

OPTIMAL TRAJECTORY GENERATION AND TRACKING FOR A  
HELICOPTER IN TAIL ROTOR FAILURE

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

BY

YUSUF ONUR ARSLAN

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

AUGUST 2022



Approval of the thesis:

**OPTIMAL TRAJECTORY GENERATION AND TRACKING FOR A  
HELICOPTER IN TAIL ROTOR FAILURE**

submitted by **YUSUF ONUR ARSLAN** in partial fulfillment of the requirements  
for the degree of **Master of Science in Aerospace Engineering, Middle East  
Technical University** by,

Prof. Dr. Halil Kalıpçılar  
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Serkan Özgen  
Head of the Department, **Aerospace Engineering**

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

**Examining Committee Members:**

Assoc. Prof. Dr. Nilay Sezer Uzol  
Aerospace Engineering, METU

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

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

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

Assist. Prof. Dr. Yıldırım Yıldız  
Mechanical Engineering, Bilkent University

Date: 23.08.2022

**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 : Yusuf Onur Arslan

Signature :

## **ABSTRACT**

### **OPTIMAL TRAJECTORY GENERATION AND TRACKING FOR A HELICOPTER IN TAIL ROTOR FAILURE**

Arslan, Yusuf Onur  
Master of Science, Aerospace Engineering  
Supervisor : Assoc. Prof. Dr. İlkey Yavrucuk

August 2022, 107 pages

Tail rotor failure is the main reason for a variety of helicopter accidents throughout aviation history. It is troublesome to control and navigate the helicopter to a predefined landing point with a non-functioning tail rotor especially if the number of available landing sites is limited. The objective of this thesis is to generate flight trajectories for a helicopter with a failed tail rotor and to track the trajectories by a controller. In this work, the generic helicopter model in FLIGHTLAB software is utilized and a tail rotor failed model is created. The path generation process is performed offline in MATLAB environment with an open-source trajectory optimization tool, OptimTraj. The trajectories are obtained for different scenarios by defining several constraints. An inversion-based controller is designed in order to bring the helicopter to a specified point in space by position trajectory tracking. An LQI controller is designed in order to change the trim condition of the helicopter by state trajectory tracking. The trajectories are given as a reference to the tracking controllers and simulations are performed with both linear and nonlinear models by communicating FLIGHTLAB and SIMULINK.

Keywords: Helicopter Tail Rotor Failure, Trajectory Generation, Trajectory Tracking, Optimization, Control

## ÖZ

### **KUYRUK ROTORU ARIZALI BİR HELİKOPTER İÇİN OPTİMAL YÖRÜNGE ÜRETİMİ VE TAKİBİ**

Arslan, Yusuf Onur  
Yüksek Lisans, Havacılık ve Uzay Mühendisliği  
Tez Yöneticisi: Assoc. Prof. Dr. İlkey Yavrucuk

Ağustos 2022, 107 sayfa

Kuyruk rotoru arızası, havacılık tarihi boyunca birçok helikopter kazasının ana nedeni olmuştur. Özellikle uygun iniş alanı sayısının sınırlı olduğu durumlarda, kuyruk rotoru işlevini yitirmiş bir helikopteri kontrol etmek ve belirli bir iniş noktasına yönlendirmek oldukça zahmetlidir. Bu tezin amacı, kuyruk rotoru arızalı bir helikopter için uçuş yörüngesi oluşturmak ve bu yörüngeyi bir kontrolcü ile takip etmektir. Bu çalışmada, FLIGHTLAB yazılımı içerisindeki genel helikopter modelinden faydalanılmış ve kuyruk rotoru arızalı model oluşturulmuştur. Patika bulma işlemi MATLAB ortamında bir açık kaynak yörünge optimizasyon aracı olan OptimTraj ile çevrimdışı olarak yapılmıştır. Yörüngeler, farklı senaryolar için bazı kısıtlamalar tanımlanarak elde edilmiştir. Helikopteri uzayda belirtilen bir noktaya konum yörüngesi takibi ile getirmek için, tersleme tabanlı bir kontrolcü tasarlanmıştır. Helikopterin denge durumunu durum yörüngesi takibi ile değiştirmek için, bir LQI kontrolcüsü tasarlanmıştır. Yörüngeler izleme kontrolcüsüne referans olarak verilmiş ve FLIGHTLAB ile SIMULINK haberleştirilerek doğrusal ve doğrusal olmayan modellerin ikisi ile de simülasyonlar yapılmıştır.

Anahtar Kelimeler: Helikopter Kuyruk Rotoru Arızası, Yörünge Üretimi, Yörünge İzleme, Optimizasyon, Kontrol



To my family, friends and teachers

## ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my supervisor Assoc. Prof. Dr. İlkey Yavrucuk for his guidance and feedback throughout this thesis work.

I would like to emphasize my special thanks to Ilgaz Doğa Okcu for his suggestions, invaluable comments, and friendship. I would also like to express my appreciation to Kaan Şansal, Göktuğ Koçak, Umut Türe, Cenk Çetin, Arda Çalışkan, Ceren Cansu Esmek, Özge Kapulu Gülbağ and Sevil Avcıoğlu for their precious advices and endless support.

I want to thank to Oğuz Han Altıntaş for his passion for science, technical assistance and friendship. Many thanks to Ali Er, Yasin Öztürk, Mehmet Berker Bavbek, Uğurcan Özalp, Ozan Can Karatepe, Hüseyin Emir Can and Hüdai Alaylı for their encouragement, empathy and patience.

Finally, I would like to express my deepest thanks to my family. Many thanks to my mother, Hatice Arslan, for her endless love and patience; to my father, Şenol Arslan, for his everlasting support and kindness; and lastly, to my sister, İrem Arslan, for her eternal dearness and support for writing this thesis.

## TABLE OF CONTENTS

ABSTRACT .....	v
ÖZ.....	vii
ACKNOWLEDGMENTS .....	x
TABLE OF CONTENTS.....	xi
LIST OF TABLES.....	xiii
LIST OF FIGURES .....	xiv
LIST OF SYMBOLS, ABBREVIATIONS AND SUBSCRIPTS .....	xviii
CHAPTERS	
1 INTRODUCTION .....	1
1.1 Motivation and Problem Definition.....	3
1.2 Literature Survey .....	5
1.3 Thesis Structure.....	7
2 HELICOPTER MATHEMATICAL MODEL .....	9
2.1 Sikorsky UH-60A Black Hawk.....	11
2.2 FLIGHTLAB.....	13
2.2.1 Model Development .....	14
2.2.2 Run-time Simulation.....	24
3 TRAJECTORY GENERATION .....	27
3.1 Trajectory Optimization.....	28
3.2 OptimTraj.....	30
3.2.1 Defining the Problem.....	31
3.3 Simple Example for Trajectory Optimization.....	34

4	TRACKING CONTROLLER DESIGN .....	39
4.1	Inversion Based Controller .....	40
4.1.1	Inner Loop.....	42
4.1.2	Outer Loop.....	46
4.2	LQI Controller.....	47
5	RESULTS AND DISCUSSION.....	51
5.1	Results for Case 1 .....	52
5.2	Results for Case 3 .....	57
5.3	Results for Case 5 .....	62
5.4	Results for Case 7 .....	67
5.5	Results for Case 8.....	72
6	CONCLUSION.....	77
6.1	Future Work .....	79
	REFERENCES .....	81
	APPENDICES	
A.	Linear Model-Nonlinear Model Response Comparison.....	87
B.	Weight Matrices of LQI Controller .....	94
C.	Results for Case 2, Case 4 and Case 6 .....	95

## LIST OF TABLES

### TABLES

Table 2.1 General Characteristics of the UH-60A Black Hawk .....	12
Table 2.2 Performancs Characteristics of the UH-60A Black Hawk .....	12
Table 2.3 Trim Variable and Trim Target Sets for Different Flight Conditions .....	20
Table 3.1 Fields of <code>problem.func</code> and Corresponding Function Handles .....	31
Table 3.2 Fields of <code>problem.bounds</code> and Corresponding Values .....	32
Table 3.3 Fields of <code>problem.guess</code> and Corresponding Values .....	32
Table 4.1 State Selection for Tracking .....	49
Table 5.1 Summary of Investigated Flight Scenarios.....	51

## LIST OF FIGURES

### FIGURES

Figure 1.1. Actions to be taken for possible situations in TRF .....	5
Figure 2.1. Sikorsky UH-60A Black Hawk helicopter during flight.....	11
Figure 2.2. UH-60A Black Hawk pitch bias control system.....	15
Figure 2.3. UH-60A Black Hawk horizontal stabilator control system.....	16
Figure 2.4. TRF response of the helicopter in different level flight trim points .....	17
Figure 2.5. Level flight trims with TRF and nominal conditions.....	18
Figure 2.6. Linear-nonlinear response to collective comparison for level flight ....	22
Figure 2.7. Linear-nonlinear response to longitudinal cyclic comparison for level flight .....	22
Figure 2.8. Linear-nonlinear response to lateral cyclic comparison for level flight	23
Figure 2.9. Linear-nonlinear response to change in rotor speed comparison for level flight .....	23
Figure 2.10. MATLAB-FLIGHTLAB connection diagram .....	25
Figure 3.1. Comparison of the analytical solution and the OptimTraj solution with 3 grids for a simple trajectory optimization problem .....	37
Figure 3.2. Comparison of the analytical solution and the OptimTraj solution with 5 grids for a simple trajectory optimization problem .....	37
Figure 3.3. Comparison of the analytical solution and the OptimTraj solution with 30 grids for a simple trajectory optimization problem .....	38
Figure 4.1. Block diagram of inversion based control system .....	41
Figure 4.2. Block diagram of the command filter .....	42
Figure 4.3. Block diagram of the PID controller .....	43
Figure 4.4. Block diagram of the collective controller .....	46
Figure 4.5. Block diagram of LQI control system.....	47
Figure 5.1. Trajectory of the helicopter for Case 1 .....	53
Figure 5.2. Inertial positions of the helicopter for Case 1 .....	53
Figure 5.3. Euler angles of the helicopter for Case 1 .....	54

Figure 5.4. Body velocities of the helicopter for Case 1 .....	54
Figure 5.5. Air data parameters of the helicopter for Case 1 .....	55
Figure 5.6. Main rotor speed of the helicopter for Case 1 .....	55
Figure 5.7. Linear model transition for Case 1 (0: 80TL, 1: 80L, 2: 80TR).....	56
Figure 5.8. Helicopter inputs for Case 1 .....	56
Figure 5.9. Trajectory of the helicopter for Case 3 .....	58
Figure 5.10. Inertial positions of the helicopter for Case 3.....	58
Figure 5.11. Euler angles of the helicopter for Case 3 .....	59
Figure 5.12. Body velocities of the helicopter for Case 3 .....	59
Figure 5.13. Air data parameters of the helicopter for Case 3 .....	60
Figure 5.14. Main rotor speed of the helicopter for Case 3 .....	60
Figure 5.15. Linear model transition for Case 3 (0: 80TL, 1: 80L, 2: 80TR).....	61
Figure 5.16. Helicopter inputs for Case 3 .....	61
Figure 5.17. Trajectory of the helicopter for Case 5 .....	63
Figure 5.18. Inertial positions of the helicopter for Case 5.....	63
Figure 5.19. Euler angles of the helicopter for Case 5 .....	64
Figure 5.20. Body velocities of the helicopter for Case 5 .....	64
Figure 5.21. Air data parameters of the helicopter for Case 5 .....	65
Figure 5.22. Main rotor speed of the helicopter for Case 5 .....	65
Figure 5.23. Linear model transition for Case 5 (0: 80TL, 1: 80L, 2: 80TR).....	66
Figure 5.24. Helicopter inputs for Case 5 .....	66
Figure 5.25. Trajectory of the helicopter for Case 7 .....	68
Figure 5.26. Inertial positions of the helicopter for Case 7.....	68
Figure 5.27. Euler angles of the helicopter for Case 7 .....	69
Figure 5.28. Body velocities of the helicopter for Case 7 .....	69
Figure 5.29. Air data parameters of the helicopter for Case 7 .....	70
Figure 5.30. Main rotor speed of the helicopter for Case 7 .....	70
Figure 5.31. Linear model transition for Case 7 (1: 60A, 2: 80A) .....	71
Figure 5.32. Helicopter inputs for Case 7 .....	71
Figure 5.33. Trajectory of the helicopter for Case 8 .....	73

Figure 5.34. Inertial positions of the helicopter for Case 8.....	73
Figure 5.35. Euler angles of the helicopter for Case 8 .....	74
Figure 5.36. Body velocities of the helicopter for Case 8.....	74
Figure 5.37. Air data parameters of the helicopter for Case 8 .....	75
Figure 5.38. Main rotor speed of the helicopter for Case 8 .....	75
Figure 5.39. Linear model transition for Case 8 (1: 80A, 2: 80L) .....	76
Figure 5.40. Helicopter inputs for Case 8 .....	76
Figure A.1. Linear-nonlinear response to collective comparison for autorotation at 40 knots .....	87
Figure A.2. Linear-nonlinear response to longitudinal cyclic comparison for autorotation at 40 knots.....	88
Figure A.3. Linear-nonlinear response to lateral cyclic comparison for autorotation at 40 knots .....	88
Figure A.4. Linear-nonlinear response to change in rotor speed comparison for autorotation at 40 knots.....	89
Figure A.5. Linear-nonlinear response to collective comparison for autorotation at 60 knots .....	89
Figure A.6. Linear-nonlinear response to longitudinal cyclic comparison for autorotation at 60 knots.....	90
Figure A.7. Linear-nonlinear response to lateral cyclic comparison for autorotation at 60 knots .....	90
Figure A.8. Linear-nonlinear response to change in rotor speed comparison for autorotation at 60 knots.....	91
Figure A.9. Linear-nonlinear response to collective comparison for autorotation at 80 knots .....	91
Figure A.10. Linear-nonlinear response to longitudinal cyclic comparison for autorotation at 80 knots.....	92
Figure A.11. Linear-nonlinear response to lateral cyclic comparison for autorotation at 80 knots.....	92



Figure A.12. Linear-nonlinear response to change in rotor speed comparison for autorotation at 80 knots.....	93
Figure C.13. Trajectory of the helicopter for Case 2.....	95
Figure C.14. Inertial positions of the helicopter for Case 2.....	96
Figure C.15. Euler angles of the helicopter for Case 2.....	96
Figure C.16. Body velocities of the helicopter for Case 2.....	97
Figure C.17. Air data parameters of the helicopter for Case 2.....	97
Figure C.18. Main rotor speed of the helicopter for Case 2.....	98
Figure C.19. Linear model transition for Case 2 (0: 80TL, 1: 80L, 2: 80TR) .....	98
Figure C.20. Helicopter inputs for Case 2 .....	99
Figure C.21. Trajectory of the helicopter for Case 4.....	99
Figure C.22. Inertial positions of the helicopter for Case 4.....	100
Figure C.23. Euler angles of the helicopter for Case 4.....	100
Figure C.24. Body velocities of the helicopter for Case 4.....	101
Figure C.25. Air data parameters of the helicopter for Case 4.....	101
Figure C.26. Main rotor speed of the helicopter for Case 4.....	102
Figure C.27. Linear model transition for Case 4 (0: 80TL, 1: 80L, 2: 80TR) .....	102
Figure C.28. Helicopter inputs for Case 4 .....	103
Figure C.29. Trajectory of the helicopter for Case 6.....	103
Figure C.30. Inertial positions of the helicopter for Case 6.....	104
Figure C.31. Euler angles of the helicopter for Case 6.....	104
Figure C.32. Body velocities of the helicopter for Case 6.....	105
Figure C.33. Air data parameters of the helicopter for Case 6.....	105
Figure C.34. Main rotor speed of the helicopter for Case 6.....	106
Figure C.35. Linear model transition for Case 6 (1: 40A, 2: 60A) .....	106
Figure C.36. Helicopter inputs for Case 6 .....	107

## LIST OF SYMBOLS, ABBREVIATIONS AND SUBSCRIPTS

### Symbols

$a$	Helicopter total translational acceleration
$a_{x_b}, a_{y_b}, a_{z_b}$	Helicopter translational acceleration on body-fixed frame
$a_{x_i}, a_{y_i}, a_{z_i}$	Helicopter translational acceleration on Earth-fixed frame
$g$	Gravitational acceleration
$J$	Cost function
$p, q, r$	Helicopter angular velocities on body-fixed frame
$u, v, w$	Helicopter translational velocities on body-fixed frame
$v_x, v_y, v_z$	Helicopter translational velocities on Earth-fixed frame
$x_i, y_i, z_i$	Helicopter position on Earth-fixed frame
$x$	Linear model states
$\delta_a$	Lateral cyclic command
$\delta_b$	Longitudinal cyclic command
$\delta_b$	Collective command
$\delta_p$	Pedal command
$\Omega$	Main rotor speed
$\phi, \theta, \psi$	Helicopter Euler angles
$\lambda$	Lagrange multiplier
$\zeta$	Damping ratio of second order transfer function
$\omega_n$	Natural frequency of second order transfer function

## Abbreviations

AoA	Angle of attack
AoS	Angle of sideslip
IAS	Indicated airspeed
IBC	Inversion based controller
LQI	Linear quadratic regulator with integral action
LQR	Linear quadratic regulator
TRF	Tail rotor failure

## Subscripts

$()_C$	Command filter output
$()_D$	Desired value of parameter
$()_{DI}$	Dynamic inverse output
$()_T$	Transformation output
$()_{trim}$	Trim value of parameter
$()_U$	PID output



# CHAPTER 1

## INTRODUCTION

A helicopter is a rotary-wing aircraft that has complex and unstable system dynamics. In spite of the complexity and instability, helicopters are highly preferred in transportation due to their vertical flight ability and agility.

In a conventional helicopter, there are one main rotor and one tail rotor. A tail rotor is a system that provides the necessary anti-torque, especially for low forward speed. A vertical fin of the helicopter can contribute to anti-torque at high forward speed.

The anti-torque is needed because of the torque on the main rotor applied by the engines in order to obey the conservation of the angular momentum law.

In case of an engine failure, the power could be lost completely. In this case, pilots perform the autorotation maneuver which is described as the state of flight with no net power requirement [1]. Autorotation is a descending maneuver in which the main rotor is turned by the action of the air moving through like a windmill. Since the required torque to turn the main rotor is not exerted by the engines, the anti-torque is not necessary for autorotation.

Tail rotor failure (TRF) is another failure condition in which the autorotation is performed by pilots. In this case, the necessary anti-torque cannot be provided by the tail rotor. Therefore, when TRF occurs, autorotation is the emergency procedure if forward speed is not sufficient to balance the torque by the vertical tail.

A helicopter is controlled by changing the pitch angles of the rotor blades. There are three main pilot controls to make this change. The first one is the collective lever which is located on the left side of pilots. Pulling up the collective levitates the swashplate mechanism and increases all main rotor blade angles by the same amount. Thus, the required thrust can be supplied by adjusting the collective. The second pilot control is the cyclic stick. The cyclic stick can travel in longitudinal and lateral directions. Therefore, it is divided into longitudinal cyclic and lateral cyclic. It is

located between legs of pilots. The cyclic tilts the swashplate mechanism to change the main rotor blade angles in different amounts. Hence, the tip path plane of the main rotor tilts and the helicopter can be controlled in pitch and roll directions. The last pilot control is the pedal pair which is located in front of pilots on the cabin floor. Pedals change pitch angles of tail rotor blades in the same amount. Therefore, sometimes it is referred as tail rotor collective [2]. By changing the tail rotor blade pitch angle with pedals, the anti-torque can be adjusted; thus, the helicopter can be controlled on the yaw axis.

In this thesis, optimal trajectory generation and tracking of the generated trajectory for a helicopter in TRF are studied. In case of TRF, pedal input would be lost. Therefore, directional control of the helicopter reduces considerably. In the thesis, the change of main rotor speed is utilized for the directional control. Using the change of rotor speed as a pilot input is possible by the engine throttle. Even though the throttle is not among the main pilot inputs, it is an engine input that can change the exerted torque on the rotor; hence, it can change the rotor speed. Especially for old helicopters, the throttle which was mounted at the end of the collective lever could be controlled by pilots. However, in today's helicopters, this input is controlled by an engine controller. Thus, the throttle can be a solution for increasing or decreasing the rotor speed for engine-powered flights. Although throttle can be used to increase the rotor speed in autorotation, it cannot be used to decrease the speed. For this case, a mechanism that decreases the kinetic energy of the rotor must be suggested. Applying collective would increase drag force on the blades while it increases the lift. However, it cannot be a good candidate because low collective and low rotor speed could be required at the same time in a trajectory. After spending some time on this problem, a rotor brake is proposed to decrease the rotor speed in autorotation. In fact, rotor brake is used in helicopters but after the helicopter lands and the engines are shut down. In this study, the existence of a sophisticated rotor brake is assumed to decrease the rotor speed when it is necessary.

In order to generate flight trajectories, an open-source trajectory optimization tool, which is called as OptimTraj, is used. It is very user-friendly and a trajectory

optimization problem can be solved by the direct collocation method by defining an objective function, system dynamics, path constraints, and boundary constraints [3]. In order to reduce the computation time, this work utilizes linear helicopter models with interpolation for the system dynamics. The linear helicopter models are created by linearizing the generic model in FLIGHTLAB around the trim points in level flight, in 20-degree left and right bank turns at 80 knots, and in autorotation at 40, 60 and 80-knot conditions. The trajectories obtained for the helicopter to fly a specified position in space for the cases that the initial trim is level flight and to change its trim condition for the cases that the initial trim is autorotation.

Next, in order to follow the generated trajectories, tracking controllers are developed. An inversion based controller is designed in order for position tracking and an LQI controller is designed in order for state tracking. The controllers are tested with linear and nonlinear model simulations. For nonlinear model simulations, real-time feature of FLIGHTLAB is utilized by communicating with SIMULINK. The helicopter response to TRF is not included in the trajectory generation and tracking process in this study.

## **1.1 Motivation and Problem Definition**

In case of a mechanical, structural, or hydraulic failure on pedal input, the control in the yaw axis could be lost. This study assumes any of these possibilities could be the cause of a TRF. Commercial helicopters suffered 470 accidents because of a TRF between 1963 and 1997 [4]. FAA states that TRF is the third major cause of fatal helicopter accidents between 1996 and 2007 [5].

When TRF occurs, the recommended emergency procedure is to go into autorotation and land the helicopter in a safe zone as soon as possible. In this study, however, it is assumed that when TRF occurs, the pilot cannot perform autorotation for landing due to lack of an available landing area. Such a situation is investigated for two different possibilities: TRF in high forward velocity that vertical fin is sufficient to provide required anti-torque, and, TRF in low forward speed.

For TRF in high forward velocity, the sideslip angle would be much higher so that the vertical fin can create sufficient moment. Although the pedal control is absent, the directional control can be satisfied with the lateral cyclic due to highly coupled dynamics and with the change in rotor speed. Hence, the helicopter might succeed in arriving at a specified location.

For the case of a TRF at low speed, there are two options: the pilot goes into autorotation in order to eliminate the need for anti-torque and accelerates to stay in level flight, or, the helicopter would spin around while maintaining its altitude. This work does not cover the spinning option. On the other hand, the transition from autorotation to level flight is studied in the same manner as bringing the helicopter to a specific location at a high forward speed. Both cases are basically trajectory optimization problems that aim to change the state of the helicopter from an initial condition to a terminal condition by minimizing defined cost function while satisfying some constraints.

Although there are many studies for trajectory generation, autorotation, path following, and TRF in helicopters, a study that covers all these subjects has not been come across within the literature. In this thesis, it is proposed that a helicopter in TRF does not have to autorotate when landing is not safe. It can fly to a certain position where autorotation could be performed safely. The case that brings the helicopter from autorotation to level flight is also studied. These scenarios are depicted in Figure 1.1. In addition, in the absence of pedal input, the change of main rotor speed is proposed in order for the directional control of the helicopter in TRF. These are the main contributions of this thesis.

Studying this subject might be helpful for predicting the abilities of a helicopter in TRF beforehand and to reduce pilot workload after the failure. Thus, this subject might have a role in designing helicopters to achieve some requirements for TRF.



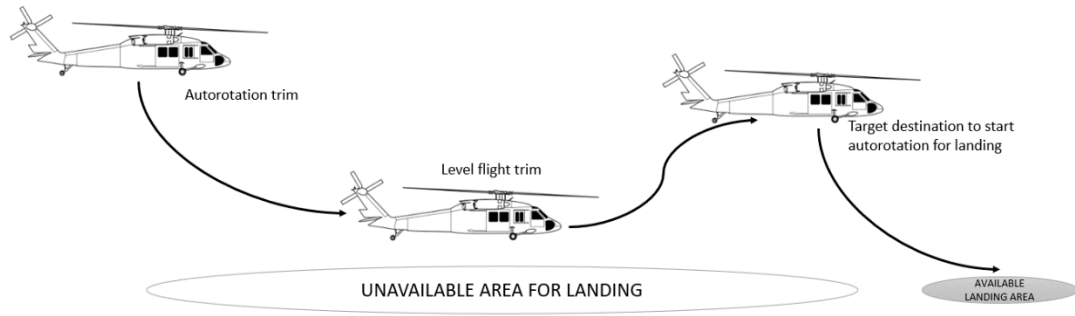


Figure 1.1. Actions to be taken for possible situations in TRF

## 1.2 Literature Survey

Yomchinda et al. [6] performed autorotation path planning for an unmanned helicopter in engine failure. The landing location is pre-specified and the computed trajectory is followed until the touchdown. The inner loop flight controller uses dynamic inversion and model following control to stabilize the vehicle. An outer loop trajectory following controller uses a combination of feed-forward control inputs and a PID controller to reduce path error. Flightlab is utilized for simulations.

Chowdhary et al. [7] managed to guide a fixed-wing UAV under actuator failure and structural damage. The flight path is defined by waypoints. The commands are used by an outer-loop linear controller to generate feasible attitude commands. The inner-loop attitude control is achieved by using either a linear attitude controller or a neural network-based model reference adaptive controller. GT Twinstar UAV is used for flight tests.

Bottaso et al. [8] performed planning and tracking of trajectories for the simulation of maneuvers with multibody models. A maneuver is defined as an optimal control problem. This definition feeds into a planning layer that determines the vehicle trajectory by solving the optimization problem for a reduced model. The resulting trajectory feeds into a control layer that tracks the computed optimal path using a nonlinear model predictive formulation and steers the multibody model.

Garcia et al. [9] presented a fuzzy controller capable of waypoint navigation under tail rotor failure on a small unmanned helicopter. In the study, the helicopter is navigated while it is spinning due to anti-torque. Both loss of tail rotor and tail actuator lockout cases are investigated. For simulations, X-Plane is used.

Budiyono and Wibowo [10] designed an optimal tracking controller for the R-50 helicopter model to follow predefined trajectories. LQR method is utilized with the presence of input constraints for path tracking.

Abbeel et al. [11] presented the first autonomous controller to successfully pilot an RC helicopter during an autorotation descent and landing. Differential dynamic programming is used to design the controller. The flight tests are conducted with the XCell Tempest helicopter.

Enns and Si [12] studied the reconfiguration of the helicopter flight control for main actuator failures. Requisite upper controls geometry is developed for RFC to have significant cross-coupling. The paper then described several reconfiguration strategies used to form the RFC system, including a swashplate reconfiguration strategy that controls any two of the three main rotor control axes for any main rotor actuator failure.

Rogers et al. [13] developed tau-based controllers for autorotation in order to guide a pilot with visual aids to apply required control inputs. In the study, a tau-based trajectory generation scheme has been formulated to produce a forward speed profile to be tracked during the flare for landing at a desired touchdown point. Pilot-in-the-loop simulations are performed with HELIFLIGHT-R simulator.

Bibik and Narkiewicz [14] developed an efficient method to control the helicopter after power failure, which would cover both emergency landings and flyaways (for OEI). The method developed is based on linear-quadratic control, modified to allow variation of both the performance index and the constraints during the flight. The helicopter control is performed autonomously, initiated with a prescribed time delay after power failure.

Rajendran and Gu [15] designed a multi-loop PID controller to control the helicopter when the tail rotor control stuck occurs in hovering flight. Robustness in terms of stabilizing ability is analyzed and the fault tolerant control scheme is tested with nonlinear model simulations. ADS-33PRF level 2 requirements could be satisfied.

Serrano et al. [16] presented a trajectory tracking control for a UAV helicopter under environmental disturbances. In the study, a pre-established trajectory is followed by the helicopter with a linear algebra based controller. The controller parameters are tuned by Monte Carlo simulations to find the minimum value for a trajectory error cost function.

Taamallah et al. [17] presented the first real-time feasible model-based trajectory planning and trajectory tracking system for a small-scale helicopter UAV in autorotation. A high fidelity nonlinear model is used. In addition, the study includes a very comprehensive literature introduction.

Wang et al. [18] obtained safe landing trajectories formulating a nonlinear optimal control problem based on a nonlinear helicopter model and geometry constraints due to the MR actuator stuck. The direct method is used to solve the optimal control problem. UH-60 model is used for simulations.

Lee et al. [19] applied reinforcement learning algorithm for autorotation of an unmanned helicopter. The proposed method is evaluated by simulations based on a point-mass model of a modified OH-58A helicopter.

### **1.3 Thesis Structure**

This thesis contains six chapters. The first chapter gives an overview of helicopters and their control. Next, the motivation of the thesis is given and the problem of interest is defined. Finally, a literature survey about helicopter trajectory generation and tracking is presented.

The second chapter includes helicopter modeling and helicopter models used in the past and the present. Next, the UH-60A Black Hawk helicopter and FLIGHTLAB are introduced. Moreover, model development for the study and the infrastructure for run-time simulations are explained. Initial analyses performed by the model are presented.

The third chapter details the trajectory generation process. After an optimization problem is formulated, an open-source program to solve it is introduced. Next, the optimization problem which is studied in this thesis is defined. Lastly, a simple trajectory optimization problem is solved by the program to compare with the analytical results.

The fourth chapter explains tracking controller development. Initially, an inversion based controller is introduced with inner and outer loops separately. Then, the LQI controller is detailed.

The fifth chapter contains results and discussion part for the investigated scenarios. The generated trajectories and the tracking simulations with linear and nonlinear models are presented for several helicopter parameters.

The final chapter gives the conclusion of the thesis and enlightens the path for future work.

## CHAPTER 2

### HELICOPTER MATHEMATICAL MODEL

Mathematical modeling is the process of describing the motion of a dynamical system in terms of mathematical equations using Newton's second law of motion. Mathematical models are used for predicting the response of the system to any initial condition or any external input. For complex systems, such as aircrafts, the mathematical model is composed of submodels of subsystems. The dynamical relations in a system are generally represented by differential equations. It is almost impossible to find an analytical solution for these equations for an aircraft. Therefore, numerical methods are implemented to find a solution. However, solving differential equations for an aircraft numerically could be complicated and would require high computing power. For this purpose, in addition to the mathematical model, computer programs which are called comprehensive analysis tools are developed to solve these equations. Comprehensive analysis tools may be composed of multidisciplinary programs that requires knowledge from different engineering areas of expertise. These tools could perform trim, linearization, dynamic simulation, noise prediction, analyses of aeroelasticity, structural load, vibration, and aerodynamics.

Flight dynamics models are very useful in both design phase and testing phase of an aircraft. Before the real aircraft is built, its flight characteristics can be predicted; moreover, experienced pilots can give feedbacks about the aircraft by performing real-time simulations in the design process so that design decisions can be made accordingly. Flight dynamics models are also used for simulations for the cases that conducting a flight test is too risky or too expensive such as emergency scenarios like engine failure, TRF, MR actuator malfunction, etc. Thanks to the mathematical model, likewise, the autopilot can be designed and tested without integrating into the real aircraft. This also reduces the design effort and the cost of production for autopilot.

The ability to predict helicopter behavior in a cheap, riskless and quick way leads the sector companies to invest and develop helicopter flight dynamics models [20] [21] [22] [23] [24]. The minimum complexity model is one of the first successful attempts to model helicopter dynamics [25]. This model represents basic helicopter dynamics including rotor flapping, inflow and the rigid body dynamics so that computational load is lightened for real-time applications. Although the minimum complexity model is very old, it still can be used for basic flight mechanics analyses especially for academic purposes. As a result of a need for high fidelity modeling of UH-60A Black Hawk and scientific research, GenHel is come out in the 1980s by Sikorsky Aircraft Company and NASA Ames Research Center. The model continued to be developed over the years and it is validated [26]. Comprehensive Analytical Model of Rotorcraft Aerodynamics and Dynamics (CAMRAD II) is another widely used helicopter mathematical modeling tool in the industry which is developed by Johnson Aeronautics. Although it is capable of analyzing performance, time response and stability characteristics, it is mainly preferred for structural dynamics, vibration and load analyses [21]. Advanced Rotorcraft Technologies Inc. has been developing a comprehensive helicopter simulation tool named FLIGHTLAB since 1982 [27]. FLIGHTLAB is a real-time blade element rotorcraft simulation tool that has viscous vortex particle method and finite state dynamic wake capabilities [28] [29].

Although there are many helicopter flight dynamics modeling tools, in this thesis, the generic model in FLIGHTLAB is preferred due to fidelity level and availability. Additionally, its run-time application feature and interfacing with other programs have an effect on making this decision. The generic model is an example model for users to reference for starting to model a helicopter. Its design parameters are very close to UH-60A Black Hawk helicopter. Therefore, it can be assumed that this study is conducted for the UH-60A Black Hawk helicopter.

## 2.1 Sikorsky UH-60A Black Hawk

Sikorsky UH-60A Black Hawk is a four-blade, twin-engine, medium-weight utility helicopter manufactured by Sikorsky Aircraft. The UH-60A Black Hawk and its variants are widely used in military services in several countries for over four decades.

The UH-60A Black Hawk is a conventional utility helicopter having four-blade main and tail rotors. It has an articulated main rotor using elastomeric bearings and its tail rotor is canted by 20 degrees. Its horizontal tail is moveable in order to increase the stability for different flight conditions. It is powered by two General Electric T700 turboshaft engines and it can carry 11 people. Its general characteristics and performance characteristics are shown in Table 2.1 and Table 2.2.



Figure 2.1. Sikorsky UH-60A Black Hawk helicopter during flight

Table 2.1 General Characteristics of the UH-60A Black Hawk

General Characteristics	
Crew	2 pilots + 2 crew
Capacity	2640 lb of cargo internally
Fuselage length	50 ft 7.5 in
Height including TR	16 ft 10 in
Fuselage width	7 ft 9 in
Empty weight	12511 lb
Maximum takeoff weight	2200 lb
MR diameter	53 ft 8 in

Table 2.2 Performancs Characteristics of the UH-60A Black Hawk

Performancs Characteristics	
Maximum speed	159 kt
Cruise speed	152 kt
Range	1199 nmi
Servise ceiling	19000 ft
Rate of climb	1646 ft/min



## 2.2 FLIGHTLAB

FLIGHTLAB is a commercial software for modeling the flight dynamics of an aircraft. It is mainly used in rotorcraft industry; however, it allows to construct fixed-wing models too. Using the dynamic components such as hinge, damper, spring, etc. in its component library, whole aircraft can be modeled with the desired level of fidelity. Physics based model which is brought together by the dynamic components makes easier to find out mistakes in the model. Moreover, solution methods are defined as a component as well and they can be selected by the user for each dynamic component.

FLIGHTLAB consists of two products: Development System and Run-time System. The Development System is used for creating the flight dynamics model of the aircraft and performing the flight dynamics analyses. The Development System can be also used to generate run-time models for real time applications. The Development System toolbox has different modules. FLIGHTLAB Modeling Editor (FLME) is a graphical interface for creating the aircraft model. By providing the pre-allocated values for parameters of components, an aircraft can be built easily without diving into coding to connect each aircraft component. Different methods for rotor modeling and desired fidelity can be defined by the user. Control System Graphical Editor (CSGE) is a graphical interface for modeling the flight control system of the aircraft. CSGE allows the control system to be represented by the block diagrams. For flight dynamics analyses, Analysis Workspace & Utilities (Xanalysis) module is provided. Xanalysis is a graphical interface for performing trim analyses, linear and nonlinear time and frequency response analyses, and, linearization of nonlinear model. In addition, FLIGHTLAB uses the programming language, SCOPE, in which users can produce their own scripts for modeling and analysis.

On the other hand, the Run-time System is used for real-time applications. The model generated by the Development System can be executed. Input and output signals from external devices are communicated and a driver called SIMPHONY can simulate the response of the aircraft in real-time.

Modeling a helicopter by writing kinematical and dynamical relations is not in the scope of this thesis. In this study, the generic model in FLIGHTLAB is used. In order to represent the faulty tail rotor and also to reduce the complexity, some modifications are made in the generic model. The Development System is utilized for making these modifications, performing flight dynamics analyses and creating the run-time model. The Run-time System is employed for real-time simulations.

### **2.2.1 Model Development**

The generic model in FLIGHTLAB is a very representative model of UH-60A Black Hawk helicopter. The model can demonstrate basic dynamical behaviour of the Black Hawk helicopter.

The main rotor of the model has 4 blades rotating counter-clockwise direction with 27 rad/sec and its radius is 27 feet. There is a 3-degree forward tilt at its shaft. The main rotor is modeled with flapping and lead-lag dynamics. Elastic behaviour of the blades is not concerned in the model. For the inflow modeling, Peters/He three state methodology is selected. The engine is modeled as an ideal engine which implies that rotor speed would not be affected by the pilot controls and the flight conditions.

For the tail rotor, Bailey rotor model is implemented. The rotor has 4 blades and its radius is 5.5 ft. Bailey rotor model does not have aerodynamic tables. Instead of tables, effective drag coefficients and lift curve slope value can generate aerodynamic force and moment. In addition, the cant angle is 20 degrees.

In the default settings of the generic model, several stability enhancement systems, which are control mixing, pitch bias controller, SAS and movable horizontal tail, exist as in the UH-60A Black Hawk [30]. Briefly, control mixing is the mechanical mixing of pilot controls. In the Black Hawk, pedal tilts the main rotor swashplate in longitudinal direction, whereas collective tilts the swashplate in both directions. Furthermore, collective has an effect on the tail rotor as well. Next, pitch bias controller is added to enhance longitudinal static stability. It feeds back longitudinal

airspeed, pitch angle and pitch rate and drives the longitudinal cyclic as shown in Figure 2.2 [30]. SAS is the stability augmentation system that exist in almost all helicopters and airplanes. Its goal is to terminate the rates on the aircraft and, hence, to provide a more stable flight. Moveable horizontal tail generates the tail force in necessary amount and necessary direction by adjusting its incidence angle. In the Black Hawk the incidence is adjusted according to pitch rate, lateral acceleration, longitudinal airspeed, and collective input as shown in Figure 2.3 [30]. However, in order to reduce complexity and to develop a new controller suitable for the purpose of the thesis, those systems are deactivated.

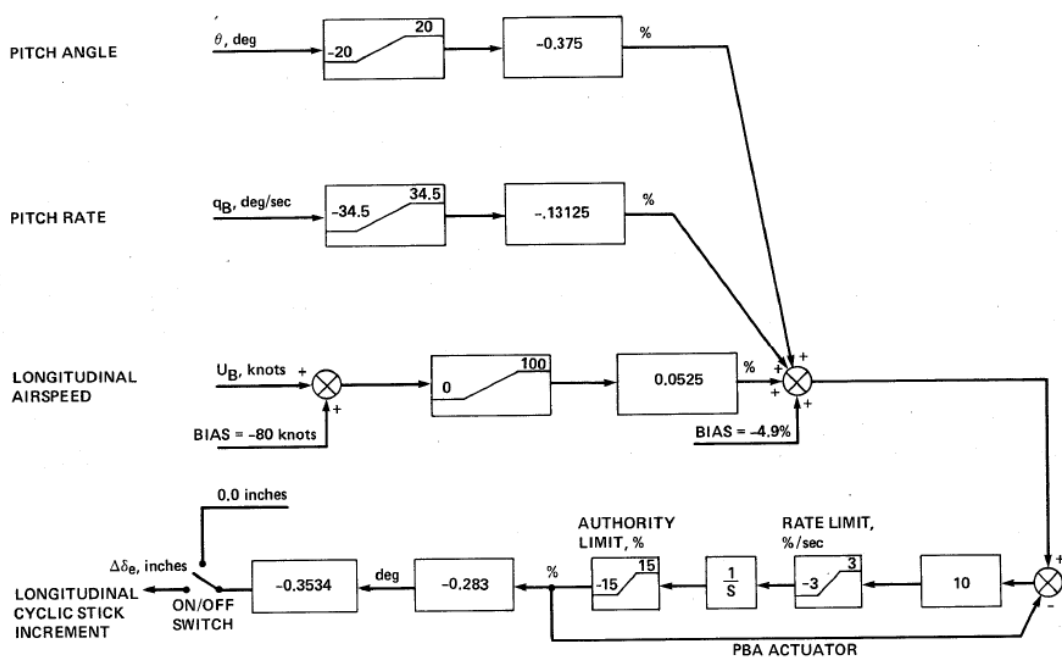


Figure 2.2. UH-60A Black Hawk pitch bias control system

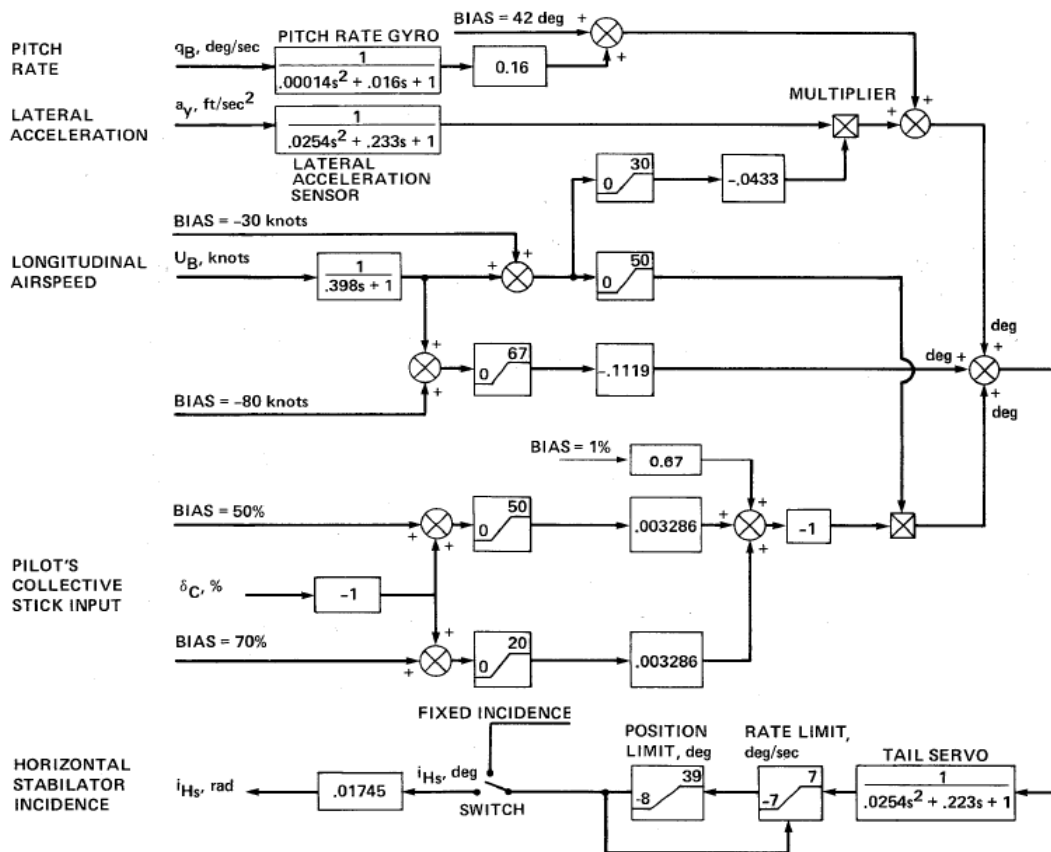


Figure 2.3. UH-60A Black Hawk horizontal stabilator control system

In this thesis, a trajectory is generated and is followed for a helicopter in tail rotor failure. For this purpose, tail rotor failure is modeled. In the generic model in FLIGHTLAB, there is a tail rotor failure switch which terminates the force and moment produced by the tail rotor. The switch is turned on in order to get the tail rotor in failure condition.

After making the modifications, the model is ready for initial analyses. Firstly, the effect of TRF switch is experimented with level flight for different forward speeds. As can be noticed in Figure 2.4, the helicopter gains rates for all directions after the failure is injected at  $t = 1$  second; however, the yaw response is more violent than others as expected. Another observation of this experiment is that as the airspeed

increases the severity of TRF response decreases. This happens because the vertical fin can take the load of the tail rotor as airspeed increases.

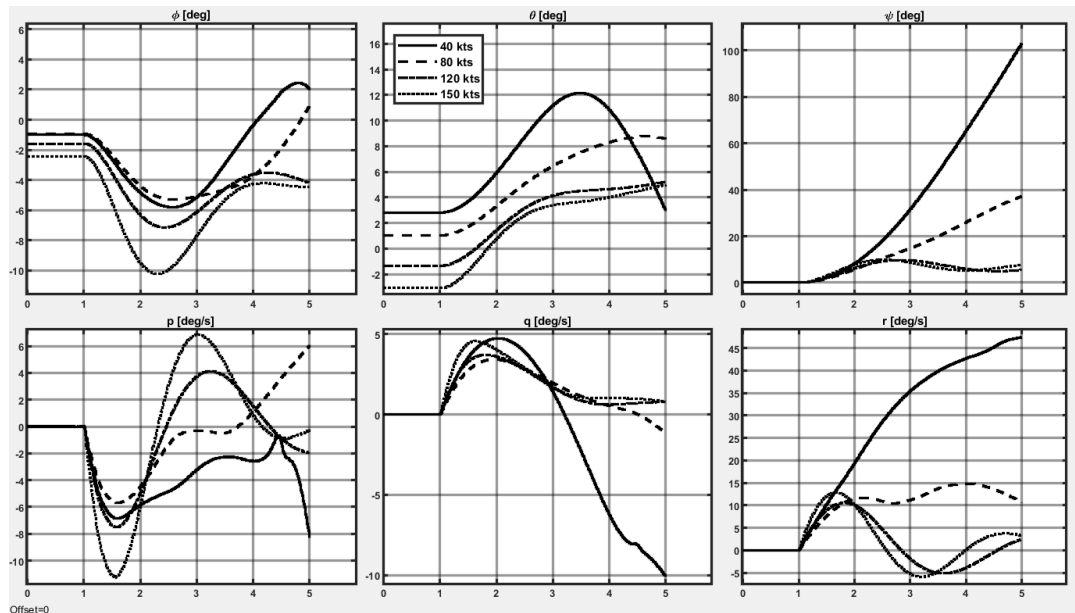


Figure 2.4. TRF response of the helicopter in different level flight trim points

Secondly, the characteristics of the helicopter are investigated by trim analyses using FLIGHTLAB. Trim variables and trim targets must be determined in order to identify the flight condition. Trim variable and trim target sets are summarized in Table 2.3. For level flight trim, trim targets are set as derivatives of body velocities and body rates whereas trim variables can be set as collective, longitudinal cyclic, lateral cyclic, pedal, roll angle and pitch angle. However, pedal fixed trims are required in order to compare the effect of TRF with the normal condition. Therefore, the pedal is omitted from the trim variable set and sideslip is added to the set. Figure 2.5 shows that the tail rotor is more dominant over the vertical fin below 80 knots and as the airspeed increases, the need for sideslip angle in magnitude reduces. For TRF condition, level flight trim is not achieved below 77 knots which implies that the helicopter would spin if it is in TRF.

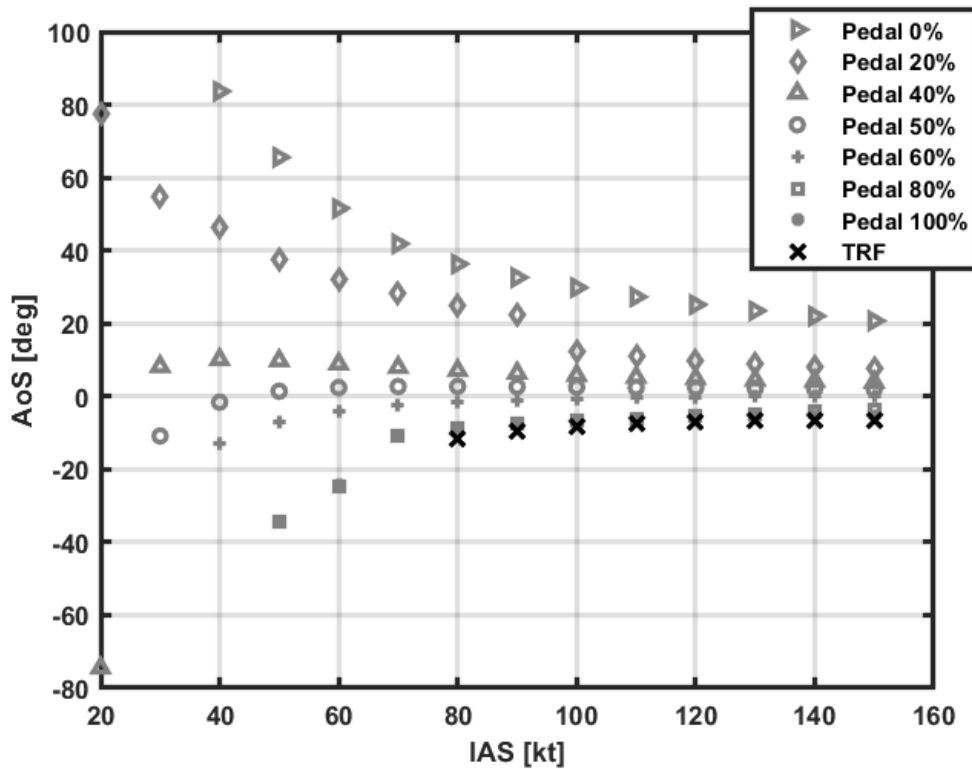


Figure 2.5. Level flight trims with TRF and nominal conditions

Since the helicopter in TRF can stay in level flight at 80 knots, the linear model for this trim condition is chosen for the flight path generation. In addition, in order to reach the specified final point, the helicopter may have to maneuver. However, path generation using the linear model of 80 knots level flight alone will not be realistic for the cases that the helicopter needs to maneuver. For such a case, even if a solution is found with the OptimTraj, nonlinear simulation results would reveal the trouble. In linear simulations, this problem would not be apparent. Thus, the linear models for left and right coordinated turns with 20 degrees bank angle are included. In both trajectory generation and trajectory tracking processes, the linear model whose states are between two linear model's equilibrium points is found by interpolating those two linear models.

For the purpose of formation of coordinated turn models, the helicopter is trimmed at that flight condition as previously done for level flight model. By freezing the roll

angle as 20 degrees for right turn and -20 degrees for left turn, the trim variables are set as collective, longitudinal cyclic, lateral cyclic, pitch angle, sideslip angle and body rates. On the other hand, trim targets are set as derivatives of body velocities, that of body rates and that of roll and pitch angles.

It is previously mentioned that below 77 knots airspeed, the vertical fin is not sufficient to hold the helicopter in trim position in level flight. Therefore, the helicopter continues to spinning unless the rotor torque is not terminated by autorotation maneuver. Hence, in order to find the trajectory that brings the helicopter from autorotation to level flight, autorotation trim analyses are performed. In this case, trim variables are set as collective, longitudinal cyclic, lateral cyclic, pitch angle, roll angle, sideslip angle and flight path angle. Next, trim targets are set as derivatives of body velocities, derivatives of body rates and specific horsepower. Floating flight path angle and targeting zero horsepower allows helicopter to descend without any power consumption.

Although the spinning case is not covered in this study, the trim condition is investigated. For this scenario, trim variables are set as collective, longitudinal cyclic, lateral cyclic, pitch angle, roll angle and body rates. On the other hand, trim targets are set as derivatives of inertial velocities, derivatives of body rates and derivatives of roll and pitch angles. Hence, it is aimed that the helicopter would fly without descending at low speeds. Surprisingly, the helicopter is trimmed at 40 and 60 knots. However, derivative of the yaw angle is 131 deg/s at 60 knots and 175 deg/s at 40 knots. Assuming pilots sit 10 feet away from the helicopter center of gravity, they would be exposed to about 3.5 g in the longitudinal direction. This is a harsh condition for pilots as expected.

Table 2.3 Trim Variable and Trim Target Sets for Different Flight Conditions

Flight Condition	<i>Trim Variables</i>	<i>Trim Targets</i>
Level flight	$\delta_c, \delta_b, \delta_a, \phi, \theta, \beta$	$\dot{u}, \dot{v}, \dot{w}, \dot{p}, \dot{q}, \dot{r}$
Coordinated Turn	$\delta_c, \delta_b, \delta_a, \theta, \beta, p, q, r$	$\dot{u}, \dot{v}, \dot{w}, \dot{p}, \dot{q}, \dot{r}, \dot{\phi}, \dot{\theta}$
Autorotation	$\delta_c, \delta_b, \delta_a, \phi, \theta, \beta, \gamma$	$\dot{u}, \dot{v}, \dot{w}, \dot{p}, \dot{q}, \dot{r}, P$
Spin	$\delta_c, \delta_b, \delta_a, \phi, \theta, p, q, r$	$\dot{v}_x, \dot{v}_y, \dot{v}_z, \dot{p}, \dot{q}, \dot{r}, \dot{\phi}, \dot{\theta}$

After the equilibrium points are found, linearization is applied to those points in order to generate linear models. For linearization, FLIGHTLAB requires the selection of input-state-output sets. In this study, instead of pedal, main rotor speed is utilized for directional control of the helicopter in TRF. FLIGHTLAB allows to select the derivative of the main rotor speed in the input set. Besides, in addition to standard body states for describing the motion of a rigid body, main rotor speed is included to the state set in order to observe its effect on the helicopter. Hence, linear models for level flight, right and left coordinated turn with 20 degrees bank angle at 80 knots, and, for autorotation at 40, 60 and 80 knots are generated. These models are obtained as in state-space form. Input-state-output sets are given in (2.1). It should be noted that although the pedal is included in the input set, the corresponding column in the control matrices of all linear models is zero because TRF switch disables its effect on the helicopter.

$$\begin{aligned}
 u &= [\delta_c, \delta_b, \delta_a, \delta_p, \dot{\Omega}]^T \\
 x &= [\Omega, \phi, \theta, u, v, w, p, q, r]^T \\
 y &= [\phi, \theta, \psi, u, v, w, p, q, r, \Omega]^T
 \end{aligned} \tag{2.1}$$

Finally, nonlinear and linear responses to all inputs are compared in order to check that the linear models truly represent the nonlinear behavior of the helicopter.



According to the linear theory, it is assumed that the nonlinear model would respond in the same manner as the linear model responds if the states are in the neighborhood of the equilibrium point. Example comparisons are given for level flight at 80 knots for all inputs from Figure 2.6 to Figure 2.9. It should be noted that these simulations are performed by applying a step input at  $t = 1$  second as much as 1 % for the pilot controls and  $1 \text{ rad/s}^2$  for the derivative of rotor speed. It is clearly seen that for 1 % change in pilot controls, the linear model would behave like the nonlinear model which implies that the states would remain in the neighborhood. Conversely, for  $1 \text{ rad/s}^2$  change in the derivative of the rotor speed, the models diverge from each other. This might happen if the input is too large for the states to stay in the neighborhood of the equilibrium point. Furthermore, TRF condition might narrow the neighborhood at that airspeed because it is known that the helicopter cannot stay in level flight below 77 knots and its characteristics suddenly change. Therefore, this result gives an idea about the sensitivity of the derivative of rotor speed. Since the responses are in the same manner at the beginning of the input, it is concluded that the linear model sufficiently represents the nonlinear helicopter model. Likewise, the comparisons for other models which can be seen in Appendix A are analyzed and the same conclusion is made for all linear models.

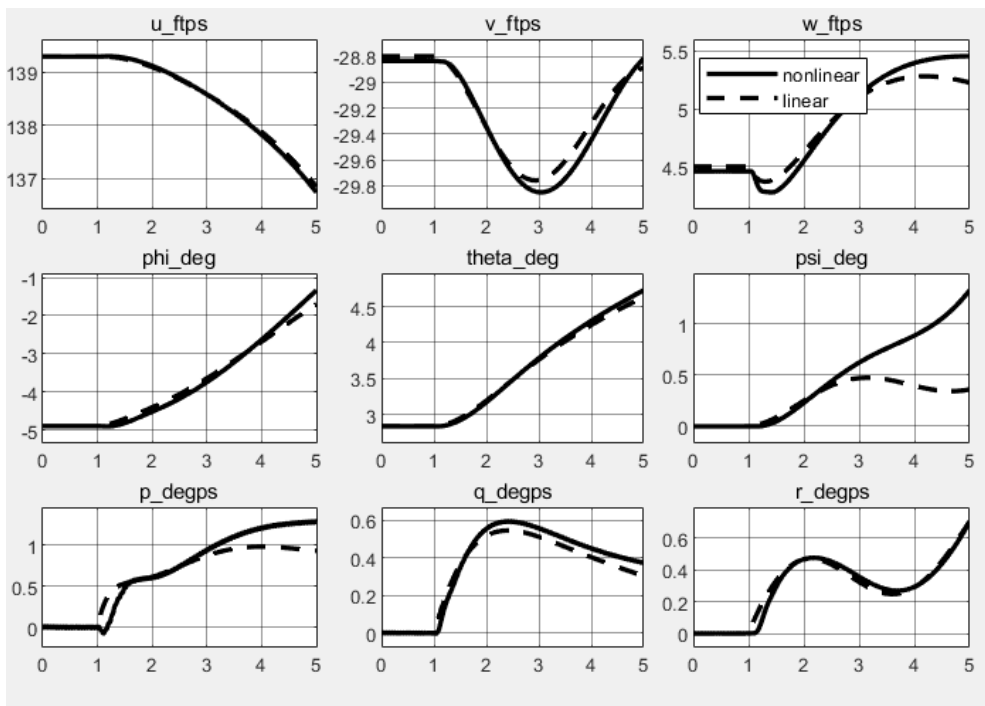


Figure 2.6. Linear-nonlinear response to collective comparison for level flight

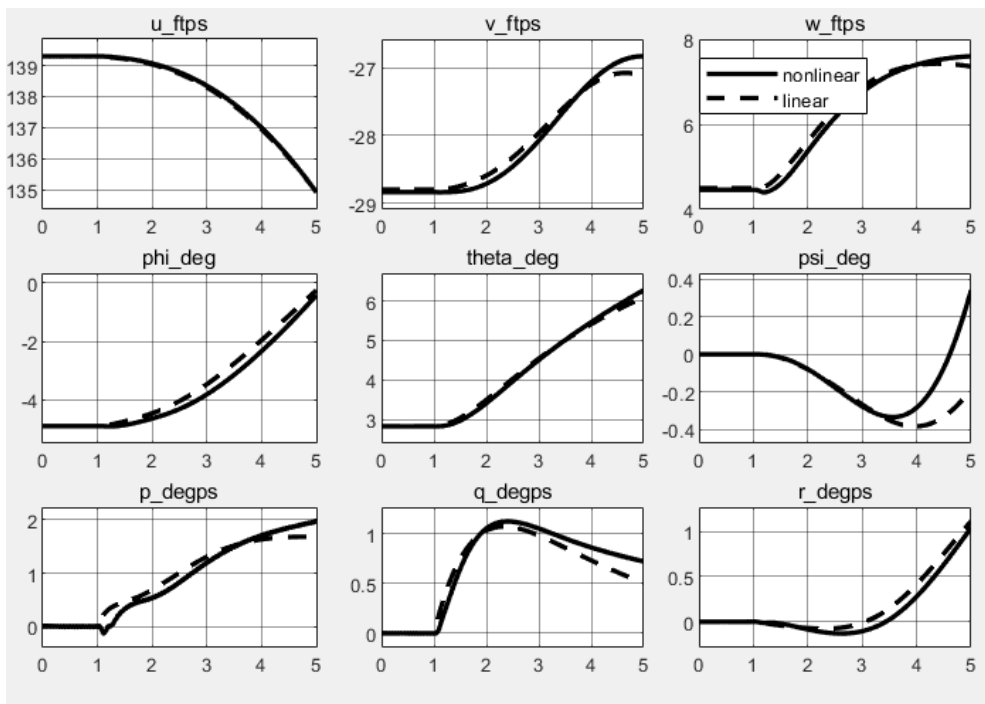


Figure 2.7. Linear-nonlinear response to longitudinal cyclic comparison for level flight

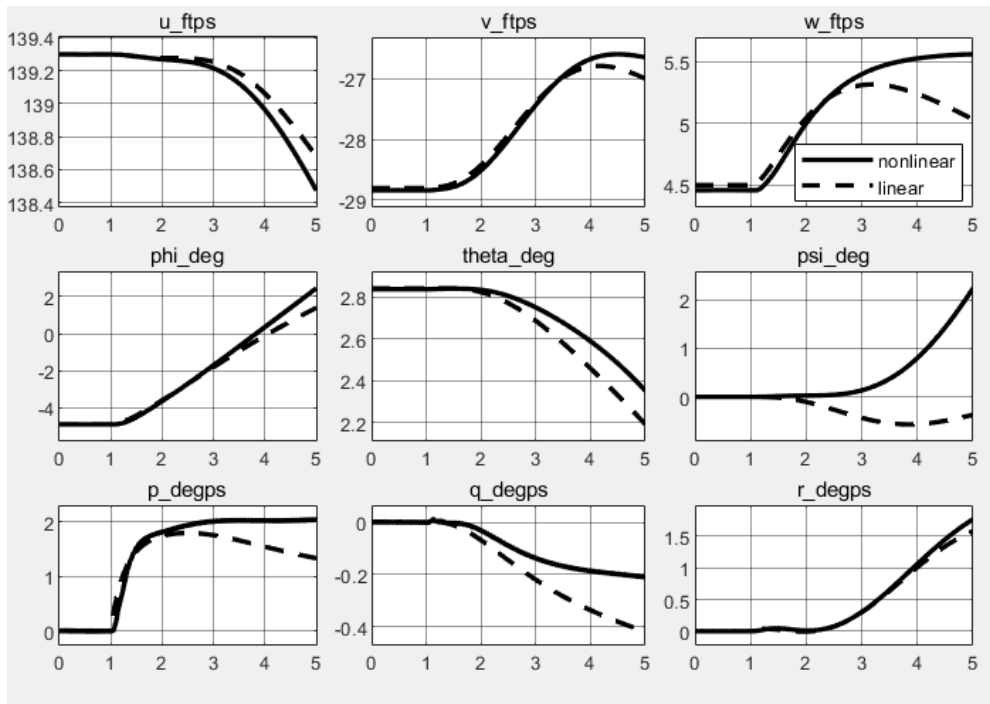


Figure 2.8. Linear-nonlinear response to lateral cyclic comparison for level flight

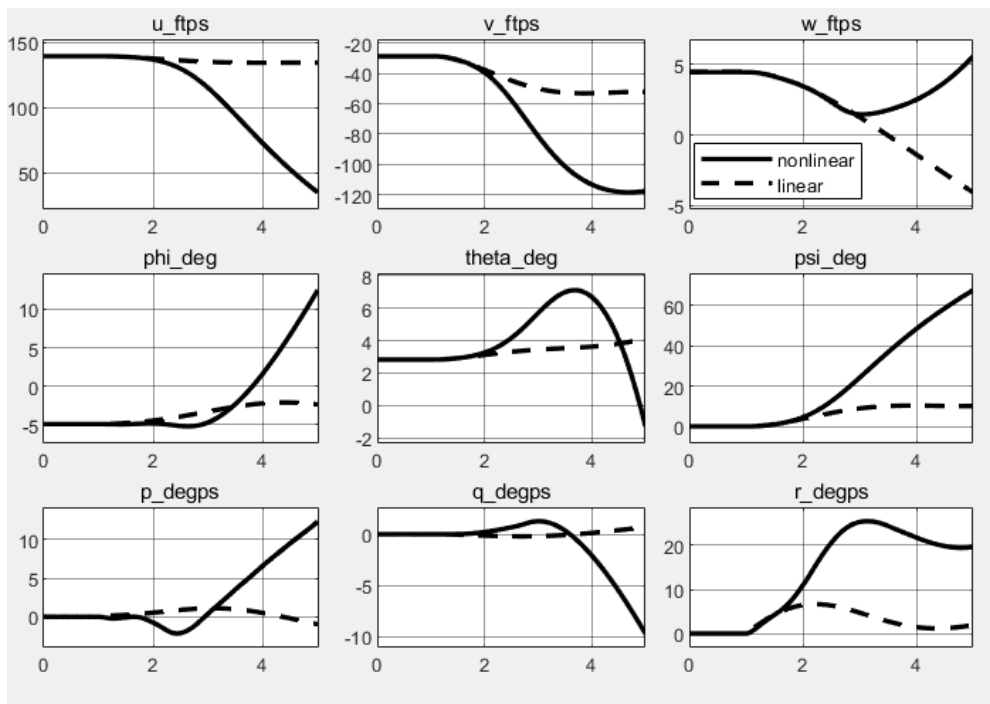


Figure 2.9. Linear-nonlinear response to change in rotor speed comparison for level flight

### **2.2.2 Run-time Simulation**

As mentioned previously, the Run-time System allows users to perform real-time applications. The generated black-box model can be interfaced with external systems such as pilot control loaders, cockpit buttons, screens, etc. This black-box model is called as FCM model.

In this work, the controller is developed in SIMULINK environment. Therefore, the Run-time System is utilized in order to communicate the model with the controller. It includes a graphical user interface; however, it can interact with a host program through the FLIGHTLAB Communication Utilities (FLCOMMS).

The FCM model is driven by SIMPHONY module. A C-script is prepared for driving the FCM model and for input-output signal regulation. It transfers the data to the shared memory of SIMPHONY. UDP communication protocol is implemented in a C-script and IP/Port numbers are provided according to IP/Port numbers in UDPsend and UDPreceive blocks in SIMULINK.

MATLAB-FLIGHTLAB connection is simply schematized in Figure 2.10. An input signal is sent from the Plant to the FCM model for a time instant and it starts pending for the FCM model outputs. SIMPHONY drives the model according to the received input signal and the model outputs are computed. After the outputs are received by the Plant, SIMULINK starts operating. The model outputs are fed back to the Controller and new inputs are calculated for the next time instant.

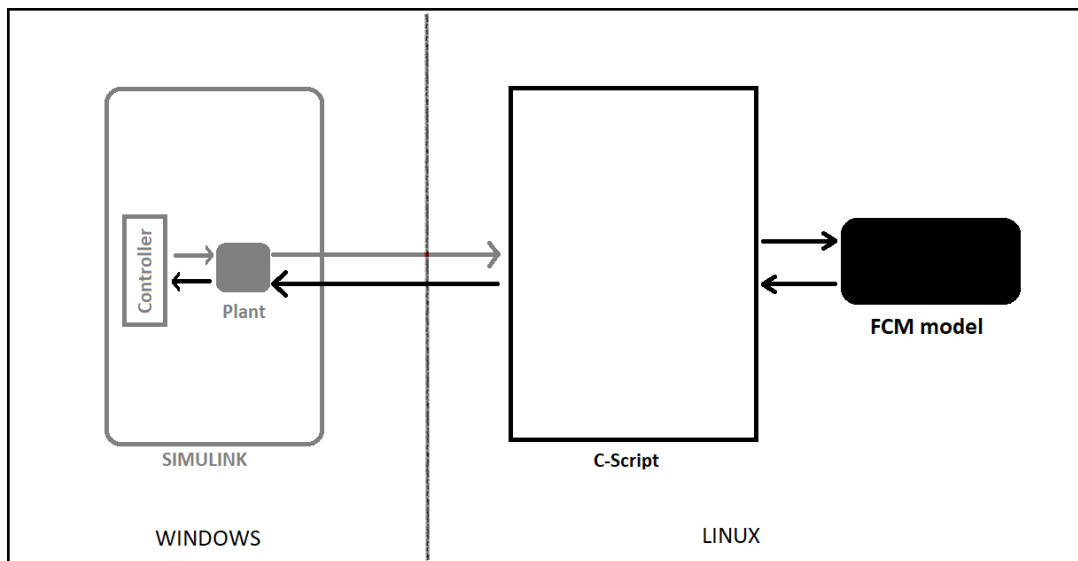


Figure 2.10. MATLAB-FLIGHTLAB connection diagram



## CHAPTER 3

### TRAJECTORY GENERATION

In this study, the optimal trajectories are investigated for a helicopter in TRF. Since the helicopter of interest is lack of pedal control, it cannot perform any maneuver commanded. Its system dynamics are pretty different from an ordinary helicopter. Therefore, instead of making a reasonable guess, trajectory generation is performed in order to find the flight path that the helicopter in TRF actually can fly on. In order to find a feasible trajectory, the problem is implemented as an optimization problem.

Solving a trajectory optimization problem analytically for a helicopter is very difficult. The system dynamics might include nonlinear terms and many states so that its complex character can be represented well. Therefore, numerical methods are implemented for solving optimization problems.

There are many commercial or open-source programs in order to solve optimization problems [31] [32] [33] [34]. In this work, an open-source trajectory optimization tool, called OptimTraj, is utilized to solve the problem [3]. OptimTraj is a MATLAB library in which many function scripts, helpful demo problems and an understandable manual exist.

OptimTraj requires system dynamics equations in order to integrate the states through the collocation grid points. Upper and lower bounds for time, state and control can be introduced. After defining equality and inequality constraints and, of course, objective function, the problem can be solved, if a feasible solution exists.

In this chapter, the general formula of a trajectory optimization problem and solution methodology are presented. Next, OptimTraj is introduced and how the trajectory problem of interest is defined is explained. Finally, a simple trajectory optimization problem solution with analytical method and with OptimTraj are compared.

### 3.1 Trajectory Optimization

A single-phase continuous-time trajectory optimization problem is considered in this study. The general formula of the problem is given as,

$$\min_{t_0, t_F, x(t), u(t)} J_B(t_0, t_F, x(t_0), x(t_F)) + \int_{t_0}^{t_F} J_P(\tau, x(\tau), u(\tau)) d\tau \quad (3.1)$$

The constraints of the optimization problem can be defined with:

System dynamics,

$$\dot{x} = f(t, x(t), u(t)) \quad (3.2)$$

Boundary constraints,

$$C_B(t_0, t_F, x(t_0), x(t_F)) \leq 0 \quad (3.3)$$

Path constraints,

$$C_P(t, x(t), u(t)) \leq 0 \quad (3.4)$$

Bounds on state, control and time,

$$\begin{aligned} x_l &\leq x(t) \leq x_u \\ u_l &\leq u(t) \leq u_u \\ t_l &\leq t_0 < t_F \leq t_u \end{aligned} \quad (3.5)$$

In order to solve an optimization problem, numerical methods are used. Only some problems that their complexity is very low can be solved analytically. Direct collocation method is one of the numerical methods that is developed to solve



trajectory optimization problems. The direct collocation is also named direct transcription because it transcribes the problem into a nonlinear programming. Therefore, this method is outlined as “first discretize, then optimize” [35].

The integral expressions in the objective function or in the system dynamics can be approximated by different discretization methods. In the study, the trapezoidal rule is selected for the discretization of the Lagrange term in the objective function and for the defects calculation with system dynamics.

$$\int_{t_0}^{t_F} J_P(\tau, x(\tau), u(\tau)) d\tau \approx \sum_{k=0}^{N-1} \frac{(J_{P_k} + J_{P_{k+1}})}{2} (t_{k+1} - t_k) \quad (3.6)$$

$$\zeta_k = x_{k+1} - x_k - \frac{(f_k + f_{k+1})}{2} (t_{k+1} - t_k) \quad (3.7)$$

The constraints can be collected as,

$$c(x) = [C_{B_0} \zeta_0 C_{P_0} \zeta_1 C_{P_1} \dots \zeta_N C_{P_N} C_{B_F}]^T \quad (3.8)$$

For the equality constraints,

$$c(x) = 0 \quad (3.9)$$

For the inequality constraints,

$$c(x) \leq 0 \quad (3.10)$$

Hence, the trajectory optimization constraints are represented as nonlinear programming constraints, and then, the problem can be handled by parameter optimization solvers.

For position navigation in level flight, minimum control is implemented as the objective function because the helicopter is in a failure condition and large disturbances are not desired. For the transition cases from autorotation to level flight, minimum altitude descent is implemented as the objective; hence, the minimum allowable height from the ground for TRF can be estimated.

For minimum control,

$$\min_{t_0, t_F, x(t), u(t)} \int_{t_0}^{t_F} u(\tau)^2 d\tau \quad (3.11)$$

For minimum descent,

$$\min_{t_0, t_F, x(t), u(t)} \int_{t_0}^{t_F} z_i(\tau, x(\tau), u(\tau))^2 d\tau \quad (3.12)$$

### 3.2 OptimTraj

OptimTraj is a MATLAB library designed for solving continuous-time single-phase trajectory optimizations problems [3]. It finds the optimal trajectory that minimizes a cost function and satisfies a set of constraints defined by the user. The optimal trajectory is the sequence of states that moves the dynamical system from the initial condition to the final condition in state-space. The control sequence that forms the optimal trajectory is called optimal control.

OptimTraj is an open-source program and it is very user-friendly. The source code can be read clearly and how the software works can be understood. There are many example problems in the package to guide the users. New scripts and functions can be added for custom usage.

In OptimTraj, the numerical method that the trajectory optimization problem would be solved by can be selected. Available methods for the current version are

trapezoidal direct collocation, Hermite-Simpson direct collocation, Chebyshev-Lobatto orthogonal collocation and 4<sup>th</sup> order Runge-Kutta Multiple Shooting. In addition, a wrapper exists for calling GPOPS-II; however, since GPOPS-II is a commercial program, it is not an option for this study.

### 3.2.1 Defining the Problem

An optimization problem of interest must be clearly defined as the struct type in MATLAB with necessary fields [36]. `problem.func` is a struct that a set of user-defined function handles for the dynamics, objective and constraint functions. Fields of `problem.func` are shown in Table 3.1 with belonging function handles. `problem.bounds` is a struct that provides all constant bounds on time, state and control. Its fields are either scalar or a column vector and they are shown in Table 3.2 with corresponding values. `problem.guess` contains an initial guess for OptimTraj to start solving the problem. In Table 3.3, fields of `problem.guess` struct and corresponding values are shown. Lastly, `problem.options` provides solution options for the underlying nonlinear program and itself. In this struct, options such as solution method, number of grids, accuracy etc. can be determined.

Table 3.1 Fields of `problem.func` and Corresponding Function Handles

Fields of <code>problem.func</code>	<i>Corresponding function handles</i>
<code>problem.func.dynamics</code>	$f(t, x(t), u(t))$
<code>problem.func.pathObj</code>	$J_P(t, x(t), u(t))$
<code>problem.func.pathCst</code>	$C_P(t, x(t), u(t))$
<code>problem.func.bndObj</code>	$J_B(t_0, t_F, x(t_0), x(t_F))$
<code>problem.func.bndCst</code>	$C_B(t_0, t_F, x(t_0), x(t_F))$

Table 3.2 Fields of `problem.bounds` and Corresponding Values

Fields of <code>problem.bounds</code>	<i>Corresponding values</i>
<code>problem.bounds.initialTime.low</code>	$t_0^-$
<code>problem.bounds.initialTime.upp</code>	$t_0^+$
<code>problem.bounds.finalTime.low</code>	$t_F^-$
<code>problem.bounds.finalTime.upp</code>	$t_F^+$
<code>problem.bounds.initialState.low</code>	$x_0^-$
<code>problem.bounds.initialState.upp</code>	$x_0^+$
<code>problem.bounds.finalState.low</code>	$x_F^-$
<code>problem.bounds.finalState.upp</code>	$x_F^+$
<code>problem.bounds.state.low</code>	$x^-$
<code>problem.bounds.state.upp</code>	$x^+$
<code>problem.bounds.control.low</code>	$u^-$
<code>problem.bounds.control.upp</code>	$u^+$

Table 3.3 Fields of `problem.guess` and Corresponding Values

Fields of <code>problem.guess</code>	<i>Corresponding values</i>
<code>problem.guess.time</code>	$t_g$
<code>problem.guess.state</code>	$x_g$
<code>problem.guess.control</code>	$u_g$

In order to find the optimal flight trajectory of a helicopter in TRF, the problem is defined expressly in `OptimTraj`. Since several initial and final condition scenarios are investigated in this thesis, the fields are filled according to the needs of each case.

Firstly, the helicopter dynamics are defined with a function handle. This function returns the state derivatives that are found by interpolating two linear models in state-

space form. The interpolation process is done by calculating the distance of the current states to the trim states of the linear models. For this, norm operation is used with a weighting vector so that the influence of the states that are sensitive to define a flight condition is increased. Besides, unit differences between the states are minimized. For each scenario, the necessary linear models are given to OptimTraj.

Secondly, objective function and constraints are defined. In the study, there are basically two different goals according to the scenarios investigated. The first one is navigating the helicopter to a final location in space with minimum control effort, and the second one is bringing the helicopter from one trim condition to another with minimum descent. While, for the first case, control minimization is defined in the path objective handle, for the second case, descent minimization is defined there with a function that calculates the final inertial z-position. In the first case, the final position of the helicopter is defined in the path constraint with a function handle. This function returns the difference between the final position of the helicopter and the desired position. OptimTraj takes the output as an equality constraint; and as expected, it must satisfy this constraint in a tolerance for a feasible solution. Since the inertial positions of the helicopter ( $x_i, y_i, z_i$ ) do not exist in the states of the linear models, they need to be calculated from the rigid body states. That is why the position constraint is written in a function handle. On the other hand, for the second case, the helicopter states are forced to a final trim value by defining bounds on the final states which indicates that the helicopter would be in the desired trim at the final time. Since the states are the parameters that actually define a trim condition, they are explicit and are not needed to be calculated by a function. Thus, the final states are easily defined in OptimTraj.

Next, desired bounds on the state and the control are defined. Since there is no time limit, the upper bound on time is set as a large number. For the first case, the bounds on the final states and on the states along the trajectory are defined in such a way that the helicopter is near the neighborhood of the level trim. For the second case, the bounds on the final states are defined as mentioned before. The bounds on the states

along the path are stretched so that the states can reach their final value. In both cases, the bounds on the controls allow the states to reach their own limits.

Finally, initial guess and solution options are defined in OptimTraj. For the solution method, trapezoidal direct collocation is selected for all scenarios. The number of grid points is determined according to the final time. For large final time values, 150 grids are used whereas 50 grids are selected for the short ones. Constraint tolerances are kept very small like in the order of  $1e-10$ . Hence, the optimization problem is defined completely, and it is ready to be solved by OptimTraj.

### 3.3 Simple Example for Trajectory Optimization

A simple trajectory optimization problem is chosen in order to test OptimTraj. An analytical solution can be found in this problem and it is compared with the numerical result of OptimTraj.

The system is given by the following state equations

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \quad (3.13)$$

is transferred from the initial state  $x^T(0) = [1 \ 0]$  to the final state  $x^T(t_F) = [1 \ 0]$  while minimizing the cost function

$$J = \int_0^{t_F} (1 + u^2) dt \quad (3.14)$$

In order to find the optimal trajectory, firstly, the Hamiltonian is formed

$$\begin{aligned} J_B &= 0 \\ J_P &= 1 + u^2 \\ H &= 1 + u^2 + \lambda_1 x_2 + \lambda_2 u \end{aligned} \quad (3.15)$$

The necessary conditions for optimality are

$$\begin{aligned}\frac{\partial H}{\partial u} &= 0 \\ \frac{\partial H}{\partial x} &= -\dot{\lambda} \\ \frac{\partial H}{\partial \lambda} &= \dot{x}\end{aligned}\tag{3.16}$$

Yielding

$$\begin{aligned}2u + \lambda_2 &= 0 \\ \dot{\lambda}_1 &= 0 \\ \dot{\lambda}_2 &= -\lambda_1 \\ \dot{x}_1 &= x_2 \\ \dot{x}_2 &= u\end{aligned}\tag{3.17}$$

Generalized boundary condition

$$\left[ \left( \frac{\partial J_B}{\partial x_1} - \lambda_1 \right) dx_1 \right]_{t_F} + \left[ \left( \frac{\partial J_B}{\partial x_2} - \lambda_2 \right) dx_2 \right]_{t_F} + \left[ \left( H + \frac{\partial J_B}{\partial t} \right) dt \right]_{t_F} = 0\tag{3.18}$$

Since the final states are fixed,  $dx = 0$ . Hence,

$$H(t_F) = 0\tag{3.19}$$

Solving (3.17), (3.18) and (3.19) together gives

$$\begin{aligned}t_F &= \sqrt{6} \\ u(t) &= \sqrt{2/3}t - 1 \\ x_1(t) &= \frac{\sqrt{2/3}t^3}{6} - \frac{1}{2}t^2 + 1 \\ x_2(t) &= \frac{\sqrt{2/3}t^2}{2} - t\end{aligned}\tag{3.20}$$

The same problem is defined in OptimTraj, and then, is solved by the trapezoid method with different number of grid points. The optimal control and state trajectories are found by the analytical solution and the OptimTraj solution with 3 grids, 5 grids and 30 grids are plotted in Figure 3.1, Figure 3.2 and Figure 3.3, respectively. Note that the simulation results are obtained by feeding the control values which belongs to the OptimTraj solution in order to observe the effect of step size of time. However, after the controls and states are calculated at each grid point, for trapezoidal method, the controls are approximated by linear splines and the state trajectories are approximated by quadratic splines along the segments. Since the analytical solution of the problem gives a linear trajectory for the control,  $u$ , and a quadratic trajectory for the second state,  $x_2$ , the effect of step size of time cannot be noticed for the second state,  $x_2$ . The effect can only be seen in the first state,  $x_1$  and, still, it is sufficient to show that OptimTraj obeys the system dynamics. The calculated controls can indeed transfer the system to the desired position. Increasing the number of grids gets the solution closer to the analytical result as expected. While the analytical solution gives the final time as 2.4495 seconds, OptimTraj solution with 30 grids gives the final time as 2.4523 seconds. This test case gives a clue about the accuracy of OpimTraj.



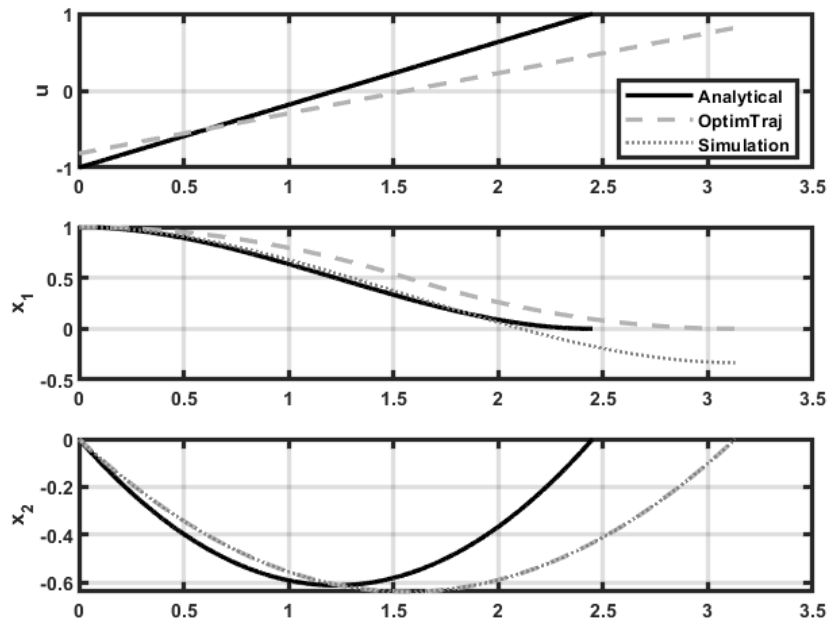


Figure 3.1. Comparison of the analytical solution and the OptimTraj solution with 3 grids for a simple trajectory optimization problem

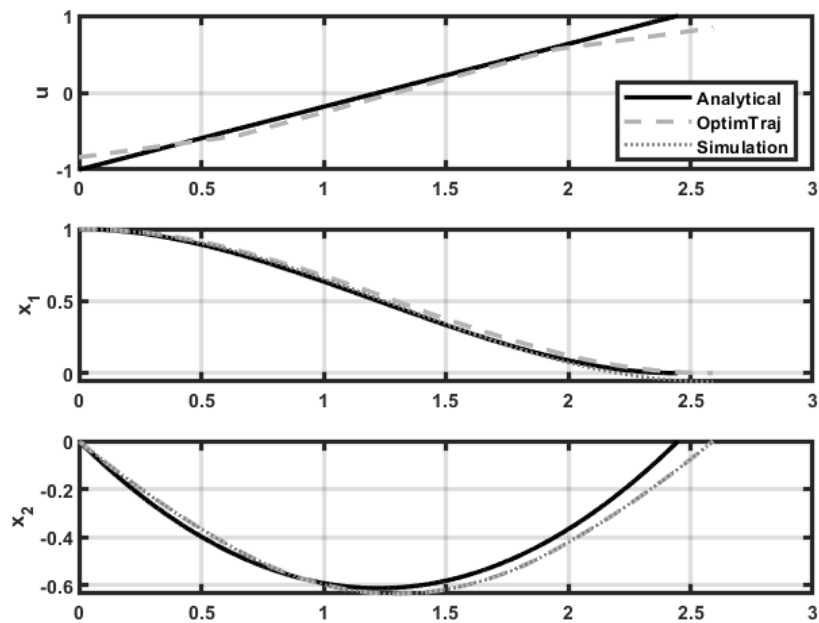


Figure 3.2. Comparison of the analytical solution and the OptimTraj solution with 5 grids for a simple trajectory optimization problem

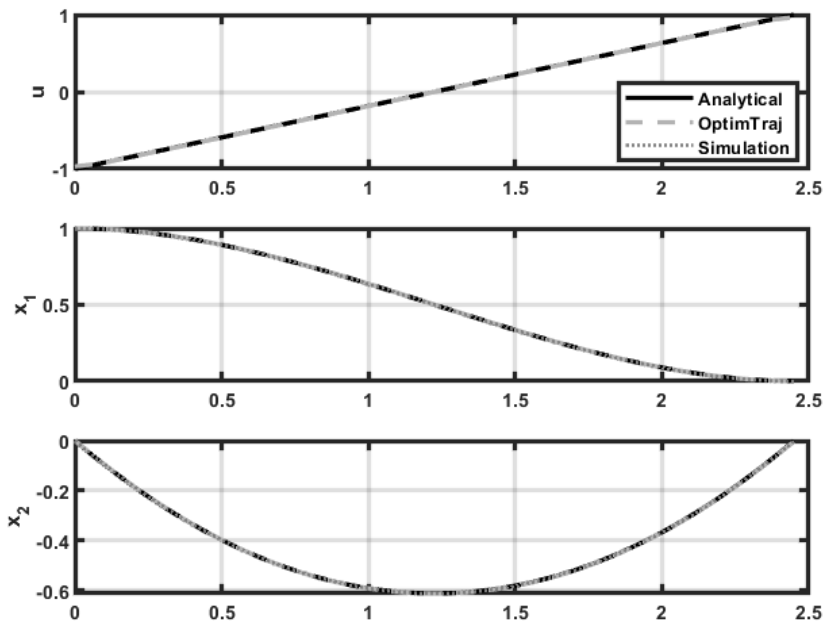


Figure 3.3. Comparison of the analytical solution and the OptimTraj solution with 30 grids for a simple trajectory optimization problem

## CHAPTER 4

### TRACKING CONTROLLER DESIGN

Flight control systems are designed in order to stabilize the aircraft and reduce the work load on the pilots. Because of the inherent instability of helicopters, control systems have much more importance for helicopter pilots. These systems are designed according to some guidelines for rotary wing flight control applications. For helicopters, Aeronautical Design Standard 33E-PRF (ADS-33E) form a basis for flight control engineers by defining the desired responses for input channels and modes of a helicopter [37]. Controllers are designed to satisfy these handling quality requirements in the helicopter response. In this study, however, autonomous flight under failure condition is investigated for only academic purposes. Therefore, any requirement on these design specifications is not checked during tracking controller development for the thesis.

After the reference trajectory is generated, research of a controller is focused on in order to follow the path. In helicopters, the longitudinal and the lateral dynamics are strongly coupled. Therefore, these dynamics cannot be handled separately. On the other hand, the rotational states are much faster than the translational dynamics. This allows to built the controller structure as cascaded. Dynamic inversion for the outer loop and linear model inversion for the inner loop are efficient and suitable methods for helicopter position tracking [6] [38] [39] [40]. In addition, for the scenarios in which the helicopter changes its trim condition, tracking of model states is investigated. LQI method in which an integrator is present in the forward path and acts as an output feedback controller is selected for this problem [41].

Adapting the controller to present condition of the aircraft is a common method for applications in aerospace industry at which the real system can be represented by linear models for different trim points [42]. In this study, the linear models are

interpolated to approximate the current state of the helicopter; thus, the controller could perform better.

In this chapter, two control methods are introduced in order to achieve trajectory tracking of a helicopter in TRF for two scenarios. The first one is for the position tracking and it is an inversion based controller. The second one is for the state tracking and it is an LQI controller.

#### **4.1 Inversion Based Controller**

Inversion based controller is developed in a cascade form. It consists of an inner and an outer loops. Main objective of the inner loop is to track the generated Euler angles and the total acceleration. On the other hand, the outer loop is responsible for tracking the position references. The block diagram of the control system developed in SIMULINK is depicted in Figure 4.1.

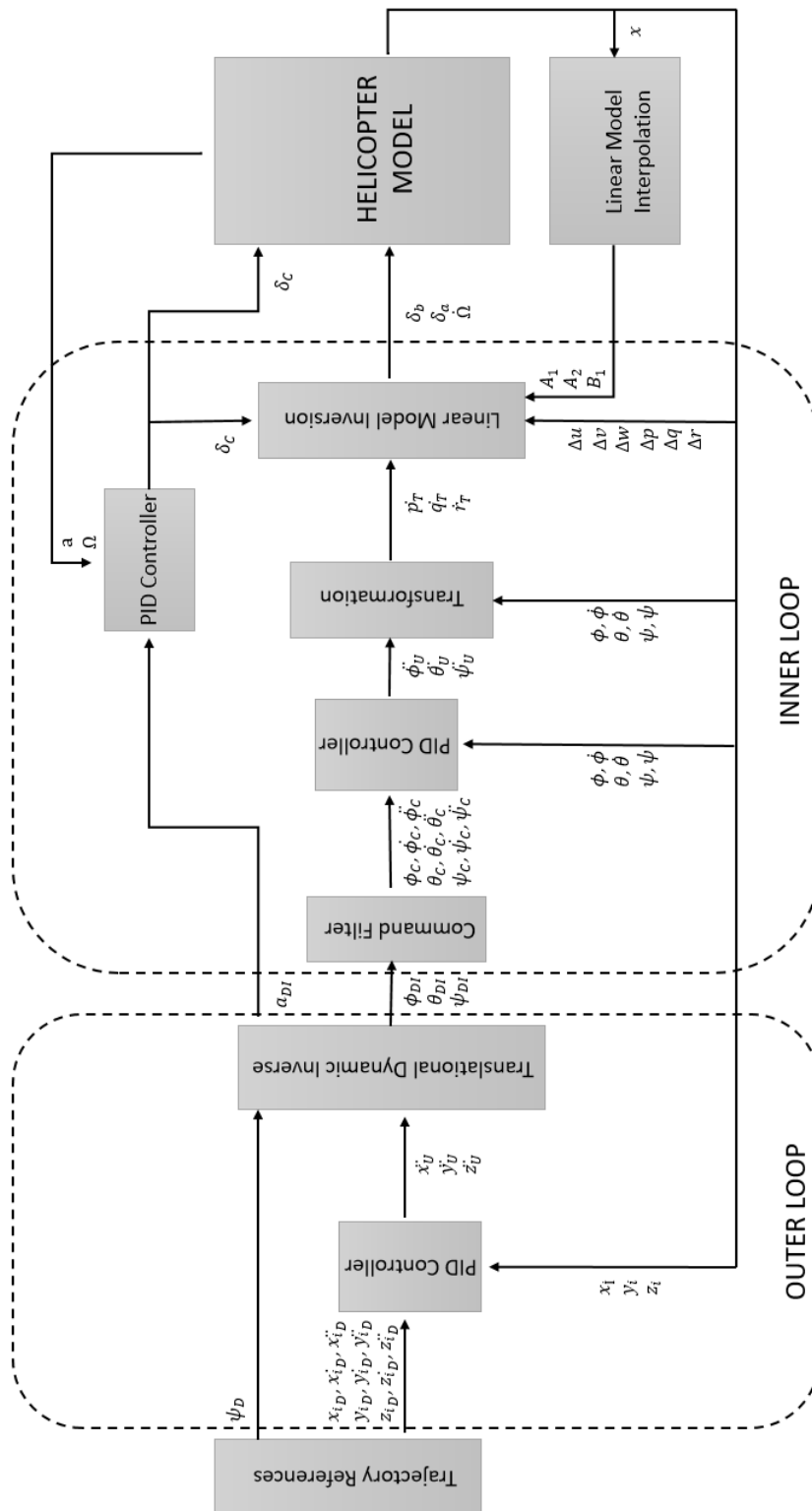


Figure 4.1. Block diagram of inversion based control system

### 4.1.1 Inner Loop

Rotational states are controlled in the inner loop by using the linear model inversion method. It is assumed that cyclic and pedal are responsible for the attitudes and collective is responsible for the total acceleration of a helicopter under nominal conditions [43]. However, in this study, instead of the pedal, the derivative of the speed of main rotor is utilized for the directional control of a helicopter in TRF.

The inner loop has two branches. In the first branch, Euler angles which are generated by the outer loop are passed through a second order command filter. In this block shown in Figure 4.2, the desired angle commands and their first and second derivatives are generated. The natural frequency of the command filter is selected to be slower than the dynamics of the attitude PID controller. Natural frequency of the command filter is chosen as 3 rad/s and the damping ratio is 0.8.

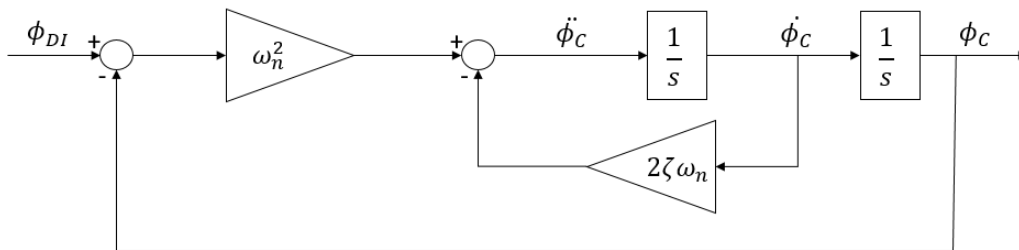


Figure 4.2. Block diagram of the command filter

The measured Euler angles and their derivatives are fed back to the PID block in the inner loop. The block diagram of the PID controller is depicted in Figure 4.3. While the error signals are operated with proportional, integral and derivative gains, the second derivative of the command signal is fed forward. Hence, sufficient transient

and steady-state responses are obtained by tuning these gains. It is noted that the outputs of this block correspond to the desired second derivatives of the Euler angles.

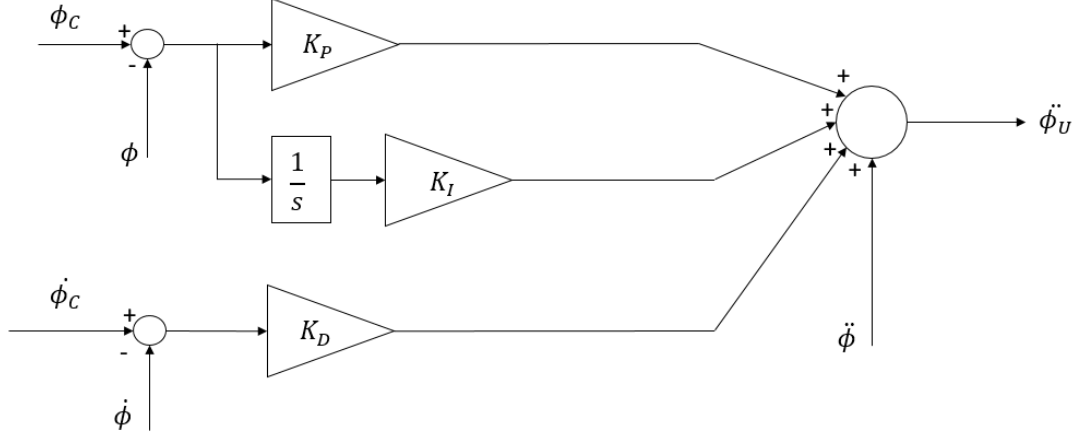


Figure 4.3. Block diagram of the PID controller

In the transformation block, by using (4.1), the Euler angles and their first and second derivatives are transformed into body angular accelerations. Here, the Euler angles and their first derivatives are the measurements coming from the model; whereas, the second derivatives come from the PID block. The outputs which are the body angular accelerations are the command signals for the linear model inversion block.

$$\begin{aligned}
 \dot{p} &= \ddot{\phi} - \ddot{\psi} \sin \theta - \dot{\theta} \dot{\psi} \cos \theta \\
 \dot{q} &= \ddot{\theta} \cos \phi - \dot{\phi} \dot{\theta} \sin \phi + \ddot{\psi} \sin \phi \cos \theta + \dot{\phi} \dot{\psi} \cos \phi \cos \theta - \dot{\theta} \dot{\psi} \sin \phi \sin \theta \\
 \dot{r} &= -\ddot{\theta} \sin \phi - \dot{\phi} \dot{\theta} \cos \phi + \ddot{\psi} \cos \phi \cos \theta - \dot{\phi} \dot{\psi} \sin \phi \cos \theta - \dot{\theta} \dot{\psi} \cos \phi \sin \theta
 \end{aligned} \tag{4.1}$$

Next, the linear model inversion method is implemented to decouple rotational and translational states. The linear model used in this study contains a 9-by-9 system matrix and a 9-by-5 control matrix. The inputs and states are given in (2.1) in

Helicopter Mathematical Model chapter. The decoupled system is given in (4.2) in state-space representation where  $x_1$  contains the body rates,  $x_2$  contains the body velocities and collective input,  $A_1$  is the stability matrix that contains the effects of rotational states on themselves,  $A_2$  is the stability matrix that contains the effects of translational states on rotational states,  $B_1$  is the control matrix that contains the effects of controls on rotational states, and finally,  $u_1$  contains longitudinal cyclic, lateral cyclic and rotor speed input. Collective is added to  $x_2$  vector as a slow translational state because its dynamic effect is slower than the other controls [44].

$$\dot{x}_1 = A_1 x_1 + A_2 x_2 + B_1 u_1 \quad (4.2)$$

$$\begin{aligned} x_1 &= [p, q, r]^T \\ x_2 &= [u, v, w, \delta_c]^T \\ u_1 &= [\delta_b, \delta_a, \dot{\Omega}]^T \end{aligned} \quad (4.3)$$

$A_1$ ,  $A_2$  and  $B_1$  are defined in MATLAB syntax in (4.4).

$$\begin{aligned} A_1 &= A([7 \ 8 \ 9], [7 \ 8 \ 9]) \\ A_2 &= [A([7 \ 8 \ 9], [4 \ 5 \ 6]), B([7 \ 8 \ 9], 1)] \\ B_1 &= B([7 \ 8 \ 9], [2 \ 3 \ 5]) \end{aligned} \quad (4.4)$$

By manipulating (4.2),  $u_1$  can be found as in (4.5).

$$u_1 = B_1^{-1}(\dot{x}_1 - A_2 x_2 - A_1 x_1) \quad (4.5)$$



It is noted that the body angular accelerations are the outputs of the transformation block, the body rates and the body velocities are the measurements coming from the helicopter model and the collective is the output of the collective controller block.

In the second branch of the inner loop, desired total acceleration which is generated by the outer loop is used in order to produce collective input. The vertical component of the thrust vector of the main rotor is the main contributor of the helicopter total acceleration. Therefore, relating collective with the total acceleration is a very reasonable idea. The acceleration command is processed in the collective controller by using the acceleration which is calculated from the model outputs as feedback signal. The calculation of the helicopter acceleration is given in (4.6). Note that the commanded acceleration is given in inertial frame; therefore, body acceleration of the helicopter is transformed.

$$\begin{bmatrix} a_{x_i} \\ a_{y_i} \\ a_{z_i} \end{bmatrix} = L_{EB} \left( \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} + \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \right) - \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \quad (4.6)$$

Where  $L_{EB}$  is the transformation matrix from body-fixed reference frame to Earth-fixed reference frame which is given in (4.7) [45].

$$L_{EB} = \begin{bmatrix} \cos\theta\cos\psi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\ \cos\theta\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \quad (4.7)$$

The collective controller is composed of a PI controller and a P controller. While PI controller processes the acceleration error signal, the P controller processes the error signal of main rotor speed. P controller for the rotor speed is added in order to prevent too high or too low rotor speed values. Thus, input effort on the rotor speed is passed to the collective. The block diagram of the collective controller is shown in Figure 4.4.

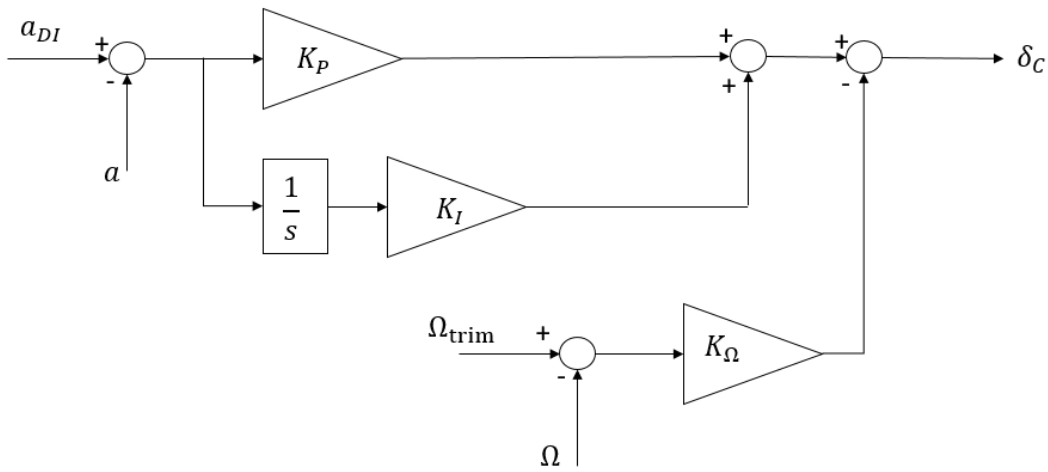


Figure 4.4. Block diagram of the collective controller

#### 4.1.2 Outer Loop

The inertial positions are controlled in the outer loop. It receives the position and heading references found by OptimTraj and generates roll, pitch and acceleration commands to the inner loop [43]. Heading angle reference is directly passed to the inner loop without any change. In the outer loop, a command filter is not used. The necessary derivatives of inertial positions are calculated offline and given as reference signals to the PID block.

The measured inertial positions and their derivatives are fed back to the PID block in the outer loop. As with the PID controller in the inner loop, while the error signals are operated with proportional, integral and derivative gains, the second derivative of the command signal is fed forward. Hence, sufficient transient and steady-state responses are obtained by tuning these gains. It is noted that the outputs of this block correspond to the desired inertial accelerations.

In translational dynamic inverse, it is assumed that the body accelerations in x and y-directions are assumed to be small compared to body acceleration in z-direction. Hence, by using (4.8), the desired roll and pitch angles can be calculated [43]. The translation dynamic inverse block composes (4.9).

$$\begin{bmatrix} a_{x_i} \\ a_{y_i} \\ a_{z_i} - g \end{bmatrix} = L_{EB} \begin{bmatrix} a_{x_b} \\ a_{y_b} \\ a_{z_b} \end{bmatrix} \quad (4.8)$$

$$a = \sqrt{a_{x_i}^2 + a_{y_i}^2 + (a_{z_i} - g)^2}$$

$$\phi \approx \sin^{-1} \left( \frac{-a_{x_i} \sin \psi + a_{y_i} \cos \psi}{a_{total}} \right) + \phi_{trim} \quad (4.9)$$

$$\theta \approx \tan^{-1} \left( \frac{-a_{x_i} \cos \psi + a_{y_i} \sin \psi}{a_{z_i} - g} \right) + \theta_{trim}$$

## 4.2 LQI Controller

Linear quadratic regulator with integral action is implemented to track the helicopter states. It is an optimal solution for tracking states of a linear system in which the performance index in (4.10) is minimized. It consists of a state feedback gain set and a feedforward integral gain set. The block diagram of the control system developed in SIMULINK is depicted in Figure 4.5.

$$J = \int_0^{\infty} (x^T Q x + u^T R u) dt \quad (4.10)$$

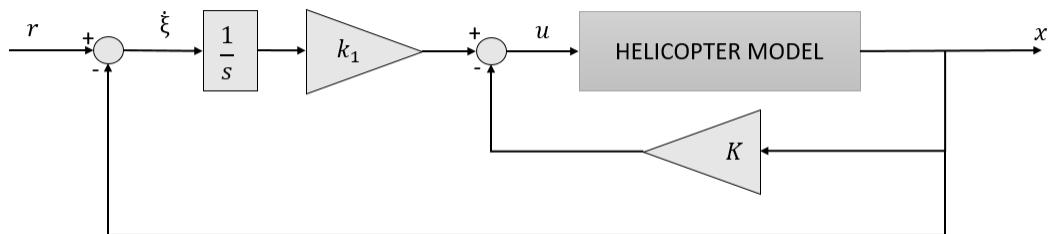


Figure 4.5. Block diagram of LQI control system

The optimal gains are found by using .

$$K = R^{-1}B^T P \quad (4.11)$$

Where  $P$  is a positive-definite matrix that results from Riccati equation in (4.12).

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \quad (4.12)$$

The matrices of the resulting regulator are given by (4.13).

$$\hat{A} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \quad \hat{B} = \begin{bmatrix} B \\ 0 \end{bmatrix} \quad (4.13)$$

The solution of (4.12) would give the gains in the form of,

$$\hat{K} = [K \quad -k_1] \quad (4.14)$$

Where  $K$  is the gain set for the stabilization and  $k_1$  is the gain set for the integral action. Hence, the closed-loop system can be written by (4.15) [46].

$$\begin{aligned} \begin{bmatrix} \dot{x} \\ \dot{\xi} \end{bmatrix} &= \begin{bmatrix} A - BK & Bk_1 \\ -C & 0 \end{bmatrix} \begin{bmatrix} x \\ \xi \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r \\ y &= [C \quad 0] \begin{bmatrix} x \\ \xi \end{bmatrix} \end{aligned} \quad (4.15)$$

Note that the system is full state feed back. And, the input is given by (4.16).

$$u = -[K \quad -k_1] \begin{bmatrix} x \\ \xi \end{bmatrix} \quad (4.16)$$

Since the initial and the final models are different for each case, the weighting matrix for state error,  $Q$ , and the weighting matrix for the input effort,  $R$  are determined for each case. Values for these matrices are given in the Appendix B.

In addition, for a fictional condition that all states can be tracked, the controller would bring the helicopter that perfectly fits its linear model to the desired final trim condition. However, tracked state set is defined for each case because only four states can be tracked and the states that must be selected could differ for each case. This means that only the states in the set are given as the reference and the others are just regulated. The tracked state selection for the flight scenarios is shown in Table 4.1. It can be noticed that for the transition from autorotation to level flight, the body forward speed is changed with the body lateral speed. This modification is made because while the forward speed is more effective for the transition between autorotation trims, the lateral speed is more effective for the transition from autorotation to level flight. For all cases, vertical speed, roll angle and pitch angle are considered to be important for defining trim condition.

Table 4.1 State Selection for Tracking

Flight Scenarios	<i>Selected States</i>
Autorotation at 40 kt to Autorotation at 60 kt	$\phi, \theta, u, w$
Autorotation at 60 kt to Autorotation at 80 kt	$\phi, \theta, u, w$
Autorotation at 80 kt to Level Flight at 80 kt	$\phi, \theta, v, w$



## CHAPTER 5

### RESULTS AND DISCUSSION

The generated trajectory and tracking performance of the controllers are presented in this chapter. Trajectory generation with OptimTraj and tracking simulations with the linear models and the nonlinear model are plotted together in each figure.

There have been investigated 8 scenarios for the helicopter in TRF; however, 5 of them are presented in this chapter. The rest of the scenarios can be found in Appendix C. The scenarios that have been investigated are summarized in Table 5.1. Note that the initial trim is level flight at 80 knots for cases 1,2,3,4 and 5; moreover, the initial position is at the origin for all cases.

Table 5.1 Summary of Investigated Flight Scenarios

Number of the Flight Scenario	<i>Trajectory Description</i>	<i>Controller</i>
Case 1	Flight to inertial position of [5000, 500, -200] ft	IBC
Case 2*	Flight to inertial position of [5000, -500, -200] ft	IBC
Case 3	Flight to inertial position of [5000, -2500, 0] ft	IBC
Case 4*	Flight to inertial position of [5000, 2500, 0] ft	IBC
Case 5	Flight to inertial position of [5000, 0, 300] ft	IBC
Case 6*	Autorotation at 40 kt to autorotation at 60 kt	LQI
Case 7	Autorotation at 60 kt to autorotation at 80 kt	LQI
Case 8	Autorotation at 80 kt to level flight at 80 kt	LQI

\* Presented in Appendix C

## 5.1 Results for Case 1

In Case 1, the helicopter in level flight at 80 knots is commanded to gently maneuver to right and climb. The resulting flight path is shown in Figure 5.1 and the inertial position variables are plotted in Figure 5.2. It can be noticed that both linear and nonlinear models follow  $x$  and  $y$  position references. However, for the  $z$ -direction, the nonlinear model could not achieve to follow the path. The Euler angles and the body velocities are presented in Figure 5.3 and Figure 5.4, respectively. The descent can be seen from  $w$  also. When the Euler angles are analyzed, it is seen that the roll and heading angles are very close to their optimal values for both models. On the other hand, the pitch angle of the nonlinear model is lower than the optimal values throughout the simulation. The effect of this situation can be seen in the angle of attack in Figure 5.5. Another parameter that needs attention is the rotor speed which is plotted in Figure 5.6. Although the optimal rotor speed and the rotor speed of the linear model are very close, the nonlinear results are different. This difference could be related to the forward speed. The optimal solution requires increasing the rotor speed. However, increasing rotor speed would require more anti-torque and would yaw the helicopter. In addition, it is known that the helicopter cannot stay at level flight below 77 knots for nominal rotor speed. Increasing the rotor speed would increase the trim speed too. Therefore, the rotor speed is not increased in order to keep up with the longitudinal and lateral path. Obviously, this complex dynamics is not captured well by the linear models and the optimal trajectory can be tracked successfully.

In Figure 5.7, the transition between the linear models is shown. It can be noticed that as the roll angle increases, the model gets closer to the right turn model (80TR). The model transition plot is suitable with the other outputs of the simulations. The linear result is closer to OptimTraj result than the nonlinear transition value. Lastly, the helicopter inputs are shown in Figure 5.8.



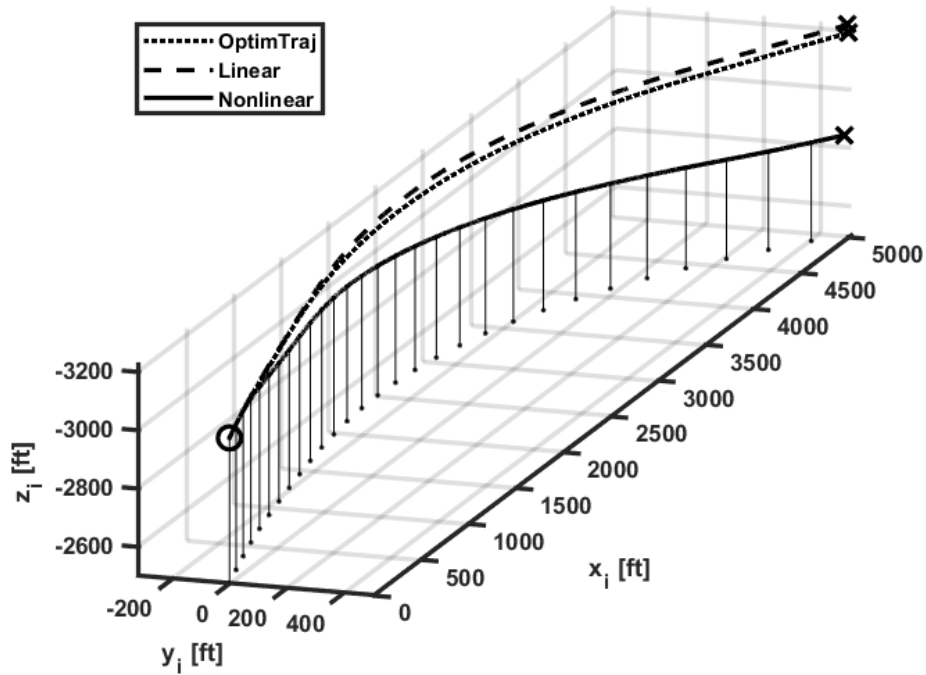


Figure 5.1. Trajectory of the helicopter for Case 1

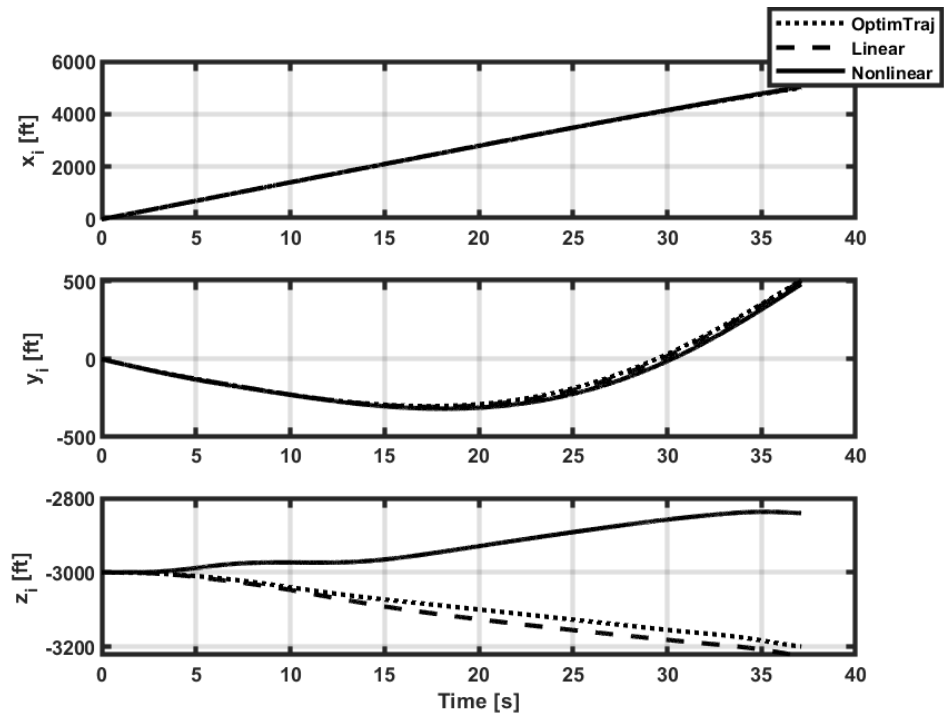


Figure 5.2. Inertial positions of the helicopter for Case 1

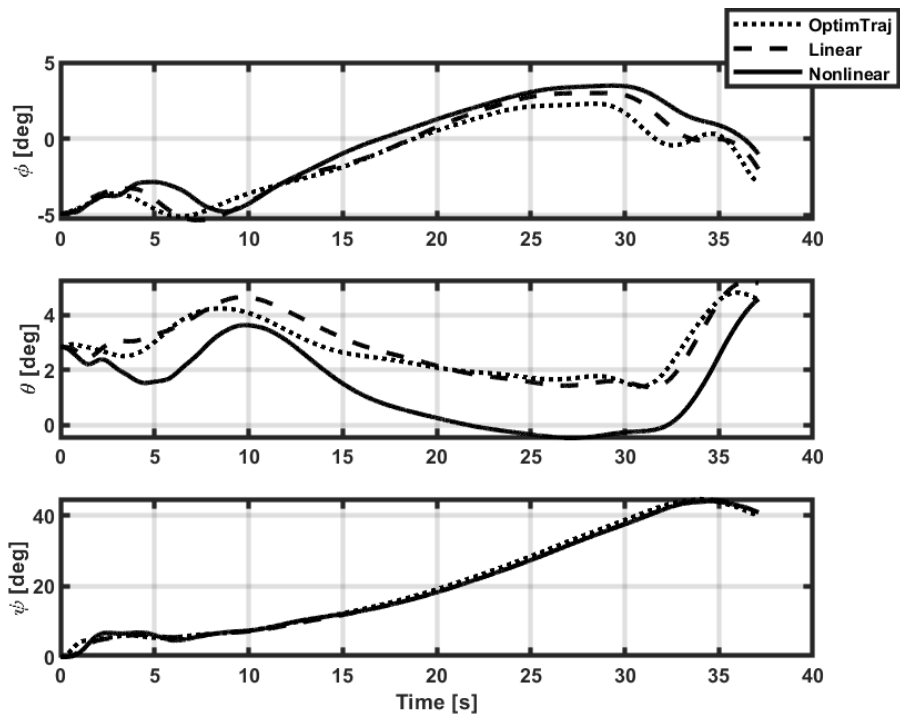


Figure 5.3. Euler angles of the helicopter for Case 1

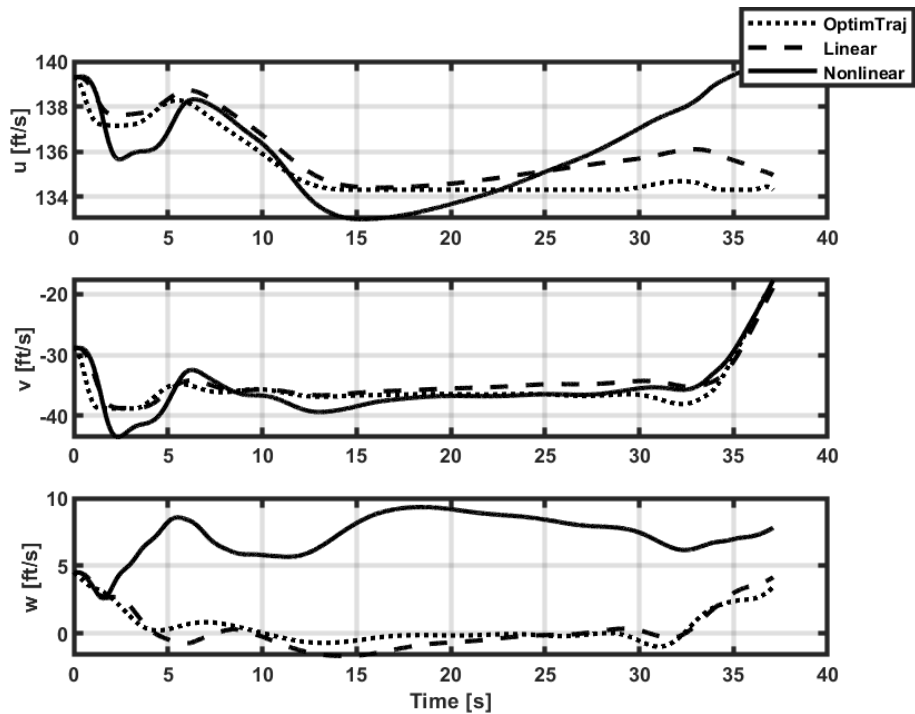


Figure 5.4. Body velocities of the helicopter for Case 1

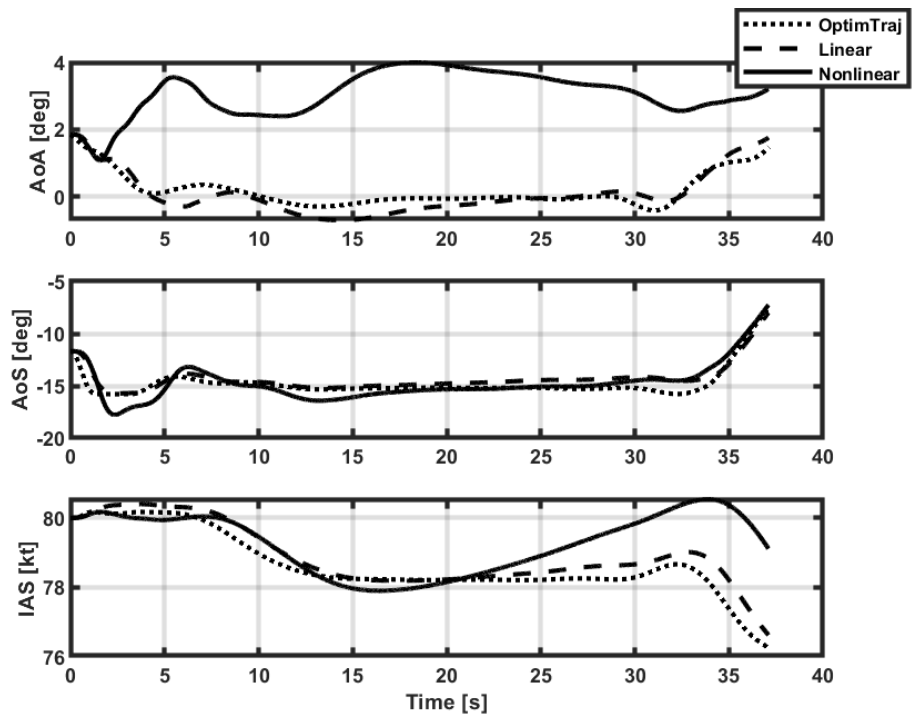


Figure 5.5. Air data parameters of the helicopter for Case 1

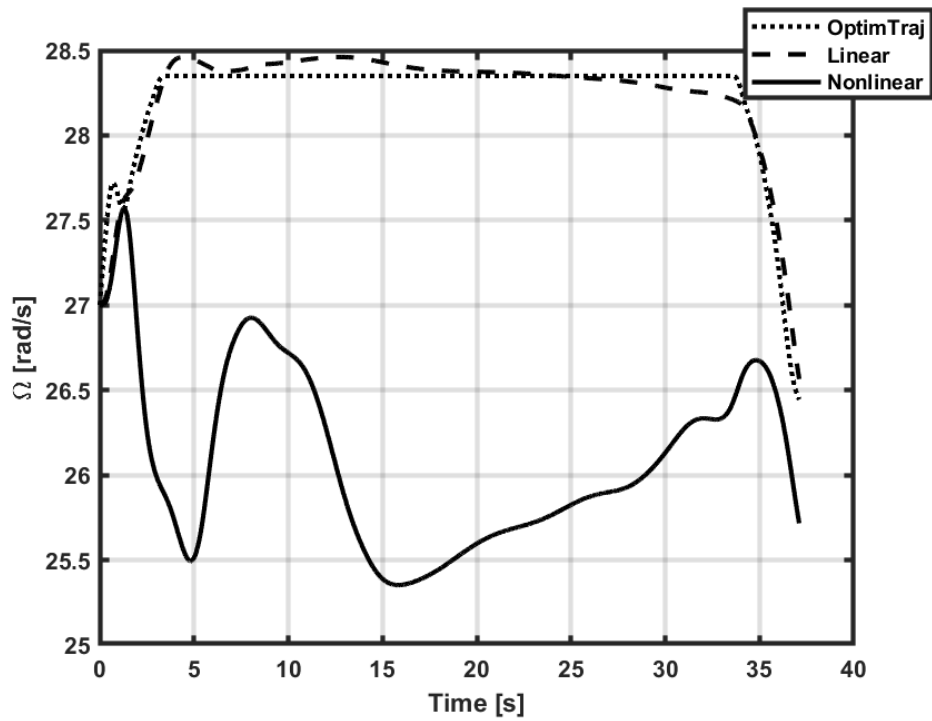


Figure 5.6. Main rotor speed of the helicopter for Case 1

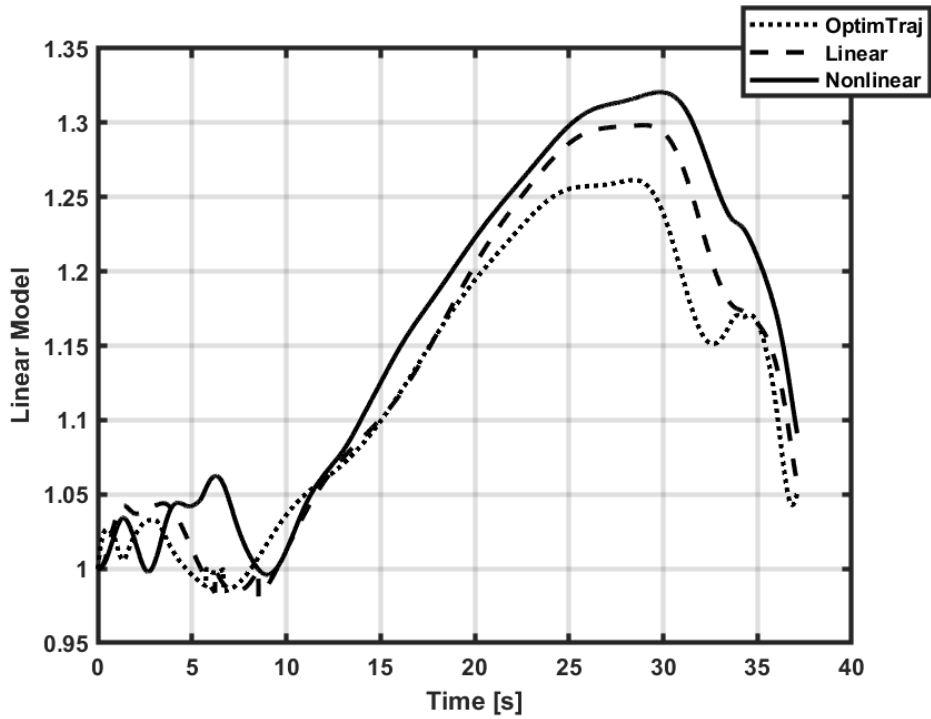


Figure 5.7. Linear model transition for Case 1 (0: 80TL, 1: 80L, 2: 80TR)

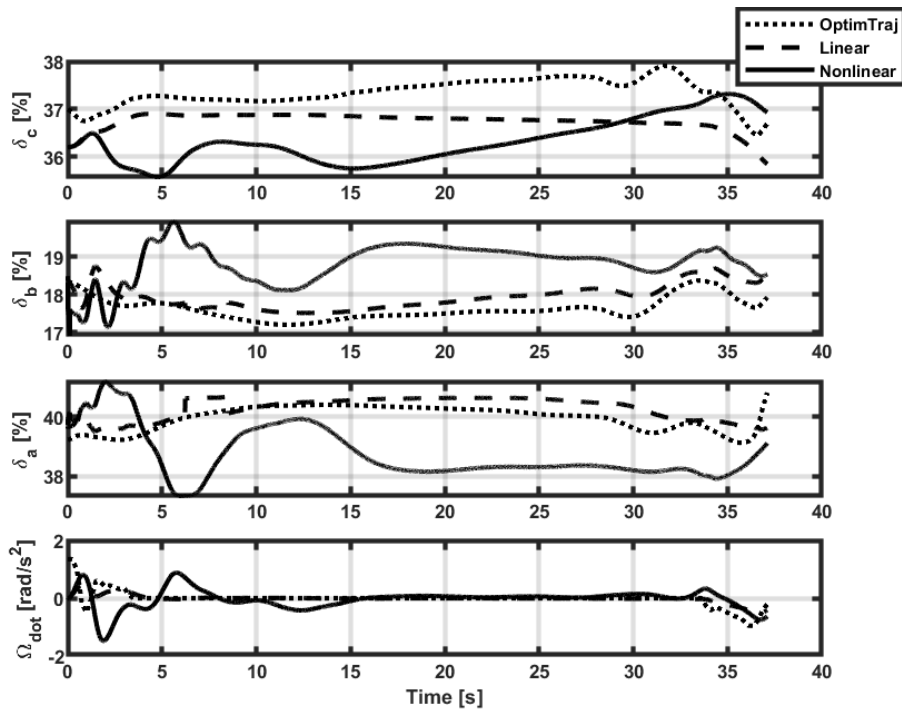


Figure 5.8. Helicopter inputs for Case 1

## 5.2 Results for Case 3

In Case 3, the helicopter in level flight at 80 knots is commanded to maneuver for left turn without an altitude change. The resulting flight path is shown in Figure 5.9 and the inertial position variables are plotted in Figure 5.10. It is seen that both linear and nonlinear models can track the position references with minor differences. The Euler angles and the body velocities are presented in Figure 5.11 and Figure 5.12, respectively. In both figures, it is seen that the parameters are close to their optimal counterpart. However, it can be noticed again that the performance of the controller is not perfectly good in the z-axis. Nevertheless, it supplies a sufficient contribution.

The air data information is given in Figure 5.13. They all are close to optimal values and are compatible with the other variables. The rotor speed is plotted in Figure 5.14. It can be stated that rotor speed is changed for all studies in similar ways.

In Figure 5.15, the transition between the linear models is shown. It can be noticed that as the roll angle decreases, the model gets closer to the left turn model (80TL). The model transition plot is suitable with the other outputs of the simulations. It is clearly seen here that both models acted almost the same. Lastly, the helicopter inputs are shown in Figure 5.16.

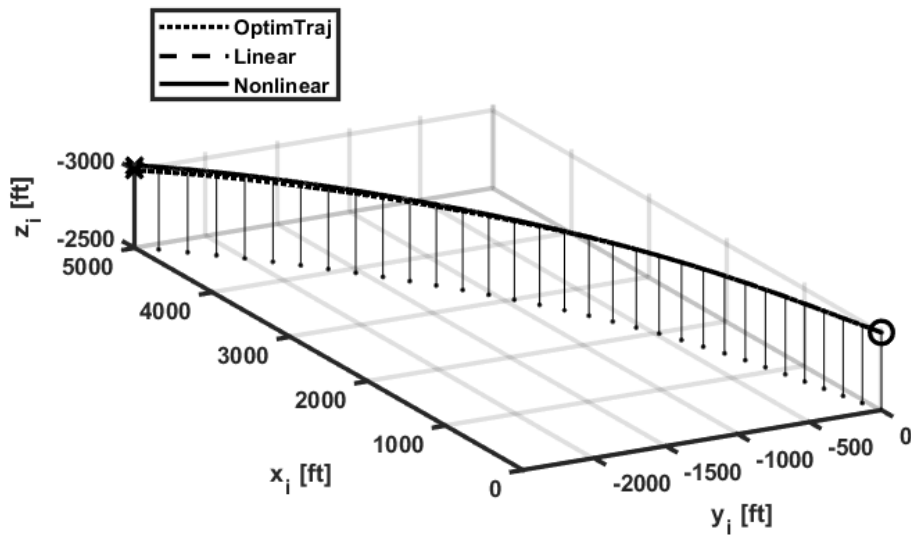


Figure 5.9. Trajectory of the helicopter for Case 3

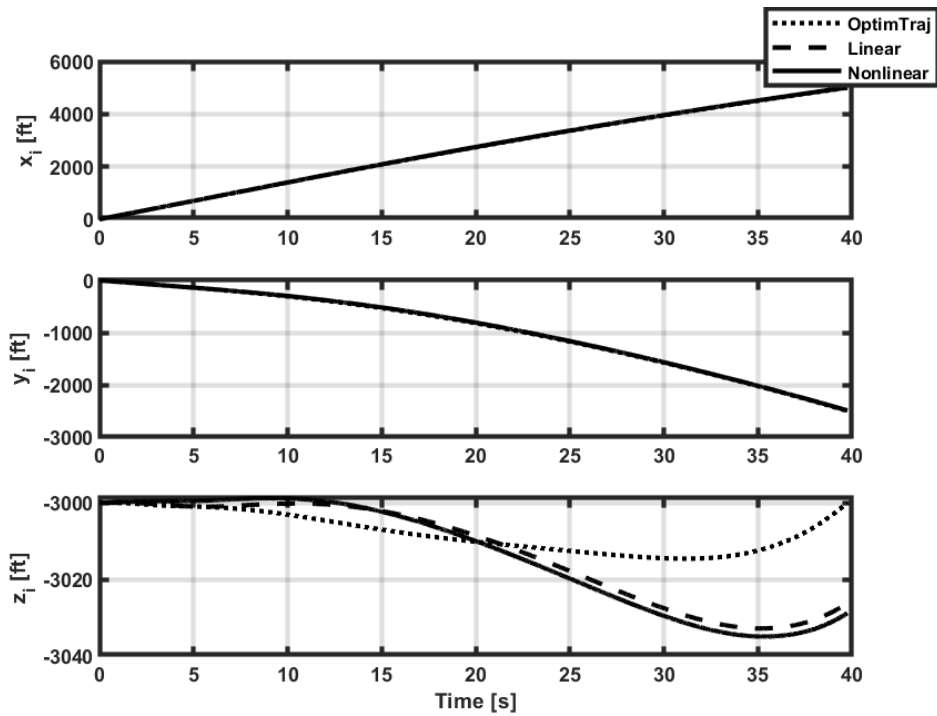


Figure 5.10. Inertial positions of the helicopter for Case 3

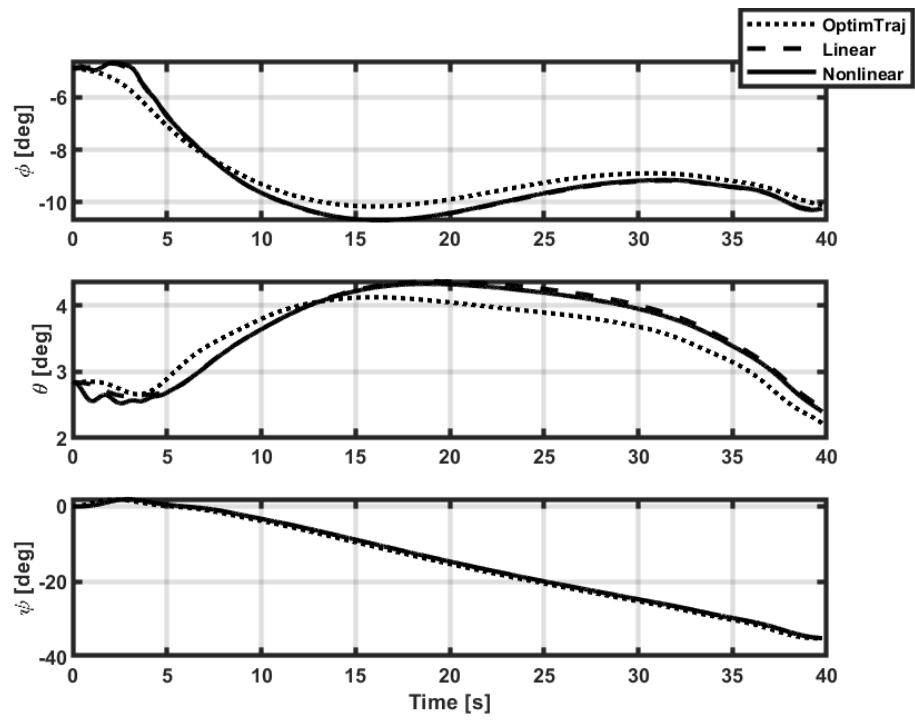


Figure 5.11. Euler angles of the helicopter for Case 3

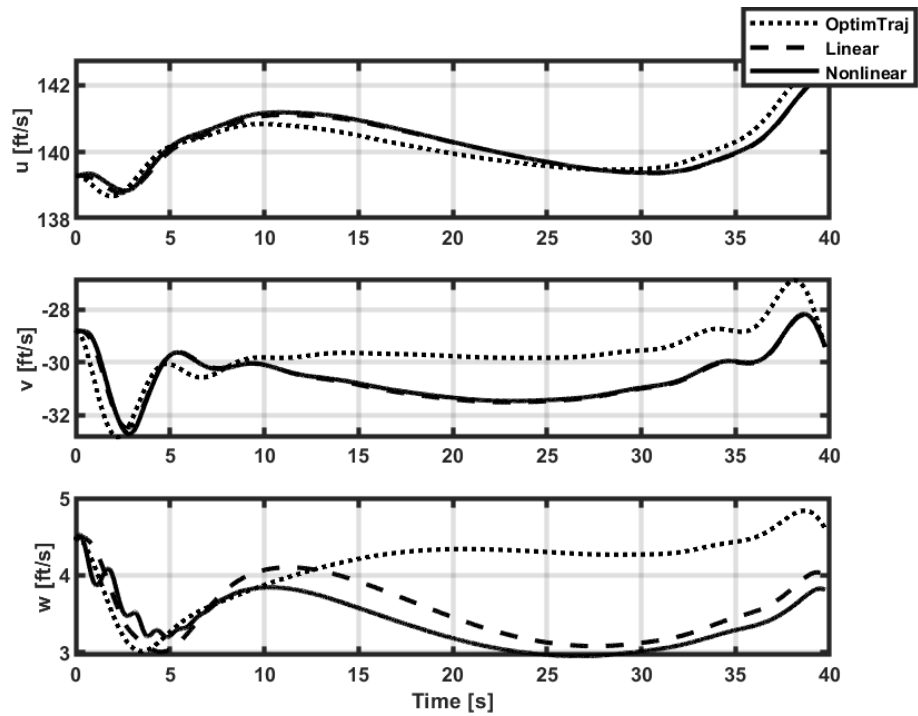


Figure 5.12. Body velocities of the helicopter for Case 3

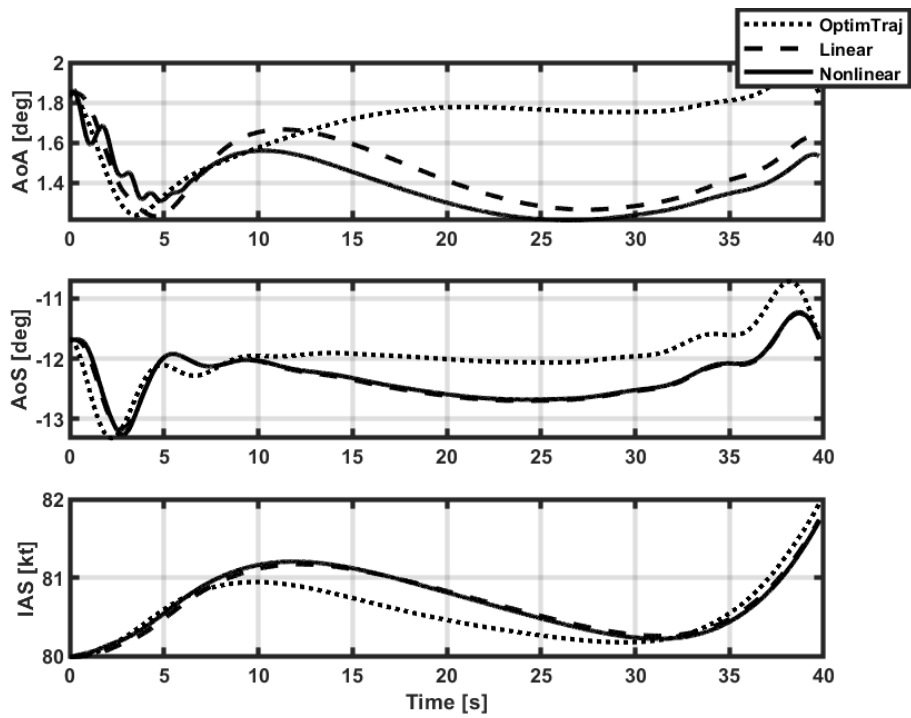


Figure 5.13. Air data parameters of the helicopter for Case 3

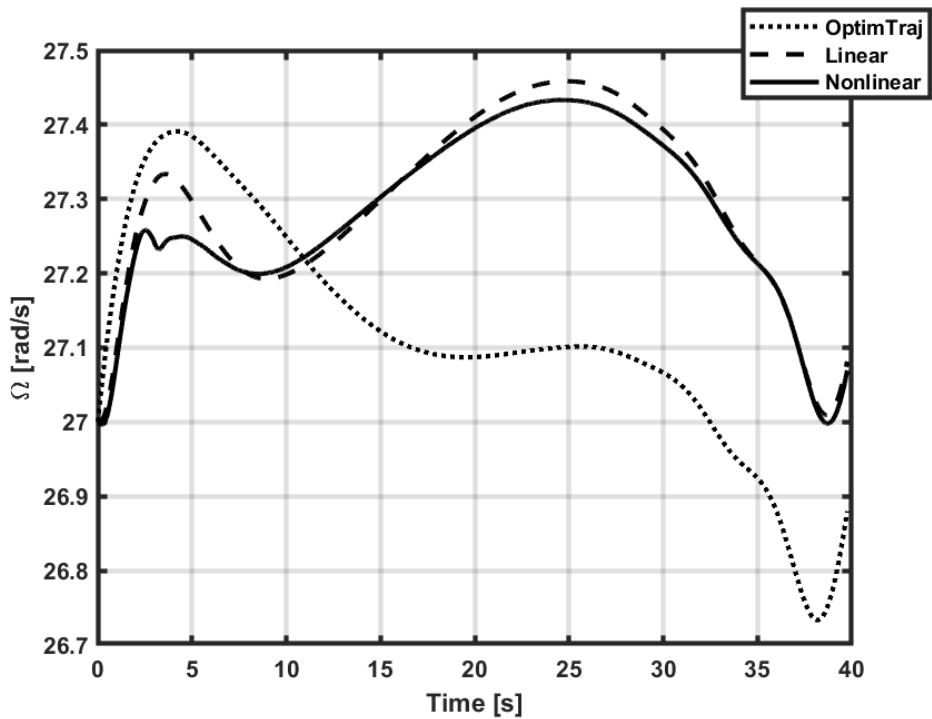


Figure 5.14. Main rotor speed of the helicopter for Case 3



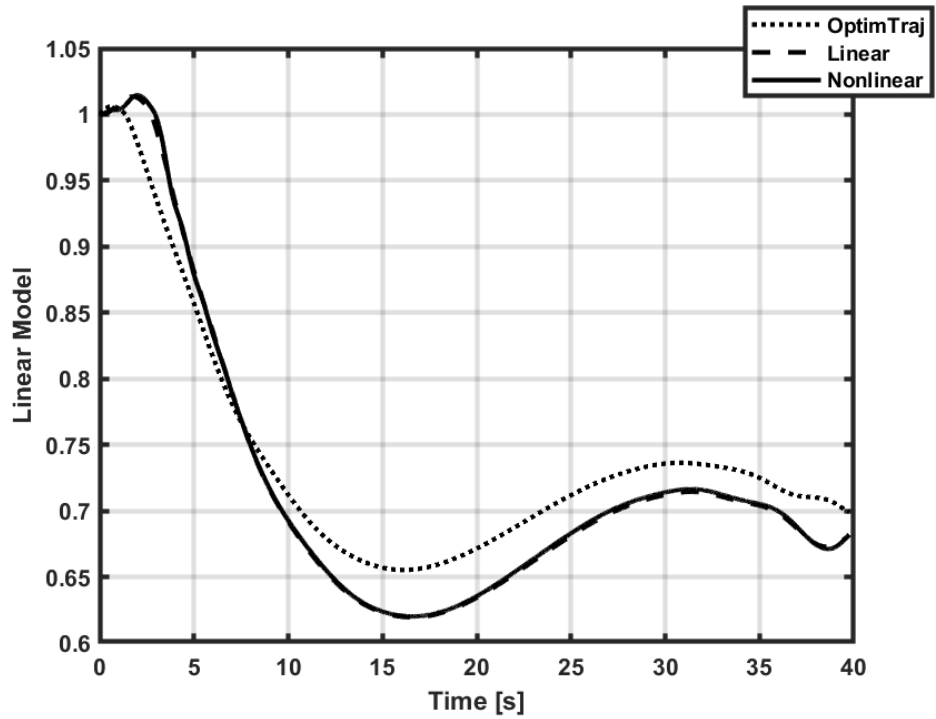


Figure 5.15. Linear model transition for Case 3 (0: 80TL, 1: 80L, 2: 80TR)

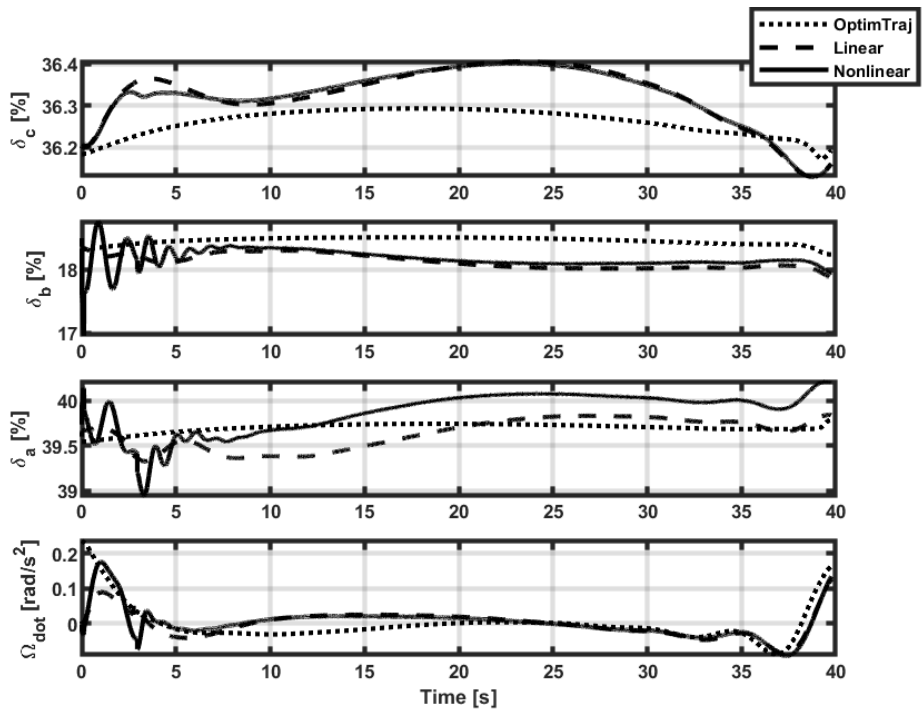


Figure 5.16. Helicopter inputs for Case 3

### 5.3 Results for Case 5

In Case 5, the helicopter in level flight at 80 knots is commanded to descend without turning. This scenario is added in order to evaluate the performance of the controller for descent condition.

The resulting flight path is depicted in Figure 5.17 and the inertial position variables are plotted in Figure 5.18. It is seen that both linear and nonlinear models can track the position references with minor differences. At the final time, the nonlinear model helicopter has a little lower altitude.

The Euler angles and the body velocities are shown in Figure 5.19 and Figure 5.20, respectively. In both figures, it is seen that the parameters are close to their optimal counterpart. It can be stated that comparing the previous cases, the performance of the controller in the z-direction is better for this case.

The air data information is given in Figure 5.21. They all are close to optimal values and are suitable for the other variables. The rotor speed is plotted in Figure 5.22. After the first reduction in rotor speed, both model values fluctuate; however, it is more severe in the nonlinear model. In the final, all study values increase so that the lateral position of the helicopter could be achieved.

The transition between the linear models is shown in Figure 5.23. It can be noticed that there is a small jump in all results. The reason for the jump is the model switching by the controller for model interpolation between two linear models. The helicopter takes a left turn at the beginning; then, it turns right with a bigger radius. This maneuver explains the model transition plot. It is seen that both models acted similarly. Lastly, the helicopter inputs are shown in Figure 5.24.

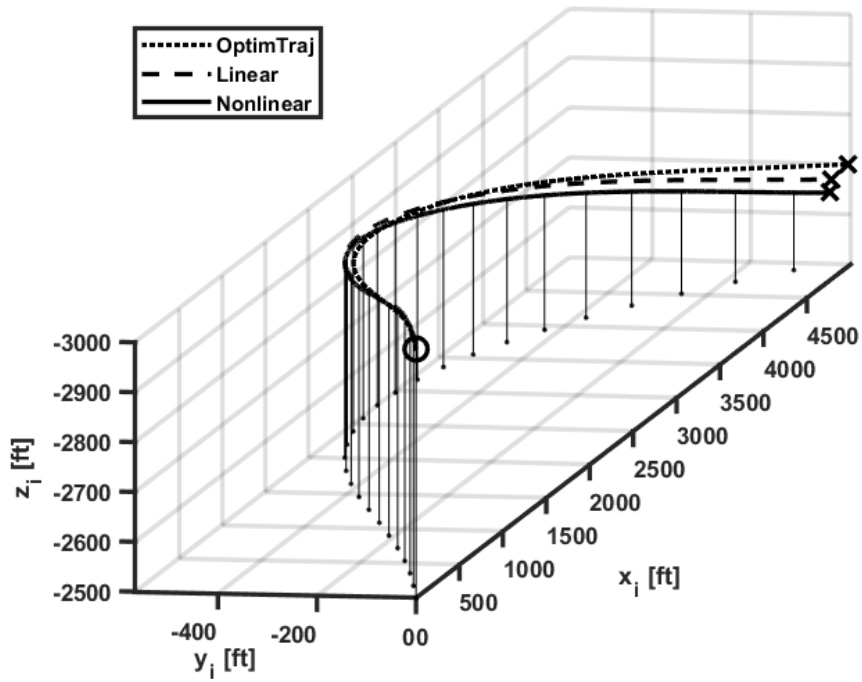


Figure 5.17. Trajectory of the helicopter for Case 5

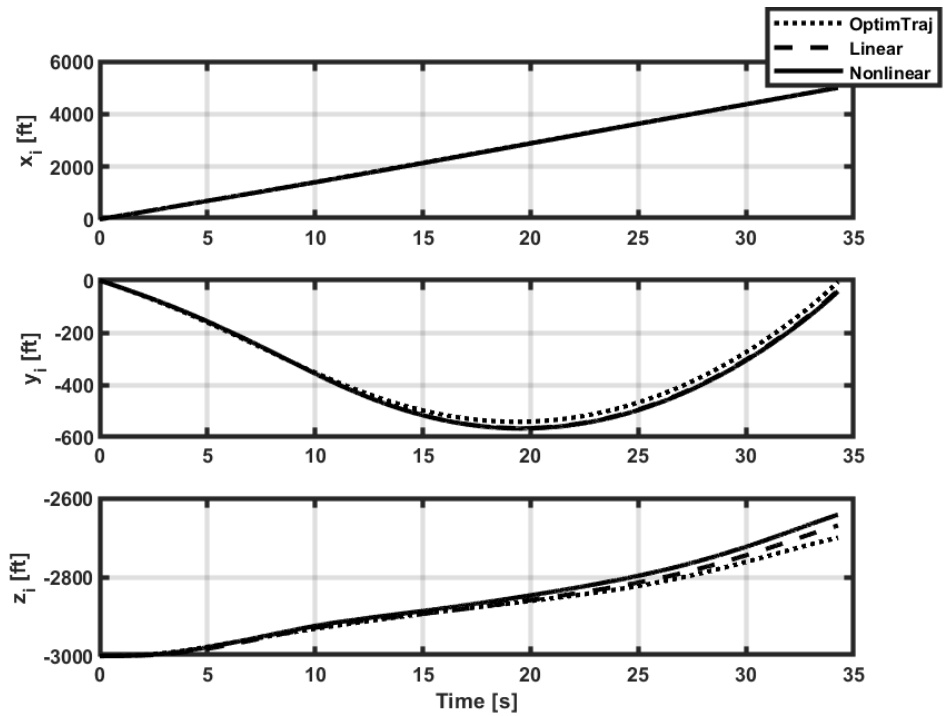


Figure 5.18. Inertial positions of the helicopter for Case 5

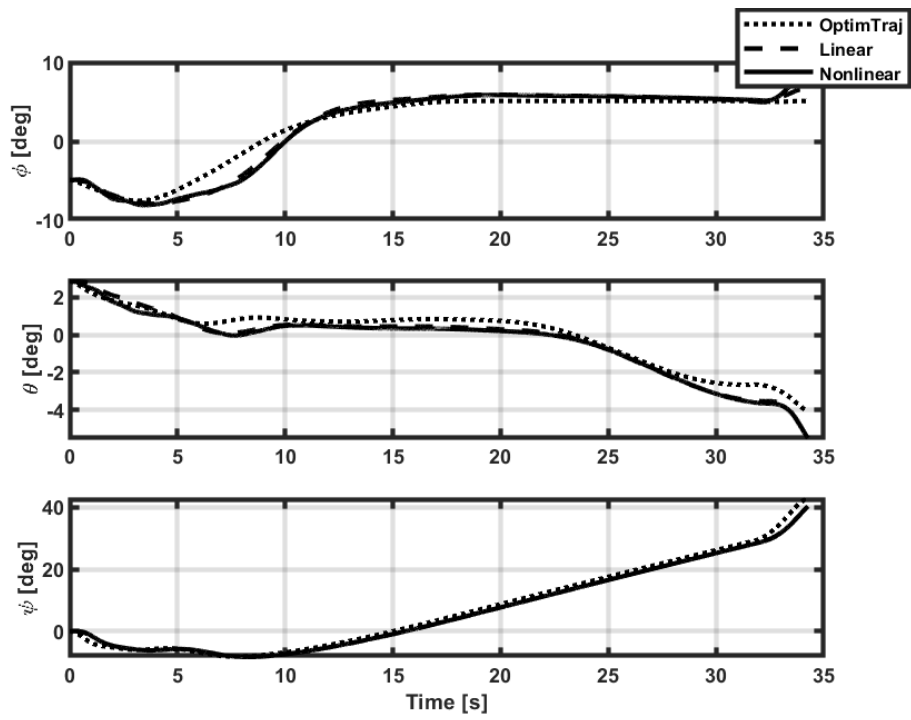


Figure 5.19. Euler angles of the helicopter for Case 5

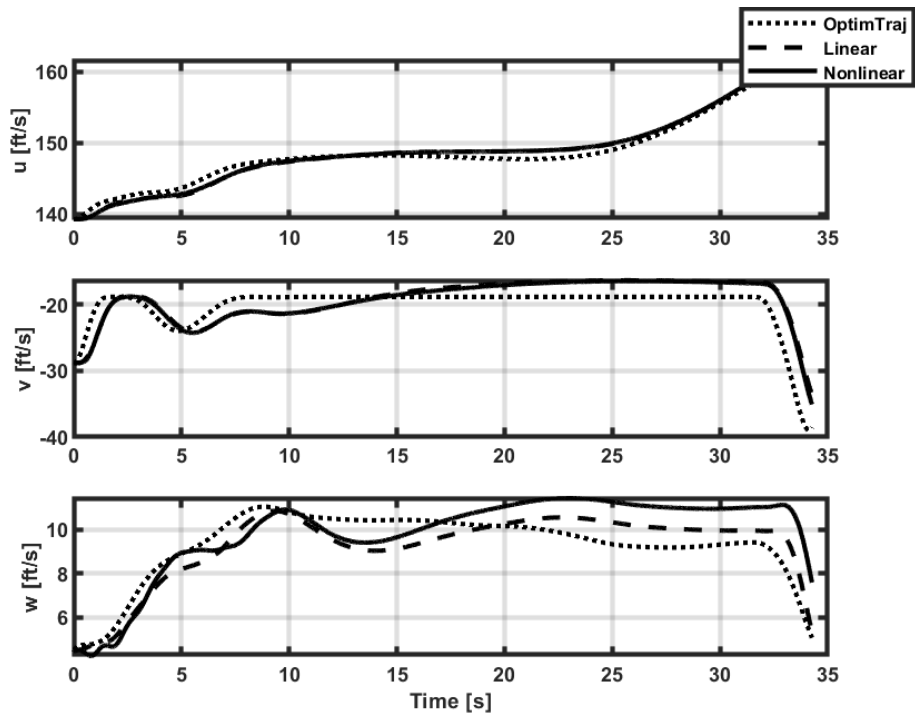


Figure 5.20. Body velocities of the helicopter for Case 5

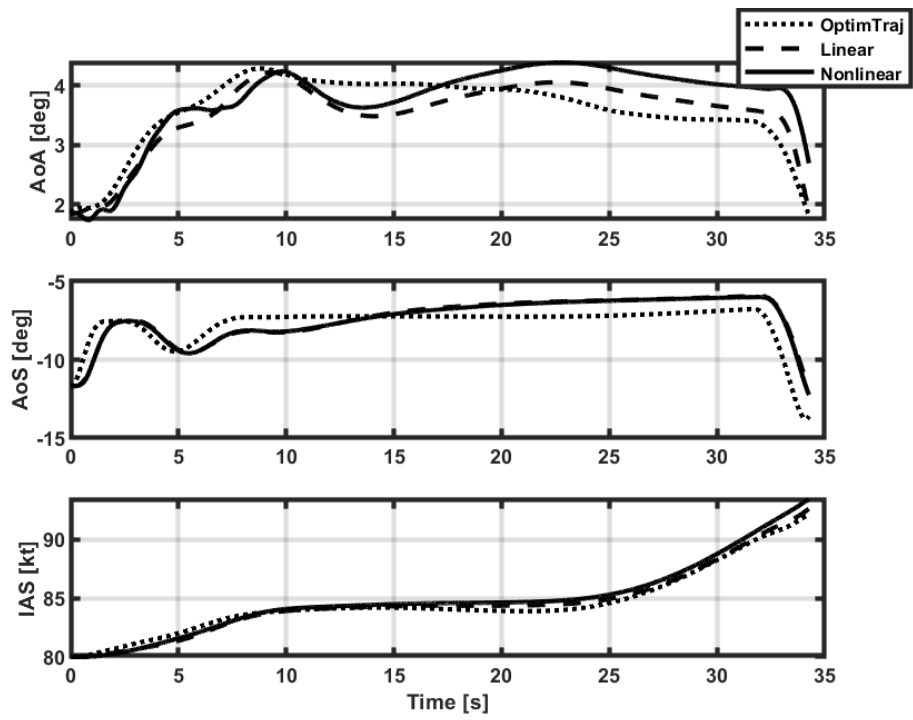


Figure 5.21. Air data parameters of the helicopter for Case 5

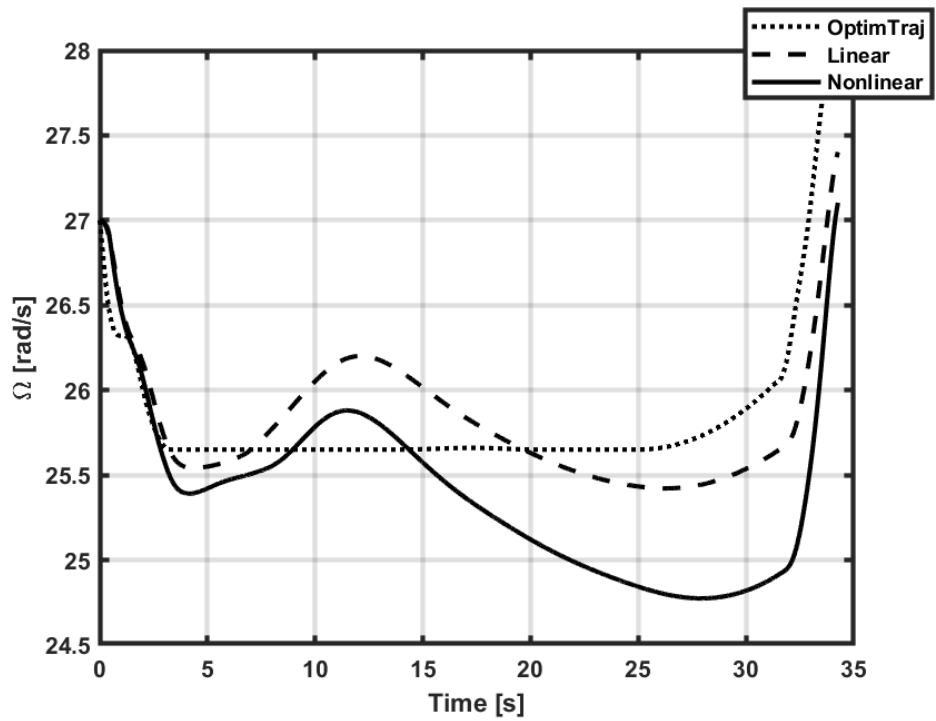


Figure 5.22. Main rotor speed of the helicopter for Case 5

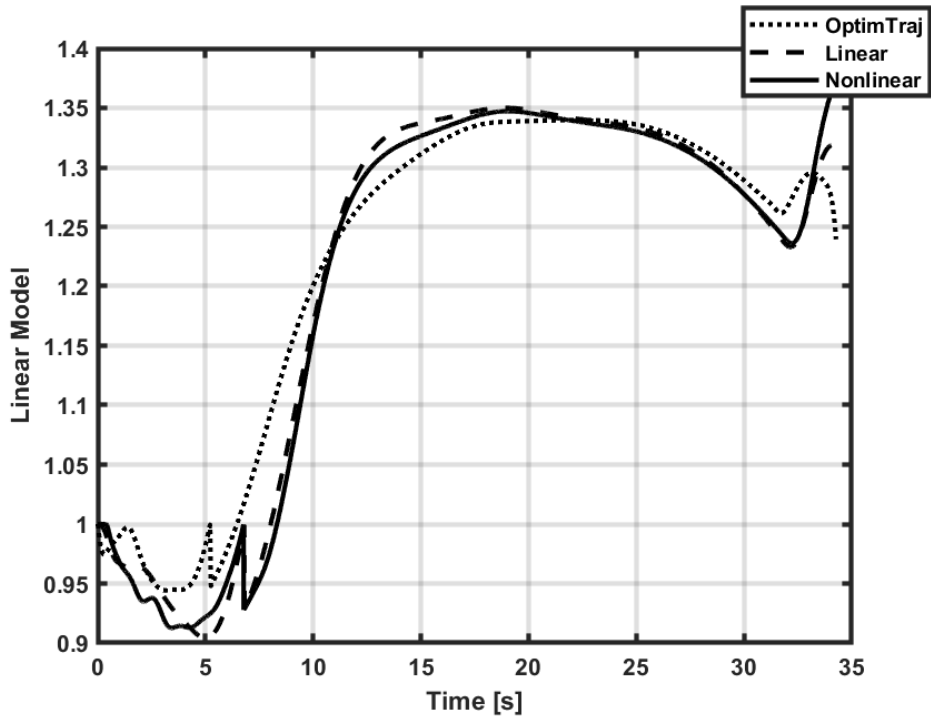


Figure 5.23. Linear model transition for Case 5 (0: 80TL, 1: 80L, 2: 80TR)

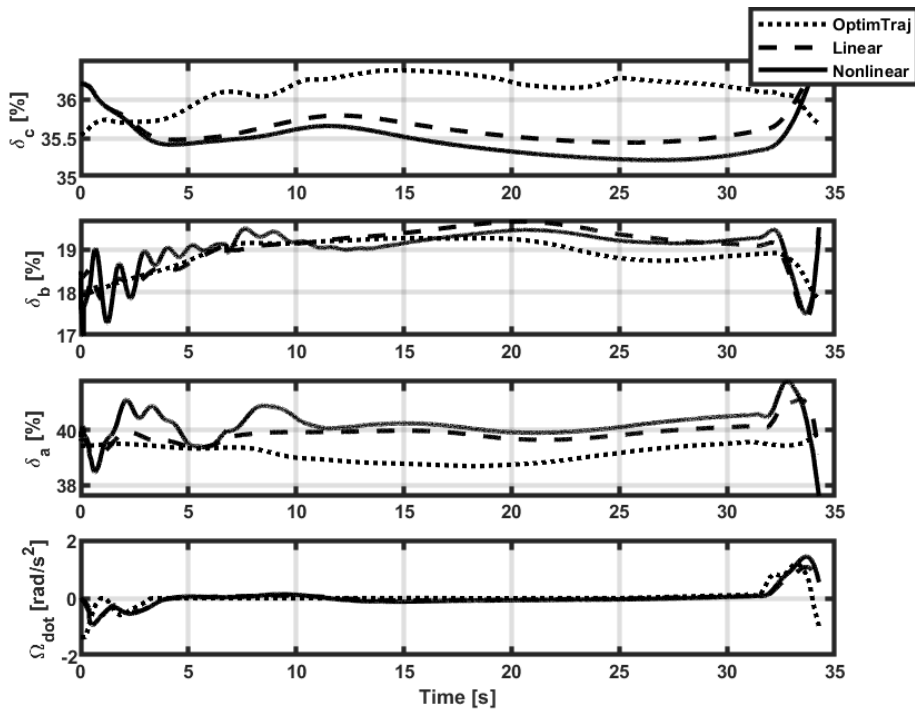


Figure 5.24. Helicopter inputs for Case 5

## 5.4 Results for Case 7

In Case 7, the transition of the helicopter from autorotation at 60 knots to autorotation at 80 knots is investigated. It should be reminded that the tracking set consists of  $\phi, \theta, u$  and  $w$  for this case.

The resulting flight path is presented in Figure 5.25 and the inertial position values are given in Figure 5.26. Here, it can be commented that the inertial x and z positions of both models are very close to the ones calculated by trajectory optimization. On the other hand, the inertial y positions of the models are quite different from OptimTraj.

The Euler angles and the body velocities are shown in Figure 5.27 and Figure 5.28, respectively. Although the controller could track the pitch angle, it fails to track the roll angle. Likewise, the controller is successful to track the longitudinal velocity components,  $u$  and  $w$ , but it fails to follow the lateral one. This situation leads to lost control in the lateral direction which was seen in the inertial y position deviation. However, this loss is intentional. Since the number of the elements of the tracking set is limited as explained in Chapter 4, controlling the longitudinal states is chosen.

The air data information is given in Figure 5.29. Similar inferences can be made here too. The rotor speed is presented in Figure 5.30. It is only regulated by the controller and reaching too high or too low values is prevented.

The transition between the linear models is shown in Figure 5.31. It is seen that both models acted similarly and closed to the optimal result. It is worth mentioning that although the helicopter is in perfect trim at 80 knots autorotation when the linear model value is 2, it is impossibly hard to reach this value. Therefore, achieving 1.9 or above for this parameter is considered a success. For this case, its value for the nonlinear model is 1.914.

Lastly, the helicopter inputs are given in Figure 5.32. Abrupt changes can be seen in optimal values due to large time step and the lack of actuator model in linear models.

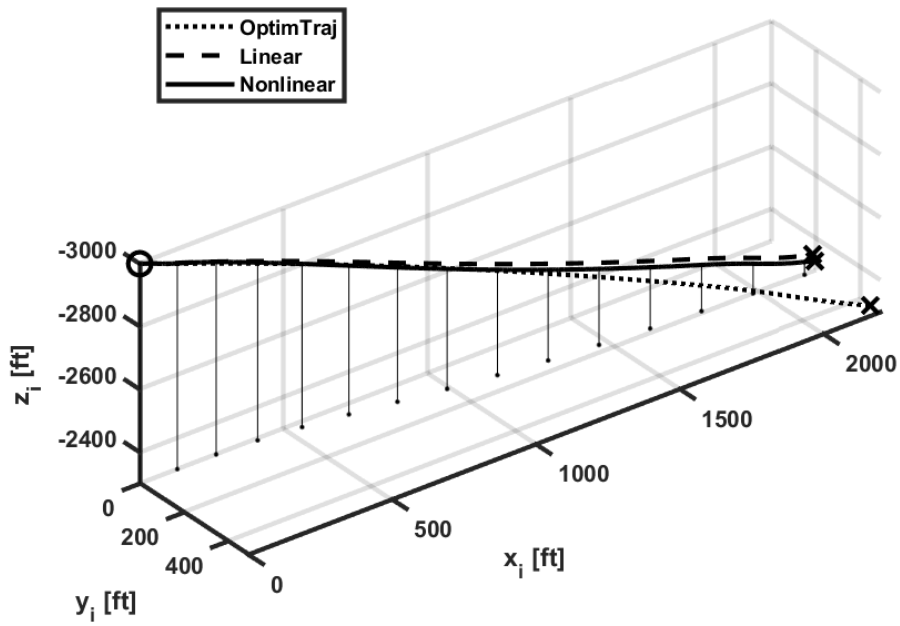


Figure 5.25. Trajectory of the helicopter for Case 7

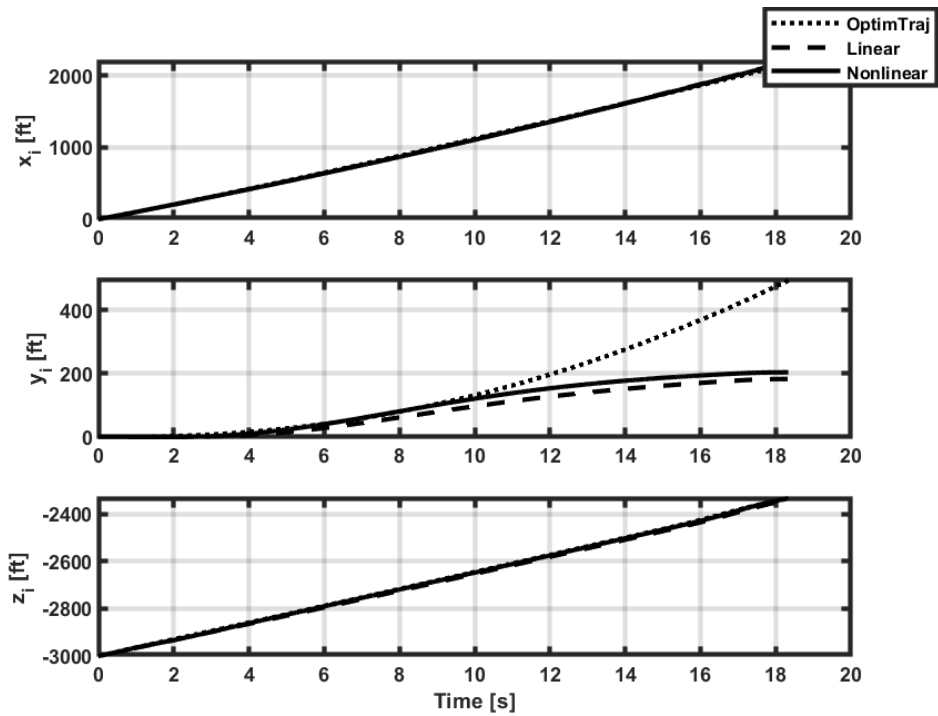


Figure 5.26. Inertial positions of the helicopter for Case 7



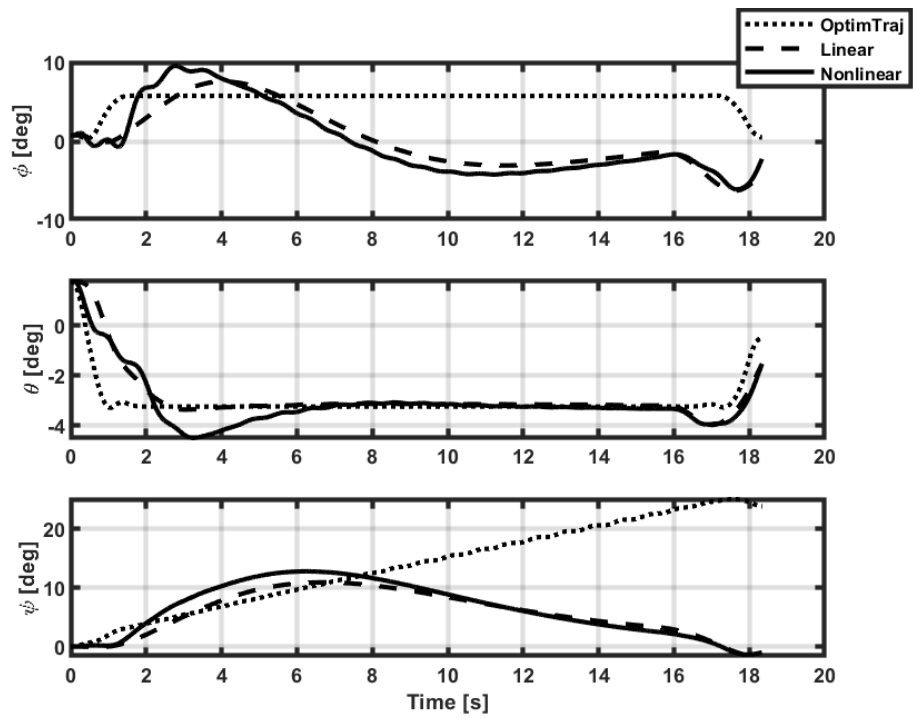


Figure 5.27. Euler angles of the helicopter for Case 7

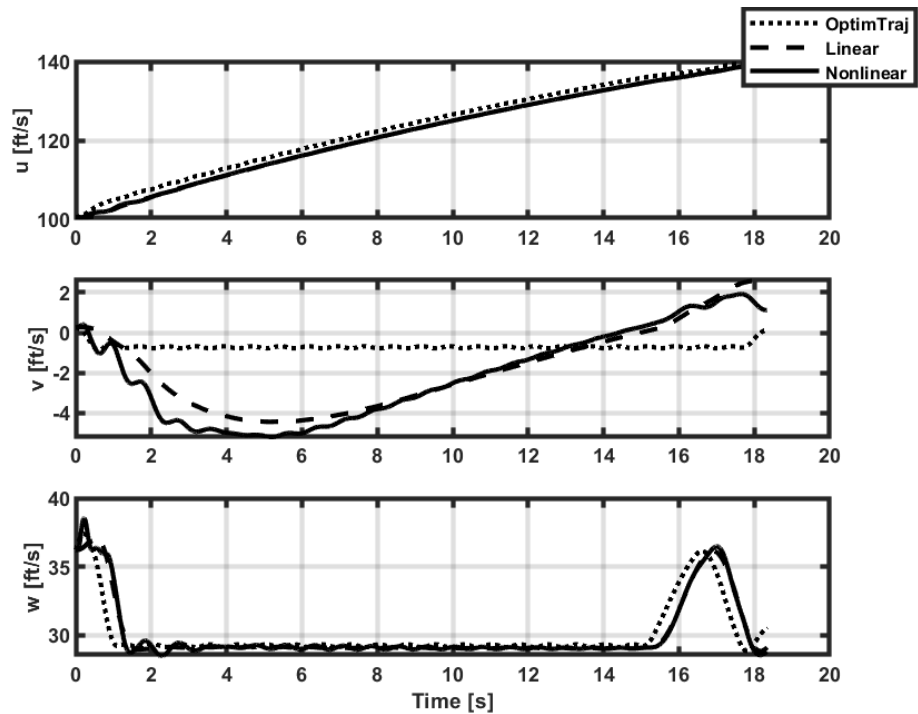


Figure 5.28. Body velocities of the helicopter for Case 7

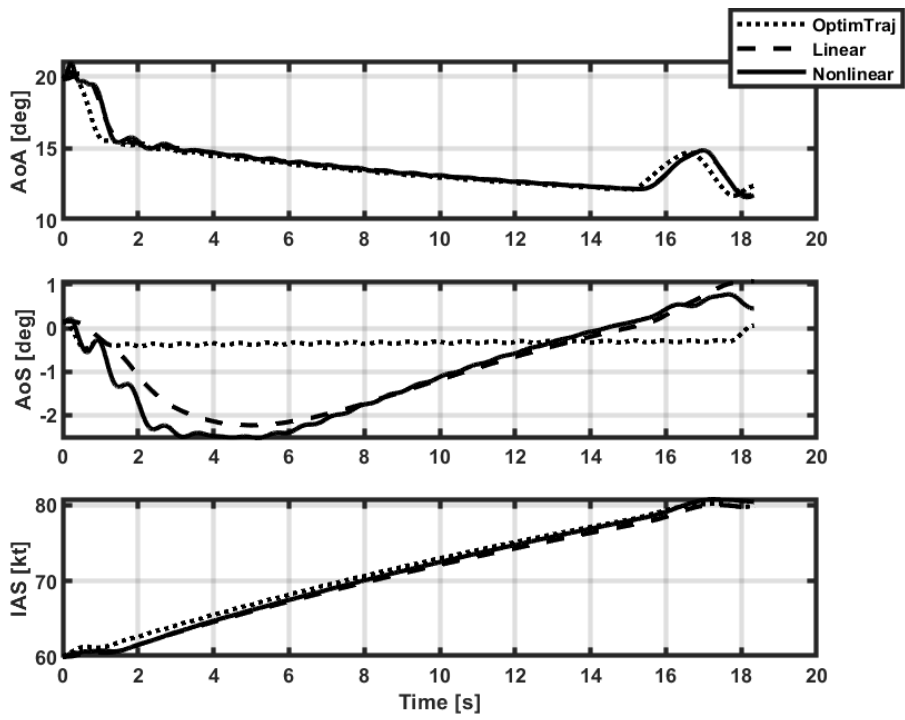


Figure 5.29. Air data parameters of the helicopter for Case 7

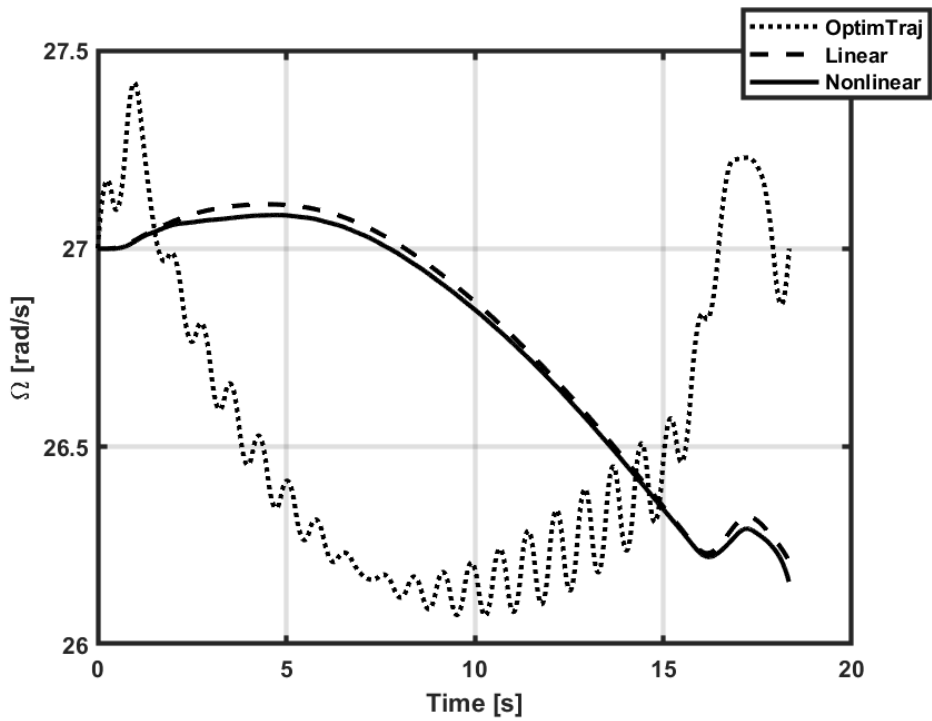


Figure 5.30. Main rotor speed of the helicopter for Case 7

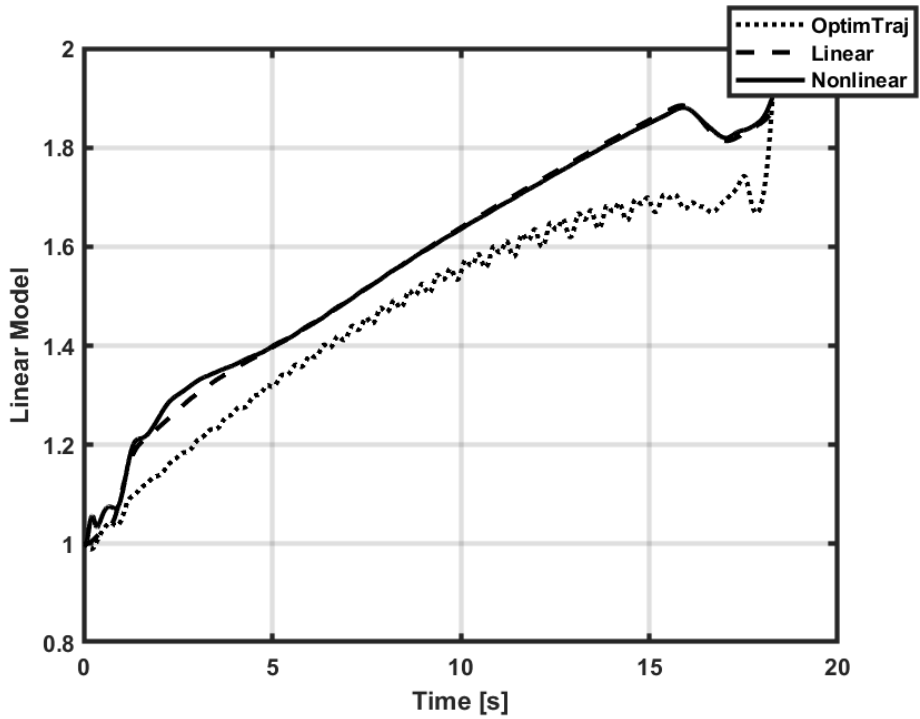


Figure 5.31. Linear model transition for Case 7 (1: 60A, 2: 80A)

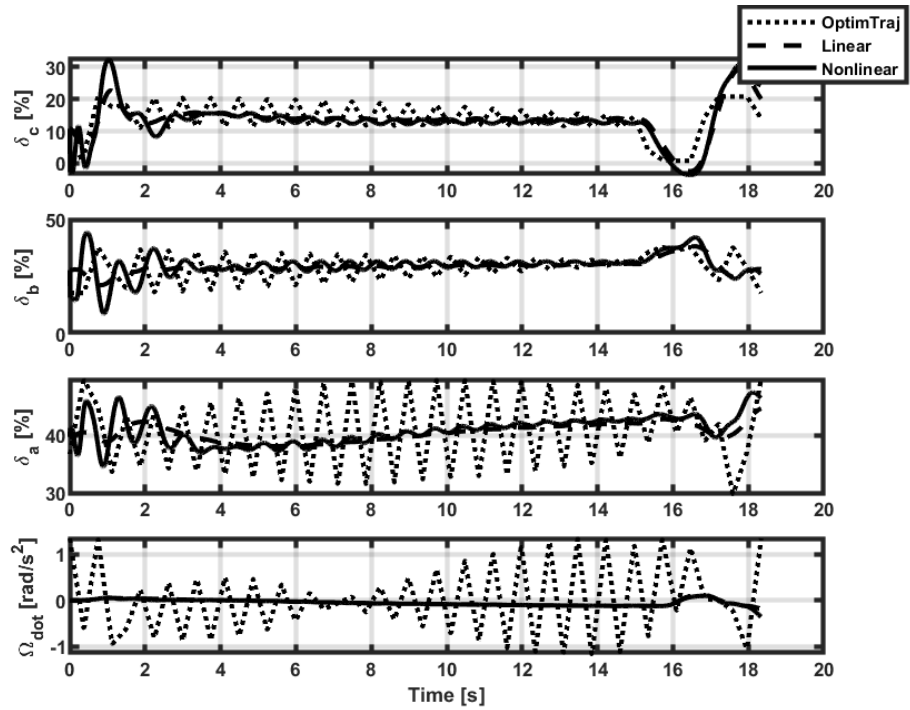


Figure 5.32. Helicopter inputs for Case 7

## 5.5 Results for Case 8

In Case 8, the transition of the helicopter from autorotation at 80 knots to level flight at 80 knots is investigated. It should be reminded that the tracking set consists of  $\phi$ ,  $\theta$ ,  $v$  and  $w$  for this case.

The resulting flight path is presented in Figure 5.33 and the inertial position values are given in Figure 5.34. Here, it can be commented that the inertial x and z positions of both models are close to the ones calculated by trajectory optimization. On the other hand, the inertial y positions of the models are a little different from OptimTraj.

The Euler angles and the body velocities are given in Figure 5.35 and Figure 5.36, respectively. It is hard to say that the controller could track the roll and pitch angles. On the contrary, the controller is successful to track the body velocities. In order to be precise on tracking of  $v$  and  $w$ , the weights of other states are lowered. That is why the angle tracking is poor.

The air data information is shown in Figure 5.37. It can be noticed that the necessary sideslip angle is achieved. The rotor speed is presented in Figure 5.38. It is only regulated by the controller and reaching too high or too low values is prevented.

The transition between the linear models is shown in Figure 5.39. It is seen that both models acted similarly and get closer to level flight trim like the optimal result. It is noted that although the helicopter is in perfect trim at 80 knots level flight when the linear model value is 2, it is impossibly hard to reach this value. Therefore, achieving 1.9 or above for this parameter is considered a success. For this case, its value for the nonlinear model is 1.908.

Lastly, the helicopter inputs are shown in Figure 5.40. Abrupt changes can be seen in optimal values because of the large time step and the lack of actuator model in the linear models. In simulations, this phenomenon is eliminated inherently thanks to existence of the controller and the actuator model.

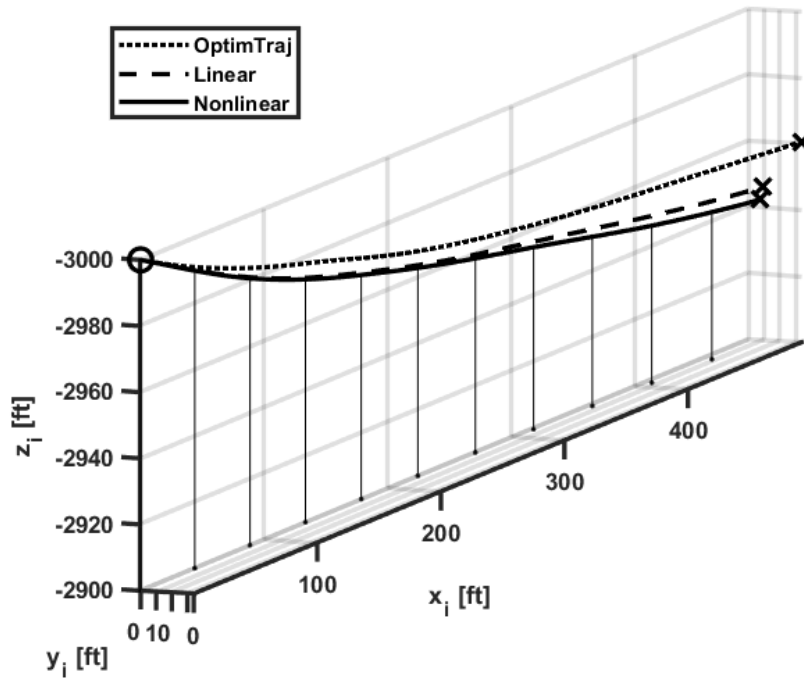


Figure 5.33. Trajectory of the helicopter for Case 8

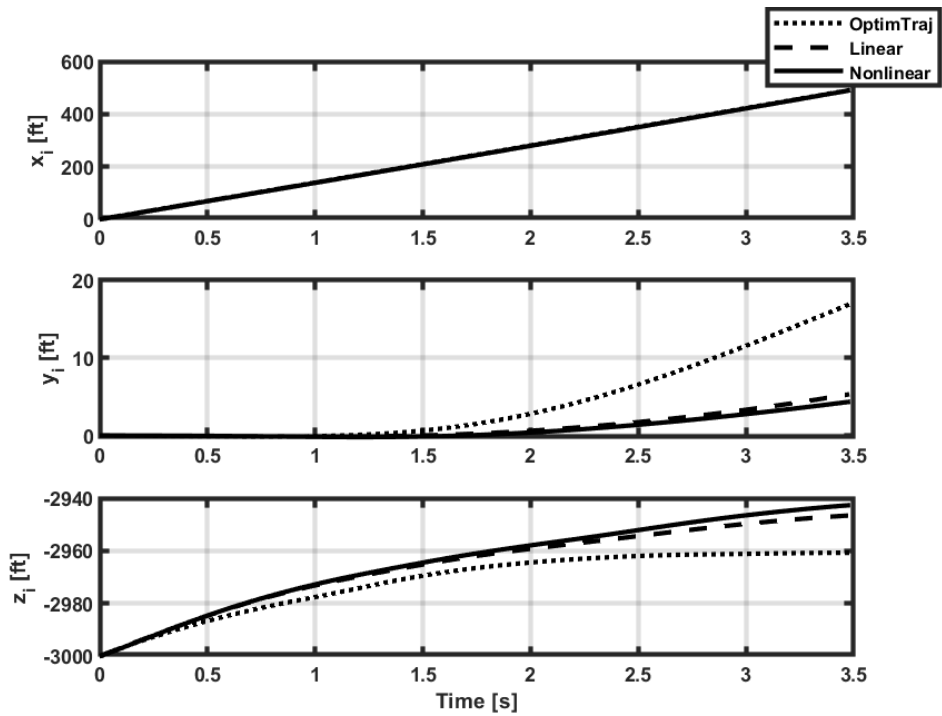


Figure 5.34. Inertial positions of the helicopter for Case 8

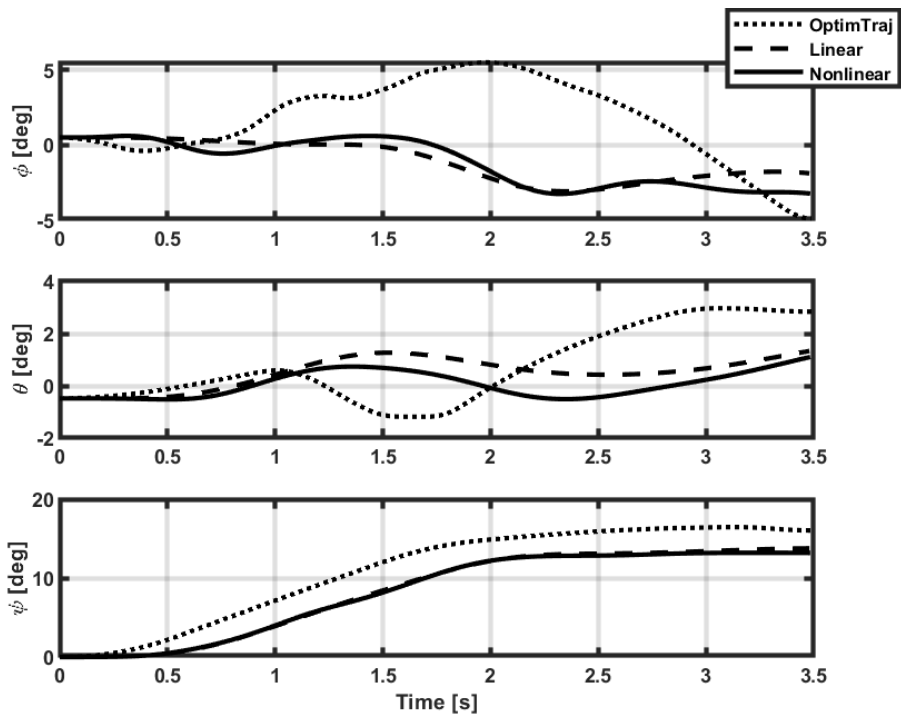


Figure 5.35. Euler angles of the helicopter for Case 8

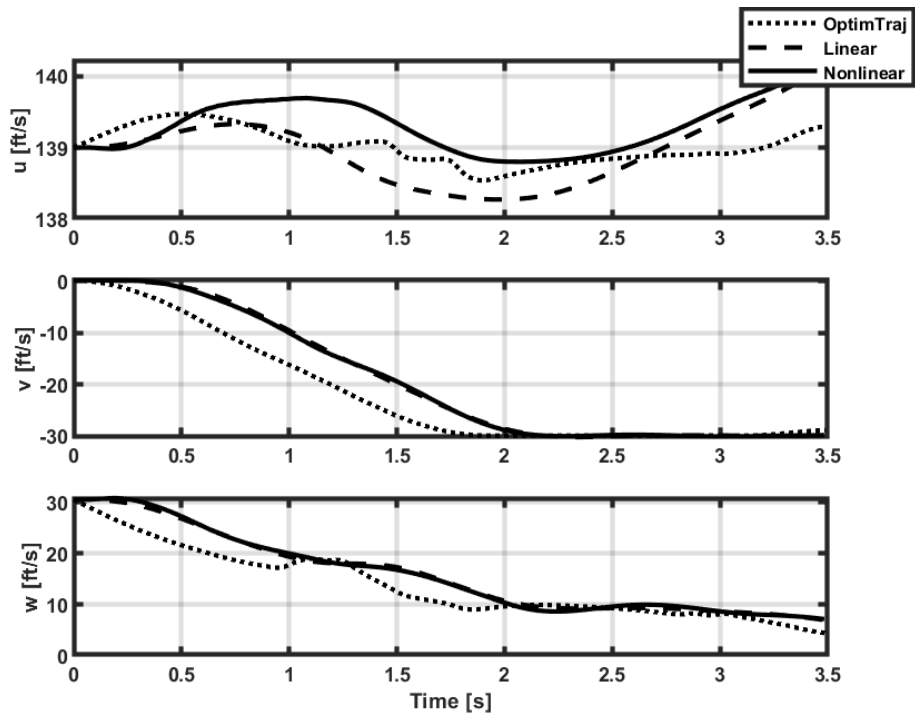


Figure 5.36. Body velocities of the helicopter for Case 8

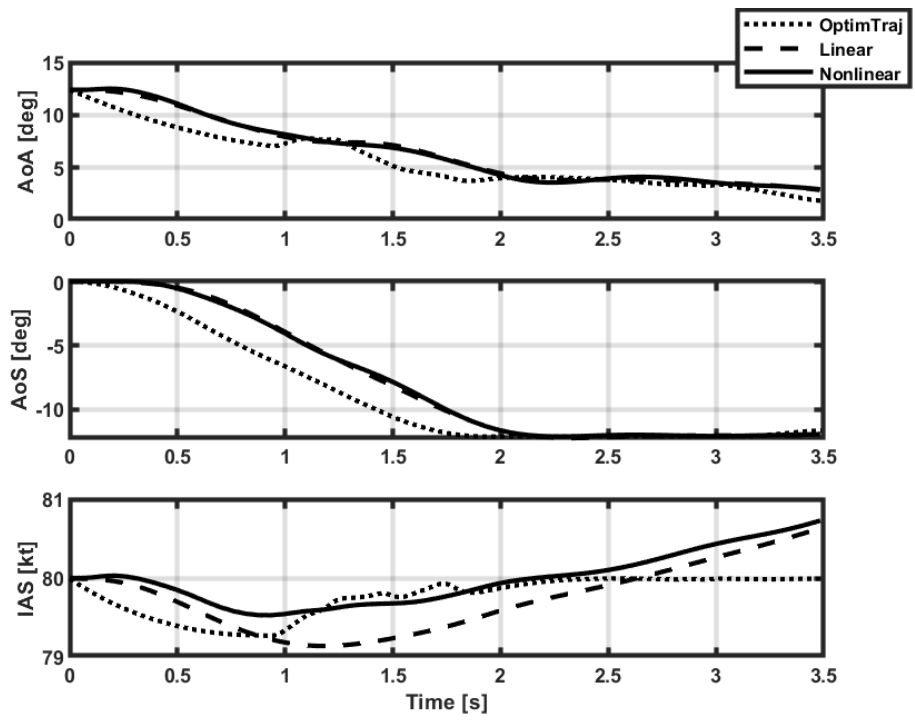


Figure 5.37. Air data parameters of the helicopter for Case 8

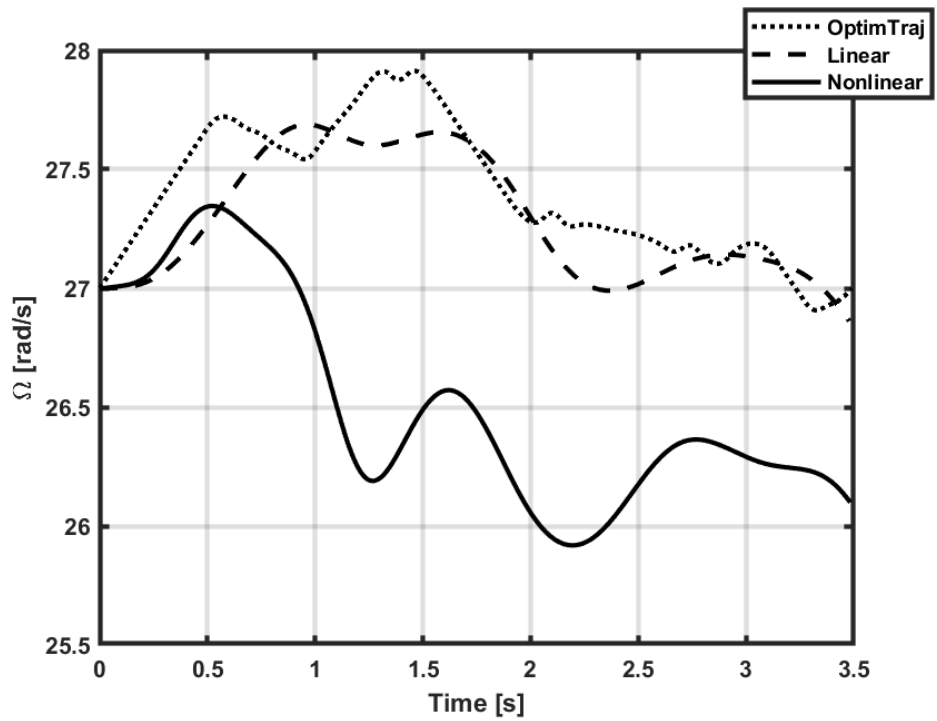


Figure 5.38. Main rotor speed of the helicopter for Case 8

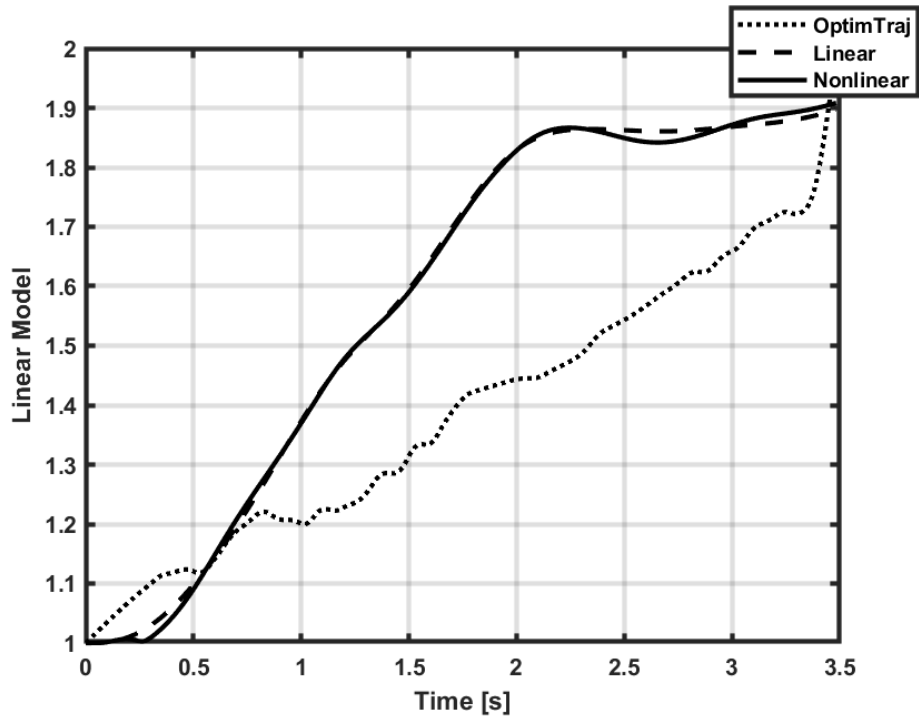


Figure 5.39. Linear model transition for Case 8 (1: 80A, 2: 80L)

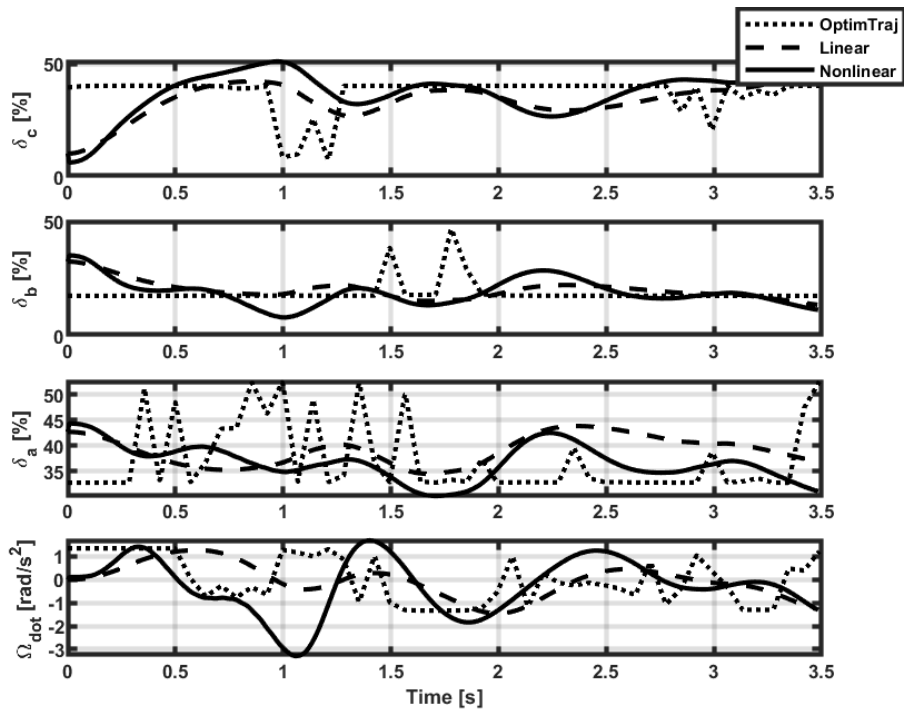


Figure 5.40. Helicopter inputs for Case 8



## CHAPTER 6

### CONCLUSION

In this thesis, trajectory generation and trajectory tracking for a helicopter in TRF is investigated. The study is started with the development of the helicopter mathematical model. The generic model in FLIGHTLAB which is a representative sample model of the UH-60A Black Hawk helicopter is utilized by making some simplifications on the model. Mechanical mixing on pilot controls, pitch bias controller, SAS, and horizontal stabilator control are turned off in order to reduce the complexity of tracking controller design. In the generic model, tail rotor failure can be injected by a switch that terminates the force and moment produced by the tail rotor. With the help of the comprehensive analysis feature of FLIGHTLAB, after trim points are found for several flight conditions, the linear models belonging to those trim points are produced in state-space form by applying linearization. Furthermore, the nonlinear model with the investigated initial conditions is created as a black box model in order to use for run-time simulations.

The mathematical model shows that there are two possibilities in a tail rotor failure condition. These possibilities are dependent on the helicopter's forward speed. If the helicopter has enough forward speed, it can be stabilized in level flight condition thanks to its vertical tail. On the other hand, if the forward speed is too low, the helicopter can be stabilized with autorotative flight; otherwise, it would spin around due to a need for an anti-torque. It should be noted that trim points where the altitude is preserved are obtained for the spinning around case; however, it is not included in the study.

In this thesis, for directional control, changing the main rotor speed is utilized because of the absence of pedal control. Therefore, the derivative of the rotor speed is added to the conventional control matrix and the rotor speed itself is added to the

state matrix in the linear models. This is the main contribution of the thesis to the literature.

Next, the trajectory generation is performed for the helicopter in TRF. In order to solve the trajectory optimization problem, OptimTraj is used by defining the objective function, the constraints, and bounds on states and controls. For the system dynamics, linearized helicopter models for different flight conditions are implemented. In order to widen the equilibrium point neighborhood, the linear models are interpolated according to the helicopter states. OptimTraj uses the collocation method by dividing the time sequence into grid points to solve a trajectory optimization problem. The trajectories are generated for the helicopter in level flight to fly to a specified inertial position with minimum control usage and for the helicopter in autorotation to change its trim condition with minimum descent.

After the reference trajectories are obtained, two tracking controllers are designed for the position tracking in level flight and the state tracking for trim transition. For the position tracking, dynamic inversion and linear model inversion are implemented as cascaded structure in the first controller. For the state tracking, a linear quadratic regulator with integral action method is used in the second controller. In this controller, the tracked states are defined individually for each case because only four states can be tracked. In addition, interpolation of the linear models is performed in both controllers to increase their accuracy.

Finally, the results are presented with optimized trajectories by OptimTraj and tracking simulations conducted by linear and nonlinear models. 8 scenarios have been investigated. The five of them belong to position control of the helicopter in TRF for level flight. The helicopter is commanded to fly to different target points. It is seen that the helicopter could achieve longitudinal and lateral positions; however, it fails when it is commanded to climb. The performance of the controller is fine when altitude change is not commanded, and the performance is much better for low altitude command.

The other scenarios are the state control of the helicopter in TRF for trim transition. Transitions from 40 knots autorotation to 60 knots autorotation, from 60 knots autorotation to 80 knots autorotation and from 80 knots autorotation to 80 knots level flight are studied. Although there is limited authority on the states, it is accepted that the helicopter could successfully move from one trim point to another. This study also estimates the necessary altitude for the helicopter to be able to stay in the air if a tail rotor failure happens.

## **6.1 Future Work**

There are several different topics for possible future work. First of all, different trim conditions can be defined between the coordinated turn trim points. Thus, the precision of model interpolation would increase, and using more accurate system dynamics would positively affect all the phases of this study.

Secondly, in the trajectory optimization process, although a feasible solution can be found, sometimes it suffers from the sudden changes in the controls. In addition, increasing grid number makes the situation worse because it becomes impossible to respond to the helicopter to the obtained controls. Therefore, for such a case, the number of grids is kept relatively low, sacrificing the integral precision. In order to overcome this problem, a simple actuator model can be implemented in the linear system dynamics model during trajectory generation.

Next, different control methods might be tried. Although the controllers have shown sufficient performance to prove the concept of this thesis, they were not perfect. More sophisticated control methods should be implemented.

And finally, even though 8 scenarios have been investigated in this thesis, different cases should be included and on-line application should be studied in order to handle tail rotor failure condition in real life and to achieve safer flights in the future.



## REFERENCES

- [1] W. Johnson, *Rotorcraft Aeromechanics*, New York: Cambridge University Press, 2013.
- [2] G. D. Padfield, *Helicopter Flight Dynamics: The Theory and Application of Flying Qualities and Simulation Modelling*, Blackwell Publishing, 2007.
- [3] M. Kelly, "An Introduction to Trajectory Optimization: How to Do Your Own Direct Collocation," *SIAM Review*, no. 59, pp. 849-904, January 2017.
- [4] F. D. Harris, E. F. Kasper and L. E. Iseler, "U.S Civil Rotorcraft Accidents 1963 Through 1997," NASA TM-2000-209597, 2000.
- [5] R. D. Garcia and A. Brown, "Control and Limitations of Navigating a Tail Rotor/Actuator Failed Unmanned Helicopter," *Journal of Intelligent & Robotic Systems*, vol. 61, no. 1-4, pp. 5-13, 2011.
- [6] T. Yomchinda, J. F. Horn and J. W. Langelaan, "Autonomous Control and Path Planning for Autorotation of Unmanned Helicopters," in *AHS International*, 2012.
- [7] G. Chowdhary, E. N. Johnson, R. Chandramohan, M. S. Kimbrell and A. Calise, "Guidance and Control of Airplanes Under Actuator Failures," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 4, 2013.
- [8] C. L. Bottasso, C.-S. Chang, A. Croce, D. Leonello and L. Riviello, "Adaptive planning and tracking of trajectories for the simulation," *Computer Methods in Applied Mechanics and Engineering*, vol. 195, no. 50-51, pp. 7052-7072, 2006.
- [9] R. D. Garcia, K. P. Valavanis and A. Kandel, "Autonomous Helicopter Navigation During A Tail Rotor Failure Utilizing Fuzzy Logic," in *Mediterranean Conference on Control and Automation*, Athens, 2007.

- [10] A. Budiyo and S. S. Wibowo, "Optimal Tracking Controller Design for a Small Scale Helicopter," *Journal of Bionic Engineering*, vol. 4, no. 4, p. 271–280, 2007.
- [11] P. Abbeel, A. Coates, T. Hunter and A. Y. Ng, "Autonomous Autorotation of an RC Helicopter," in *Experimental Robotics: The Eleventh International Symposium*, 2008.
- [12] R. Enns and J. Si, "Helicopter Flight-Control Reconfiguration for Main Rotor Actuator Failures," *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 4, 2003.
- [13] J. Rogers, B. Eberle, M. Jump and N. Cameron, "Time-to-Contact-Based Control Laws for Flare Trajectory Generation and Landing Point Tracking in Autorotation," in *AHS International*, Phoenix, 2018.
- [14] P. Bibik and J. Narkiewicz†, "Helicopter Optimal Control After Power Failure Using Comprehensive Dynamic Model," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 4, 2012.
- [15] S. Rajendran and D.-w. Gu, "Fault Tolerant Control of a Small Helicopter with Tail Rotor Failures in Hovering Mode," in *11th IEEE International Conference on Control & Automation*, Taichung, 2014.
- [16] M. E. Serrano, D. C. Gandolfo and G. . J. Scaglia, "Trajectory Tracking Controller for Unmanned Helicopter Under Environmental Disturbances," *ISA Transactions*, vol. 106, pp. 171-180, 2020.
- [17] S. Taamallah, X. Bombois and P. M. Van den Hof, "Trajectory Planning and Trajectory Tracking for A Small-scale Helicopter in Autorotation," *Control Engineering Practice*, vol. 58, pp. 88-106, 2017.

- [18] Y. Wang, C. Jiang, Y. Zhang and H. Wang, "Helicopter Safe Landing Trajectory after Main Rotor Actuator Failures," *Applied Sciences*, vol. 10, no. 8, p. 2917, 2020.
- [19] D. J. Lee, H. Bang and K. Baek, "Autorotation of an Unmanned Helicopter by a Reinforcement Learning Algorithm," in *AIAA Guidance, Navigation and Control Conference and Exhibit*, Honolulu, 2008.
- [20] M. Ballin, "Validation of A Real-time Engineering Simulation of the UH-60A Helicopter," NASA TM-88360, 1987.
- [21] W. Johnson, "Rotorcraft Dynamics Models for a Comprehensive Analysis," in *American Helicopter Society 54th Annual Forum*, Palo Alto, 1998.
- [22] Yavrucuk İlkey, O. Tarımcı, M. Katırcıoğlu, E. Kubalı and E. Yılmaz, "A New Helicopter Simulation and Analysis Tool: HELIDYN+," in *36th European Rotorcraft Forum*, Paris, 2010.
- [23] O. A. Bauchau, "DYMORE : A Finite Element Based Tool for the Analysis of Nonlinear Flexible Multibody Systems," in *Georgia Institute of Technology*, Atlanta, 2001.
- [24] P. D. Talbot, B. E. Tinling, W. A. Decker and R. T. N. Chen, "A Mathematical Model of A Single Main Rotor Helicopter for Piloted Simulation," NASA-TM-84281, 1982.
- [25] R. K. Heffly and M. A. Mnich, "Minimum Complexity Helicopter Simulation Math Model," NASA CR 177476, 1988.
- [26] C. R. Frost, W. S. Hindson, E. Morales and G. E. Tucker, "Design and Testing of Flight Control Laws on the RASCAL Research Helicopter," in *AIAA Modeling and Simulation Technologies Conference and Exhibit*, Monterey, 2002.

- [27] Advanced Rotorcraft Technology, "The Company Overview," [Online]. Available: <https://www.flightlab.com/>. [Accessed 2022].
- [28] R. W. D. Val, "A Real-time Multi-body Dynamics Architecture for Rotorcraft Simulation," in *Proceedings of the RAeS conference*, London, 2001.
- [29] C. He and J. Zhao, "Modeling Rotor Wake Dynamics with Viscous Vortex," *AIAA Journal*, vol. 47, no. 4, 2009.
- [30] K. B. Hilbert, "A Mathematical Model of the UH-60 Helicopter," NASA TM 85890, 1984.
- [31] J. Currie and D. I. Wilson, "Opti: Lowering the Barrier Between Open Source Optimizers and the Industrial MATLAB User," in *Conference: Foundations of Computer-Aided Process Operations*, Savannah, 2012.
- [32] J. Andersson, J. Gillis, G. Horn, J. B. Rawlings and M. Diehl, "CasADi – A Software Framework for Nonlinear Optimization," *Mathematical Programming Computation*, vol. 11, pp. 1-36, 2019.
- [33] M. A. Patterson and A. V. Rao, "GPOPS-II: A MATLAB Software for Solving Multiple-Phase Optimal Control Problems Using hp-Adaptive Gaussian Quadrature Collocation Methods and Sparse Nonlinear Programming," *ACM Transactions on Mathematical Software*, vol. 1, no. 1, pp. 1-37, 2014.
- [34] J. K. Moore and A. J. van den Bogert, "opty: Software for Trajectory Optimization and Parameter Identification Using Direct Collocation," *Journal of Open Source Software*, vol. 3, no. 21, p. 300, 2018.
- [35] M. Diehl and S. Gros, *Numerical Optimal Control*, University of Freiburg, 2017.
- [36] M. P. Kelly, *OptimTraj Users Guide*, 2018.



- [37] Anon., *ADS-33E-PRF, Aeronautical Design Standard Performance Specification: Handling Qualities Requirements for Military Rotorcraft*, 2000.
- [38] R. T. Rysdyk and A. J. Calise, "Adaptive Model Inversion Flight Control for Tilt-Rotor Aircraft," *Journal of Guidance, Control, and Dynamics*, vol. 22, no. 3, 1999.
- [39] A. J. Calise and R. T. Rysdyk, "Nonlinear Adaptive Flight Control Using Neural Networks," *IEEE Control Systems Magazine*, vol. 18, no. 6, pp. 14-25, 1998.
- [40] S. Avcioğlu, *Controller Design and Simulation for A Helicopter During Target Engagement*, MSc. Thesis, Middle East Technical University, 2011.
- [41] S. Das and K. Halder, "Missile Attitude Control via a Hybrid LQG-LTR-LQI Control Scheme with Optimum Weight Selection," in *First International Conference on Automation, Control, Energy and Systems (ACES)*, Adisaptagram, 2014.
- [42] W. J. Rugh and J. S. Shamma, "Research on Gain Scheduling," *Automatica*, vol. 36, no. 10, pp. 1401-1425, 2000.
- [43] J. V. R. Prasad, A. J. Calise, Y. Pei and J. E. Corban, "Adaptive Nonlinear Controller Synthesis and Flight Test Evaluation on An Unmanned Helicopter," in *IEEE International Conference on Control Applications*, Kohala Coast, 1999.
- [44] O. Tarımcı, *Adaptive Controller Applications for Rotary Wing Aircraft Models of Varying Simulation Fidelity*, MSc. Thesis, Middle East Technical University, 2009.
- [45] B. Etkin, *Dynamics of Atmospheric Flight*, Dover Publications Inc., 2005.

[46] R. L. Williams II and D. A. Lawrence, *Linear State-Space Control Systems*, John Wiley & Sons, Inc., 2007.

## APPENDICES

### A. Linear Model-Nonlinear Model Response Comparison

In this appendix, the model comparisons which are not given in Model Development part are presented. Linear and nonlinear model responses for 1 unit change in collective, longitudinal cyclic, lateral cyclic and derivative of rotor speed are shown for autorotation trims at 40, 60 and 80 knots.

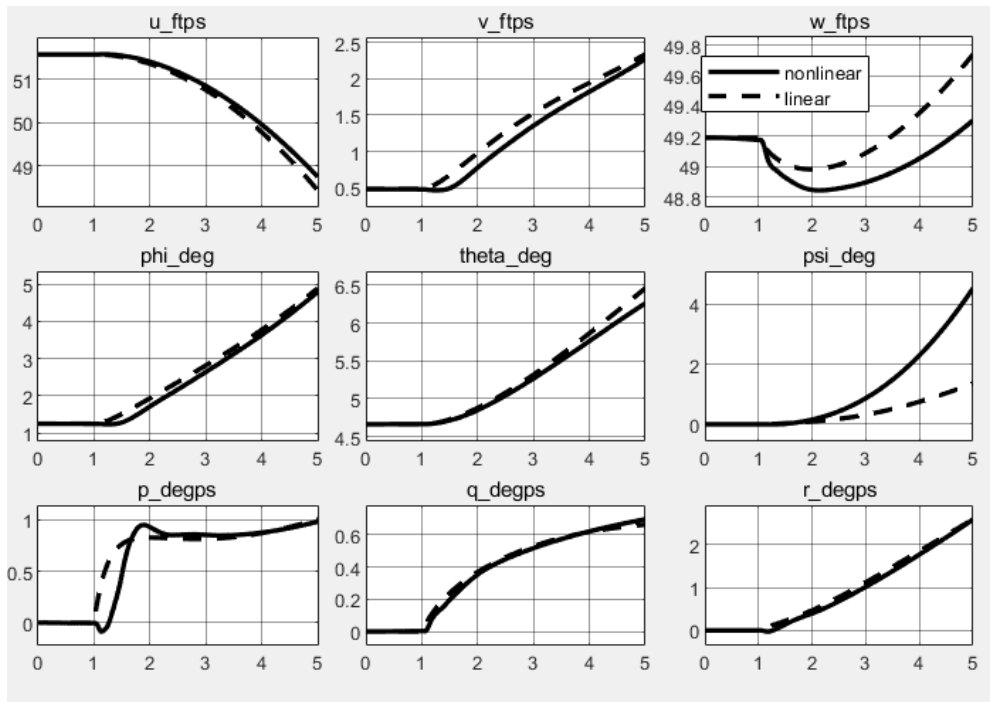


Figure A.1. Linear-nonlinear response to collective comparison for autorotation at 40 knots

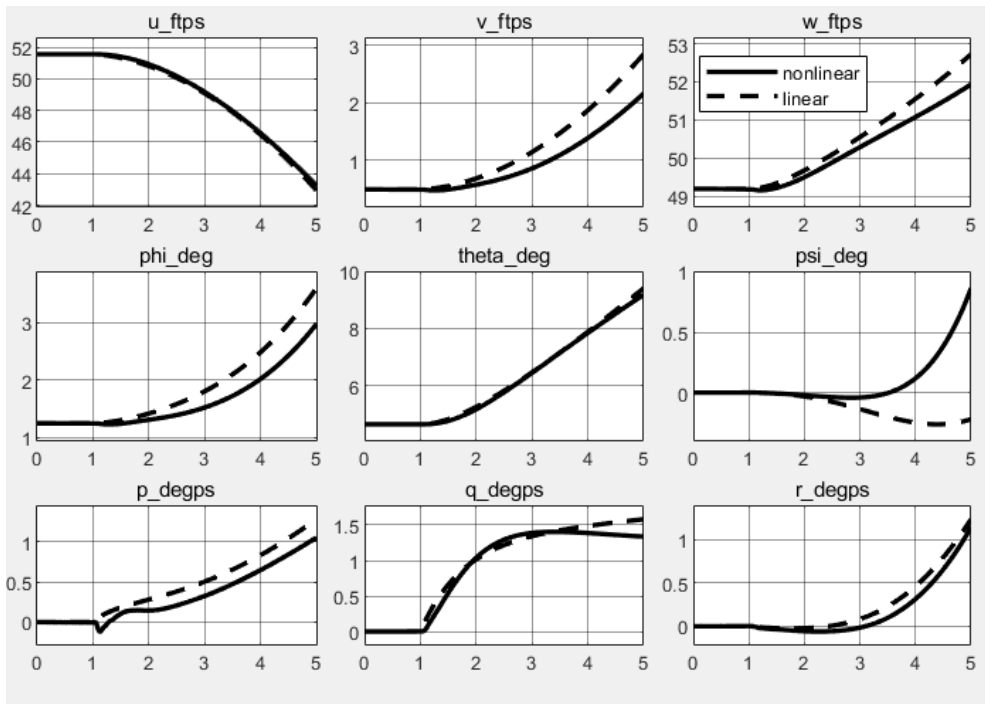


Figure A.2. Linear-nonlinear response to longitudinal cyclic comparison for autorotation at 40 knots

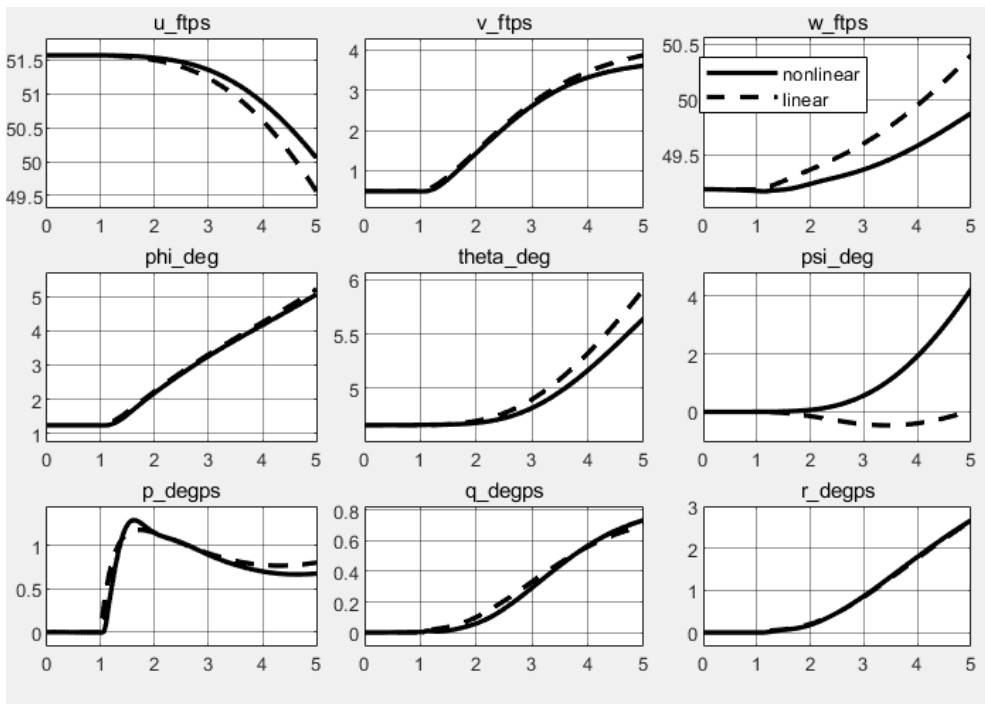


Figure A.3. Linear-nonlinear response to lateral cyclic comparison for autorotation at 40 knots

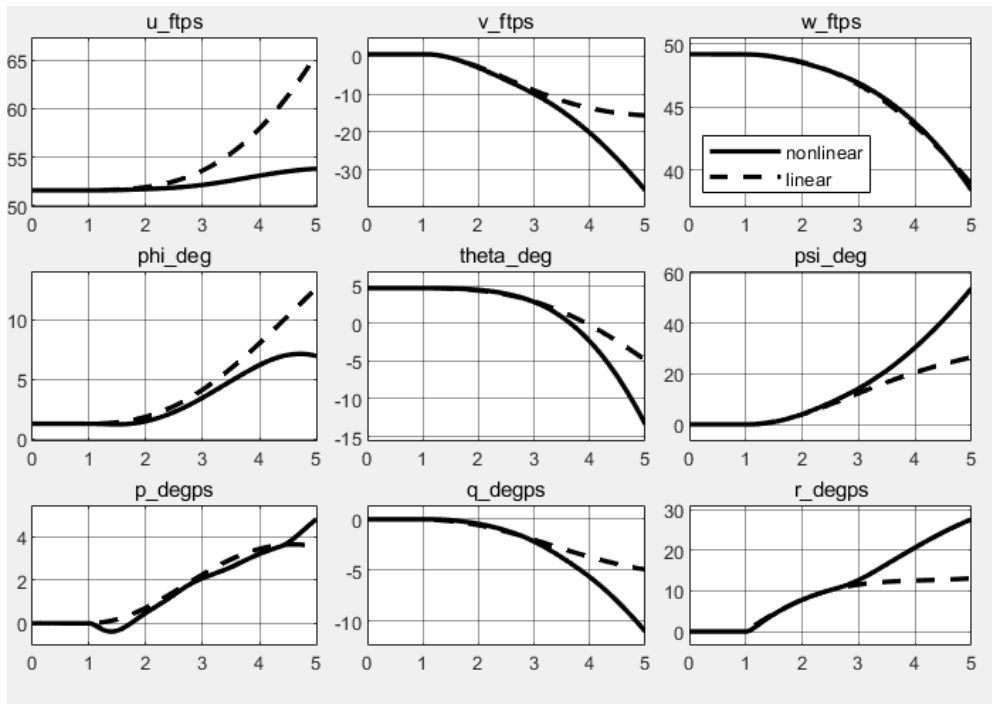


Figure A.4. Linear-nonlinear response to change in rotor speed comparison for autorotation at 40 knots

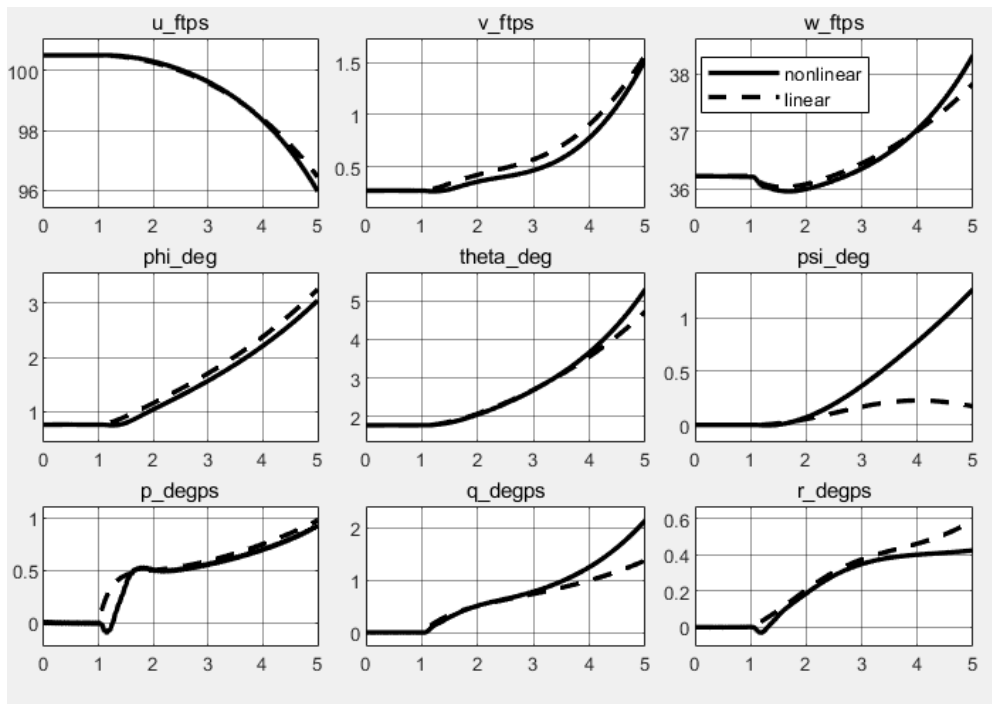


Figure A.5. Linear-nonlinear response to collective comparison for autorotation at 60 knots

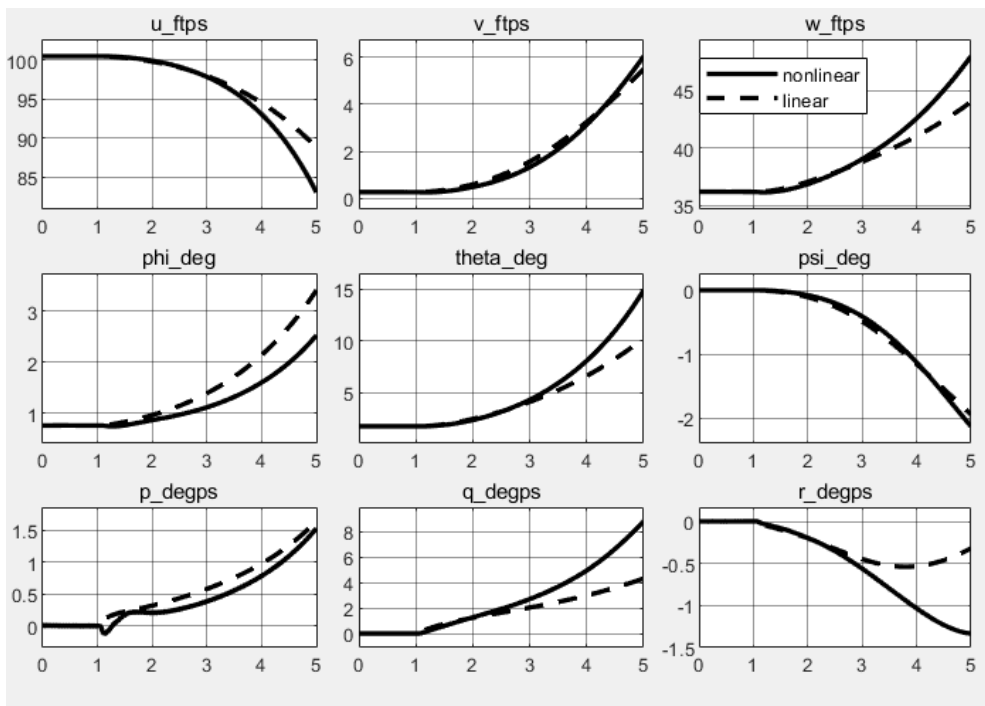


Figure A.6. Linear-nonlinear response to longitudinal cyclic comparison for autorotation at 60 knots

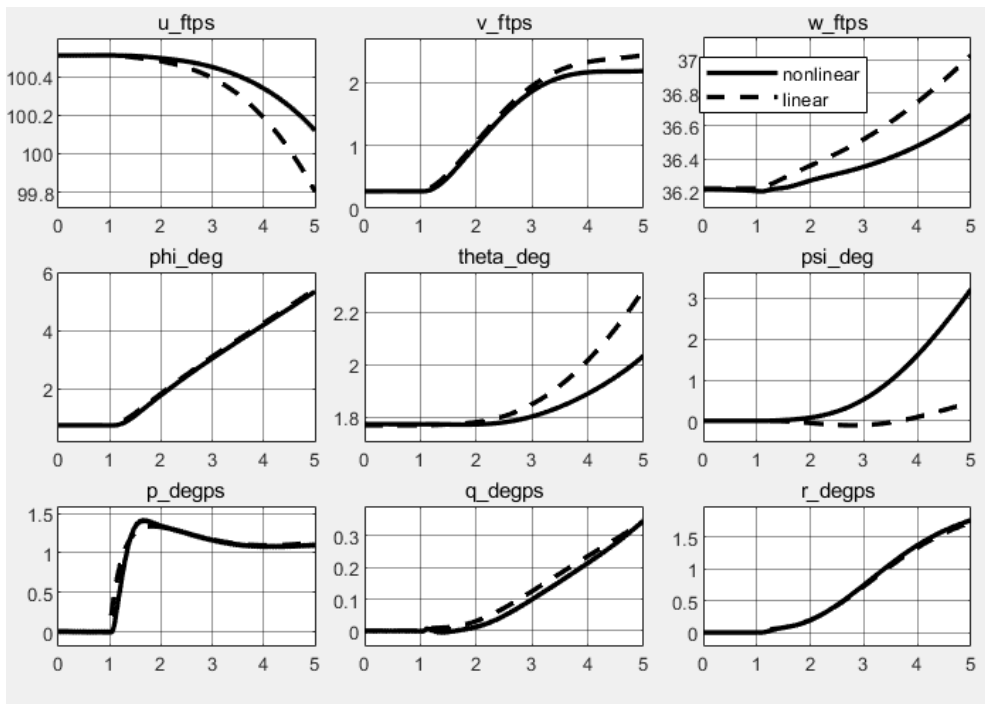


Figure A.7. Linear-nonlinear response to lateral cyclic comparison for autorotation at 60 knots

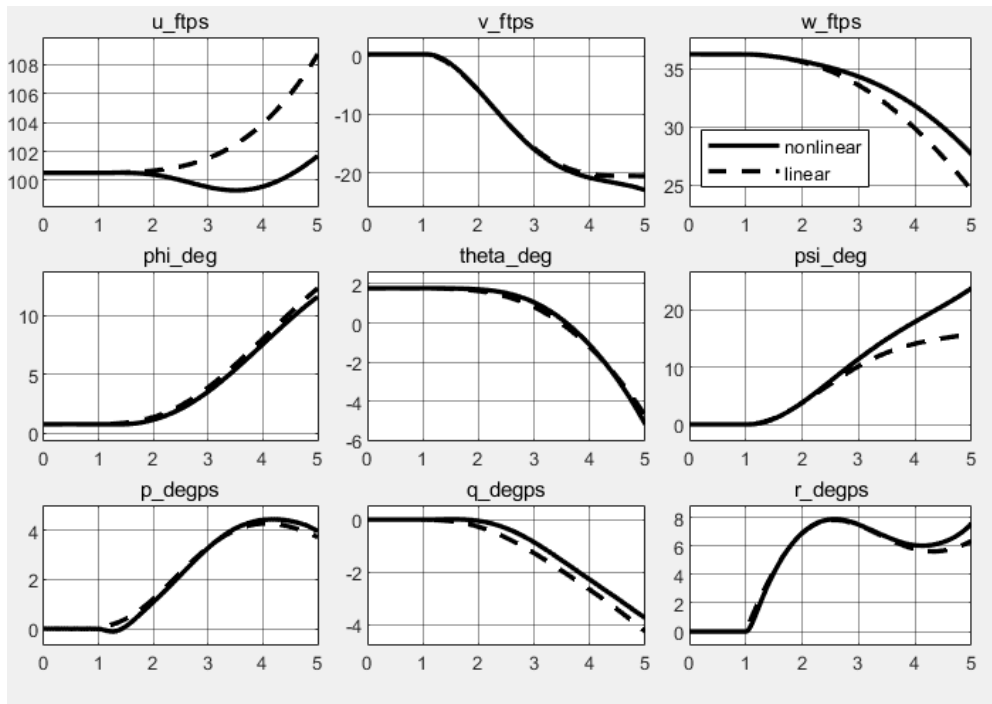


Figure A.8. Linear-nonlinear response to change in rotor speed comparison for autorotation at 60 knots

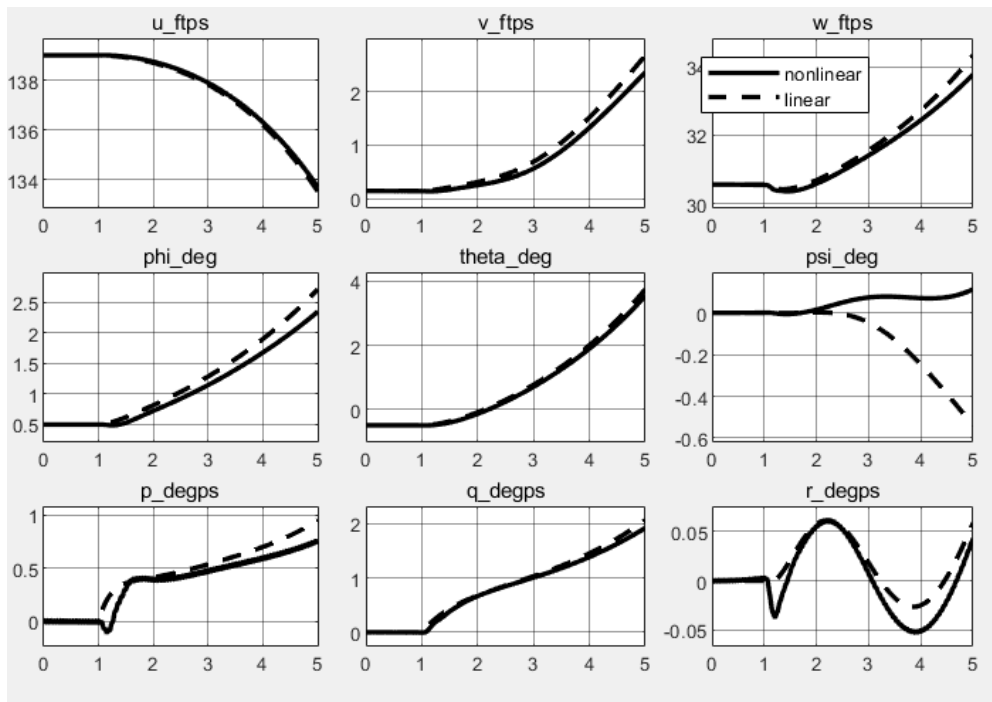


Figure A.9. Linear-nonlinear response to collective comparison for autorotation at 80 knots

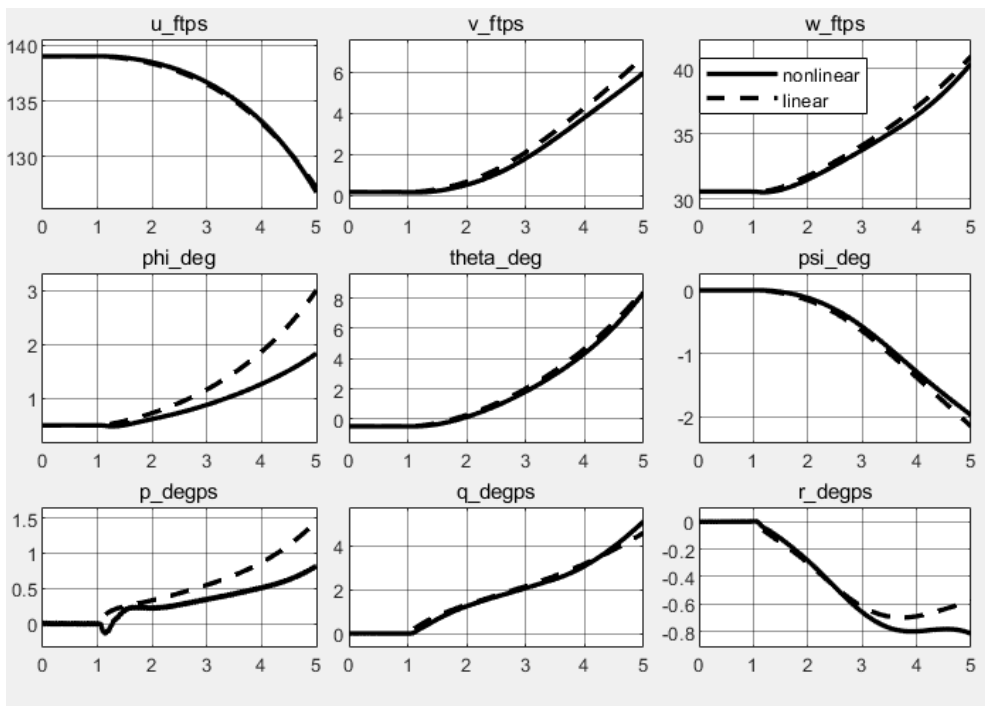


Figure A.10. Linear-nonlinear response to longitudinal cyclic comparison for autorotation at 80 knots

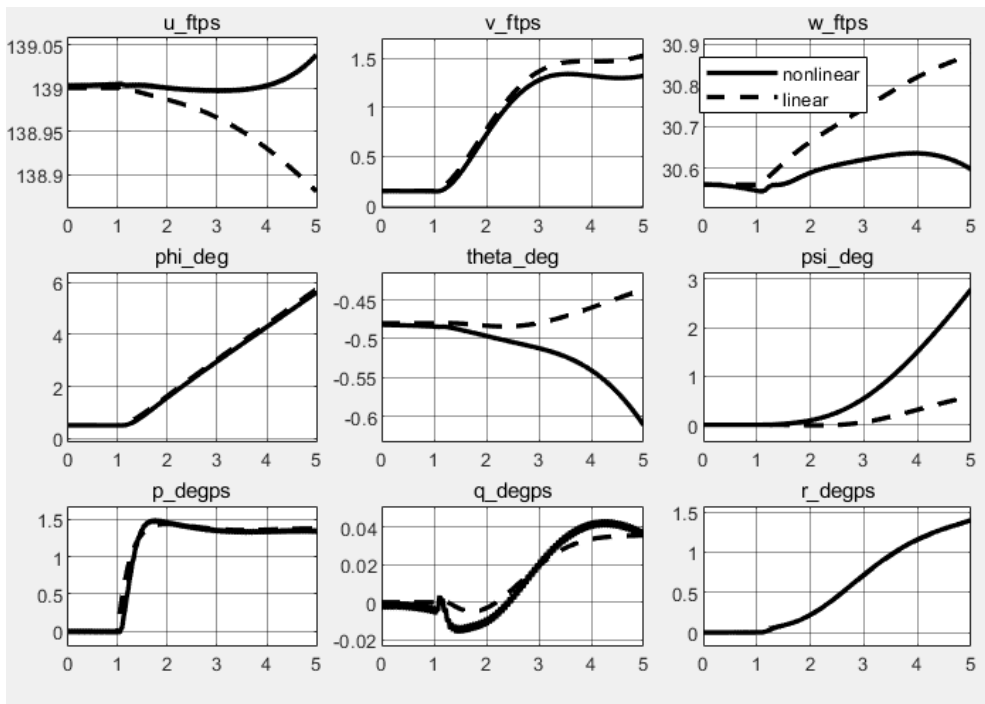


Figure A.11. Linear-nonlinear response to lateral cyclic comparison for autorotation at 80 knots



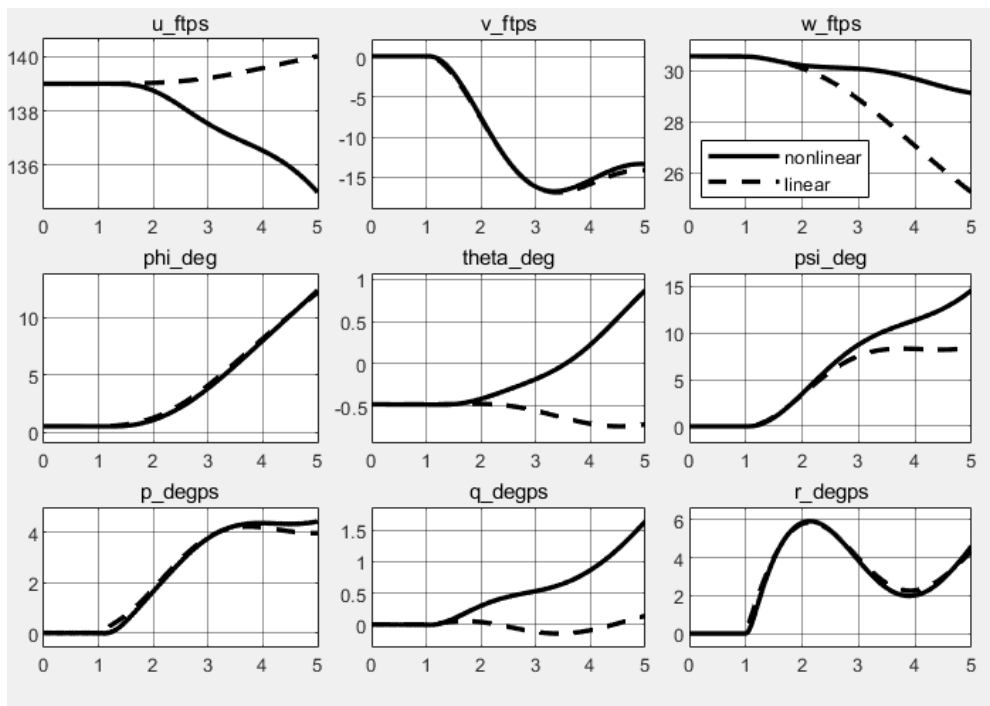


Figure A.12. Linear-nonlinear response to change in rotor speed comparison for autorotation at 80 knots

## B. Weight Matrices of LQI Controller

In this appendix, weight matrices of the LQI controller are given in (B.1), (B.2) and (B.3) which are implemented in Case 6, Case 7 and Case 8, respectively. Note that the value which penalizes the pedal input is chosen to be very high so that the controller does not send command. All matrices are selected to be diagonal.

$$Q = [10000; 0; 0; 0; 3; 0; 5730; 5730; 5730; 11460; 57.3; 200; 200] [I_{13 \times 13}]$$
$$R = [0.1; 0.1; 0.1; 1e10; 100] [I_{5 \times 5}] \quad (\text{B.1})$$

$$Q = [10000; 0; 0; 0; 1; 0; 0.573; 5.73; 0.573; 28.65; 11460; 100; 100] [I_{13 \times 13}]$$
$$R = [0.1; 0.1; 0.1; 1e10; 1000] [I_{5 \times 5}] \quad (\text{B.2})$$

$$Q = [10; 10; 0; 0; 0; 0; 5.73; 5.73; 0.573; 28.65; 5730; 100; 100] [I_{13 \times 13}]$$
$$R = [0.1; 0.1; 0.1; 1e10; 1] [I_{5 \times 5}] \quad (\text{B.3})$$

### C. Results for Case 2, Case 4 and Case 6

In this appendix, the trajectory generation and the tracking simulation results are presented for Case 2, Case 4 and Case 6 which are not given in Results and Discussion chapter.

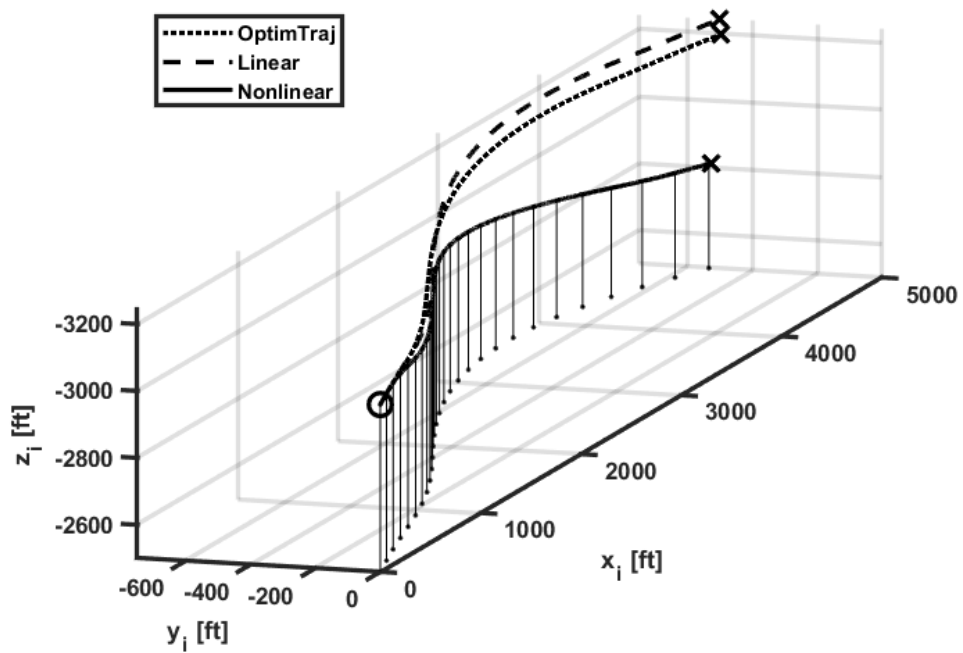


Figure C.13. Trajectory of the helicopter for Case 2

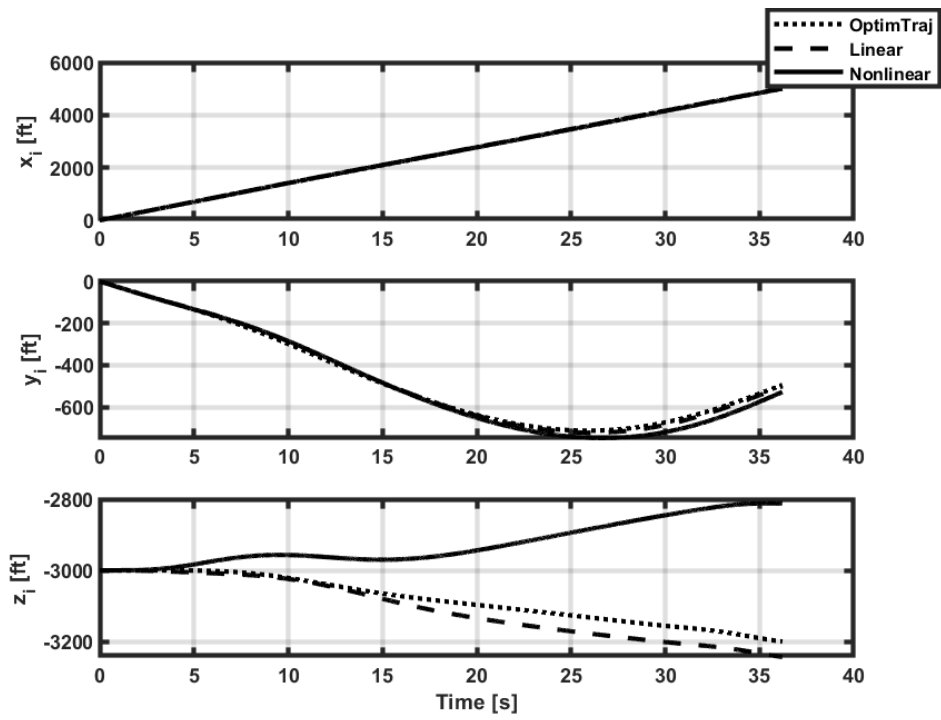


Figure C.14. Inertial positions of the helicopter for Case 2

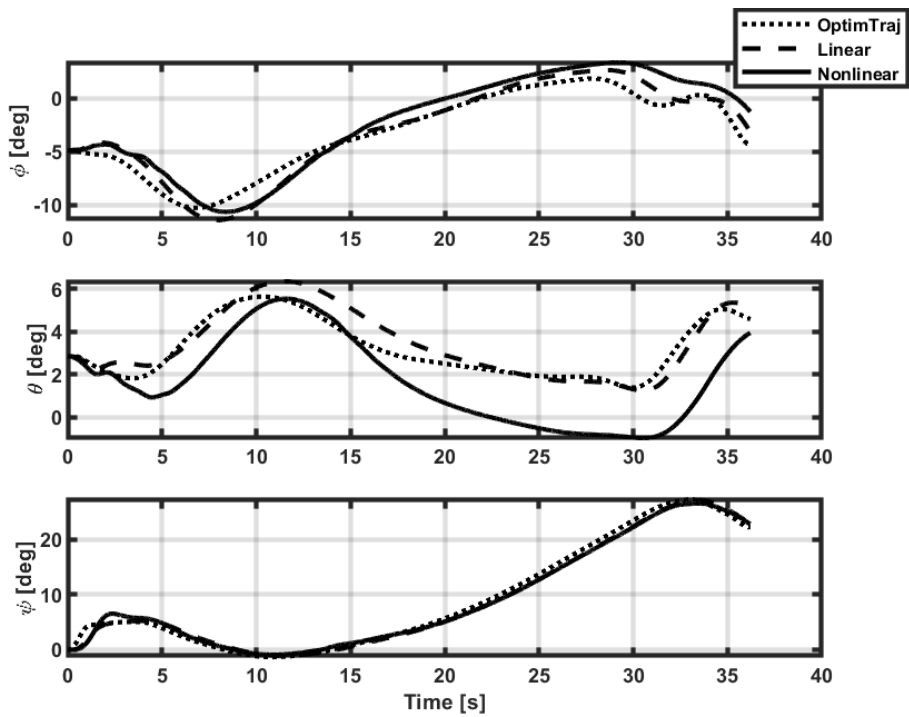


Figure C.15. Euler angles of the helicopter for Case 2

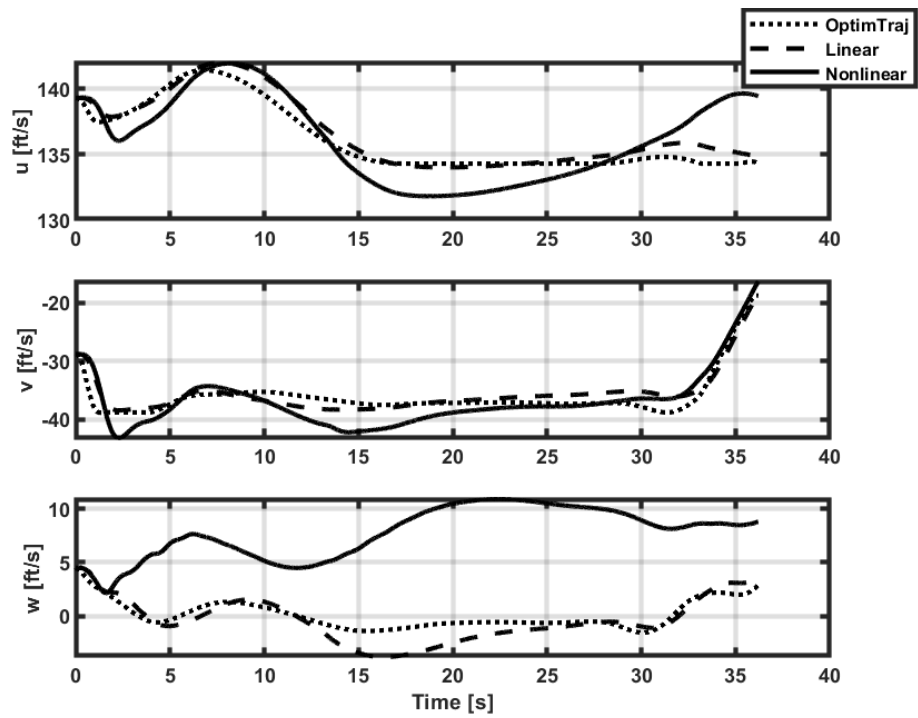


Figure C.16. Body velocities of the helicopter for Case 2

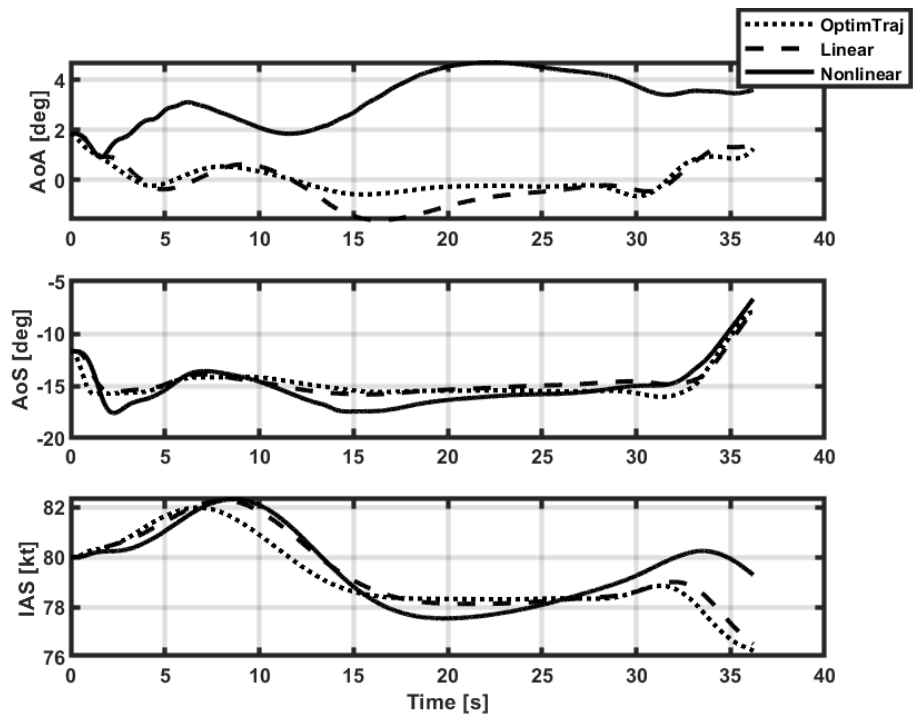


Figure C.17. Air data parameters of the helicopter for Case 2

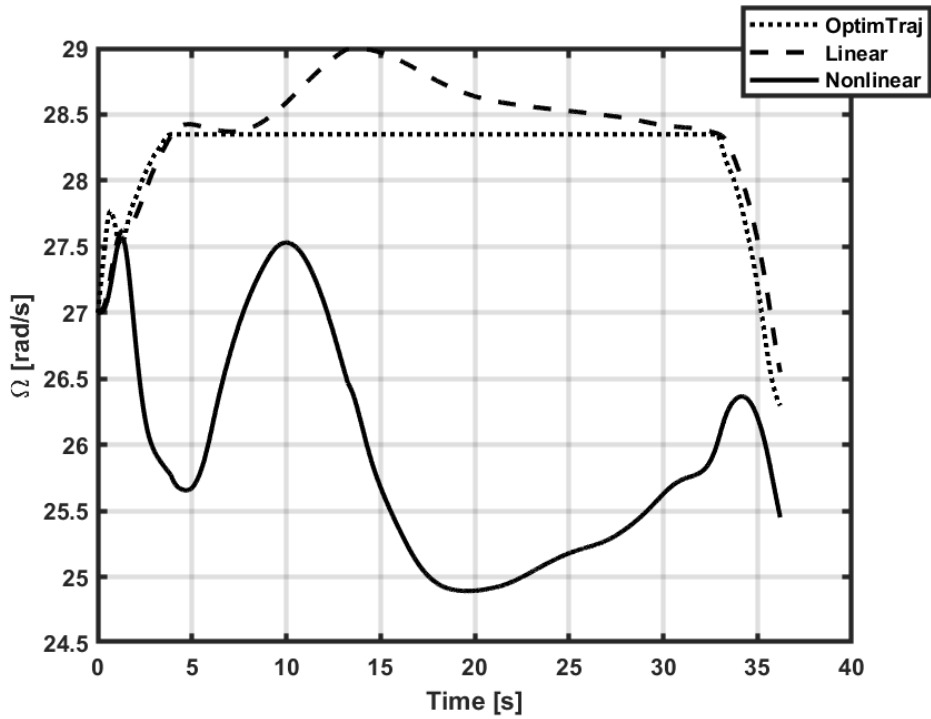


Figure C.18. Main rotor speed of the helicopter for Case 2

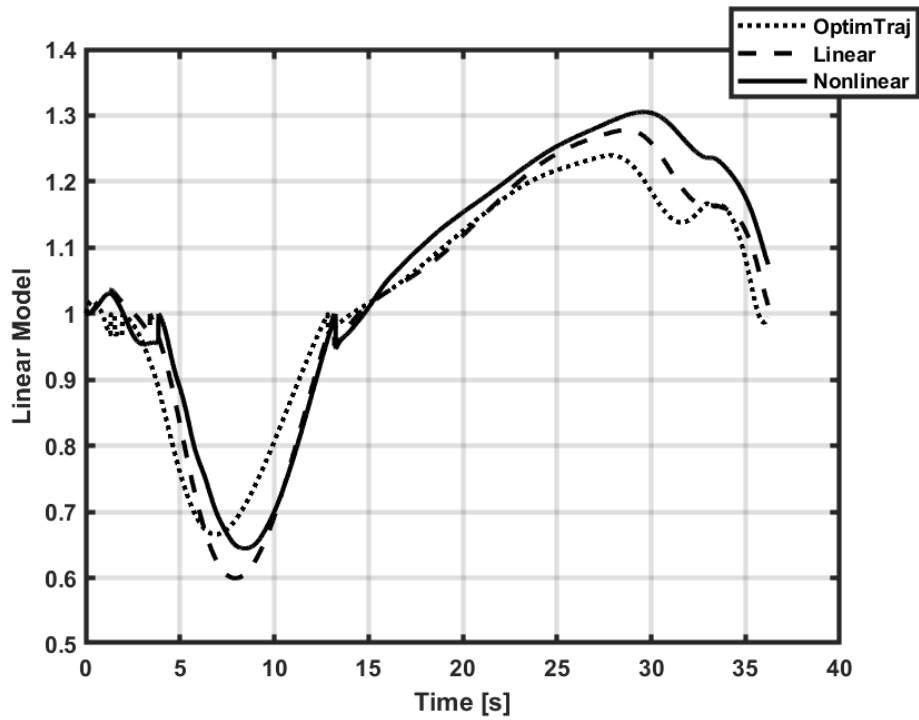


Figure C.19. Linear model transition for Case 2 (0: 80TL, 1: 80L, 2: 80TR)

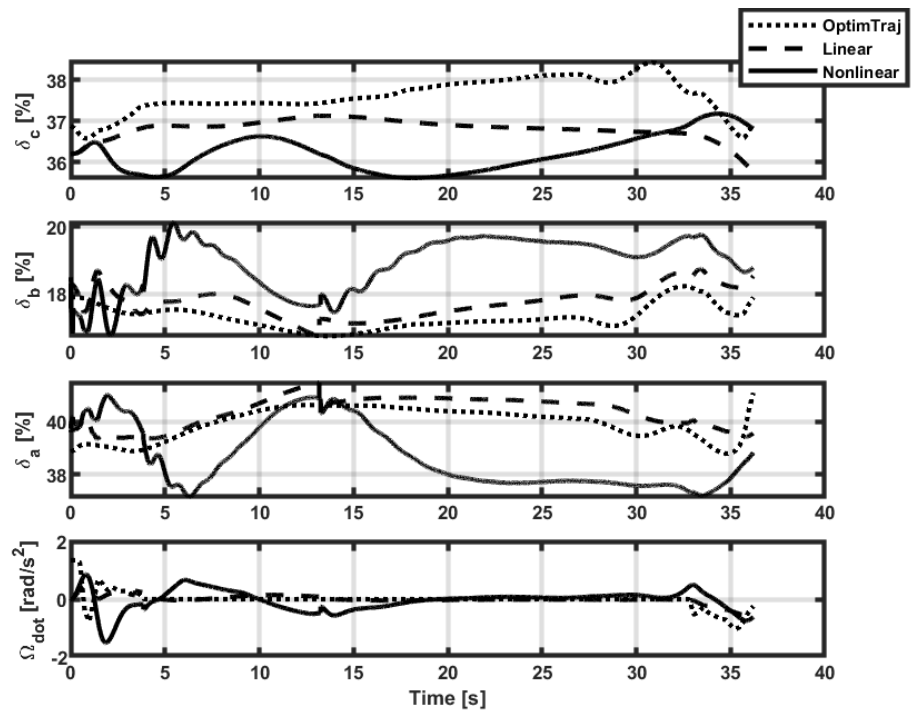


Figure C.20. Helicopter inputs for Case 2

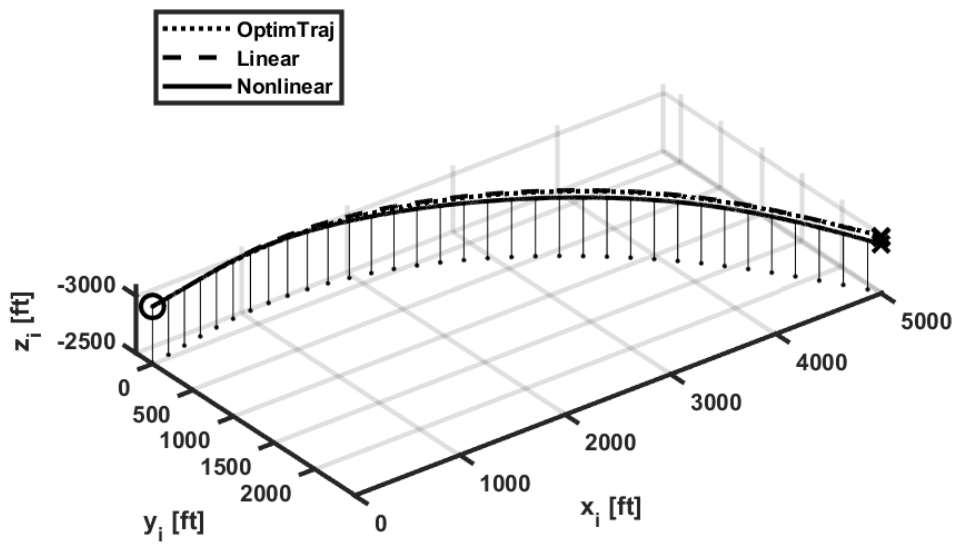


Figure C.21. Trajectory of the helicopter for Case 4

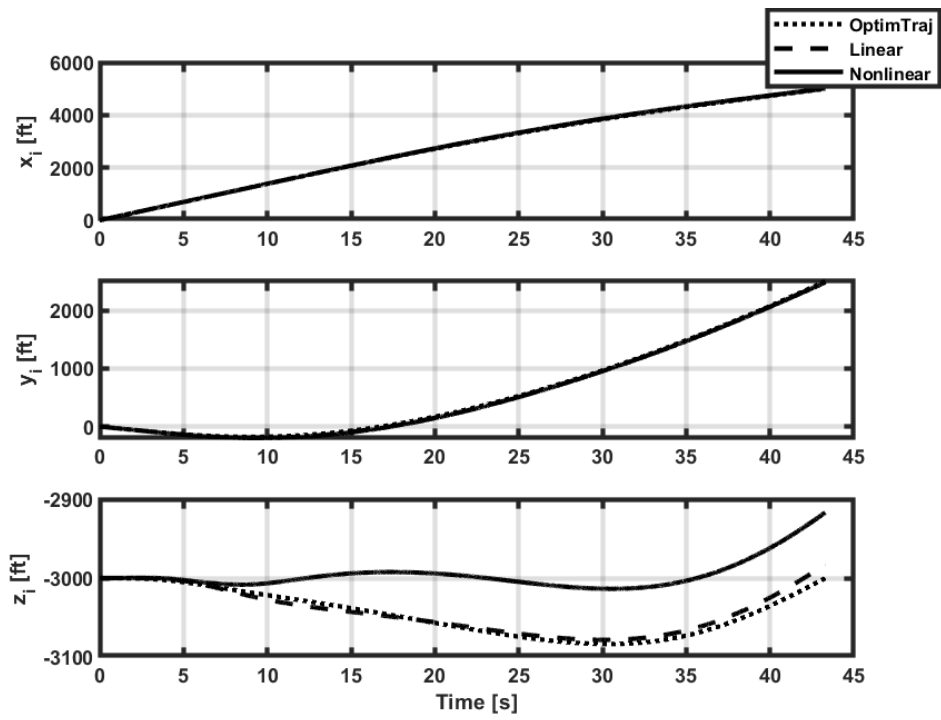


Figure C.22. Inertial positions of the helicopter for Case 4

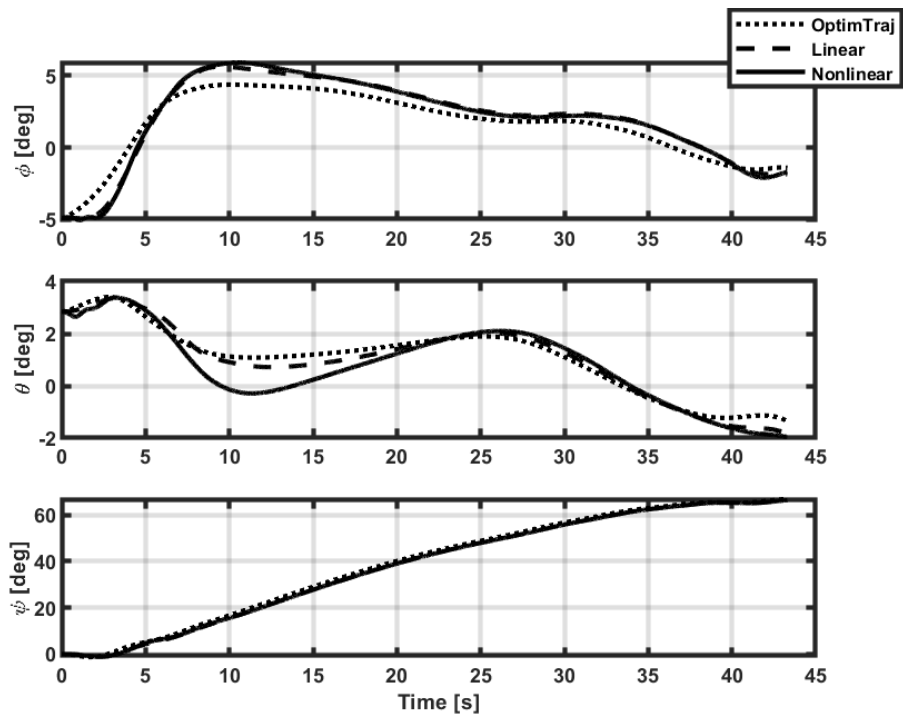


Figure C.23. Euler angles of the helicopter for Case 4



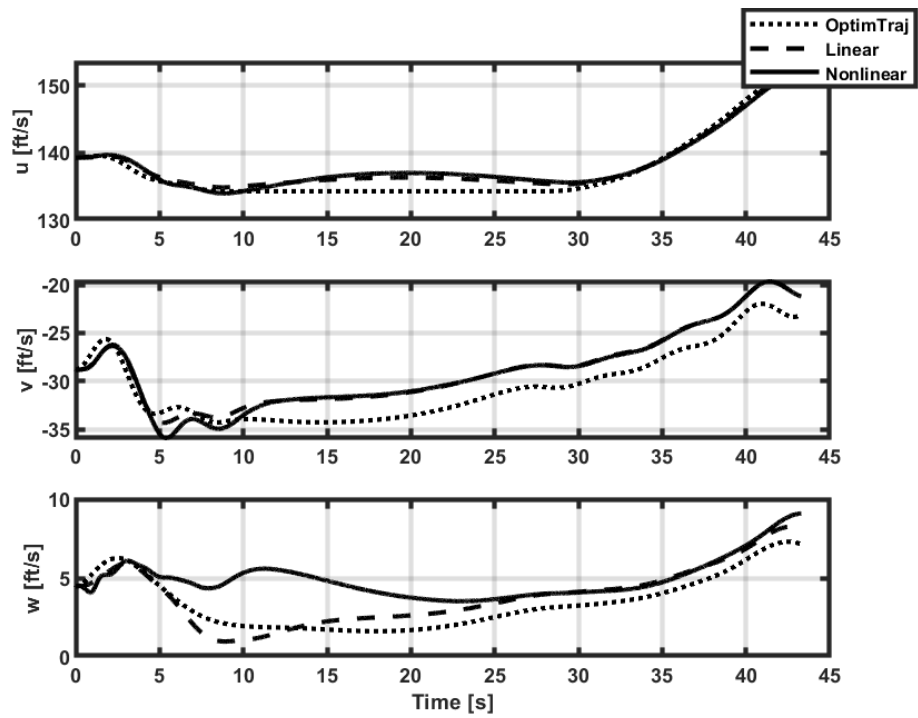


Figure C.24. Body velocities of the helicopter for Case 4

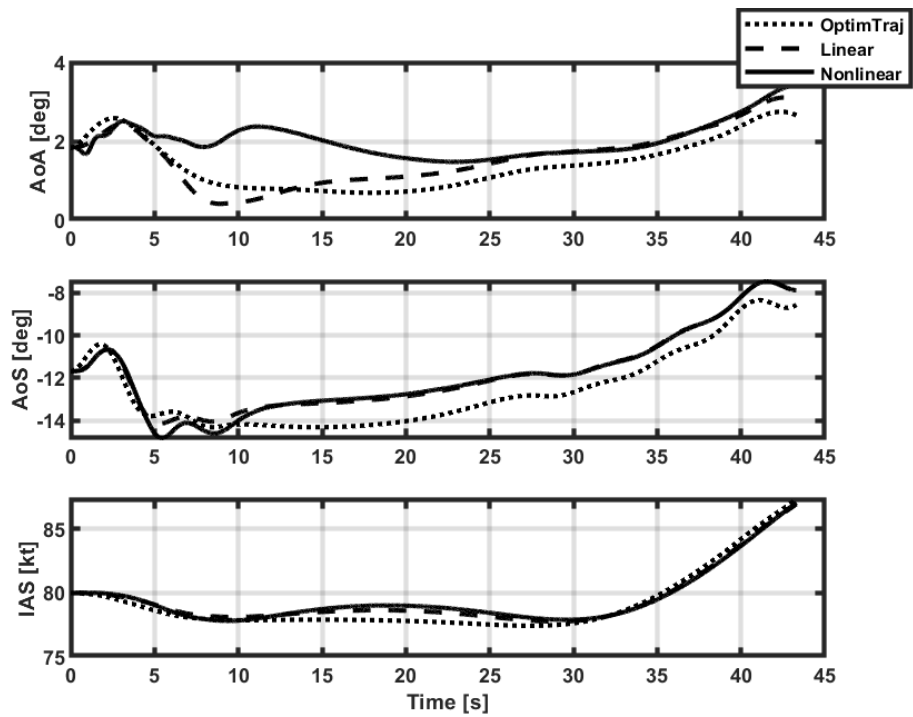


Figure C.25. Air data parameters of the helicopter for Case 4

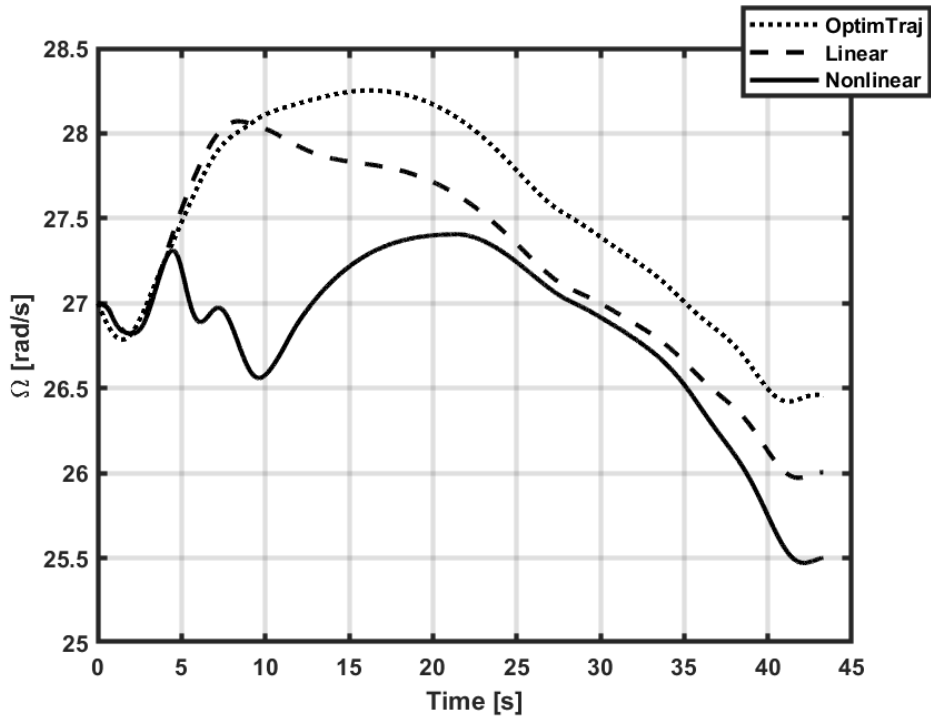


Figure C.26. Main rotor speed of the helicopter for Case 4

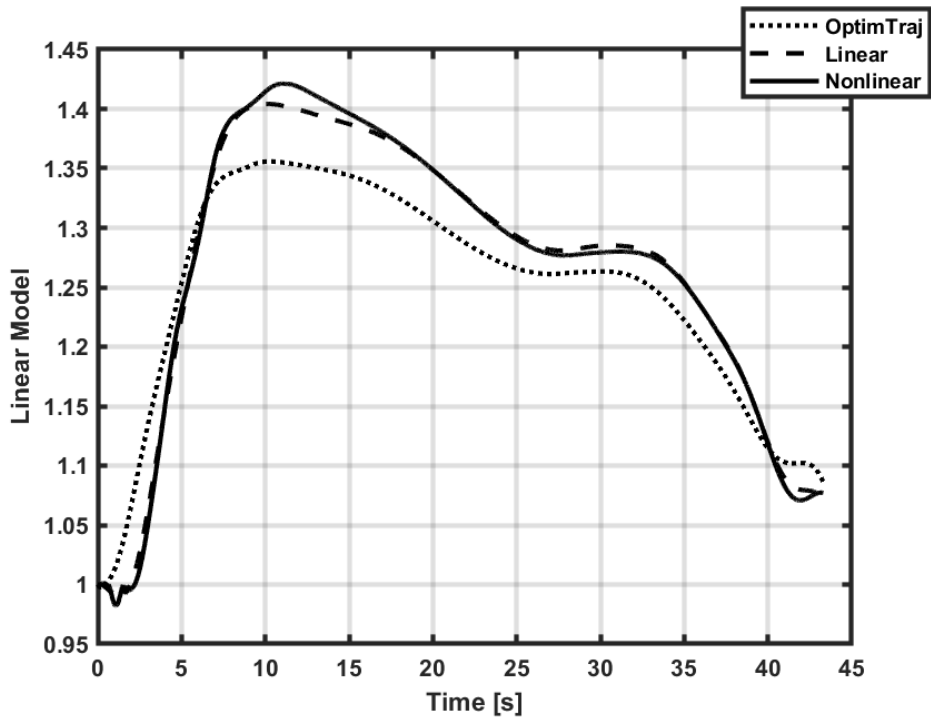


Figure C.27. Linear model transition for Case 4 (0: 80TL, 1: 80L, 2: 80TR)

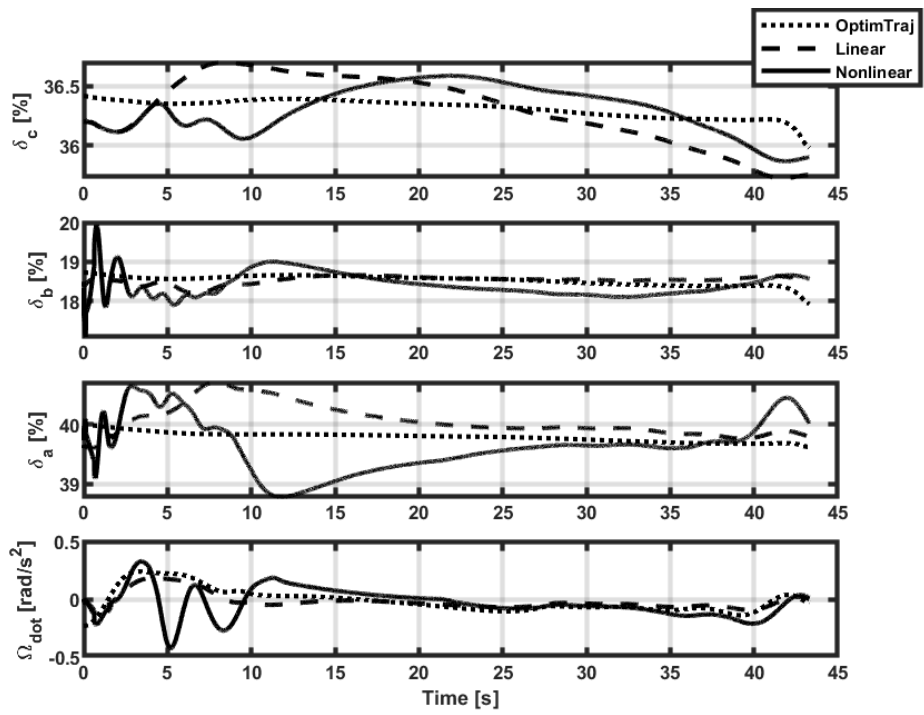


Figure C.28. Helicopter inputs for Case 4

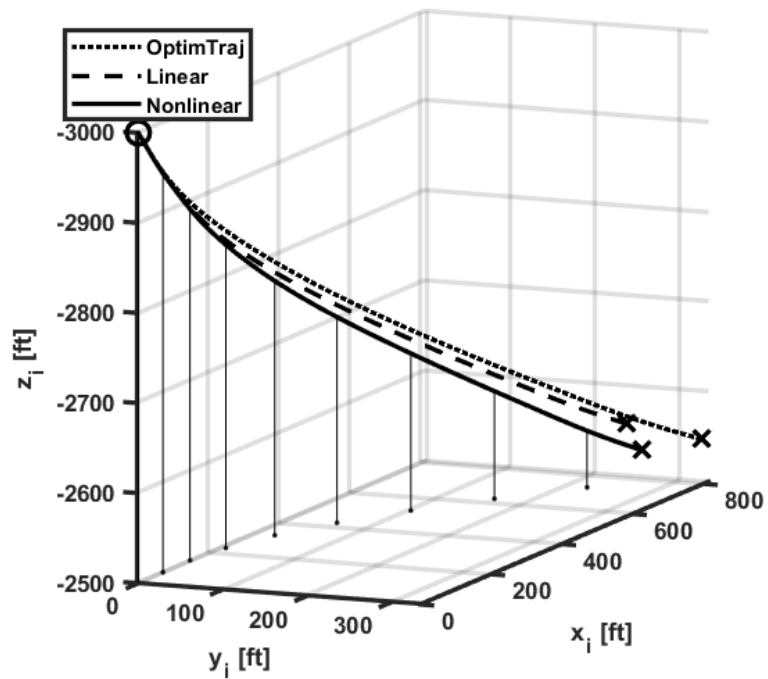


Figure C.29. Trajectory of the helicopter for Case 6

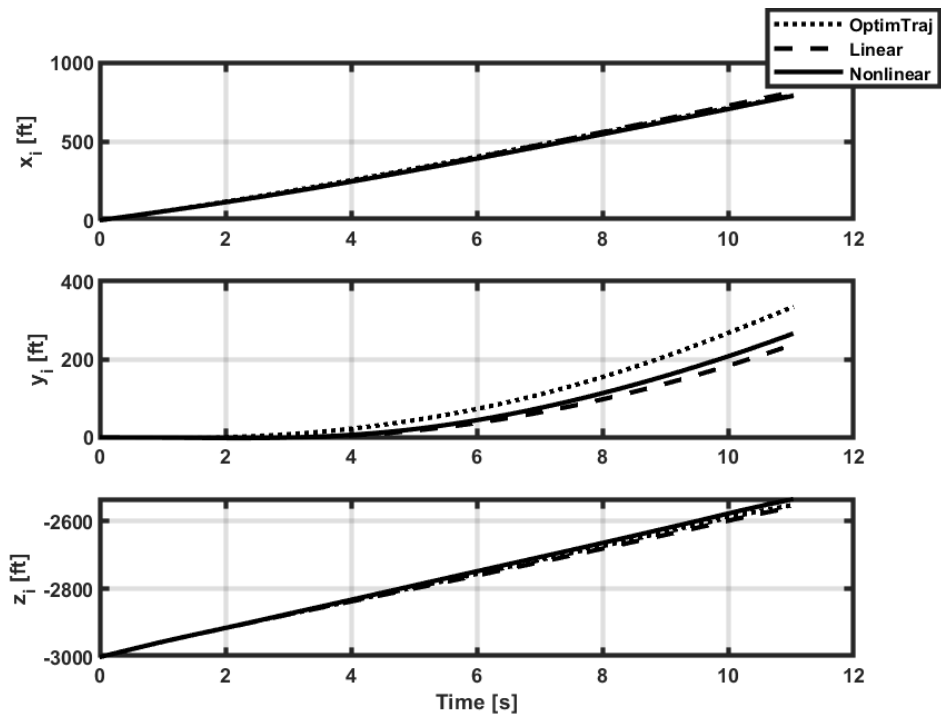


Figure C.30. Inertial positions of the helicopter for Case 6

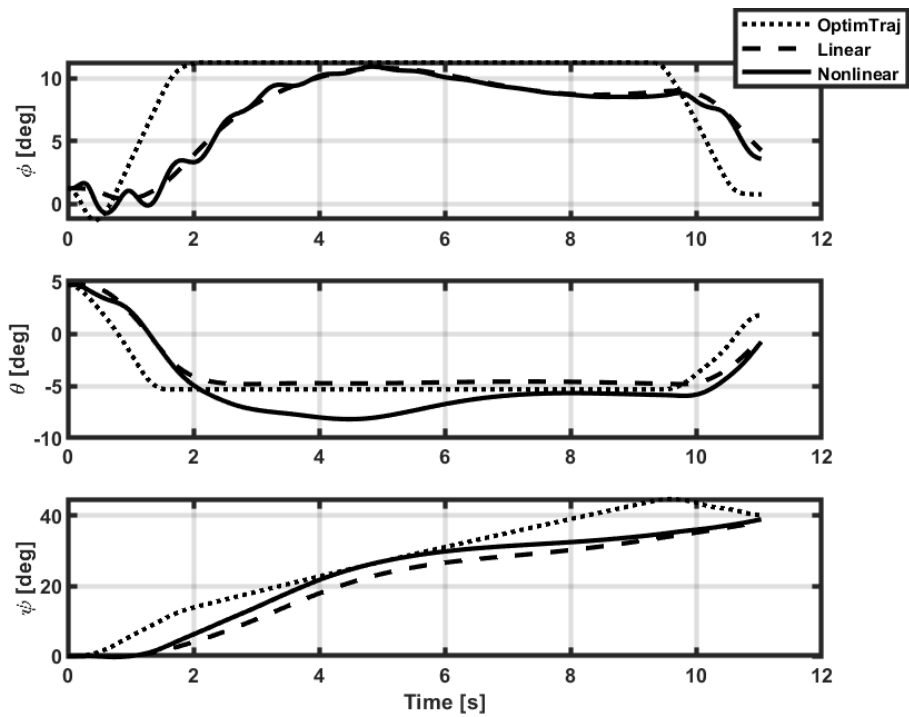


Figure C.31. Euler angles of the helicopter for Case 6

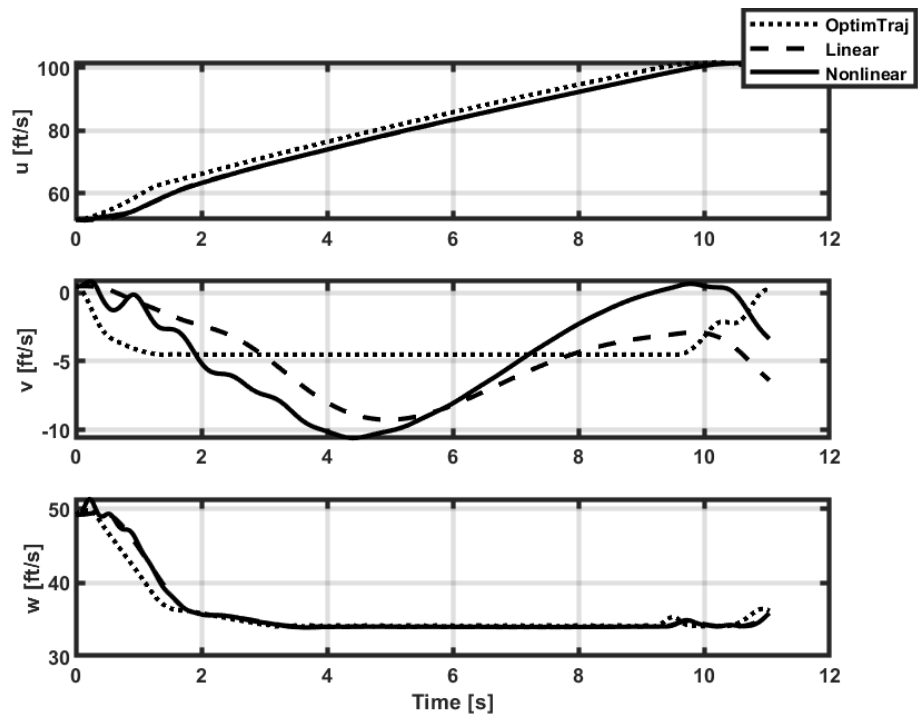


Figure C.32. Body velocities of the helicopter for Case 6

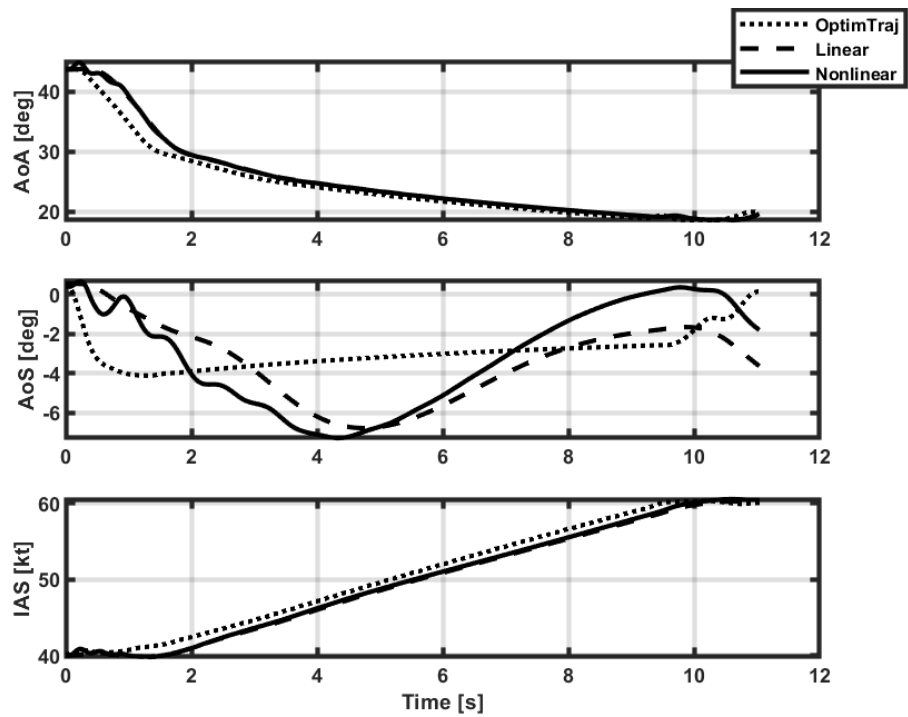


Figure C.33. Air data parameters of the helicopter for Case 6

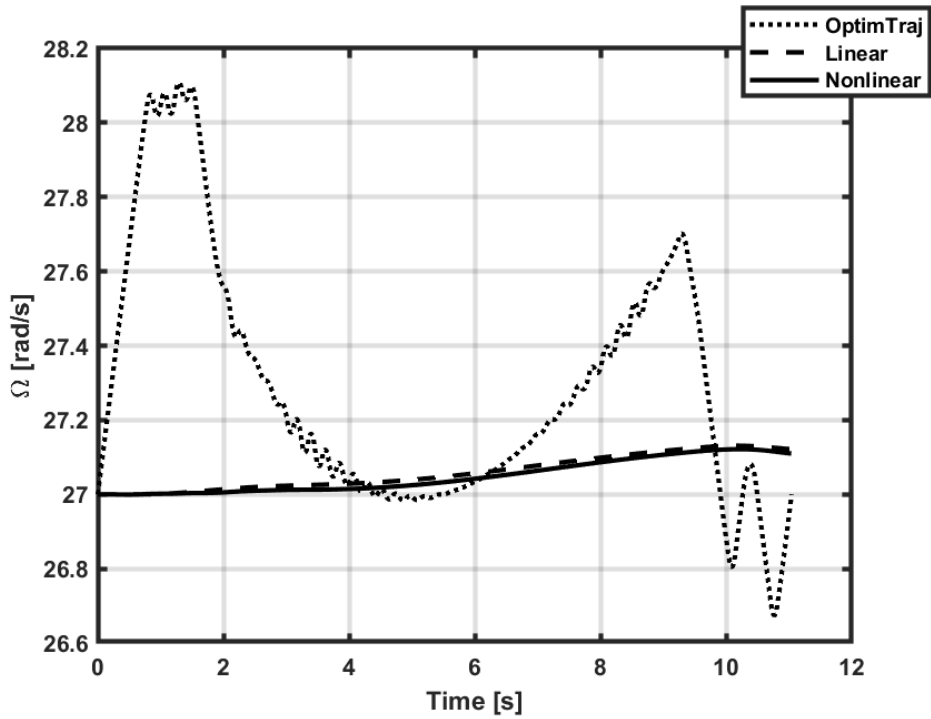


Figure C.34. Main rotor speed of the helicopter for Case 6

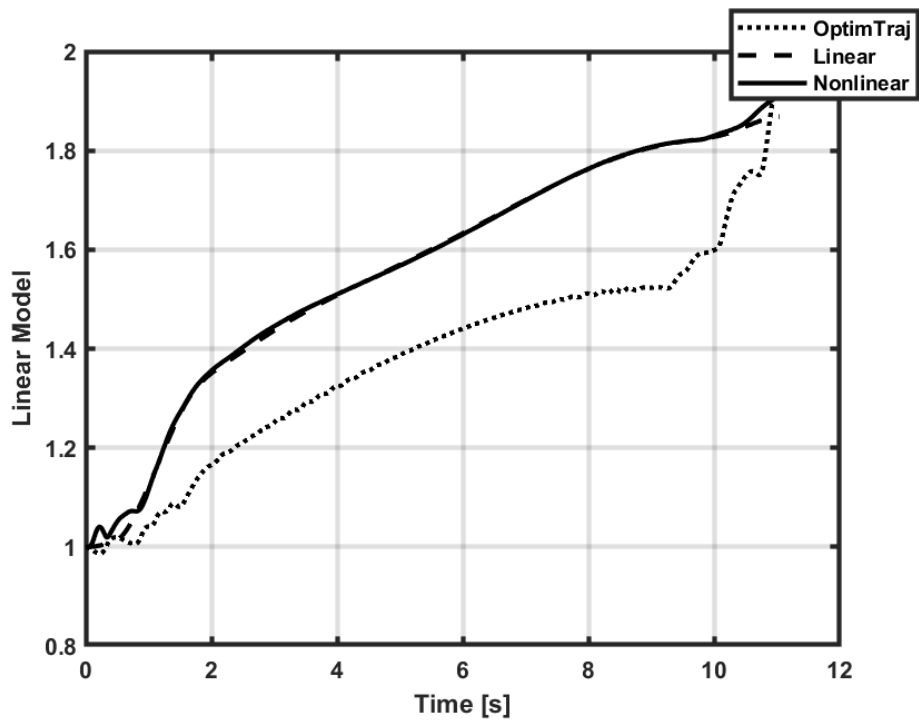


Figure C.35. Linear model transition for Case 6 (1: 40A, 2: 60A)

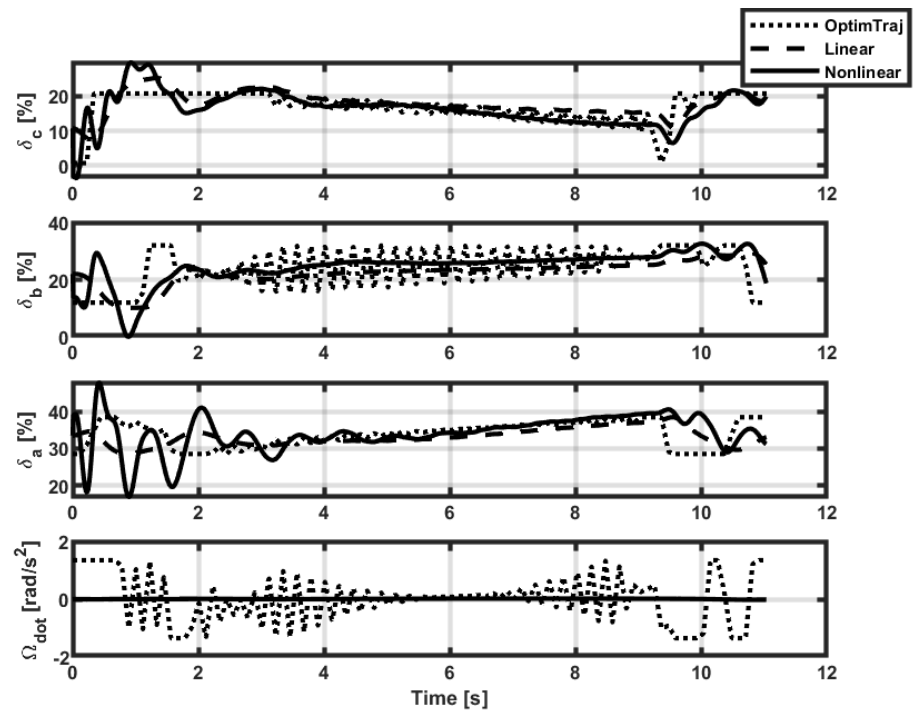


Figure C.36. Helicopter inputs for Case 6