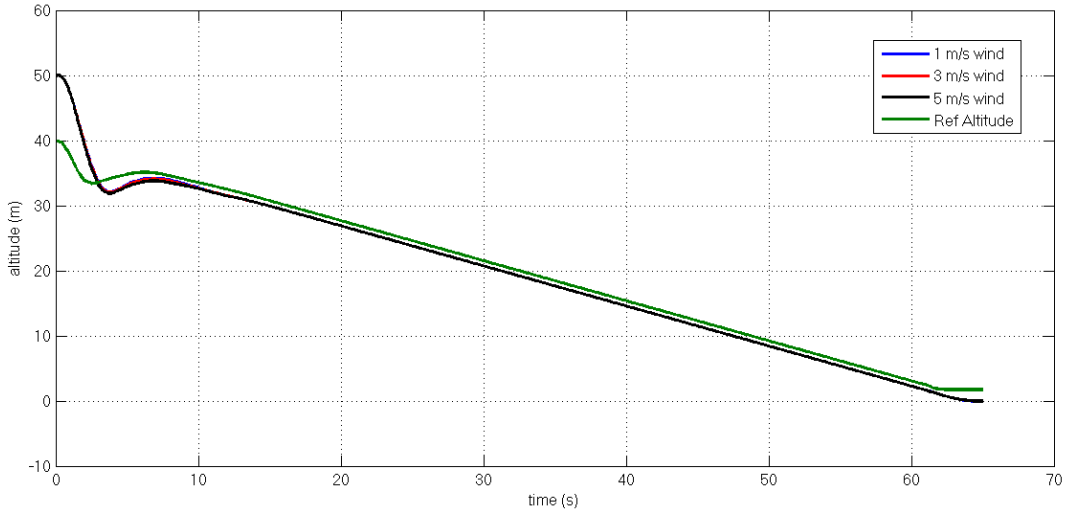

Figure 87 - Altitude response 1st SMC with CTG

Sliding mode controller also has a good altitude response like LQT controller. The offset is smaller and it did not effected any of wind speeds applied.

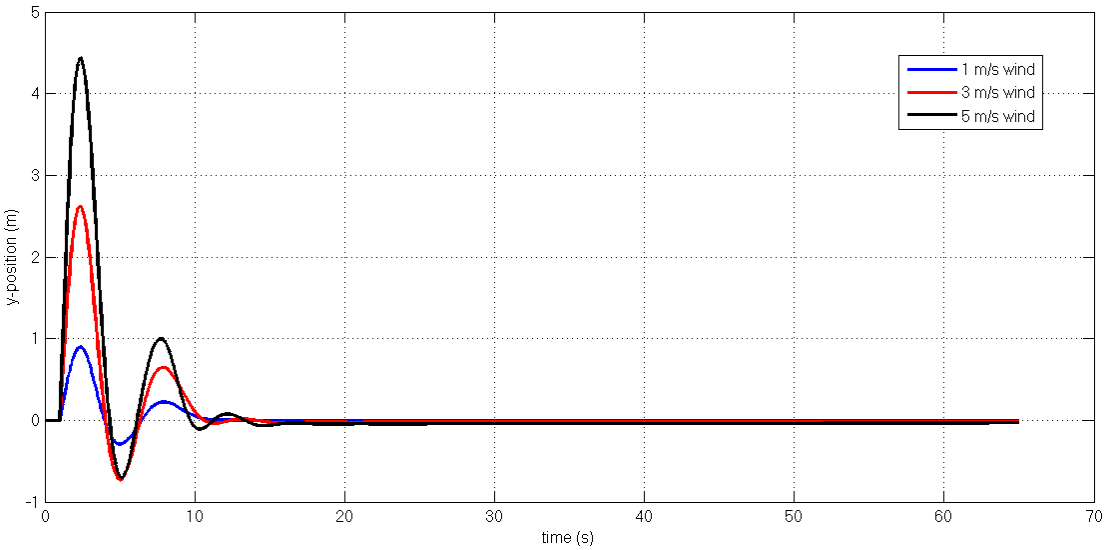

Figure 88 - Y-position plot for 1st SMC with CTG

The lateral response of SMC is very similar to LQT response. The response of controller is as fast as LQT response. Also there is no steady state error observed. The aircraft maintained its y-position successfully.

2nd SMC with CTG

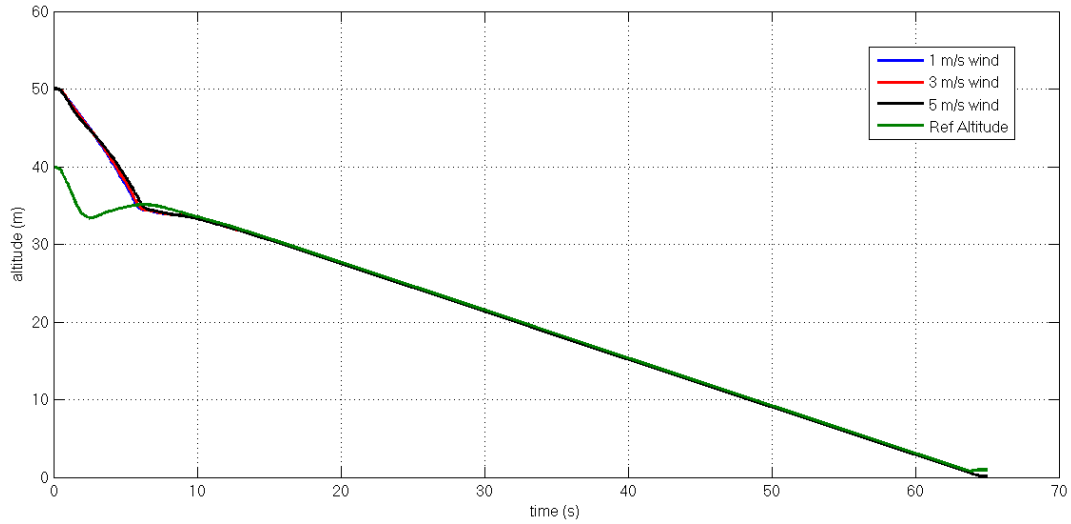

Figure 89 – Altitude response of 2nd SMC with CTG

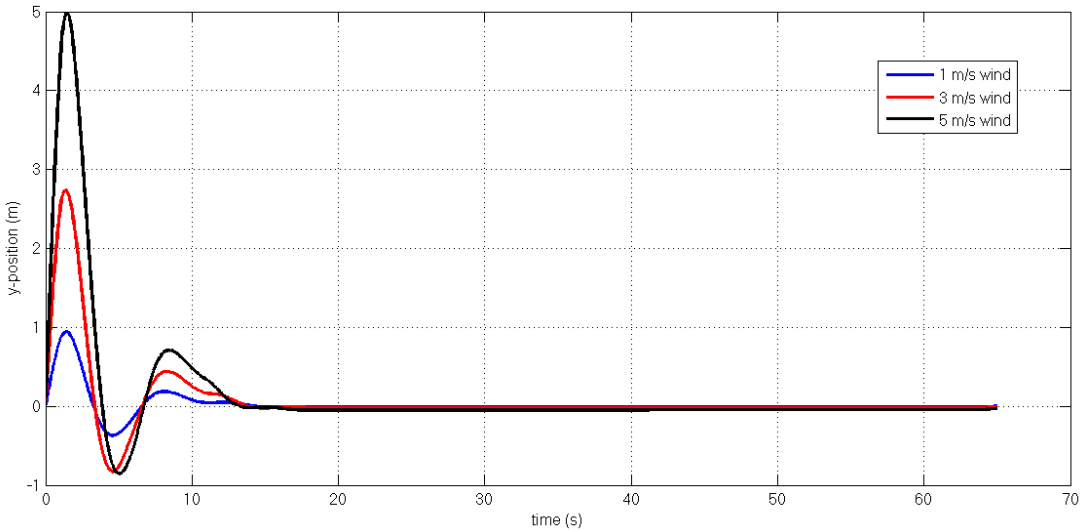

Figure 90 - Y-position of 2nd SMC with CTG

The altitude tracking performance of 2nd SMC is better that normal 1st SMC. The lateral responses are quite similar.

NMPC

Finally, Model predictive controller is tested:

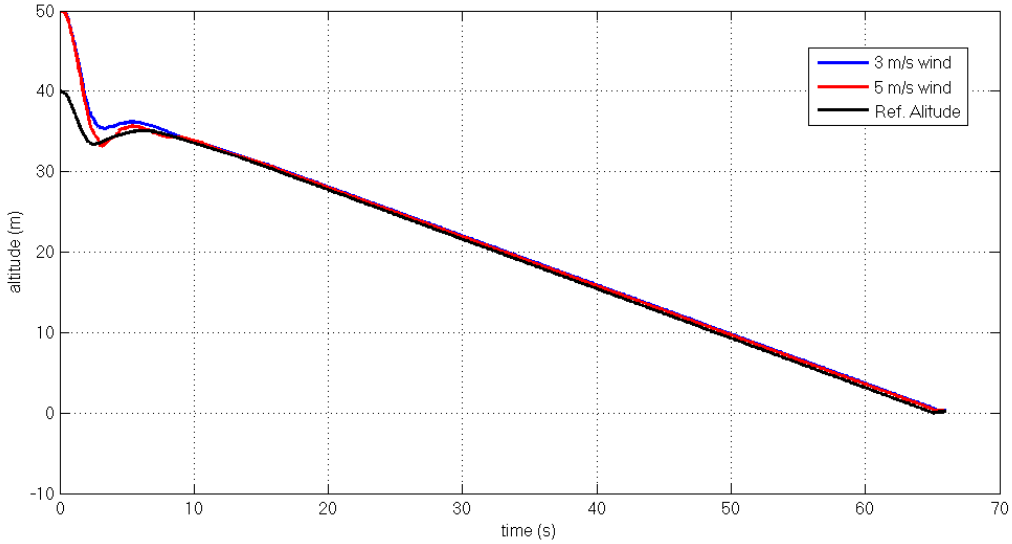

Figure 91 – Altitude response of NMPC

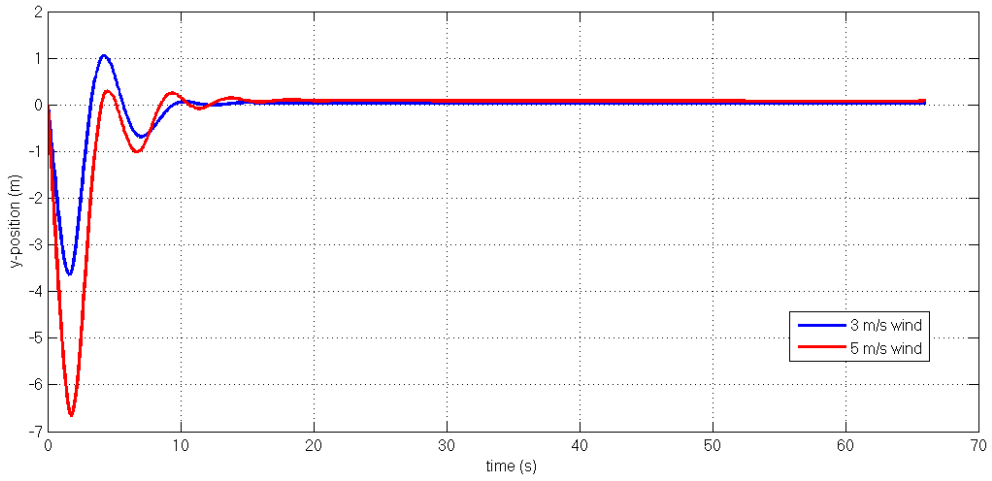

Figure 92 – Y-position plot of NMPC

MPC has a very good tracking performance, the settling time of lateral responses of SMC and MPC are nearly the same but the wind effected the MPC controller more than SMC.

5.2 Scenario 2

Continuous tail winds are applied to the aircraft form the starting time of the simulation, in this scenario. Wind speeds are 1m/s, 3m/s and 5m/s. Altitude and speed responses are presented. Tail winds are challenging situations, since the airspeed of the aircraft can drop dangerously and if it exceeds stall speed it may cause the UAV to crash. So it is expected that the speed controller to adjust thrust that the airspeed of the aircraft will not exceed dangerous levels.

PID with LTG

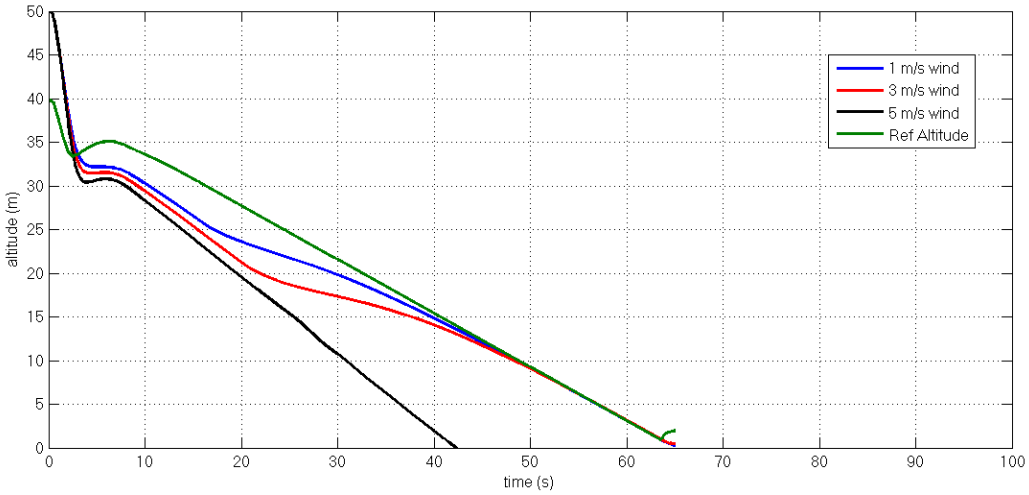

Figure 93 - Altitude response of PID with LTG

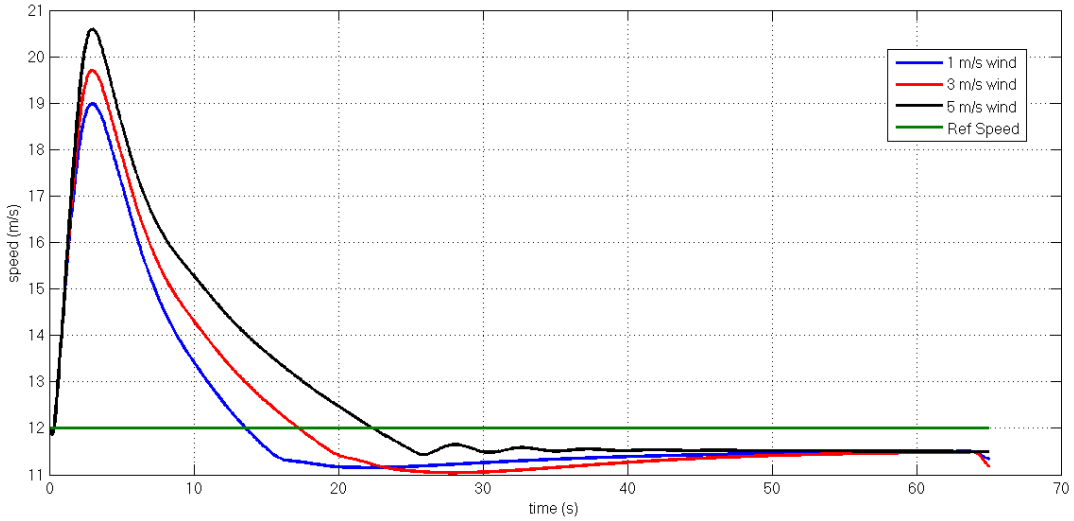

Figure 94 - Speed response of PID with LTG

It can be observed that the PID controller with LTG didn't manage to sustain desired altitude under 5 m/s tailwinds. Aircraft hits the ground. Aircraft speed is changing quickly to get the aircraft into reference altitude and a small offset is observed.

PID with CTG

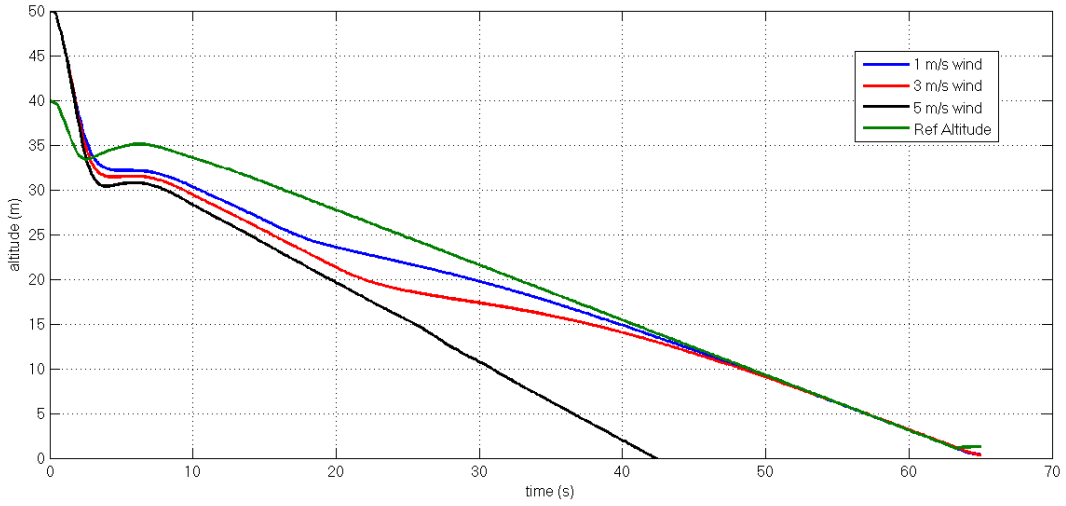

Figure 95 - Altitude response of PID with CTG

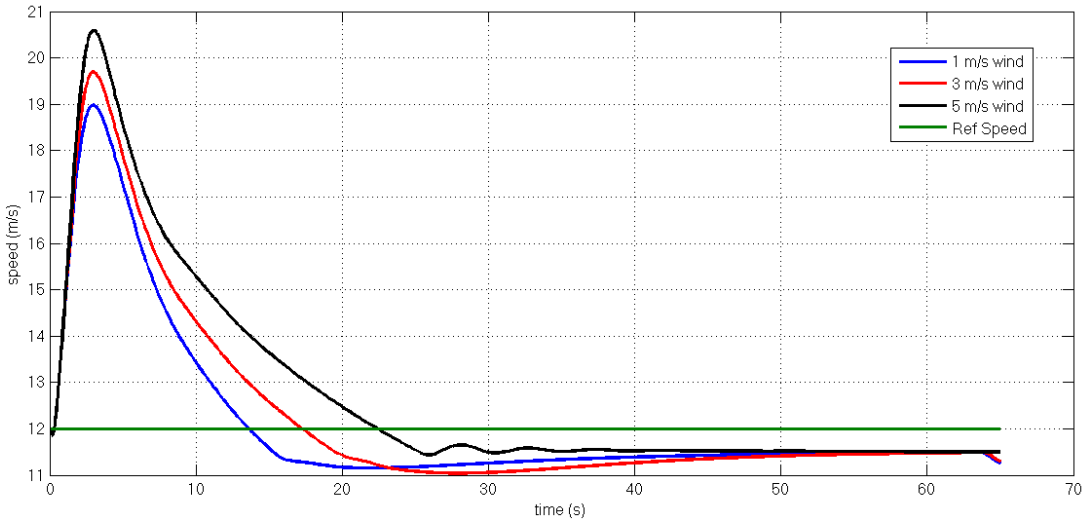

Figure 96 - Speed response of PID with CTG

PID with CTG has very similar response like the previous controller. The controller can't maintain the altitude and aircraft collapses to ground as previous controller setup.

LQT with CTG

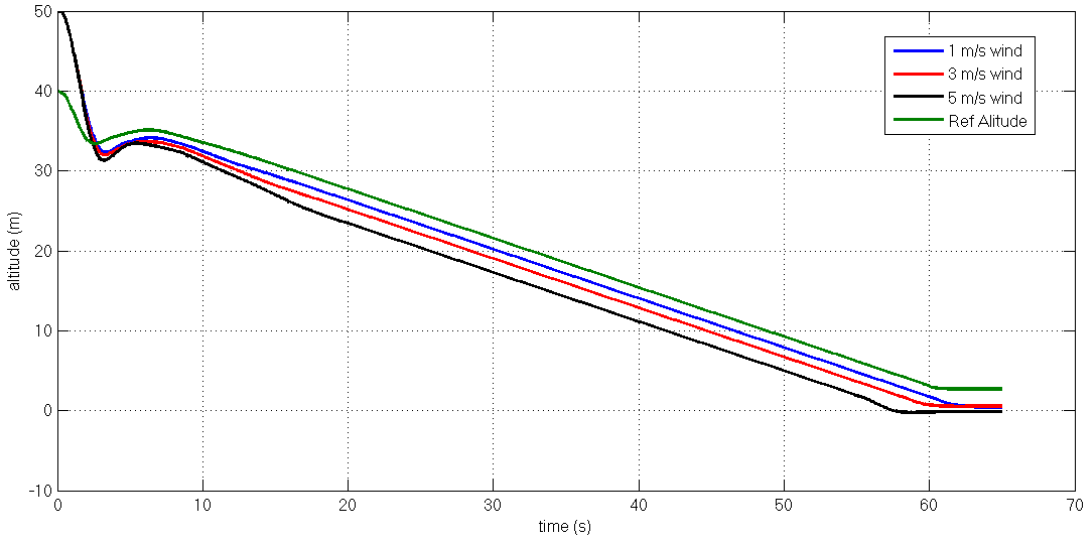

Figure 97 - Altitude response of LQT with CTG

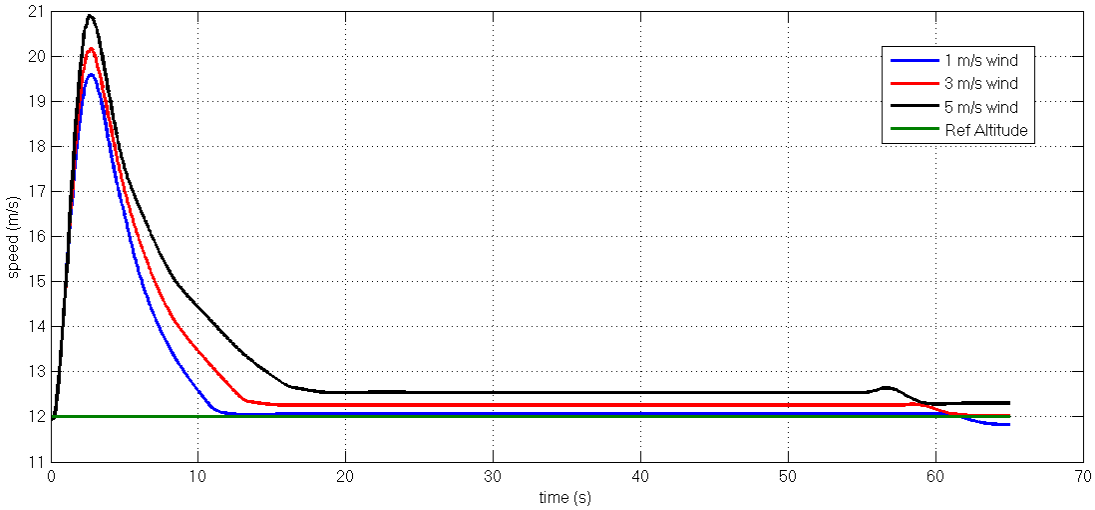

Figure 98 - Speed response of LQT with CTG

The LQT with CTG has also a good altitude response, however an offset is observed at both speed and altitude values. Hopefully, this will not affect the landing in a bad manner. Since LQR is a proportional state feedback control system (doesn't have an integrator), LQR is not offset error free when it is subjected by a constant disturbance.

1st SMC with CTG

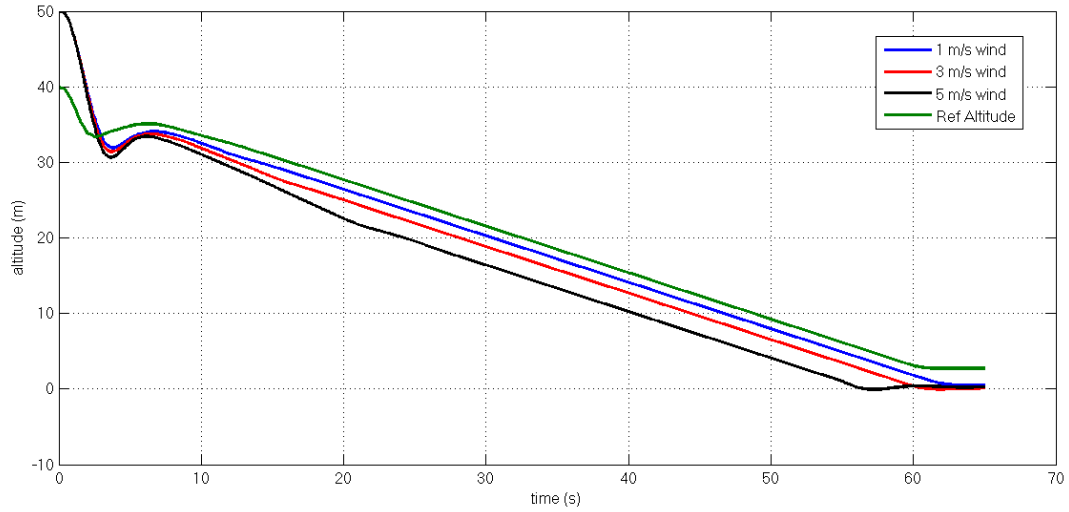

Figure 99 - Altitude response of 1st SMC with CTG

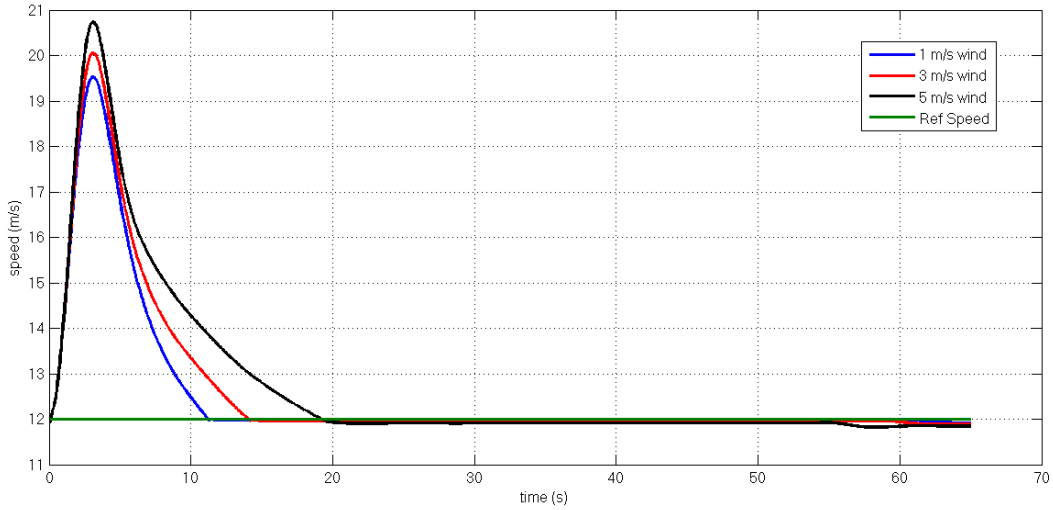

Figure 100 - Speed response of 1st SMC with CTG

Similar altitude response observed with SMC, with an offset aircraft tracks the desired altitude. While under 5 m/s tail wind, the flare maneuver caused sudden altitude drop which will result a harsh landing. No steady state error seen in speed response of SMC.

2nd SMC with CTG

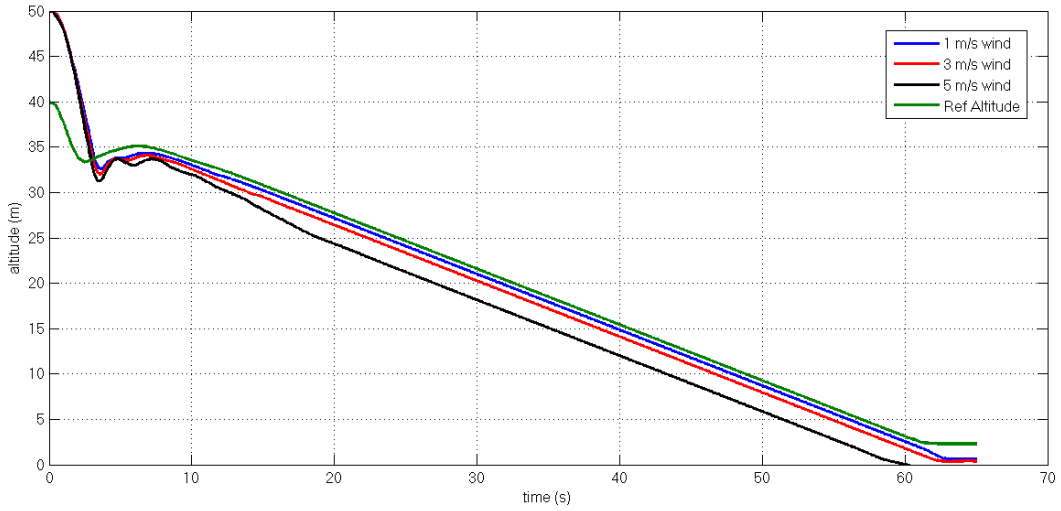

Figure 101 – Altitude response of 2nd SMC with CTG

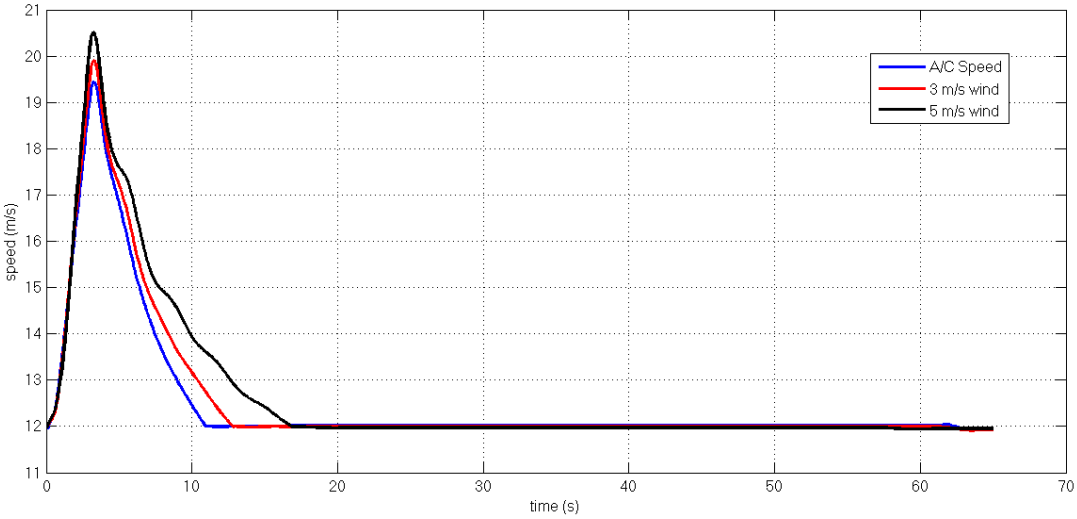

Figure 102 – Speed response of 2nd SMC with CTG

The 2nd SMC controller has better result with respect to PID and LQT controllers. The altitude is responses are similar to 1st SMC, offset still occurs and a harsh landing under 5 m/s wind still exists.

NMPC

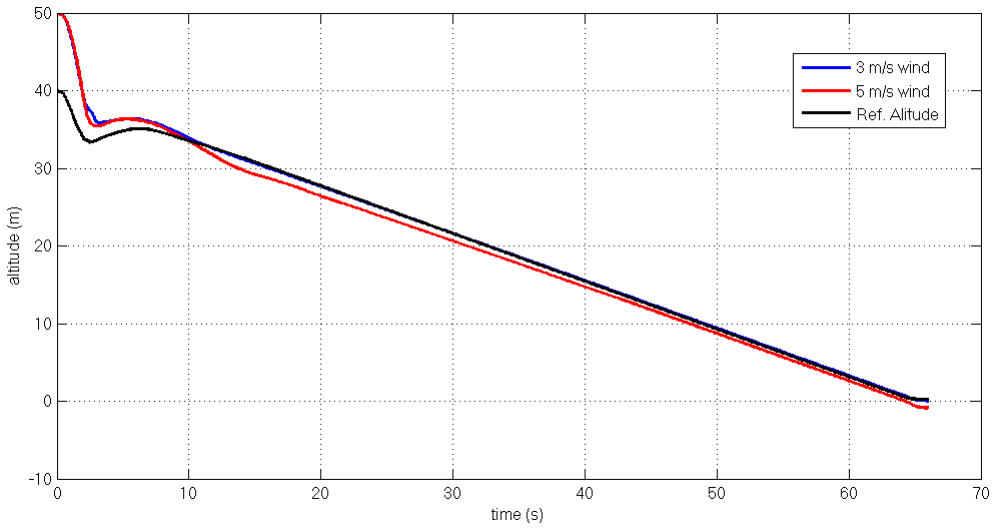

Figure 103 – Altitude response of NMPC

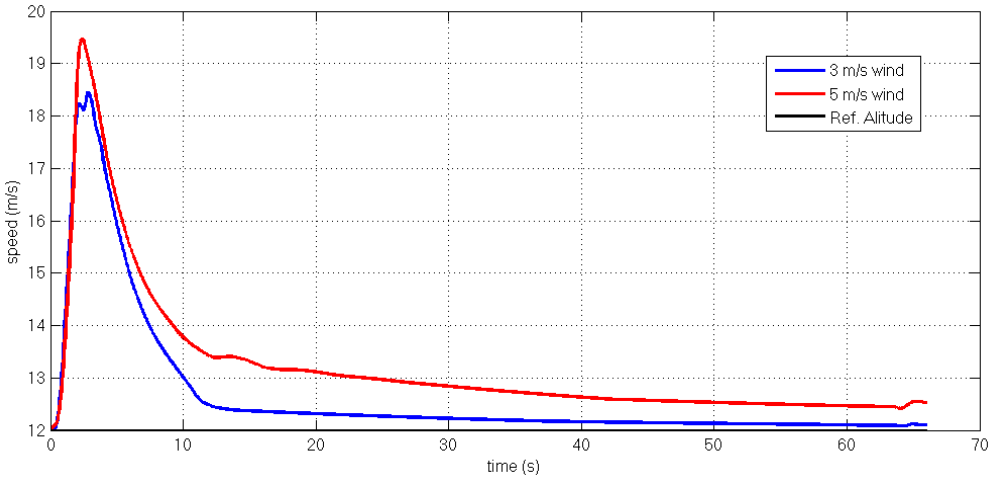

Figure 104 – Speed response of NMPC

It can be observed that, although speed value has some steady state error, altitude response has very small offset, the tracking performance of MPC controller is the best in all controllers. The MPC will land the aircraft safely to the ground.

5.3 Scenario 3

In this scenario, head wings (coming from north) are applied to the aircraft. Altitude and speed responses are observed if they are convenient for landing. Head wings are favorable for takeoff and landing phases of aircrafts, since an airfoil moving towards a headwind can produce more lifting than moving towards a tailwind.

PID with LTG

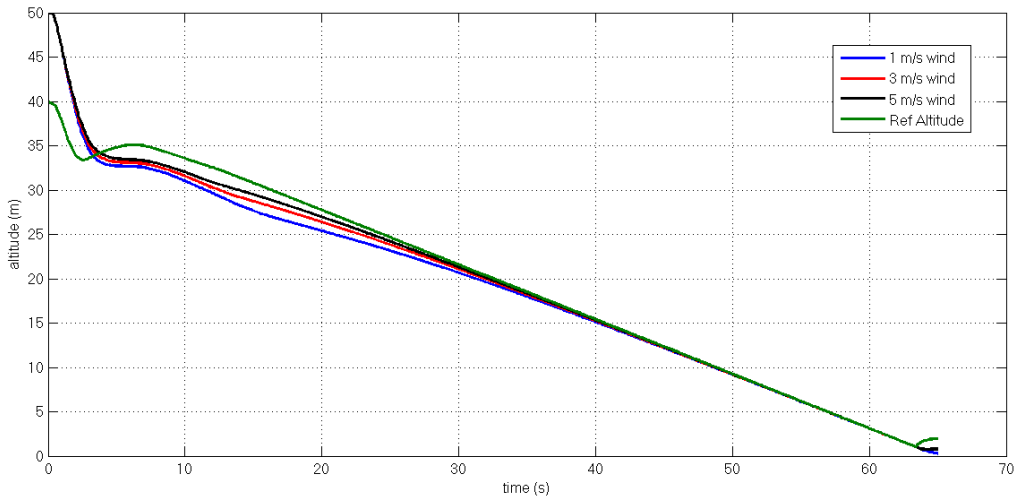

Figure 105 - Altitude response of PID with LTG

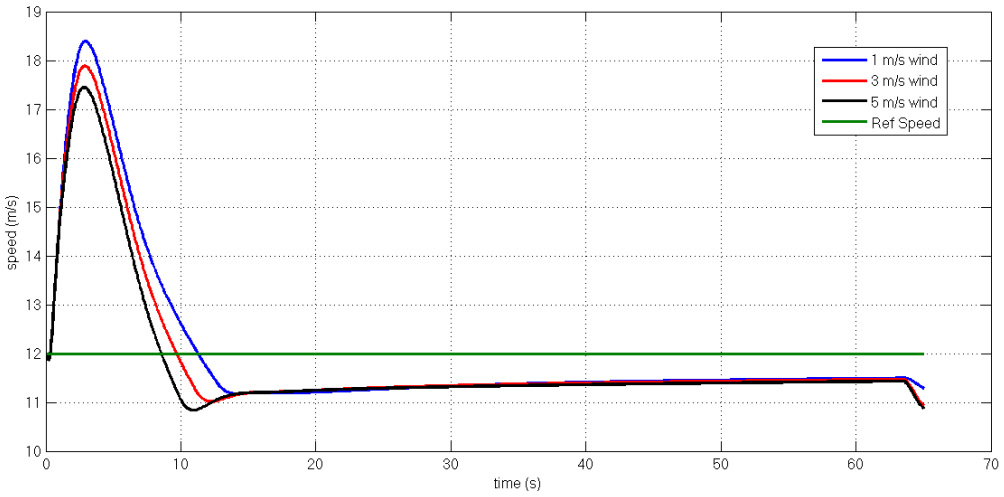

Figure 106 - Speed response of PID with LTG

As the head wind gets stronger the aircraft manages to converge to optimal landing path faster since it is desirable that while take-off and landing to have some head wind. Which will produce more lift and increase the performance of longitudinal maneuvers. PID with LTG setup handles this king of wind easily. By increasing thrust command aircraft manages to keep desired reference speed.

PID with CTG

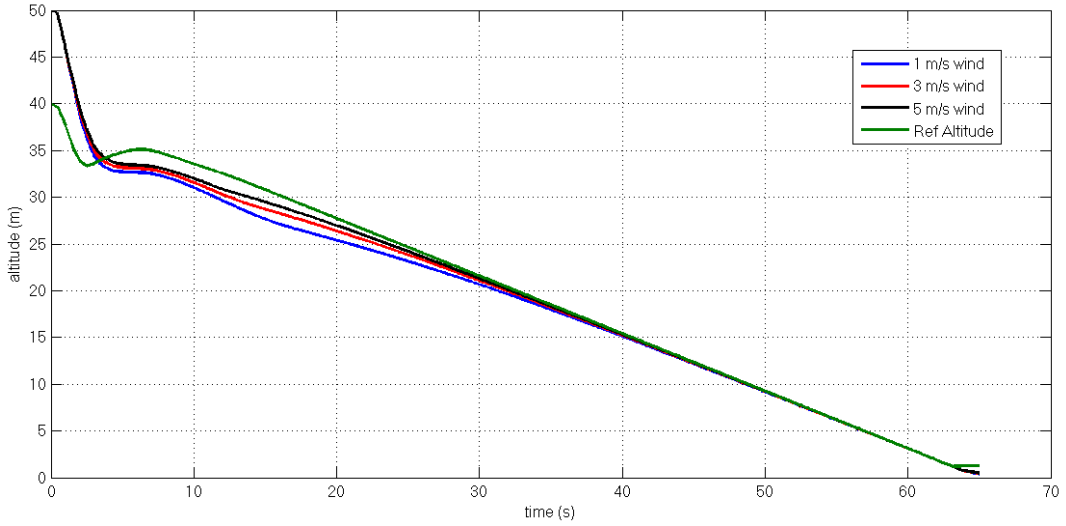

Figure 107- Altitude response of PID with CTG

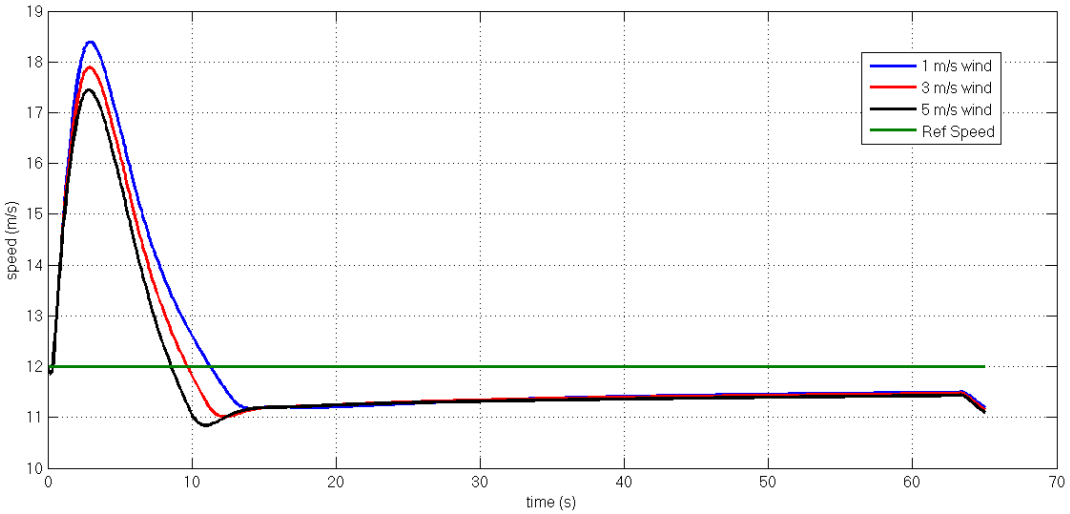

Figure 108 - Speed response of PID with CTG

Both of PID controllers with LTG and CTG has similar responses. The tracking performance of both controllers is good. And the speed responses aren't exceeding limits.

LQT with CTG

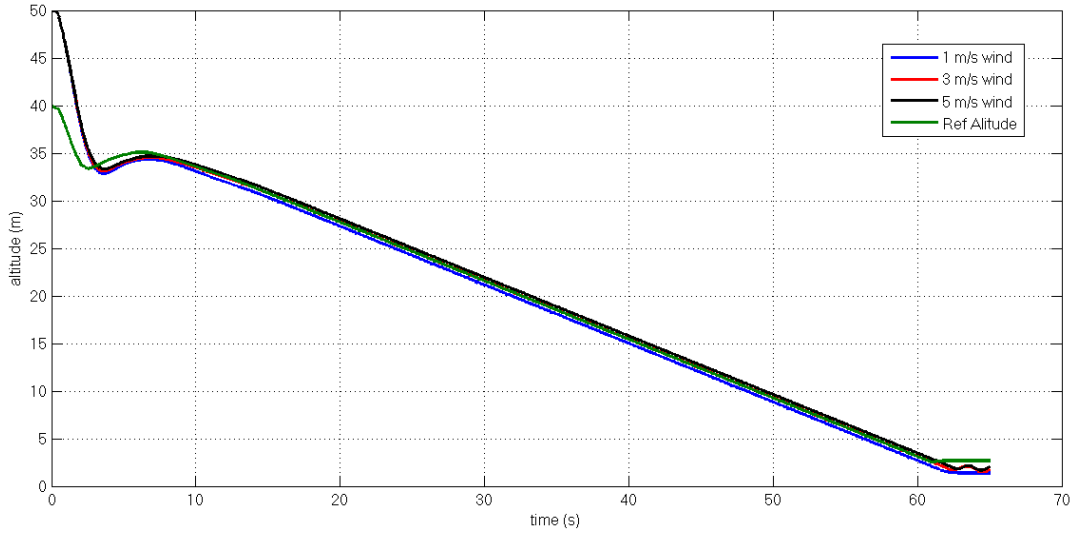

Figure 109 - Altitude response of LQT with CTG

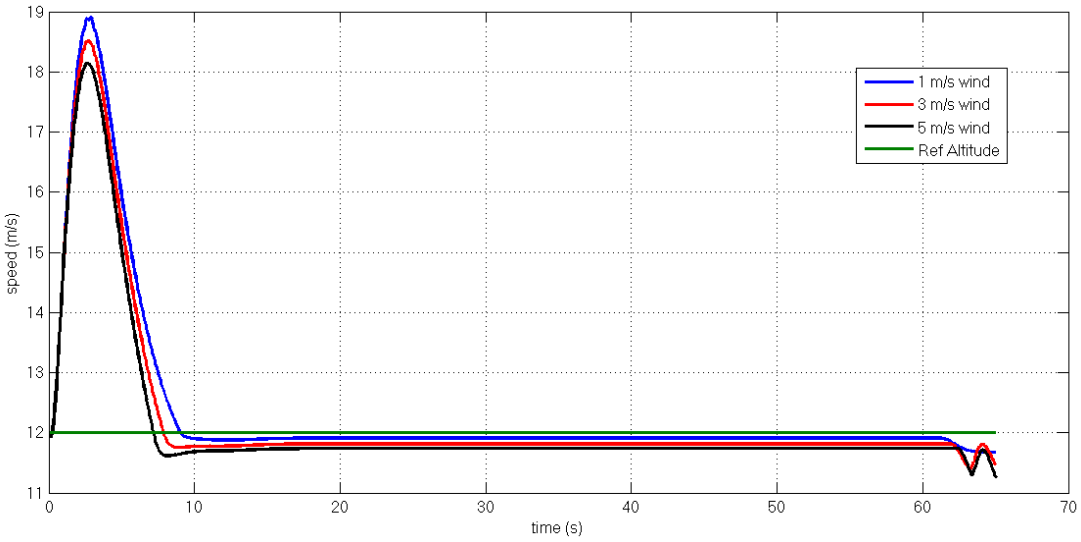

Figure 110 - Speed response of LQT with CTG

The altitude tracking performance under head winds up to 5 m/s of LQT is very good. Very small offsets observed at speed responses under different wind values.

1st SMC with CTG

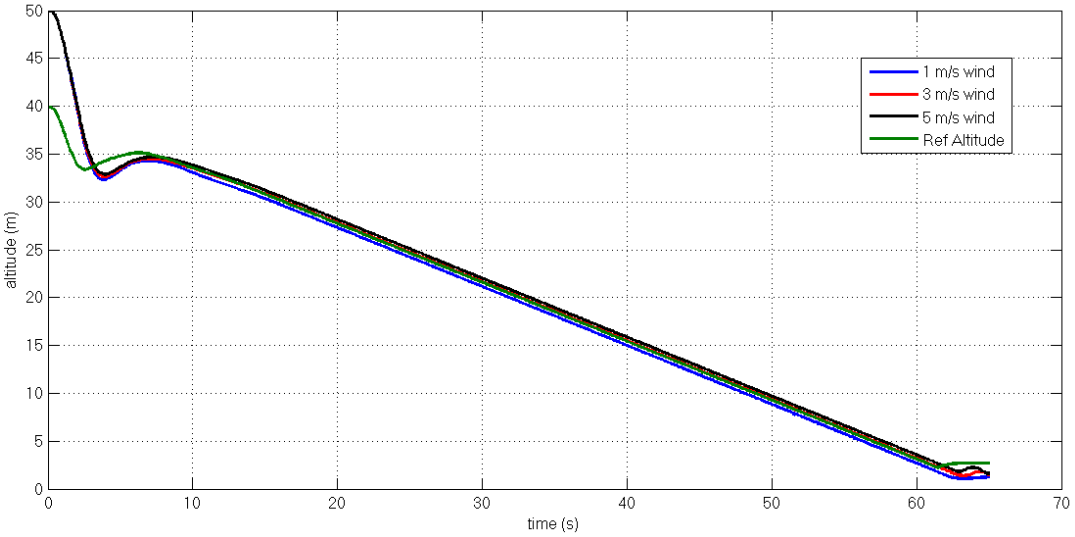

Figure 111 – Altitude response of 1st SMC with CTG

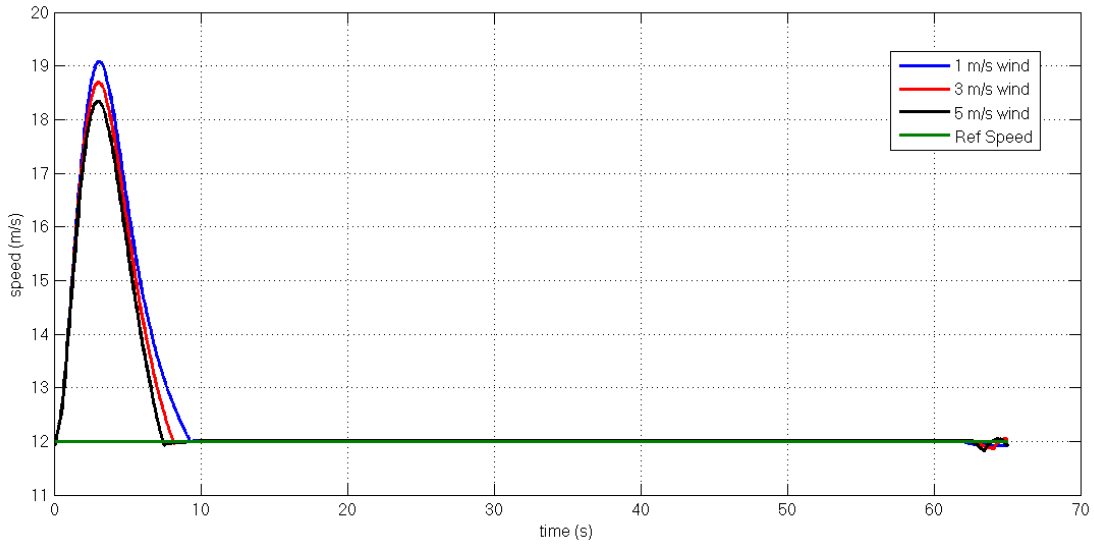

Figure 112 - Speed response of 1st SMC with CTG

The speed response of SMC is very similar to LQT. We can understand from results that, all controllers manage to track the desired landing corridor and slightly effected by head wings.

2nd SMC with CTG

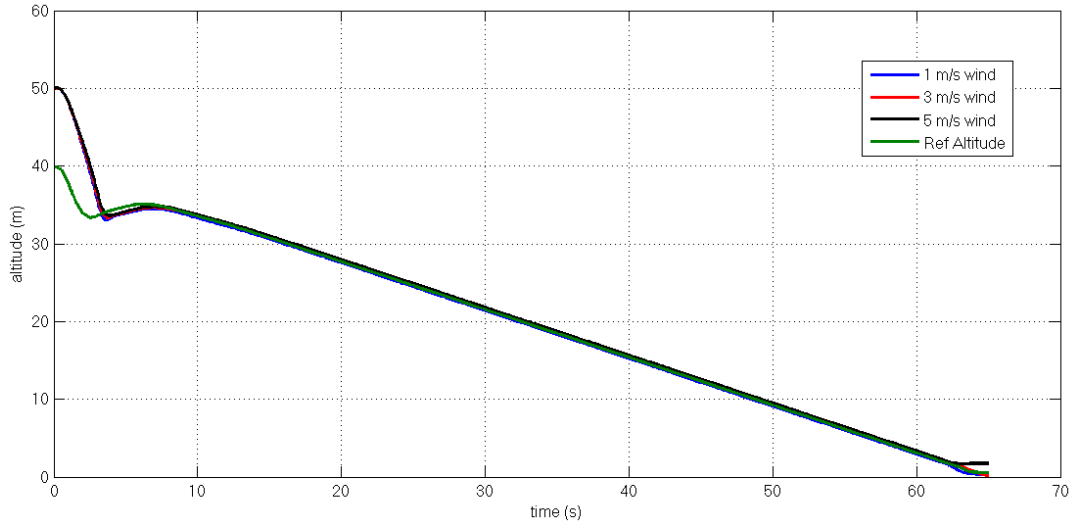

Figure 113 - Altitude response of 2nd SMC with CTG

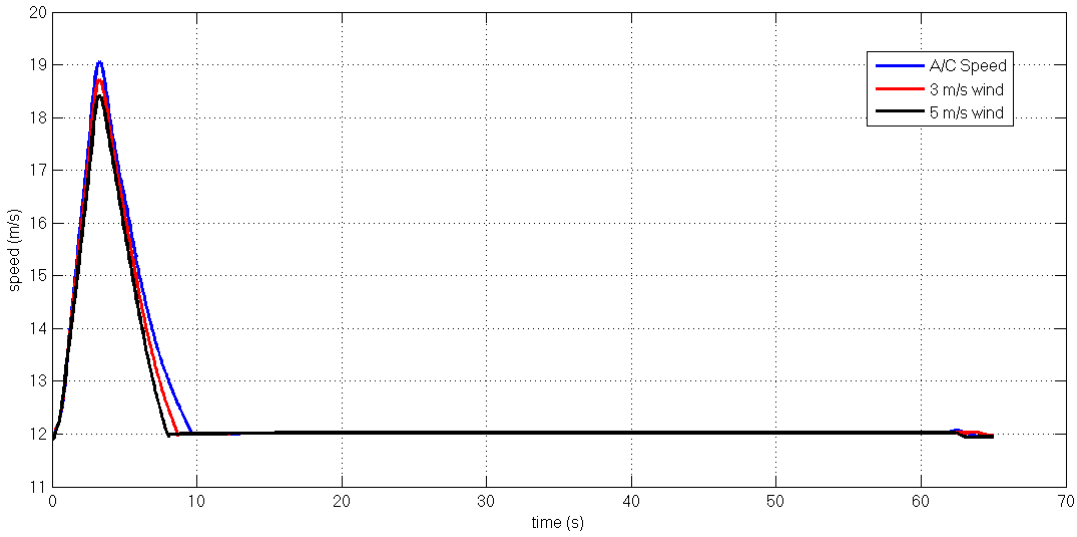

Figure 114 - Speed response of 1st SMC with CTG

2nd sliding mode controller, where the sliding surface has defined as the error of both speed and altitude, has more robust result compared to other SMC. Since the speed is kept under 17 m/s and better tracking performance observed.

NMPC

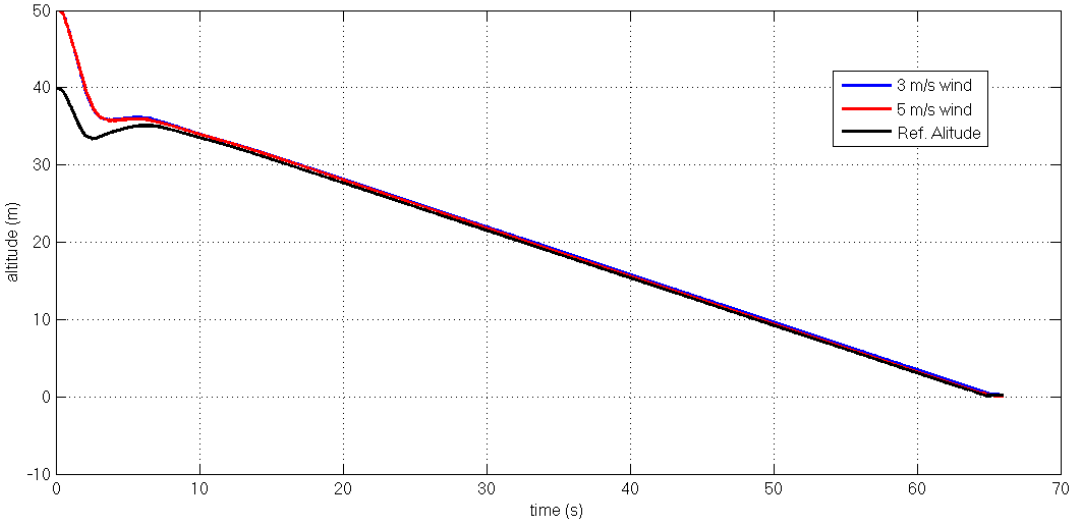

Figure 115 – Altitude response of SMC with CTG

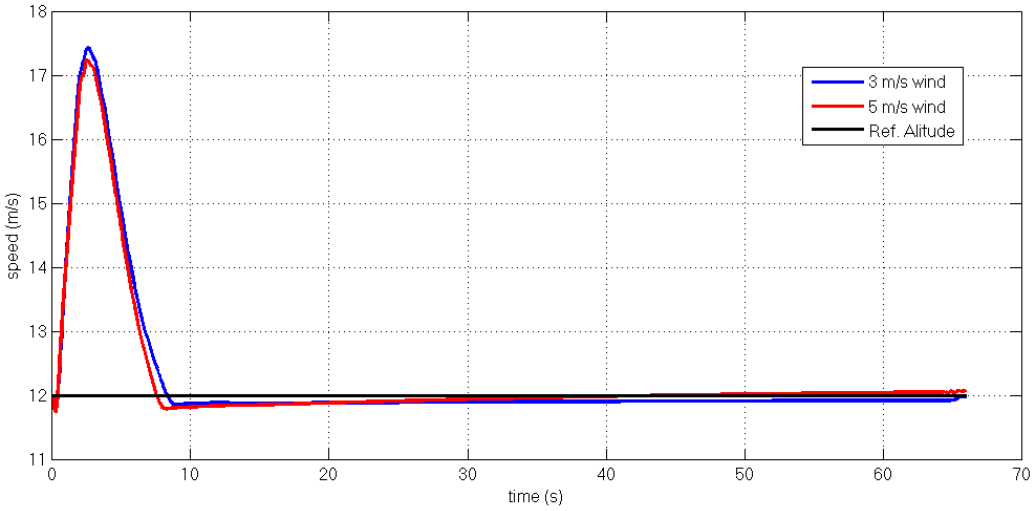

Figure 116 – Speed response of MPC with CTG

The model predictive controller also has the best tracking performance of all controllers in this scenario. Altitude tracking is very good and speed value does not exceed limits and converges to desired speed which is 12 m/s quickly.

5.4 Scenario 4

In this scenario the wind is continuous and coming from 45^{o} ahead of aircraft. So the components of wind consist of some head and cross winds. The controller have to cope with lateral disturbance and airspeed disturbance effects. Results are gives as:

PID with LTG

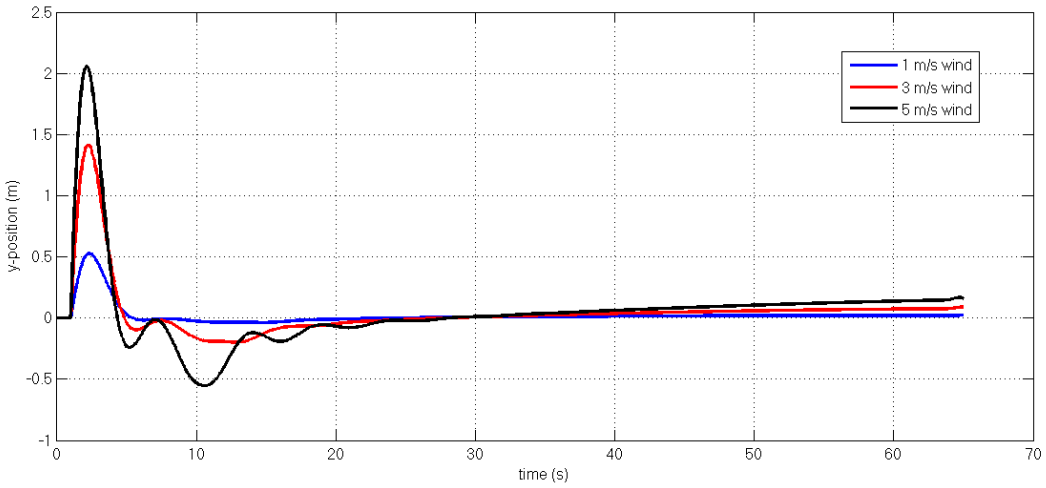

Figure 117 - Y-axis Position Plot for PID with LTG

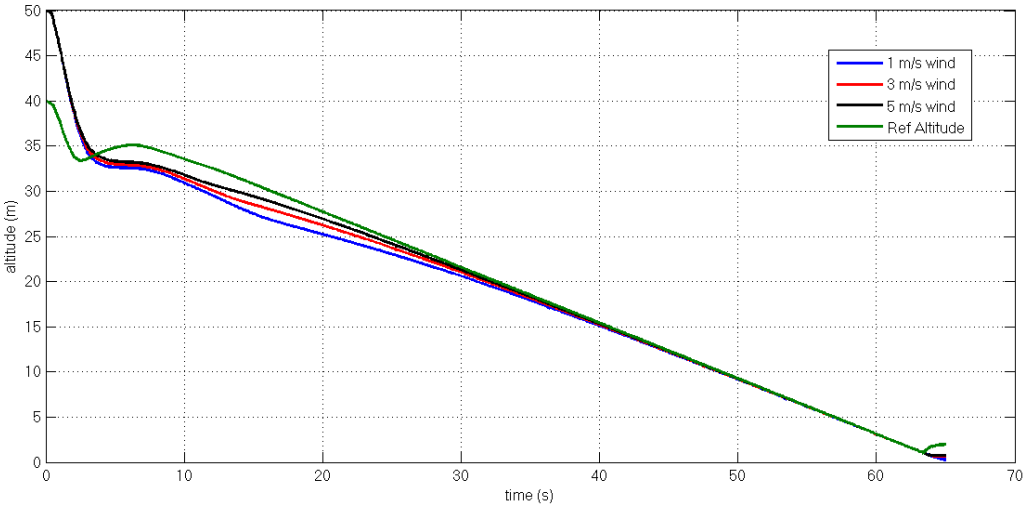

Figure 118 - Altitude response of PID with LTG

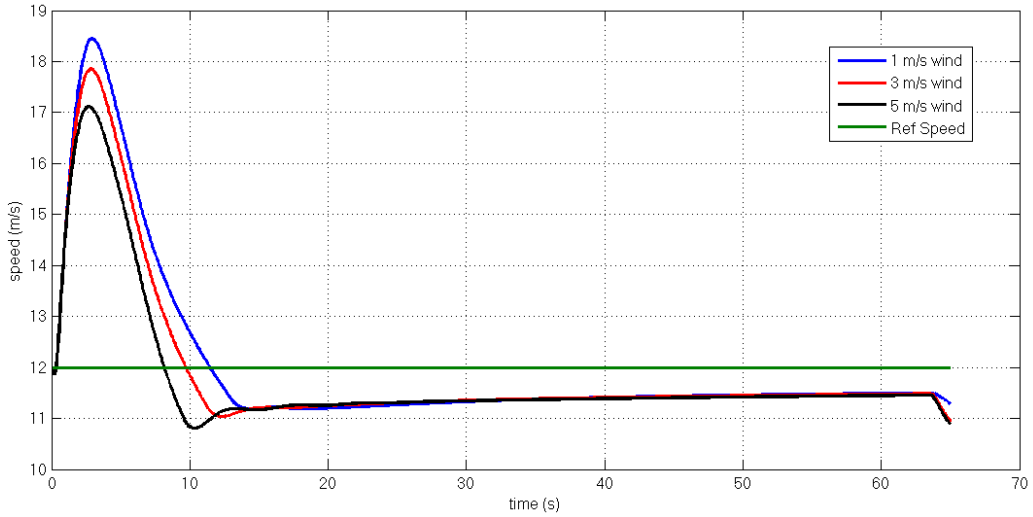

Figure 119 - Speed response of PID with LTG

PID with lateral track guidance manages to keep desired track the speed is in acceptable region and y-position error is small, still at strong winds, lateral oscillations observed which is negative performance criteria.

PID with CTG

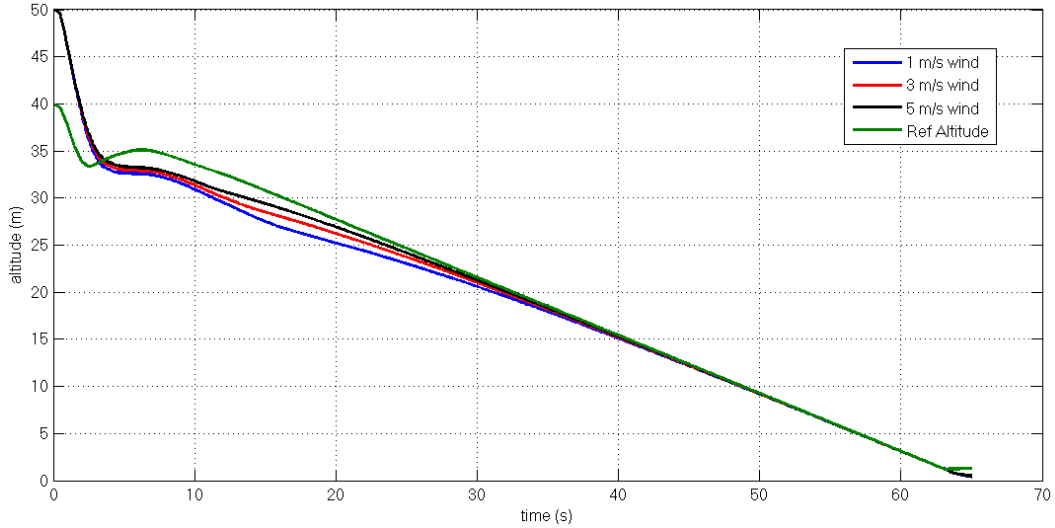

Figure 120 - Altitude response of PID with CTG

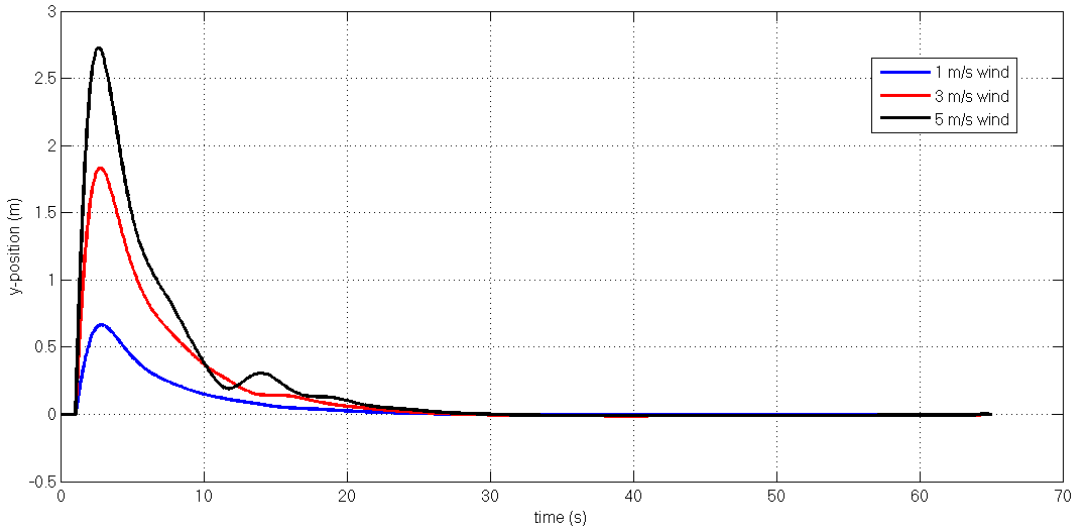

Figure 121 - Y-axis Position Plot for PID with CTG

The lateral response is smoother than PID with LTG but a little slow so if this kind of wind appears at a close point to touch-down, aircraft may not be able to land to runway precisely.

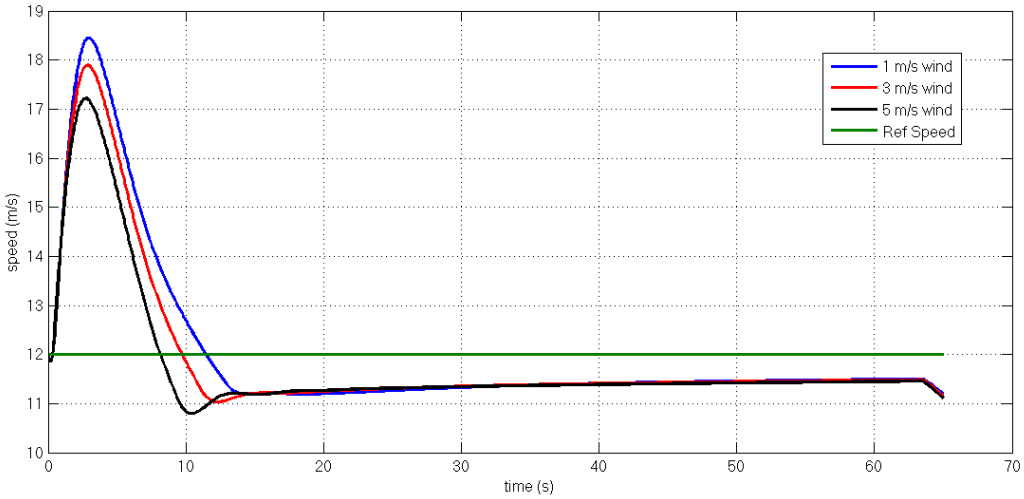

Figure 122 - Speed response of PID with CTG

Controller 1 with CTG is faster with respect to Controller 1 LTG however, it has some amount of overshoot. Still, both of them manages to keep the aircraft on track. It can also be observed that altitude controller is not affected from long-term winds. The same response is expected from all other controllers.

LQT with CTG

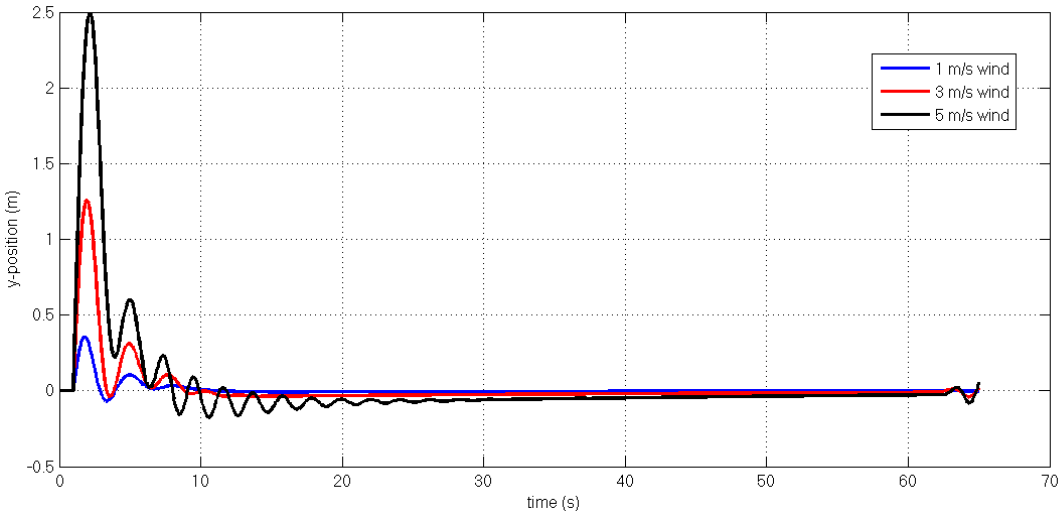

Figure 123 - Y-axis Position Plot for LQT with CTG

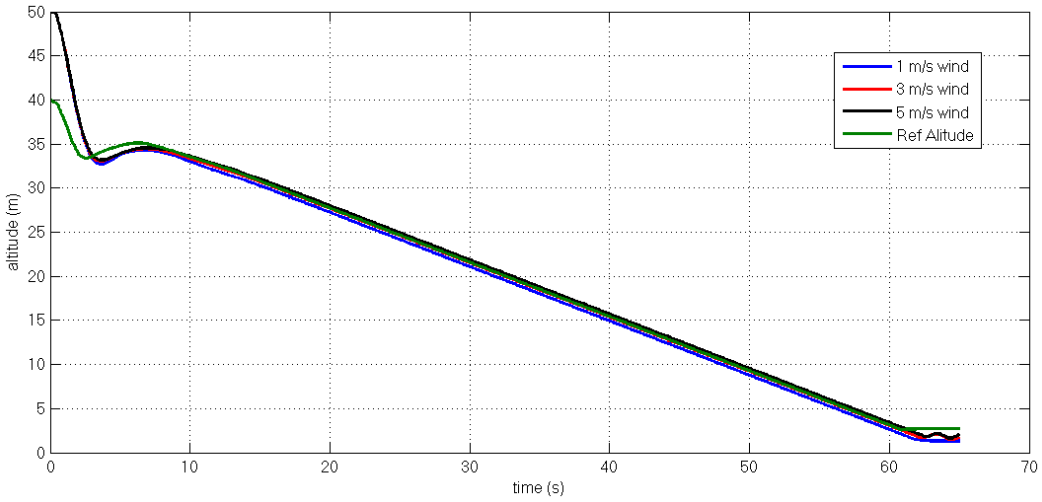

Figure 124 – Altitude response of LQT with CTG

Settling time of LQR is much better than PID controllers. The altitude tracking is smoother and robust. But at strong winds lateral response of aircraft is oscillatory. Which imlies that stronger winds coming 45º may cause stability problems.

1st SMC with CTG

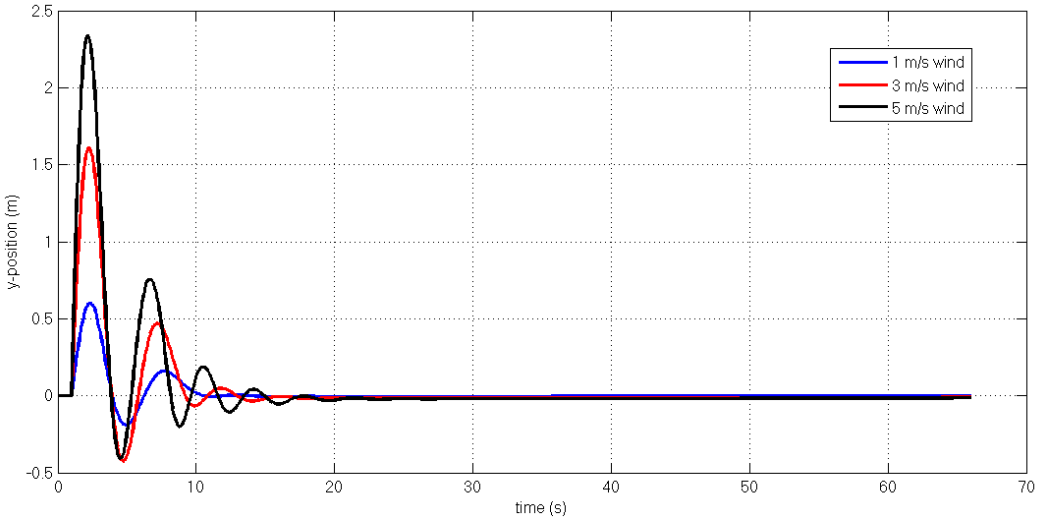

Figure 125 - Y-axis Position Plot for 1st SMC with CTG

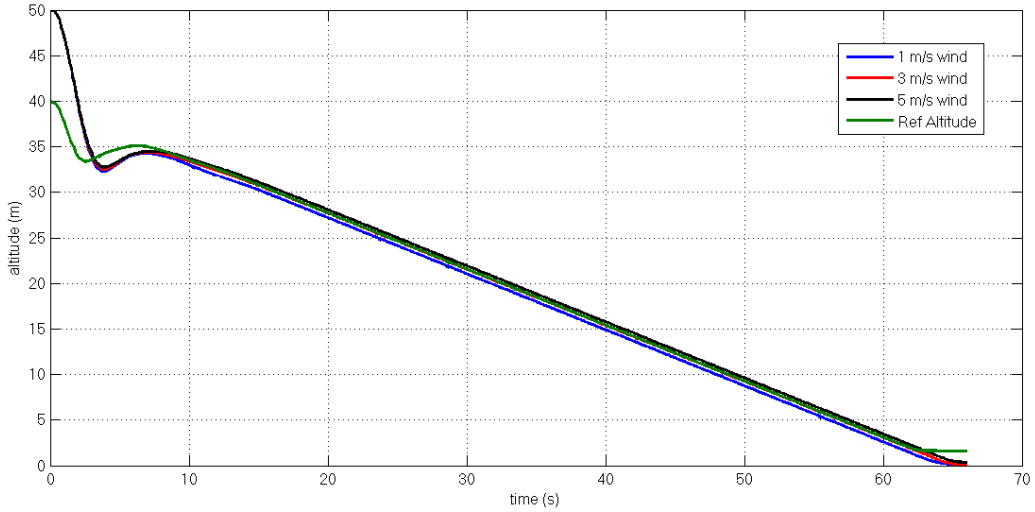

Figure 126 - Altitude response of 1st SMC with CTG

Results Show that LQT with CTG has good lateral response but a small steady state error is observed where SMC with CTG has no Steady state error. Altitude responses of both controllers are similar and sufficient for a safe landing.

2nd SMC with CTG

Since the lateral responses are very similar to 1st SMC, they are not presented. The altitude and speed plots of controller are given as:

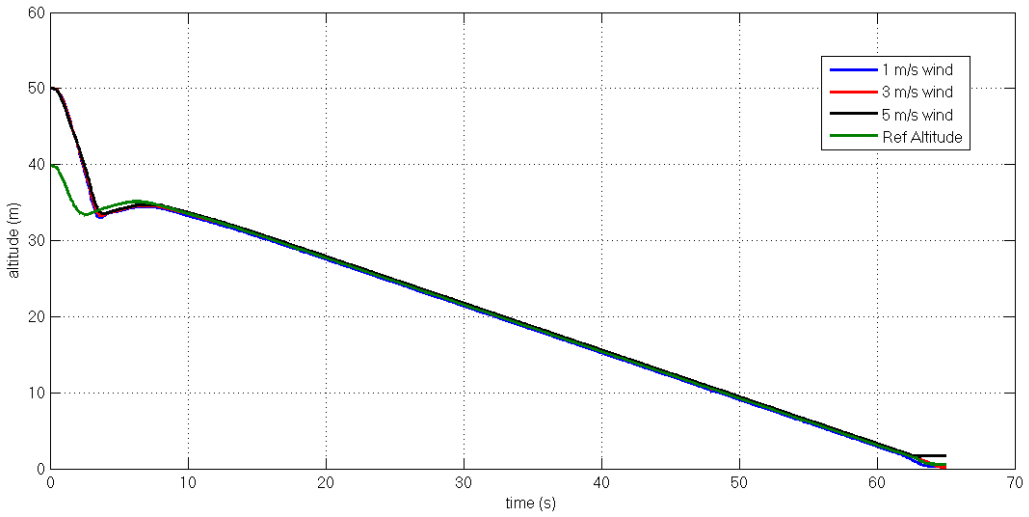

Figure 127 - Altitude response of 2nd SMC with CTG

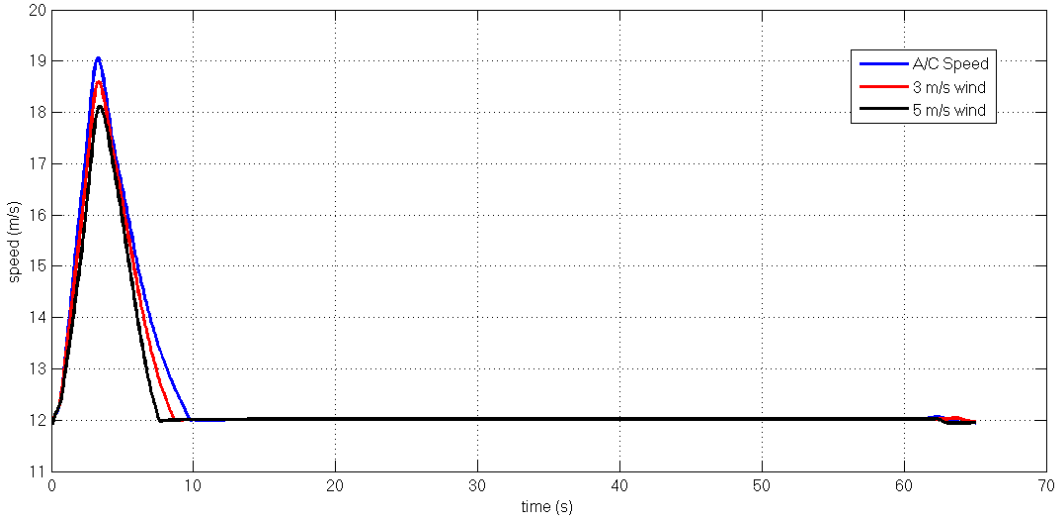

Figure 128 – Speed response of 2nd SMC with CTG

As expected the altitude response of the aircraft is much slower than other SMC, but the speed overshoot is also much smaller. Tracking performance is sufficient enough under winds coming 45^{o} ahead.

NMPC

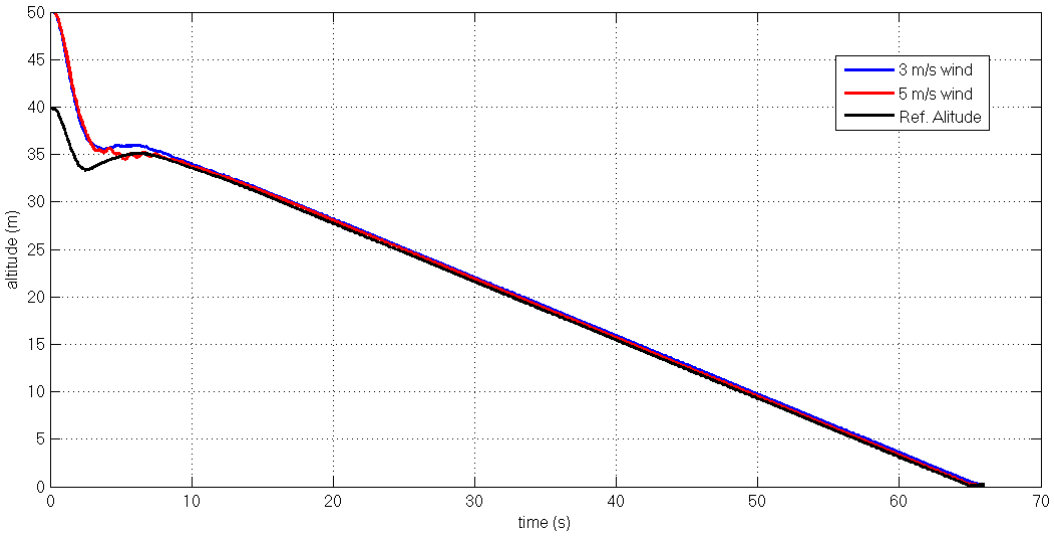

Figure 129 – Altitude response of NMPC

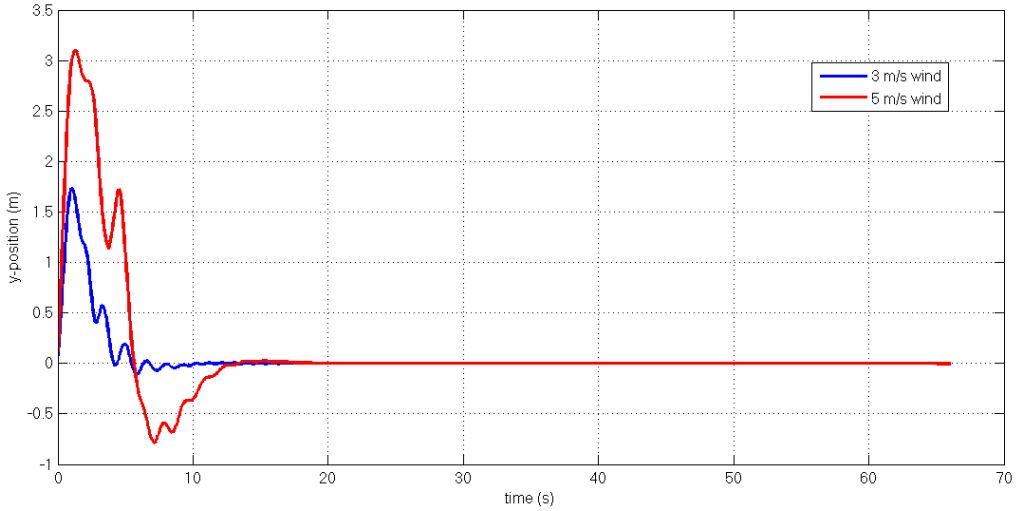

Figure 130 – Y-position plot for NMPC

Model predictive controller moves to desired altitude quickly and tracks it successfully, but the lateral response is also better compared to other controllers, since there's no oscillating path and steady state error.

5.5 Scenario 5

In this scenario, wind is coming from 135 degree (North-west) and it is continuous. Here the aircraft must handle both lateral disturbance and relative speed drop, caused by tail wind component.

PID with LTG

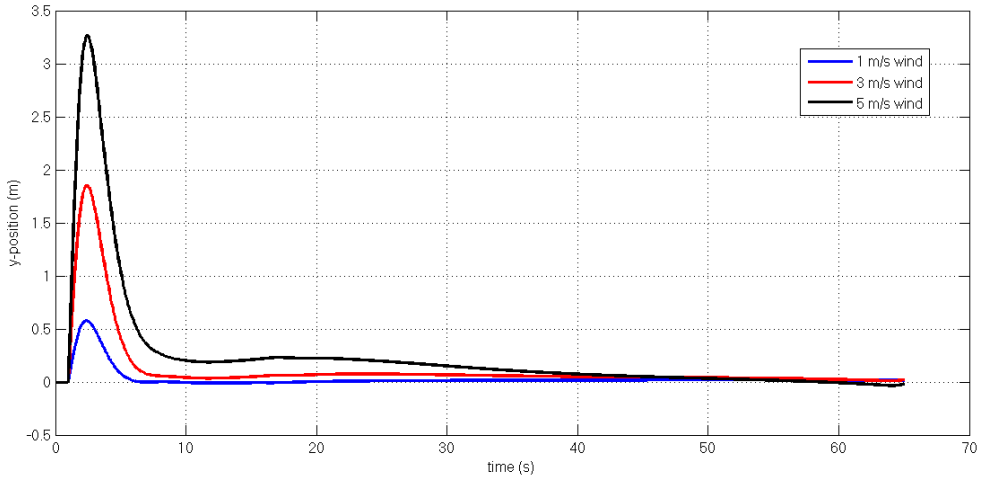

Figure 131 - Y-axis Position Plot for PID with LTG

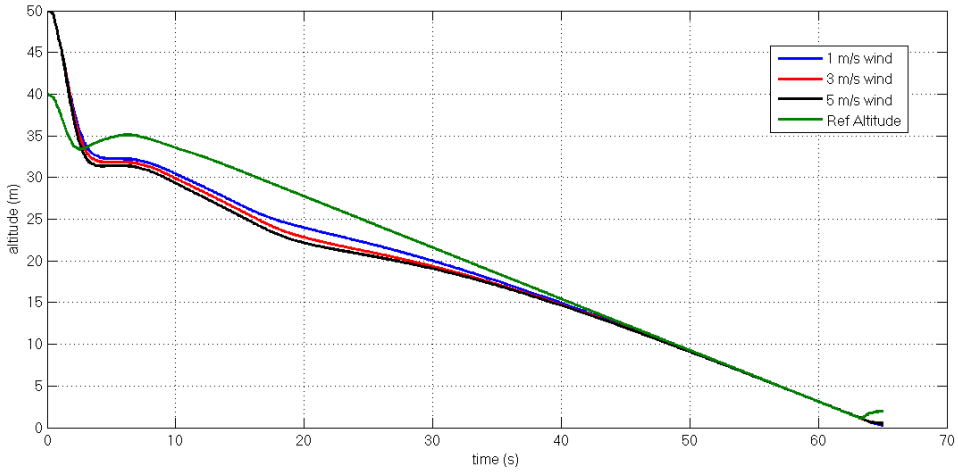

Figure 132 – Altitude response of PID with LTG

PID with LTG closely manages to land safely, if the wind speed exceeds its current speed the controller will no longer be able to maintain the altitude of aircraft which will cause a possible crash. As in the direct tail wind scenario, very long time is required to get in to desired track of landing.

PID with CTG

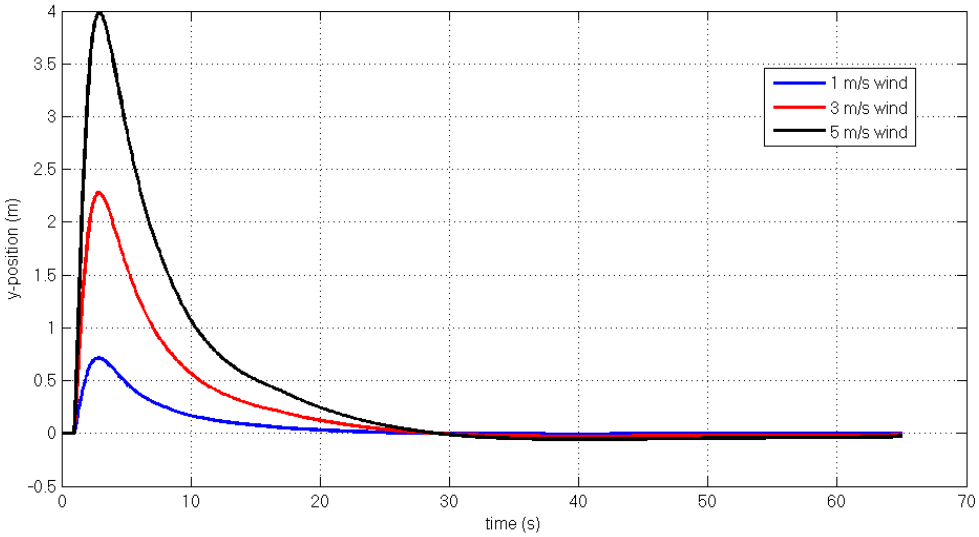

Figure 133 - Y-axis Position Plot for PID with CTG

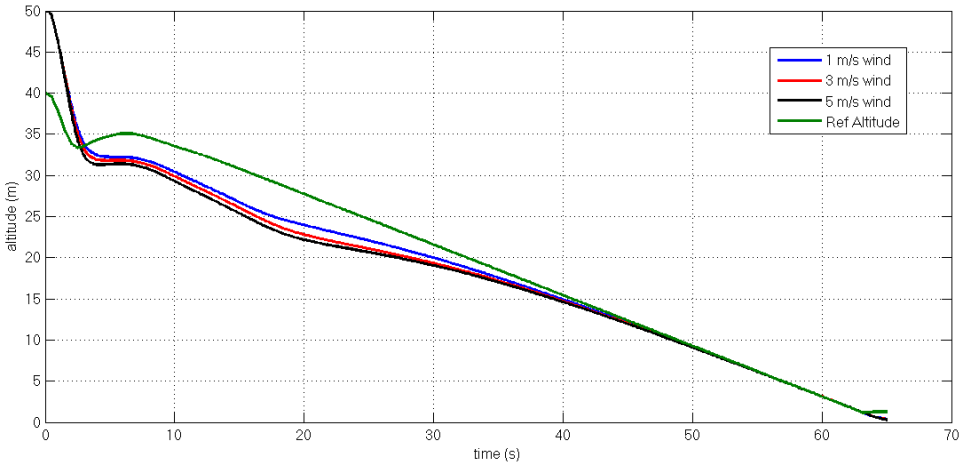

Figure 134 - Altitude response of PID with CTG

It can be observed from the figures that Controller 1 with LTG is not adequate for winds coming from southeast. The result would be the same if the wind was coming from south-west. It is seen that the speed of the controller/guidance is slow regarding Controller 1 with CTG, hence not suitable for landing under such winds.

LQT with CTG

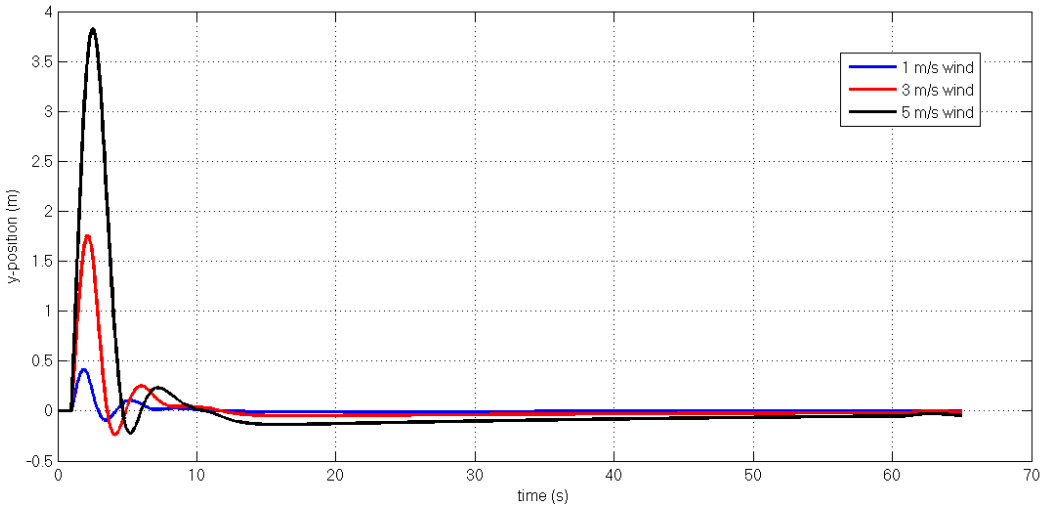

Figure 135 – Y-axis Position Plot for LQT with CTG

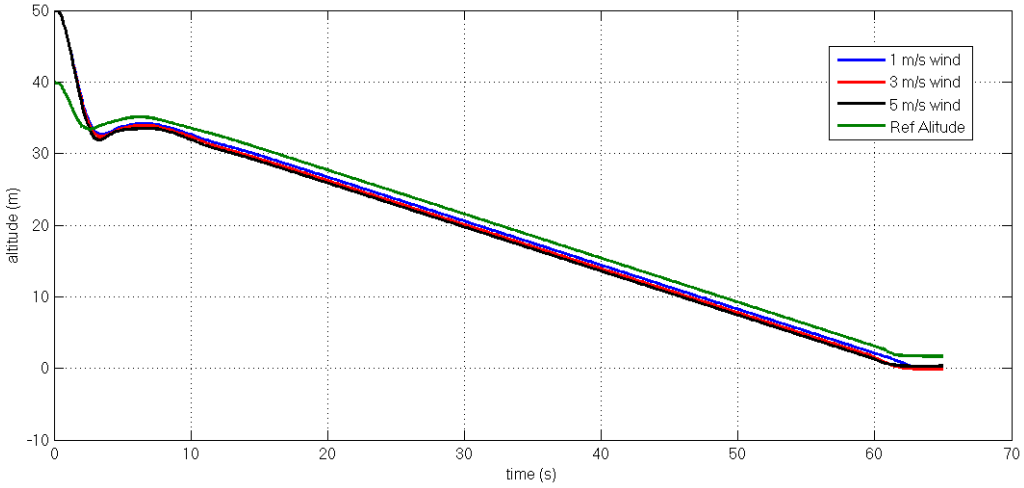

Figure 136 – Altitude response of LQT with CTG

With a small offset in altitude response LQT also lands the aircraft safely. It is observed that the altitude offset gets bigger with the wind disturbance so under strong continuous winds LQT will not be a logical solution unless some modifications (like adding a differential) are made.

SMC with CTG

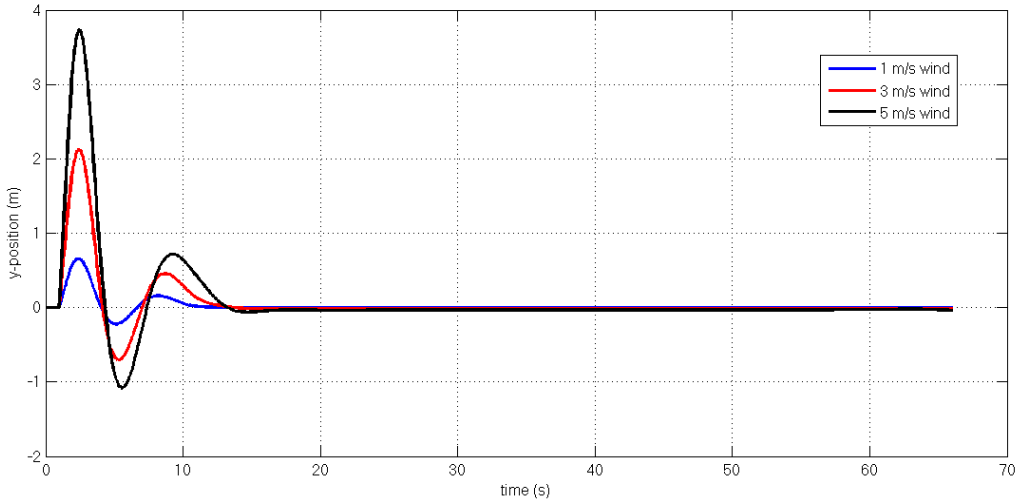

Figure 137 - Y-axis Position Plot for 1st and 2nd SMC with CTG

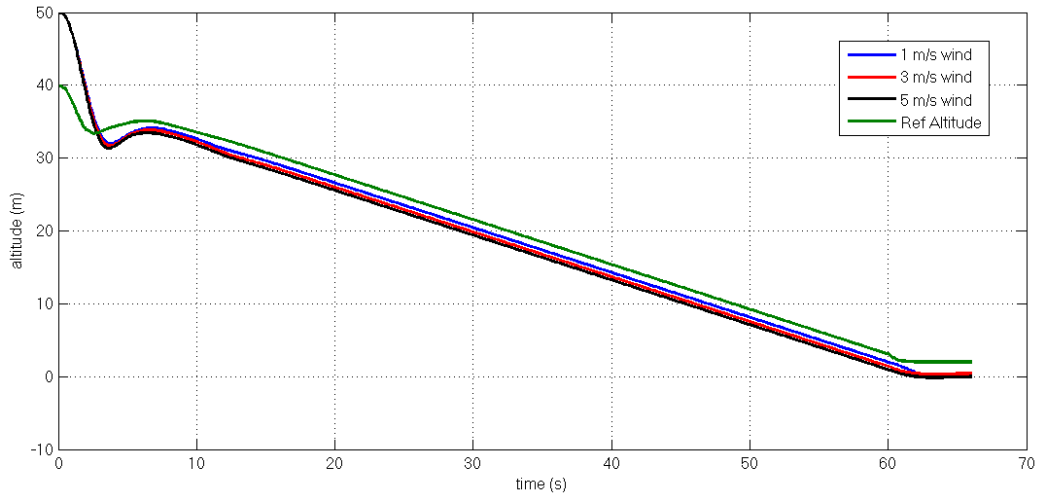

Figure 138 - Altitude response of SMC with CTG

Very similar plots are obtained with SMC with previous controller LQT. The offset problem still exists but the lateral response is more accurate compared to LQT.

NMPC

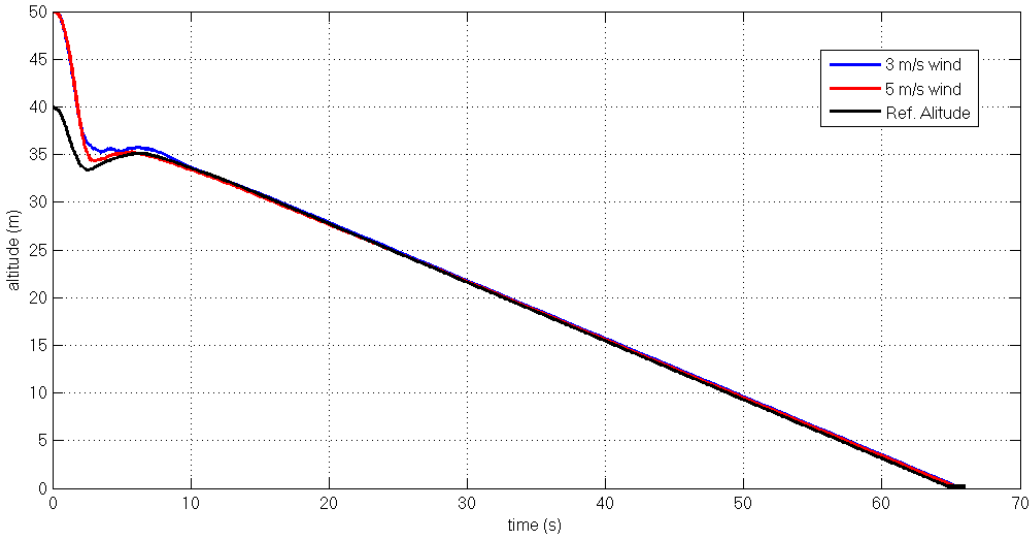

Figure 139 – Altitude response of NMPC

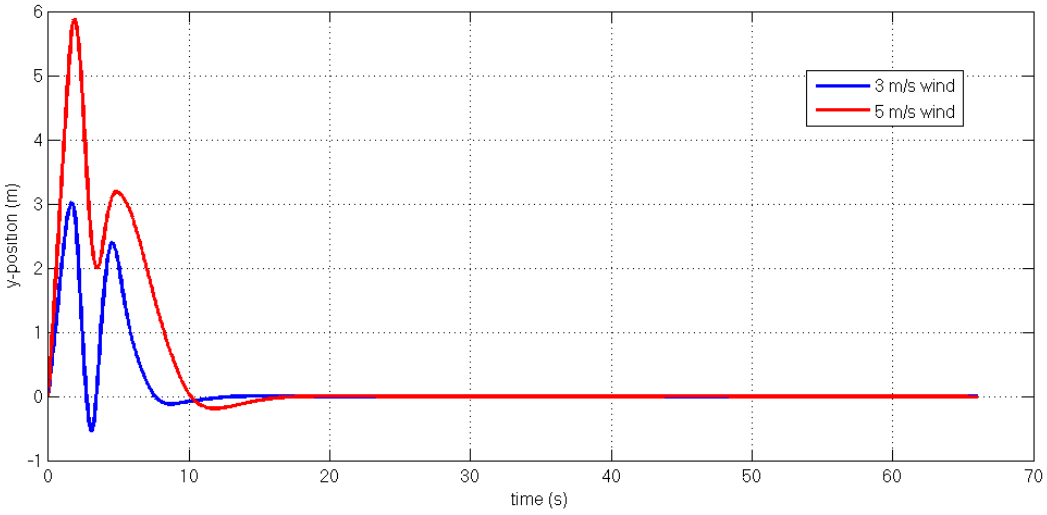

Figure 140 – Y-position plot of NMPC

As seen in the Figures 139 and 140, the general tracking performance of MPC is better compared to other controllers tested. The altitude response is robust and no offset observed. Y-position correction characteristics is similar to SMC and LQT and no steady state error is observed.

5.6 Scenario 6

A gust wind (not continuous) with duration 1 sec. is applied close to touchdown point. The performances of controllers are investigated in order to have a safe landing. Since Telemaster UAV is a small aircraft the lateral tolerances of y-position are defined as $\pm$ 1m. It's expected from controllers that to fix heading and y-position before touch-down occur.

PID with LTG

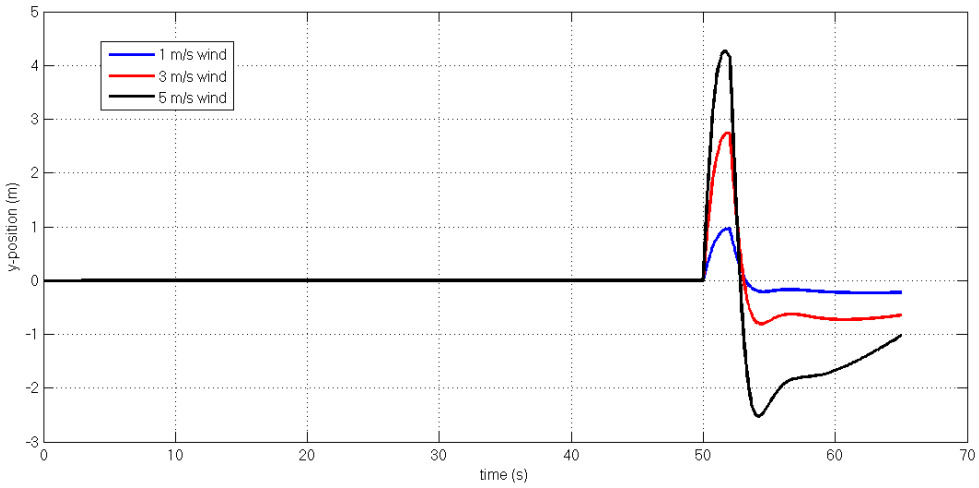

Figure 141 - Y-axis Position Plot for PID with LTG

Very poor lateral response observed with Lateral track guidance, especially winds stronger than 3 m/s. The y-position error is very big.

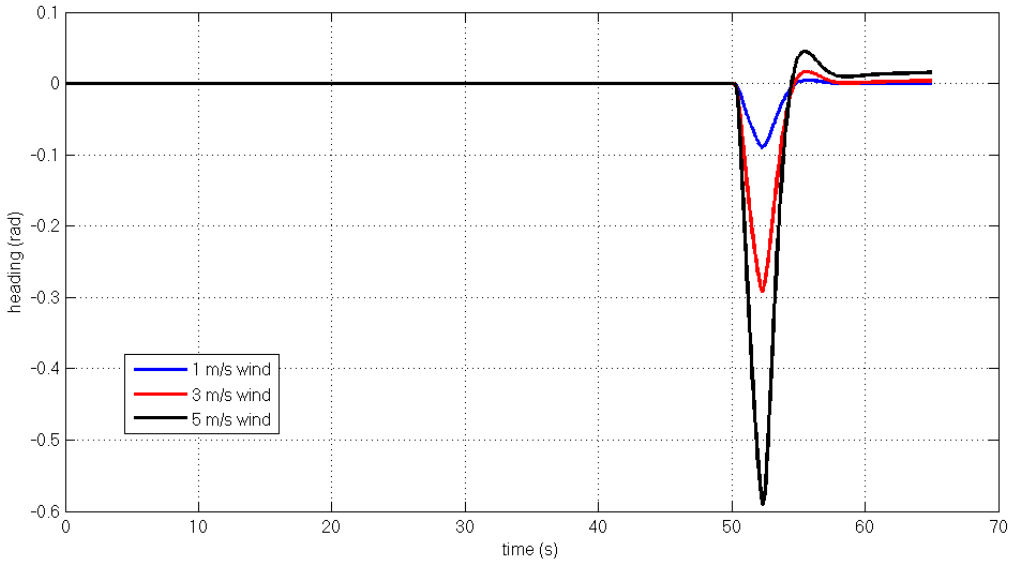

Figure 142 - Heading Plot for PID with LTG (in radians)

PID with LTG has the poorest lateral performance compared to other controller setups. 5m/s crosswind couldn't be tolerated in time.

PID with CTG

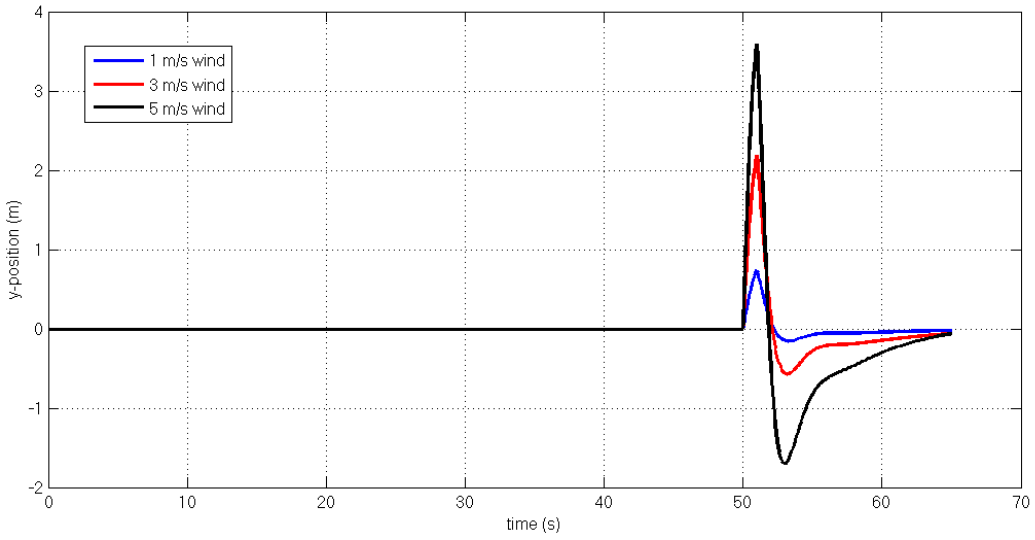

Figure 143 - Y-axis Position Plot for PID with CTG

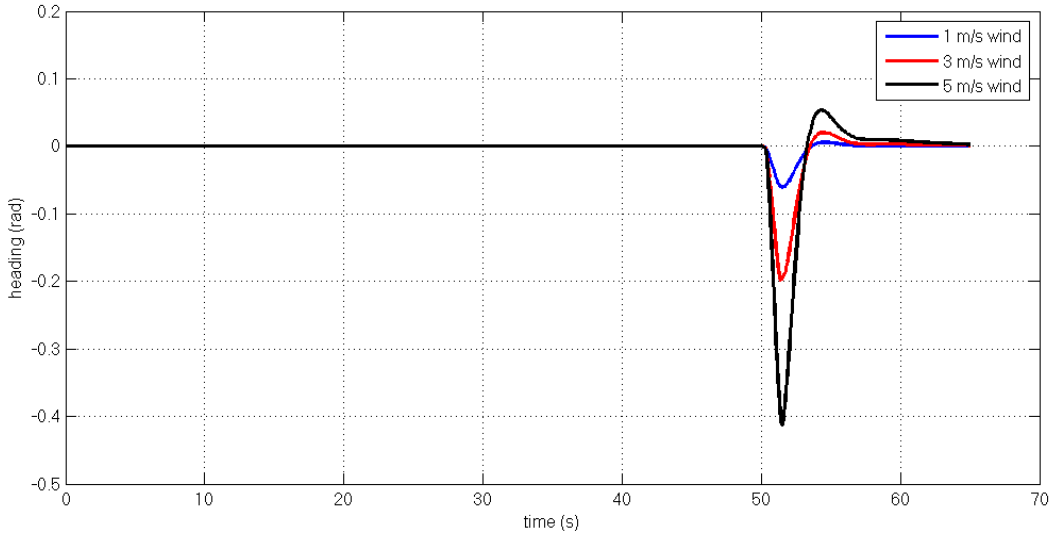

Figure 144 - Heading Plot for PID with CTG (in radians)

Results show that, the PID controller with CTG is sufficient enough to land the aircraft in wind gusts at a time close to the touchdown. The y-position is in the acceptable tolerances. Also the heading of the aircraft is smaller than 0.05 radians which implies us that this landing is successful. The response of both PID controller modified by LTG and CTG has similar y-position and heading responses, but PID with CTG has less y-position error.

LQT with CTG

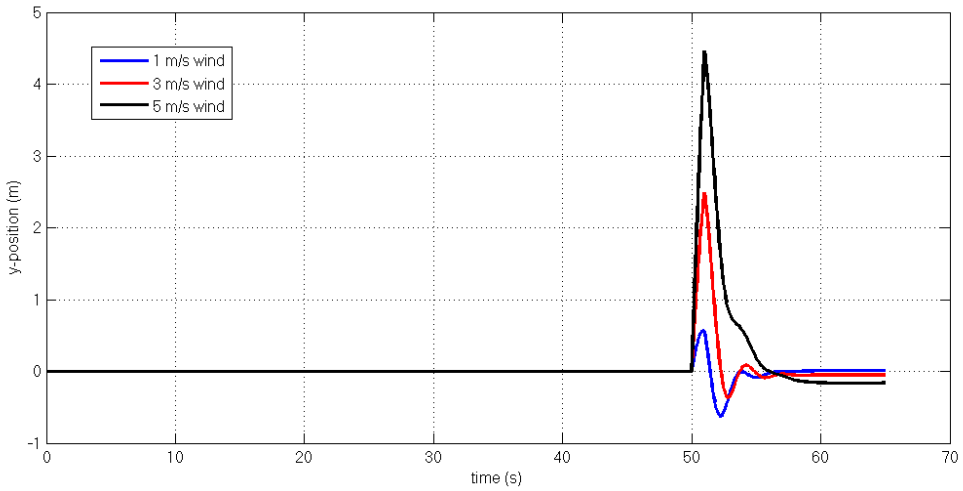

Figure 145 - Y-axis Position Plot for LQT with CTG

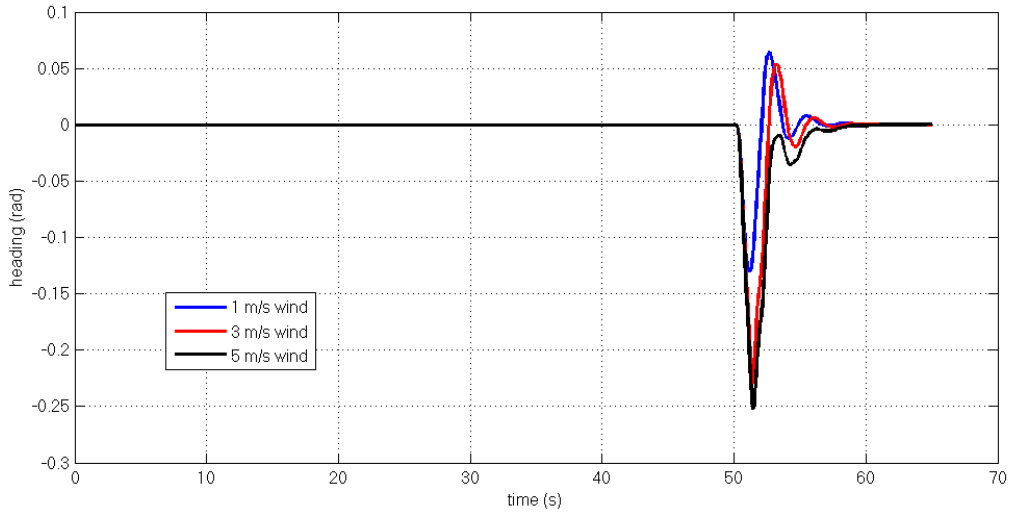

Figure 146 - Heading Plot for LQT with CTG (in radians)

LQT with CTG has a fast response but an offset can be observed in the y-position track. But it is still superior to PID controllers since the heading of aircraft is almost zero in all wind speed cases.

SMC with CTG

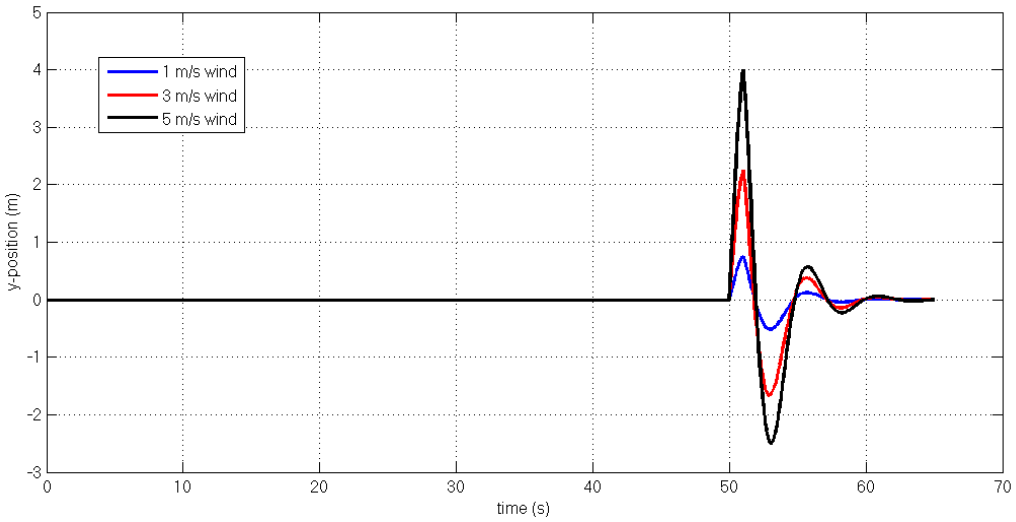

Figure 147 - Y-axis Position Plot for SMC with CTG

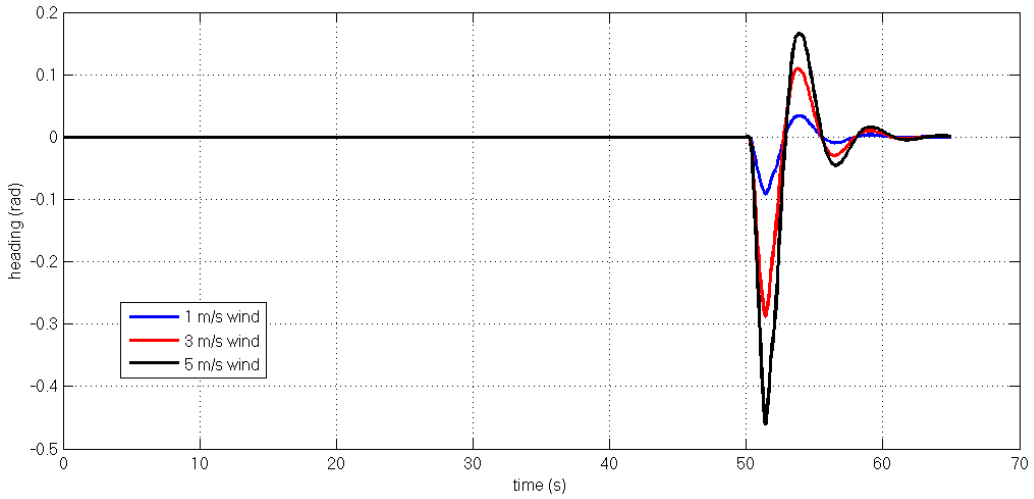

Figure 148 - Heading Plot for SMC with CTG (in radians)

SMC with CTG is a little bit slower than LQR but it has no steady state error in the y-position plot. The controller is good enough for landing under cross winds.

NMPC

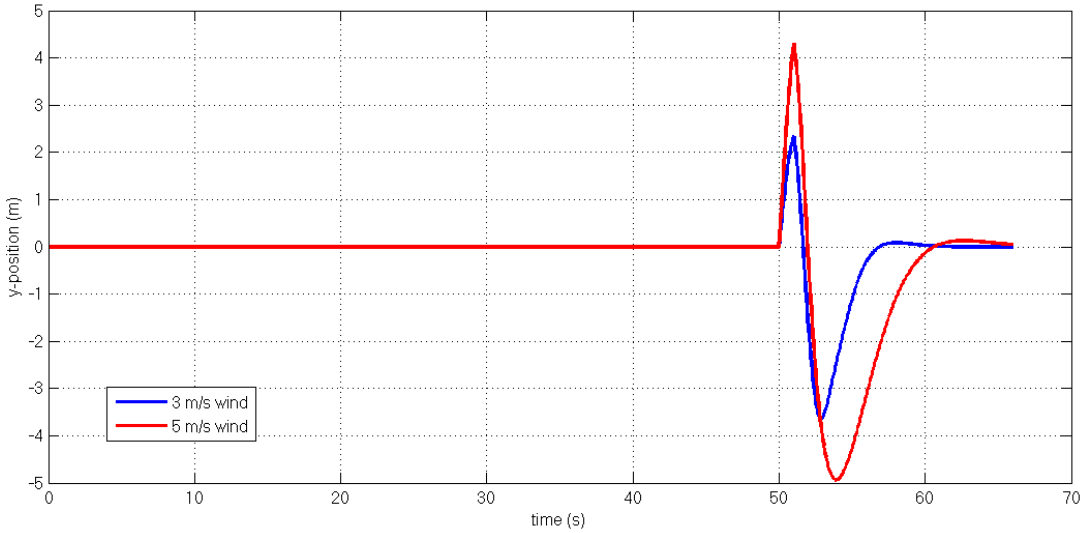

Figure 149 – Y-position plot of NMPC

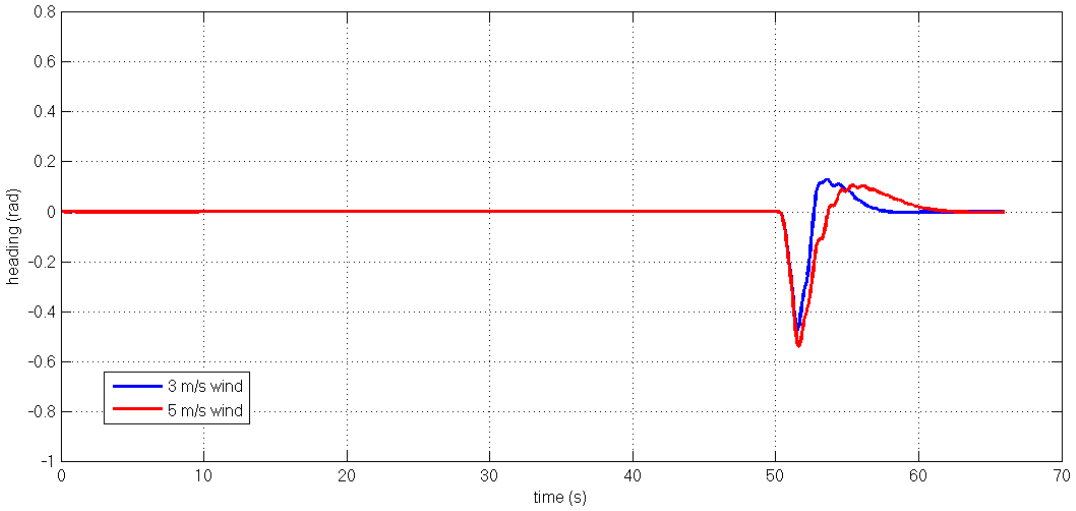

Figure 150 – Heading plot of NMPC (in radians)

The response of MPC is not quicker but robust than any other controller tested. The tracking performance in lateral axes is good and no offset observed.

5.7 Scenario 7

In this scenario, 2 gust cross-winds with 1sec duration is applied just before touch-down. The LQT, SMC and NMPC are tested, since they have better performance. It is expected that the controllers to get in the desired lateral trajectory.

LQT with CTG

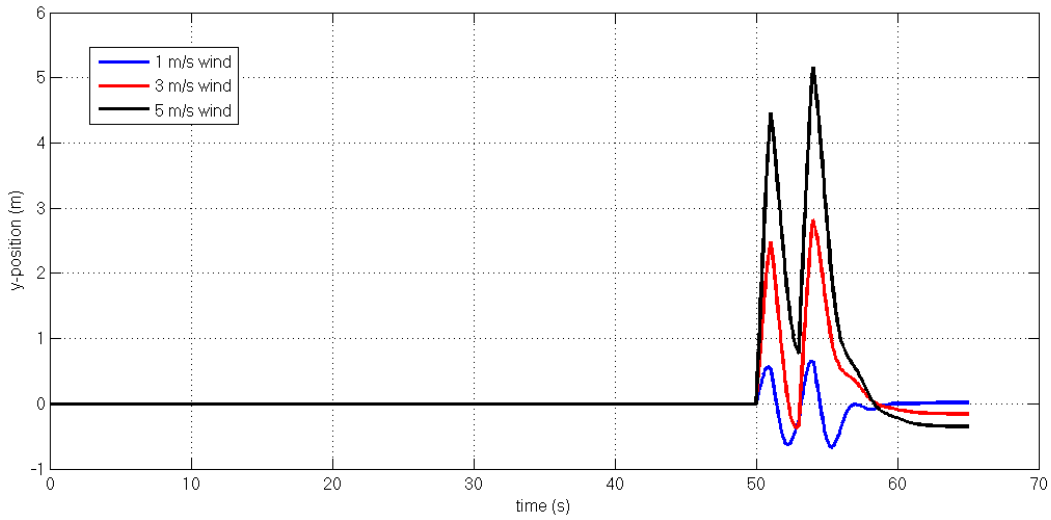

Figure 151 - Y-axis Position Plot for LQT with CTG

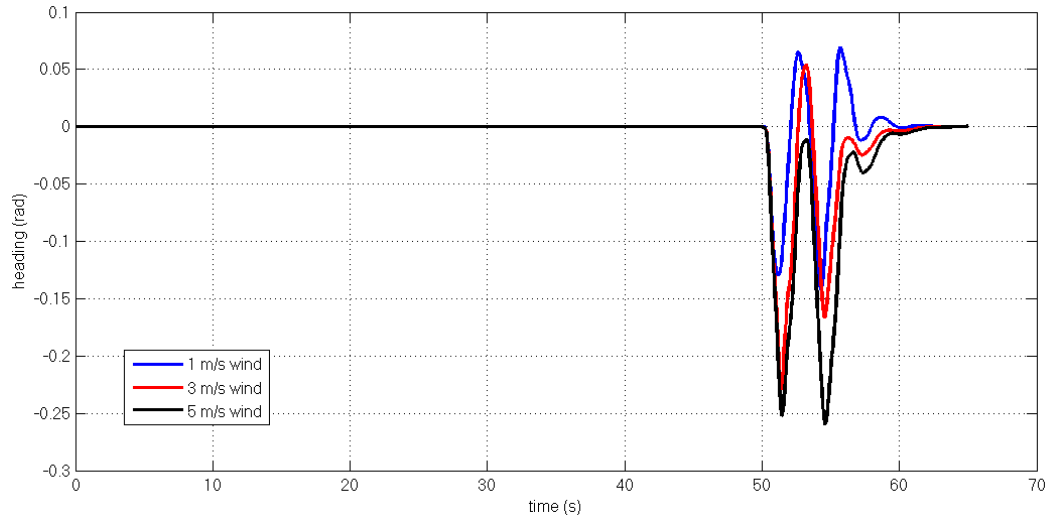

Figure 152 - Heading Plot for LQT with CTG (in radians)

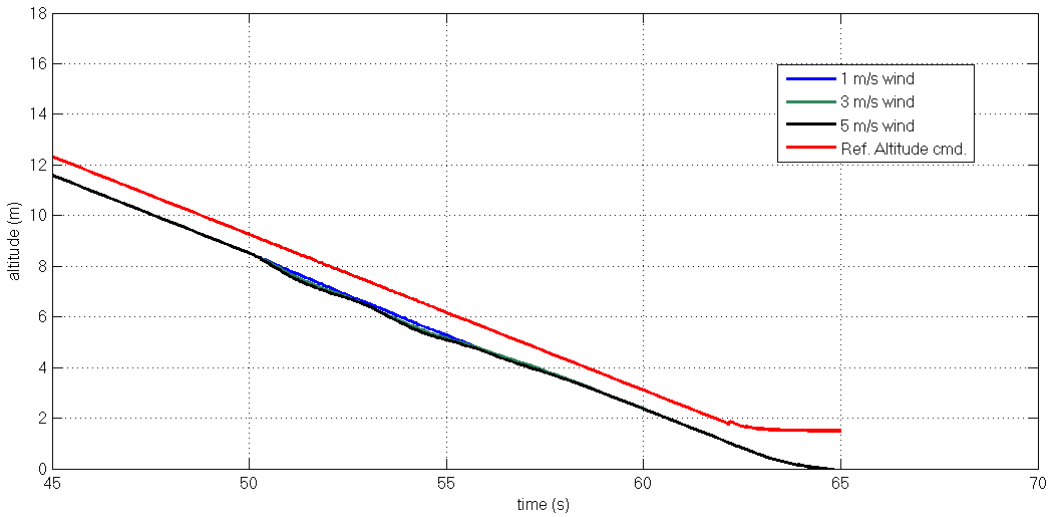

Figure 153 - Altitude Plot for LQT with CTG (15 m/s wind)

LQT with CTG handles with 2 gust winds. The altitude drop is not in dangerous levels but there is an offset in y-position. Still it is appropriate for landing.

1st SMC with CTG

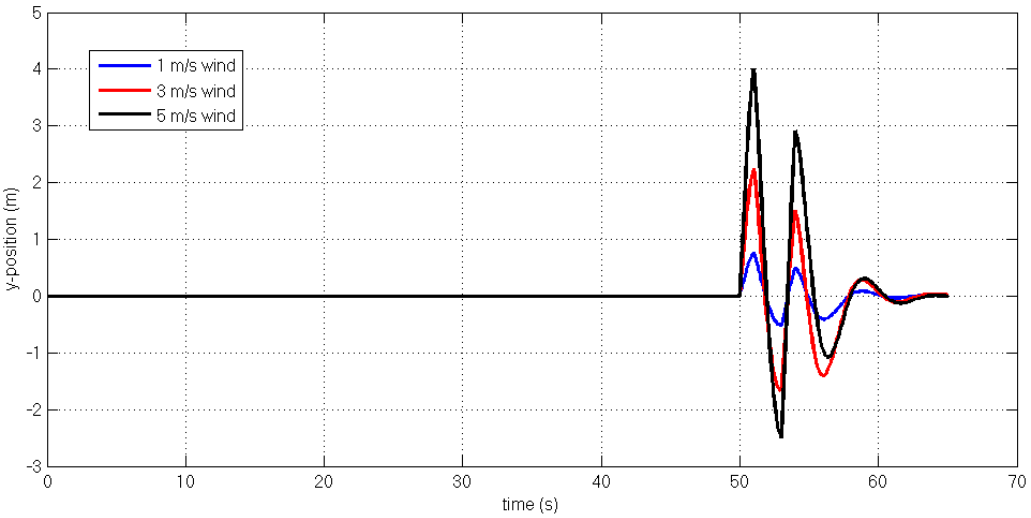

Figure 154 - Y-axis Position Plot for 1st SMC with CTG

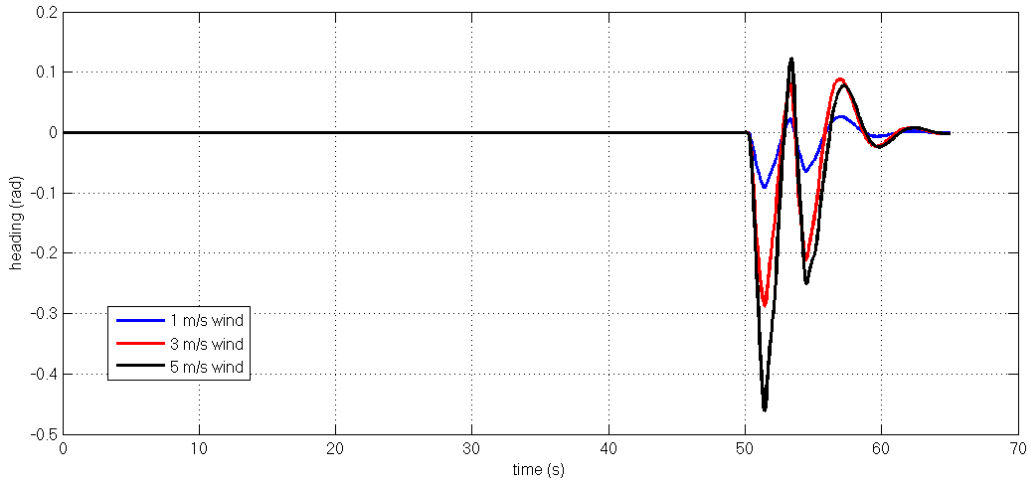

Figure 155 - Heading Plot for 1st SMC with CTG (in radians)

SMC manages to drive the aircraft in desired lateral track better than LQT, however altitude drops are higher which could be dangerous while landing. Still the landing action can be performed without any accidents.

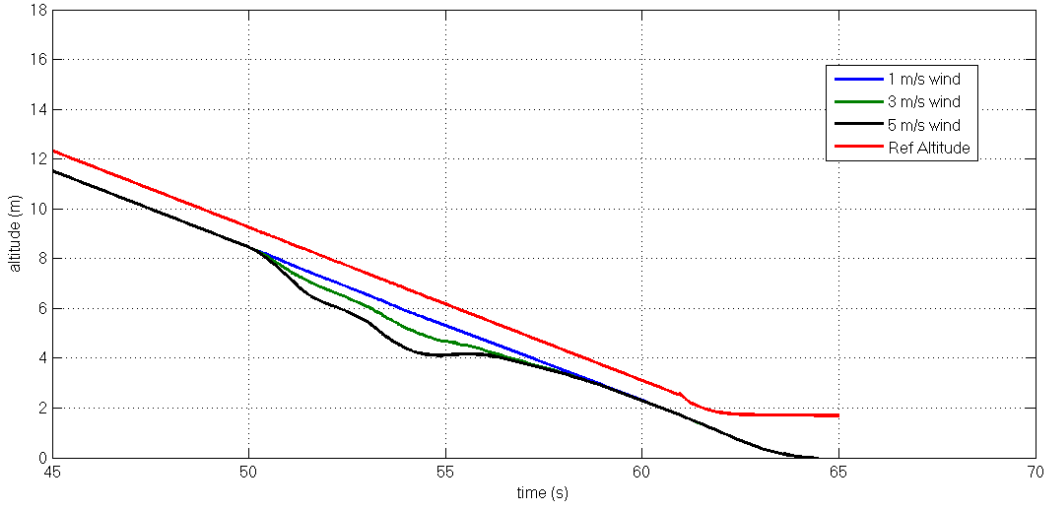

Figure 156 - Altitude response of 1st SMC with CTG

2nd SMC with CTG

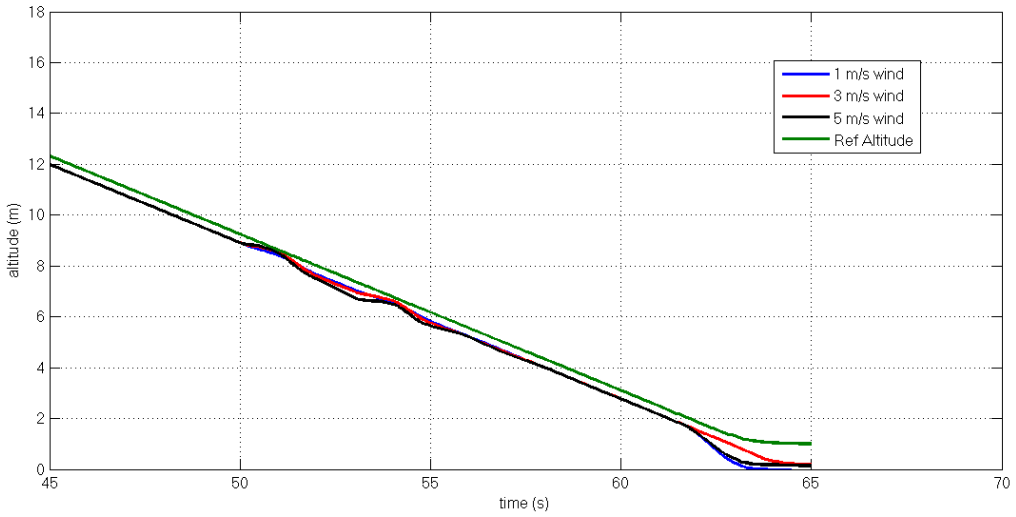

Figure 157 – Altitude response of 1st SMC with CTG

2nd SMC has very similar lateral response like 1st SMC, so it is not presented here. The altitude drops are much smaller which will make this controller better than 1st SMC while landing.

NMPC

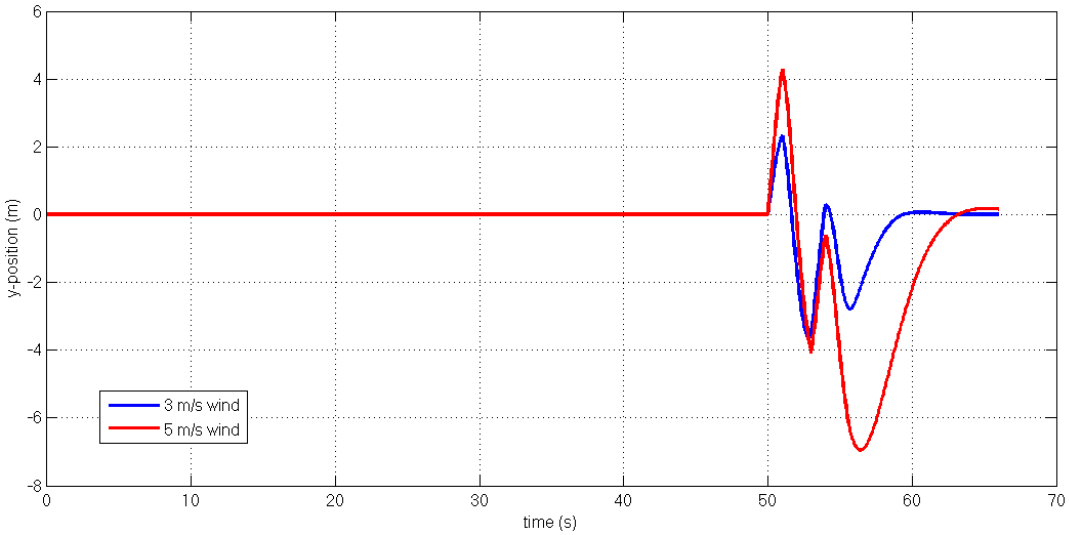

Figure 158 – Y-position plot of NMPC

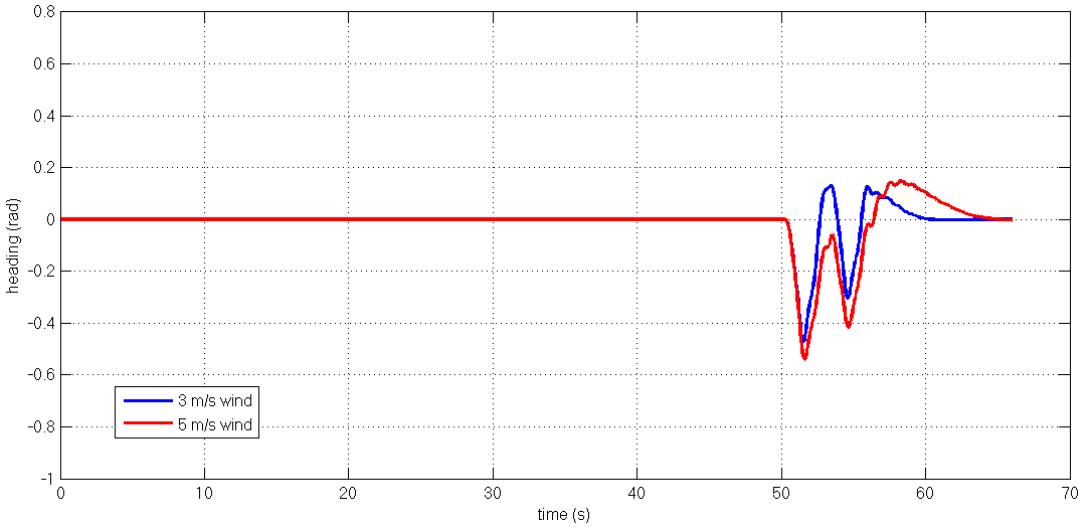

Figure 159 – Heading plot for NMPC (in radians)

Heading and y-position error of NMPC is less oscillatory, but a little slow, since lateral maneuvers required longer prediction horizons. Eventually, NMPC manages to get aircraft in desired track robustly. And altitude drops can compete with 2nd SMC which has very good altitude results.

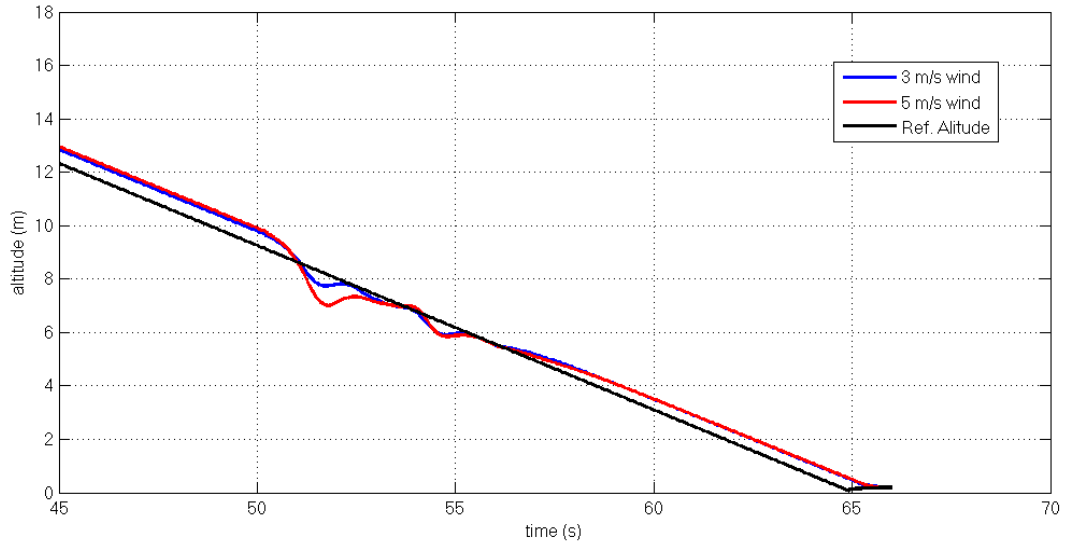

Figure 160 – Altitude response of NMPC

5.8 Scenario 8

LQT, SMC and NMPC are analyzed with three gust winds with duration 1 sec. where the first two are in same but the last gust is in opposite direction just before touch-down, which makes this scenario a challenging one.

LQT with CTG

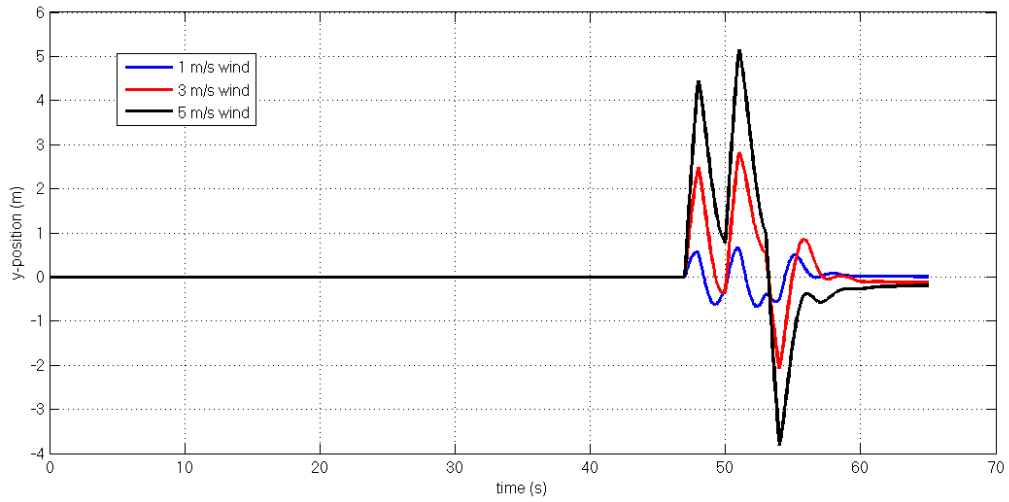

Figure 161 - Y-axis Position Plot for LQT with CTG

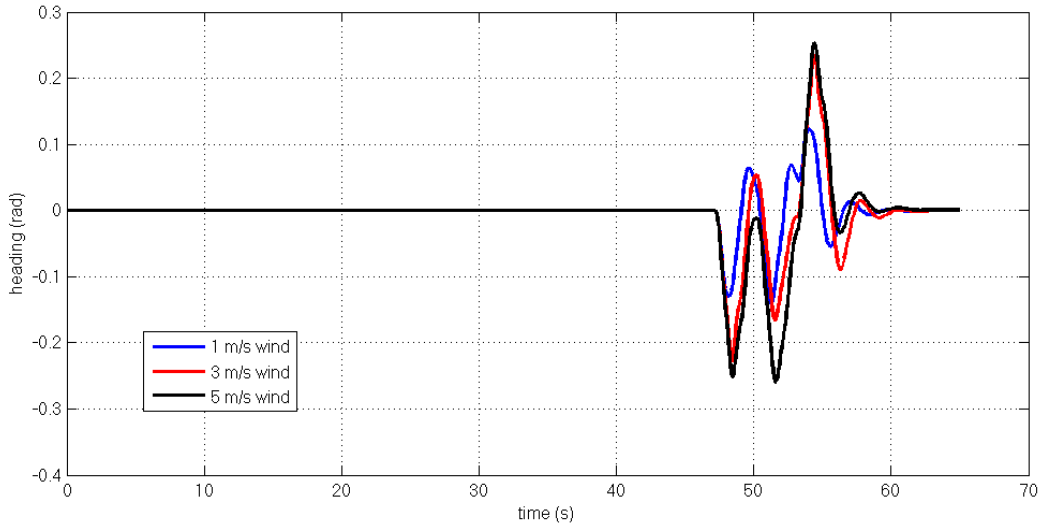

Figure 162 - Heading Plot for LQT with CTG (in radians)

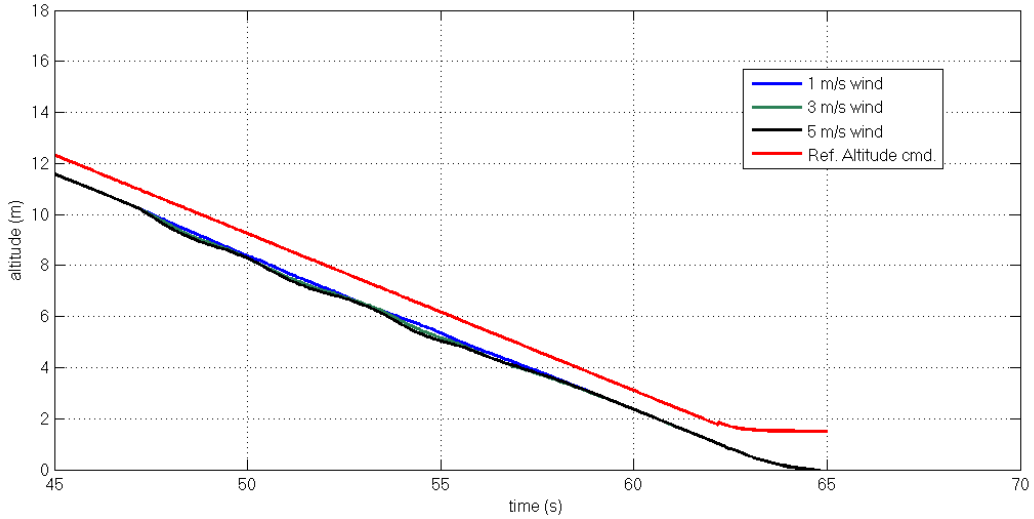

Figure 163 - Altitude response of LQT with CTG

LQT manages to keep aircraft in desired track for altitude and lateral, like in previous case (two gust winds coming in same direction). The altitude drop is very low which makes LQT the best controller in this scenario. It is observed that LQT has very good performances under instant disturbances, but under continuous disturbances there is a big chance that we will see a steady state error.

1st SMC with CTG

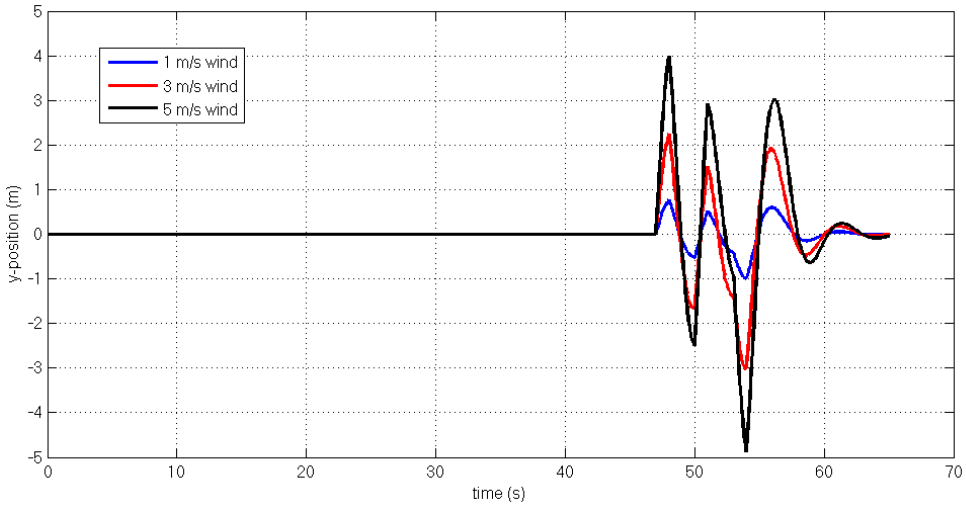

Figure 164 - Y-axis Position Plot for 1st SMC with CTG

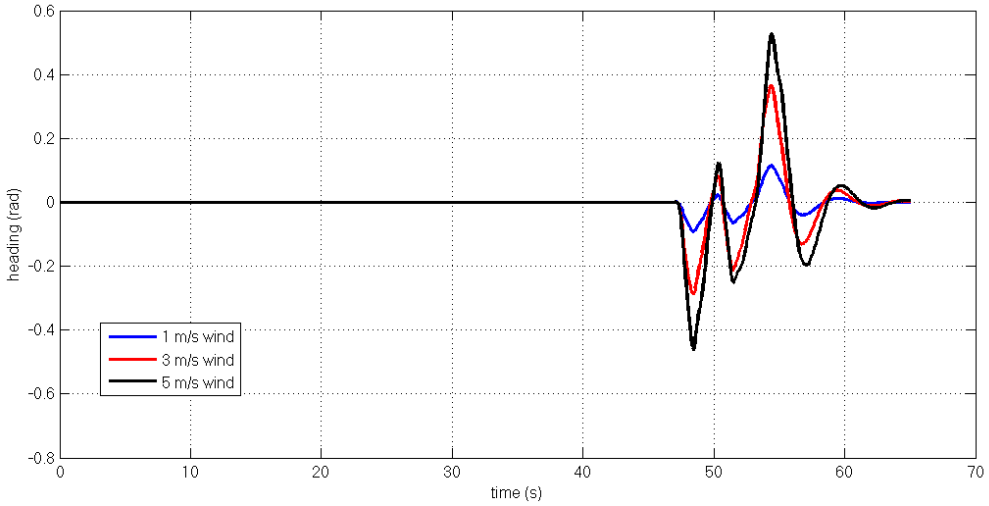

Figure 165 - Heading Plot for 1st SMC with CTG (in radians)

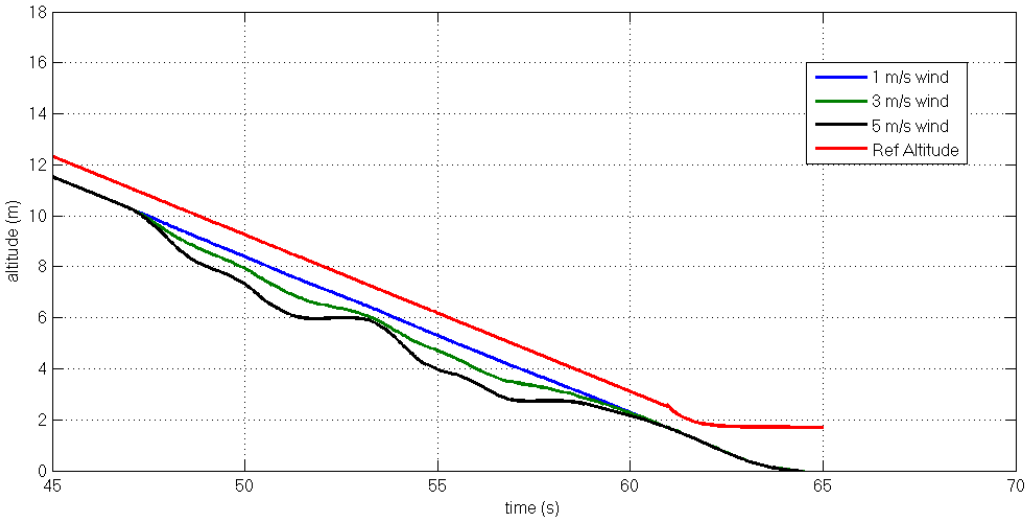

Figure 166 - Altitude response of 1st SMC with CTG

Some dangerous altitude drops can be seen from the figure 166. The aircraft manages to land safely in this case but a wind with more duration would crash the aircraft. SMC with CTG has similar lateral response with LQT.

2nd SMC with CTG

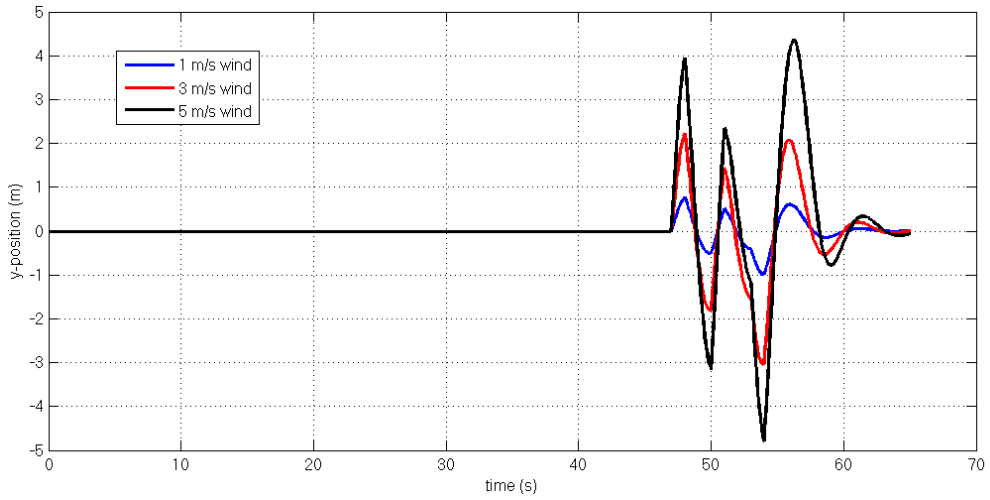

Figure 167 - Y-axis Position Plot for 2nd SMC with CTG

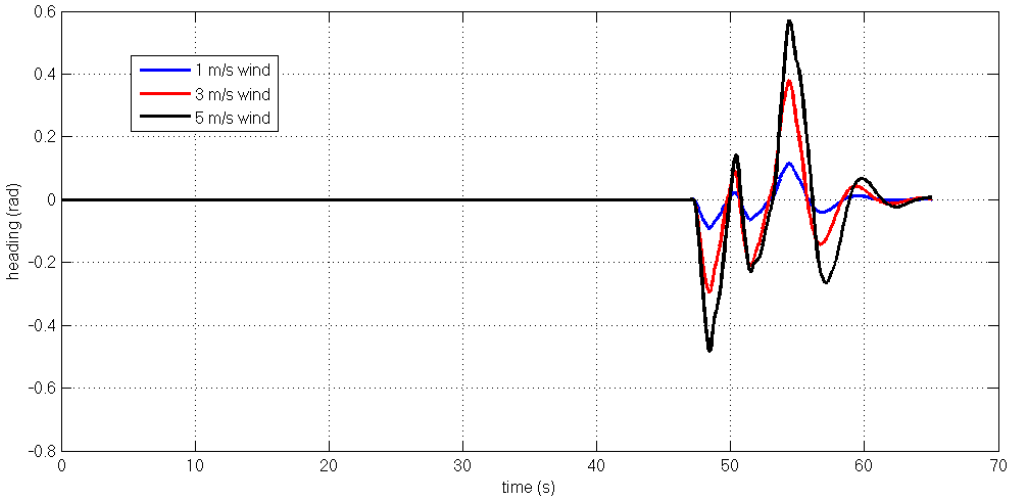

Figure 168 - Heading plot for 2nd SMC with CTG (in radians)

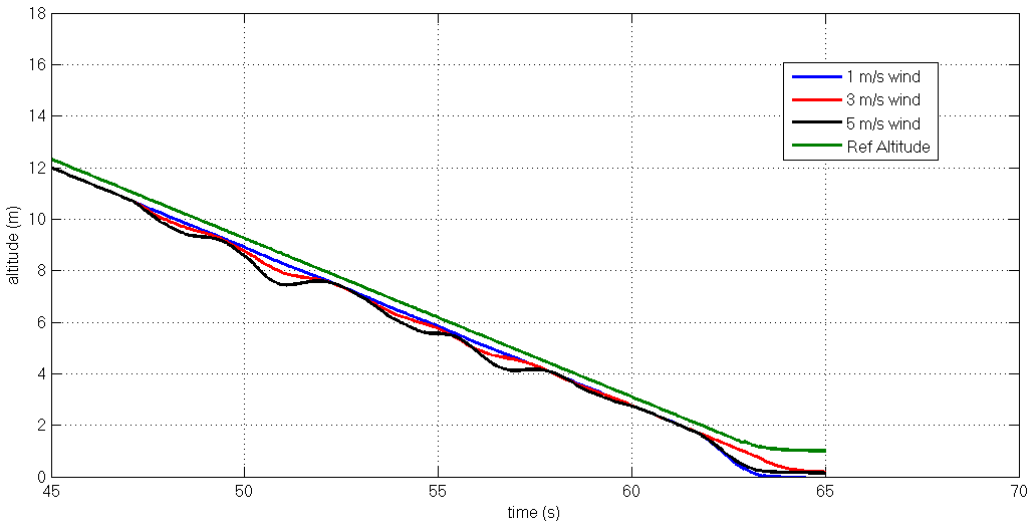

Figure 169 – Altitude response of 2nd SMC with CTG

Lateral responses of 2nd SMC is very similar to 1st SMC. As expected the altitude drop is smaller. This performance is appropriate for landing. Still there is small offset observed, despite the disturbance effect is gone. It was expected that to get in to desired altitude when the disturbance fades.

NMPC

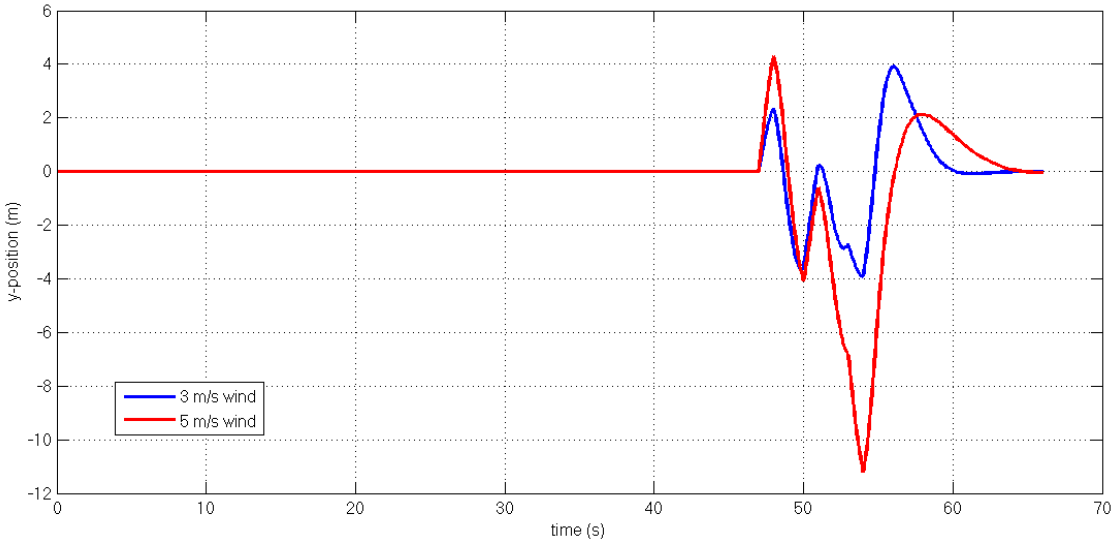

Figure 170 – Y-position plot of NMPC

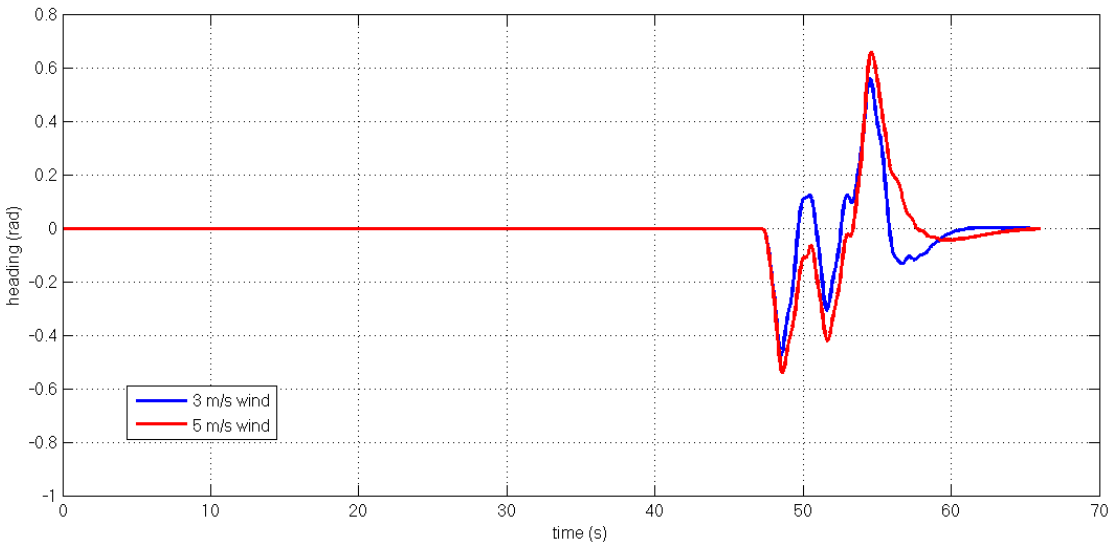

Figure 171 – Heading plot for NMPC (in radians)

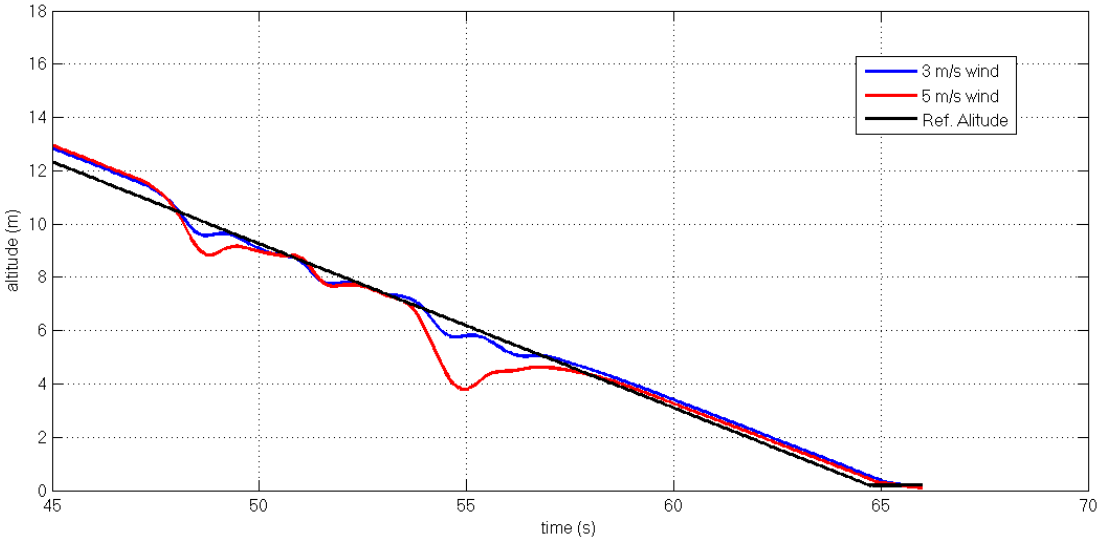

Figure 172 – Altitude response of NMPC

Model Predictive Controller has slower but less oscillatory lateral response compared to other controllers. Still it manages to keep the desired track. Altitude drops seen under 5 m/s winds which may be dangerous.

5.9 Scenario 9

Note that the disturbances were ignored in all scenarios done by now. In this scenario the performances of linear quadratic tracker, sliding mode controller and model predictive controller are tested under some Gaussian white noise. The aircraft approaches the runway and a steady tail wind (5 m/s) is present. While close to touch-down point three gust winds hit the aircraft.

LQT

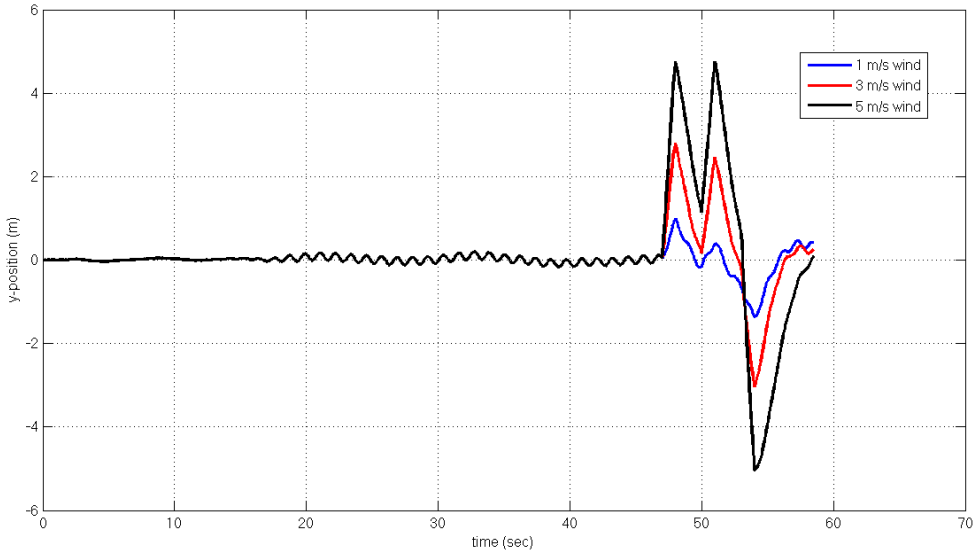

Figure 173 – Y-position plot of LQT with CTG

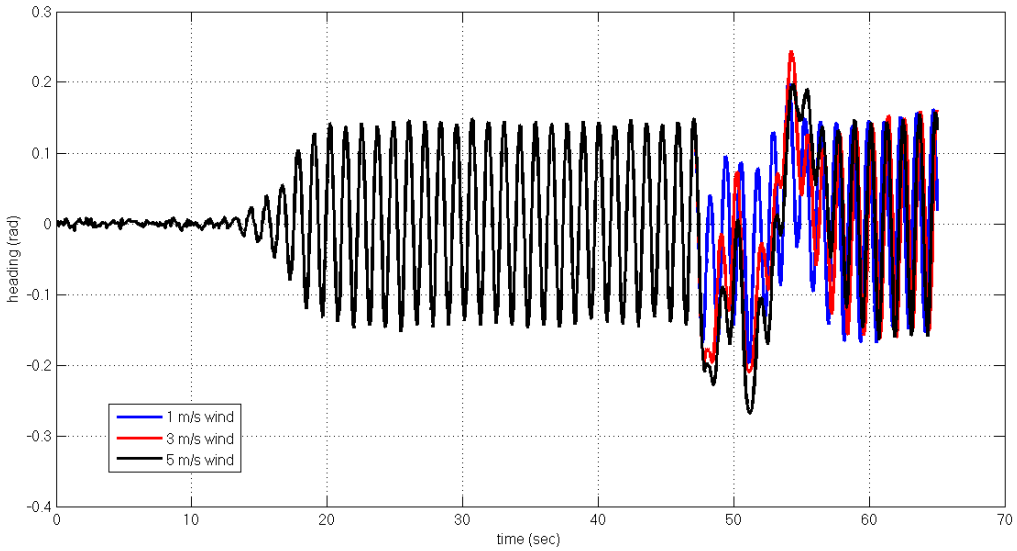

Figure 174 - Heading Plot for LQT with CTG (in radians)

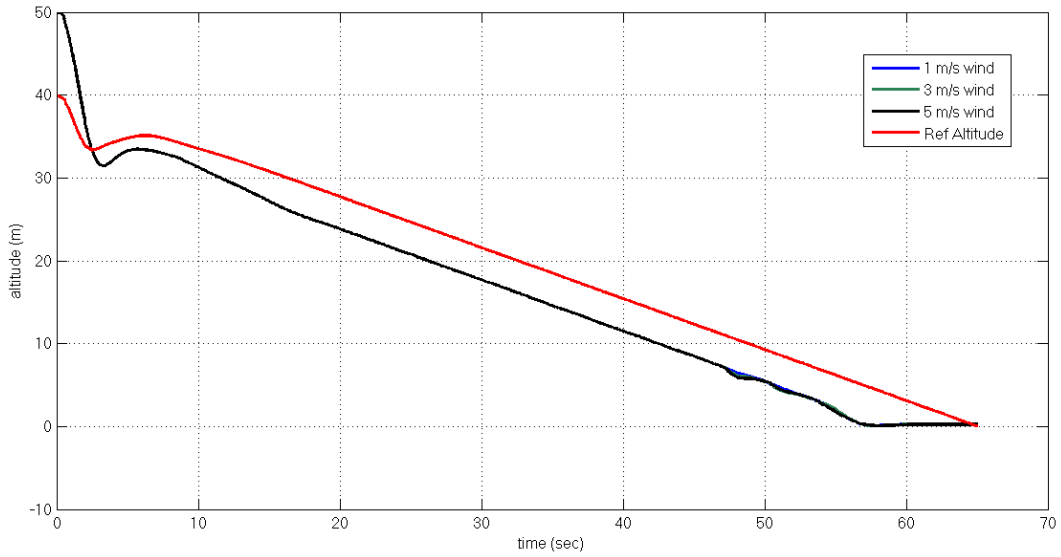

Figure 175 - Altitude response of LQT with CTG

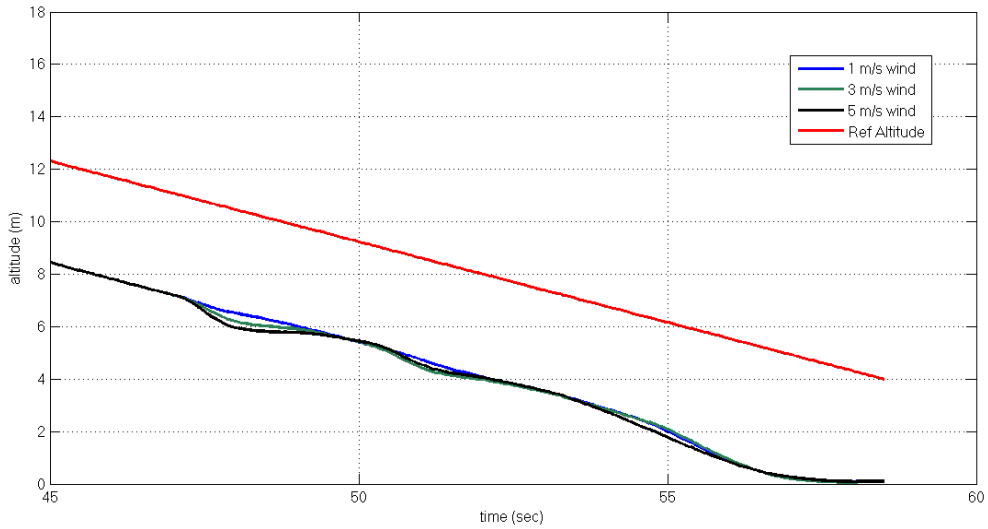

Figure 176 - Altitude response of LQT with CTG (zoomed)

As expected an offset between the desired landing trajectory and the flight path is observed. This is may be a serious problem since the flare mechanism in this thesis is design to increase the pitch angle when the altitude reaches some point. So the plane will land sooner than expected and miss the runway. Although the lateral controller manages to fix the y-position error in time, the heading of aircraft starts to oscillate under tail wind and disturbance effects which may cause some damage to the plane.

1st SMC

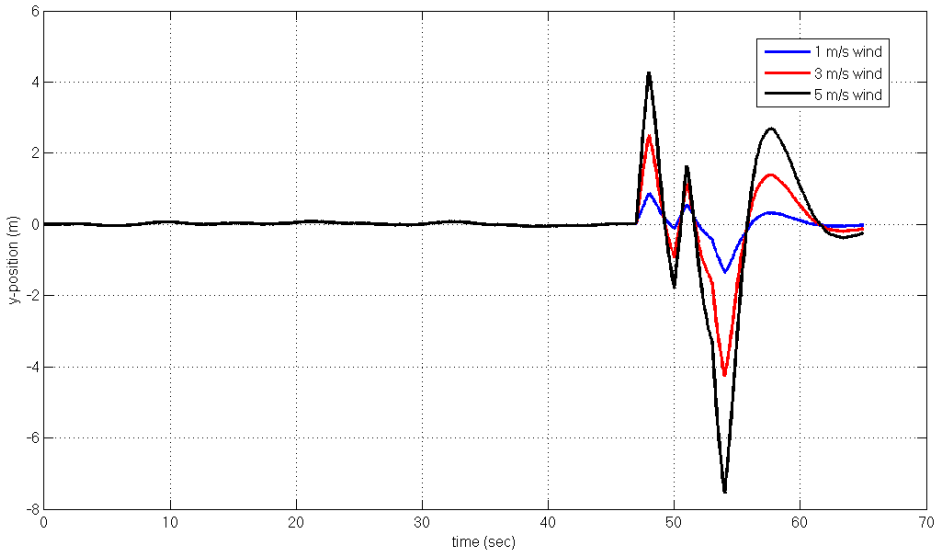

Figure 177 - Y-position plot of SMC with CTG

Sliding mode controller keeps the aircraft in desired y-position better than linear quadratic tracker, since it has no oscillatory response and the y-position error at touch-down is smaller. Still, y-position error is present under 5m/s cross-winds.

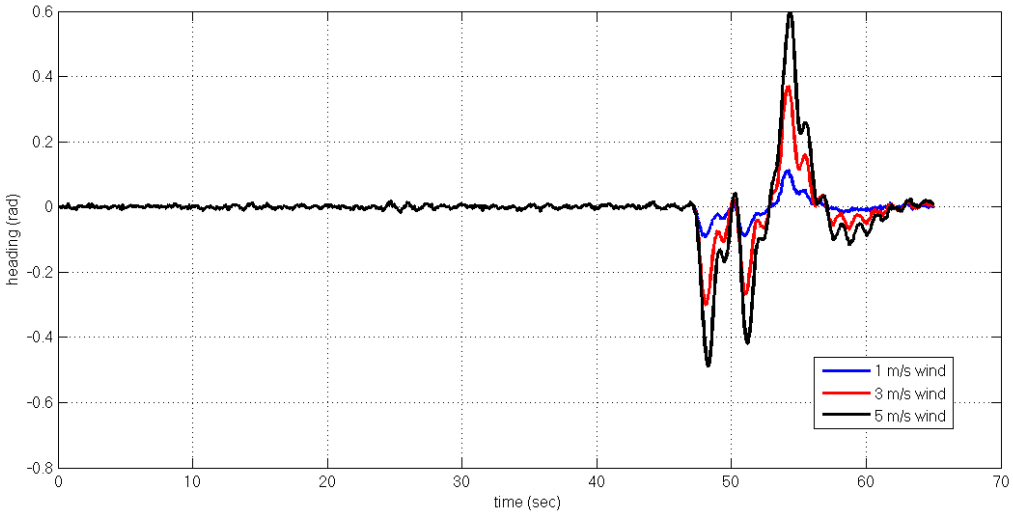

Figure 178 - Heading Plot for SMC with CTG (in radians)

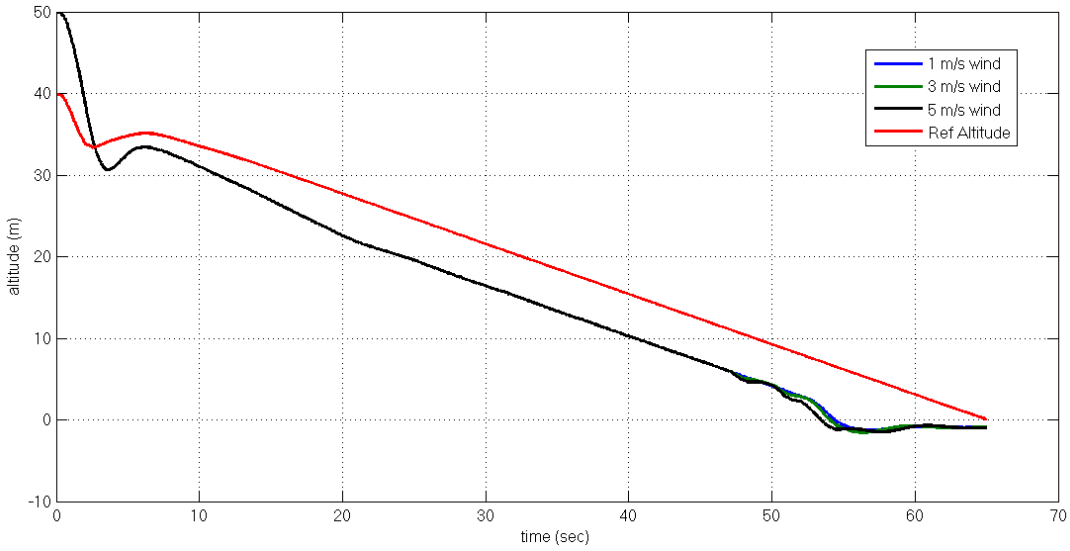

Figure 179 - Altitude response of LQT with CTG

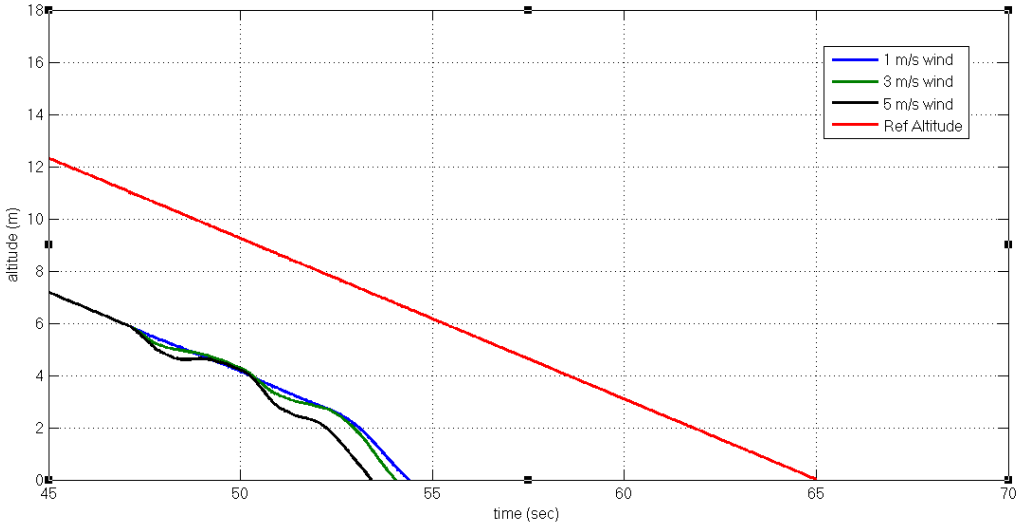

Figure 180 - Altitude response of SMC with CTG (zoomed)

The heading response illustrated in figure 178 is better damped than LQT case but still oscillations are present. Like linear quadratic tracker, altitude trajectory of sliding mode controller has some offset which result in a crash to the ground under crosswinds coming near touch-down point. The altitude drop is critical while performing lateral movements. Sliding mode controller fails this scenario.

NMPC

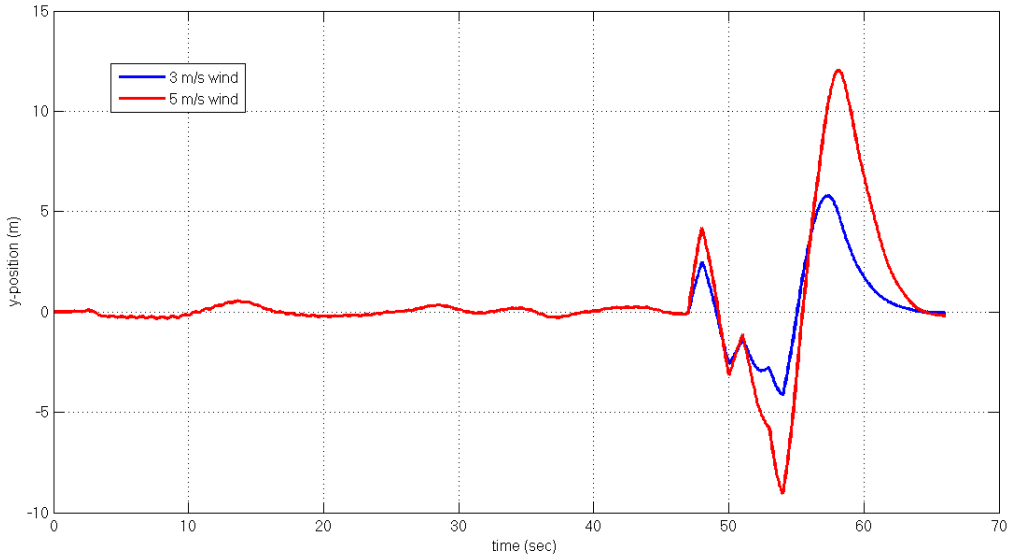

Figure 181 - Y-position plot of NMPC

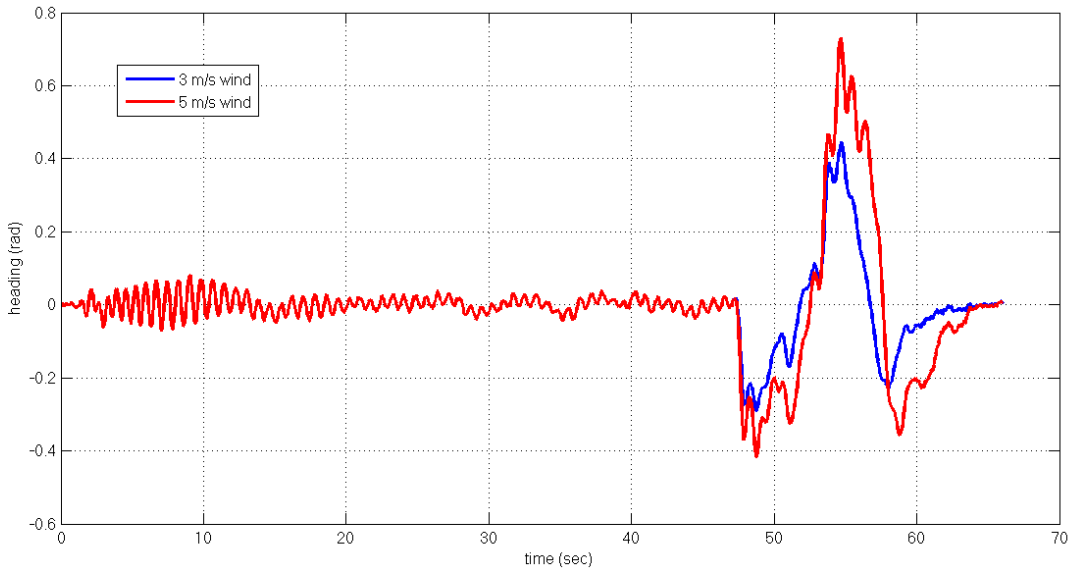

Figure 182 - Heading Plot for NMPC (in radians)

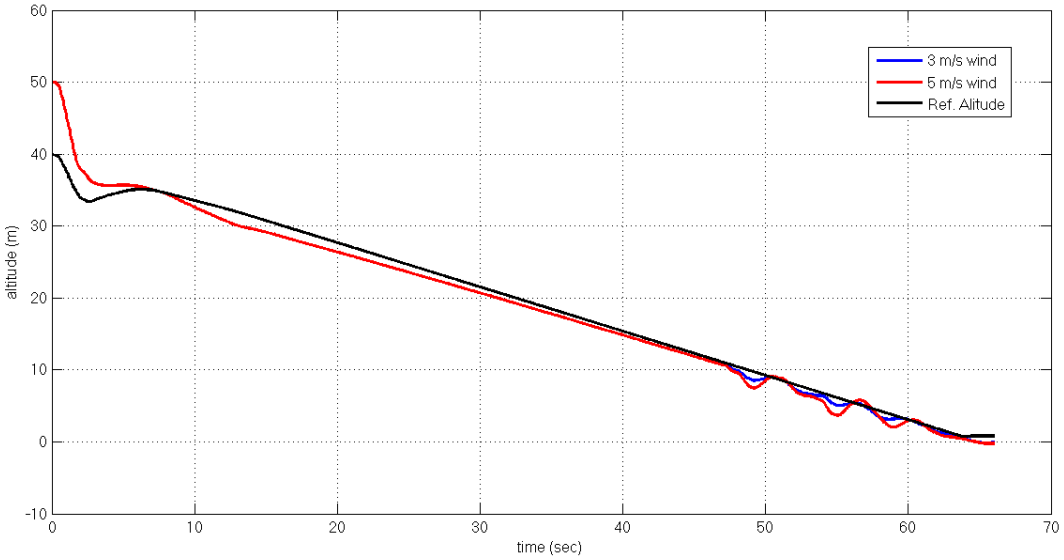

Figure 183 - Altitude response of NMPC

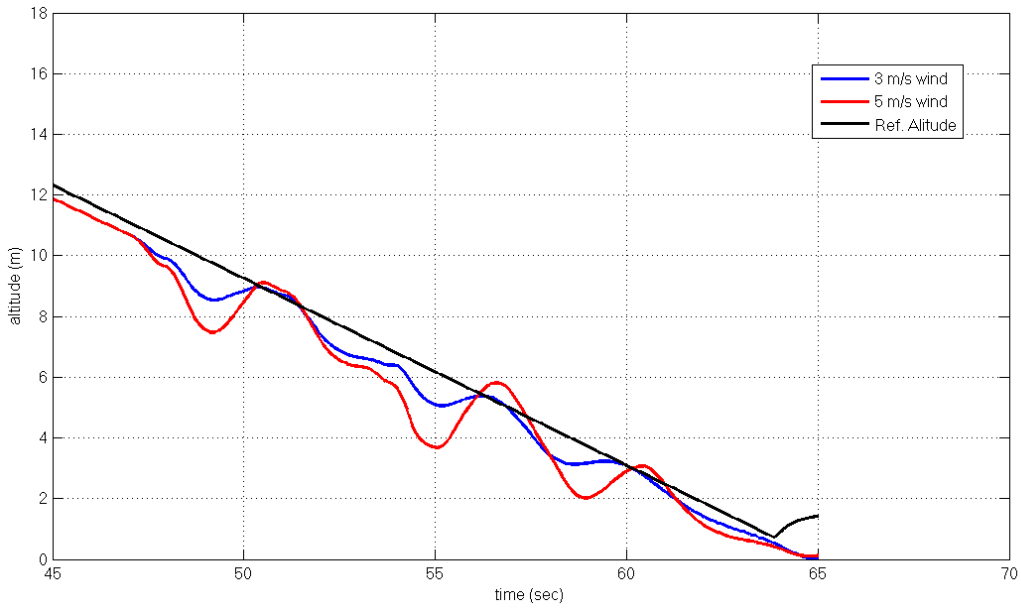

Figure 184 - Altitude response of NMPC

By looking figure 181, it can be easily said that model predictive controller has the most reliable y-position error correction compared to other controllers. The heading is near zero while landing. Although altitude drops are bigger than linear quadratic tracker case, this controller tracks the desired landing corridor better than other controllers under tail winds which represents the most dangerous landing situation.

CONCLUSION

The mathematical model of Telemaster UAV was implemented on MATLAB/Simulink. The dynamic and static stability derivatives were computed using DATCOM software. Several control methods were tested with predefined scenarios which consists of wind effects during landing.

First, PID controllers was designed to control aircraft. It was observed that PID controllers aren't sufficient enough to control correlated states like, speed and altitude. Than LQT (Linear Quadratic Tracker) and SMC (Sliding Mode Controller) are designed with the linearized form of nonlinear model. Two types of sliding mode controllers are tested. One of them uses only altitude error to give elevator commands while second one uses both speed and altitude errors to give elevator commands. The second type of SMC can adjust the balance of speed of convergence of aircraft to the desired reference altitude between keeping the speed in acceptable region. Model Predictive Control has also tested. Which is an optimization based control method.

A guidance block has designed that allows the aircraft to follow waypoints and two types of guidance methods are implemented. The purpose of these methods are to get the aircraft in desired trajectory and reject outer disturbances. With Lateral Track Guidance, the smoothness of movement of the aircraft can be adjusted and the Cross Track Guidance is used mainly for calculating the wind effects and try to cope with them.

Since the landing phase is the most critical phase of flight, lateral and longitudinal motions of aircraft are controlled in order to track some track. Optimal paths for several altitudes and distances to the runway are obtained using steepest decent algorithm which uses dichotomous method as one dimensional search. After that, the obtained optimal paths are interpolated in order to use them any initial state. Heading,

speed and altitude states should be in specific interval for a safe landing. The performances of designed controllers are tested in adverse weather conditions while landing phase of flight. The results showed that, LQT and SMC are superior to PID controllers with similar performances. In overall MPC has a very good tracking performance for all desired states (y-position, speed and altitude).

It is observed from the simulations that modern control techniques (LQT, SMC) are better than PID controllers. The comparison between LQT, SMC and NMPC is harder, since none of them is superior in all cases. Formulation of the standard LQ problem results in proportional state feedback control law. The controller doesn't have an integrator. This implies that the steady state of LQR is error free. So in continuous disturbances LQT generally results in a steady state error but in instant disturbances LQT manages to keep desired trajectory better than all other controllers. Sliding mode controller has very similar responses like LQT. It is observed that using additional sliding surfaces can improve the performance of the SMC controller. Offset issues are still present in this controller. Still it is superior to PID controllers. Model Predictive Controller doesn't have the faster responses but the robust ones. The tracking performance of NMPC is very good even under continuous disturbance effects are present unlike LQT. As a conclusion, all controllers test have advantages and disadvantages, changing with the scenario. It can be said that LQT, SMC and NMPC are better that PID controllers but making a decision between these three controllers is hard.

REFERENCES

[1] K.T. Guthrie, "Linear Parameter-Varying Path Following Control of a Small Fixed Wing Unmanned Aerial Vehicle", M.Sc. Thesis, Virginia Polytechnic Institute and State University, 2013.

[2] R. Frezza, "Path Following for Air Vehicles in Coordinated Flight", International Conference on Advanced Intelligent Mechatronics, pp. 884-889, 1999.

[3] S. Saripalli, "Visually Guided Landing of an Unmanned Aerial Vehicle", IEEE Transactions on Robotics and Automation, vol. 19, no. 3, pp. 371-380, 2003.

[4] D.R. Nelson, D.B. Barber, T.W. McLain, "Vector Field Path Following for Miniature Air Vehicles", IEEE Transactions on Robotics, vol. 23, no. 3, pp. 519-529, 2007.

[5] S. Suresh, N. Kannan, "Direct Adaptive Neural Flight Control System for an Unstable Unmanned Aircraft", Applied Soft Computing, vol. 8, pp. 937-948, 2008.

[6] H.Y. Chao, Y.C. Cao, Y.Q. Chen, "Autopilots for Small Unmanned Aerial Vehicles: A Survey", International Journal of Control, Automation, and Systems, vol. 8, no. 1, pp. 36-44, 2010.

[7] D. Royer, "Design of an Automatic Landing System for the Meridian UAV using Fuzzy Logic", M.Sc. Thesis, University of Kansas, 2010.

[8] S. Adiansyah, "Mazari UAV Adaptive Autopilot System Design and Implementation", M.Sc. Thesis, American University of Sharjah, 2013.

[9] S. Sunil, "Development and Testing of a Simulation Environment for an Unmanned Aerial Vehicle", M.Sc. Thesis, Carleton University, 2013.

[10] J. Osborne, R. Rysdyk, "Waypoint Guidance for Small UAVs in Wind", in Infotech@Aerospace, Sep. 2005.

[11] J.D.B. Gonzalez, H.R. Cortes, "Adaptive energy based control for the longitudinal dynamics of a fixed-wing aircraft", In 2014 American Control Conference, 2014.

[12] V. Kargın, “Design of an Autonomous Landing Control Algorithm for a Fixed Wing UAV”, M.Sc. Thesis, METU, 2007.

[13] B. Galbraith, “Aircraft Coefficient Comparisons Between Datcom and Published Data”, Holy Cows, Inc., 2011. American Control Conference (ACC), 2014.

[14] E.L. Duke, R.F. Antoniewicz, K.D. Krambeer, “Derivation and Definition of a Linear Aircraft Model”, NASA Reference publication, 1988.

[15] M. Ulu, “Sliding Mode Guidance of an Air-to-Air Missile”, M.Sc. Thesis, METU, 2013.

[16] N. Anton, R.M. Botez, D. Popescu, “Stability derivatives for a delta-wing X-31 aircraft validated using wind tunnel test data”, Proceedings of the Institution of Mechanical Engineers Part G Journal of Aerospace Engineering, vol. 225, pp. 403-416, 2010.

[17] M. Rauw, “A Simulink Toolbox for Flight Dynamics and Control Analysis”, 2001.

[18] J.R. Leigh, “Control Theory 2nd Edition”, The Instution of Electrical Engineers, 2004.

[19] M. Hanköylü, “Landing Autopilot Design for an UAV”, M.Sc. Thesis, METU, 2011.

[20] K. Ogata, “Modern Control Engineering”, Prentice Hall, 2010.

[21] R. SaravanaKumar, K. Vinoth Kumar, K.K. Ray, “Sliding Mode Control of Induction Motor using Simulation Approach”, UCSNS International Journal of Computer Science and Network Security, vol. 9, no. 10, 2009.

[22] A. Visioli, “Practical PID Control”, Springer, 2006.

[23] K.J. Astrom, T. Hagglund, “PID Controllers, 2nd Edition”, Instrument Society of America, 1995.

[24] R.S. Christiansen, “Design of an Autopilot for Small Unmanned Aerial Vehicles”, A Thesis Submitted to the Faculty of Brigham Young University, August 2004.

[25] D. McLean, “Automatic Flight Control Systems”, Prentice Hall, 1990.

[26] J.P. Hespana, “Linear Systems Theory”, Princeton University Press, 2009.

[27] E. Barbieri, R. Alba-Flores, “Real-time Infinite Horizon Linear-Quadratic Tracking Controller for Vibration Quenching in Flexible Beams”, IEEE Conference on Systems, Man, and Cybernetics, vol. 1, pp. 38-43, 2006.

[28] Utkin, V.I., “Variable structure systems with sliding modes”, IEEE Trans. Automat. Contr., Vol. AC-22, no. 2, pp. 212-222, 1977.

[29] W. S. Levine, “Control System Advanced Methods”, CRC Press Taylor & Francis Group, 2011.

[30] B. Bandyopadhyay, F. Deepak, and K.S. Kim, “Sliding Mode Control Using Novel Sliding Surfaces”, SpringerVerlag, Berlin Heidelberg, 2009.

[31] I.H. Johansen, “Autopilot Design for Unmanned Aerial Vehicles”, M.Sc. Thesis, Norwegian University of Science and Technology, 2016.

[32] S. Arıbal, “Development of an autopilot for automatic landing of an unmanned aerial vehicle”, M.Sc. Thesis, METU, 2011.

[33] C. Edwards, S.K. Spurgeon, “Sliding Mode Control Theory and Applications”, Taylor and Francis Ltd, 1998.

[34] D.G. Luenberger, “Linear and Nonlinear Programming, Second Edition”, Addison Wesley, 1984.

[35] L. Grüne, J. Pannek, “Nonlinear Model Predictive Control Theory and Algorithms”, Springer, 2011.

[36] M.A. Abbas, “Non-linear Model Predictive Control for Autonomous Vehicles”, M.Sc. Thesis, University of Ontario Institute of Technology, 2011.

[37] http://www.delphiforfun.org/programs/math_topics/WindTriangle.htm , April 12th, 2011.

[38] M. Niculescu, “Lateral Track Control Law for Aerosonde UAV”, AIAA 2001-0016, 2001.

APPENDICES

A.DATCOM MODEL

There are six main parts to define an aircraft in DATCOM. These parts are addressed as namelists in the syntax of digital DATCOM uses. Namelists are defined as follows:

FLTCON - defines the flight conditions

SYNTHS - locates the cg, wing, horizontal tail, and vertical tail with respect to a reference line

BODY - defines the body geometry

WGPLNF - defines the wing planform geometry

HTPLNF - defines the horizontal tail geometry

VTPLNF - defines the vertical tail geometry

A.1 The fltcon namelist

Under FLTCON namelist, the flight conditions are defined such as Mach number(s), altitude(s), and angle of attacks.

NMACH - number of Mach numbers to be run. We will only run at one Mach.

MACH - the predeterminedMach numbers to be run. We will run at Mach = 0.05.

NALPHA - the number of angles of attack to test. We will test at 18 AOA's.

ALSCHD - the schedule of angles of attack. We will use (in deg): -10,-8,-6,-4, -2, 0, 2, 4, 6, 8, 10, 12, 14,16,17,18 (eight teen total)

NALT - number of altitudes to run. We will test only one altitude.

ALT - the altitudes to run. We will test at 100 meters.

WT - the weight of the aircraft. We will use 41N.

A.2 The synths namelist

Aircraft's center of gravity location locations as well as the positions of wing and tail surfaces are defined in this section. The horizontal measurements are taken from the nose of the aircraft. The vertical measurements are taken from a reference line which is placed at the center of the aircraft.

XCG - the horizontal position of the c.g. It's defined as 0.43 meters.

ZCG - the vertical position of the c.g. with respect to the reference line. Arbitrarily chosen as 0.003 meters.

XW - the horizontal position of the apex of the wing. Entered as 0.325 meters.

ZW - the vertical position of the wing apex wrt the reference line. Entered as 0.088 meters.

ALIW - the incidence of the wing in degrees. Taken as 0 degree.

XH - the horizontal position of the apex of the horizontal tail. Entered as 1.474 meters.

ZH - the vertical position of the horiz. tail apex wrt the reference line. Entered as 0.074 meters.

ALIH - incidence of the horizontal tail. Taken as 0 degrees.

XV - the horizontal position of the apex of the vertical tail. Entered as 1.474 meters.

ZV - the vertical position of the vertical tail apex wrt the reference line. It's 0.074 meters.

The following figure shows the graphical representation of definitions which are used in some of the namelists.

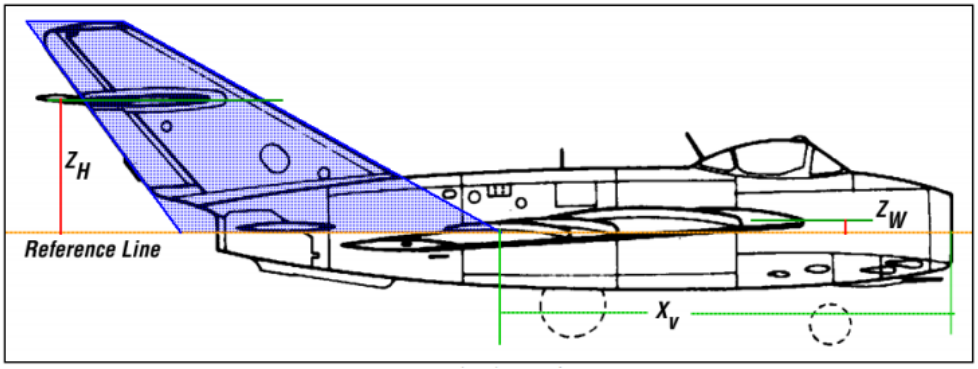

Figure 185 - Side view of aircraft with parameters that DATCOM uses.

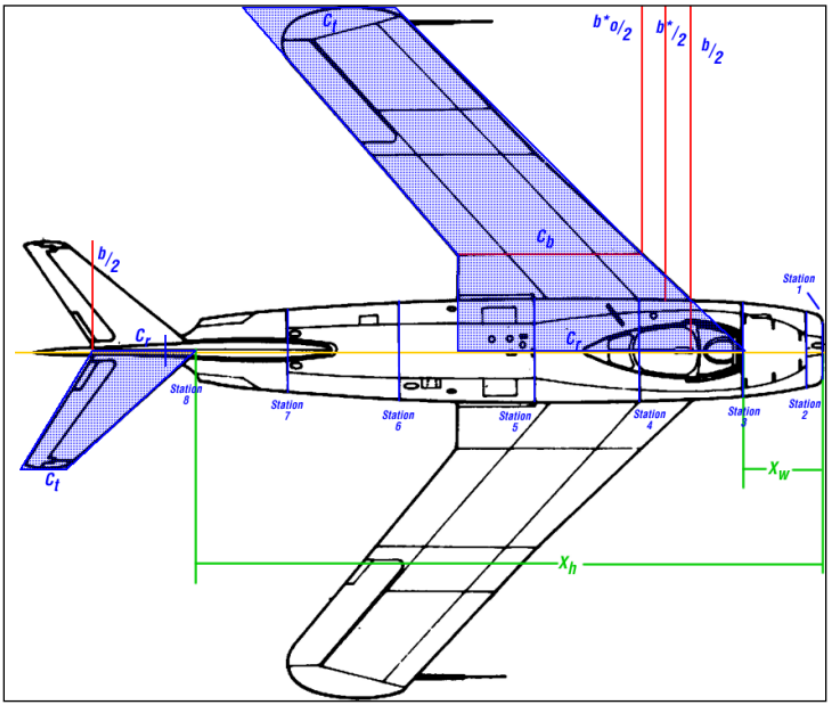

Figure 186 – Top view of aircraft with parameters that DATCOM uses.

A.3 The body namelist

Note that from the drawing above, the body has been divided into 8 stations. It is required to have horizontal distances of every station from the nose of the aircraft as well as the cross sectional area of each station. The variables we defined under this namelist is given below:

NX - number of body stations.

X - horizontal distance of each station.

S - cross sectional area at each station.

Table 5 – Horizontal distances of sections and their areas of Telemaster UAV.

Station	1	2	3	4	5	6	7	8
X (meter)	0.0	0.0391	0.854	0.2070	0.2649	3864	0.5975	1.3589
S(area,meter^2)	0.0085	0.01444	0.0174	0.019	0.031	0.031	0.028	0.003

A.4 The wgplnf namelist

In this section wing planform is defined by the parameters described in figure 2. The following variables are used to implement wing planform in DATCOM:

CHRDTP - the length of the chord at the tip of the wing. Measured to be 0.30 meters.

SSPNE - the "exposed" semi-span, which is the "b*/2". This dimension is from the side of the fuselage to the tip chord. Entered as 0.85 meters.

SSPN - the theoretical semi-span, which is b/2. This dimension is from the root chord to the tip chord. Entered as 0.90 meters.

CHRDR - the length of the chord at the root of the wing. Measured to be 0.30 meters.

SAVSI - the sweep of the wing at the inboard panel (refer to the manual). 0 degrees in our case.

SAVSO - the sweep of the wing at the outboard panel. A stated before, it will be 0 deg.

TWISTA - the twist angle of the wing (wash-out). For the Telemaster, it is 0 deg.

CHSTAT - the % of the mac at which the sweep angle will be referenced. it is 0 .

DHDAHI - the dihedral of the inboard panel. If the inboard and outboard panel dihedral is the same (constant dihedral across the wing), then only DHDADI is inputted. For the Telemaster, the wing dihedral is 3 deg.

TYPE - different planform types available. Will be set to 1. In our case straight tapered planform.

A.5 The htplnf namelist

The horizontal tail is described in this section which uses same variables desciribed in WGPLNF namelist.

CHRDTP is 0.22 meters.

SSPNE is 0.28 meters.

SSPN is 0.30 meters.

CHRDR is 0.22 meters.

SAVSI is 0 degrees.

CHSTAT is 0.25.

A.6 The vtplnf namelist

The vertical tail is described in this section which uses same variables desciribed in WGPLNF namelist.

CHRDTP is 0.16 meters.

SSPNE is 0.25 meters.

SSPN is 0.25 meters.

CHRDR is 0.22 meters.

SAVSI is 20 degrees.

CHSTAT is 0.25.