DEVELOPMENT OF AN INTELLIGENT MODEL PREDICTION CONTROLLER FOR AUTONOMOUS HELICOPTERS

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

BY

ŞEVKET ESER KUBALI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
AEROSPACE ENGINEERING

MAY 2016

Approval of the thesis:

DEVELOPMENT OF AN INTELLIGENT MODEL PREDICTION CONTROLLER FOR AUTONOMOUS HELICOPTERS

submitted by **ŞEVKET ESER KUBALI** in partial fulfillment of the requirements for the degree of **Master of Science in Aerospace Engineering Department, Middle East Technical University** by,

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

Prof. Dr. Ozan Tekinalp
Head of Department, **Aerospace Engineering** ______________

Assoc. Prof. Dr. İlkay Yavrucuk
Supervisor, **Aerospace Engineering Department, METU** ______________

Examining Committee Members:

Prof. Dr. Ozan Tekinalp
Aerospace Engineering Department, METU ______________

Assoc. Prof. Dr. İlkay Yavrucuk
Aerospace Engineering Department, METU ______________

Prof. Dr. M. Kemal Leblebicioğlu
Electrical and Electronics Engineering Department, METU ______________

Prof. Dr. Metin U. SALAMCI
Mechanical Engineering Department, Gazi University ______________

Assist. Prof. Dr. Ali Türker Kutay
Aerospace Engineering Department, METU ______________

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: ŞEVKET ESER KUBALI

Signature :

ABSTRACT

DEVELOPMENT OF AN INTELLIGENT MODEL PREDICTION
CONTROLLER FOR AUTONOMOUS HELICOPTERS

Kubali, Şevket Eser
M.S., Department of Aerospace Engineering
Supervisor : Assoc. Prof. Dr. İlkay Yavrucuk

May 2016, 81 pages

In this thesis, a new PID gain update law using linear least squares regression is introduced as an adaptive control method for autonomous helicopters. In addition, future prediction analyses are conducted for error dynamics of the closed loop system using recursive linear least squares regression. Combining these two concepts with classical PID controller, an intelligent PID controller is obtained. On the other hand, using PID controllers, a flight controller with three control loops is developed to demonstrate the capabilities of the new intelligent controller and PID controllers of second and third control loops of this flight controller are replaced by the newly developed intelligent controller. Thus, a new intelligent flight controller is acquired with model prediction and adaptation abilities. Several challenging maneuvers are carried out in virtual environment for the flight controller that has no adaptation ability and the new intelligent flight controller using the same initially stable PID gains to investigate the success of the new intelligent controller.

Keywords: adaptive control, pid controller, least squares regression, optimization, helicopter, simulation

ÖZ

OTONOM HELİKOPTERLER İÇİN AKILLI BİR MODEL TAHMİN KONTROLCÜSÜ GELİŞTİRİLMESİ

Kubali, Şevket Eser
Yüksek Lisans, Havacılık ve Uzay Mühendisliği Bölümü
Tez Yöneticisi : Doç. Dr. İlkay Yavrucuk

Mayıs 2016 , 81 sayfa

Bu tezde, yeni bir PID kontrolcüsü kazanç ayarı güncelleme kuralı, doğrusal en küçük kareler ilişkilendirmesi kullanılarak yeni bir uyarlanabilir kontrol yöntemi olarak sunulmuştur. Ayrıca, kapalı devre sistemlerin hata dinamiği için yinelemeli doğrusal en küçük kareler ilişkilendirmesi kullanarak gelecek tahmin analizleri yürütülmüştür. Bu iki kavram, klasik PID kontrolcüsüyle birleştirilerek, yeni bir akıllı PID kontrolcüsü elde edilmiştir. Bunun yanısıra, yeni aklıllı kontrolcünün yeteneklerini göstermek için PID kontrolcüler kullanarak üç kontrol döngülü bir uçuş kontrolcüsü geliştirilmiş ve bu uçuş kontrolcüsünün ikinici ve üçüncü kontrol döngülerindeki PID kontrolcüleri yeni geliştirilen kontrolcüyle değiştirilmiştir. Böylece, model tahmin ve uyarlanma yetenekleri olan yeni bir akıllı uçuş kontrolcüsü elde edilmiştir. Yeni akıllı kontrolcünün başarısını inceleyebilmek için uyarlanma yeteneği olmayan ilk uçuş kontrolcüsüyle yeni akıllı uçuş kontrolcüsü, sanal ortamda çeşitli ve zorlayıcı manevralara tabi tutulmuşlardır.

Anahtar Kelimeler: uyarlanabilir kontrol, pid kontrolcüsü, en iyileştirme, en küçük kareler ilişkilendirmesi, helikopter, simülasyon

To my family

ACKNOWLEDGMENTS

I would like to thank my supervisor Associate Professor İlkay Yavrucuk for his guidance and friendship. It was a great honor and opportunity to work with him for nearly eleven years as well as a student and a professional engineer. My aerospace perspective is shaped and improved a lot by the virtue of him. I would also like to thank him for nine lectures during my aerospace education.

I would like to thank to Yunus Emre Arslantaş and Gönenç Gürsoy for their endless support, guidance and friendship, without them it would be very hard to complete this thesis. I would like to express my thanks to Onur Tarımcı for his friendship and very well prepared master thesis which is a very good handbook for this study. I would also like to thank to Deniz Yılmaz for his friendship and guidance in Simulation, Control and Avionics Laboratory.

I would like to thank to my directors at TAI, Akif Çetintaş and Özcan Ertem for giving me a work leave for a duration of two months to complete my thesis.

I am very grateful to all people who guide, trust and support me during my whole education. I appreciate the support of my cousin Serdar Somyürek who take over the work load of my website bitzfree.com during thesis period. I would also thank to Ali Şahin, Alper Gümüştepe and Esra Arslan for their close friendship and support. I could not even graduate from METU-AEE without them.

A special gratitude and love goes to my family for their unique unfailing support and trust. I could not think the completion of this thesis without their endless help.

TABLE OF CONTENTS

CHAPTERS

APPENDICES

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

Alt	Altitude
D_n	n^{th} coefficient of error modeling for derivative channel
deg	degree
E	Least Squares Sum of modeling errors of linearization for Least Squares Regression
e	Error
F_x	Resultant Force in body x-axis
F_y	Resultant Force in body y-axis
ft	Feet
g	Gravitational Acceleration
Hz	Hertz
L_{VB}	Transformation Matrix from Body Frame to North-East-Down Navigation Frame
L_{BV}	Transformation Matrix from North-East-Down Navigation Frame to Body Frame
m	Mass of the helicopter
I_n	n^{th} coefficient of error modeling for integral channel
K	Gain of the PID controller
N	Number of data points for Least Squares Regression
P	Number of past values for Least Squares Regression
P_n	n^{th} coefficient of error modeling for proportional channel
R	Number of recursive usages of Least Squares Regression for future prediction
p	Body roll rate (Angular velocity over body x-axis)
q	Body pitch rate (Angular velocity over body y-axis)
r	Body yaw rate (Angular velocity over body z-axis)
S	Success ratio
t	Time
u	Command input to controller
U_ϕ	Command input for desired Euler roll angle
U_θ	Command input for desired Euler pitch angle
U_ψ	Command input for desired Euler yaw angle
U_{X_N}	Command input for desired north position in NED navigation frame
U_{X_E}	Command input for desired east position in NED navigation frame
U_{X_D}	Command input for desired down position in NED navigation frame
u	Velocity in body x-axis
v	Velocity in body y-axis
w	Velocity in body z-axis
X_N	North Position in NED navigation frame
X_E	East Position in NED navigation frame

X_D	Down Position in NED navigation frame
X	Position in body x-axis
Y	Position in body y-axis
Z	Position in body z-axis
$\frac{1}{s+1}$	Transfer function in Laplace domain
α	Coefficient matrix in Least Squares Regression
Δ	Modeling error of linearization for Least Squares Regression
δ_e	Longitudinal cyclic control input
δ_c	Collective control input
δ_a	Lateral cyclic control input
δ_p	Pedal control input
η	Learning rate
Φ	State matrix in Least Squares Regression
ϕ	Euler roll angle
θ	Euler pitch angle
ψ	Euler yaw angle
ω_n	Natural frequency
ξ	Damping ratio

Subscripts and Superscripts

0_b	Body
0_c	Command
0_d	Desired
0_D	Derivative channel
0_I	Integral channel
0_k	k^{th} value of the state in a data set
0_P	Proportional channel
0_{ref}	Reference test
0_{new}	New test
0_{trim}	Trim value
0_v	Vehicle
0^T	Transpose of the matrix
0^{-1}	Inverse of the matrix
$\hat{0}$	Instant Prediction
$\dot{0}$	First derivative with respect to time
$\ddot{0}$	Second derivative with respect to time
$\Delta 0$	Deviation

Acronyms

3-D	Three dimension
DLS	Damped Least Squares Regression
EAC	Euler Angle Controller
BAVC	Body Angular Velocity Controller
IFC	Intelligent Flight Controller
I-PD	Improved Proportional Derivative (Controller)
I-PID	Improved Proportional Integral Derivative (Controller)
min	Minimum function
MRAC	Model Reference Adaptive Control
NED	North East Down
PI	Proportional Integral (Controller)
PD	Proportional Derivative (Controller)
PID	Proportional Integral Derivative (Controller)
RBF	Radial Basis Function
RLS	Recursive Least Squares Regression
RMS	Root Mean Square
RMSE	Root Mean Square Error
UAV	Unmanned Aerial Vehicle

CHAPTER 1

INTRODUCTION

Unmanned Aerial Vehicles (UAVs) have been a topic of research in academia for many years because of their maneuverability and versatility [1]. As the affordability of these platforms increases, their popularity increased worldwide for both military and civil aviation. Moreover, conversion of existing human piloted[2] and remotely piloted[3] aerial platforms into UAVs have been studied.

Beyond the military applications like reconnaissance, surveillance and border patrol operations[4], UAVs play a significant role for civil aviation such as aerial surveying of crops in farming, aerial footage in filmmaking [5], search and rescue operations for dangerous missions like after the nuclear accident in Fukushima[6], forest fire detection [7], inspecting power lines and pipelines [8], counting wildlife and detection of illegal hunting [9],landfill detection[10] and crowd monitoring[11].

The main objective is to control the aerial vehicle and complete the given missions even better than manned platforms. UAVs have several advantages as compared to human piloted systems. One of the most important of them is that UAVs have no perceiving and reaction time. Therefore, performance of these platforms are only limited by their controller architectures.

The conventional proportional-integral-derivative (PID) is the most used type of controller for any control applications because of their simplicity and significant performance in a wide range of operating conditions. Although its efficiency, finding the optimum gains is the most critical part of PID design. Especially for a complex system such as a helicopter due to its coupled dynamics and unstable characteristics, designing and tuning of PID controllers are very hard in practice. Besides, even a well designed PID controller with fixed parameters can hardly adapt to uncertainties and changing flight conditions[12]. Therefore, operation range of PID controllers are restricted with the initial gain settings. For these reasons, self-tuning and adaptive PID controllers are used in literature to design, tune and improve the control performance of PID controllers[13].

In this study, the Bell UH-1 Huey (UH-1H) helicopter which is a two-bladed military helicopter powered by a single turboshaft engine and also commonly used by Turkish Land Forces Aviation is converted to an unmanned platform in virtual environment by developing a full flight controller for hover and forward flight conditions. Then, a PID gain update law using linear least squares regression is presented as a new adaptive control method for autonomous helicopters. In addition, future prediction analysis are conducted for error dynamics of the closed loop system using recursive linear least squares regression. Combining these two concepts with classical PID controller, an intelligent PID controller is acquired. Excluding the velocity controllers, PID controllers of the flight controller are replaced by the newly developed intelligent controller. Thus, a new intelligent flight controller is obtained with model prediction and adaptation abilities.

After completing the design of the intelligent flight controller, the first flight controller that has no adaptation ability and the new intelligent flight controller are tested with same PID gains by conducting several challenging maneuvers to demonstrate the effectiveness of the new intelligent flight controller.

1.1 Literature Survey

Self tuning, adaptive and intelligent controller concepts are well known and widely used in literature. In 1997, an intelligent helicopter controller using artificial neural network, genetic algorithm and fuzzy logic was developed by S. Zein-Sabatto and Y. Zheng[14]. Lee et al. implemented fuzzy neural network as an adaptation method for PID controllers in 2002 [12]. In 2005, Zhang et al. used radial basis function (RBF) neural networks for PID gain adaptation [13]. Sanchez et al. used fuzzy logic to adjust the PID controller gains for an autonomous mini-helicopter in 2007 [15]. O. Tarimci developed a neural network based adaptive flight controller for AH-1S helicopter using model inversion technique in 2009 [16]. This study is the starting point of this thesis. In 2011, Sadeghzadeh et al. developed a trajectory tracking controller for a quadrotor helicopter using gain-scheduled PID and model reference adaptive control (MRAC)[17]. A self tuning PID controller for a twin rotor system was developed by P. Sahu and S. K. Pradhan in 2014. [18]. In 2015, H. Gao et al. developed a fuzzy adaptive PD controller for a quadrotor UAV[19].

Least squares regression is also popular in literature for adaptation and optimization of PID gains. In 1985, A. Brickwedde used RLS for PID pole assignment to control the speed and postion of an electrical drive with a microprocessor[20]. E. Poulin et al. used damped version of recursive linear least square regression (DLS) to find the optimum gains of the PID controller for a given transfer function in 1996 . In this study, gains of PID were calculated directly from the least squares optimization using the process gain, time constant and time delay in the coefficient matrix of DLS[21]. In 1997, a combined method of least-squares estimation, Newton–Raphson search technique and Ziegler-Nichols formulas for self tuning of PID controllers was proposed by Rad et al. [22]. In 1999, Mitsukura et al. also used Recursive Least Squares (RLS) algorithm to find the process gain, time constant and time delay of a PID controller from the deviation of RLS coefficient matrix[23]. In 2000, Grassi et al. used Least Squares Regression for loop shaping of PID controller[24]. In 2004, J. Chen and Y. Cheng used Partial Least Squares algorithm to find the process gain, time constant and time delay of a PID controller from the deviation of RLS coefficient. Differently, they used error as a state instead of using state of the system in least square regression[25]. Again Recursive Least Squares (RLS) algorithm was used to find the process gain, time constant and time delay of a PID controller from the deviation of RLS coefficient matrix with a different gain update formula by T. Yamamoto and S. L. Shah in 2007 [26]. In 2008, Wanfeng et al. used least squares support vector machines with RBF kernel to model the gradient of the system error with respect to control input and update the PID gains using this gradient[27]. Similarly, Zhao et al. used least squares support vector machines with RBF kernel to design an intelligent PID controller in 2009[28]. Recent research have focused on non-linear least squares regression for adaptive control. In 2014, Wilson et al. applied non-linear least squares regression for trajectory optimization[29].

1.2 Contribution of this Thesis

In this thesis, an intelligent flight controller is developed for a full size helicopter that has the ability to perform challenging maneuvers better than a human pilot. As an original contribution to the literature, a new PID gain update law using Linear Least Squares Regression is proposed. In addition, prediction of future values of the closed loop error is achieved by using Recursive Linear Least Squares Regression. Combining these two concepts, a new intelligent controller for autonomous helicopters is obtained that has an effective PID gain scheduling capability.

1.3 Thesis Structure

The structure of this thesis is as follows, Adaptation and model prediction concepts are given in Chapter 1 as an introduction. In addition, results of literature survey about adaptive PID control methods and least squares regression applications for adaptive control are mentioned in Chapter 1.

In Chapter 2, Least Squares Regression and Recursive Least Squares Regression methods are explained in detail. The limitations of Least Squares Regression are also described in this chapter. In addition, implementation of Recursive Least Squares Regression for prediction is mentioned in this chapter.

The main contribution of this thesis which is a least squares based adaptive controller with a new PID gain update law is presented in Chapter 3. Besides, an intelligent controller design that is the second contribution of this thesis is expressed in this chapter.

Architecture of the Intelligent Flight Controller is described in Chapter 4. Implementation of the intelligent controller is also explained in this chapter.

Modeling the UH-1H is described in Chapter 5. Mass, inertia and geometric data of

UH-1H are given in this chapter.

Chapter 6 includes the simulation results of six challenging maneuvers for both a flight controller with classical PID controllers and the Intelligent Flight Controller.

Finally, a brief summary of the thesis is given in Chapter 7. Conclusions and future work are also discussed in this chapter.

CHAPTER 2

METHODOLOGY

In this study, an adaptive and model predictive PID controller is developed by using Least Squares Regression which is widely used for online adaptive control and real time parameter estimation [30]. Least Squares Regression finds an optimum solution by minimizing the sum of the squares of the errors for overdetermined systems [31]. This method is used to derive a new PID gain update law that is detailed in Chapter 3 and also an optimum statistical model to predict the future value of a state from its previous values for an asymptotically stable closed loop system. For a given time frame, Least Squares Regression is used recursively to predict the next future values. This recursive usage is called as Recursive Least Squares Regression in literature [32].

2.1 Linear Least Squares Regression

A state can be modeled using the past values of itself within a given time frame **P**[33].

$$x_k = a_0 + a_1 \cdot x_{k-1} + a_2 \cdot x_{k-2} + \cdots + a_n \cdot x_{k-n} + \Delta_k \tag{2.1}$$

$$\alpha^T = \begin{bmatrix} a_0 & a_1 & a_2 & \cdots & a_n \end{bmatrix} \tag{2.2}$$

$$\Phi_k^T = \begin{bmatrix} 1 & x_{k-1} & x_{k-2} & \cdots & x_{k-n} \end{bmatrix} \tag{2.3}$$

$$x_k = \Phi_k^T \alpha + \Delta_k \tag{2.4}$$

where Δ_k is the modeling error,

$$\Delta_k = x_k - \Phi_k^T \alpha \tag{2.5}$$

Sum of the least squares of the modeling errors for N data points can be expressed as follows,

$$E(\alpha, N) = \sum_{k=1}^{N} (x_k - \Phi_k^T \alpha)^2 \tag{2.6}$$

$$E(\alpha, N) = \sum_{k=1}^{N} \Big[(x_k - \Phi_k^T \alpha)^T (x_k - \Phi_k^T \alpha) \Big]$$

$$E(\alpha, N) = \sum_{k=1}^{N} (x_k^T x_k - x_k^T \Phi_k^T \alpha - \alpha^T \Phi_k x_k + \alpha^T \Phi_k \Phi_k^T \alpha) \tag{2.7}$$

Note that $x_k = \Phi_k^T \alpha + \Delta_k$ and $(\alpha^T \Phi_k x_k)^T = x_k^T \Phi_k^T \alpha$,

$$(\alpha^T \Phi_k x_k)^T = x_k^T (x_k - \Delta_k) \tag{2.8}$$

Since x_k is a (1×1) vector, $\alpha^T \Phi_k x_k$ is a scalar value and equal to its own transpose.

$$\alpha^T \Phi_k x_k = x_k^T \Phi_k^T \alpha \tag{2.9}$$

Then equation (2.7) can be simplified as,

$$E(\alpha, N) = \sum_{k=1}^{N} (x_k^T x_k - 2\alpha^T \Phi_k x_k + \alpha^T \Phi_k \Phi_k^T \alpha) \tag{2.10}$$

Differentiating both sides with respect to α,

$$\frac{\partial E}{\partial \alpha} = \sum_{k=1}^{N} (-2\Phi_k x_k + 2\Phi_k \Phi_k^T \alpha)$$

$$\frac{\partial E}{\partial \alpha} = 2 \sum_{k=1}^{N} (-\Phi_k x_k + \Phi_k \Phi_k^T \alpha) \tag{2.11}$$

Second derivative of $E(\alpha, N)$ with respect to α,

$$\frac{\partial^2 E}{\partial \alpha^2} = 2\sum_{k=1}^{N}(\Phi_k^T \Phi_k) \tag{2.12}$$

Due to the definition of Φ_k, Φ_k cannot be a zero vector and the multiplication of a non-zero vector Φ_k by its transpose always gives a positive value.

$$\forall\, \Phi_k,\ (\Phi_k^T \Phi_k) > 0 \implies \frac{\partial^2 E}{\partial \alpha^2} > 0 \tag{2.13}$$

Therefore, $E(\alpha, N)$ always decreases as $k \to N$ and has a global minimum at $k = N$ for N data points.

To satisfy the necessary condition for a relative extremum, equating the first derivative of $E(\alpha, N)$ to zero is enough to find the best modeling parameter $\hat{\alpha_k}$ that minimizes $E(\alpha, N)$ according to the second derivative test of $E(\alpha, N)$.

Then the equation (2.11) becomes,

$$0 = \sum_{k=1}^{N}(-\Phi_k x_k + \Phi_k \Phi_k^T \hat{\alpha_k}) \tag{2.14}$$

Rearranging for the modeling paramater $\hat{\alpha_k}$,

$$\sum_{k=1}^{N}(\Phi_k x_k) = \sum_{k=1}^{N}(\Phi_k \Phi_k^T \hat{\alpha_k})$$

$$\hat{\alpha_k} = \left(\sum_{k=1}^{N} \Phi_k \Phi_k^T\right)^{-1} \left(\sum_{k=1}^{N} \Phi_k x_k\right) \tag{2.15}$$

2.2 Recursive Linear Least Squares Regression

Assume $\alpha^T = \begin{bmatrix} a_0 & a_1 & a_2 & \cdots & a_n \end{bmatrix}$ is constant for a limited time frame **R**, then x_k can be used to calculate the next **R** states.

$$x_k = a_0 + a_1 \cdot x_{k-1} + a_2 \cdot x_{k-2} + \cdots + a_n \cdot x_{k-n}$$

$$x_{k+1} = a_0 + a_1 \cdot x_k + a_2 \cdot x_{k-1} + \cdots + a_n \cdot x_{k-n+1}$$

$$x_{k+2} = a_0 + a_1 \cdot x_{k+1} + a_2 \cdot x_k + \cdots + a_n \cdot x_{k-n+2}$$

$$\vdots$$

$$x_{k+R} = a_0 + a_1 \cdot x_{k+R-1} + a_2 \cdot x_{k+R-2} + \cdots + a_n \cdot x_{k-n+R}$$

The recursive least squares algorithm allows the prediction of the next **R** future states assuming a constant modeling parameter α which is calculated from equation (2.15) using x_k and the **P** past values of x.

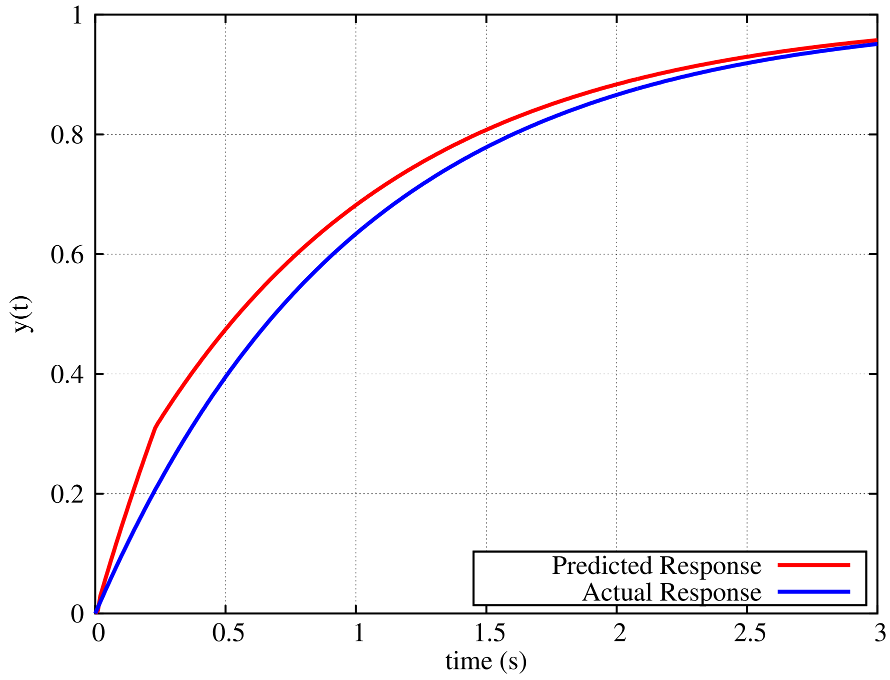


Figure 2.1: Prediction of Unit Step Input

An example of using Recursive Least Squares Regression for prediction is shown in figure 2.1. Unit step input is applied at the beginning of the simulation for system $\frac{1}{s+1}$. In this example, Recursive Least Squares Regression predicts 35^{th} next value using 20 values in the past.

2.3 Limitations

As in most statistical model, Least Squares Regression has some limitations. Most common problems of Least Squares Regression are outlier sensitivity, non- linearity, dealing with high numbers of variables, dependencies between independent variables, heteroskedasticity and variances in independent variables.[34].

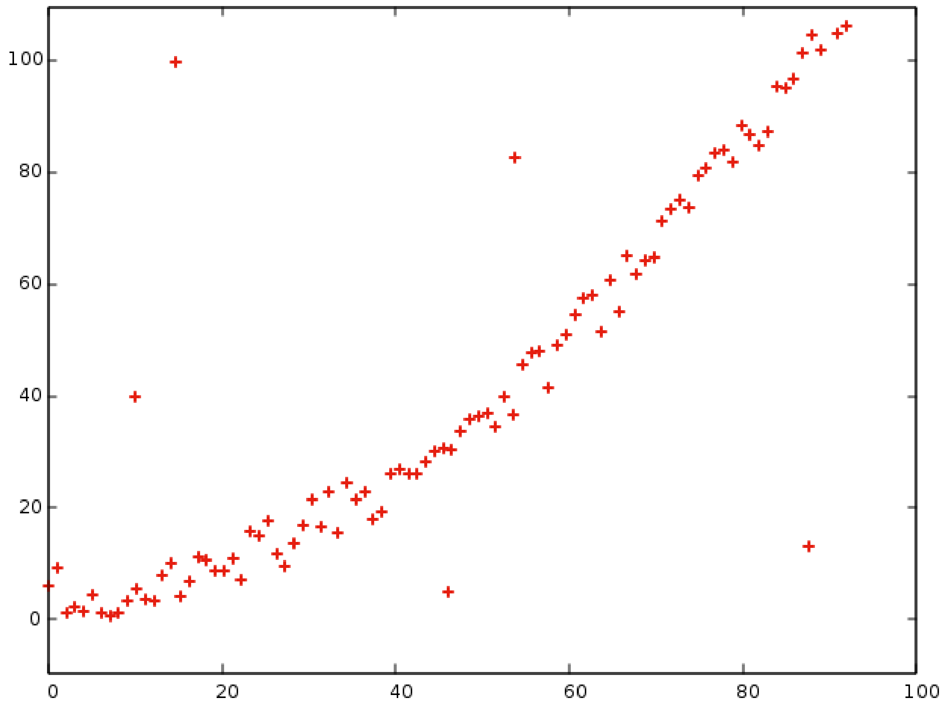


Figure 2.2: Outliers in a Data Set

Outlier is a distinct point in a data set which lies outside of the overall pattern[35] as shown in figure 2.2. For a continuous system with a very tiny step size, outliers are not expected and hence outliers does not create a problem for the least square regression. Hence, choosing a step size of 0.01 seconds and using a continuous integration method like Euler on MATLAB Simulink is enough to minimize outliers. Effect of non-linearity of the system is eliminated by choosing a tiny step size and decreasing the number of independent variables of regression. Modeling errors due to high numbers of variables is avoided by choosing the number of independent variables much smaller than the number of available data points[34].

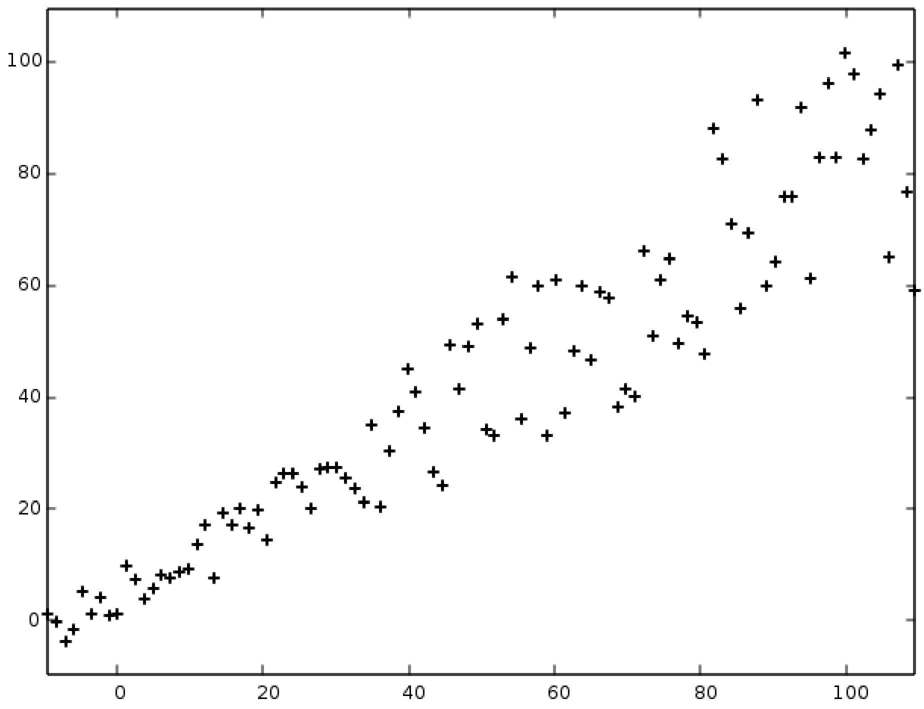


Figure 2.3: Heteroskedasticitic Data

Since, least square regression is applied for a stable closed loop system, heteroskedasticity, i.e. inconstant variance is unexpected as depicted in figure 2.3. Moreover, for a stable system, variance of independent variables does not create a disturbance for regression.

CHAPTER 3

LEAST SQUARES BASED ADAPTIVE CONTROL

Adaptive control is achieved by modeling the error dynamics and finding the optimum PID gains using Least Squares Regression. For a quick gain optimization, proper initial PID gains should be used. It is also possible to start from any initial guess like a system identification process. In this thesis, an asymptotically stable closed loop system is selected as a starting point to increase the success of the optimization. In addition, the limitations of Least Squares Regression which are mentioned in Chapter 2 restrict the usage of this method for mostly stable systems.

3.1 Modeling Error Dynamics

The output of conventional PID controller is the weighted sum of proportional, integral and derivative channels.

$$u_k = K_P\, e_k + K_I \int e_k\, dt + K_D\, \dot{e_k} \tag{3.1}$$

where K_P, K_I and K_D are the proportional, integral and derivative gains respectively.

Error that is minimized by PID controller is the difference between the desired and the current values of a state.

$$e_k = x_{dk} - x_k \tag{3.2}$$

Each channel of a PID controller is modeled using Least Square Regression as shown below,
Proportional Channel:

$$e_k = P_0 + P_1 x_{k-1} + P_2 x_{k-2} + \ldots + P_n x_{k-n} + \Delta_{Pk} \tag{3.3}$$

Integral Channel:

$$\int e_k \, dt = I_0 + I_1 x_{k-1} + I_2 x_{k-2} + \ldots + I_n x_{k-n} + \Delta_{Ik} \tag{3.4}$$

Derivative Channel:

$$\frac{d\,e_k}{dt} = D_0 + D_1 x_{k-1} + D_2 x_{k-2} + \ldots + D_n x_{k-n} + \Delta_{Dk} \tag{3.5}$$

The modeling errors of the each channel are Δ_{Pk}, Δ_{Ik} and Δ_{Dk} respectively.

Error dynamics is in matrix form,

$$\begin{bmatrix} e_k \\ \int e_k \, dt \\ \dot{e_k} \end{bmatrix} = \begin{bmatrix} P_0 & P_1 & P_2 & \cdots & P_n \\ I_0 & I_1 & I_2 & \cdots & I_n \\ D_0 & D_1 & D_2 & \cdots & D_n \end{bmatrix} \begin{bmatrix} 1 \\ x_{k-1} \\ x_{k-2} \\ \vdots \\ x_{k-n} \end{bmatrix} + \begin{bmatrix} \Delta_{Pk} \\ \Delta_{Ik} \\ \Delta_{Dk} \end{bmatrix} \tag{3.6}$$

3.2 PID Gain Update Law

Modeling errors are used for updating the PID gains using a learning rate η for each channel as follows,

$$\Delta K_{Pk} = \eta_P \, \Delta_{Pk} \tag{3.7}$$

$$\Delta K_{Ik} = \eta_I \, \Delta_{Ik} \tag{3.8}$$

$$\Delta K_{Dk} = \eta_D \, \Delta_{Dk} \tag{3.9}$$

PID gain update law in matrix form,

$$\begin{bmatrix} \Delta K_{Pk} \\ \Delta K_{Dk} \\ \Delta K_{Ik} \end{bmatrix} = \begin{bmatrix} \eta_P & 0 & 0 \\ 0 & \eta_I & 0 \\ 0 & 0 & \eta_D \end{bmatrix} \left(\begin{bmatrix} e_k \\ \int e_k \, dt \\ \dot{e_k} \end{bmatrix} - \begin{bmatrix} P_0 & P_1 & P_2 & \cdots & P_n \\ I_0 & I_1 & I_2 & \cdots & I_n \\ D_0 & D_1 & D_2 & \cdots & D_n \end{bmatrix} \begin{bmatrix} 1 \\ x_{k-1} \\ x_{k-2} \\ \vdots \\ x_{k-n} \end{bmatrix} \right) \tag{3.10}$$

An adaptive controller is obtained from the modeling errors of the Linear Least Squares Regression of the error dynamics. These modeling errors are multiplied by a learning rate for each channel and added to the gains of the PID controller to calculate the new PID gains for the next time step as shown below,

$$K_{Pk+1} = K_{Pk} + \Delta K_{Pk} \tag{3.11}$$

$$K_{Ik+1} = K_{Ik} + \Delta K_{Ik} \tag{3.12}$$

$$K_{Dk+1} = K_{Dk} + \Delta K_{Dk} \tag{3.13}$$

According to the proof in section 2.1, as the number of data points for least squares regression increases, the modeling error of the least square regression decreases. Thus, increasing the number of data points that have similar variance in a closed domain of the target state minimizes the regression error. An asymptotically stable system satisfies this condition. As stated in Chapter 2, Least Square Regression is applied for an asymptotically stable closed loop system. Therefore, the least squares based adaptive controller is also asymptotically stable and PID gain updates go to zero for sufficiently small learning rates.

The limitations of this new PID gain update law depend on the limitations of the Linear Least Squares Regression that is mentioned in Chapter 2.

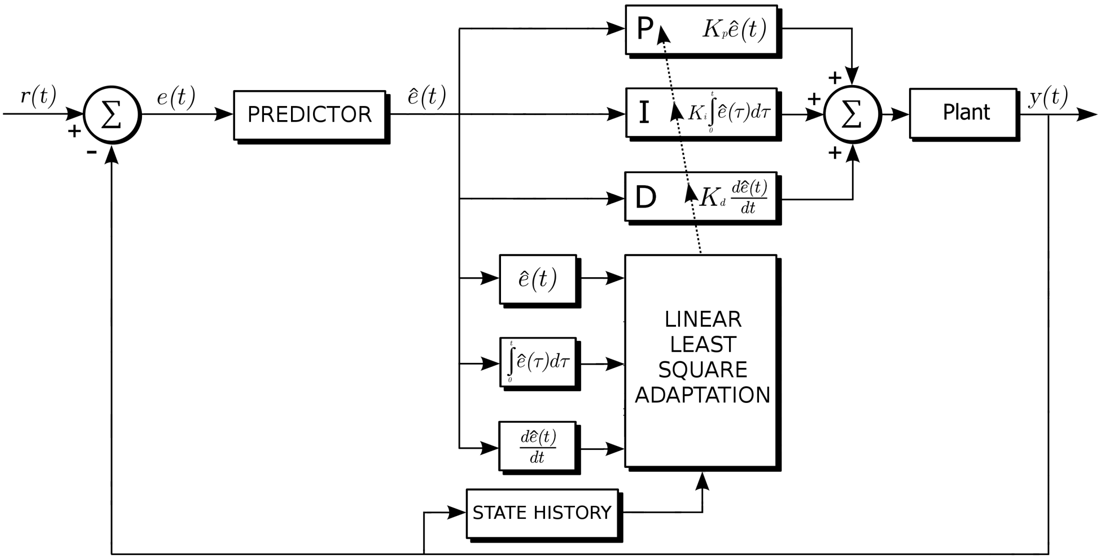


Figure 3.1: Block Diagram of the Intelligent Controller

3.3 Intelligent Controller Design

Classical PID controller design is improved by adding prediction and adaptation capabilities that convert the PID to an intelligent controller as depicted in figure 3.1. In order to achieve this, an error predictor is developed using Recursive Linear Least Squares Regression which predicts the 35^{th} future value of the error from 20 past values of the error. Instead of feeding the instantaneous error, this predicted error is used as an input to the controller. In addition to the predictor, a state history block is implemented to log the 20 past values of the state. Error dynamics of the system is modeled using these past values with Least Squares Regression. After adaptation is completed in Least Squares based Adaptive Controller by updating PID gains with the errors of the error dynamics as explained in equation 3.10, the optimum PID gains are obtained.

CHAPTER 4

INTELLIGENT FLIGHT CONTROLLER ARCHITECTURE

The main objective of this thesis is to design an intelligent model prediction controller for an autonomous helicopter. For this purpose, adaptive controller architectures [16] and [36] are analysed and improved without using a neural network for adaptation. In addition, the controller is optimized for both hover and forward flight conditions. Finally, an intelligent model prediction controller is implemented and an Intelligent Flight Controller (IFC) is obtained.

4.1 Flight Controller Design

As in the previous controller design[16], flight controller consist of a trajectory generator, outer loop for position control and inner loop for attitude control. In addition, body angular velocity controller is added as a third loop. Decoupling the position and attitude control, improves the controller efficiency for faster rotational dynamics. In addition to decoupling, the controller has two command filters for both inner loop and outer loop to eliminate oscillations. Also, an actuator model is included in the inner loop for swashplate dynamics.

The previous design[16] has an effective feed forward mechanism for PID controller that increases the performance of the controller significantly. This feed forward mechanism is arisen by the summation of the second derivative of the command with the control output of PID and thus an improved PID controller (I-PID) is obtained as illustrated in figure 4.1.

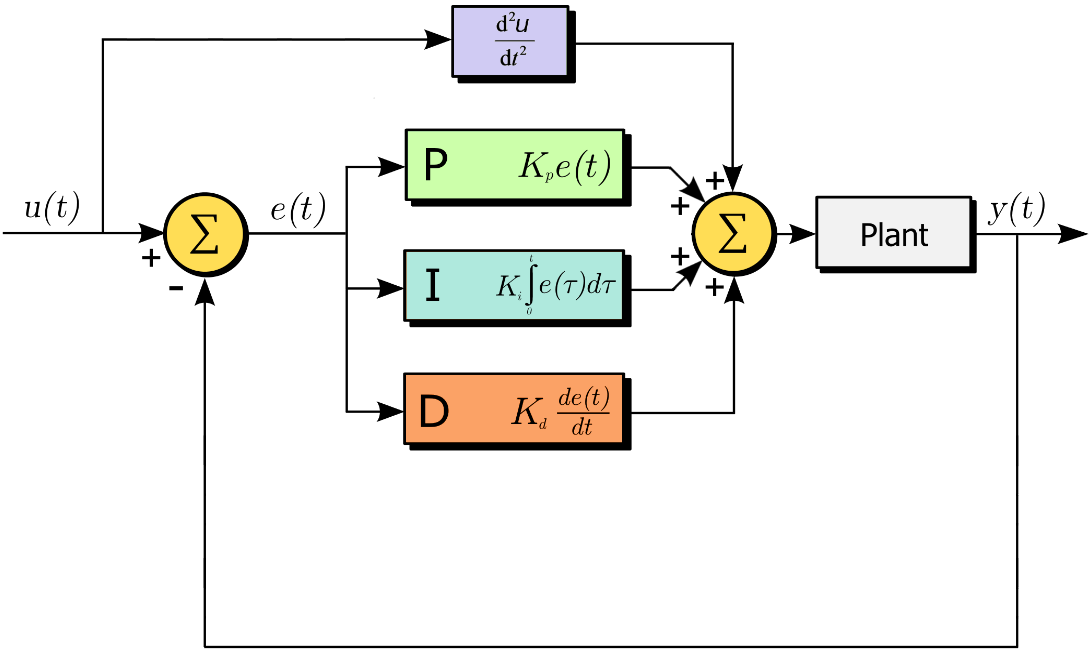


Figure 4.1: Block Diagram of the I-PID Controller

In the previous design[16], model inversion technique is used to generate control inputs. An error always occurs due to the approximations for inversion. Moreover, as the complexity and fidelity of the model increases, it gets harder to invert the model and model inversion error increases. An online learning capable neural network based adaptive controller is used to overcome this modeling error in the previous study[16]. In spite of using an adaptive controller, the model inversion has still a disadvantage that the stability and control matrices are assumed to be time invariant and linearized for only one flight condition. Because of this assumption, as flight condition changes from the initial trim point where the linearization is done, model inversion error increases. Therefore, since the model inversion method has a restricted usage due to the assumption of the constant stability and the control matrices of the system, instead of using the model inversion method in the inner loop, a Body Angular Velocity Controller (BAVC) is implemented as a third loop (most-inner loop) for the new flight controller.

4.1.1 Actuator Model

A second order filter is used as an actuator model. The natural frequency of the actuator model is chosen as 70 rad/s to be faster than helicopter dynamics. The damping ratio is selected as 0.6 to prevent the flattening out the dynamics. In addition, the actuator model has angle and rate saturations to model the actual swashplate dynamics. The transfer function of the actuator model is as shown below,

$$\frac{4900}{s^2 + 84\,s + 4900} \tag{4.1}$$

4.1.2 Command Filter

A second order command filter is also used for both outer and inner loops to eliminate the control oscillations. The natural frequency of the command filter is chosen as 1 rad/s to have a slower dynamics than the controllers of the position and attitude channels. The damping ratio of the command filter is chosen as 0.8 to have an extra feed forward effect for controllers.The transfer function of the command filter is as follows,

$$\frac{1}{s^2 + 1.6\,s + 1} \tag{4.2}$$

Actuator model and command filter settings are taken from the previous study. But, there is an obligatory change for the Acutator Model, since the helicopter is AH-1S in the previous thesis[16] and has different control margins from UH-1H. Actuator limits of UH-1H are shown in the table 4.1. These limits are obtained during the development of Heli-Dyn[37].

Table 4.1: Actuator Limits

	Longitudinal Swashplate	Collective	Lateral Swashplate	Pedal
Angle Limit (deg)	± 8	0 - 20	± 8	-5 - 20
Rate Limit (deg/s)	± 10	± 10	± 10	± 10

4.1.3 Trajectory Generator

Trajectory Generator consists of maneuver libraries that generate the necessary position commands for the north, east, altitude and heading channels according to the desired maneuver. These commands are sent to the outer loop after filtered by the Command Filter. Complex maneuvers like pull-up pushover, slalom, coning, and pirouette maneuvers are selected to challenge the adaptive controller for a realistic flight. In addition to these maneuvers that are used in the previous study [16], the flight controller is also tested for two more manuvers which are combination of slalom and pull up - push over maneuver and 3-D cone maneuver respectively.

4.1.4 Outer Loop

The Outer loop is responsible for controlling the three positions in North-East-Down (NED) navigation frame. In more detail, the outer loop receives the target north and east positions, altitude and heading commands from the trajectory generator after smoothing by a second order filter and sends the desired roll, pitch and yaw Euler angles to the inner loop. Desired Euler angles are produced in Translational Dynamic Inverse Block from the outputs of I-PID velocity controllers. Differently from the reference[16], instead of using position controllers, velocity controllers are used in the outer loop to increase the controller performance. Hence, oscillations and steady state errors are decreased especially for north channel.

Heading command is sent to inner loop without any change in the outer loop. North and East channels are controlled by I-PID controllers. Transient performance of the controller is improved by using a derivative controller and an integral controller is included to minimize the steady state error for accurate path tracking.

Altitude is also controlled in the outer loop from desired down velocity instead of controlling from the desired total body acceleration in the inner loop as in the references [16] and [36]. Although only an integral controller is used for collective control in reference [36], a proportional controller is included in order to increase stability. According to the simulation tests, derivative controller is not effective for altitude channel, hence only a PI controller is used for altitude control.

4.1.4.1 Translational Dynamic Inverse Block

This block converts the accelerations from North-East-Down frame to body frame using the transformation matrix L_{VB}. Then from the body accelerations, desired roll and pitch angles are acquired.

$$\begin{bmatrix} \ddot{X}_V \\ \ddot{Y}_V \\ \ddot{Z}_V \end{bmatrix} = L_{VB}(\phi, \theta, \psi) \begin{bmatrix} \ddot{X}_B \\ \ddot{Y}_B \\ \ddot{Z}_B \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \tag{4.3}$$

Since $L_{VB}^{-1} = L_{BV}$,

$$\begin{bmatrix} \ddot{X}_B \\ \ddot{Y}_B \\ \ddot{Z}_B \end{bmatrix} = L_{BV}(\phi, \theta, \psi) \begin{bmatrix} U_{XN} \\ U_{XE} \\ U_{XAlt} - g \end{bmatrix} \tag{4.4}$$

In order to simplify the computations, the required cyclic and pedal control forces are assumed to be much smaller than the collective control force and they are neglected. In addition, body x-axis and y-axis aerodynamic force components F_x and F_y are assumed to be much smaller than body z-axis aerodynamic force component F_z and they are also neglected[36].

Using these approximations, desired pitch and roll Euler angles are obtained from the following equations,

$$\phi_d \approx \arcsin(\frac{-U_{XN}\, sin(\psi_d) + U_{XE}\, cos(\psi_d)}{\sqrt{U_{XN}^2 + U_{XE}^2 + (U_{XAlt} - g)^2}}) + \phi_{trim} \tag{4.5}$$

$$\theta_d \approx \arctan(\frac{U_{XN}\, cos(\psi_d) + U_{XE}\, sin(\psi_d)}{U_{XAlt} - g}) + \theta_{trim} \tag{4.6}$$

4.1.5 Inner Loop

In the inner loop, desired roll and pitch angles are received from the outer loop and processed to generate desired body angular accelerations for the innermost loop. For this process, the relation matrix between Euler angle rates and body angular velocities is used as shown in equation 4.7.

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -sin(\theta) \\ 0 & cos(\phi) & sin(\phi)\,cos(\theta) \\ 0 & -sin(\phi) & cos(\phi)\,cos(\theta) \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{4.7}$$

Since the integral controller has adverse effects for attitude control, for each attitude channel an I-PD controller is used. Transient performance of the attitude channels are improved by implementing a derivative controller for each channel. The commands for attitude controllers are passed through the command filter and taken by Euler Angle Controller (EAC) which consists of three I-PD attitude controllers. The outputs of EAC are sent to the innermost loop to generate longitudinal cyclic, lateral cyclic and pedal controls from desired body angular velocities.

Inner loop also includes the innermost loop that is responsible from controlling the body angular velocities.

4.1.5.1 Innermost Loop

As mentioned before, necessary control inputs are generated according to the desired body angular velocities which are received from the inner loop. Th innermost loop or in other words Body Angular Velocity Controller (BAVC) generates longitudinal cyclic, lateral cyclic and pedal controls from desired body angular velocities using I-PID controllers for each channel.

4.2 Controller Optimization for Hover and Forward Flight

The new flight controller is capable of both flying at hover and forward flight conditions. Therefore, gain scheduling is used for the velocity controller and the command filter to optimize the transition between hover and forward flight. The gain scheduling is achieved by setting nonlinear gain equations. First, variable derivative and integral gains are used for east velocity controller rather than constant ones. The following equations are used for the gain scheduling,

$$I = min\left(\frac{V_{Total}}{250}, 1\right) + 0.02 \cdot \left(\frac{V_{Total}}{10}\right)^{2} \tag{4.8}$$

$$D = \frac{0.7 \cdot V_{Total} + 13}{8} + 0.1 \cdot min\left(\frac{V_{Total}}{50}, 1\right) \tag{4.9}$$

where velocities are in knots and V_{Total} is obtained from,

$$V_{Total} = min\left(\sqrt{V_N^2 + V_E^2}\,,\, 50\sqrt{2}\right) \tag{4.10}$$

Second, damping ratios for north and east channels of the command filter in the outer loop are changed according to the 4.11 and 4.12 instead of using constant values. The damping ratios are increased with respect to forward flight speed as shown in figure 4.2 to increase the stability of the transition from hover to forward flight.

$$\xi_{North} = 0.8\, +\, min(0.001 \cdot |\Delta V_{North}|^{2.1}, 30) \tag{4.11}$$

$$\xi_{East} = 0.8\, +\, min(0.001 \cdot |\Delta V_{East}|^{2.5}, 30) \tag{4.12}$$

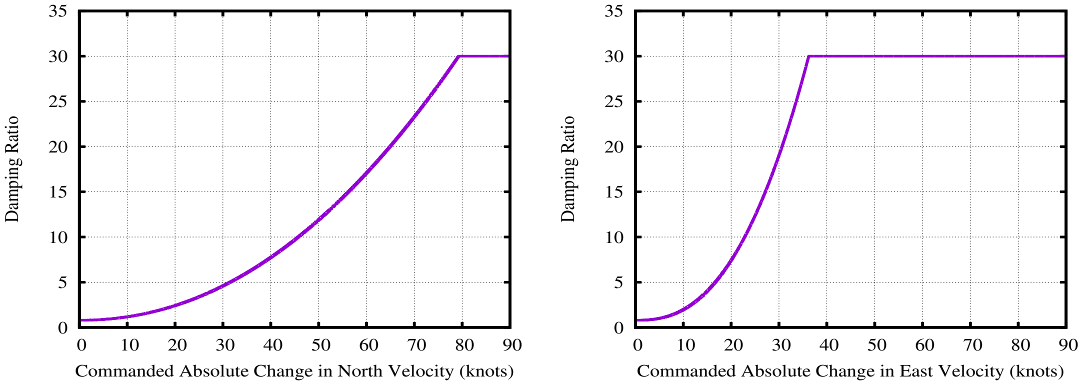


Figure 4.2: Command Filter Damping Changes for North and East Veloctiy Commands

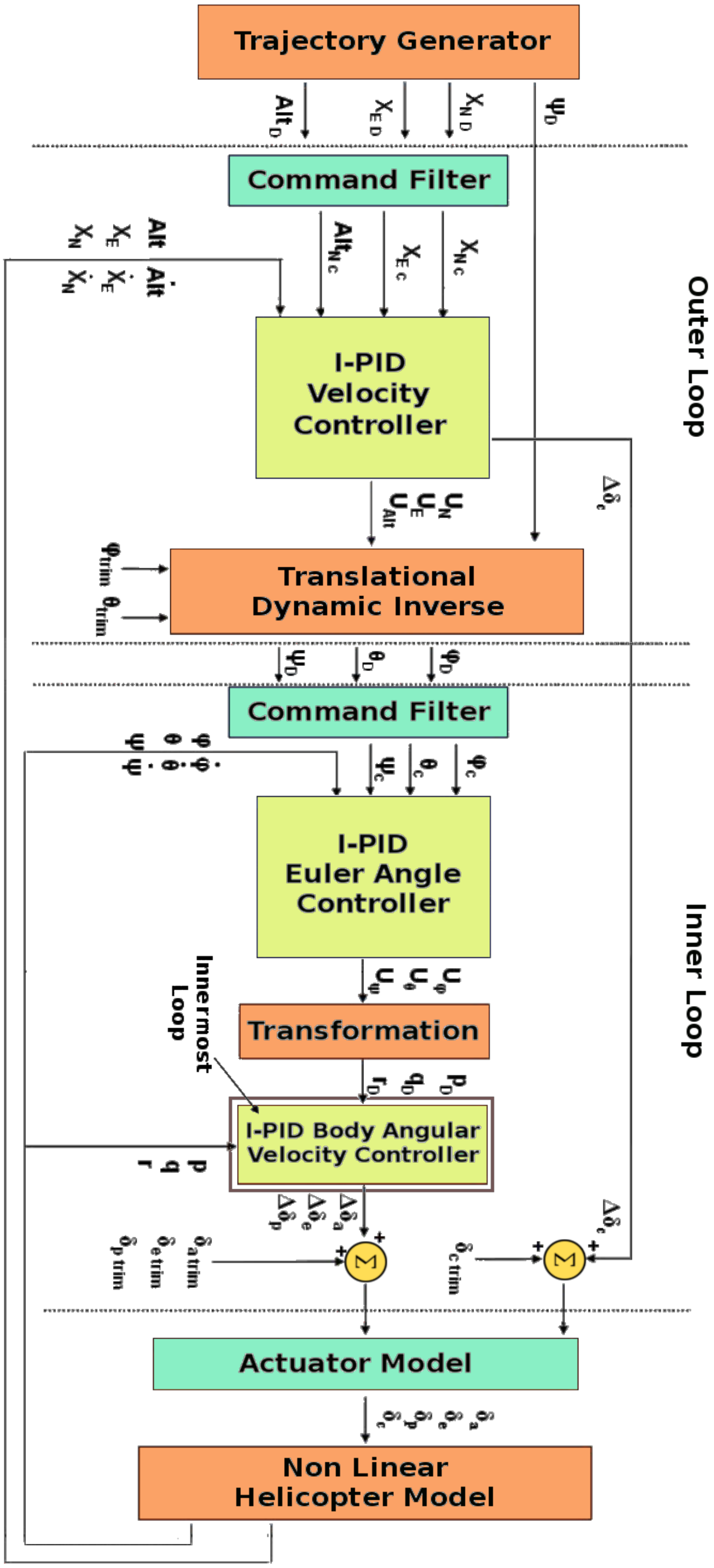

Trajectory Generator
Command Filter
I-PID Velocity Controller
Translational Dynamic Inverse
Command Filter
I-PID Euler Angle Controller
Transformation
I-PID Body Angular Velocity Controller
Actuator Model
Non Linear Helicopter Model
Outer Loop
Inner Loop
Innermost Loop

Figure 4.3: Intelligent Flight Controller Design

4.3 Intelligent Controller Implementation

In this study, capabilities of I-PID controller are extended by adding model prediction and adaptation abilities which leads to an intelligent controller as described in Chapter 3. Each of the I-PID controllers in the inner loop (EAC and BAVC) is replaced with this intelligent I-PID controller. Thus, a new adaptive and model predictive Intelligent Flight Controller (IFC) is acquired. As mentioned before, IFC has three control loops as shown in figure 4.3. The outer loop of IFC has only I-PID controllers. But, the inner and the innermost loop contains Intelligent I-PID Controllers.

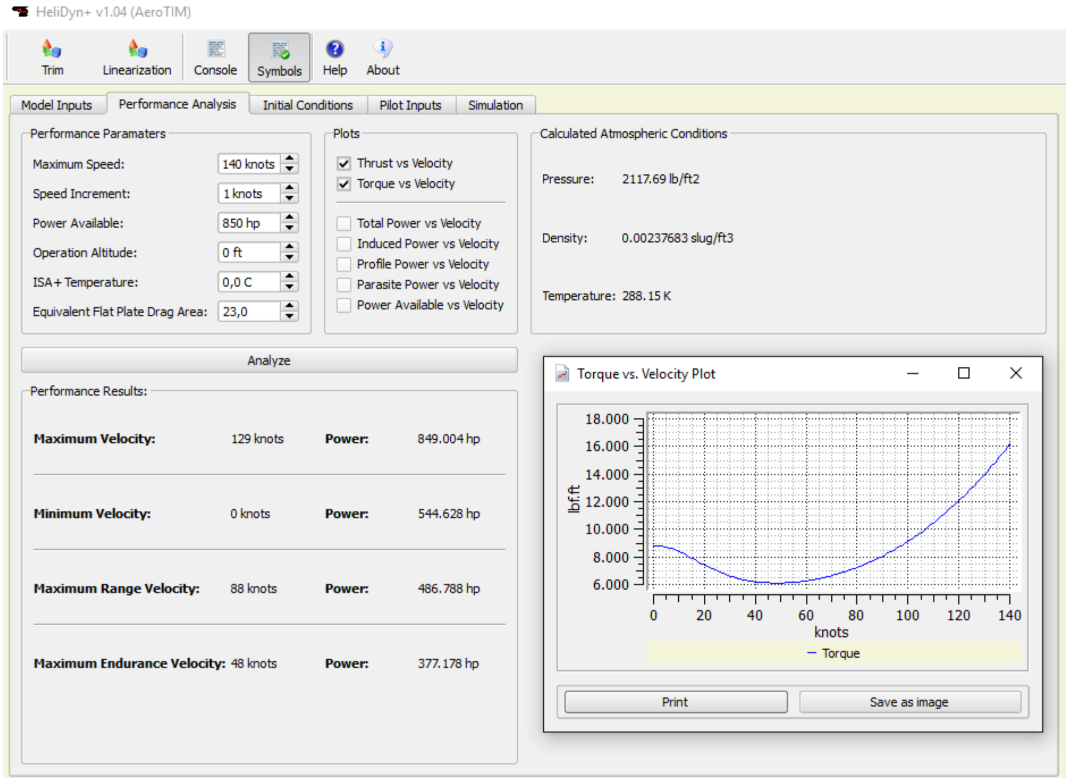
CHAPTER 5

MODELING HELICOPTER

For the simulation tests, the UH-1H helicopter is chosen since it is a known helicopter in literature[38]. A software for rotorcraft modeling and simulation called Heli-Dyn[37] is used for modeling the helicopter.

Heli-Dyn was firstly developed by AeroTIM with the support of TUBITAK and KOSGEB. The author of this thesis is also a member of the core development team of the first several versions of Heli-Dyn. It is a dynamic modeling tool which is capable of modeling a helicopter, trimming, linearizing around any trim points and also conducting basic performance analysis that supports for different ISA temperature conditions as shown in figure 5.1. In addition, any external simulation and software environments can be integrated with Heli-Dyn. Providing geometric, inertial and aerodynamic data any helicopter can be modeled using this software.

Heli-Dyn uses a component build-up technique for generating the whole helicopter model. In component built-up method, the forces and moments generated by each component of the helicopter are integrated at the center of gravity of the helicopter leading the usage of 6-DOF (Degree of Freedom) rigid body dynamics. This method allows an interchangeability for the users and developers. Any component can be replaced with a more sophisticated high fidelity model or there may be lots of versions for a specific component with different fidelities as shown in figure 5.2.



Performance Paramaters

Maximum Speed: 140 knots

Speed Increment: 1 knots

Power Available: 850 hp

Operation Altitude: 0 ft

ISA+ Temperature: 0,0 C

Equivalent Flat Plate Drag Area: 23,0

Plots

☑ Thrust vs Velocity

☑ Torque vs Velocity

□ Total Power vs Velocity

□ Induced Power vs Velocity

□ Profile Power vs Velocity

□ Parasite Power vs Velocity

□ Power Available vs Velocity

Calculated Atmospheric Conditions

Pressure: 2117.69 lb/ft2

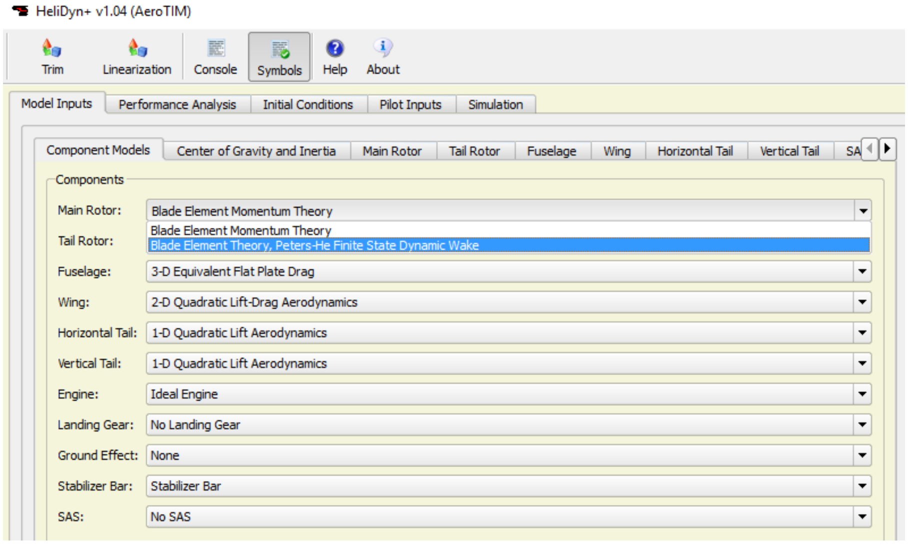
Density: 0.00237683 slug/ft3

Temperature: 288.15 K

Performance Results:

Maximum Velocity:	129 knots	**Power:**	849.004 hp
Minimum Velocity:	0 knots	**Power:**	544.628 hp
Maximum Range Velocity:	88 knots	**Power:**	486.788 hp
Maximum Endurance Velocity:	48 knots	**Power:**	377.178 hp

Figure 5.1: Performance Analysis Using Heli-Dyn



Components

Main Rotor: Blade Element Momentum Theory

Tail Rotor: Blade Element Momentum Theory / Blade Element Theory, Peters-He Finite State Dynamic Wake

Fuselage: 3-D Equivalent Flat Plate Drag

Wing: 2-D Quadratic Lift-Drag Aerodynamics

Horizontal Tail: 1-D Quadratic Lift Aerodynamics

Vertical Tail: 1-D Quadratic Lift Aerodynamics

Engine: Ideal Engine

Landing Gear: No Landing Gear

Ground Effect: None

Stabilizer Bar: Stabilizer Bar

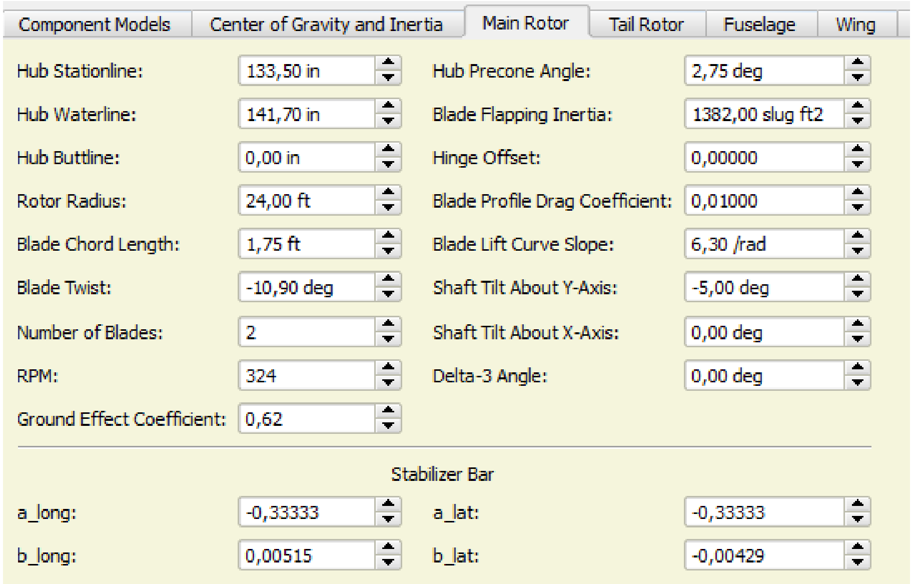
SAS: No SAS

Figure 5.2: Modeling Components of Heli-Dyn

Main rotor, tail rotor, fuselage, wing, horizontal tail, vertical tail, engine, landing gear and stabilizer bar are the basic components in Heli-Dyn for modeling a helicopter. Moreover, the software offers a ground effect model. Lastly, a SAS model can be added to ease the usage of the helicopter or for testing purposes.

Each model component includes different models with various fidelities. For the main rotor component, there are two versions; the well known "Minimum Complexity" model[39] which uses Blade Element Momentum Theory and Blade Element Moementum with Peters-He Inflow model[40]. These models were validated by the development team in 2008[41]. The minimum complexity model uses first order flapping, uniform inflow and an iterative approach to the classic momentum and Glauert Theories for force and moment calculations. Peters-He Inflow model of Heli-Dyn uses 3-state Peters-He inflow models for main rotor blade element solutions.

The main difference between these two main rotor models is the main rotor thrust and inflow calculations. While, minimum complexity accepts flow distribution as uniform and calculates thrust from momentum theory, Peters-He inflow distribution model computes thrust by sectioning the main rotor blades. For real time performance considerations, section number is limited to three for the inflow model in Heli-Dyn. The Peters-He inflow model has some advantages on blade element momentum models like minimum complexity such that it generates an improved pressure distribution across a rotor plane including tip loss[41]. In this thesis, the minimum complexity model is used for the main rotor and tail rotor components. Geometric inputs for the main rotor component and stabilizer bar are given in figure 5.3 and the tail rotor geometry inputs are specified as shown in figure 5.4



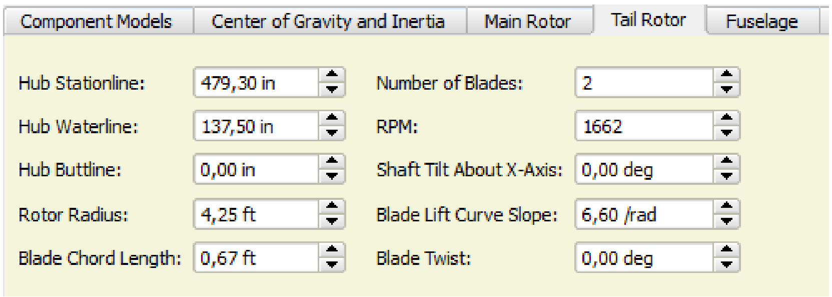
Hub Stationline:	133,50 in	Hub Precone Angle:	2,75 deg
Hub Waterline:	141,70 in	Blade Flapping Inertia:	1382,00 slug ft2
Hub Buttline:	0,00 in	Hinge Offset:	0,00000
Rotor Radius:	24,00 ft	Blade Profile Drag Coefficient:	0,01000
Blade Chord Length:	1,75 ft	Blade Lift Curve Slope:	6,30 /rad
Blade Twist:	-10,90 deg	Shaft Tilt About Y-Axis:	-5,00 deg
Number of Blades:	2	Shaft Tilt About X-Axis:	0,00 deg
RPM:	324	Delta-3 Angle:	0,00 deg
Ground Effect Coefficient:	0,62		

Stabilizer Bar

a_long:	-0,33333	a_lat:	-0,33333
b_long:	0,00515	b_lat:	-0,00429

Figure 5.3: Geometric Inputs for Main Rotor Component

The data of geometric inputs of main rotor and tail rotor are the default values in the HeliDyn software for UH-1H and are validated in the study[41].

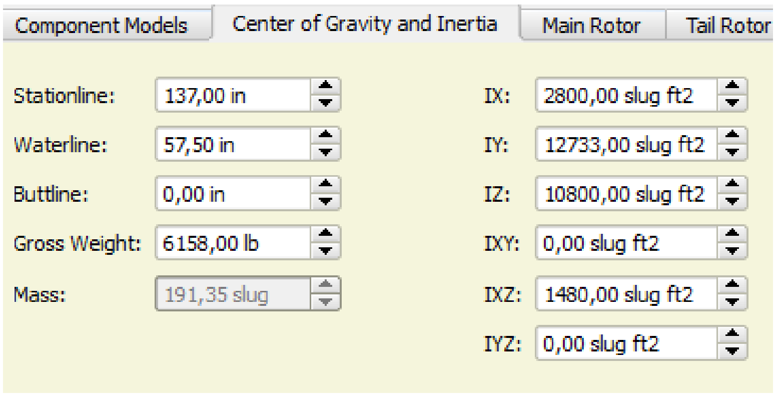


Hub Stationline:	479,30 in	Number of Blades:	2
Hub Waterline:	137,50 in	RPM:	1662
Hub Buttline:	0,00 in	Shaft Tilt About X-Axis:	0,00 deg
Rotor Radius:	4,25 ft	Blade Lift Curve Slope:	6,60 /rad
Blade Chord Length:	0,67 ft	Blade Twist:	0,00 deg

Figure 5.4: Geometric Inputs for Tail Rotor Component

Other basic components such as wings, horizontal stabilizer, vertical tail, etc. are modeled by simple aerodynamic equations with constant coefficients[42].

Mass and inertia data are taken from the reference[38] as stated in figure 5.5.



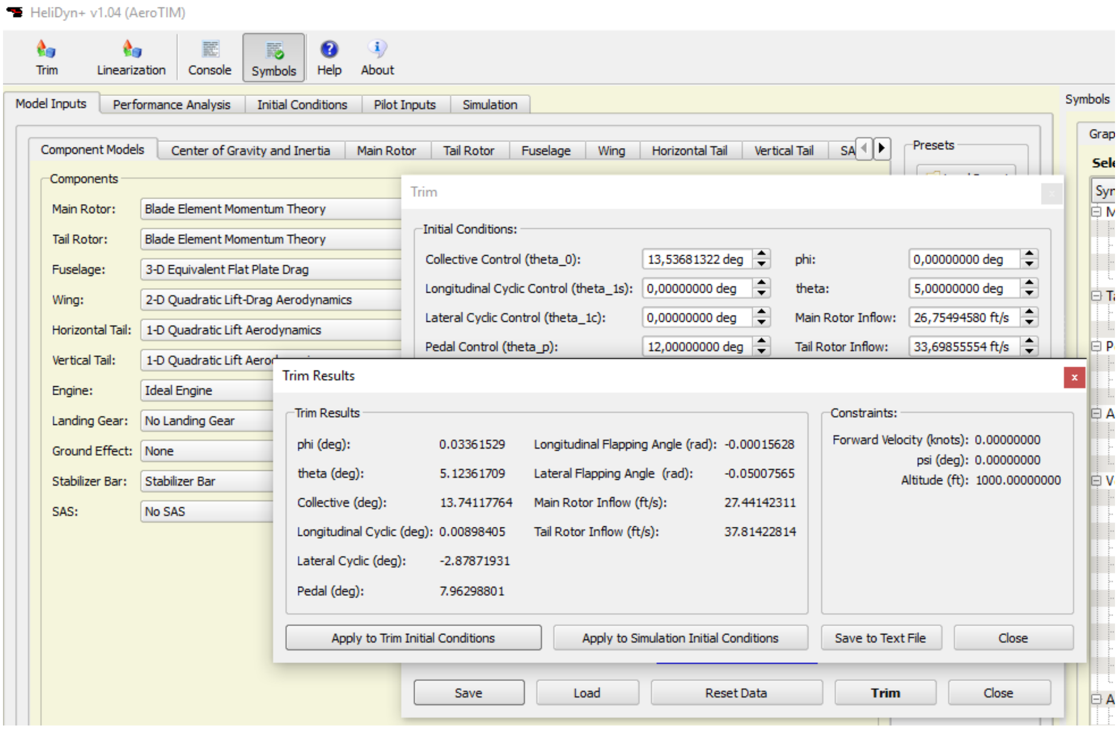
Stationline:	137,00 in	IX:	2800,00 slug ft2
Waterline:	57,50 in	IY:	12733,00 slug ft2
Buttline:	0,00 in	IZ:	10800,00 slug ft2
Gross Weight:	6158,00 lb	IXY:	0,00 slug ft2
Mass:	191,35 slug	IXZ:	1480,00 slug ft2
		IYZ:	0,00 slug ft2

Figure 5.5: Mass and Inertia Data of UH-1H

In this study, Heli-Dyn v1.04 is used to model the UH-1H with the default geometry and components settings as shown below,

- Main Rotor: Blade Element Momentum Theory
- Tail Rotor: Blade Element Momentum Theory
- Fuselage: 3-D Equivalent Flat Plate Drag
- Wing: 2-D Quadratic Lift Aerodynamics
- Horizontal Tail: 1-D Quadratic Lift Aerodynamics
- Vertical Tail: 1-D Quadratic Lift Aerodynamics
- Engine: Ideal Engine
- Landing Gear: None
- Ground Effect: None
- Stabilizer Bar: Default
- SAS: None

Since all simulations start from 1000 ft hover and the minimum altitude for each maneuver is higher than 750 ft during flight, ground effect and landing gear models are not used. Using these settings, UH-1H model is trimmed at 1000 ft for hover as depicted in figure 5.6. The simulation tests are started from this trim point with the initial conditions and corresponding trim controls.



Components

Main Rotor: Blade Element Momentum Theory
Tail Rotor: Blade Element Momentum Theory
Fuselage: 3-D Equivalent Flat Plate Drag
Wing: 2-D Quadratic Lift-Drag Aerodynamics
Horizontal Tail: 1-D Quadratic Lift Aerodynamics
Vertical Tail: 1-D Quadratic Lift Aerod
Engine: Ideal Engine
Landing Gear: No Landing Gear
Ground Effect: None
Stabilizer Bar: Stabilizer Bar
SAS: No SAS

Initial Conditions:

Collective Control (theta_0): 13,53681322 deg
phi: 0,00000000 deg
Longitudinal Cyclic Control (theta_1s): 0,00000000 deg
theta: 5,00000000 deg
Lateral Cyclic Control (theta_1c): 0,00000000 deg
Main Rotor Inflow: 26,75494580 ft/s
Pedal Control (theta_p): 12,00000000 deg
Tail Rotor Inflow: 33,69855554 ft/s

Trim Results

phi (deg): 0.03361529
Longitudinal Flapping Angle (rad): -0.00015628
theta (deg): 5.12361709
Lateral Flapping Angle (rad): -0.05007565
Collective (deg): 13.74117764
Main Rotor Inflow (ft/s): 27.44142311
Longitudinal Cyclic (deg): 0.00898405
Tail Rotor Inflow (ft/s): 37.81422814
Lateral Cyclic (deg): -2.87871931
Pedal (deg): 7.96298801

Constraints:

Forward Velocity (knots): 0.00000000
psi (deg): 0.00000000
Altitude (ft): 1000.00000000

Figure 5.6: Hover Trim Results of UH-1H at 1000 ft

CHAPTER 6

SIMULATION RESULTS

Simulation analyses are conducted at 1000 ft above sea level for hover trim condition. A test bench as depicted in figure 6.1 for simulations is prepared in MATLAB-Simulink environment according to the Intelligent Flight Controller design in figure 4.3. It is assumed that all measurements are available and therefore, the system is observable. The I-PID Controller and the Intelligent Flight Controller are tested in several challenging maneuvers. Root Mean Square Error(RMSE) analyses of these maneuvers are conducted to compare the flight controllers. In virtue of the RMSE analyses, the success of the Intelligent Flight Controller is seen obviously.

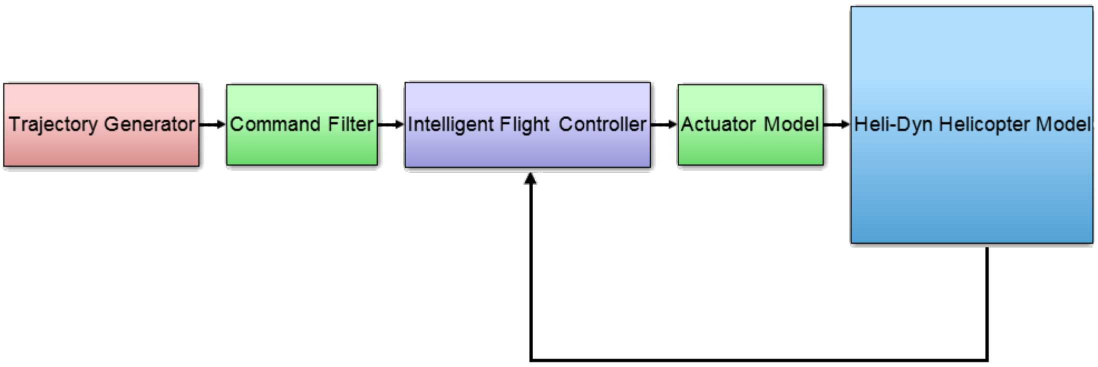


Figure 6.1: Block Diagram of the Test Bench for Simulation Tests

6.1 Root Mean Square Analysis

The root mean square is a special case of power mean. The power mean is a generalized mean which is in the form,

$$M_p(a_1, a_2, \ldots, a_n) = \left(\tfrac{1}{n} \sum_{k=1}^{n} a_k^p \right)^{1/p} \tag{6.1}$$

where $a_k \geq 0$ and p is a real number in the domain $[-\infty, +\infty]$.

And for $p = 2$ which is the M_2 power mean is the root mean square (RMS),

$$M_2(a_1, a_2, \ldots, a_n) = \sqrt{\frac{1}{n} \sum_{k=1}^{n} a_k^2} \tag{6.2}$$

The RMS can be extended to RMSE for error analysis using the equation 3.2.

$$RMSE = \sqrt{\frac{1}{n} \sum_{k=1}^{n} (x_d - x)^2} \tag{6.3}$$

The RMSE analysis is used for comparing difference between the desired and actual values of the position and attitude channels to find the best flight controller for each maneuver. The results of the RMSE analysis for each maneuver are illustrated in figure A.1.

6.1.1 RMSE Analysis for a Single State

For comparing two tests for a single state like roll angle, quotient of the RMSE values of the state is used as shown below,

$$S = 100 \times \left(1 - \frac{RMSE_{new}}{RMSE_{ref}} \right) \tag{6.4}$$

where S is the success ratio, $RMSE_{ref}$ is the RMSE value of the reference test and $RMSE_{new}$ is the RMSE value of the new test.

6.1.2 RMSE Analysis for Multiple States

For comparing two tests for multiple states like the combination of roll, pitch and yaw angles as attitude, ratio of Euclidean Norms of the RMSE values of these states are used as shown below,

$$S = 100 \times \left(1 - \sqrt{\frac{RMSE_{1_{new}}^2 + RMSE_{2_{new}}^2 + \cdots + RMSE_{n_{new}}^2}{RMSE_{1_{ref}}^2 + RMSE_{2_{ref}}^2 + \cdots + RMSE_{n_{ref}}^2}}\right) \tag{6.5}$$

where S is the success ratio, $RMSE_{n_{ref}}$ is the RMSE value of the reference test for n^{th} state in the comparison list and $RMSE_{n_{ref}}$ is the RMSE value of the new test for n^{th} state in the comparison list.

6.2 Maneuvers

As mentioned before, six challenging maneuvers are selected to demonstrate the adaptation abilities of the Intelligent Flight Controller with respect to the I-PID controller with constant gains. These maneuvers are pull up - push over, slalom, pull-up - push over - slalom, coning, pirouette and 3-D cone respectively. All these maneuvers are started from hover trim point at 1000 ft above sea level with the initial trim controls and simulated at 100 Hz using Euler integration method in MATLAB Simulink.

As described in Chapter 4, the Intelligent Flight Controller has ability to change the Inner Loop controller gains during flight. Both I-PID and the Intelligent Flight Controller are started from the same gains and performance of the controllers are analyzed.

Initial controller gains for the inner loop are selected as,

- Euler Angle Controller
  - ϕ Channel: K_P=10, K_I=0, K_D=5
  - θ Channel: K_P=10, K_I=0, K_D=5
  - ψ Channel: K_P=10, K_I=0, K_D=5
- Body Angular Velocity Controller
  - p Channel: K_P=20, K_I=1, K_D=10
  - q Channel: K_P=30, K_I=1, K_D=15
  - r Channel: K_P=20, K_I=1, K_D=10

The intelligent controller needs a closed loop stable system due to the limitations of Least Squares Regression as stated in Chapter 2 and 3. Otherwise, intelligent controller may still adapt and control the helicopter if Least Squares Regression succeeds to model the error dynamics of the unstable system. Therefore, these initial PID gains are chosen after testing them to have a stable system at least 30 seconds for each maneuver with I-PID controller. After determination of the initial PID gains, six complex maneuvers are tested for both controllers. The first three maneuvers and the last one are conducted for 300 seconds. However, simulations for coning and pirouette maneuvers are limited to 120 seconds for both controllers because of the difficulty of these maneuvers.

6.2.1 Pull Up - Push Over Maneuver

The longitudinal channel of the flight controller is tested with the pull up - push over maneuver for 300 seconds. A sinusoidal 100 ft peak-peak altitude change within 40 seconds time period and a constant 50 knots north velocity are expected in this maneuver [16].

Starting from the initial positions $X_N = 0, X_E = 0, X_D = -1000$ and $\psi = 0$, the commanded position equations for pull up-push over maneuver,

$$X_N = \int V_N \, dt \tag{6.6}$$

$$X_E = 0 \tag{6.7}$$

$$X_D = -1000 - A \int sin(\frac{2\,\pi}{T}\, X_N)\, dt \tag{6.8}$$

$$\psi = 0 \tag{6.9}$$

where A=50 ft, T=40 seconds and V_N=50 knots.

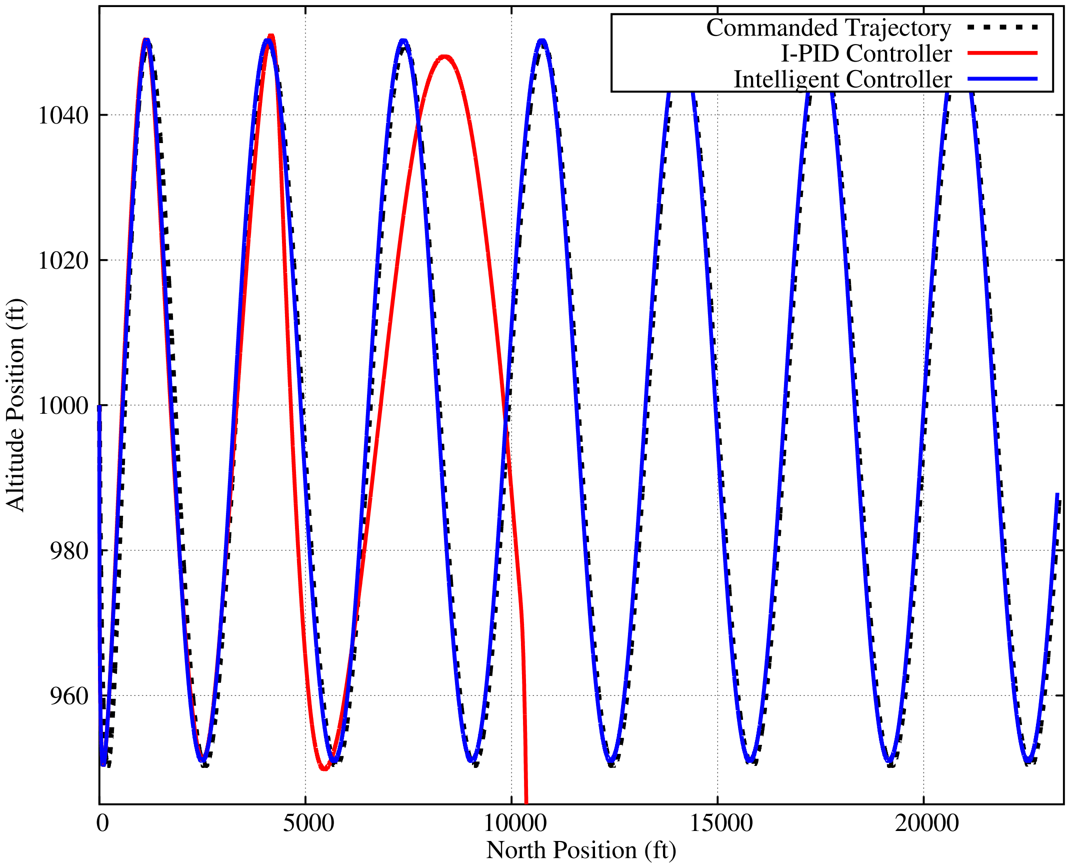


Figure 6.2: Trajectory of the Helicopter on X-Z Plane

The I-PID controller can stand for about 100 seconds without any adaptation, but the chosen initial gains are not suitable for this maneuver and the helicopter crashes before the simulation ends. However, the Intelligent Flight Controller is adapted itself quickly and controls the helicopter until the end of the simulation. Divergent path followed by I-PID controller and also the trajectory followed by the Intelligent Flight Controller are shown in in figure 6.2.

Adaptation of PID gains of the Euler Angle Controller of the Intelligent Flight Controller is shown in figure 6.3. Derivative gains are nearly constant and integral gains are applied periodically after learning is completed. Differently, there are oscillations in proportional gains, but the amplitude of the oscillations are insignificant.

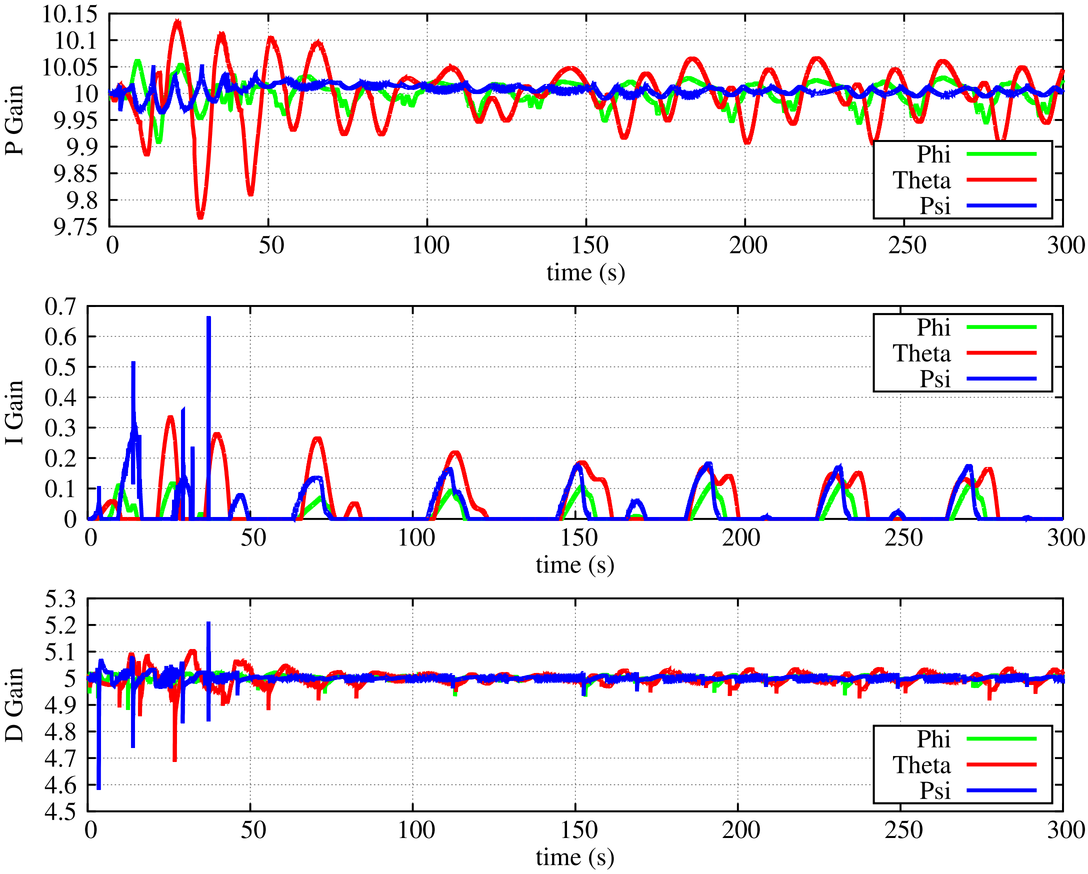


Figure 6.3: Adaptation History of the Euler Angle Controller Gains

The amplitude of oscillations in proportional channel for theta control is larger than other channels. This difference is expected since the maneuver challenges the longitudinal stability of the controller directly by periodic climbs and dives.

Initial short-time peaks for integral and derivative gains of theta and psi channels are related with the adaptation process and transition from hover to forward flight. As helicopter reaches to 50 knots forward speed, adaptation gets easier and these peaks fade out.

The stability of the PID gains of the Euler Angle Controller of the Intelligent Flight Controller provides a correct trajectory tracking as seen in figure 6.2.

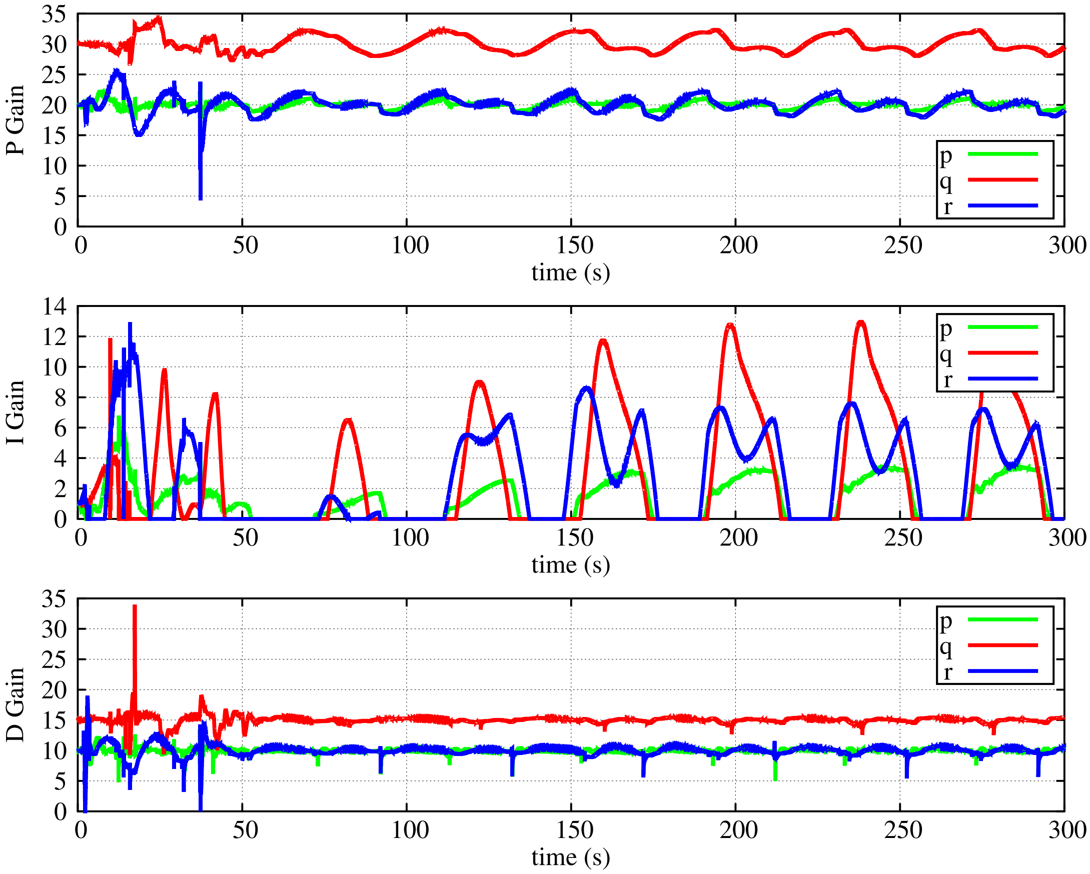


Figure 6.4: Adaptation History of the Body Angular Velocity Controller Gains

Learning regime of PID gains of the Body Angular Velocity Controller of the Intelligent Flight Controller is illustrated in figure 6.4. Proportional and derivative gains have continuous oscillations with negligible amplitudes. As integral gains of the Euler Angle Controller, integral gains of the Body Angular Velocity Controller are applied periodically after learning is completed. Although integral gains seem unstable at the beginning, the peak amplitude of the integral gains are not changed after adaptation is completed.

The stability of the PID gains of the Body Angular Velocity Controller provides stable PID gains for Euler Angle Controller as shown in figure 6.3.

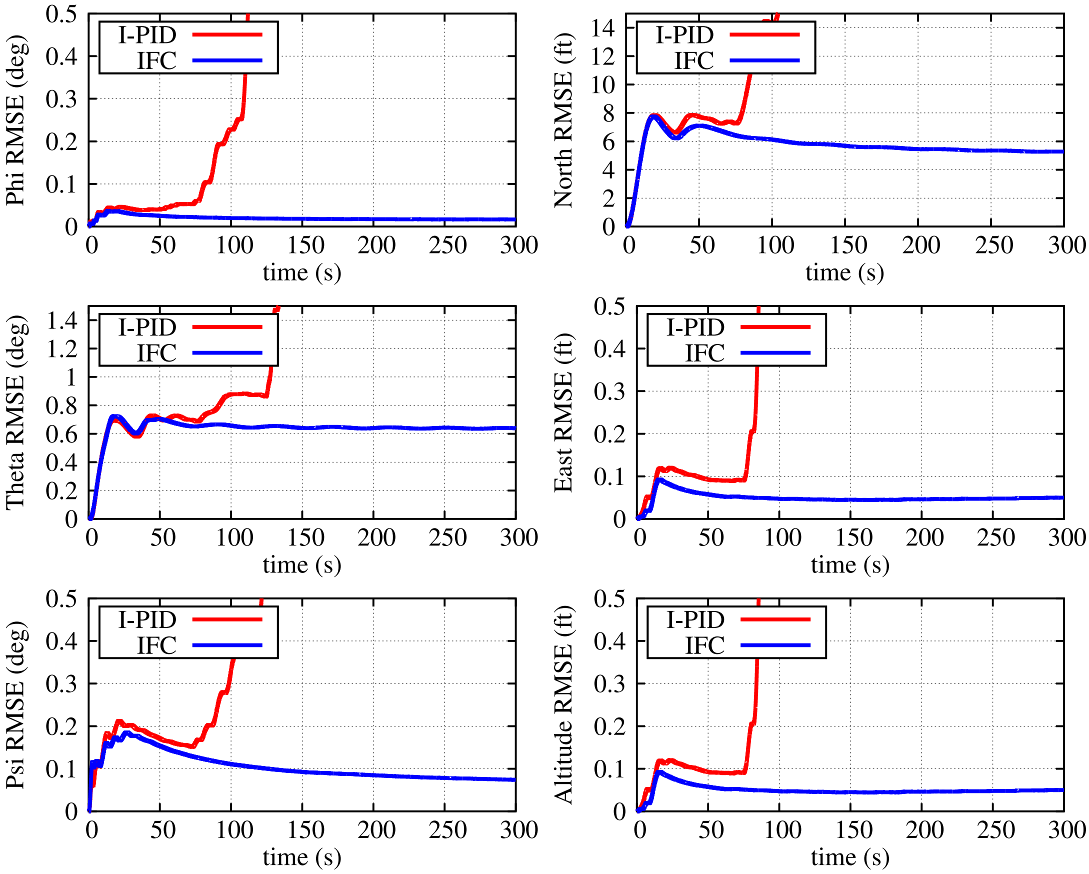


Figure 6.5: RMSE Analyses of the Positions of the Helicopter in Pull Up - Push Over Maneuver

When comparing the root mean square errors of six positions for both controllers from figure 6.5, effectiveness of the Intelligent Flight Controller is seen obviously for all positions. In addition to the root mean square error analysis, it is seen from in figure 6.6 that the Intelligent Flight Controller completes the given mission with periodic but stable control inputs.

In consequence, the reference trajectory for pull up - push over maneuver is followed successfully by the Intelligent Flight Controller. However, I-PID controller with fixed gains lose the control and helicopter hit the ground within 150 seconds. Therefore, simulation for I-PID controller is stopped at 150^{th} second as it can be seen on figure 6.6.

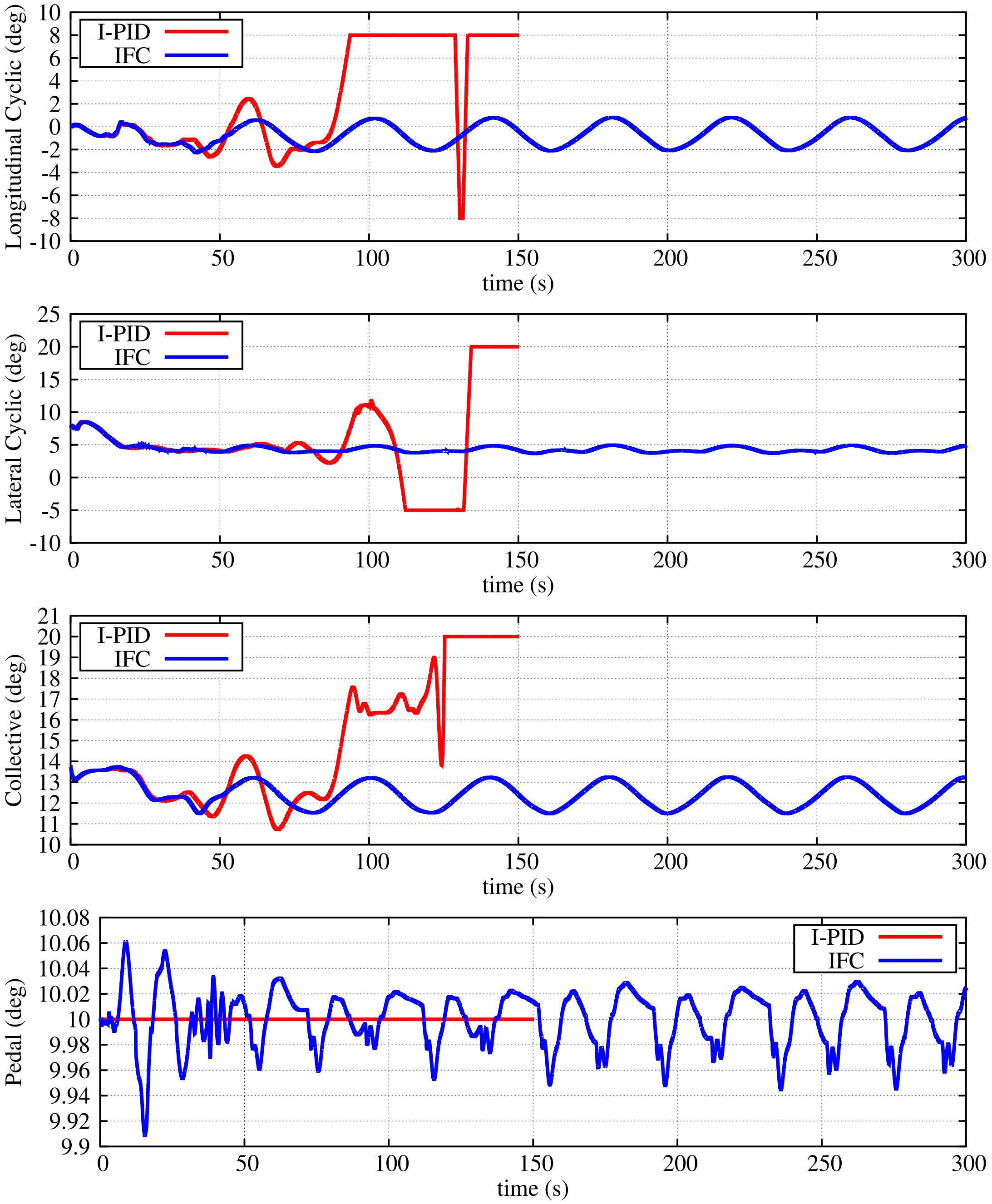


Figure 6.6: Control Inputs Generated by I-PID and IFC in Pull Up - Push Over Maneuver

6.2.2 Slalom Maneuver

The lateral channel of the flight controller is tested in the slalom maneuver for 300 seconds. A sinusoidal 100 ft peak-peak east position change within 100 seconds time period and a constant 10 knots north velocity are expected in this maneuver [16].

Starting from the initial positions $X_N = 0, X_E = 0, X_D = -1000$ and $\psi = 0$, the commanded position equations for slalom maneuver,

$$X_N = \int V_N \, dt \tag{6.10}$$

$$X_E = A \int sin(\frac{2\,\pi}{T}\, X_N)\, dt \tag{6.11}$$

$$X_D = -1000 \tag{6.12}$$

$$\psi = 0 \tag{6.13}$$

where A=50 ft, T=100 seconds and V_N=10 knots.

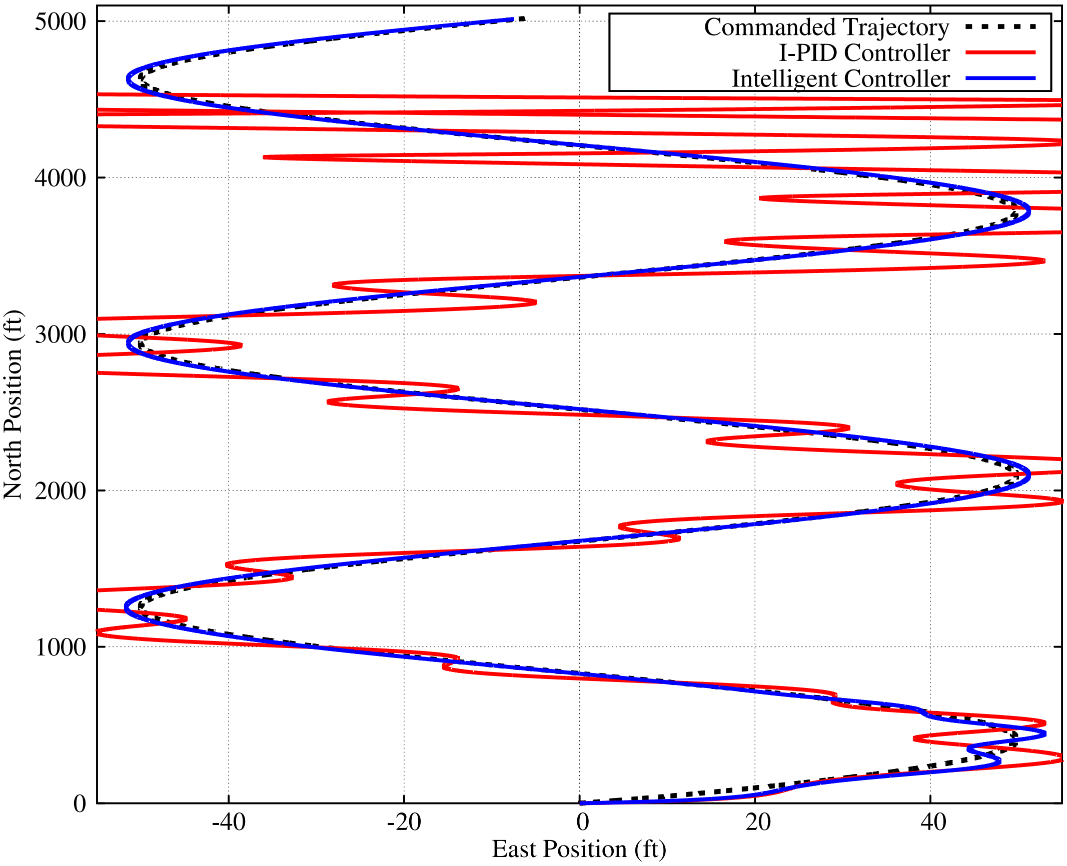


Figure 6.7: Trajectory of the Helicopter on X-Y Plane

The I-PID controller can follow the desired trajectory with oscillations for about 250 seconds without any adaptation. However, these oscillations continuously increases due to the incorrect initial gains for this maneuver and the helicopter crashes before end of the simulation. The process which leads to crash is seen obviously from the controller inputs in figure 6.11.

Unlike I-PID controller, the Intelligent Flight Controller completes the learning within 50 seconds and completes this test successfully. Divergence of the path followed by I-PID controller and also the effort of the Intelligent Flight Controller are shown in figure 6.7.

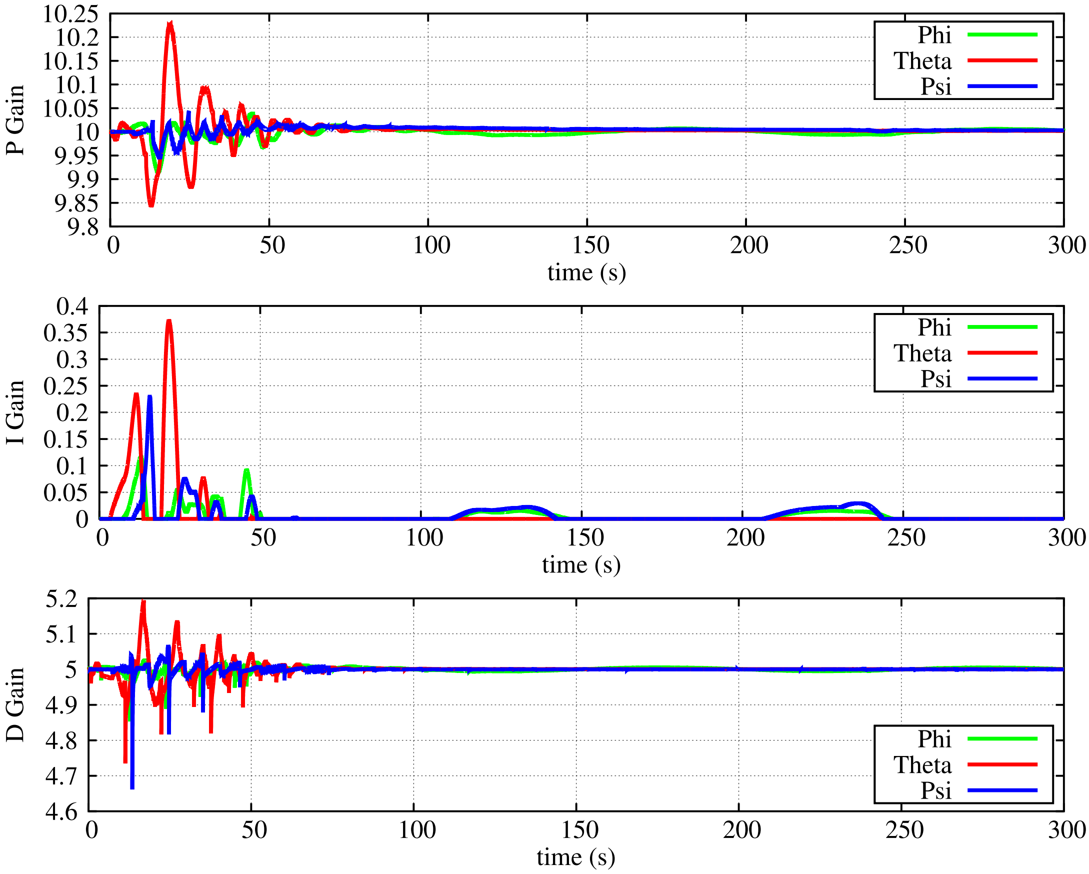


Figure 6.8: Adaptation History of the Euler Angle Controller Gains

Adaptation of PID gains of the Euler Angle Controller of the Intelligent Flight Controller is shown in figure 6.8. Proportional and derivative gains are nearly constant and integral gains are applied periodically with a very low frequency and insignificant amplitude after learning is completed.

Initial short-time peaks with small amplitudes for integral and derivative gains of theta and psi channels are related with the adaptation process and transition from hover to forward flight. As helicopter reaches to 10 knots forward speed, these peaks fade out due to the completion of adaptation.

The stability of the PID gains of the Euler Angle Controller of the Intelligent Flight Controller provides an accurate trajectory as shown in figure 6.7.

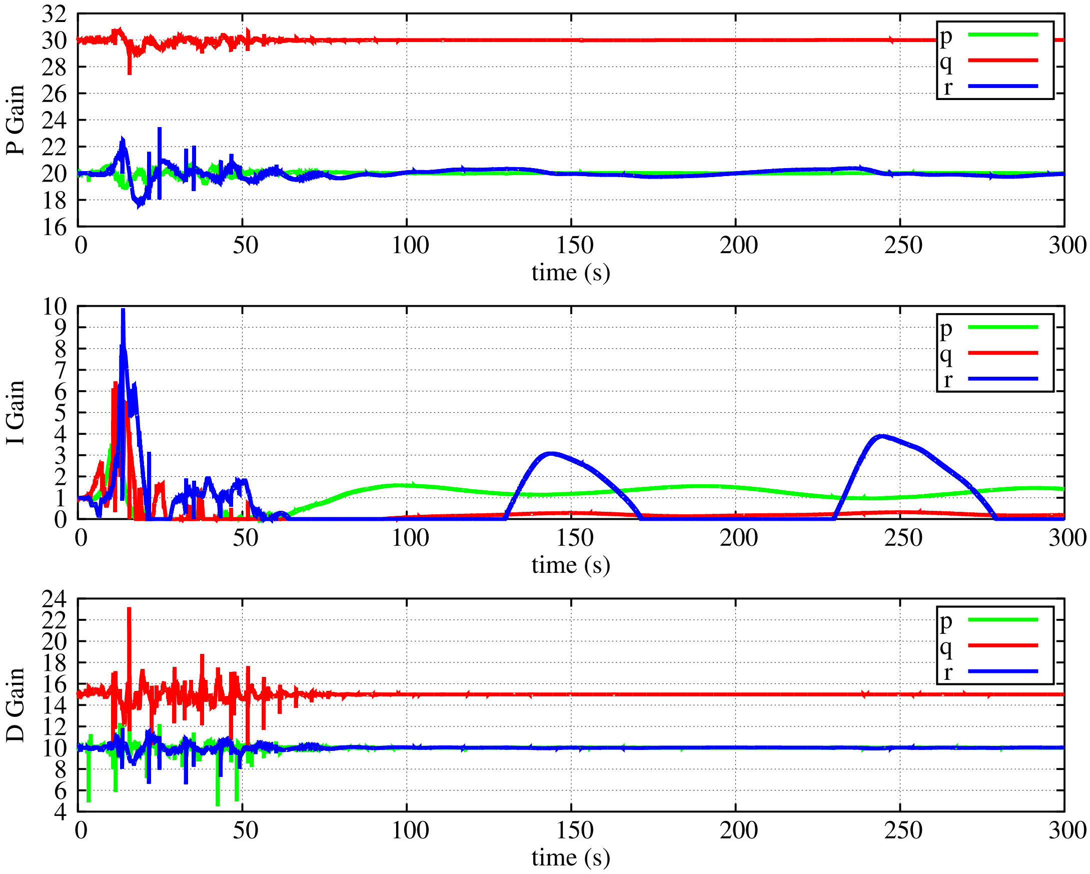


Figure 6.9: Adaptation History of the Body Angular Velocity Controller Gains

PID gains of the Body Angular Velocity Controller of the Intelligent Flight Controller are also stable and depicted in figure 6.9. After learning process is completed, proportional and derivative gains are nearly constant until the end of the simulation. Integral gains of p and q channels of the Body Angular Velocity Controller are also constant. However, integral gain of r channel is applied periodically with a very low frequency after learning is completed. Because of very low frequency, integral gain of r channel does not cause an unexpected disturbance for 300 seconds.

The stability of the PID gains of the Body Angular Velocity Controller provides stable PID gains for Euler Angle Controller as shown in figure 6.8.

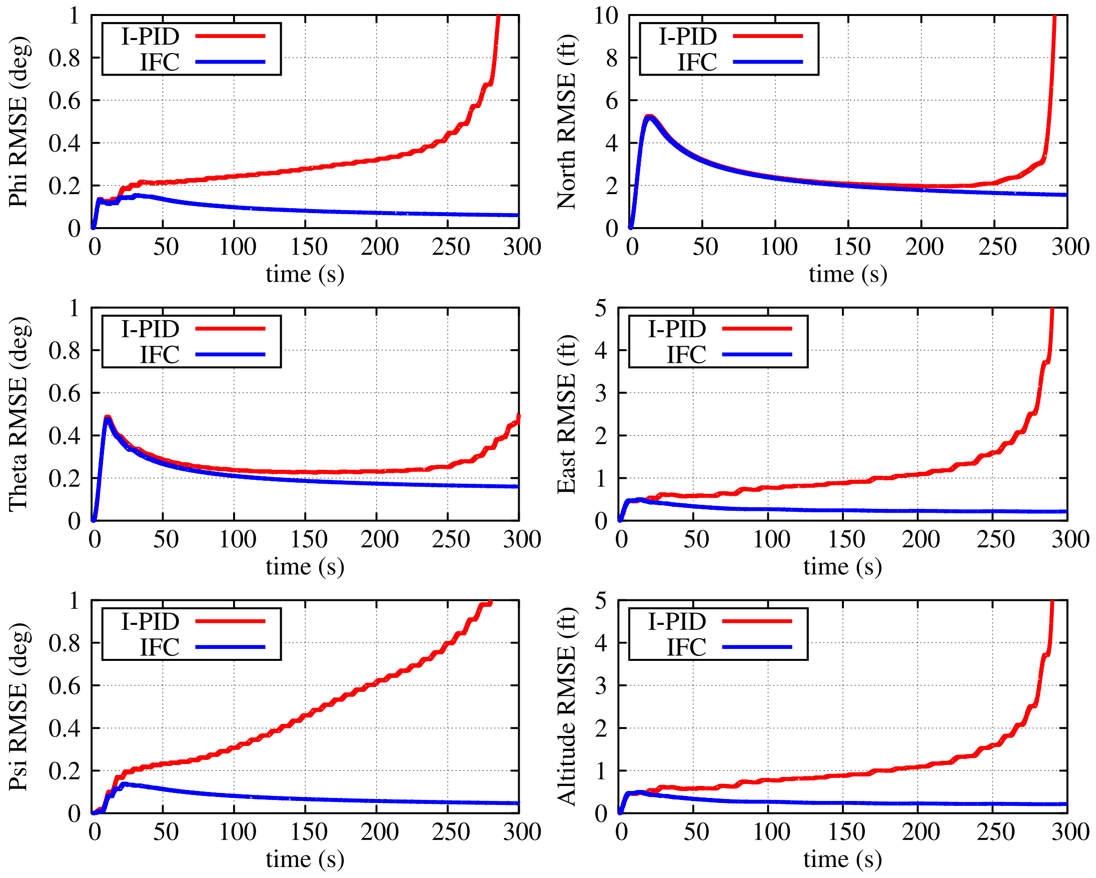


Figure 6.10: RMSE Analyses of the Positions of the Helicopter in Slalom Maneuver

The root mean square errors of six positions for both controllers as shown in figure 6.10 indicate the effectiveness of the Intelligent Flight Controller for all positions. It is also seen in figure 6.10 that I-PID controller with constant gains lose control in all positions after 250 seconds. In addition to the root mean square error analysis, it is seen from in figure 6.11 that the Intelligent Flight Controller completes the given mission with nearly constant control inputs. However, control deflections of the I-PID controller with fixed gains oscillate increasingly.

Consequently, the reference desired trajectory for slalom maneuver is accomplished by the Intelligent Flight Controller without any bias. However, I-PID controller with fixed gains cannot control the helicopter and helicopter crashes after about 250 seconds as it can be seen on figure 6.7.

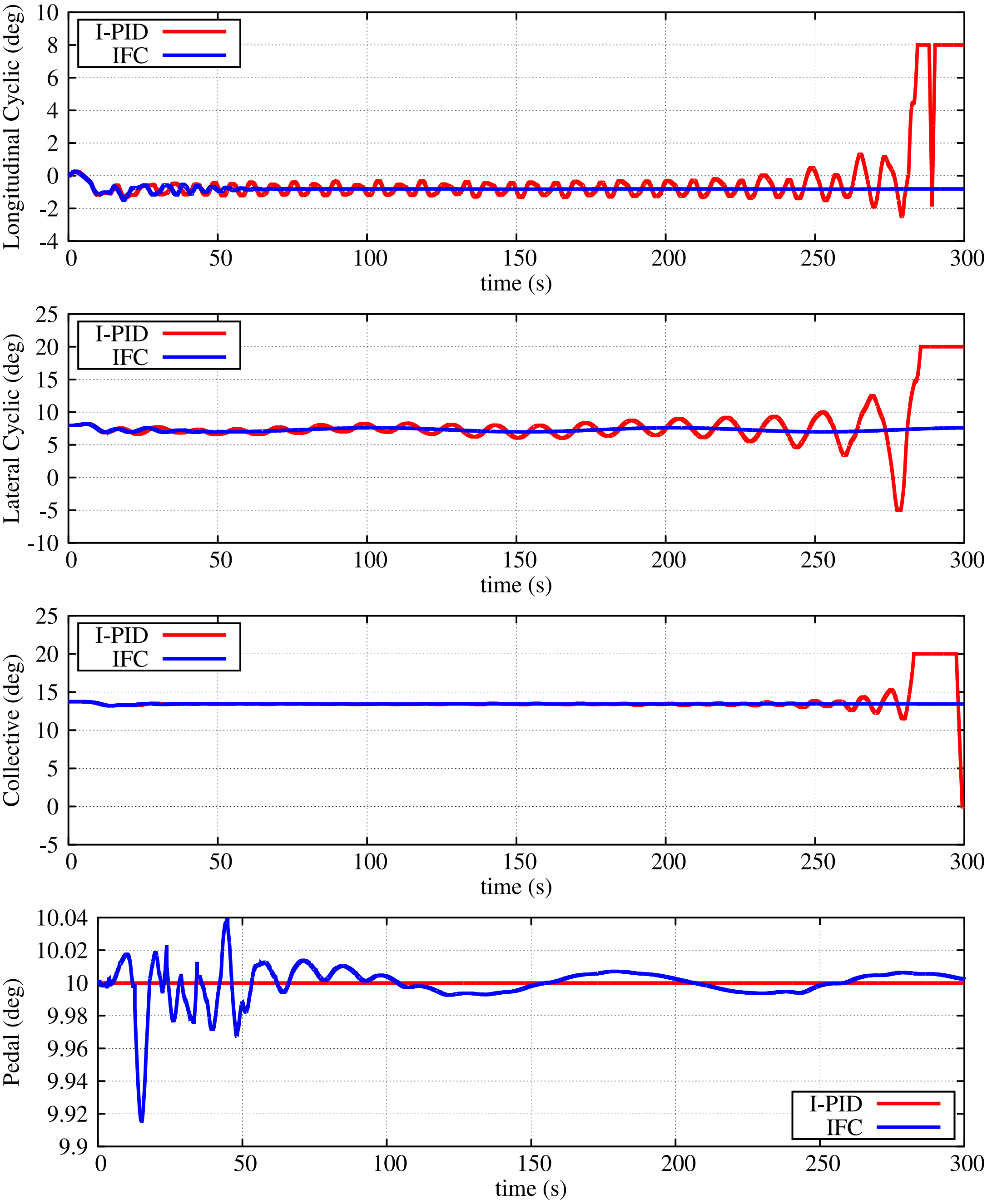


Figure 6.11: Control Inputs Generated by I-PID and IFC in Slalom Maneuver

6.2.3 Pull Up - Push Over - Slalom Maneuver

This maneuver is a combination of pull up - push over and slalom maneuvers. Both lateral and longitudinal channels of the flight controller are tested in pull up - push over - slalom maneuver for 300 seconds. A sinusoidal 100 ft peak-peak east position change within 100 seconds time period and a sinusoidal 100 ft peak-peak altitude change within 50 seconds time period are expected in this maneuver. The forward velocity target is 50 knots constant during flight.

Starting from the initial positions $X_N = 0, X_E = 0, X_D = -1000$ and $\psi = 0$, the commanded position equations for slalom maneuver,

$$X_N = \int V_N \, dt \tag{6.14}$$

$$X_E = A \int sin(\frac{2\,\pi}{T_1} X_N) \, dt \tag{6.15}$$

$$X_D = -1000 - A \int sin(\frac{2\,\pi}{T_2} X_N) \, dt \tag{6.16}$$

$$\psi = 0 \tag{6.17}$$

where A=50 ft, T_1=100 seconds, T_2=50 seconds and V_N=50 knots.

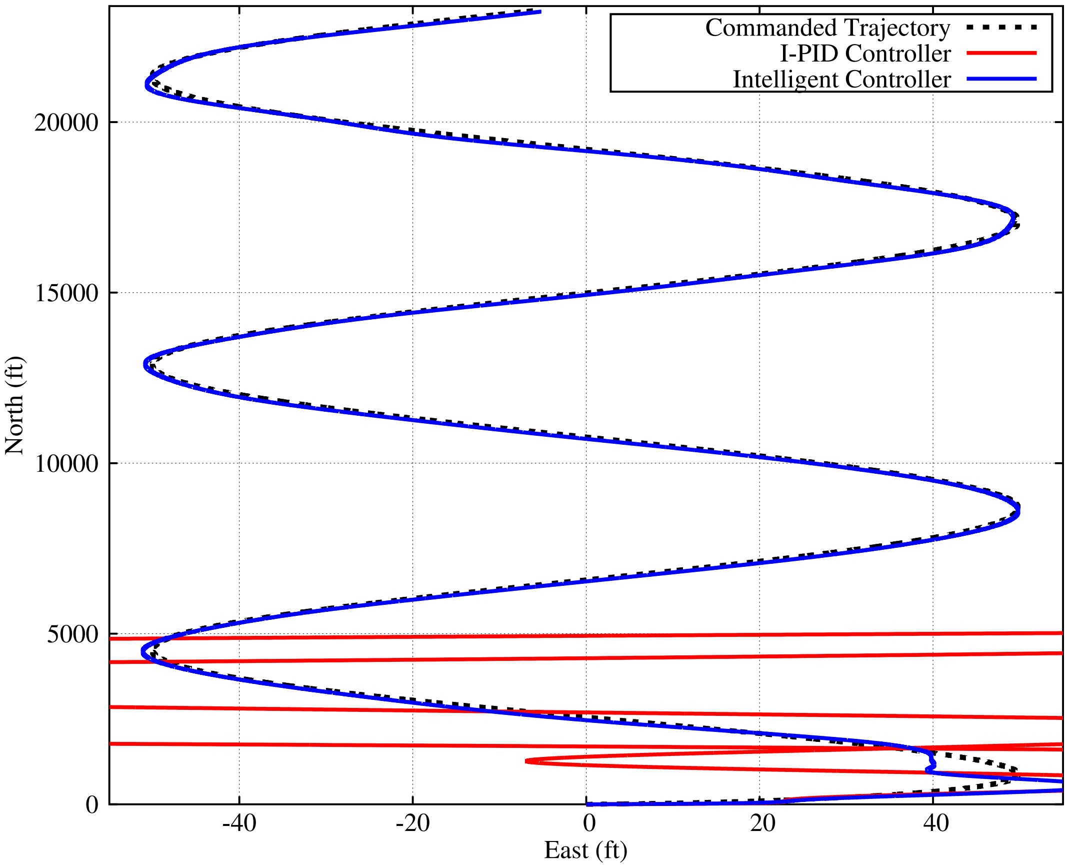


Figure 6.12: Trajectory of the Helicopter on X-Y Plane

At the end of the simulation, the I-PID controller cannot follow the reference trajectory and the helicopter crashes after about 120 seconds. Although I-PID controller can follow the North-Altitude trajectory for about 60 seconds as seen in figure 6.13, due to the unstable characteristics of the helicopter in lateral channel as shown in figure 6.12, initial gains are not enough to control in lateral channel and attitude of the helicopter crashes with I-PID controller.

The process of crash is seen obviously from growing oscillations in the longitudinal cyclic, lateral cyclic and collective controls as shown in figure 6.17. These oscillations are started after about 30 seconds for longitudinal cyclic, lateral cyclic and collective controls and lead to crash of the helicopter after about 120 seconds.

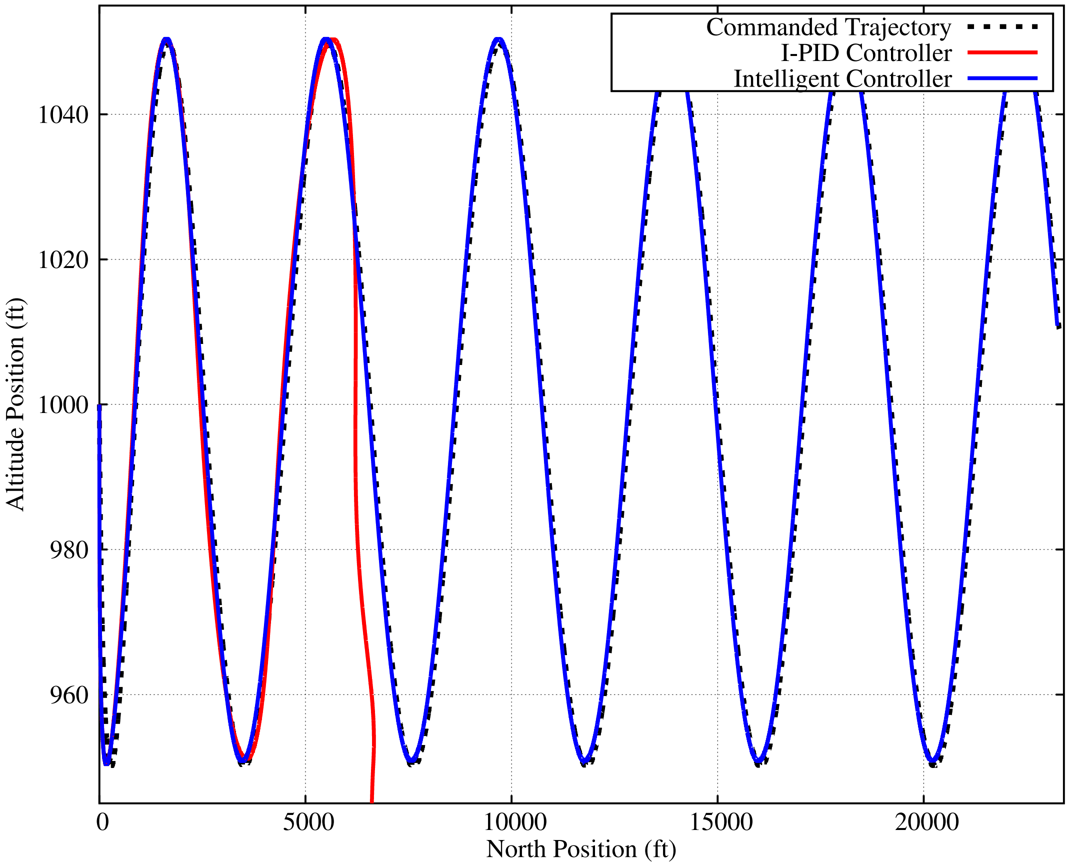


Figure 6.13: Trajectory of the Helicopter on X-Z Plane

However, the Intelligent Flight Controller is adapted itself within the first 20 seconds and maintains the control of the helicopter for 300 seconds as shown in figures 6.12 and 6.13. Thus, the helicopter follows correct trajectories for both North-Altitude and North-East channels with the Intelligent Flight Controller.

Adaptation of PID gains of the Euler Angle Controller of the Intelligent Flight Controller is shown in figure 6.14. Compulsion of the Intelligent Flight Controller is understood from the oscillations of PID gains. Derivative gains are more stable. Integral gains are applied periodically even after learning is completed. Although the oscillations in proportional gains shows an unstable regime, the amplitude of the oscillations are small in magnitude and they cannot force the Intelligent Flight Controller to lead a crash during the simulation period.

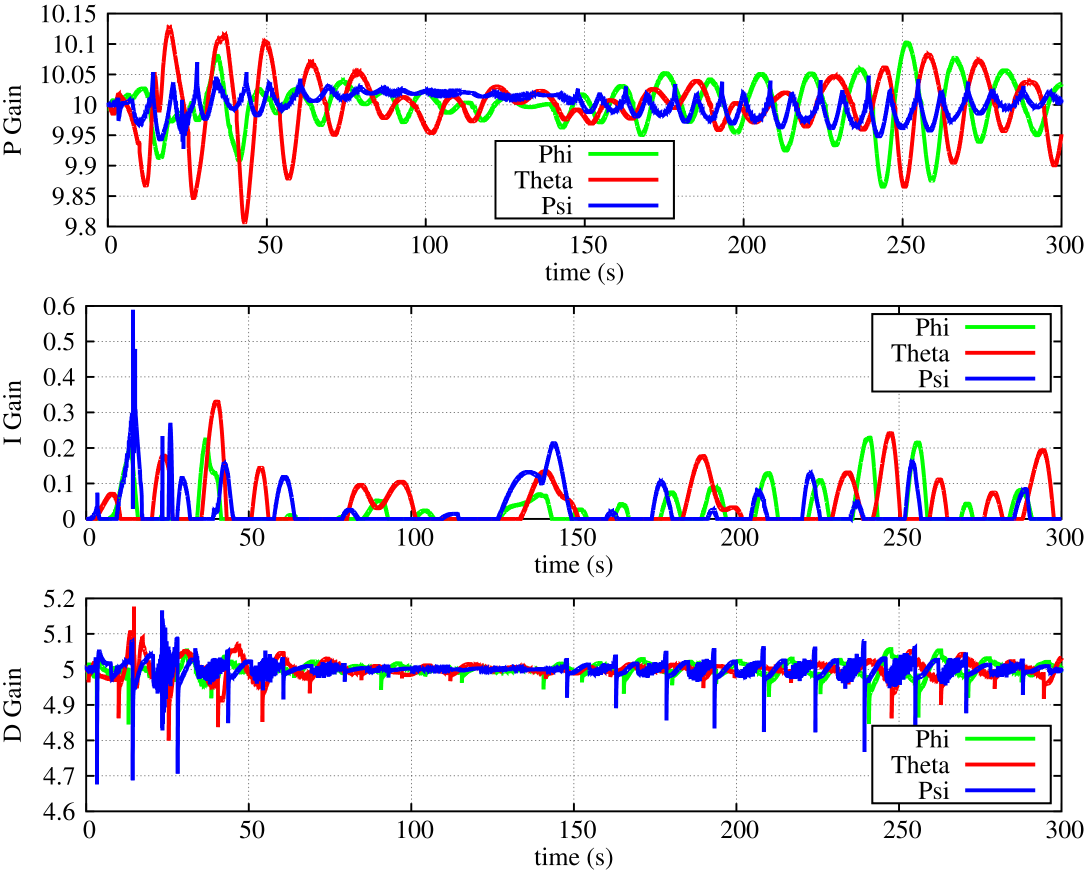


Figure 6.14: Adaptation History of the Euler Angle Controller Gains

The amplitude of oscillations in proportional channel for theta control is larger than other channels. This difference is expected since the maneuver challenges the longitudinal stability of the controller directly by pull up - push over maneuver. In addition, minor oscillations are existed in phi and psi channels due to the difficulty of the slalom maneuver. These oscillations of the PID gains of the Euler Angle Controller cannot deter the Intelligent Flight Controller to follow correct trajectories for both North-Altitude and North-East channels as seen in figures 6.12 and 6.13.

Initial short-time peaks with small amplitudes for integral and derivative gains of theta and psi channels also occur in this maneuver. As helicopter reaches to 50 knots forward speed, adaptation gets easier and these peaks are fade out.

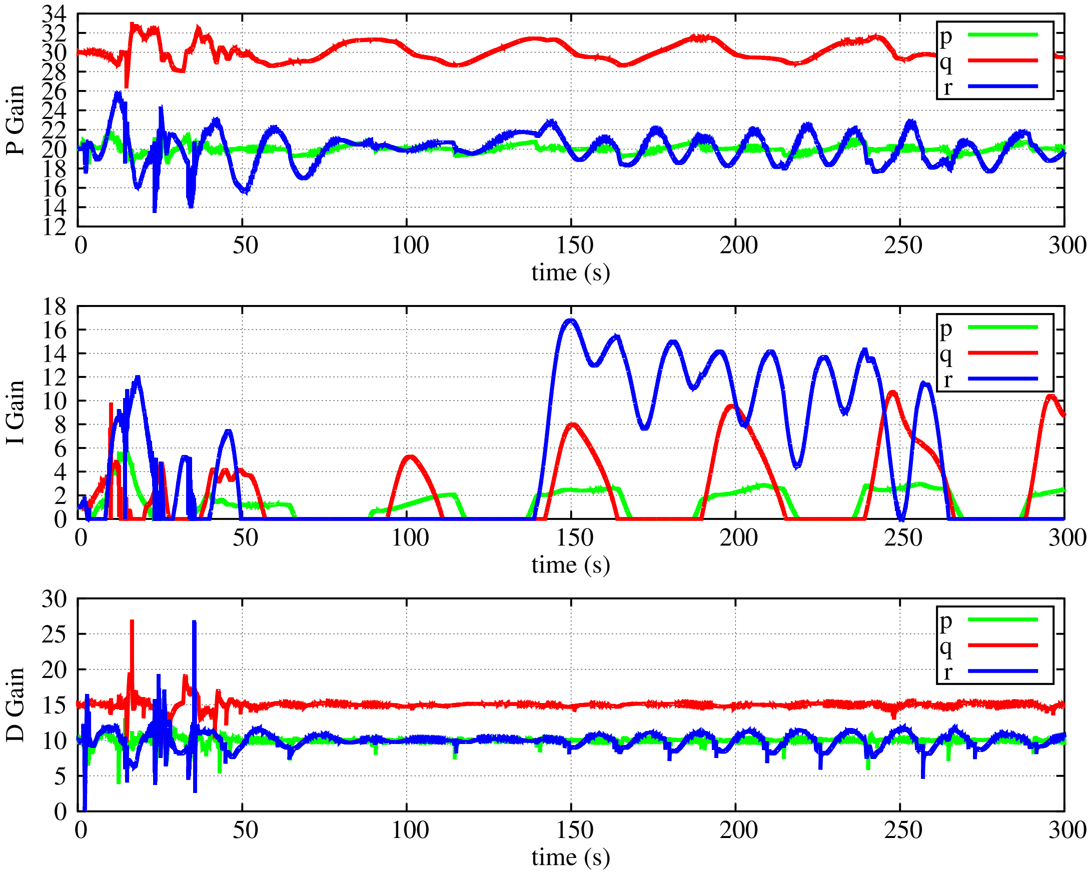


Figure 6.15: Adaptation History of the Body Angular Velocity Controller Gains

Learning process of PID gains of the Body Angular Velocity Controller of the Intelligent Flight Controller is shown in figure 6.15. Proportional and derivative gains behave sinusoidally with small magnitudes. As integral gains of the Euler Angle Controller, integral gains of the Body Angular Velocity Controller are applied periodically after learning is completed except integral gain of r controller. Between about 140^{th} and 265^{th} seconds, integral gain of r controller suddenly increases and then fades out.

These gain oscillations show the complexity of this maneuver. In spite of this complexity, the Intelligent Flight Controller can control the helicopter until the end of the simulation and follows the desired trajectory.

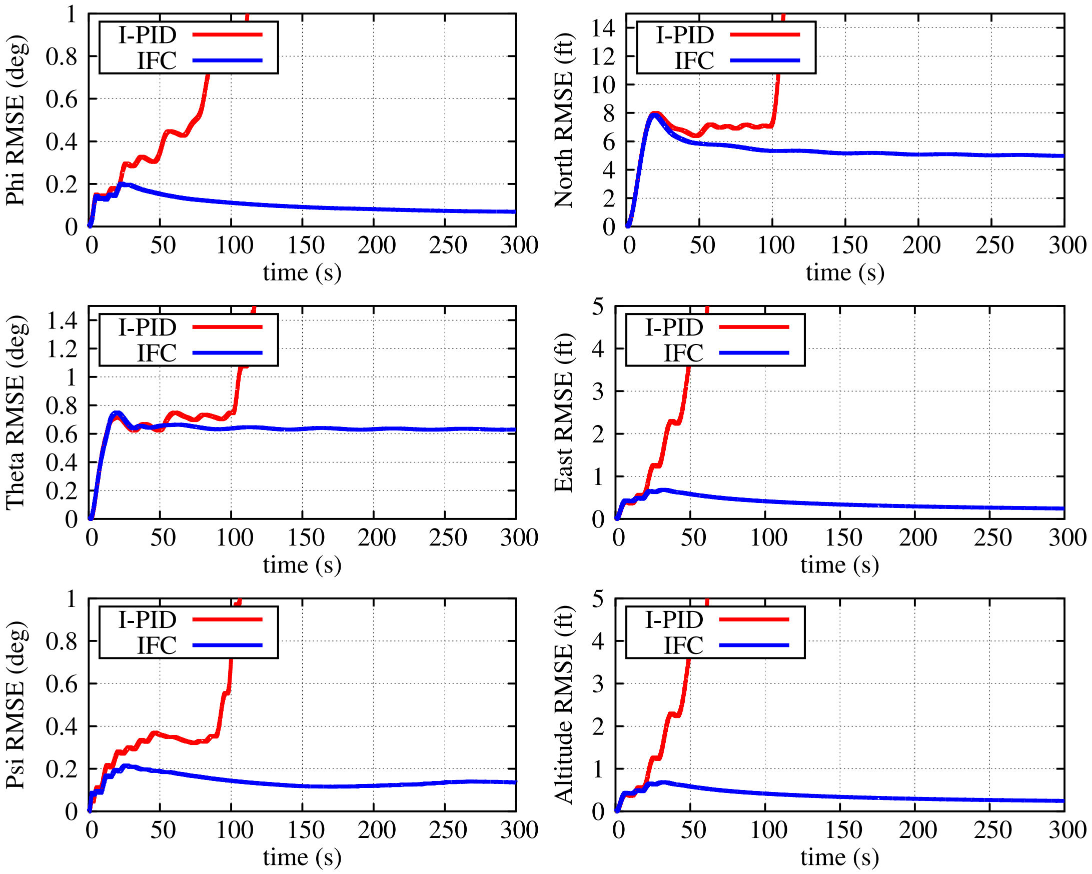


Figure 6.16: RMSE Analyses of the Positions of the Helicopter in Pull Up - Push Over - Slalom Maneuver

Root mean square errors of six positions for both controllers from figure 6.16 show the effectiveness of the Intelligent Flight Controller for all positions. According to RMSE analysis, I-PID controller lose the control of the helicopter nearly at the beginning. But, the Intelligent Flight Controller maintains the control until the end of the simulation. In addition to the root mean square error analysis, it is shown in figure 6.17 that the Intelligent Flight Controller accomplishes the given mission with periodic but stable control inputs.

To sum up, the reference trajectory for the pull up - push over - slalom maneuver is followed successfully by the Intelligent Flight Controller. However, I-PID controller with constant gains lose the control and thus the helicopter hit the ground within 100 seconds.

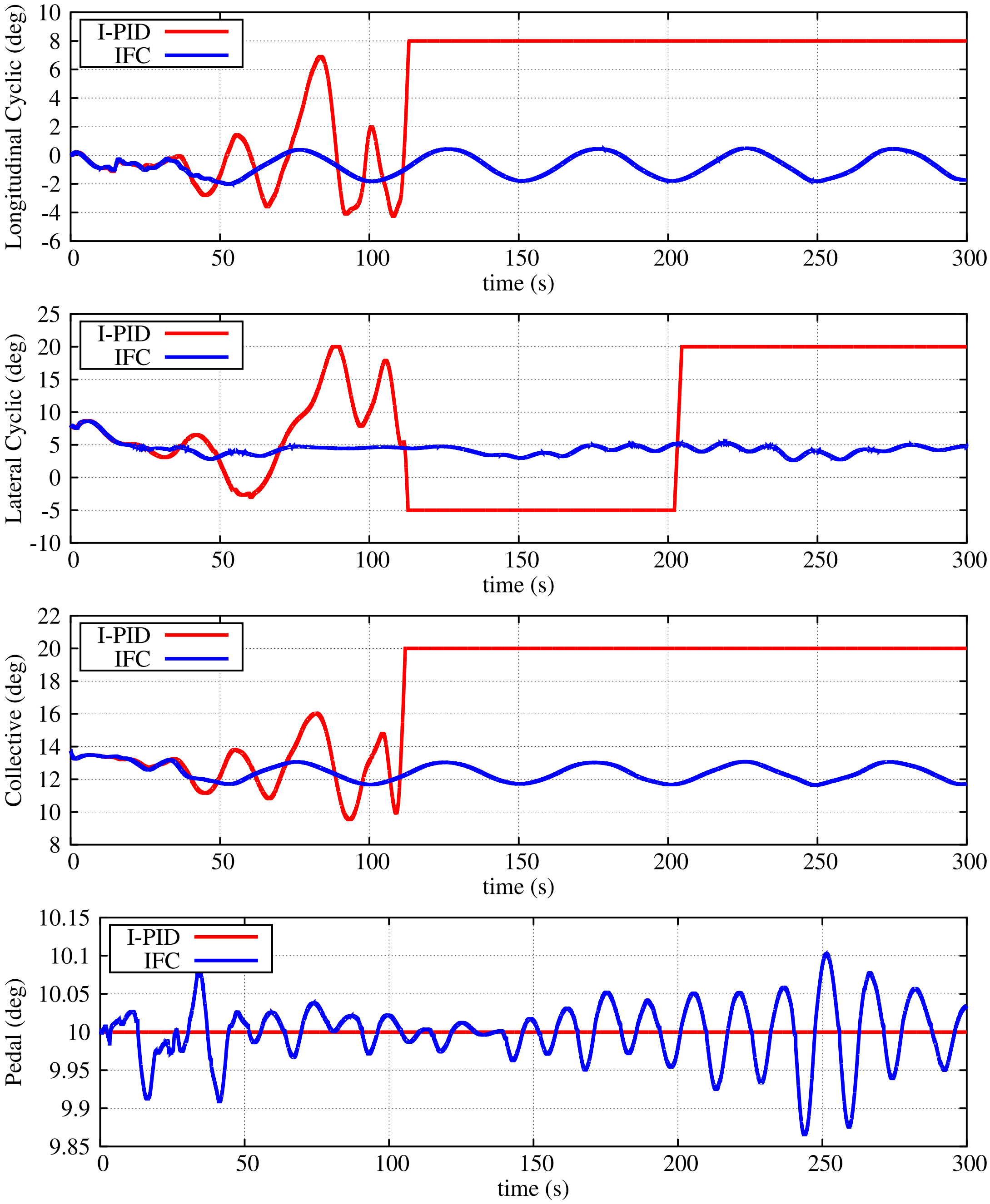


Figure 6.17: Control Inputs Generated by I-PID and IFC in Pull Up - Push Over - Slalom Maneuver

6.2.4 Coning Maneuver

Both lateral and longitudinal channels of the flight controller are tested in the coning maneuver in the case of a constant heading change. A continuously rounding path with a decreasing radius is expected along north and east positions like drawing a cone in the case of a constant heading change. Total velocity is 8 knots and heading rate is 4.58 deg/s during flight. [16].

Starting from the initial positions $X_N = 0, X_E = 0, X_D = -1000$ and $\psi = 0$, the commanded position equations for coning maneuver,

$$X_N = \int V_{Total}\, sin\left(\int \psi\, dt\right)\, dt \tag{6.18}$$

$$X_E = \int V_{Total}\, cos\left(\int \psi\, dt\right)\, dt \tag{6.19}$$

$$X_D = -1000 \tag{6.20}$$

$$\dot{\psi} = \frac{V_{Total}}{R} \tag{6.21}$$

where R=100 ft and V_{Total}=8 knots.

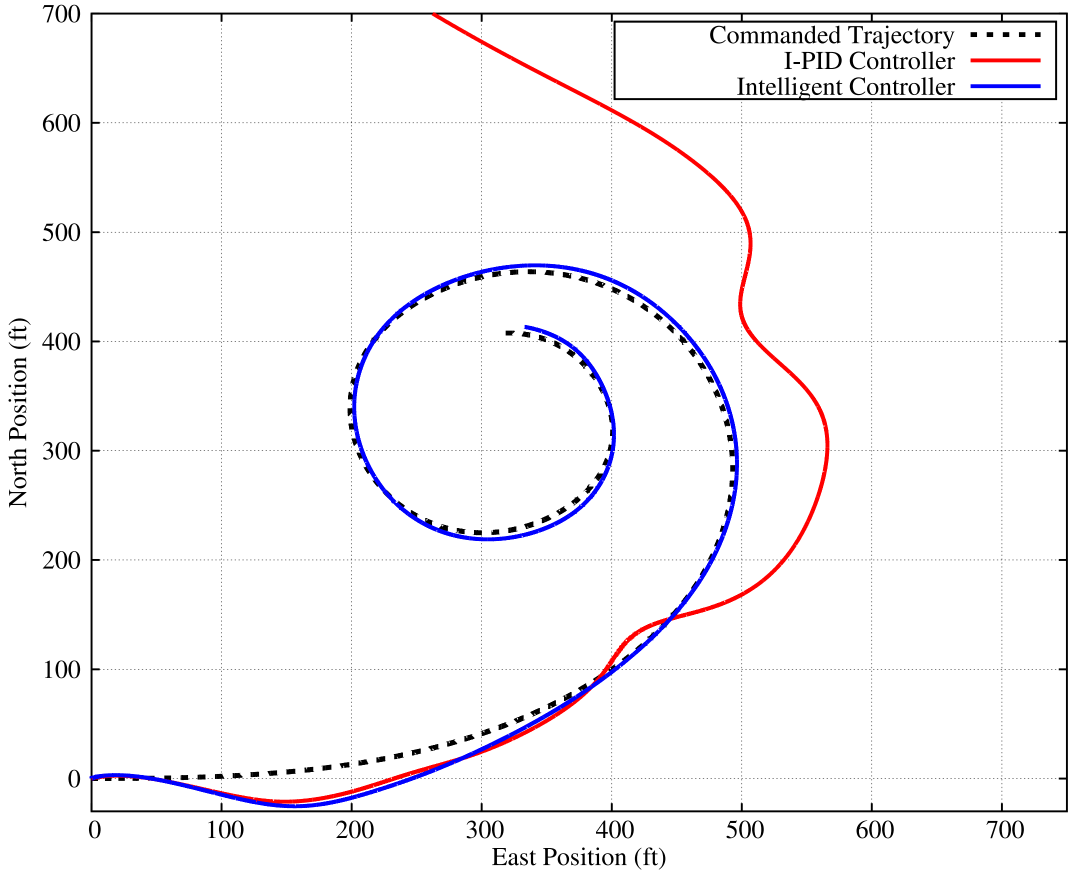


Figure 6.18: Trajectory of the Helicopter on X-Y Plane

This maneuver is hard to accomplish due to the constant heading rate and continuously decreasing radius of the trajectory. Therefore, the simulation is limited to 120 seconds for both controllers. At the beginning, helicopter accelerates to reach 8 knots to start to draw a cone. The Intelligent Flight Controller reaches to 8 knots in total velocity and follows the trajectory as expected. The Intelligent Flight Controller completes two loops in this maneuver.

However, the I-PID controller cannot start to draw a cone due to the incorrect gains for this maneuver. In spite of following the same trajectory with the Intelligent Flight Controller for about 10 seconds, the I-PID controller leaves the path within 60 seconds as seen in figure 6.18.

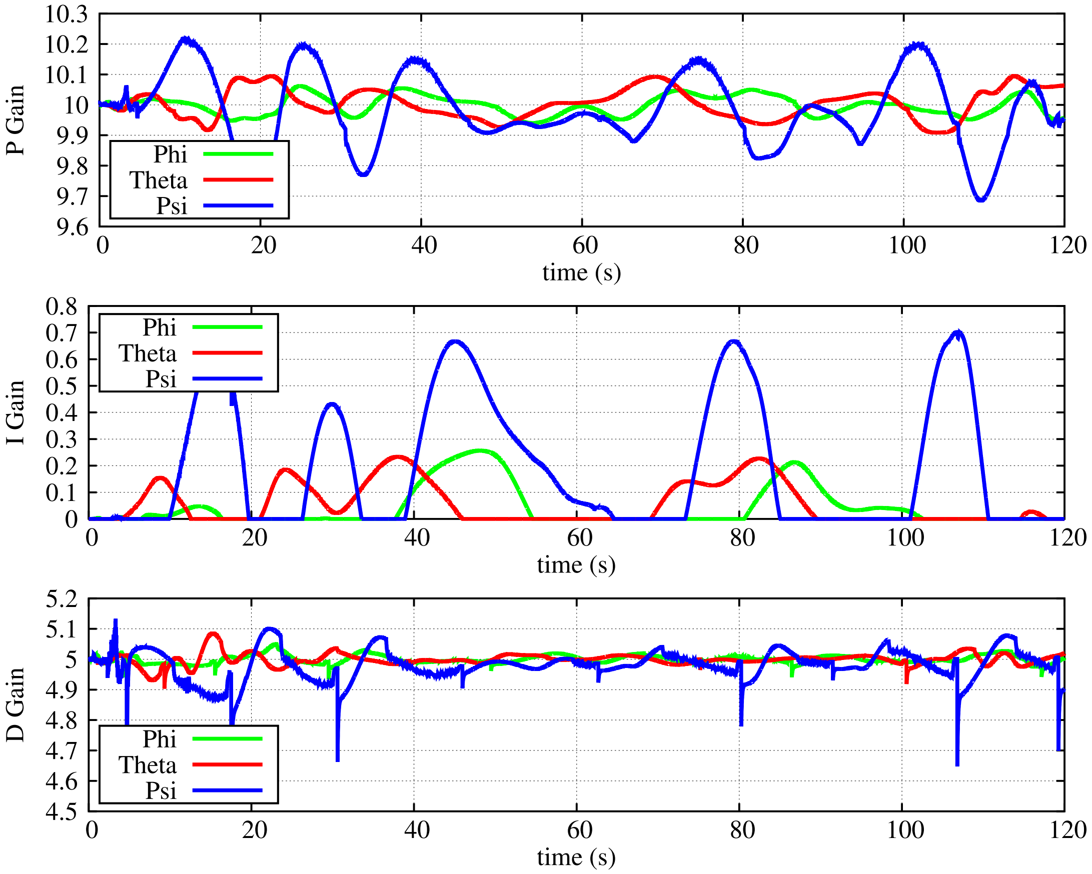


Figure 6.19: Adaptation History of the Euler Angle Controller Gains

Adaptation process of PID gains of the Euler Angle Controller of the Intelligent Flight Controller is shown in figure 6.19. Because of the complexity of the coning maneuver, PID gains of the Euler Angle Controller are not as smooth as in the previous maneuvers.

There are sinusoidal oscillations for all gains, but the amplitude of these oscillations are small. In spite of the oscillations, the Intelligent Flight Controller can follow the desired trajectory in this maneuver. The most oscillatory gains of the Euler Angle Controller are belonged to psi controller because of the constant heading rate and continuously decreasing radius of the trajectory of the coning maneuver.

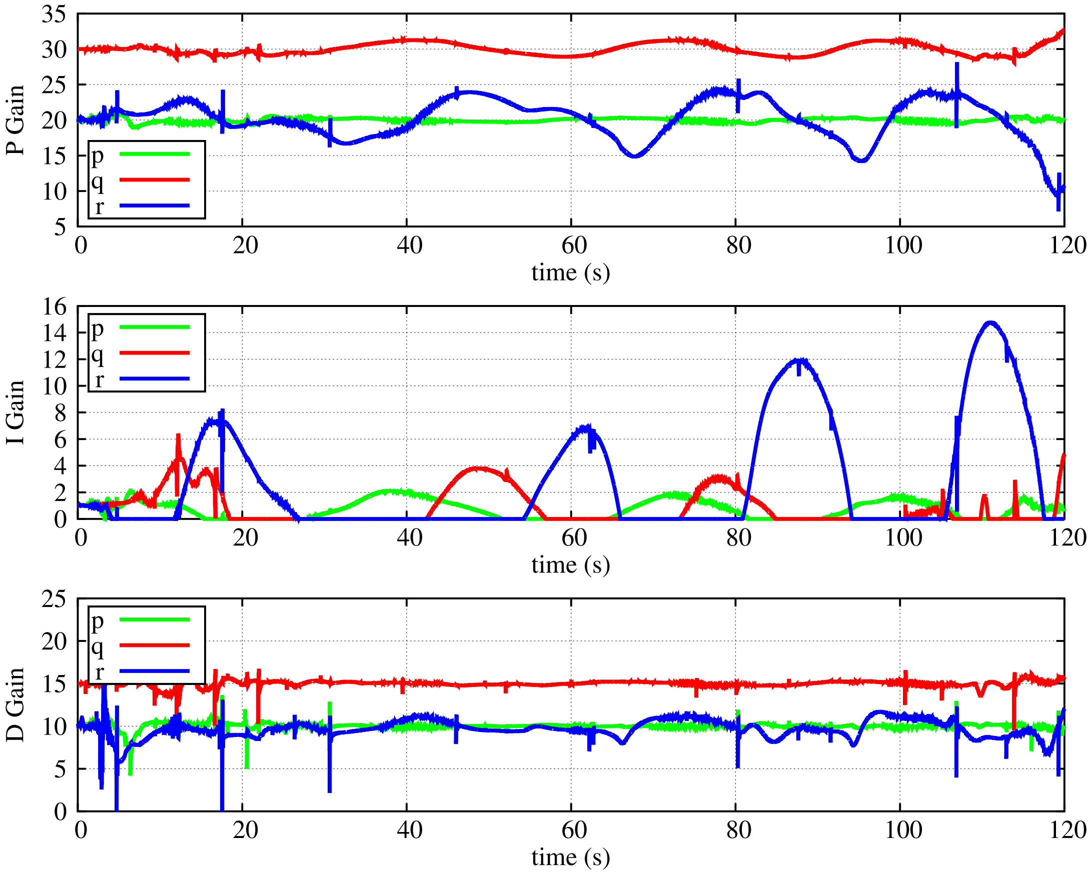


Figure 6.20: Adaptation History of the Body Angular Velocity Controller Gains

Learning process of PID gains of the Body Angular Velocity Controller of the Intelligent Flight Controller is illustrated in figure 6.20. Proportional and derivative gains have continuous oscillations with small amplitudes. As integral gains of the Euler Angle Controller, integral gains of the Body Angular Velocity Controller are applied periodically after learning is completed.

As in the Euler Angle Controller, the most oscillatory gains of the Body Angular Velocity Controller are belonged to psi controller because of the constant heading rate and continuously decreasing radius of the trajectory of the coning maneuver.

In spite of these oscillations, the Intelligent Flight Controller can control the helicopter until the end of the simulation and follows the desired trajectory.

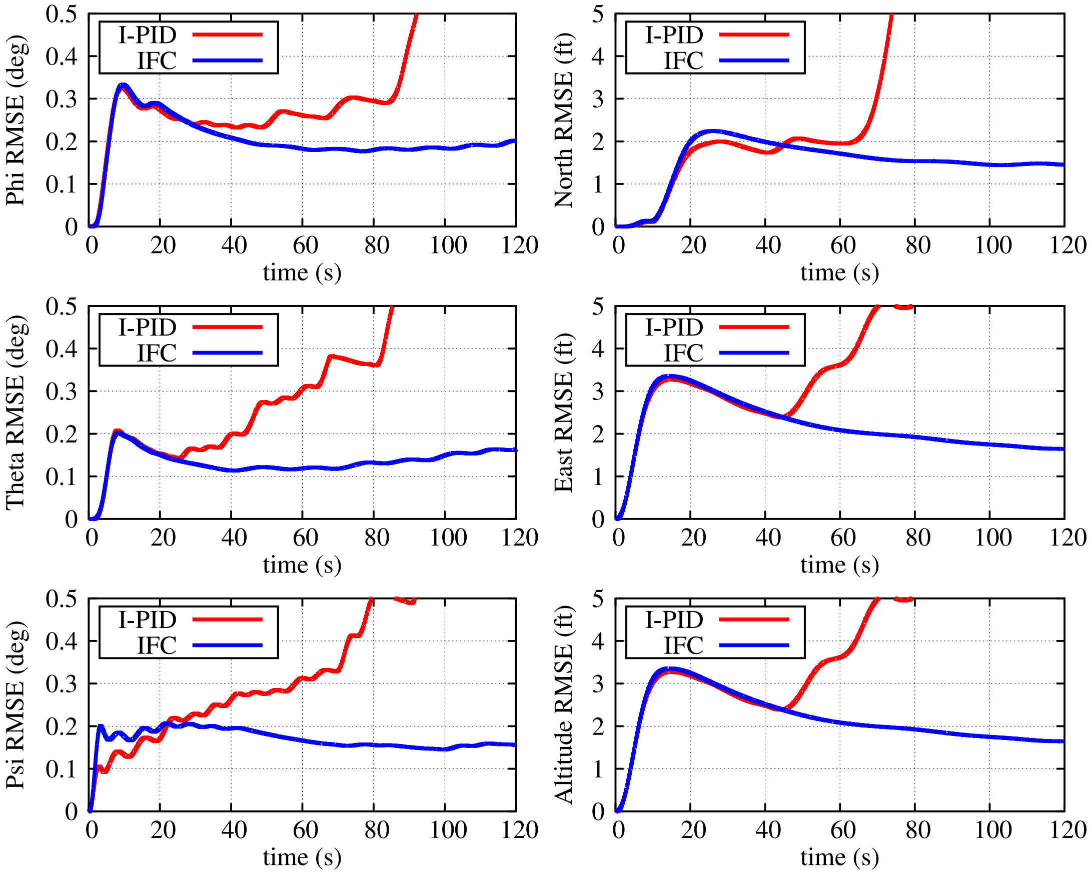


Figure 6.21: RMSE Analyses of the Positions of the Helicopter in Coning Maneuver

Root mean square error analyses of six positions for both controllers from figure 6.21 show the effectiveness of the Intelligent Flight Controller for all positions. According to RMSE analysis, the trajectory followed by I-PID controller diverges from the reference nearly at the beginning. But, the Intelligent Flight Controller maintains the tacking the reference trajectory until the end of the simulation. In addition to the root mean square error analysis, it is shown in figure 6.22 that the Intelligent Flight Controller accomplishes the given mission with nearly constant control inputs.

In consequence, the reference trajectory for coning maneuver is followed successfully by the Intelligent Flight Controller. However, I-PID controller with fixed gains cannot start to draw a cone and follows a different path rather than the reference trajectory as seen on figure 6.18.

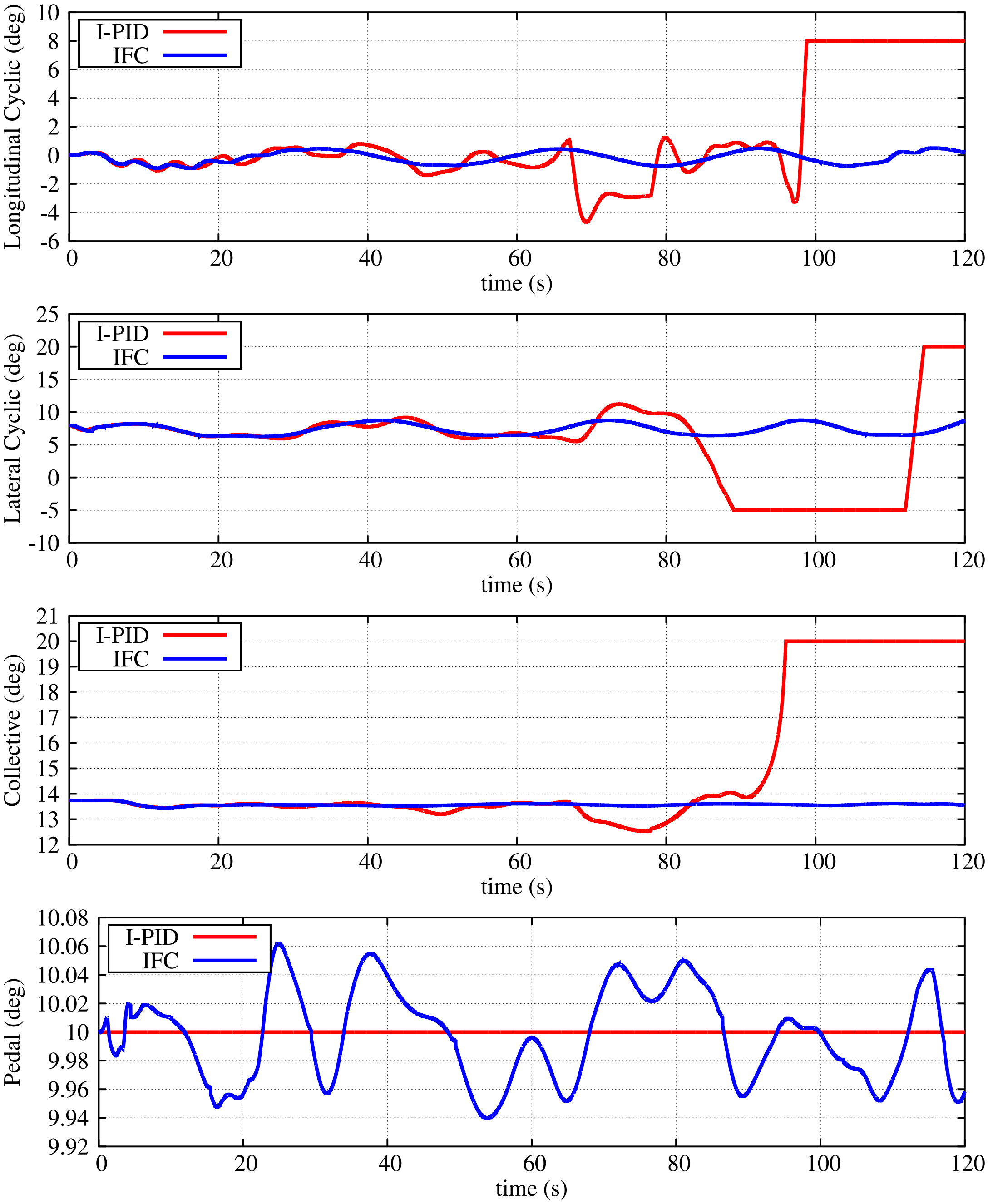


Figure 6.22: Control Inputs Generated by I-PID and IFC in Coning Maneuver

6.2.5 Pirouette Maneuver

Both lateral and longitudinal channels of the flight controller are tested in the pirouette maneuver in the case of a constant heading change. A continuously rounding path with a decreasing radius is expected along north and east positions. But, in this case heading rate is faster and constant north velocity is slower than coning maneuver. Hence, this condition causes to turn around a single point which is the definition of pirouette. 6.88 deg/s yaw rate in reverse direction and a constant 6 knots total velocity are expected in this maneuver [16].

Starting from the initial positions $X_N = 0, X_E = 0, X_D = -1000$ and $\psi = 0$, the commanded position equations for pirouette maneuver,

$$X_N = \int V_{Total}\, sin\left(\int \psi\, dt\right)\, dt \tag{6.22}$$

$$X_E = \int V_{Total}\, cos\left(\int \psi\, dt\right)\, dt \tag{6.23}$$

$$X_D = -1000 \tag{6.24}$$

$$\dot{\psi} = -2\,\frac{V_{Total}}{R} \tag{6.25}$$

where R=100 ft and V_{Total}=6 knots.

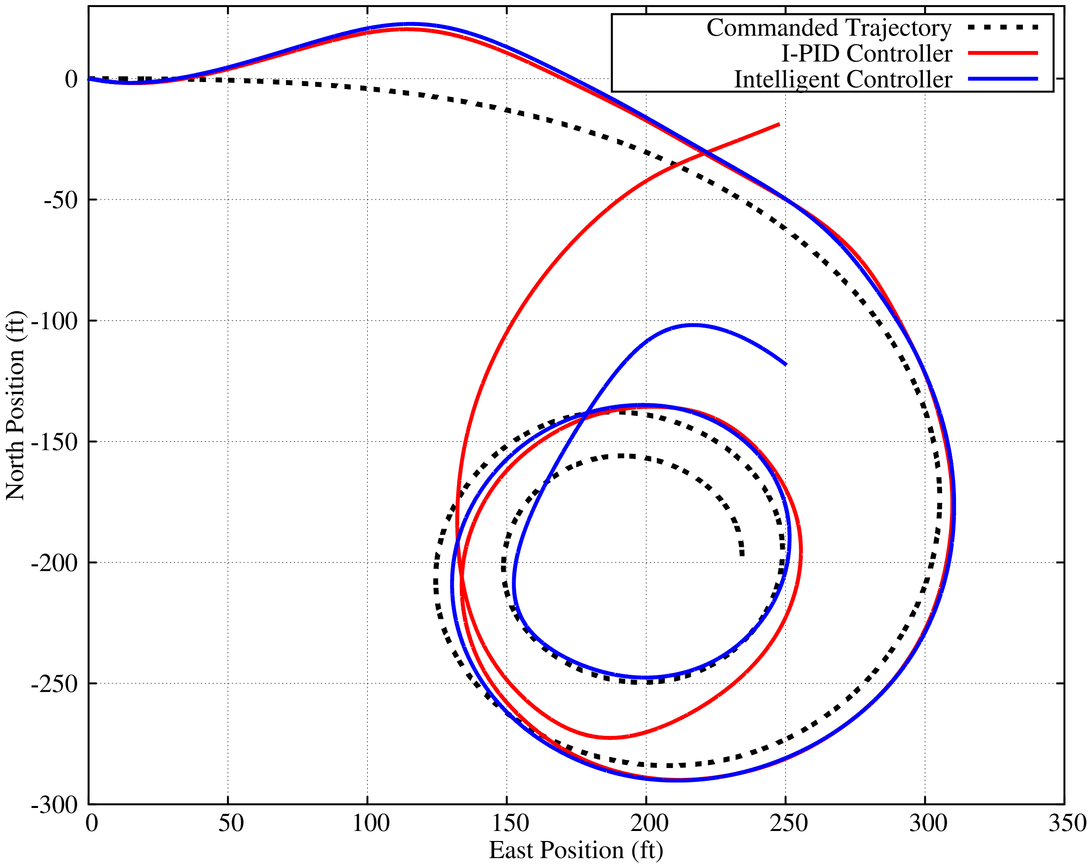


Figure 6.23: Trajectory of the Helicopter on X-Y Plane

This maneuver is hard to accomplish due to the constant heading rate and continuously decreasing radius of the trajectory like coning maneuver. Therefore, the simulation is limited to 120 seconds for both controllers. At the beginning, helicopter accelerates to reach 6 knots from hover to start to draw a cone with a faster yawing rate than coning maneuver.

Both I-PID and the Intelligent Flight Controller are achieved to follow the trajectory for pirouette maneuver. However, the Intelligent Flight Controller is able to perform a better pirouette and starts to try a second loop before leaving the trajectory unlike the I-PID controller as seen in figure 6.23.

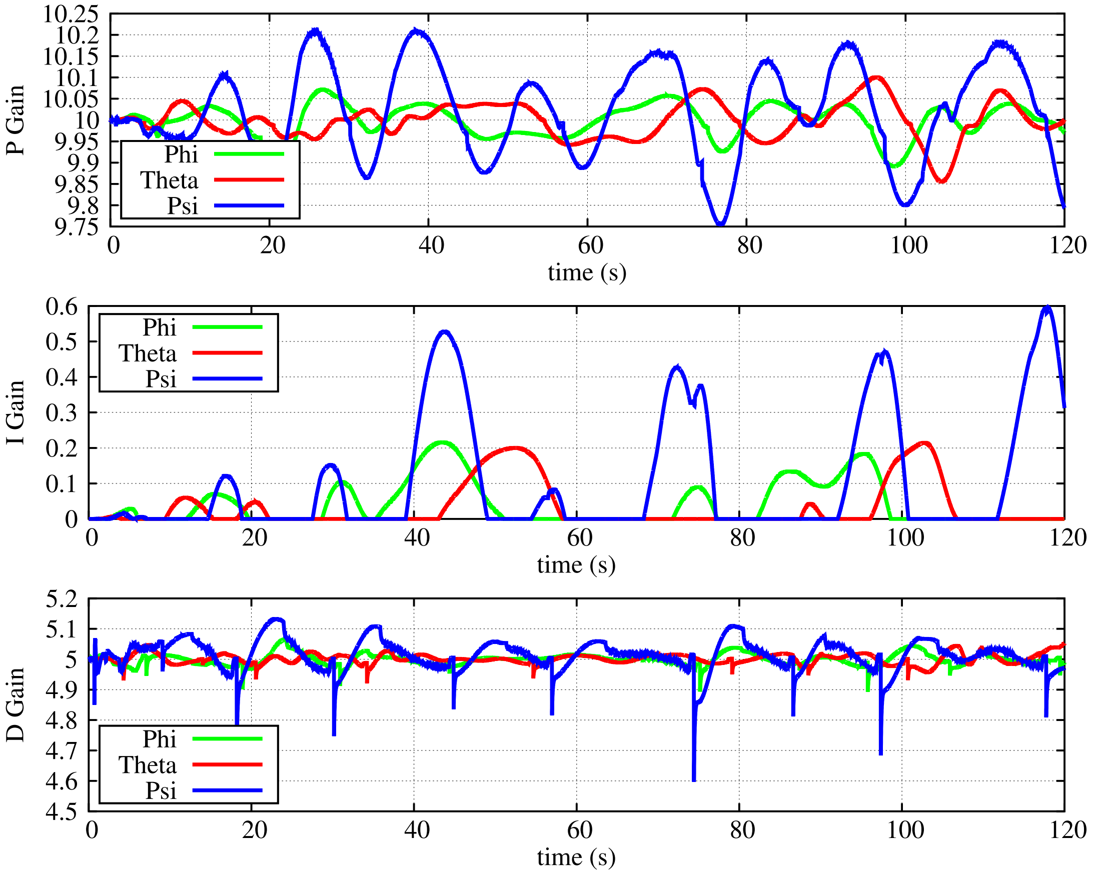


Figure 6.24: Adaptation History of the Euler Angle Controller Gains

Learning process of PID gains of the Euler Angle Controller of the Intelligent Flight Controller is depicted in figure 6.24. Because of the complexity of the pirouette maneuver, PID gains of the Euler Angle Controller are not as smooth as in the pull up - push over and slalom maneuvers.

There are sinusoidal oscillations for all gains, but the amplitude of these oscillations are small. In addition, there are short-time small peaks for derivative gains. In spite of the oscillations and small peaks, the Intelligent Flight Controller can follow the desired trajectory in this maneuver. As in the coning maneuver, the most oscillatory gains of the Euler Angle Controller are the psi controller gains because of the constant heading rate and continuously decreasing radius of the trajectory of the pirouette maneuver.

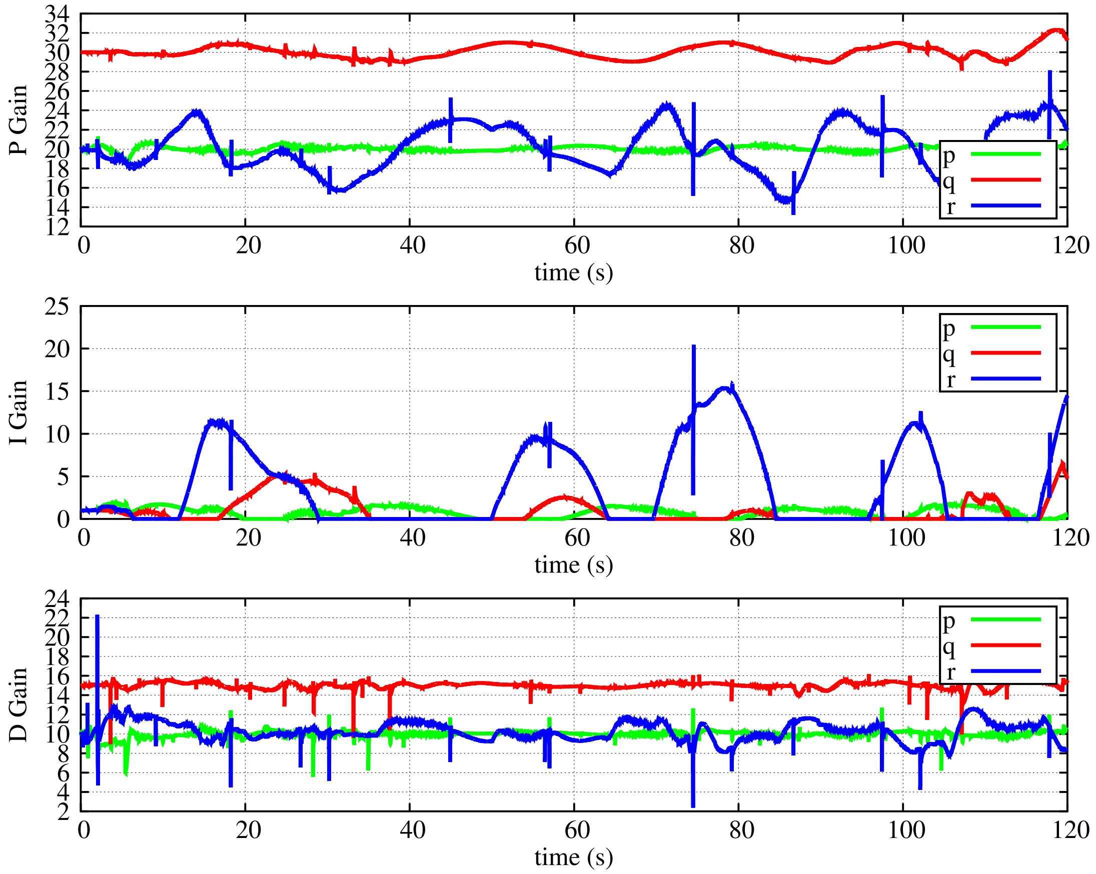


Figure 6.25: Adaptation History of the Body Angular Velocity Controller Gains

Adaptation process of PID gains of the Body Angular Velocity Controller of the Intelligent Flight Controller is shown in figure 6.25. Proportional, derivative and integral gains have continuous oscillations with small amplitudes and short-time small peaks except r controller.

Amplitudes of peaks in derivative gain, amplitudes of oscillations in proportional and integral gains of r controller are not so small because of the constant heading rate and continuously decreasing radius of the trajectory of the pirouette maneuver.

In spite of these oscillations and peaks, the Intelligent Flight Controller can control the helicopter until the end of the simulation and follows the desired trajectory.

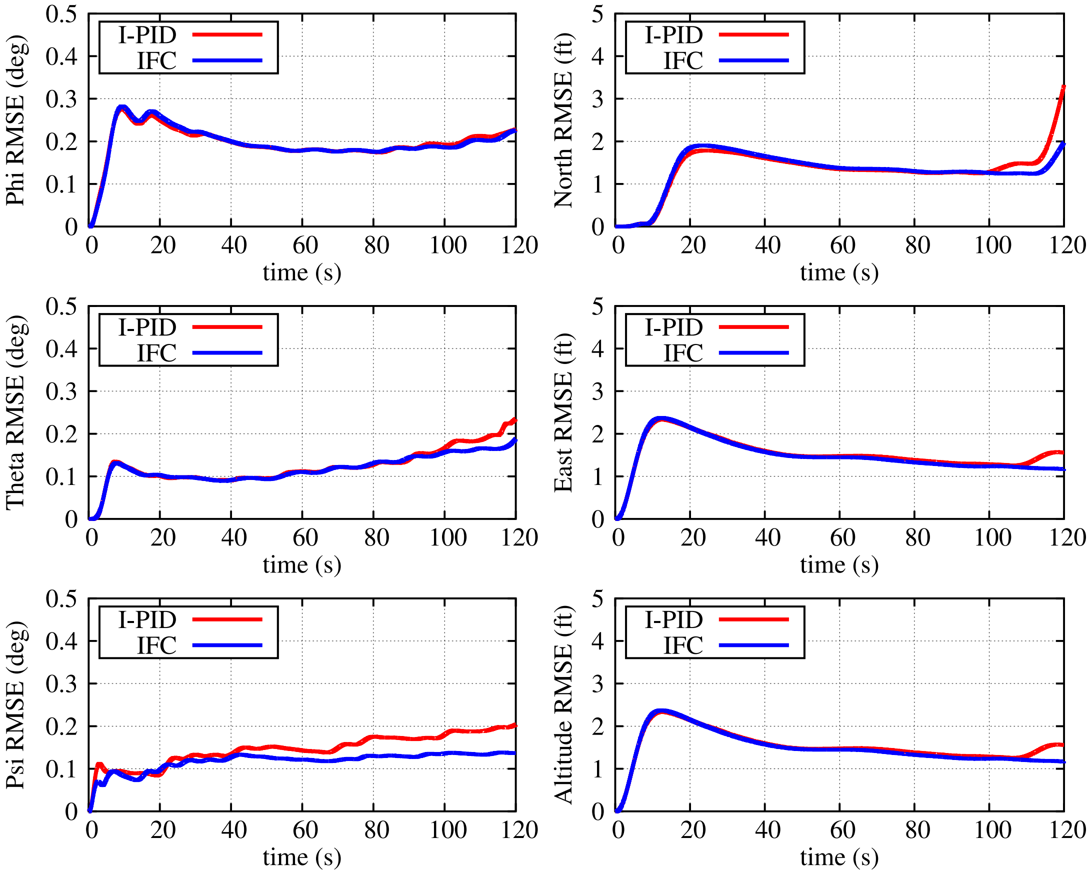


Figure 6.26: RMSE Analyses of the Positions of the Helicopter in Pirouette Maneuver

As it is shown in figure 6.26, root mean square error analyses of six positions for both controllers are similar until the last 20 seconds. The Intelligent Flight Controller is still more effective for all positions. According to RMSE analysis, the trajectory followed by I-PID controller diverges from the reference in the last 20 seconds. But, the Intelligent Flight Controller maintains to track the reference trajectory until the end of the simulation. In addition to the root mean square error analysis, it is shown in figure 6.27 that the Intelligent Flight Controller completes the given mission with more stable control inputs.

In conclusion, the Intelligent Flight Controller is more successful for following the desired trajectory than the I-PID controller with fixed gains in pirouette maneuver as seen on figure 6.23.

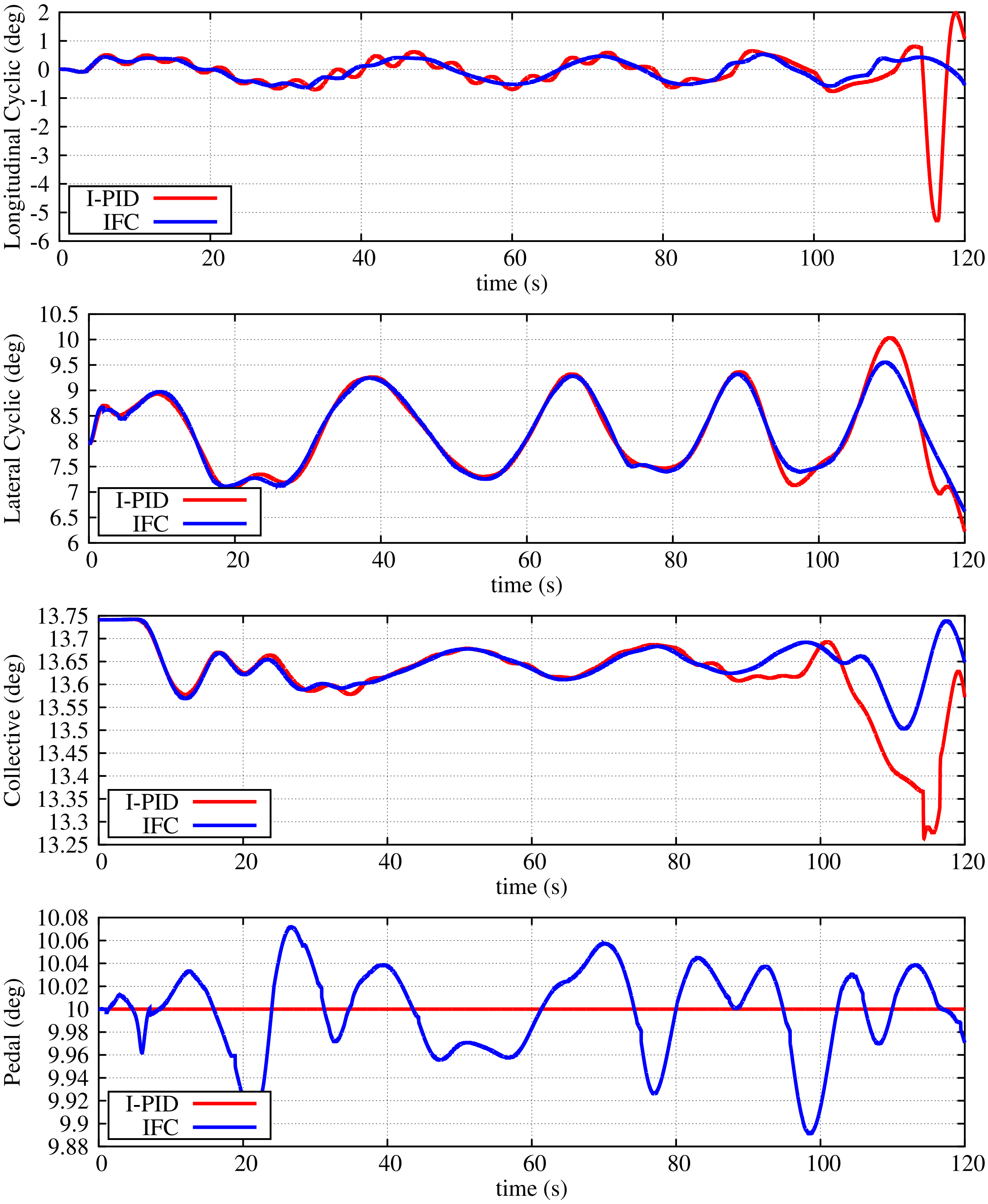


Figure 6.27: Control Inputs Generated by I-PID and IFC in Pirouette Maneuver

6.2.6 3-D Cone Maneuver

The last maneuver is the combination of all maneuvers to challenge the flight controller as much as possible. A continuously rounding path with a decreasing radius like drawing a cone is expected for all positions. 3.44 deg/s yaw rate and a constant

3 knots total velocity are expected in this maneuver. At the beginning, the maneuver is much simpler for flight controllers. But, as the radius of the cone decreases, path tracking becomes increasingly challenging due to the constant heading rate.

Starting from the initial positions $X_N = 0, X_E = 0, X_D = -1000$ and $\psi = 0$, the commanded position equations for 3-D Cone maneuver,

$$X_N = \int V_{Total}\, sin\left(\textstyle\int \psi\, dt\right)\, dt + \int \left|sin\left(\textstyle\int \psi\, dt\right)\right| dt \tag{6.26}$$

$$X_E = \int V_{Total}\, cos\left(\textstyle\int \psi\, dt\right)\, dt \tag{6.27}$$

$$X_D = -1000 + \int V_{Total}\, sin\left(\textstyle\int \psi\, dt\right)\, dt \tag{6.28}$$

$$\dot{\psi} = 2\,\frac{V_{Total}}{R} \tag{6.29}$$

where R=100 ft and V_{Total}=3 knots.

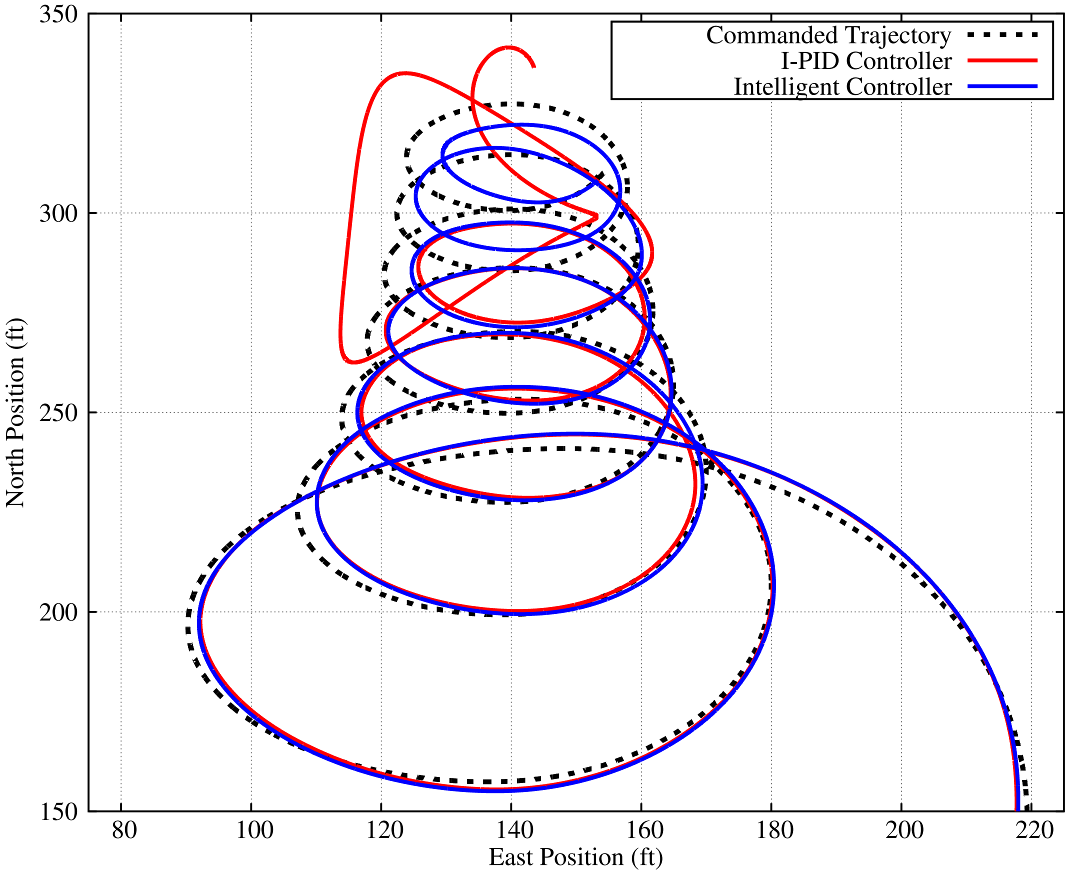


Figure 6.28: Trajectory of the Helicopter on X-Y Plane

As the simulation continues, completion of the loops in this maneuver gets harder an harder for both controllers due to the constant heading rate and continuously decreasing radius of the trajectory like coning maneuver. Unlike the coning and pirouette maneuvers, the simulation is conducted for 300 seconds for this maneuver because of the slower total velocity. At the beginning, helicopter accelerates to reach 3 knots from hover to start to draw a 3-D cone with a constant yaw rate.

Both I-PID and the Intelligent Flight Controller are able to follow the trajectory for 3-D cone maneuver until the last 40 seconds. In the last 40 seconds of the simulation, I-PID controller leaves the trajectory as seen in figure 6.28. However, the Intelligent Flight Controller continues to complete loops for this maneuver.

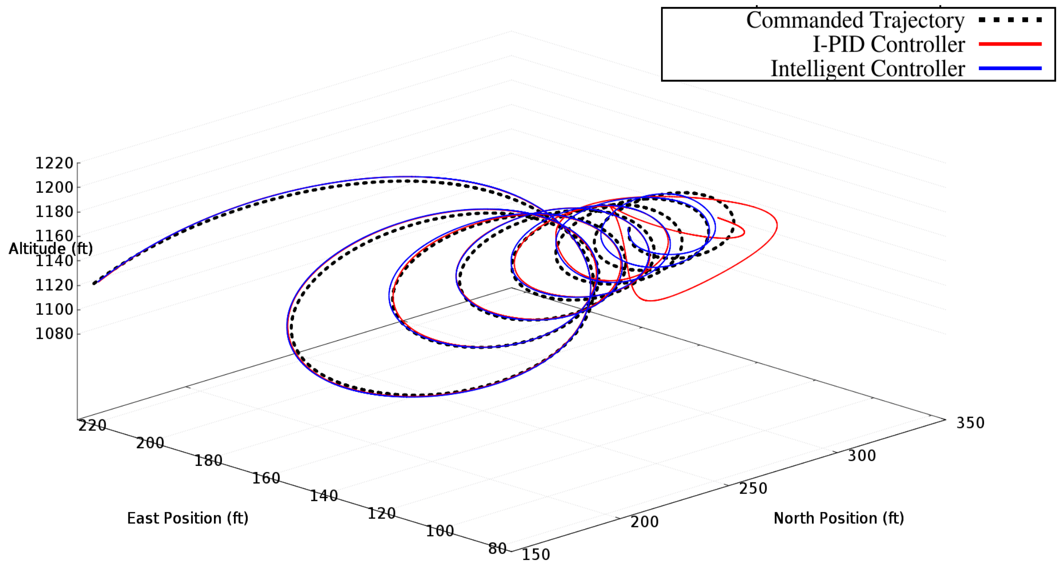


Figure 6.29: Isometric View of the Trajectory of the Helicopter

The 3-D cone shape of the followed trajectory for 3-D cone maneuver is seen in figure 6.29. In addition, effectiveness of the Intelligent Flight Controller and divergence of the I-PID controller from desired trajectory are seen obviously in figure 6.29.

Since the altitude is controlled by the outer loop, no difference is expected in altitude channel for both I-PID and the Intelligent Flight Controller for this maneuver. Because, rate of the altitude change is not so much as in pull up - push over maneuver. In addition, only the PID gains of the inner and the innermost loops are changed by Intelligent Flight Controller.

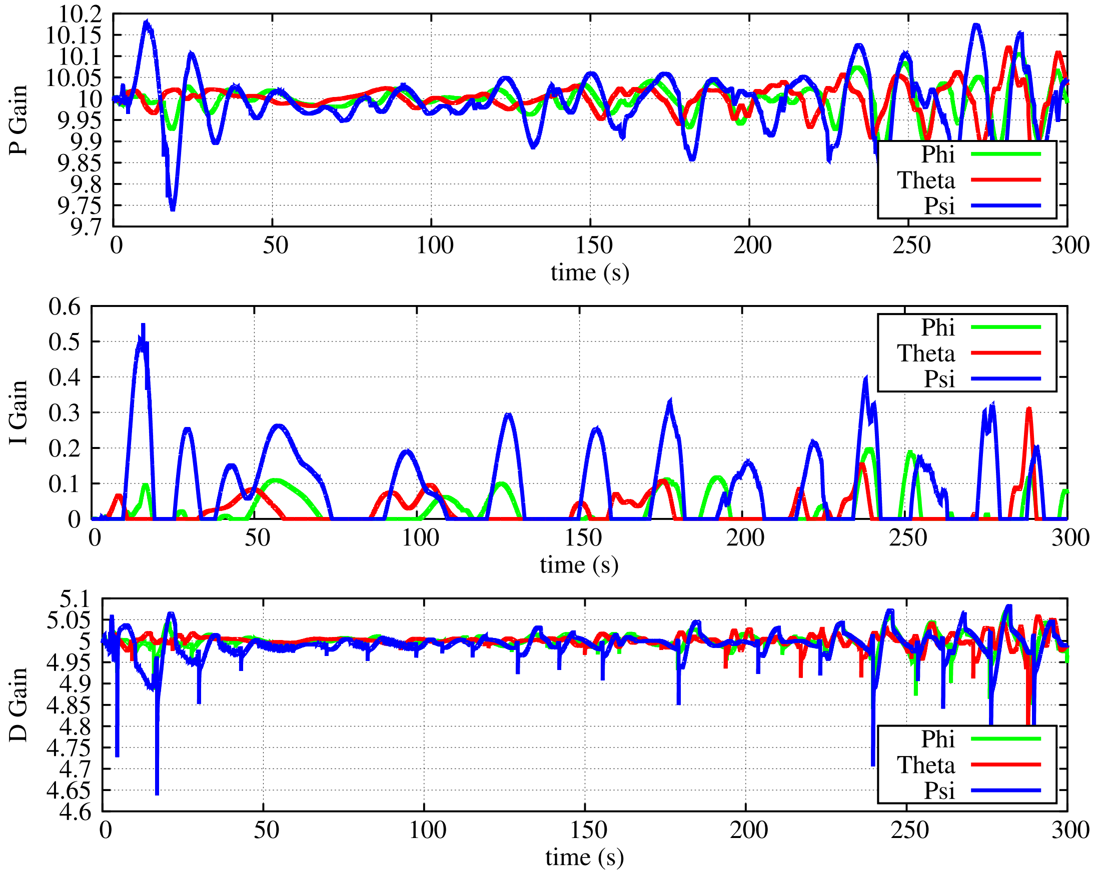


Figure 6.30: Adaptation History of the Euler Angle Controller Gains

Learning process of PID gains of the Euler Angle Controller of the Intelligent Flight Controller is depicted in figure 6.30. Because of the complexity of the 3-D cone maneuver, PID gains of the Euler Angle Controller are oscillatory as in coning and piroutte maneuvers.

The amplitude of the PID gain oscillations are small in magnitude. In addition, there are short-time small peaks for derivative gains. In spite of the oscillations and small peaks, the Intelligent Flight Controller can follow the desired trajectory better than I-PID controller in this maneuver. As in the coning and piroutte maneuvers, the most oscillatory gains of the Euler Angle Controller are the psi controller gains because of the constant heading rate and continuously decreasing radius of the trajectory of the 3-D cone maneuver.

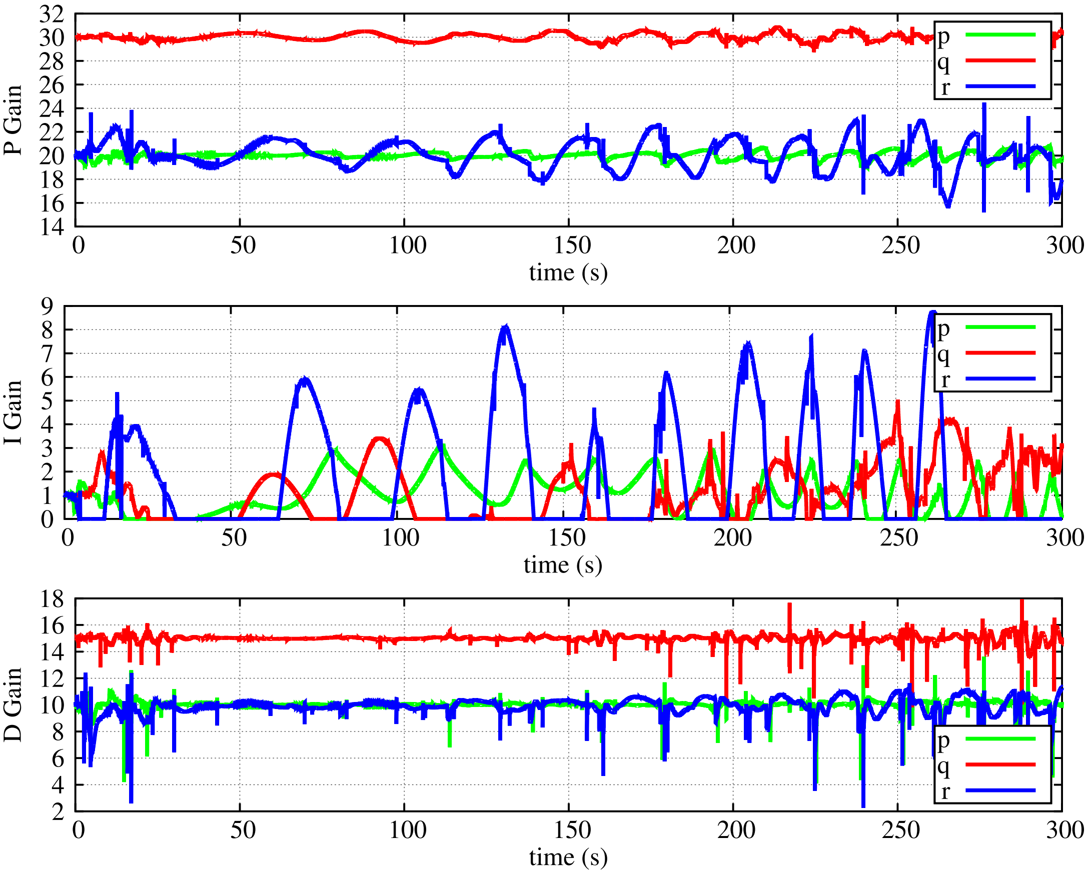


Figure 6.31: Adaptation History of the Body Angular Velocity Controller Gains

Adaptation process of PID gains of the Body Angular Velocity Controller of the Intelligent Flight Controller is shown in figure 6.31. Proportional and derivative gains are nearly constant. In addition derivative gains have small short-time peaks and there are periodic oscillations with small amplitudes in integral gains of p and q controllers.

Amplitudes of peaks in derivative gains, amplitudes of oscillations in proportional and integral gains of r controller are not so small because of the constant heading rate and continuously decreasing radius of the trajectory of the 3-D cone maneuver.

Although existence of these oscillations and peaks, the Intelligent Flight Controller can control the helicopter better than the I-PID controller for 3-D cone maneuver.

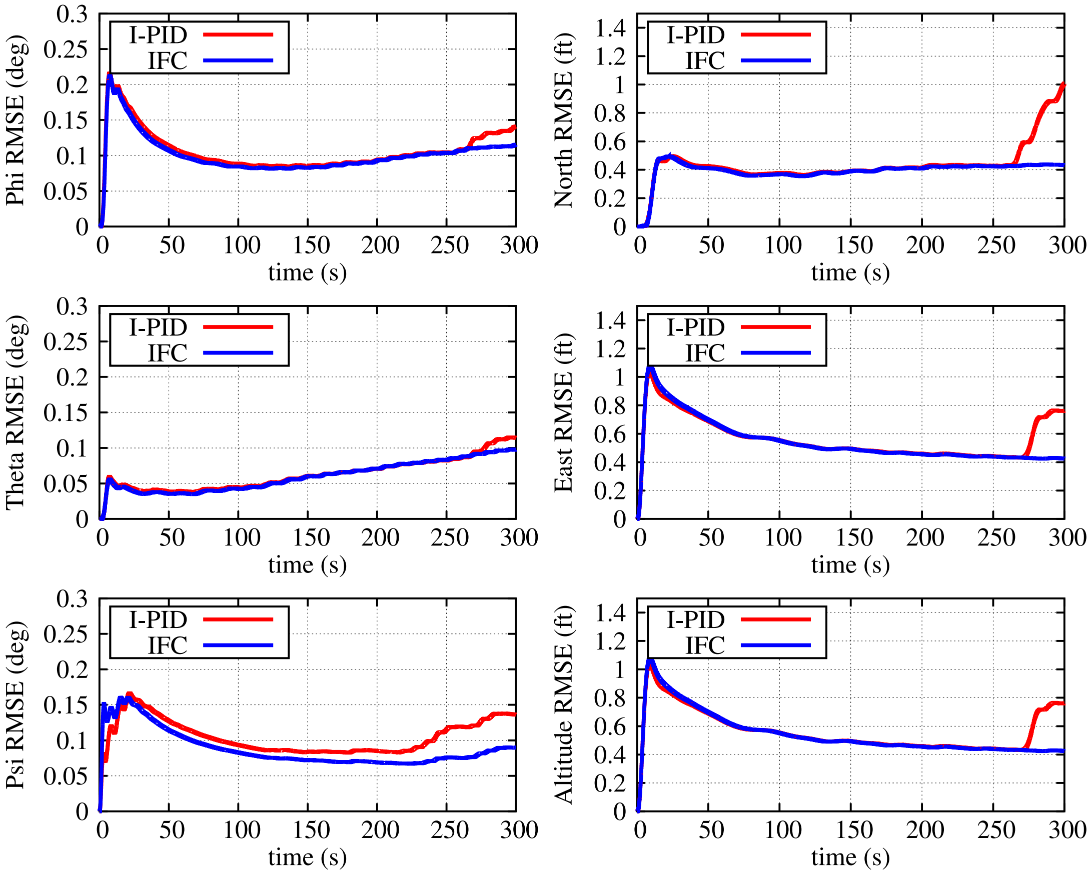


Figure 6.32: RMSE Analyses of the Positions of the Helicopter in 3-D Cone Maneuver

As it is shown in figure 6.32, root mean square error analyses of six positions for both controllers are similar until the last 40 seconds. The Intelligent Flight Controller is still more effective for all positions. According to RMSE analysis, the trajectory followed by I-PID controller diverges from the reference in the last 40 seconds. However, the Intelligent Flight Controller maintains to follow the reference trajectory until the end of the simulation. In addition to the RMSE analysis, it is shown in figure 6.33 that the IFC completes the given mission with more stable control inputs.

To sum up, the IFC is more successful for following the desired trajectory than the I-PID controller with fixed gains in 3-D cone maneuver as seen on figure 6.28.

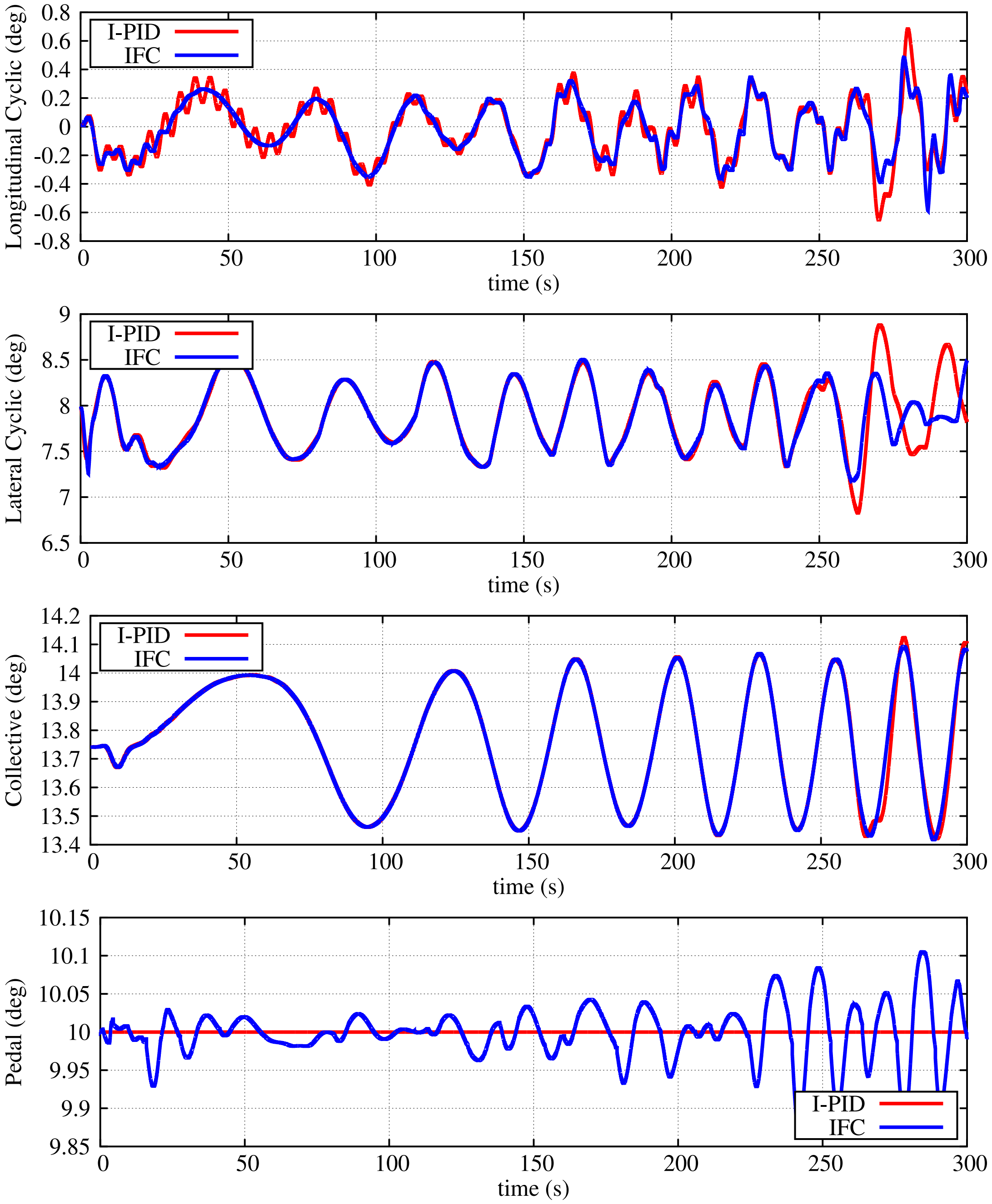


Figure 6.33: Control Inputs Generated by I-PID and the Intelligent Flight Controller in 3-D Cone Maneuver

CHAPTER 7

CONCLUSION

In this study, a new intelligent flight controller is developed with model prediction and adaptation abilities for a full sized autonomous helicopter. Testing platform for the intelligent controller is selected as the UH-1H helicopter. To achieve this, UH-1H helicopter is converted to an unmanned platform in virtual environment by developing a full flight controller for hover and forward flight conditions.

For a contribution to the literature, a PID gain update law using linear least squares regression is proposed as a new adaptive control method. Moreover, future prediction analysis are conducted for error dynamics of the closed loop system using recursive linear least squares regression. These two concepts are combined with conventional PID controller and an intelligent PID controller is acquired.

Excluding the velocity controllers, PID controllers of the flight controller are replaced by the newly developed intelligent PID controller. Thus, a new intelligent flight controller is obtained with model prediction and adaptation abilities for autonomous helicopters.

Finally, several complex maneuvers are conducted to challenge the intelligent flight controller. The first flight controller that has no adaptation ability and the new intelligent flight controller are tested with same initally stable PID gains to demonstrate the effectiveness of the new intelligent flight controller.

All challenging maneuvers are completed successfully by the new Intelligent Flight Controller (IFC). However, these complex maneuvers are too aggressive for the flight controller with fixed PID gains. Therefore, most of the maneuvers are ended due to crash of the helicopter with this flight controller although using the same initial PID gains with IFC.

However, the new controller IFC is not more effective than a very well tuned PID. The results of the comparison between IFC and a very well tuned PID is not given in this study because of the difference in the success ratio of these methods is approximately 1-3%. As stated in the introduction section, tuning the gains of the PID controllers is very hard in practice. Therefore, acquiring a very well tuned PID is not an easy process without using any advanced techniques like adaptive control. The main objective for an adaptive controller is to find the optimum PID gains while maximizing given performance index. The new controller IFC achieves such objective while satisfying constraints, even starting from a very rough initial PID gains.

To sum up, a new PID update law using Linear Least Squares Regression is proposed for adaptation. Implementing the acceleration of the command, an I-PID controller is obtained. This I-PID controller is converted to a new intelligent I-PID controller by adding adaptation and prediction abilities using Linear Least Squares Regression and Recursive Linear Least Squares Regression, respectively. On the other hand, a flight controller with three control loops is developed using I-PID controllers to convert UH-1H to an autonomous helicopter. Then, the I-PID controllers of the second and third loops of this flight controller are replaced by newly developed intelligent I-PID controllers. Consequently, an effective intelligent flight controller is acquired for autonomous helicopters.

7.1 Future Work

Suggested improvements for this thesis are listed below,

- PID gain update law can be extended for Proportional Integral Derivative Acceleration (PIDA) controllers.
- Damped Linear Least Squares regression can be used for gain optimization using forgetting factors for back-stepping.
- Non-Linear Least Squares regression methods can be used for adaptation of

PID gains and prediction of the error.

- Controller can be challenged for more complex maneuvers.
- Initial PID gains can be selected from an unstable closed loop condition to analyze the fidelity range of the intelligent flight controller.
- Different gain update rules can be applied for the same intelligent flight controller.

REFERENCES

[1] Cai, G., Chen, B., Peng, K., Dong, M., and Lee, T., Industrial Electronics, IEEE Transactions on **55** (2008) 3426.

[2] Azeem, S., Autonomous unmanned aerial vehicles, Technical report, The George Washington University, 2012.

[3] Beharrell, A., Converting an inexpensive rc helicopter into an autonomous vehicle, Master's thesis, University Of Southampton, 2011.

[4] Haddal, C. C. and Gertler, J., Homeland security: Unmanned aerial vehicles and border surveillance, Technical report, U.S. Congressional Research Service, 2010, Retrieved from: http://nsarchive.gwu.edu/NSAEBB/NSAEBB527-Using-overhead-imagery-to-track-domestic-US-targets/documents/EBB-Doc24.pdf, Last Accessed: May 22, 2016.

[5] Nicas, J., Drones find fans among farmers, filmmakers, Retrieved from: http://www.wsj.com/articles/, 2014, Article ID: SB10001424052702304732804579425342990070808, Last Accessed: May 22, 2016.

[6] Autonomous drone developed for Fukushima reactors (11.06.2015). Retrieved from: http://www.bbc.com/news/blogs-news-from-elsewhere-33096077, Last Accessed: May 22, 2016.

[7] Zhang, L. et al., Advanced Science and Technology Letters **81** (2015) 69.

[8] Smith, R., Utilities turn to drones to inspect power lines and pipelines, Retrieved from: http://www.wsj.com/articles/utilities-turn-to-drones-to-inspect-power-lines-and-pipelines-1430881491, 2015, Last Accessed: May 22, 2016.

[9] Hedges, S. and O'Brien, T., Monitoring Elephant Populations and Assessing Threats (2012).

[10] Gasperini, D., Allemand, P., Delacourt, C., and Grandjean, P., International Journal of Environmental Technology and Management **17** (2014) 1.

[11] Khaleghi, A. M. et al., Agent-based hardware-in-the-loop simulation for uav/ugv surveillance and crowd control system, in *Proceedings of the 2013 Winter Simulation Conference: Simulation: Making Decisions in a Complex World*, pages 1455–1466, IEEE Press, 2013.

[12] Lee, C., Lee, Y., and Teng, C., A novel robust pid controllers design by fuzzy neural network, in *American Control Conference, 2002. Proceedings of the 2002*, volume 2, pages 1561–1566 vol.2, 2002.

[13] Ming-guang, Z., Xing-gui, W., and qiang Liu, M., Adaptive pid control based on rbf neural network identification, in *Tools with Artificial Intelligence, 2005. ICTAI 05. 17th IEEE International Conference on*, pages 3 pp.–683, 2005.

[14] Zein-Sabatto, S. and Zheng, Y., Intelligent flight controllers for helicopter control, in *Neural Networks,1997., International Conference on*, volume 2, pages 617–621 vol.2, 1997.

[15] Sanchez, E. N., Becerra, H. M., and Velez, C. M., Information Sciences **177** (2007) 1999 , Including Special Issue on Hybrid Intelligent Systems.

[16] Tarimci, O., Adaptive controller applications for rotary wing aircraft models of varying simulation fidelity, Master's thesis, METU, 2009.

[17] Sadeghzadeh, I., Mehta, A., Zhang, Y., and Rabbath, C.-A., Fault-tolerant trajectory tracking control of a quadrotor helicopter using gain-scheduled pid and model reference adaptive control, in *Annual Conference of the Prognostics and Health Management Society*, volume 2, 2011.

[18] Sahu, P. and Pradhan, S., Multi variable self tuning pid control of a twin rotor mimo system, in *Advanced Communication Control and Computing Technologies (ICACCCT), 2014 International Conference on*, pages 381–385, 2014.

[19] Gao, H., Liu, C., Guo, D., and Liu, J., Fuzzy adaptive pd control for quadrotor helicopter, in *Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2015 IEEE International Conference on*, pages 281–286, 2015.

[20] Brickwedde, A., Industry Applications, IEEE Transactions on **IA-21** (1985) 1154.

[21] Éric Poulin, Pomerleau, A., Desbiens, A., and Hodouin, D., Automatica **32** (1996) 71 .

[22] Rad, A., Lo, W. L., and Tsang, K., Industrial Electronics, IEEE Transactions on **44** (1997) 717.

[23] Mitsukura, Y., Yamamoto, T., and Kaneda, M., A design of self-tuning pid controllers using a genetic algorithm, in *American Control Conference, 1999. Proceedings of the 1999*, volume 2, pages 1361–1365 vol.2, 1999.

[24] Grassi, E., Tsakalis, K. S., Dash, S., Gaikwad, S., and Stein, G., Adaptive/self-tuning pid control by frequency loop-shaping, in *Decision and Control, 2000. Proceedings of the 39th IEEE Conference on*, volume 2, pages 1099–1101 vol.2, 2000.

[25] Chen, J. and Cheng, Y.-C., Industrial & engineering chemistry research **43** (2004) 5888.

[26] Yamamoto, T. and Shah, S., Design of a performance-adaptive pid controller, in *Control Conference (ECC), 2007 European*, pages 4799–4804, 2007.

[27] Wanfeng, S., Shengdun, Z., and Yajing, S., Adaptive pid controller based on online lssvm identification, in *Advanced Intelligent Mechatronics, 2008. AIM 2008. IEEE/ASME International Conference on*, pages 694–698, 2008.

[28] Zhao, J., Li, P., and song Wang, X., Intelligent pid controller design with adaptive criterion adjustment via least squares support vector machine, in *Control and Decision Conference, 2009. CCDC '09. Chinese*, pages 7–12, 2009.

[29] Wilson, A., Schultz, J., and Murphey, T., Automation Science and Engineering, IEEE Transactions on **12** (2015) 28.

[30] Tao, G., *Adaptive Control Design and Analysis*, Wiley, 2003.

[31] Bretscher, O., *Linear Algebra with Applications*, Printice Hall, 2001.

[32] Chowdhary, G. V., *Concurrent Learning For Convergence In Adaptive Control Without Persistency Of Excitation*, PhD thesis, Daniel Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, 2010.

[33] Kim, K., Rao, P., and Burnworth, J. A., IEEE Transactions on Industry Applications **46** (2010).

[34] Burger, M. and Repiský, J., Problems of linear least square regression, in *ARSA*, volume 1, pages 257–262, EDIS - Publishing Institution of the University of Zilina, 2012.

[35] Renze, John. "Outlier." From MathWorld–A Wolfram Web Resource, created by Eric W. Weisstein. http://mathworld.wolfram.com/Outlier.html, Last Accessed: May 22, 2016.

[36] Prasad, J., Calise, A., Pei, Y., and Corban, J. E., Adaptive nonlinear controller synthesis and flight test evaluation on an unmanned helicopter, in *International Conference on Control Applications*, volume 1, pages 137–142, IEEE, 1999.

[37] Official Website of Heli-Dyn: http://www.heli-dyn.com/Helidyn/Home.html, Last Accessed: May 22, 2016.

[38] Talbot, P. D. and Corliss, L. D., A mathematical force and moment model of a uh-1h helicopter for flight dynamics simulations, Technical report, NASA, Prepared in Cooperation with USAAMRDL Moffett Field, Calif., 1977, Document ID:19770024231,.

[39] Heffley, R. K. and Mnich, M. A., Minimum-complexity helicopter simulation math model, Technical report, NASA, Prepared for Army Research and Technology Labs., Moffett Field, Calif., 1988, NASA-CR-177476, NAS 1.26:177476, USAAVSCOM-TR-87-A-7.

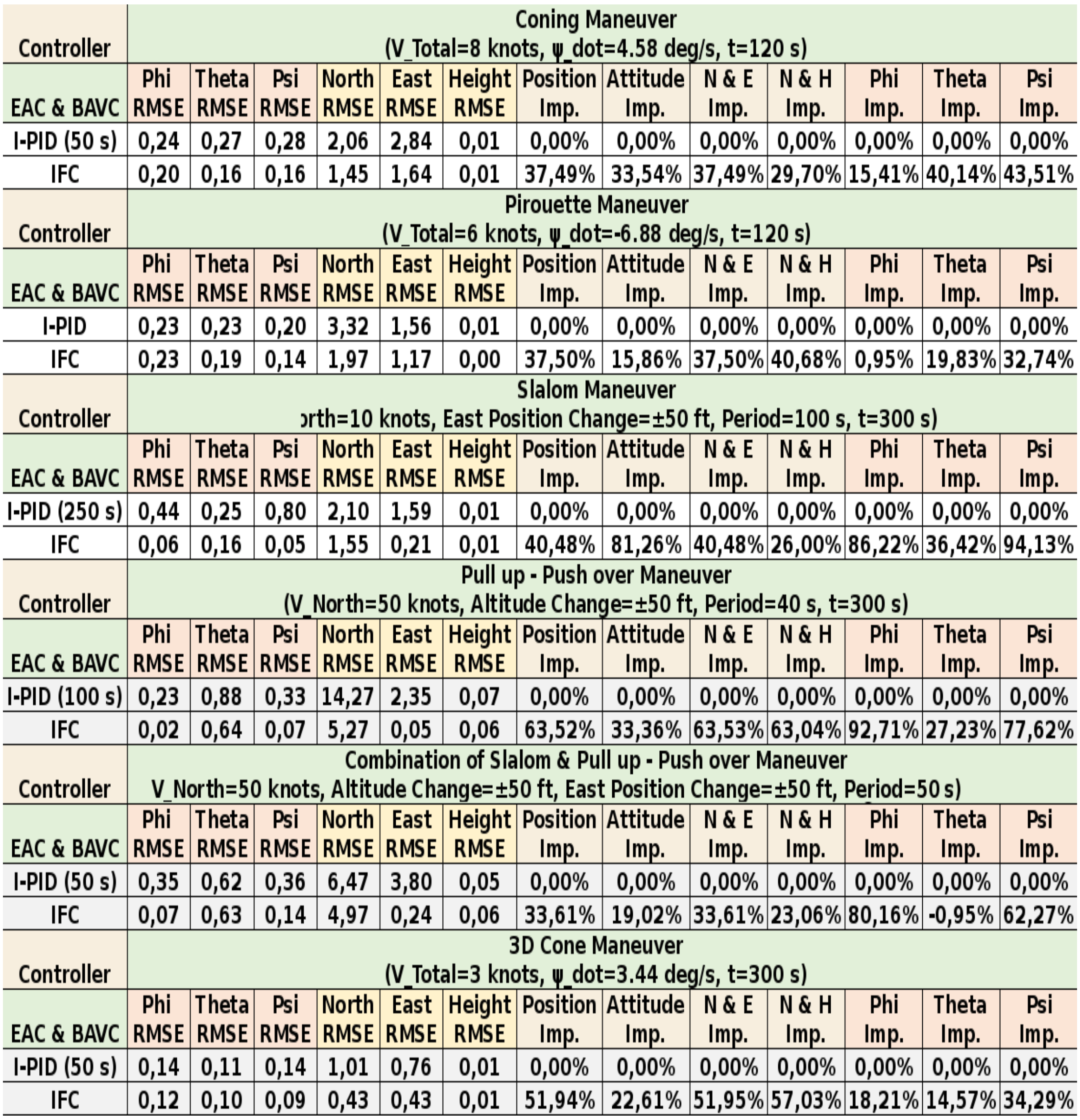
[40] Peters, D. A., Boyd, D. D., and He, C. J., Journal of the American Helicopter Society **34** (1989) 5.

[41] Yilmaz, D., Evaluation and comparison of helicopter simulation models with different fidelities, Master's thesis, METU, 2008.

[42] Yavrucuk, I., Tarimci, O., Katircioglu, M., Kubali, E., and Yilmaz, D., A new helicopter simulation and analysis tool: Helidyn+, European Rotorcraft Forum, 2010.

APPENDIX A

ROOT MEAN SQUARE ERROR ANALYSIS OF MANEUVERS



Controller	Coning Maneuver (V_Total=8 knots, ψ_dot=4.58 deg/s, t=120 s)												
EAC & BAVC	Phi RMSE	Theta RMSE	Psi RMSE	North RMSE	East RMSE	Height RMSE	Position Imp.	Attitude Imp.	N & E Imp.	N & H Imp.	Phi Imp.	Theta Imp.	Psi Imp.
I-PID (50 s)	0,24	0,27	0,28	2,06	2,84	0,01	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
IFC	0,20	0,16	0,16	1,45	1,64	0,01	37,49%	33,54%	37,49%	29,70%	15,41%	40,14%	43,51%
Controller	Pirouette Maneuver (V_Total=6 knots, ψ_dot=-6.88 deg/s, t=120 s)												
EAC & BAVC	Phi RMSE	Theta RMSE	Psi RMSE	North RMSE	East RMSE	Height RMSE	Position Imp.	Attitude Imp.	N & E Imp.	N & H Imp.	Phi Imp.	Theta Imp.	Psi Imp.
I-PID	0,23	0,23	0,20	3,32	1,56	0,01	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
IFC	0,23	0,19	0,14	1,97	1,17	0,00	37,50%	15,86%	37,50%	40,68%	0,95%	19,83%	32,74%
Controller	Slalom Maneuver orth=10 knots, East Position Change=±50 ft, Period=100 s, t=300 s)												
EAC & BAVC	Phi RMSE	Theta RMSE	Psi RMSE	North RMSE	East RMSE	Height RMSE	Position Imp.	Attitude Imp.	N & E Imp.	N & H Imp.	Phi Imp.	Theta Imp.	Psi Imp.
I-PID (250 s)	0,44	0,25	0,80	2,10	1,59	0,01	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
IFC	0,06	0,16	0,05	1,55	0,21	0,01	40,48%	81,26%	40,48%	26,00%	86,22%	36,42%	94,13%
Controller	Pull up - Push over Maneuver (V_North=50 knots, Altitude Change=±50 ft, Period=40 s, t=300 s)												
EAC & BAVC	Phi RMSE	Theta RMSE	Psi RMSE	North RMSE	East RMSE	Height RMSE	Position Imp.	Attitude Imp.	N & E Imp.	N & H Imp.	Phi Imp.	Theta Imp.	Psi Imp.
I-PID (100 s)	0,23	0,88	0,33	14,27	2,35	0,07	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
IFC	0,02	0,64	0,07	5,27	0,05	0,06	63,52%	33,36%	63,53%	63,04%	92,71%	27,23%	77,62%
Controller	Combination of Slalom & Pull up - Push over Maneuver V_North=50 knots, Altitude Change=±50 ft, East Position Change=±50 ft, Period=50 s)												
EAC & BAVC	Phi RMSE	Theta RMSE	Psi RMSE	North RMSE	East RMSE	Height RMSE	Position Imp.	Attitude Imp.	N & E Imp.	N & H Imp.	Phi Imp.	Theta Imp.	Psi Imp.
I-PID (50 s)	0,35	0,62	0,36	6,47	3,80	0,05	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
IFC	0,07	0,63	0,14	4,97	0,24	0,06	33,61%	19,02%	33,61%	23,06%	80,16%	-0,95%	62,27%
Controller	3D Cone Maneuver (V_Total=3 knots, ψ_dot=3.44 deg/s, t=300 s)												
EAC & BAVC	Phi RMSE	Theta RMSE	Psi RMSE	North RMSE	East RMSE	Height RMSE	Position Imp.	Attitude Imp.	N & E Imp.	N & H Imp.	Phi Imp.	Theta Imp.	Psi Imp.
I-PID (50 s)	0,14	0,11	0,14	1,01	0,76	0,01	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
IFC	0,12	0,10	0,09	0,43	0,43	0,01	51,94%	22,61%	51,95%	57,03%	18,21%	14,57%	34,29%

Figure A.1: RMSE Analysis of Maneuvers