

**REPUBLIC OF TURKEY  
YILDIZ TECHNICAL UNIVERSITY  
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES**

**DEVELOPMENT OF MOTION CUEING ALGORITHM FOR  
SIMULATORS**

**ŞEMSETTİN NUMAN SÖZEN**

**ADVISER**

**ASSOC. PROF. DR. VASFİ EMRE ÖMÜRLÜ**

**İSTANBUL, 2016**

**REPUBLIC OF TURKEY  
YILDIZ TECHNICAL UNIVERSITY  
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES**

**DEVELOPMENT OF MOTION CUEING ALGORITHM FOR  
SIMULATORS**

**ŞEMSETTİN NUMAN SÖZEN**

**M.Sc. THESIS  
DEPARTMENT OF MECHATRONICS ENGINEERING  
MECHATRONICS PROGRAM**

**ADVISER  
ASSOC. PROF. DR. VASFİ EMRE ÖMÜRLÜ**

**İSTANBUL, 2016**

**REPUBLIC OF TURKEY**  
**YILDIZ TECHNICAL UNIVERSITY**  
**GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES**

**DEVELOPMENT OF MOTION CUEING ALGORITHM FOR  
SIMULATORS**

A thesis submitted by Şemsettin Numan SÖZEN in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE** is approved by the committee on 14.07.2016 in Department of Mechatronics, Mechatronics Engineering Program.

**Thesis Adviser**

Assoc. Prof. Dr. Vasfi Emre ÖMÜRLÜ  
Yıldız Technical University

**Approved By the Examining Committee**

Assoc. Prof. Dr. Vasfi Emre ÖMÜRLÜ  
Yıldız Technical University

Asst. Prof. Dr. Cüneyt YILMAZ, Member  
Yıldız Technical University

Prof. Dr. Halil Rıdvan ÖZ, Member  
Aydın University



This study was supported by the Science, Industry and Technology Ministry of Turkey  
Grant No: 0563-STZ-2013-2.

## ACKNOWLEDGEMENTS

---

I would like to thank to my dear adviser Associated Professor Vasfi Emre ÖMÜRLÜ for making such important project initiated on behalf of our country, for trusting me in this project and provide everything to us during our studies, to my co-worker Berkay VOLKANER with whom we performed everything together, to the Science, Industry and Technology Ministry of Turkey for supporting this study, to Havelsan Inc. to provide us a place in the department of Simulator, Education and Test Systems Center to perform our studies in the company and support us in everything, to Furkan İnşaat and İdris AYDIN who is the General Manager of the company for providing us all the material, moral and accommodation support, to Mehmet Can BAŞKAN, Mustafa ALTUNAY, Hakan KARACAOVALI, Afşın KÜÇÜK and Murat AYGÜN who are the employers of Havelsan Inc. and we don't hesitate to call in every respects, to Amega Inc. and Ömer Faruk ÖZDEMİR who is the Automation Manager of the company and supported us every time we need as well as constructing the mechanical structure of the simulator system.

This work was dedicated to my family.

June, 2016

Şemsettin Numan SÖZEN

## TABLE OF CONTENTS

---

	Page
LIST OF SYMBOLS .....	vii
LIST OF ABBREVIATIONS.....	viii
LIST OF FIGURES .....	ix
LIST OF TABLES .....	xi
ABSTRACT.....	xii
ÖZET .....	xv
CHAPTER 1	
INTRODUCTION .....	1
1.1 Literature Review .....	1
1.2 Objective of the Thesis .....	5
1.3 Hypothesis .....	5
CHAPTER 2	
SIMULATOR SYSTEM .....	6
2.1 Kinematic Analysis of 6 DOF Parallel Manipulator - Stewart Platform.....	7
2.1.1 Inverse Kinematic Analysis .....	8
2.1.2 Forward Kinematic Analysis .....	12
2.2 Coordinate Transformation.....	13
2.3 Communication .....	17
CHAPTER 3	
MOTION CUEING AND SYSTEM INTEGRATION .....	18
3.1 Vestibular System .....	18
3.2 Classical Motion Cueing Algorithm .....	20
3.2.1 Tilt Coordination.....	22
3.2.2 Calibration Process of Classical Washout Filters .....	23
3.3 System Integration .....	27
3.3.1 Integration of 6 DOF Low Payload Motion Platform .....	27

3.3.1.1	Platform Motion System .....	28
3.3.1.2	Motion Controller Units .....	29
3.3.1.3	Mechanical and Software Switching Elements .....	30
3.3.1.4	Computer Systems .....	31
3.3.1.5	Command Control, Pedal and Screen Systems .....	31
3.3.1.6	Force and Acceleration (IMU) Sensors .....	31
3.3.1.7	IO Data Processing Cards Compartment .....	32
3.3.2	Frequency Responses of the Platform .....	33
CHAPTER 4		
RESULTS AND DISCUSSION .....		35
REFERENCES .....		57
APPENDIX-A		
CALIBRATION PROCESS OF CLASSICAL MOTION CUEING ALGORITHM ...		59
APPENDIX-B		
FORWARD KINEMATIC M. FILE .....		62
APPENDIX-C		
LONGITUDINAL MODE OPTIMAL MOTION CUEING ALGORITHM DESIGN		65
APPENDIX-D		
JOINT AVIATION REQUIREMENTS STANDARDS .....		69
CURRICULUM VITAE.....		67

## LIST OF SYMBOLS

---

$J$	Inertia matrix
$H$	Input – output force converter
$F_{ext}$	External force
$M_{ext}$	External moment
$R$	Rotation matrix
$F$	Actuator force
$\ddot{i}$	Translational acceleration
$\alpha$	Angular acceleration
$\psi$	Roll angle
$\theta$	Pitch angle
$\varphi$	Yaw angle
$C$	Cosine
$S$	Sine
$P$	Top plate positions
$B$	Bottom plate positions
$T$	Transformation matrix
$\tau$	Time constant
$G_{sc}$	Gain of semi-circular canals
$K_{oto}$	Gain of otoliths
$\zeta$	Damping factor of washout filters
$\omega$	Cornering frequency of washout filters
$\varphi_{x,y}$	Tilt angles
$G$	Gravity
$W(s)$	Transfer function matrix



## LIST OF ABBREVIATIONS

---

AC	Alternative Current
CAG	Center of Lower Universal Joints
CoG	Center of Gravity
DOF	Degrees of Freedom
EEG	Electroencephalogram
FFS	Full Flight Simulator System
IAS	Inertial Coordinate Frame of Linear Actuators
IMU	Inertial Measurement Unit
IMU	IMU Coordinate Frame
IO	Input – Output
JAR	Joint Aviation Requirements Standards
LCD	Liquid Crystal Display
TV	Television
Mech	Collection of Mechanical Coordinate Frames
NASA	The National Aeronautics and Space Administration
SAS	Seat Coordinate Frame
UDP	The User Datagram Protocol
VEST	Vestibular System Coordinate Frame
XML	Extensible Markup Language

## LIST OF FIGURES

	Page
Figure 2.1 Representation of simulator system integration .....	7
Figure 2.2 6 DOF Stewart Platform Structure .....	7
Figure 2.3 Orientation of P about x- axis.....	8
Figure 2.4 Orientation of P about y- axis.....	9
Figure 2.5 Orientation of P about z- axis.....	9
Figure 2.6 Forward Kinematic Analysis Flowchart.....	12
Figure 2.7 Representation of coordinate frame of the flight simulator system .....	13
Figure 2.8 Coordinate set scheme of the platform.....	14
Figure 2.9 FlightGear input-output interface .....	19
Figure 2.10 UDP communication model .....	21
Figure 3.1 Vestibular System representation.....	23
Figure 3.2 Structure of classical motion cueing algorithm .....	25
Figure 3.3 Tilt coordination effect.....	26
Figure 3.4 Motion limits for pitch motion .....	27
Figure 3.5 Motion limits for roll motion.....	28
Figure 3.6 Motion limits for yaw motion .....	29
Figure 3.7 Representation of reference axes of the aircraft.....	29
Figure 3.8 Results of workspace analysis of the manipulator .....	29
Figure 3.9 Representation of block diagram of the calibration process .....	30
Figure 3.10 Step response of W11 filter .....	36
Figure 3.11 Step response of W12 filter .....	36
Figure 3.12 Step response of W21 filter .....	37
Figure 3.13 Step response of W22 filter .....	37
Figure 3.14 Bode diagrams for W11 and W11d filters .....	38
Figure 3.15 Bode diagrams for W12 and W12d filters .....	38
Figure 3.16 Bode diagrams for W21 and W21d filters .....	39
Figure 3.17 Bode diagrams for W22 and W22d filters .....	39
Figure 3.18 Continuous-time optimal motion cueing algorithm Simulink model .....	40
Figure 3.19 Discrete-time optimal motion cueing algorithm Simulink model .....	40
Figure 3.20 Optimal washout filter solution procedure .....	41
Figure 3.21 6 DOF low payload flight simulator system .....	42
Figure 3.22 Motor and system .....	42
Figure 3.23 Controller group .....	43
Figure 3.24 Mechanical switching elements.....	44
Figure 3.25 Command controls and pedals .....	45
Figure 3.26 Acceleration (IMU) sensor.....	46

Figure 3.27	Force sensor.....	46
Figure 3.28	IO data processing cards.....	47
Figure 4.1	Platform motion during takeoff.....	49
Figure 4.2	Acceleration through x- axis, filtered x- acceleration and platform x- position .....	50
Figure 4.3	Roll motion about counter clockwise direction.....	50
Figure 4.4	Roll motion about clockwise direction.....	50
Figure 4.5	Roll velocity and roll angle of the aircraft .....	50
Figure 4.6	Performance of the platform roll motion.....	51
Figure 4.7	Pitch motion of the aircraft through negative pitch direction .....	51
Figure 4.8	Pitch motion of the aircraft through positive pitch direction .....	51
Figure 4.9	Pitch velocity, filtered pitch velocity and platform pitch angle .....	51
Figure 4.10	Helicopter motion through x- axis and results .....	52
Figure 4.11	Helicopter motion through y- axis and results .....	52
Figure 4.12	Helicopter motion through z- axis and results.....	53
Figure 4.13	Results for roll motion of the helicopter .....	54
Figure 4.14	Results for pitch motion of the helicopter .....	55
Figure 4.15	Results for yaw motion of the helicopter .....	56
Figure A.1	Simulink block in which received datas from the aircraft are scaled.....	61
Figure A.2	Representation of Simulink block contains washout filters .....	61
Figure A.3	Representation of "conversion" block including saturation values in Simulink .....	62
Figure A.4	Interior structure of "conversion" block (saturation - red blocks).....	62
Figure D.1	Typical test profile for 2 hours .....	70

## LIST OF TABLES

---

	Page
Table 3.1 Human motion system parameters.....	23
Table 3.2 Threshold values of human motion perception.....	24
Table 3.3 Translational motion limits of the platform.....	28
Table 3.4 Defined IP addresses.....	43
Table 3.5 Frequency responses of the platform.....	47
Table 3.6 Frequency responses in decibel.....	48
Table A.1 Calibration parameters of washout filters.. . . . .	61

## ABSTRACT

---

# DEVELOPMENT OF MOTION CUEING ALGORITHM FOR SIMULATORS

Şemsettin Numan SÖZEN

Department of Mechatronics Engineering

MSc. Thesis

Adviser: Assoc. Prof. Dr. Vasfi Emre ÖMÜRLÜ

Flight simulators are widely used for commercial and military purposes for training of pilots. Additionally, they are increasingly employed for ground, marine, and space vehicles. Simulator systems help us to understand driver/pilot behavior in simulations as close to real flight. Real aircraft flights are very expensive and have numerous risks. Flight simulators are safe and practical structures. During The Second World War, flight simulators were required in military. Technological developments in analog computers contributed to flight simulator developments. Following years, some specific options like aerodynamics and flight behavior were incorporated in simulators. There is no specific standard for motion cueing, but in literature, four major algorithms are currently present; classical, optimal, adaptive and model predictive base algorithms.

Motion cueing algorithm is a must that provides the motion of a simulator within boundaries of limited robotic workspace. 6 DOF parallel manipulators are mostly used for flight simulation. Inputs to the manipulator are simulated aircraft angular velocities and translational accelerations as references to generate necessary motions. The mobile platform of the mechanism should return its neutral position in order to start the next motion sequence. This process is called as “Washout”. Washout motion must be under human motion sensation thresholds which is covered in literature by several researchers. For the sake of sensing continuous translational accelerations, the platform tilts itself and uses gravity vector to create artificial motion sensation translationally which is called “Tilt Coordination”. Force effect caused by continuous acceleration of aircraft is also provided by Tilt Coordination. Tilt coordination is provided by applying 2<sup>nd</sup> order low-pass filters for x- and y- axes in the algorithm.

Basically a simulator consists of a robotic manipulator and a simulator room on the top and kinematic analysis of the manipulator is crucial to obtain meaningful motion results.

Translational and rotational positions obtained from the output of motion cueing algorithm are transmitted to the manipulator as actuator extensions calculated by inverse kinematics. Coordinate transformations in a robotic system are another important topic because motion sensation must be at the eye point of the pilot/driver. The major issue in simulators is motion cueing algorithm that relates the simulator input to the aircraft input. This is possible by the communication between motion control computer and simulation computer. Communication protocol in presented system in this study receives the translational accelerations and angular velocities from the simulation package program called FlightGear via UDP protocol. For this purpose, an XML script file is generated to receive related motion control inputs and take them into the motion cueing algorithm to be processed. If aircraft dynamic model is used in simulation control computer via Matlab/Simulink, it is needed to receive related motion parameters as packages by using package input/output block.

Classical motion cueing algorithm constitutes the basis of motion cueing algorithms and includes 1<sup>st</sup> order and 2<sup>nd</sup> order high-pass washout filters that are used to bring the manipulator to its neutral position in its workspace under the threshold values of human sensation for all motions. Thus, vestibular system model is necessary to check the motion whether it is under the threshold value or not. Vestibular system is in inner ear of a human and the system consists of semi-circular canals sensing rotational motions and otoliths sensing translational motions. Calibration process of classical washout filters is also important in motion cueing algorithm design. There are position saturation values through x-, y- and z- axes and angle saturation values about x-, y- and z- axes of the manipulator according to its production catalog. If filter gains are higher than the saturation values that manipulator is capable of, there is possibility of the occurrence of mechanical failures. Tilt coordination is another important topic because force effect caused by continuous acceleration of aircraft is provided by tilt coordination. Tilt coordination is provided by applying 2<sup>nd</sup> order low-pass filters for x- and y- axes in the algorithm. In literature there are not many works explaining the calibration process of washout filters.

Previously, we implemented classical washout filter on a small scale desktop platform. After a significant optimization was applied, a full scale flight simulator mechanical structure has been constructed and classical washout filter was implemented. The system is on a 6x6 parallel manipulator and is driven by AC motors. The classical washout filters were implemented on Matlab – Simulink environment with Real-Time Windows Target. This algorithm has first and second order high pass filters for high frequency components and first order low pass filters for low frequency components of aircraft flight variables. Also, all software implementations are established on Matlab. Flight data are obtained from virtual flight software with 5 Hz sampling rate.

In this study, system integration of a low payload flight simulator is explained, results of classical motion cueing algorithm, vestibular system validations and design and calibration process of classical washout filter are presented and comparison between results obtained from Boeing 777-300ER and Sikorsky flights are shown. Many tests according to system requirement documents are performed and results of these tests are also presented. Furthermore, accuracy of motions obtained from the output of the manipulator is proven by using forward kinematics. As well as classical motion cueing algorithm, optimal washout filter design and its implementation are explained in this study. At the end of the paper conclusions are presented.

**Key words:** Flight Simulator, Motion Cueing, Washout Filter Calibration, Vestibular System, System Integration.

# SİMÜLATÖRLER İÇİN HAREKET ALGI ALGORİTMASI GELİŞTİRİLMESİ

Şemsettin Numan SÖZEN

Mekatronik Anabilim Dalı

Yüksek Lisans Tezi

Tez Danışmanı: Doç. Dr. Vasfi Emre ÖMÜRLÜ

Uçuş benzeticileri, pilotların eğitilmesi amacıyla ticari ve askeri alanlarda geniş bir kullanım alanına sahiptirler ve kara, deniz ve uzay araçları için de kullanım oranları giderek artmaktadır. Benzetici sistemleri, benzetim ortamı içerisinde sürücü/pilot davranışını olabildiğince gerçek uçuşa yakın olarak anlamamıza yardımcı olmaktadır. Gerçek uçuşların çok pahalı ve çeşitli risklere sahip olması nedeniyle benzeticiler daha güvenli ve pratik yapılardır. İkinci Dünya Savaşı sırasında uçuş benzeticilerine gereksinim duyulmuştur. Analog bilgisayarlardaki teknolojik gelişmeler ise uçuş benzeticilerinin geliştirilmesine katkı sağlamıştır. Takribi yıllarda, aerodinamik gibi bazı özel konular da benzetici sistemler içerisine birleştirilmiştir. Hareket algı yönetimi için belirli bir standart yoktur fakat literatürde klasik, optimal, adaptif ve model öngörülü kontrol algoritması olmak üzere dört ana algoritma yapısı mevcuttur.

Hareket algı yönetimi algoritması, bir simülatör hareketini sınırlı bir çalışma uzayı içerisinde sağlamak için bir zorunluluktur. Uçuş benzetimi için çoğunlukla 6 DOF paralel manipülatörler kullanılmaktadır. Manipülatör girişleri, gerekli hareketleri üretmek için açısal hız ve çizgisel ivme referansları olarak benzetilir. Mekanizma hareketli platformu, bir sonraki hareket dizisine başlamak için nötr pozisyona dönmelidir. Bu işlem "sıfırlama" olarak adlandırılmaktadır ve sıfırlama hareketi, literatürdeki farklı çalışmalarla elde edilmiş olan insan hareket algı eşik değerlerinin altında bir değerde olmalıdır. Sürekli çizgisel ivmelenmelerin algılanması için platform kendisini eğer ve "Eğim Koordinasyonu" olarak adlandırılan çizgisel olarak yapay bir hareket algısı üretimi için yer çekimi vektörünü kullanır. Uçuş aracının sürekli ivmelenmesinden kaynaklanan kuvvet etkisi de eğim koordinasyonu tarafından sağlanmaktadır. Eğim koordinasyonu,



algoritma içerisinde x- ve y- eksenlerinde 2 .dereceden düşük geçiren filtrelerin kullanılması aracılığıyla sağlanır.

Bir benzetici temel olarak bir robotik manipülatör ve manipülatör üzerine konumlandırılmış bir benzetici odasından oluşmaktadır bu nedenle manipülatörün kinematik analizleri anlamlı hareket sonuçları elde edebilmek için büyük önem taşımaktadır. Hareket algı yönetimi algoritmasının çıkışlarından elde edilen çizgisel ve dönel pozisyon değerleri manipülatöre ters kinematik tarafından hesaplanan bacak boyları olarak iletilmektedir. Robotik sistemler içerisindeki koordinat dönüşümü ise diğer bir önemli konudur çünkü benzetici içerisindeki hareket sürücü/pilotun göz hizasında olmalıdır.

Uçuş benzeticilerindeki esas konu benzetici hareketini uçuş aracı hareketine ilişkilendiren hareket algı yönetimi algoritmasıdır ve bu durum hareket kontrol bilgisayarı ile benetim kontrol bilgisayarı arasındaki haberleşme ile mümkündür. Bu çalışmada sunulan sistem içerisindeki haberleşme protokolü, FlightGear olarak adlandırılan uçuş benzetimi paket programından çizgisel ivme ve açısal hız değerlerini UDP protokolü aracılığıyla almaktadır. Bu amaçla, ilgili hareket kontrol girdilerini alıp hareket kontrol yönetimi algoritmasında işlenmek üzere ileten bir XML senaryo dosyası üretilir. Eğer benzetim kontrol bilgisayarı Matlab/Simulink aracılığıyla, geliştirilmiş bir uçuş aracı dinamik modeli kullanılıyor ise ilgili hareket kontrol parametreleri, paket giriş/çıkış bloğu aracılığıyla paket veriler olarak alınır.

Klasik hareket algı yönetimi algoritması hareket algı yönetimi algoritmalarının temelini oluşturur ve manipülatörü nötr pozisyonuna, her harekette, çalışma uzayı içerisinde ve insan algısı eşik değerlerinin altında götürmek için 1. ve 2. dereceden yüksek geçiren sıfırlama filtrelerini içerir. Bu nedenle, hareketin eşik değerlerinin altında olup olmadıklarını kontrol etmek için vestibüler sistem modeli gerekmektedir. Vestibüler sistem insanda iç kulağın içerisinde yer almaktadır ve dönme hareketlerinin algılanmasını sağlayan yarı dairesel kanallar ve çizgisel hareketlerin algılanmasını sağlayan kulak taşlarından oluşmaktadır.

Klasik sıfırlama filtrelerinin kalibrasyon işlemi hareket algı yönetimi algoritması tasarımında önemli olan diğer bir konudur. Sistemde, manipülatör üretim kataloğuna göre x-, y- ve z- eksenleri üzerinde pozisyon doyum değerleri ve yine bu eksenler etrafında açı doyum değerleri yer almaktadır. Eğer sıfırlama filtresi kazançları platformun yapabileceği doyum değerlerinden yüksek değerlerde ise manipülatörde mekaniksel arızaların meydana gelme olasılığı yüksektir. Literatürde, sıfırlama filtrelerinin kalibrasyon aşamalarını açıklayan çok fazla çalışma mevcut değildir.

Klasik hareket algı yönetimi algoritması öncelikle küçük ölçekli bir masaüstü platform üzerinde denenmiş ve önemli bir optimizasyon aşaması uygulandıktan sonra tam ölçekli bir uçuş benzetici mekanik yapısı inşa edilmiş ve klasik sıfırlama filtresi uygulanmıştır. Sistem 6x6' lık bir paralel manipülatördür ve AC motorlarla sürülmektedir. Klasik sıfırlama filtreleri Matlab/Simulink ortamında, Windows gerçek zamanlı hedef ile uygulanmıştır. Algoritma, uçuş aracı uçuş değişkenlerinin yüksek frekanslı bileşenleri için 1. ve 2. dereceden yüksek geçiren filtrelere ve düşük frekanslı bileşenleri için ise 1. dereceden düşük geçiren filtrelere sahiptir. Aynı zamanda tüm yazılım uygulamaları da Matlab üzerinde kurulmuştur. Sanal uçuş yazılımından uçuş verileri 5 Hz' lik örnekleme zamanı ile elde edilmektedir.

Bu tez çalışmasında, düşük yük kapasiteli bir uçuş simülatörünün sistem bütünleştirilmesi açıklanmış, klasik hareket algı yönetimi algoritması sonuçları, vestibüler sistem doğrulamaları ve klasik sıfırlama filtrelerinin kalibrasyon işlemi sunulmuş, Boeing 777-

300ER ve Sikorsky tipi helikopter uçuşlarından elde edilen sonuçların karşılaştırması gösterilmiştir. Sistem gereksinim dokümanları doğrultusunda ilgili testler gerçekleştirilmiş ve bu testlerin sonuçları da sunulmuştur. İlâveten, manipûlatör çıkışlarından elde edilen hareketlerin doğrulukları, düz kinematik kullanılarak ispatlanmıştır. Klasik hareket algı yönetimi algoritmasının yanısıra optimal sıfırlama filtresi tasarımı ve uygulaması da bu çalışma içerisinde açıklanmıştır. Tezin sonunda ise bu çalışmadan elde edilen çıkarımlar anlatılmıştır.

**Anahtar Kelimeler:** Uçuş Simûlatörü, Hareket Algı Yönetimi, Sıfırlama Filtresi Kalibrasyonu, Vestibüler Sistem, Sistem Bütünleştirmesi

### INTRODUCTION

#### 1.1 Literature Review

When compared to real flights, flight simulator systems are more useful, safer and cheaper and these systems allow to understand the pilot behavior as close to real flights. Nowadays, universities and industrial company laboratories are performing studies to develop new type vehicle and aircraft dynamic models for new generation flight simulators.

Simulator systems are reliable, realizable, can be implemented and by using motion information feedback they create a realistic flight feeling [1]. For this aim, employing a robust motion cueing algorithm is crucial [2].

Due to being a “man in the loop” process and having a complex structure, simulators include a complicated code algorithm so simulator design is a difficult task [3].

There are differences between real and virtual data so the examination which factors cause this issue is required. These factors are as follows:

- a. Human motion perception system (vestibular system),
- b. Visual simulation environment
- c. Audio feedback.

To understand the problems that are possible to occur in a motion perception application, following questions are asked:

- a. What does human motion perception depend on in real time applications?
- b. How is it possible to improve motion perception through changing parameters of factors described above? [4]

Human motion perception system is called “Vestibular System” and is placed in the inner ear. The system includes semi-circular canals responsible for sensing rotational motions and includes otoliths responsible for sensing translational motions. So the mathematical model of vestibular system is required and developed model was firstly used in [5] and transfer functions of vestibular system were presented in [6] study. Related signals are transmitted to the neural constructions through nerve fibers by the receptor cells of the semi-circular canals and otoliths. Otolith organ consists of two structures: Ultricle that senses horizontal motions and saccule that senses vertical motions. Vestibular system also plays a major role in balance of the human body [7], [8].

For the sake of sensing continuous translational accelerations, the platform tilts itself and uses gravity vector to create artificial motion sensation translationally which is called “Tilt Coordination”. Force effect caused by continuous acceleration of aircraft is also provided by Tilt Coordination. Tilt coordination is provided by applying 2<sup>nd</sup> order low-pass filters for x- and y- axes in the algorithm [9].

Visual scaling is another topic for image perception feeling because it creates differences for feeling. So visual scaling at optimal level affects the perception positively and visual feedback changes was examined under several visual scale factors in. In addition, depth and slope play an important role for motion perception. Studies in literature showed that to create a realistic motion perception, horizontal image needs to have a slope of 120 ° [10].

Audio feedback has less effect on motion perception because it is based on simulation acoustics and number of sound supplies. To get an accurate result, it is useful to use sound emitting object in the virtual environment [4].

Without inertial cues pilot has no information about aircraft accelerations and rotations so inertial cues are very important for simulators. Unlike real aircrafts, simulators have limited workspace for accelerations and rotations. Motion cueing algorithms are used for the purpose of providing workspace to the manipulator and they include washout filters that bring the manipulator to its neutral position [11].

Classical motion cueing algorithm was found in NASA Ames Research Center in 1969 and after that rotating coordinate filter was created in 1973. Classical motion cueing algorithm constitutes the basis of motion cueing algorithms and includes 1<sup>st</sup> order and 2<sup>nd</sup> order high-pass washout filters that are used to bring the manipulator to its neutral

position in its workspace under the threshold values of human sensation for all motions. Thus, vestibular system model transfer functions are necessary to check the motion whether it is under the threshold value or not. Calibration process of classical washout filters is also important in motion cueing algorithm design. There are position saturation values through x-, y- and z- axes and angle saturation values about x-, y- and z- axes of the manipulator according to its production catalog. If filter gains are higher than the saturation values that manipulator is capable of, there is possibility of the occurrence of mechanical failures [12].

Because the classical algorithm has a limited workspace, adaptive algorithm was developed by Parrish, Dieudonne, Bowles and Martin in 1975 and to eliminate the workspace problem adaptive algorithm was applied to the system in frequency domain. In the algorithm coefficients were different [13] and motion perception was considered as a tracking problem [14]. Manipulator should track accelerations obtained from the simulation as soon as possible [15].

“Adaptive Classical Motion Cueing Algorithm” was developed by applying adaptive washout filters in classical algorithm and adaptive washout filter gains are obtained by minimizing a cost function. Defusing incorrect cues at the actuator extensions and providing a smooth flight feeling are the advantages of the adaptive algorithm but because the structure of algorithm may vary, there are sometimes differences in motion perception. In case of exposure to any distorting effect, dynamic stability of the system is also provided by adaptive algorithm [16]. In [17], the adaptive algorithm structure was examined and implemented to the system to keep it in limits of its workspace. After developing classical washout filter, trial-and-error method is sufficient to determine necessary parameters such corner frequency, damping ratio, gain, cost and adaptive step size for adaptive algorithm [18].

Classical motion cueing algorithm was applied 3, 6 and 8 DOF manipulators and there are several studies comparing usage of the algorithm in literature. According to several studies, 3 DOF platforms are more effective for the translational feedback. On 6 DOF platforms because of more motion freedom motion is felt more intensively. Similar performance of 6 DOF platforms were observed on 8 DOF platforms [19].

In simulator systems there is possibility of occurrence of phase delay between simulation motion and the manipulator motion. In order to handle this, a flexible and effective

software should be integrated into the algorithm. Compensation methods (i.e. fuzzy logic, adaptive model, nonlinear solving) are also available to handle this [20].

Model predictive algorithm is a novel algorithm and it is presented in [21]. This is an advanced control technique and provided control actions are optimized according to a reference system model. Model predictive algorithm computes the control input by solving a finite horizon open-loop optimal control problem. Consequently, the optimization process gives an optimal control sequence by using current state of the system for initial state of the platform. There is an internal model in model predictive control and this may be thought as a limitation but because allowing the controller to attempt system dynamics it can also be considered as an advantage [22].

Generally motion cueing algorithms give good results but in some cases actuators may reach maximum actuator lengths and this may create disruption for the system [23].

Optimal control theory including vestibular system model was implemented by Sivan, Ishsalom and Huang in 1982 to reduce the cost function and it is based on a problem how to control the system as closed to desired value with respect to specified dynamics and constraints basically [24]. In this algorithm by using Newton Raphson method, Riccati equation is solved in real time. The major issue is to assign  $W(s)$  transfer function that relates the simulator control input to the aircraft control input so that the cost function is to be minimized. According to related studies it can be seen that the optimal motion cueing algorithm is more effective than the classical motion cueing algorithm and the adaptive motion cueing algorithm [25]. In [26], to decrease the pilot sensation error, “human vestibular based senseless maneuver optimal washout filter” was proposed and human motion perception threshold values obtained from several studies were presented. System was considered as an inverted pendulum. Optimal control strategy is separated into two categories: Conventional control and advanced control [27].

To make a comparison of motions Pouliot suggested two approaches: Angular velocity and forces that produced comparison of simulator and air vehicle angular velocity comparison with jerk [28].

This thesis includes an abstract, an introduction stage covers related literature review, kinematic analysis of 6 DOF parallel manipulator constitutes the mobile platform of the simulator system, classical motion cueing algorithm, its implementation to the desktop parallel manipulator and full scale manipulator and the calibration of classical washout

filters, system integration and optimal motion cueing algorithm design. At the end of the paper comparison of results obtained from different flights, conclusion and future works are presented with references used.

## **1.2 Objective of the Thesis**

This thesis aims to compare motion cueing algorithms in terms of efficiency and usability on 6 DOF FFS with respect to system requirement documents. There is no specific standard for motion cueing, but in literature, four major algorithms are currently present; classical, optimal, adaptive and model predictive base algorithms.

## **1.3 Hypothesis**

Classical motion cueing algorithm constitutes the basis of motion cueing algorithms and includes 1<sup>st</sup> order and 2<sup>nd</sup> order high-pass washout filters that are used to bring the manipulator to its neutral position in its workspace under the threshold values of human sensation for all motions. Calibration process of classical washout filters is also important in motion cueing algorithm design. There are position saturation values through x-, y- and z- axes and angle saturation values about x-, y- and z- axes of the manipulator according to its production catalog. If filter gains are higher than the saturation values that manipulator is capable of, there is possibility of the occurrence of mechanical failures. So, coordinate transformation, accurate dynamic modelling and kinematic analysis of the manipulator are crucial for flight simulator design to develop an accurate motion cueing algorithm. In literature there are not many works explaining the calibration process of washout filters.

## CHAPTER 2

---

### SIMULATOR SYSTEM

A simulator system consists of a robotic manipulator and a simulator room on the top and kinematic analysis of the manipulator is crucial to obtain meaningful motion results. Translational and rotational positions obtained from the output of motion cueing algorithm are transmitted to the manipulator as actuator extensions calculated by inverse kinematics. Coordinate transformations in a robotic system are another important topic because motion sensation must be at the eye point of the pilot/driver. The major issue in simulators is motion cueing algorithm that relates the simulator input to the aircraft input. This is possible by the communication between motion control computer and simulation computer. Communication protocol in presented system in this study receives the translational accelerations and angular velocities from the simulation package program called FlightGear via UDP protocol. For this purpose, an XML script file is needed to receive related motion control inputs and take them into the motion cueing algorithm to be processed. Representation of whole system integration on the basis of components is in Figure 2.1.



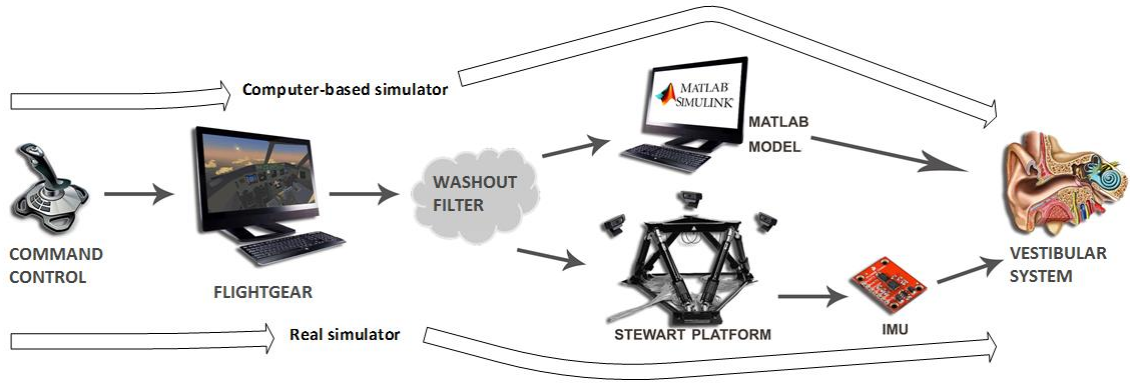


Figure 2.1 Representation of simulator system integration [29]

## 2.1 Kinematic Analysis of 6 DOF Parallel Manipulator – Stewart Platform

Stewart platform is a 6 DOF parallel manipulator including fixed and mobile platform on it and it consists of six parallel legs moving linearly and mounting fixed and mobile platform to each other. It was designed by D. Stewart in 1965 and it includes 3 translational and 3 rotational motion freedom. In this study, a Stewart platform that has 6 electrical motors will be described. Illustration of the Stewart platform can be seen in Figure 2.2.



Figure 2.2 6 DOF Stewart Platform Structure

Stewart platform has a structure based on producing force to the legs to simulate the effect of flight. It applies the force to actuators by defining current trajectory as acceleration, velocity and position. Optimization process of the motion control is needed to be performed depend on maximum acceleration and velocity values of the platform for all motions. In order to apply the force described, inverse kinematic model should be used to obtain desired leg lengths.

### 2.1.1 Inverse Kinematic Analysis

According to literature review, it can be seen that the Newton-Raphson method can be used for inverse and forward kinematics both. Although having a limited workspace can be considered as a disadvantage, position controllers can be used to increase the efficiency of workspace. In many researches the force was associated with acceleration instead of velocity.

Dynamic model of Stewart platform is as follows:

$$J \cdot \begin{bmatrix} \ddot{i} \\ \alpha \end{bmatrix} = H \cdot F + \begin{bmatrix} R \cdot F_{ext} \\ R \cdot M_{ext} \end{bmatrix} \quad (2.1)$$

In equation (2.1),  $J$  is the inertia matrix,  $H$  is the input-output force converter,  $F_{ext}$  is the external force,  $M_{ext}$  is the external moment,  $R$  is the rotation matrix,  $F$  is the actuator force,  $\ddot{i}$  is the translational acceleration and  $\alpha$  is the angular acceleration parameters.

The inverse kinematic analysis is based on finding leg lengths of platform when the central point coordinates of the mobile platform are known. For this aim the rotation matrix of the platform should be found first.

Rotation matrix is used to determine new coordinates of defining points of surfaces or objects which has been defined as surfaces in 2 or 3 dimensional space when they performed an orientation. Orientation of point P about fundamental axes is as shown in Figure 2.3, Figure 2.4 and Figure 2.5.

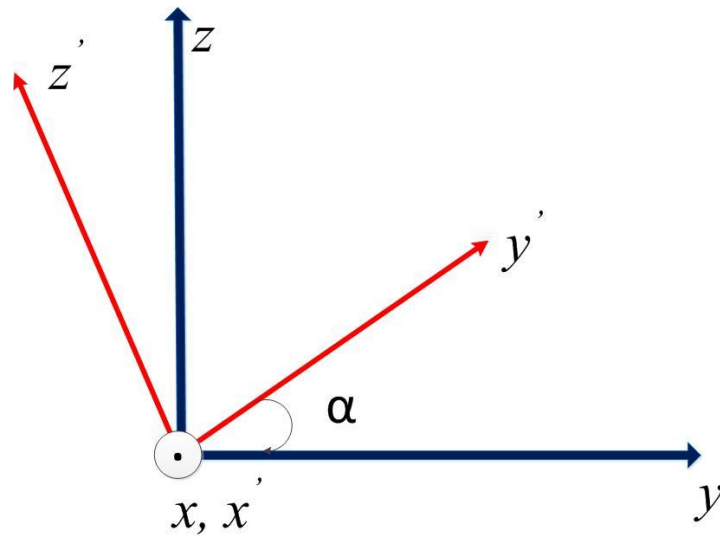


Figure 2.3 Orientation of P about x- axis

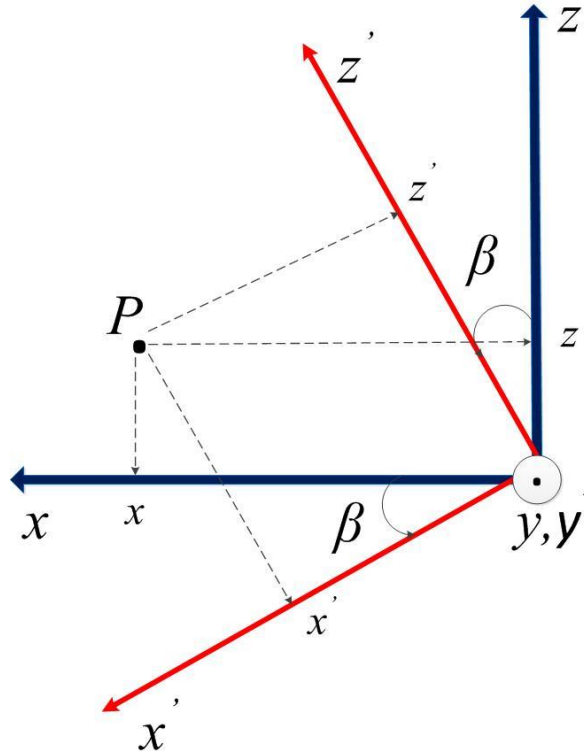


Figure 2.4 Orientation of P about y- axis

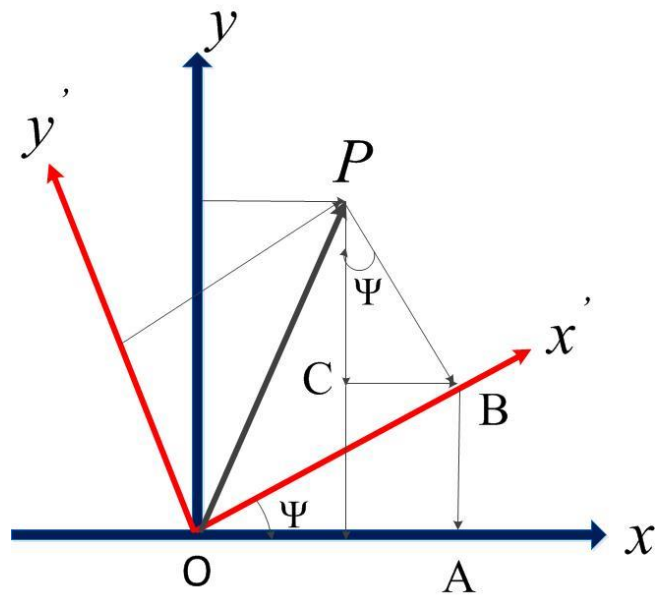


Figure 2.5 Orientation of P about z-axis

Generally there are 6 different rotation matrices as obtained below:

Rotation about x- axis:

$$P = i'x' + j'y' + k'z' = ix + jy + kz \quad (2.2)$$

$$x = |OA| - |BC| = x' \cos \psi - y' \sin \psi \quad (2.3)$$

$$y = |AB| + |PC| = x' \sin \psi + y' \cos \psi \quad (2.4)$$

$$z = z' \quad (2.5)$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R_x(\psi) \cdot \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \Rightarrow R_x(\psi) = \begin{bmatrix} C_\psi & -S_\psi & 0 \\ S_\psi & C_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

Similarly for equations of rotations about y- axis and z- axis are as follows:

$$R_y(\theta) = \begin{bmatrix} C_\theta & 0 & S_\theta \\ 0 & 1 & 0 \\ -S_\theta & 0 & C_\theta \end{bmatrix} \quad (2.7)$$

$$R_z(\varphi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C_\varphi & -S_\varphi \\ 0 & S_\varphi & C_\varphi \end{bmatrix} \quad (2.8)$$

$${}^P R_B = R_x(\psi) \cdot R_y(\theta) \cdot R_z(\varphi) \quad (2.9)$$

$$\begin{aligned} &= \begin{bmatrix} C_\psi & -S_\psi & 0 \\ S_\psi & C_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} C_\theta & 0 & S_\theta \\ 0 & 1 & 0 \\ -S_\theta & 0 & C_\theta \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & C_\varphi & -S_\varphi \\ 0 & S_\varphi & C_\varphi \end{bmatrix} \\ &= \begin{bmatrix} C_\psi \cdot C_\theta & -S_\psi & C_\psi \cdot S_\theta \\ S_\psi \cdot C_\theta & C_\psi & S_\psi \cdot S_\theta \\ -S_\theta & 0 & C_\theta \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & C_\varphi & -S_\varphi \\ 0 & S_\varphi & C_\varphi \end{bmatrix} \\ &{}^P R_B = \begin{bmatrix} C_\psi \cdot C_\theta & -S_\psi \cdot C_\varphi + C_\psi \cdot S_\theta \cdot S_\varphi & S_\psi \cdot S_\varphi + C_\psi \cdot S_\theta \cdot C_\varphi \\ S_\psi \cdot C_\theta & C_\psi \cdot C_\varphi + S_\psi \cdot S_\theta \cdot S_\varphi & -C_\psi \cdot S_\varphi + S_\psi \cdot S_\theta \cdot C_\varphi \\ -S_\theta & C_\theta \cdot S_\varphi & C_\theta \cdot C_\varphi \end{bmatrix} \quad (2.10) \end{aligned}$$

‘C’ and ‘S’ denote Cosine and Sine respectively.

Now equations of leg length and equations of joints are needed. Equations are as follows:

$$\begin{aligned} l_i^2 &= x^2 + y^2 + z^2 + R_p^2 + R_b^2 + 2 \cdot (R_{11} \cdot P_{ix} + R_{12} \cdot P_{iy}) \cdot (x - B_{ix}) + \\ &2 \cdot (R_{21} \cdot P_{ix} + R_{22} \cdot P_{iy}) \cdot (y - B_{iy}) + 2(R_{31} \cdot P_{ix} + R_{32} \cdot P_{iy}) \cdot z - 2 \cdot (x \cdot B_{ix} + y \cdot B_{iy}) \end{aligned} \quad (2.11)$$

Equation of top plate joints:

$$P_i = [R_p \cdot \cos(\sigma_i) ; R_p \cdot \sin(\sigma_i) ; 0] \quad (2.12)$$

Equation of bottom plate joints:

$$B_i = [R_b \cdot \cos(\rho_i) ; R_b \cdot \sin(\rho_i) ; 0] \quad (2.13)$$

Equations of x-, y- and z- coordinates:

$$x_i = \cos \psi \cdot \cos \theta \cdot P_{ix} + (\cos \psi \cdot \sin \theta \cdot \sin \varphi - \sin \psi \cdot \cos \varphi) \cdot P_{iy} + (\cos \psi \cdot \sin \theta \cdot \cos \varphi + \sin \theta \cdot \cos \varphi) \cdot P_{iz} + x - B_{ix} \quad (2.14)$$

$$y_i = \sin \psi \cdot \cos \theta \cdot P_{ix} + (\sin \psi \cdot \sin \theta \cdot \sin \varphi - \cos \psi \cdot \cos \varphi) \cdot P_{iy} + (\sin \psi \cdot \sin \theta \cdot \cos \varphi - \cos \psi \cdot \sin \varphi) \cdot P_{iz} + y - B_{iy} \quad (2.15)$$

$$z_i = -\sin \theta \cdot P_{ix} + \cos \theta \cdot \sin \varphi \cdot P_{iy} + \cos \theta \cdot \cos \varphi \cdot P_{iz} + z - B_{iz} \quad (2.16)$$

### 2.1.2 Forward Kinematic Analysis

Newton – Raphson method is also used in order to obtain forward kinematic equations. Obtaining related equations is as follows and the flowchart of forward kinematic that used in system is as shown in Figure 2.6:

$$F_i(X) = x^2 + y^2 + z^2 + r_p^2 + r_b^2 + 2 \cdot (r_{11} \cdot p_{ix} + r_{12} \cdot p_{iy}) \cdot (x - b_{ix}) + 2 \cdot (r_{21} \cdot p_{ix} + r_{22} \cdot p_{iy}) \cdot (y - b_{iy}) + 2 \cdot (r_{31} \cdot p_{ix} + r_{32} \cdot p_{iy}) \cdot z - 2 \cdot (x \cdot b_{ix} + y \cdot b_{iy}) - l_i^2 = 0 \quad (2.17)$$

$$X = x, y, z, \alpha, \beta, \gamma \quad i = 1, 2, \dots, 6 \quad (2.18)$$

By taking partial derivative of each variable in the function, the derivative of the function can be obtained.

$$J_{(i,j)} = \frac{dF_i}{dX_j}(X) \quad , \quad \forall \{i, j\} \in \{1, \dots, 6\} \quad (2.19)$$

By using following equation, unknowns can be found:

$$X^{(n+1)} = X^{(n)} - [J \cdot (X^{(n)})]^{-1} \cdot F(X^{(n)}) \quad (2.20)$$

The following equation can be used as initial condition:

$$\mathbf{X}^{(0)} = (x_0, y_0, z_0, \alpha_0, \beta_0, \gamma_0) \quad (2.21)$$

When necessary iteration number is reached, the iteration is stopped. Iteration is shown as  $\varepsilon$  and is as shown as follows:

$$|F_i(x)| \leq \varepsilon \quad (2.22)$$

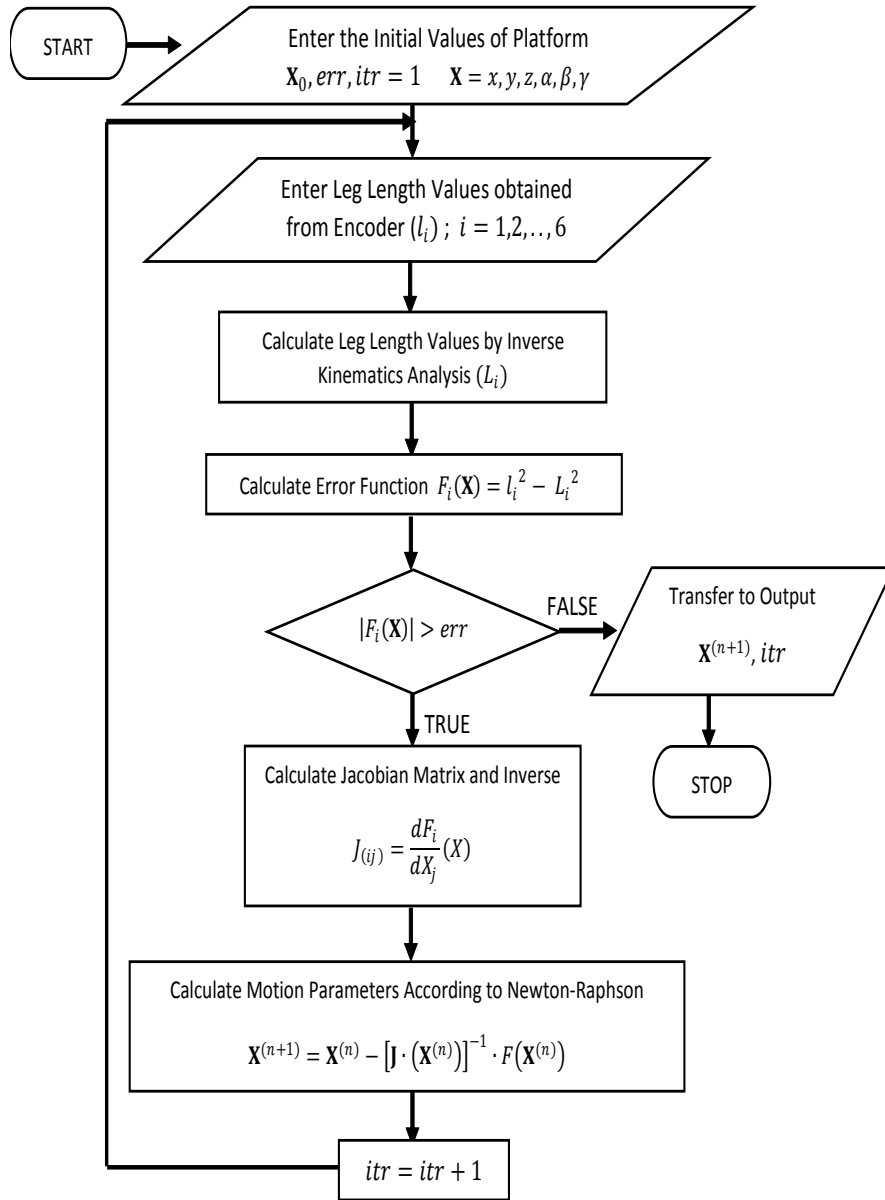


Figure 2.6 Forward kinematic analysis flowchart [29]

Forward kinematic analysis Matlab m. file is as shown in Appendix B.

## 2.2 Coordinate Transformation

Coordinate transformation in a simulator is an important topic because motion sensation must be at the eye point of the pilot/driver of the simulator in order to obtain more realistic

motion perception. Flight simulator system and related coordinates on the platform are shown in Figure 2.6. In progress, equations of coordinate transformation are shown:

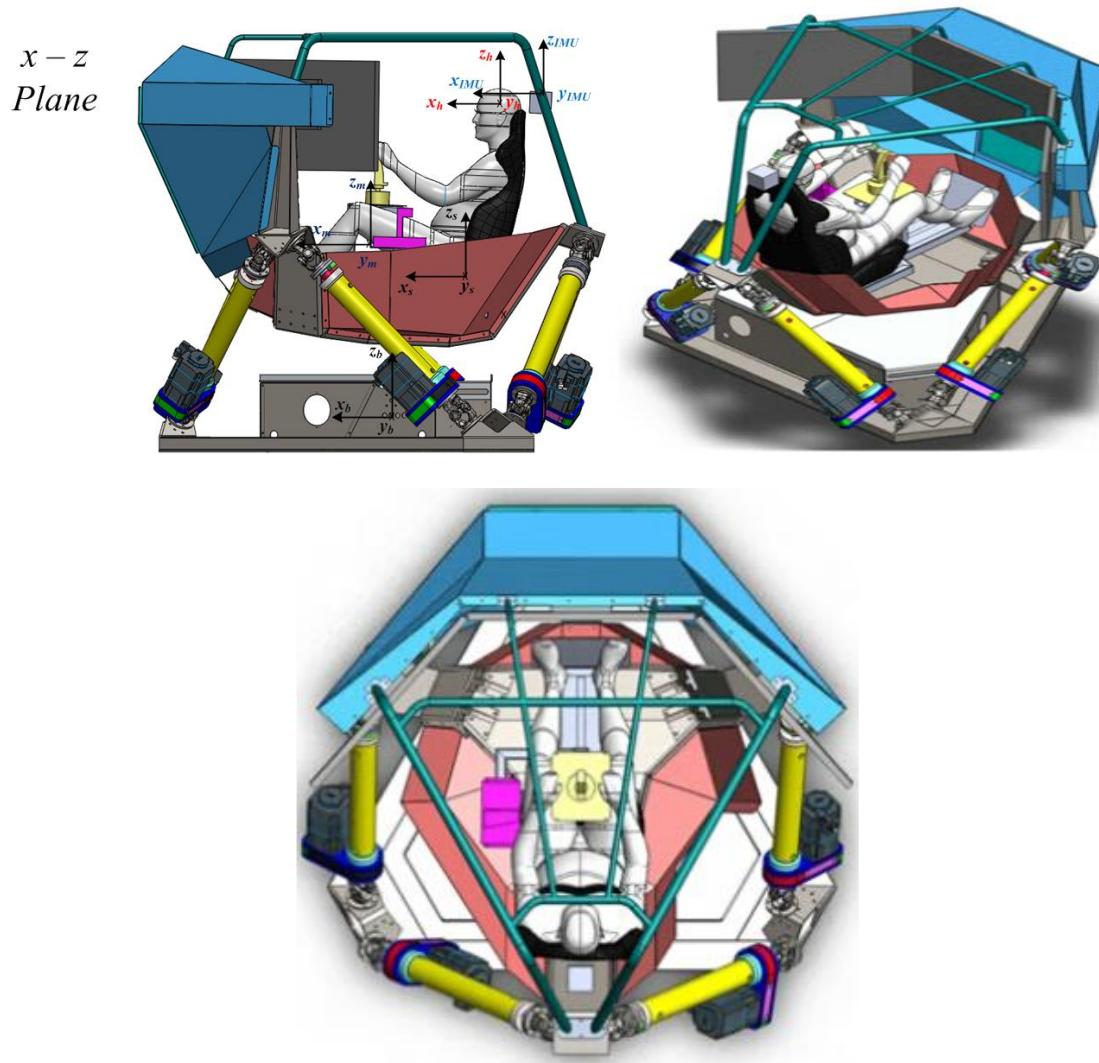


Figure 2.7 Representation of coordinate frame of the flight simulator system

$$L_b = \{x_b, y_b, z_b\} \quad (2.23)$$

$$L_m = \{x_m, y_m, z_m\} \quad (2.24)$$

$$L_s = \{x_s, y_s, z_s\} \quad (2.25)$$

$$L_{IMU} = \{x_{IMU}, y_{IMU}, z_{IMU}\} \quad (2.26)$$

$$L_h = \{x_h, y_h, z_h\} \quad (2.27)$$

Since reference coordinate frame is on x – z plane:

$${}^b\vec{r}_m = \{0, 0, c_m\} \quad (2.28)$$

$${}^b\vec{r}_s = \{-a_s, 0, c_s\} \quad (2.29)$$

$${}^b\vec{r}_h = \{-a_h, 0, c_h\} \quad (2.30)$$

$${}^b\vec{r}_{IMU} = \{-a_{IMU}, 0, c_{IMU}\} \quad (2.31)$$

The scheme of coordinate sets of the platform is as shown in Figure 2.7.

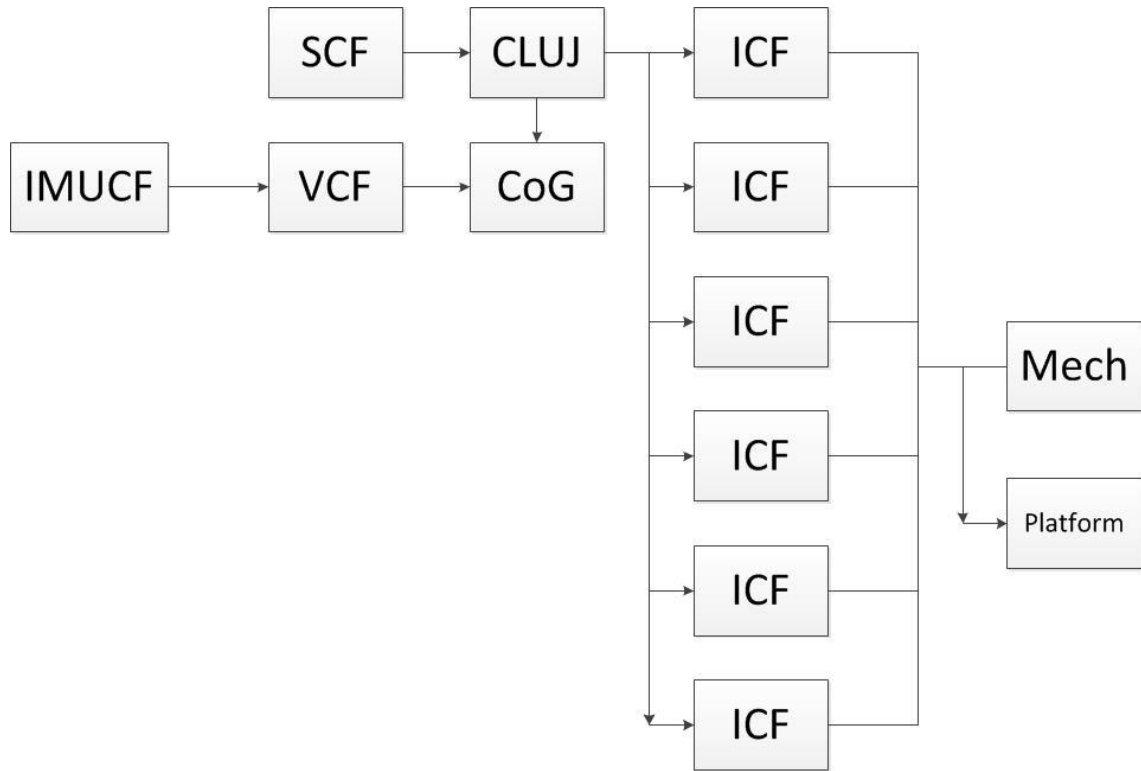


Figure 2.8 Coordinate set scheme of the platform

**CAG** : Center of Lower Universal Joint

**IAS** : Inertial Coordinate Frame of Linear Actuators

**CoG** : Center of Gravity

**VEST** : Vestibular System Coordinate Frame

**IMU** : IMU Coordinate Frame

**SAS** : Seat Coordinate Frame

**Mech** : Collection of Mechanical Coordinate Frames

All coordinate frames constitutes collection of the mechanical coordinate frames. Each frame has x-, y- and z- axis. CAG represents the motors which driving the platform. VEST



and IMU have the same reference coordinate frames. There is only a translational displacement through x- axis. Coordinate transformation equations are as follows:

$$T_m^b = \begin{bmatrix} R_{11} & R_{12} & R_{13} & 0 \\ R_{21} & R_{22} & R_{23} & 0 \\ R_{31} & R_{32} & R_{33} & c_m \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.32)$$

$$T_s^b = \begin{bmatrix} R_{11} & R_{12} & R_{13} & -a_s \\ R_{21} & R_{22} & R_{23} & 0 \\ R_{31} & R_{32} & R_{33} & c_s \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.33)$$

$$T_{IMU}^b = \begin{bmatrix} R_{11} & R_{12} & R_{13} & -a_{IMU} \\ R_{21} & R_{22} & R_{23} & 0 \\ R_{31} & R_{32} & R_{33} & c_{IMU} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.34)$$

$$T_h^b = \begin{bmatrix} R_{11} & R_{12} & R_{13} & -a_h \\ R_{21} & R_{22} & R_{23} & 0 \\ R_{31} & R_{32} & R_{33} & c_h \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.35)$$

$$T_{IMU}^b \cdot T_h^{IMU} = T_h^b \Rightarrow T_h^{IMU} = (T_{IMU}^b)^{-1} \cdot T_h^b \quad (2.36)$$

$$T_h^{IMU} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & -a_{IMU} \\ R_{21} & R_{22} & R_{23} & 0 \\ R_{31} & R_{32} & R_{33} & c_{IMU} \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} R_{11} & R_{12} & R_{13} & -a_h \\ R_{21} & R_{22} & R_{23} & 0 \\ R_{31} & R_{32} & R_{33} & c_h \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.37)$$

$$T^{-1} = \begin{bmatrix} R^T & -R^T p \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.38)$$

$$T_{IMU}^b = \begin{bmatrix} (R_{31}(R_{12}R_{23} - R_{13}R_{22}))/... & (R_{32}(R_{12}R_{23} - R_{13}R_{22}))/... & (R_{33}(R_{12}R_{23} - R_{13}R_{22}))/... & (R_{22}R_{33}a_{IMU} - R_{23}R_{32}a_{IMU} - ... \\ x - ... & x - ... & x - ... & R_{12}R_{23}c_{IMU} + R_{13}R_{22}c_{IMU})/... \\ (R_{21}(R_{12}R_{33} - R_{13}R_{32}))/... & (R_{22}(R_{12}R_{33} - R_{13}R_{32}))/... & (R_{23}(R_{12}R_{33} - R_{13}R_{32}))/... & x - ... \\ x + ... & x + ... & x + ... & (a_h(R_{22}R_{33} - R_{23}R_{32}))/... \\ (R_{11}(R_{22}R_{33} - R_{23}R_{32}))/... & (R_{12}(R_{22}R_{33} - R_{23}R_{32}))/... & (R_{13}(R_{22}R_{33} - R_{23}R_{32}))/... & x + ... \\ x, & x, & x, & (c_h(R_{12}R_{23} - R_{13}R_{22}))/... \\ (R_{21}(R_{11}R_{33} - R_{13}R_{31}))/... & (R_{22}(R_{11}R_{33} - R_{13}R_{31}))/... & (R_{23}(R_{11}R_{33} - R_{13}R_{31}))/... & x, \\ x - ... & x - ... & x - ... & (a_h(R_{21}R_{33} - R_{23}R_{31}))/... \\ (R_{31}(R_{11}R_{23} - R_{13}R_{21}))/... & (R_{32}(R_{11}R_{23} - R_{13}R_{21}))/... & (R_{33}(R_{11}R_{23} - R_{13}R_{21}))/... & x - ... \\ x - ... & x - ... & x - ... & (R_{21}R_{33}a_{IMU} - R_{23}R_{31}a_{IMU} - \\ (R_{11}(R_{21}R_{33} - R_{23}R_{31}))/... & (R_{12}(R_{21}R_{33} - R_{23}R_{31}))/... & (R_{13}(R_{21}R_{33} - R_{23}R_{31}))/... & R_{11}R_{23}c_{IMU} + R_{13}R_{21}c_{IMU})/ \\ x, & x, & x, & (c_h(R_{11}R_{23} - R_{13}R_{21}))/... \\ (R_{31}(R_{11}R_{22} - R_{12}R_{21}))/... & (R_{32}(R_{11}R_{22} - R_{12}R_{21}))/... & (R_{33}(R_{11}R_{22} - R_{12}R_{21}))/... & x, \\ x - ... & x - ... & x - ... & (R_{21}R_{32}a_{IMU} - R_{22}R_{31}a_{IMU} - ... \\ (R_{21}(R_{11}R_{32} - R_{12}R_{31}))/... & (R_{22}(R_{11}R_{32} - R_{12}R_{31}))/... & (R_{23}(R_{11}R_{32} - R_{12}R_{31}))/... & R_{11}R_{22}c_{IMU} + R_{12}R_{21}c_{IMU})/... \\ x + ... & x + ... & x + ... & (a_h(R_{21}R_{32} - R_{22}R_{31}))/... \\ (R_{11}(R_{21}R_{32} - R_{22}R_{31}))/... & (R_{12}(R_{21}R_{32} - R_{22}R_{31}))/... & (R_{13}(R_{21}R_{32} - R_{22}R_{31}))/... & x - ... \\ x, & x, & x, & (c_h(R_{11}R_{22} - R_{12}R_{21}))/... \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.39)$$

‘ $R$ ’ is the rotation matrix in the forward kinematic equations of the platform.

### 2.3 Communication

Mathematical model outputs which convert the command controls into motion in the FlightGear simulation environment are supplied to the platform via UDP protocol. During this process datas received from simulation are determined by creating an XML script file. Thus six parameters including translational accelerations and angular velocities at the position of pilot are provided to be exported.

Sample frequency of 240 Hz is sufficient for simulation. Visual simulation software frequency is calculated to be 120 Hz by using following equation:

$$frequency \succ 2 \cdot f_{\max} \quad (2.40)$$

According to JAR standards flight simulation frequency of 5 Hz is a general acceptance for Level D simulators so in this study the frequency of this level was used.

In order to get motion parameters from FlightGear, a script file containing basic reference codes used in software must be created. Reference codes used in FlightGear are as follows:

```
/orientation/roll-deg  
/orientation/pitch-deg  
/orientation/heading-deg  
  
/orientation/roll-rate-degps  
/orientation/pitch-rate-degps  
/orientation/yaw-rate-degps  
  
/orientation/side-slip-rad  
/orientation/side-slip-deg  
/orientation/alpha-deg
```

These reference codes generates parameters which are used by dynamic model run in FlightGear.

The script file exports motion parameter generated from reference codes via determined port.

Sample codes are as follows:

- **Position**

```
/position/  
/position/altitude-ft ()  
/position/altitude-agl-ft (22.46983965)  
/position/altitude-ft (28.24368289)  
/position/ground-elev-ft (-0.43513529)  
/position/ground-elev-m (-0.1326292364)  
/position/latitude-deg (37.61371436)  
/position/latitude-string (37*36 49.4N)  
/position/longitude-deg (-122.3576508)  
/position/longitude-string (-122*21 27.5W)  
/position/sea-level-radius-ft (20899648.76)
```

- **Orientation**

```
/orientation/roll-deg  
/orientation/pitch-deg
```

/orientation/heading-deg  
/orientation/roll-rate-degps  
/orientation/pitch-rate-degps  
/orientation/yaw-rate-degps  
/orientation/side-slip-rad  
/orientation/side-slip-deg  
/orientation/alpha-deg

- **Velocities**

/velocities/airspeed-kt  
/velocities/mach  
/velocities/speed-north-fps  
/velocities/speed-east-fps  
/velocities/speed-down-fps  
/velocities/uBody-fps  
/velocities/vBody-fps  
/velocities/wBody-fps  
/velocities/vertical-speed-fps  
/velocities/glideslope

- **Acceleration**

/accelerations/nlf  
/accelerations/ned/north-accel-fps\_sec  
/accelerations/ned/east-accel-fps\_sec  
/accelerations/ned/down-accel-fps\_sec  
/accelerations/pilot/x-accel-fps\_sec  
/accelerations/pilot/y-accel-fps\_sec  
/accelerations/pilot/z-accel-fps\_sec

FlightGear supports multi inputs and multi outputs. In order to export parameters generated by script file, Generic Protocol is used in FlightGear. This protocol exports datas as UDP packages. Related FlightGear options interface is as shown in Figure 2.9.



Figure 2.9 FlightGear input-output interface

This communication file may contains input or output parameters. Reference codes are used to decide which parameters will be used. A sample output file is as follows:

```
--generic=file,out,1,/tmp/data.xml,myproto
```

If the operating system used is Windows, it is suggested that number of COM which will be used to be COM9 or higher. A sample is as follows:

```
--generic=\\.\COM10,out,1,/tmp/data.xml,myproto
```

A sample XML script file is as follows:

```
<?xml version="1.0"?>
<PropertyList>
  <generic>

    <output>
      <binary_mode>>false</binary_mode>
      <line_separator></line_separator>
      <var_separator></var_separator>
      <preamble></preamble>
      <postamble></postamble>

    <chunk>
      <!-- First output chunk definition -->
    </chunk>

    <chunk>
      <!-- Next output chunk definition etc... -->
```

```

    </chunk>
</output>

<input>
  <line_separator></line_separator>
  <var_separator></var_separator>

  <chunk>
    <!-- First input chunk definition -->
  </chunk>

  <chunk>
    <!-- Next input chunk definition etc... -->
  </chunk>
</input>

</generic>
</PropertyList>

```

UDP (User Datagram Protocol) makes a procedure available to send and receive data over Ethernet from user program with a minimum of protocol mechanism. No information concerning the transferred data is returned in case of communication via UDP. The communication takes place via ports on both the send and receive sides. As opposed to TCP/IP, you do not need to program any connection buildup or closing. Following steps describe the model of the UDP communication:

- For reception, in the command you address the port that you want to use on your component for the communication job.
- When sending data you specify the IP address of the target system, the port number for the data on the target system and the port number of your component.
- You can specify whether the port should remain reserved on your end after the communication job has been executed.
- UDP is not a secured model. Therefore, data may be lost during transfer. A secured data transfer must be programmed in your application, e.g. by acknowledging the receipt of the data.
- Function `_udpReceive` allows you to transfer the data of a transfer protocol in the return structure, if several data protocols have been returned with `_udpReceive`, the “oldest” data protocol is returned. UDP communication at the SIMOTION is as shown in Figure 2.10.

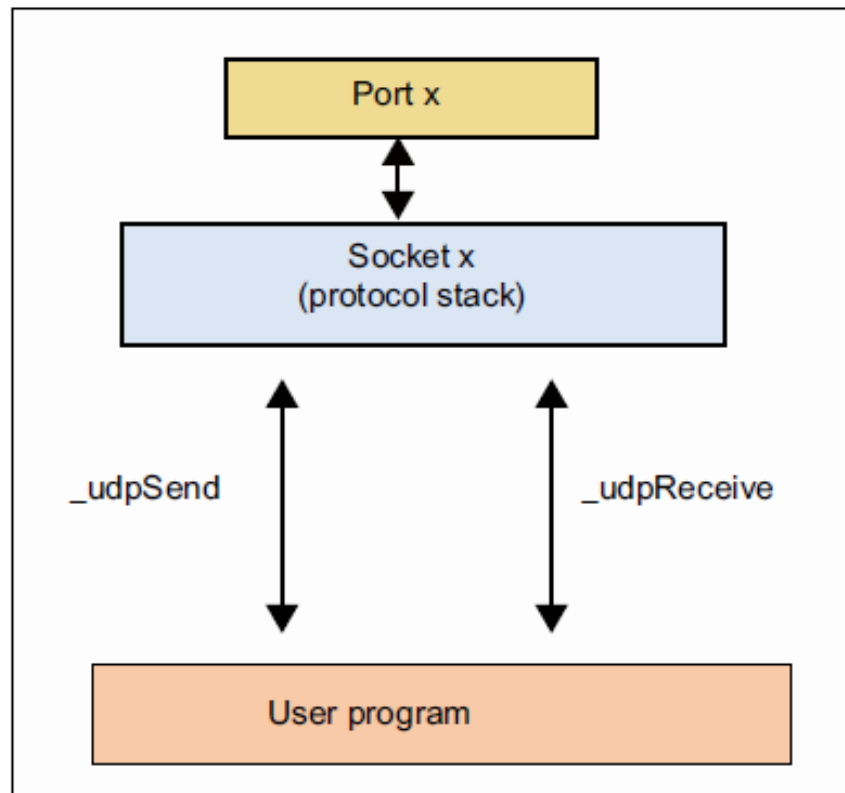


Figure 2.10 UDP communication model

### MOTION CUEING AND SYSTEM INTEGRATION

Motion cueing algorithm is a must that provides the motion of a simulator within boundaries of limited robotic workspace. 6 DOF parallel manipulators are mostly used for flight simulation. Inputs to the manipulator are simulated aircraft angular velocities and translational accelerations as references to generate necessary motions. The mobile platform of the mechanism should return its neutral position in order to start the next motion sequence. This process is called as “Washout”. Washout motion must be under human motion sensation thresholds which is covered in literature by several researchers. For the sake of sensing continuous translational accelerations, the platform tilts itself and uses gravity vector to create artificial motion sensation translationally which is called “Tilt Coordination”. Force effect caused by continuous acceleration of aircraft is also provided by Tilt Coordination. Tilt coordination is provided by applying 2<sup>nd</sup> order low-pass filters for x- and y- axes in the algorithm. In this section firstly vestibular system is going to be explained to validate that motions obtained from outputs of the motion cueing algorithm are under the threshold values of human sensation then classical and optimal motion cueing algorithm structures are going to be shown.

#### 3.1 Vestibular System

For human perception studies human motion perception system (vestibular system) constitutes the basis. Vestibular can be seen in Figure 3.1. Transfer functions between angular velocity obtained from output of the washout filters and angular velocity received from the inputs at each semi-circular canal are as follows:



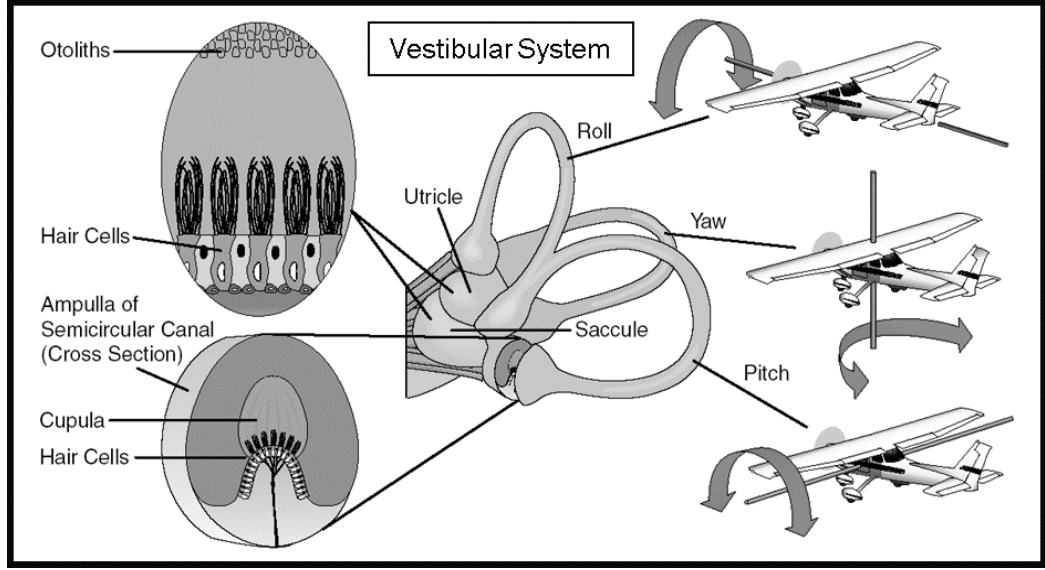


Figure 3.1 Vestibular System representation

$$W_{si}(s) = \frac{\hat{\omega}_i(s)}{\omega_i(s)} = \frac{G_{sc} \tau_1 \tau_a s^2 (1 + \tau_L s)}{(1 + \tau_a s)(1 + \tau_1 s)(1 + \tau_2 s)} \quad (3.1)$$

$\tau_1, \tau_2, \tau_a, \tau_L$  are time constants and  $G_{sc}$  is the gain of the system.

Transfer functions between translational acceleration obtained from output of the washout filters and translational acceleration received from the inputs at otoliths are as follows:

$$W_{oi}(s) = \frac{\hat{a}_i(s)}{a_i(s)} = \frac{K_{oto}(s + a_0)}{(s + b_0)(s + b_1)} \quad (3.2)$$

$K_{oto}$  is the gain and  $1/\tau_L$  is  $a_0$   $1/\tau_1$  is  $b_0$   $1/\tau_2$  is  $b_1$ .

Index  $i$  represents rotational motions in Euler angles (roll – pitch - yaw). Related vestibular system parameters obtained from literature review and human motion perception thresholds are as shown in Table 3.1.

Table 3.1 Human Motion System Parameters

	Chen [30]			Houck and Telban [8]		
	x	y	z	x	y	z
$a_0(s^{-1})$	0,08	0,08	0,08	0,1	0,1	0,1
$b_0(s^{-1})$	0,18	0,18	0,18	0,2	0,2	0,2
$b_1(s^{-1})$	1,51	1,51	1,51	62,5	62,5	N/A
$K_{oto}(s^{-1})$	1,5	1,5	1,5	0,94	0,94	0,01
	$\alpha$	$\beta$	$\gamma$	$\alpha$	$\beta$	$\gamma$
$\tau_1(s)$	6,1	5,3	10,2	5,73	5,73	5,73

$\tau_2(s)$	0,1	0,1	0	0,01	0,01	0,01
$\tau_a(s)$	30	30	30	80	80	80
$\tau_l(s)$	0	0	0	0,06	0,06	0,06
$G_{sc}$	1	1	1	28,65	28,65	28,65

Table 3.2 Threshold values of human motion perception

<b>Human Motion Perception Thresholds [26]</b>					
Roll (deg/s)	Pitch (deg/s)	Yaw (deg/s)	$a_x(m/s^2)$	$a_y(m/s^2)$	$a_z(m/s^2)$
2	2	1,6	0,17	0,17	0,28

Related signals are transmitted to the neural constructions through nerve fibers by the receptor cells of the semi-circular canals and otoliths. Otolith organ consists of two structures: Ultricle that senses horizontal motions and saccule that senses vertical motions. These structures can also be seen in Figure 3.1.

### 3.2 Classical Motion Cueing Algorithm

Classical motion cueing algorithm constitutes the basis of motion cueing algorithms and includes 1<sup>st</sup> order and 2<sup>nd</sup> order high-pass washout filters that are used to bring the manipulator to its neutral position in its workspace under the threshold values of human sensation for all motions. Thus, vestibular system model is used to check the motion whether it is under the threshold value or not. Classical motion cueing algorithm structure is as shown in Figure 3.2.

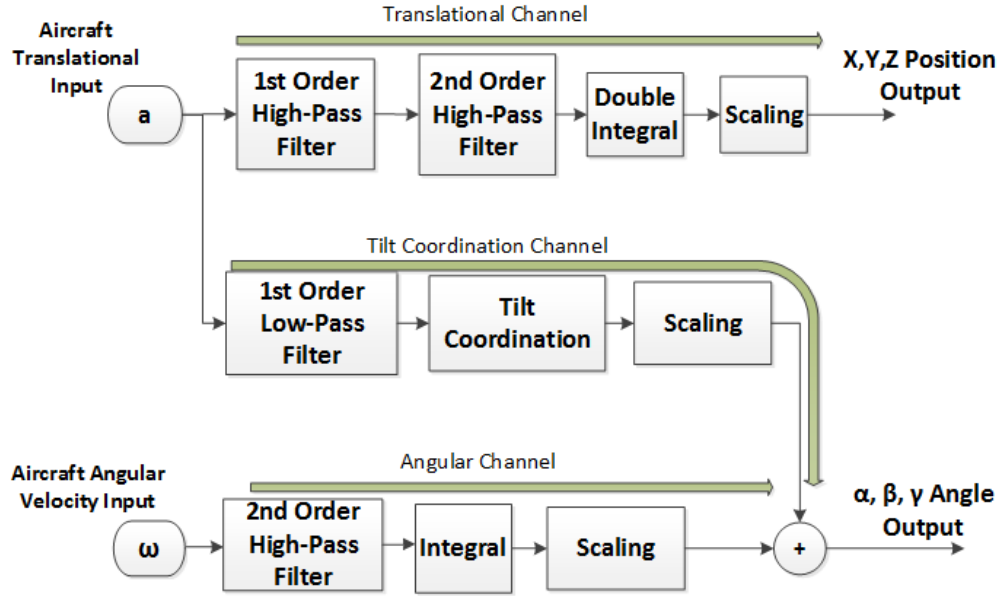


Figure 3.2 Structure of classical motion cueing algorithm

The channel which is responsible for simulating translational motion of the aircraft is the translational channel. Inputs of the translational channel are translational accelerations of the aircraft. Similarly angular channel is responsible for simulating rotational motions of the aircraft.

Signals that are filtered are transmitted to the drivers of motors as position and angle values by taking the double integrals of translational acceleration inputs and by taking the single integral of angular velocities.

During washout platform acceleration values are kept under the threshold values of human sensation in order not to feel the motion presence and provide system sufficient workspace. This combination creates a realistic flight feeling.

1<sup>st</sup> order and 2<sup>nd</sup> order high pass filter and 1<sup>st</sup> order low pass filter representations are as follows:

$$1. \quad HP = \frac{s}{s + \omega} \quad (3.3)$$

$$2. \quad HP = \frac{s}{s^2 + 2 \cdot s \cdot \omega \cdot \zeta + \omega^2} \quad (3.4)$$

$$1. \quad LP = \frac{\omega}{s + \omega}$$

(3.5)

$\zeta$  is damping factor and  $\omega$  is cornering frequency of washout filters.

Although the classical motion cueing algorithm has a simple structure it has limitations due to having constant washout filter parameters. To overcome this problem optimal and adaptive washout filter designs can be performed. So washout filter parameters are going to be calculated for each sample time of the system.

### 3.2.1 Tilt Coordination

For the sake of sensing continuous translational accelerations, the platform tilts itself and uses gravity vector to create artificial motion sensation translationally which is called “Tilt Coordination”. Force effect caused by continuous acceleration of aircraft is also provided by Tilt Coordination. Tilt coordination is provided by applying 2<sup>nd</sup> order low-pass filters for x- and y- axes in the algorithm. Tilt coordination effect is as shown in Figure ...

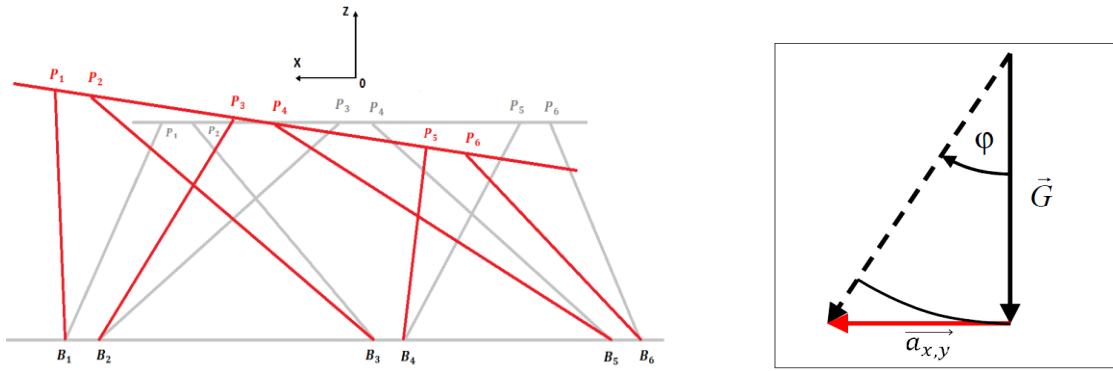


Figure 3.3 Tilt coordination effect

Tilt coordination formulation is as follows:

$$\varphi_{x,y} = \arctan\left(\frac{a_{x,y}}{G}\right) \quad (3.6)$$

$\varphi_{x,y}$  shows tilt angles and  $a_{x,y}$  shows accelerations through x- and y- directions.  $G$  is the gravity.

Tilt coordination is a low motion effect. Therefore before the calculation of tilt angles related accelerations should be passed through low pass filters. Thus aircraft inertia can be felt by the pilot. Several researches showed that on flight simulators tilt coordination effect is more effective than driving simulators [30].

Calibration process of classical washout filters is also important in motion cueing algorithm design. There are position saturation values through x-, y- and z- axes and angle

saturation values about x-, y- and z- axes of the manipulator according to its production catalog. If filter gains are higher than the saturation values that manipulator is capable of, there is possibility of the occurrence of mechanical failures.

### 3.2.2 Calibration Process of Classical Washout Filters

Calibration process of classical washout filters is also important in motion cueing algorithm design. There are position saturation values through x-, y- and z- axes and angle saturation values about x-, y- and z- axes of the manipulator according to its production catalog. If filter gains are higher than the saturation values that manipulator is capable of, there is possibility of the occurrence of mechanical failures. Parameters that will be calibrated are as shown in Table A.1.

Calibration process includes following steps:

- I. **Determination of Maximum Position and Angle Values According to Mechanical Ability of the Manipulator:** Maximum position through x-, y- and z- axes and maximum angle values of 6 DOF parallel manipulator about these axes due to its mechanical ability are obtained from catalog values of the manipulator. These values form the saturation values of signals obtained from the output of the washout filter and scaled according to related unit conversion. Position limits for translational motions and angle limits for rotational motions are as shown in Table 3.3, Figure 3.4, Figure 3.5 and Figure 3.6. “Conversion” block in Simulink including related saturation values is as shown in Figure A.3 and interior structure of this block is as shown in Figure A.4.

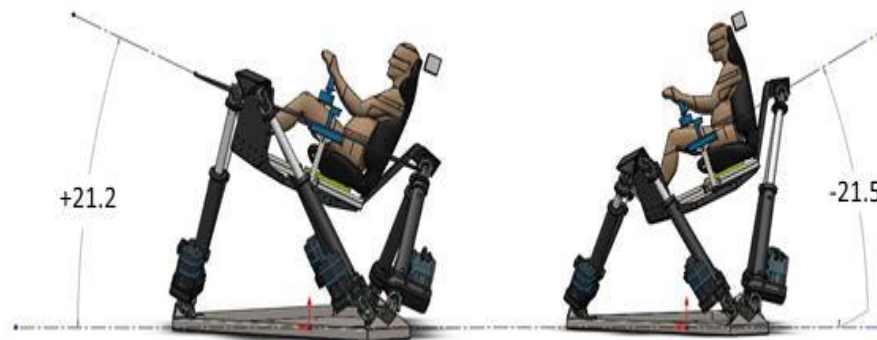


Figure 3.4 Motion limits for pitch motion

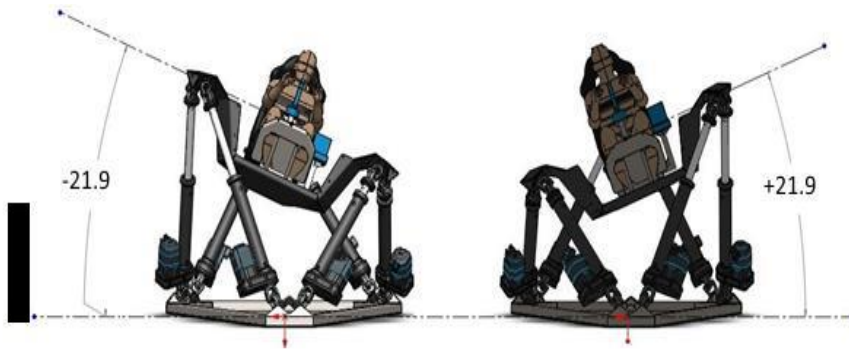


Figure 3.5 Motion limits for roll motion

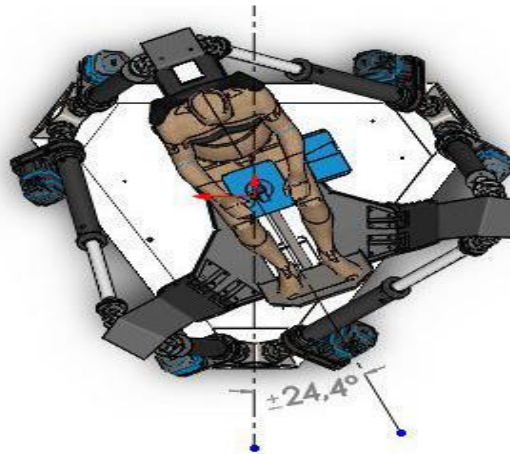


Figure 3.6 Motion limits for yaw motion

Table 3.3 Translational motion limits of the platform

Through x-axis	+460.5	-460.5	mm
Through y- axis	+425.0	-605	mm
Through z- axis	+321.0	-359	mm

- II. Determination of Maximum Translational Accelerations and Angular Velocities Manipulator Can Do:** Translational accelerations and angular velocities that achieve the manipulator to its saturations are determined by applying a square wave or a constant signal to manipulator as a reference input. These input signals are applied to

the manipulator to obtain maximum position and angle values according to reference axes that are shown in Figure 3.7.

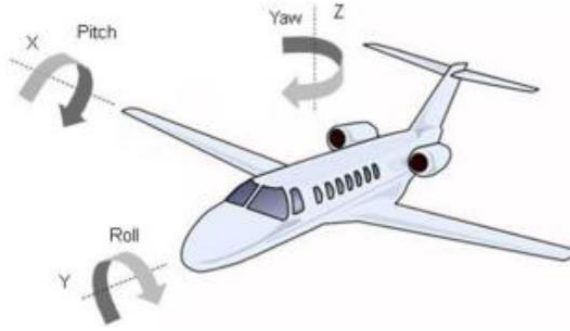


Figure 3.7 Representation of reference axes of the aircraft

**III. Creation of Scenarios for Coupled Motions:** Position and angle values and the acceleration and angular velocity values provide these positions and angles for the coupled motions due to the mechanical abilities of the manipulator are determined.

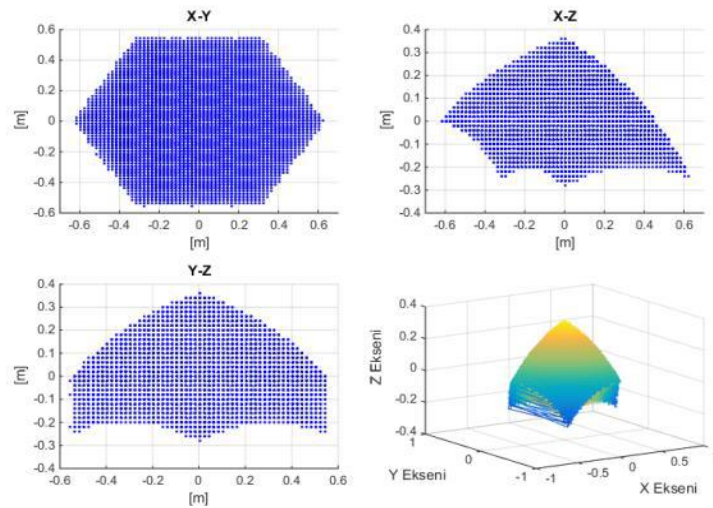


Figure 3.8 Results of workspace analysis of the manipulator

- IV. Determination of Mechanical Abilities of the Manipulator:** During the flight, by comparing the acceleration and angular velocity values obtained from the aircraft with maximum acceleration and angular velocity values of the manipulator, the output of the washout filter is multiplied by related gain value. Otherwise the manipulator will try to make a motion that will lead to mechanical failure due to its insufficient ability.
- V. Determination of Calibration Method:** According to obtained graphics, filter parameters (natural frequency ( $\omega$ ), damping ratio ( $\zeta$ )) are obtained by using trial-and-error method.

- VI. Interpretation of datas obtained from the Aircraft:** If the developed algorithm is an optimal algorithm that is controlled by using adaptive control method, because the washout filter parameters are adjusted in real time for each sampling time, there will be no need to use the trial-and-error method. The block in Simulink model in which scaling of the datas received from the aircraft can be seen in Figure A.1.
- VII. Adjustment of the Filters According to Manipulator Motions:** Another issue is achievement of the manipulator to desired input motion at optimum level. Otherwise the mechanical ability of the manipulator will be restricted and this will delay the calibration of the developed algorithm.
- VIII. Adjustment of Filter Gain Values According to Flight Datas:** If the reference input obtained as the flight data is received from a developed dynamic aircraft model, related datas must be brought to healthy and meaningful case before the calibration because the probability of occurrence that some motion may be in the opposite direction of the expected motion. This process is as shown in Simulink block that can be seen in Figure A.1.
- IX. Checking the Accuracy of Datas Received from the Aircraft:** The accuracy of the received datas can be measured through literature.
- X. Calibration of Washout Filter Parameters:** Calibration parameters of filters are as shown in Table A.1. Related filters are in the Simulink block that contains the Motion Cueing Algorithm placed just after the block that you can obtain the flight datas. Simulink block that contains the washout filters can be seen red in color in Figure A.2. Block diagram including the whole calibration process is as shown in Figure 3.9.

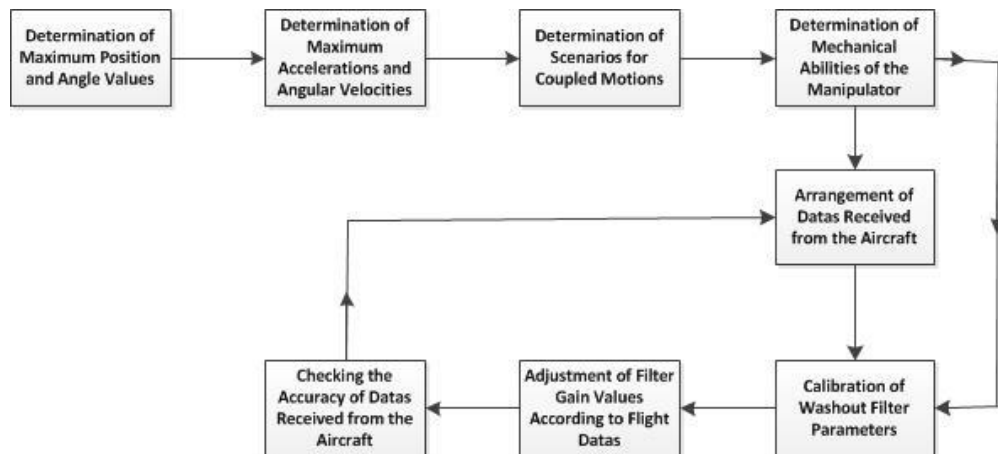


Figure 3.9 Representation of block diagram of the calibration process



### 3.3 Optimal Motion Cueing Algorithm

The algorithm development with angular velocity input for the longitudinal (pitch/surge) mode is given below [8]. The input  $u$  is formulated as:

$$u = \begin{bmatrix} \dot{\theta} \\ a_x \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (3.7)$$

where  $\dot{\theta}$  is angular velocity and  $a_x$  is the translational acceleration with each term respectively is  $u_1$  and  $u_2$ .

The sensed rotational motion is related to  $u_1$  by the semi-circular model transfer function:

$$\hat{\theta} = \frac{G_{SCC} \tau_1 \tau_a s^2 (1 + \tau_L s)}{(1 + \tau_a s)(1 + \tau_1 s)(1 + \tau_2 s)} \cdot u_1 \quad (3.8)$$

where  $\tau_1$ ,  $\tau_2$ ,  $\tau_a$  and  $\tau_L$  are the time constants of the semi-circular canals and  $G_{SCC}$  is the angular velocity threshold value. Equation (3.8) can be rewritten as:

$$\hat{\theta} = \frac{T_4 s^3 + T_3 s^2}{s^3 + T_2 s^2 + T_1 s + T_0} \cdot u_1 \quad (3.9)$$

where

$$T_0 = \frac{1}{\tau_a \tau_1 \tau_2}, \quad T_1 = \frac{\tau_a + \tau_1 + \tau_2}{\tau_a \tau_1 \tau_2}, \quad T_2 = \frac{\tau_1 \tau_2 + \tau_a (\tau_1 + \tau_2)}{\tau_a \tau_1 \tau_2}, \quad T_3 = \frac{G_{SCC}}{\tau_2} \text{ and}$$

$$T_4 = \frac{G_{SCC} \tau_L}{\tau_2}$$

and can be defined in state space notation as:

$$\begin{aligned} \dot{x}_{SCC} &= A_{SCC} x_{SCC} + B_{SCC} u \\ \hat{\theta} &= C_{SCC} x_{SCC} + D_{SCC} u \end{aligned} \quad (3.10)$$

where in observer canonical form,

$$A_{SCC} = \begin{bmatrix} -T_2 & 1 & 0 \\ -T_1 & 0 & 1 \\ -T_0 & 0 & 0 \end{bmatrix}, B_{SCC} = \begin{bmatrix} T_3 - T_2 T_4 & 0 \\ -T_1 T_4 & 0 \\ -T_0 T_4 & 0 \end{bmatrix}, C_{SCC} = [1 \ 0 \ 0] \text{ and}$$

$$D_{SCC} = [T_4 \ 0]$$

The otolith model of the vestibular system can be defined in state space notation as:

$$\begin{aligned} \dot{x}_{OTO} &= A_{OTO} x_{OTO} + B_{OTO} u \\ \hat{f}_x &= C_{OTO} x_{OTO} + D_{OTO} u \end{aligned} \quad (3.11)$$

where  $x_{OTO}$  are the otolith states, and

$$A_{OTO} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -b & -a & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -b & -a \end{bmatrix}, B_{OTO} = \begin{bmatrix} c & 0 \\ d - ac & 0 \\ e & 0 \\ 0 & f \\ 0 & h - af \end{bmatrix},$$

$$C_{OTO} = [1 \ 0 \ 0 \ 1 \ 0], D_{OTO} = [-G_{OTO} K'_{OTO} R_{sz} \ 0]$$

Semi-circular model and otolith model can be combined to form a single representation for the human vestibular model:

$$\begin{aligned} \dot{x}_v &= A_v x_v + B_v u \\ \hat{y}_v &= C_v x_v + D_v u \end{aligned} \quad (3.12)$$

where  $x_v$  and  $\hat{y}_v$  are the combined states respectively, and

$$A_v = \begin{bmatrix} A_{SCC} & 0 \\ 0 & A_{OTO} \end{bmatrix}, B_v = \begin{bmatrix} B_{SCC} \\ B_{OTO} \end{bmatrix}, C_v = \begin{bmatrix} C_{SCC} & 0 \\ 0 & C_{OTO} \end{bmatrix}, D_v = \begin{bmatrix} D_{SCC} \\ D_{OTO} \end{bmatrix}$$

We can define the vestibular state error  $x_e = x_s - x_a$  and the pilot sensation error  $e$ , resulting in:

$$\begin{aligned} \dot{x}_e &= A_v x_e + B_v u_s - B_v u_A \\ e &= C_v x_e + D_v u_s - D_v u_A \end{aligned} \quad (3.13)$$

In order to constrain the simulator motion, additional terms are included in the state equations:

$$\dot{x}_d = A_d x_d + B_d u_s \quad (3.14)$$

$$x_d = \left[ \iiint a_x dt^3 \quad \iint a_x dt^2 \quad \int a_x dt \quad \theta \right]^T \text{ and}$$

and is related to the simulator input  $u_s$  by:

$$A_d = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad B_d = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$$

The aircraft input  $u_A$  consists of filtered white noise and can be expressed as:

$$\begin{aligned} \dot{x}_n &= A_n x_n + B_n w \\ u_A &= x_n \end{aligned} \quad (3.15)$$

$$A_n = \begin{bmatrix} -\gamma_1 & 0 \\ 0 & -\gamma_2 \end{bmatrix}, \quad B_n = \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix}$$

$\gamma_1$  and  $\gamma_2$  are the first order filter break frequencies for each degree-of-freedom.

The state equations given in Equations (3.13), (3.14) and (3.15) can be combined to form the desired system equation:

$$\begin{aligned} \dot{x} &= Ax + B_n w \\ y &= [e \quad x_d]^T = Cx + Du_s \end{aligned} \quad (3.16)$$

where  $y$  is the desired output and  $x = [x_e \quad x_d \quad x_n]^T$  represents the combined states.

The combined system matrices A, B, C, D and H are then given by:

$$A = \begin{bmatrix} A_v & 0 & -B_v \\ 0 & A_d & 0 \\ 0 & 0 & A_n \end{bmatrix}, B = \begin{bmatrix} B_v \\ B_d \\ 0 \end{bmatrix}, H = \begin{bmatrix} 0 \\ 0 \\ B_n \end{bmatrix}, C = \begin{bmatrix} C_v & 0 & -D_v \\ 0 & I & 0 \end{bmatrix},$$

$$H = \begin{bmatrix} D_v \\ 0 \end{bmatrix}$$

A cost function  $J$  is then defined as:

$$J = E \left\{ \int_{t_0}^{t_1} (e^T Q e + x_d^T R_d x_s + u_s^T R u_s) dt \right\} \quad (3.17)$$

where  $E\{\}$  represents the mathematical mean of statistical variable,  $Q$  and definite matrices and  $R$  is a positive definite matrix.

The system equation and cost function can be transformed to the standard optimal control form by the following equations:

$$\dot{x} = A'x + Bu' + Hw$$

$$J' = E \left\{ \int_{t_0}^{t_1} (x^T R_1' x + u'^T R_2 u') dt \right\} \quad (3.18)$$

where

$$A = A - BR_2^{-1}R_{12}^T, u' = u_s + R_2^{-1}R_{12}^T x, R_1' = R_1 - R_{12}R_2^{-1}R_{12}^T,$$

$$R_1 = C^T G C, R_{12} = C^T G D, R_2 = R + D^T G D, G = \text{diag}[Q, R_d]$$

The cost function of Equation (3.18) is minimized when:

$$u' = R_2^{-1} B^T P x \quad (3.19)$$

where  $P$  is the solution of the following algebraic Riccati equation:

$$R_1' - PBR_2^{-1}B^T P + A'^T P + PA' = 0 \quad (3.20)$$

Substituting Equation (3.19) into Equation (3.18) and solving for  $u_s$ :

$$u_s = -[R_2^{-1}(B^T P + R_{12}^T)]x \quad (3.21)$$

and defining a matrix  $K$ , where  $u_s = Kx$  results in  $K = R_2^{-1}(B^T P + R_{12}^T)$ .

$K$  can be partitioned corresponding to the partition of  $x$  in Equation (3.16):

$$u_s = -\begin{bmatrix} K_1 & K_2 & K_3 \end{bmatrix} \begin{bmatrix} x_e \\ x_d \\ x_n \end{bmatrix} \quad (3.22)$$

Since  $x_n = u_A$ , the states corresponding to the  $x_n$  are removed:

$$\begin{bmatrix} \dot{x}_e \\ \dot{x}_d \end{bmatrix} = \begin{bmatrix} A_v & 0 & -B_v \\ 0 & A_d & 0 \end{bmatrix} \begin{bmatrix} x_e \\ x_d \\ u_A \end{bmatrix} + \begin{bmatrix} B_v \\ B_d \end{bmatrix} u_s \quad (3.23)$$

and substituting Equation (3.22) into Equation (3.23) results in

$$\begin{bmatrix} \dot{x}_e \\ \dot{x}_d \end{bmatrix} = \begin{bmatrix} A_v - B_v K_1 & -B_v K_2 \\ -B_d K_1 & A_d - B_d K_2 \end{bmatrix} \begin{bmatrix} x_e \\ x_d \end{bmatrix} + \begin{bmatrix} B_v(I + K_3) \\ -B_d K_3 \end{bmatrix} u_A \quad (3.24)$$

Then, the following equations are obtained in the Laplace domain:

$$u_s(s) = W(s) \times u_A(s) \quad (3.25)$$

where

$$W(s) = \begin{bmatrix} K_1 & K_2 \end{bmatrix} \begin{bmatrix} sI - A_v + B_v K_1 & B_v K_2 \\ B_d K_1 & sI - A_d + B_d K_2 \end{bmatrix}^{-1} \begin{bmatrix} B_v(I + K_3) \\ B_d K_3 \end{bmatrix} - K_3$$

$W(s)$  is a matrix of the optimized open-loop transfer functions linking the simulator inputs  $u_s$  to the aircraft inputs  $u_A$ .

Continuous-time Bode diagrams and step input responses of the obtained filters are as follows:

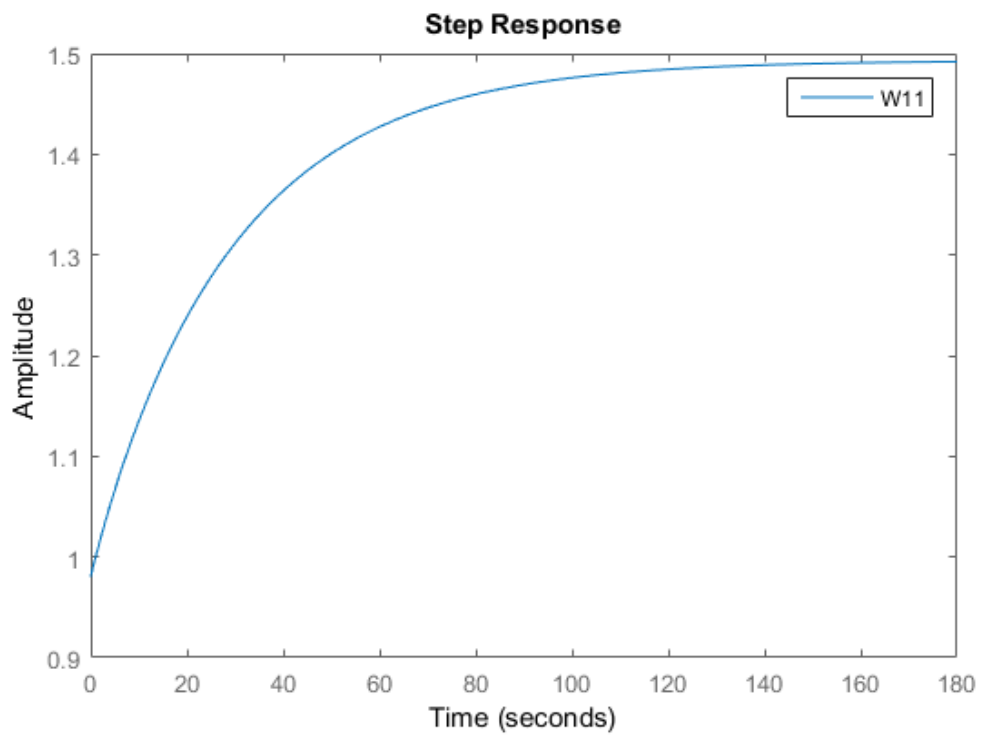


Figure 3.10 Step response of W11 filter

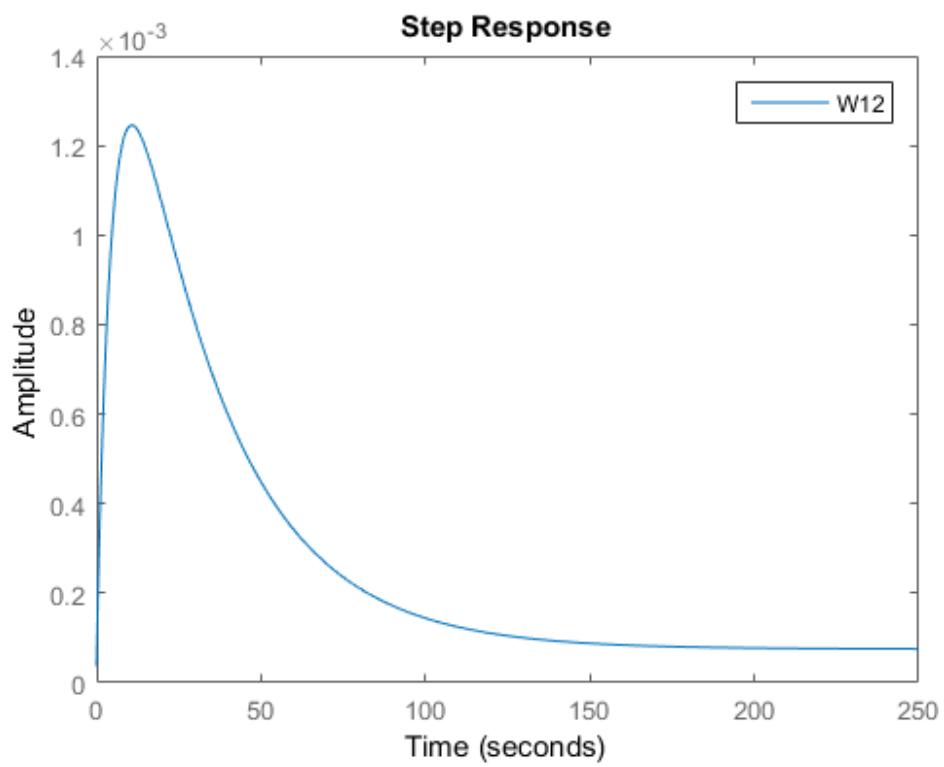


Figure 3.11 Step response of W12 filter

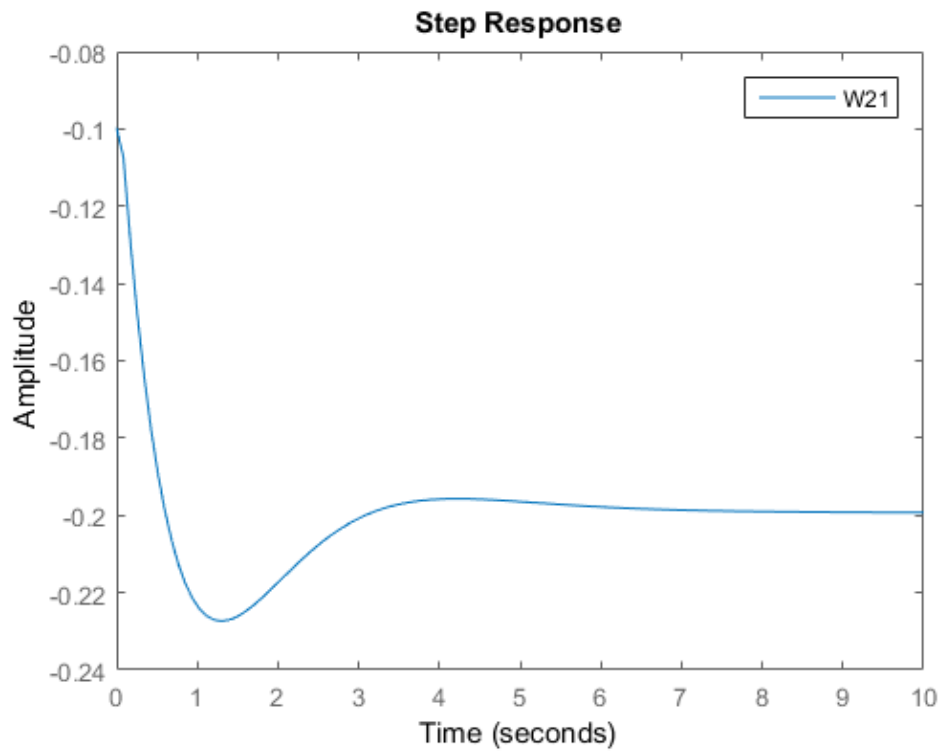


Figure 3.12 Step response of W21 filter

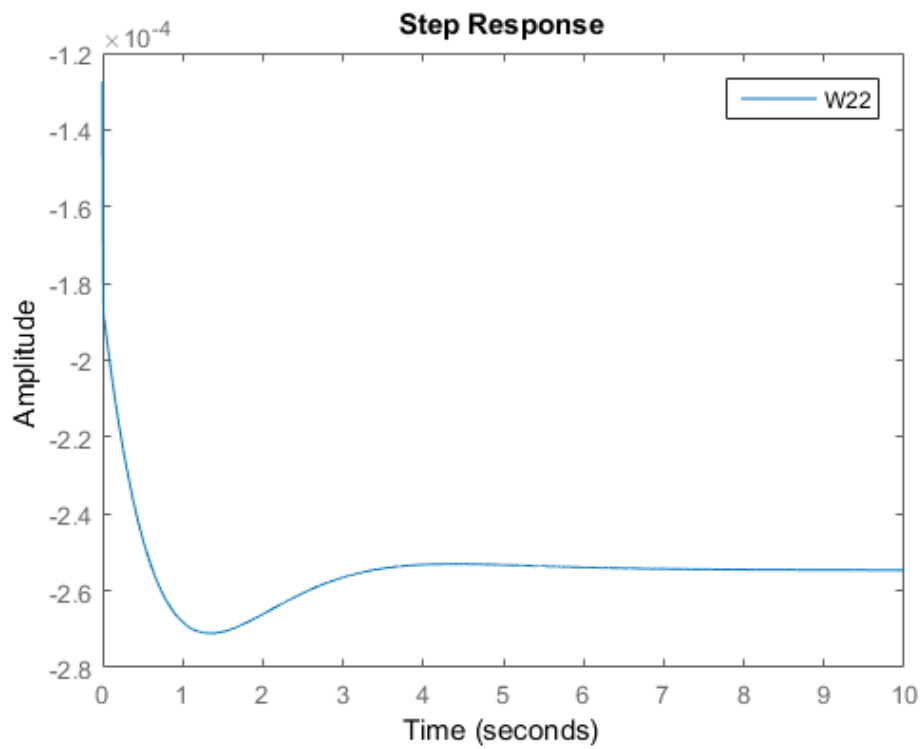


Figure 3.13 Step response of W22 filter

Following bode plots were obtained by using first-order-hold (foh) method and sampling time of 0.001.

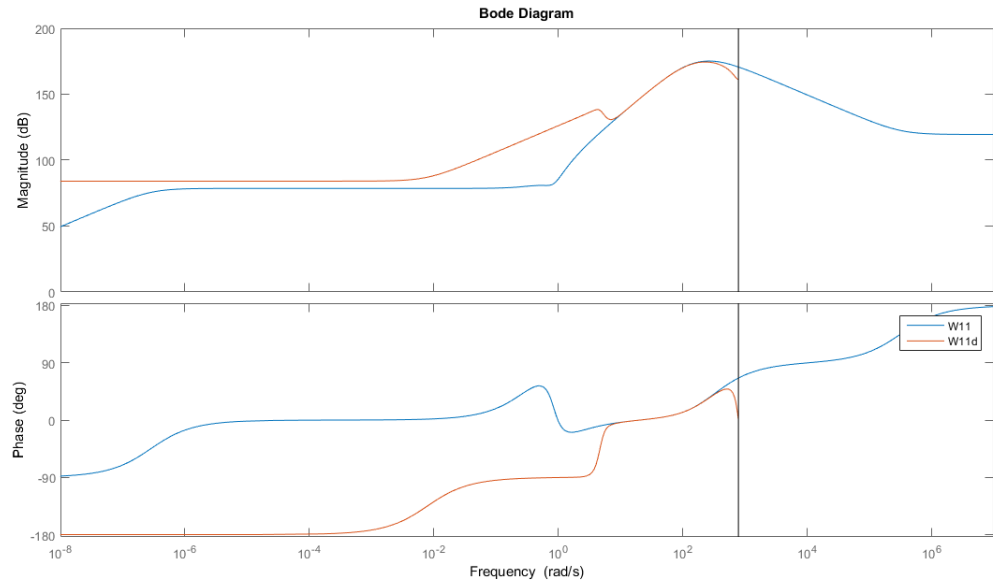


Figure 3.14 Bode diagrams for W11 and W11d filters (d  $\rightarrow$  discrete)

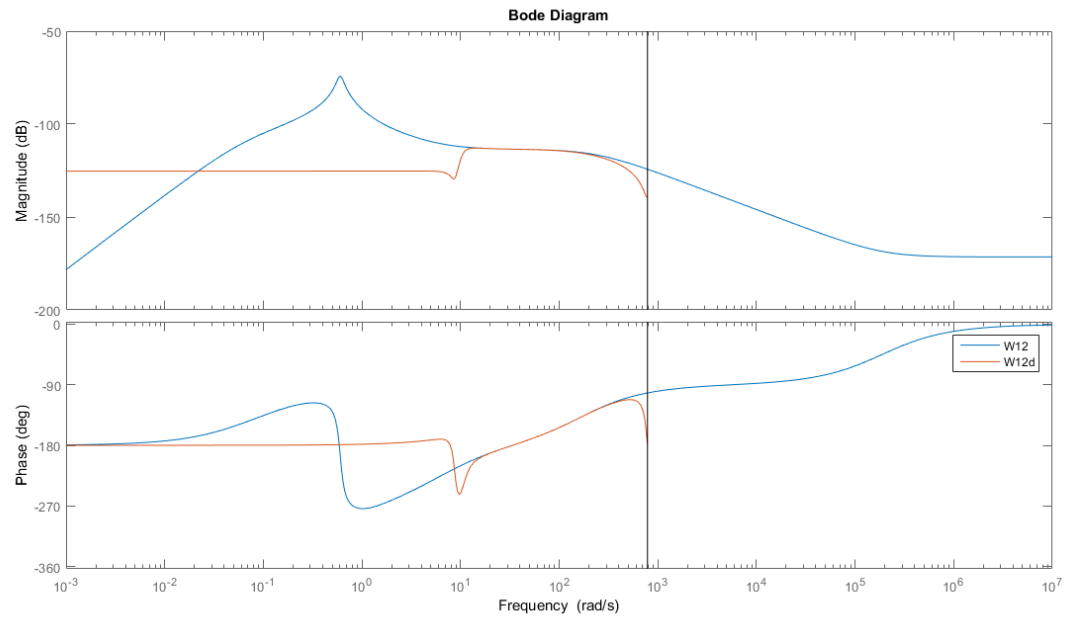


Figure 3.15 Bode diagrams for W12 and W12d filters



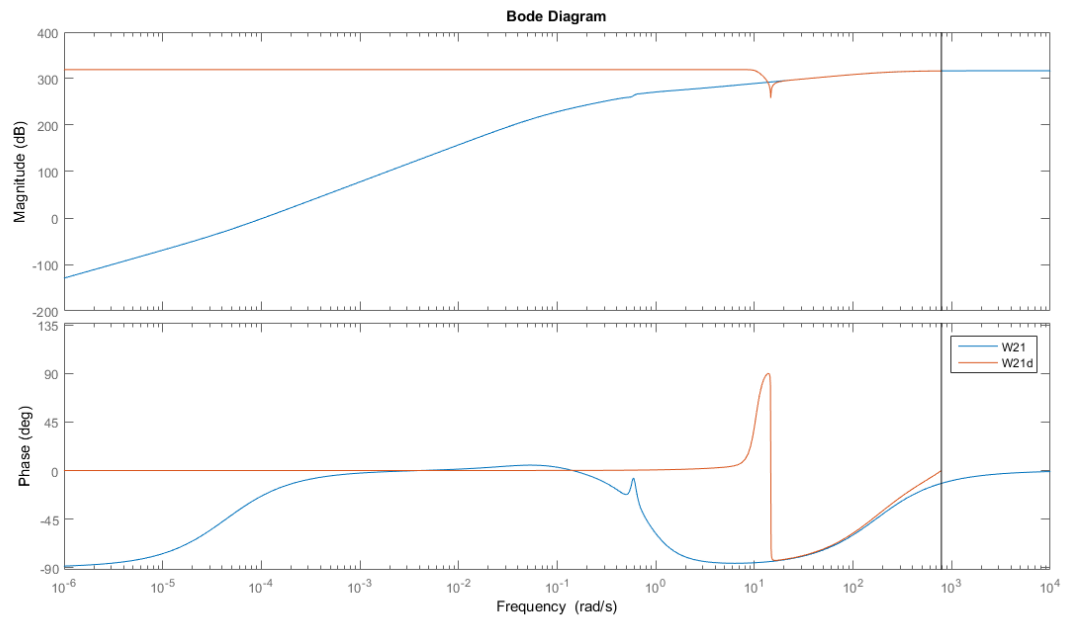


Figure 3.16 Bode diagrams for W21 and W21d filters

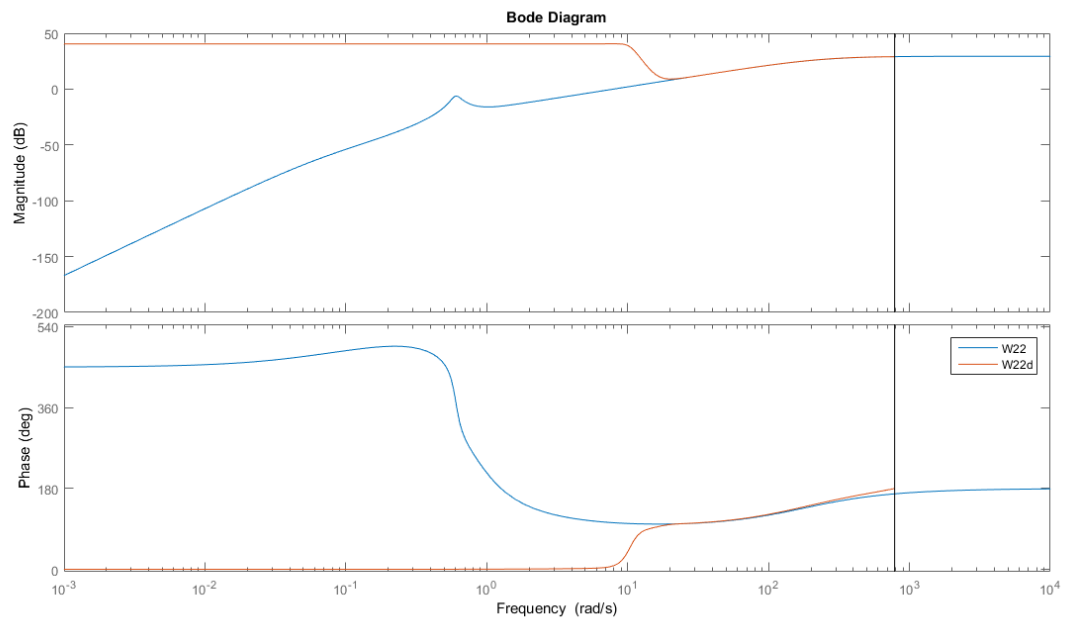


Figure 3.17 Bode diagrams for W22 and W22d filters

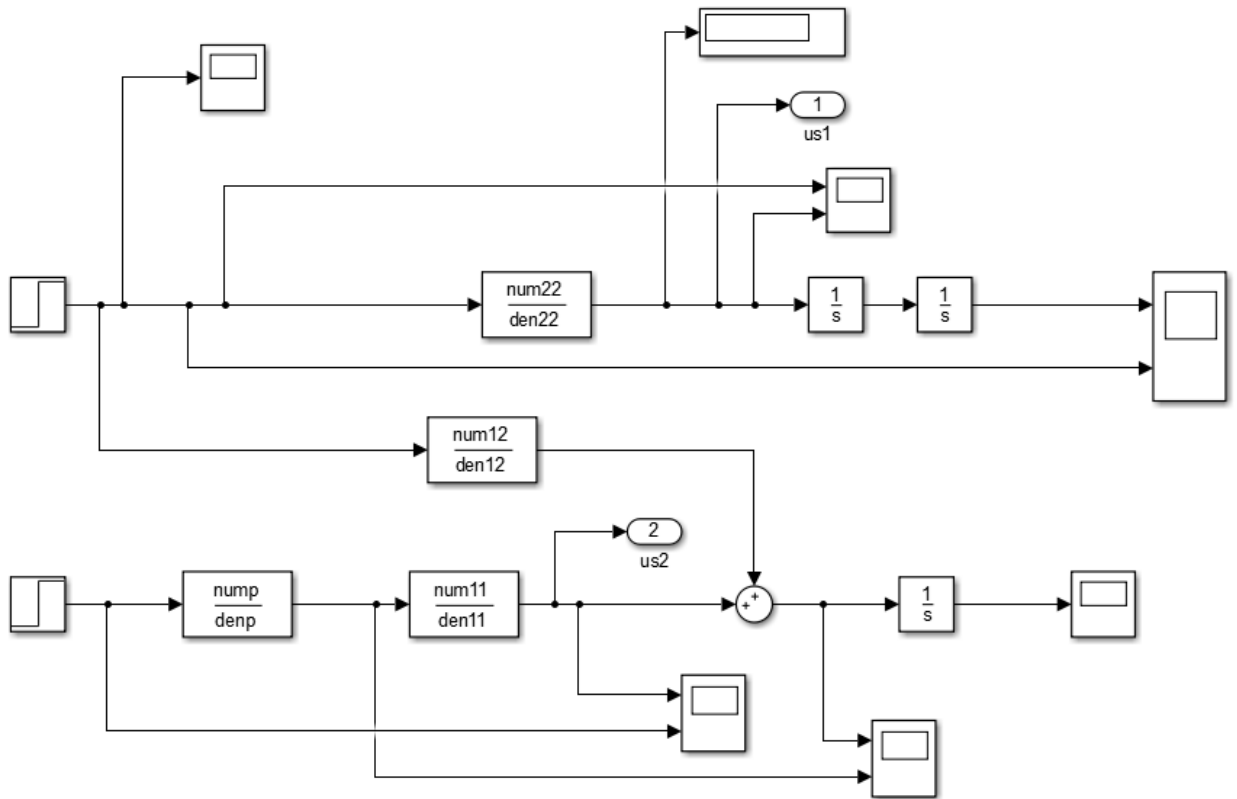


Figure 3.18 Continuous-time optimal motion cueing algorithm Simulink model

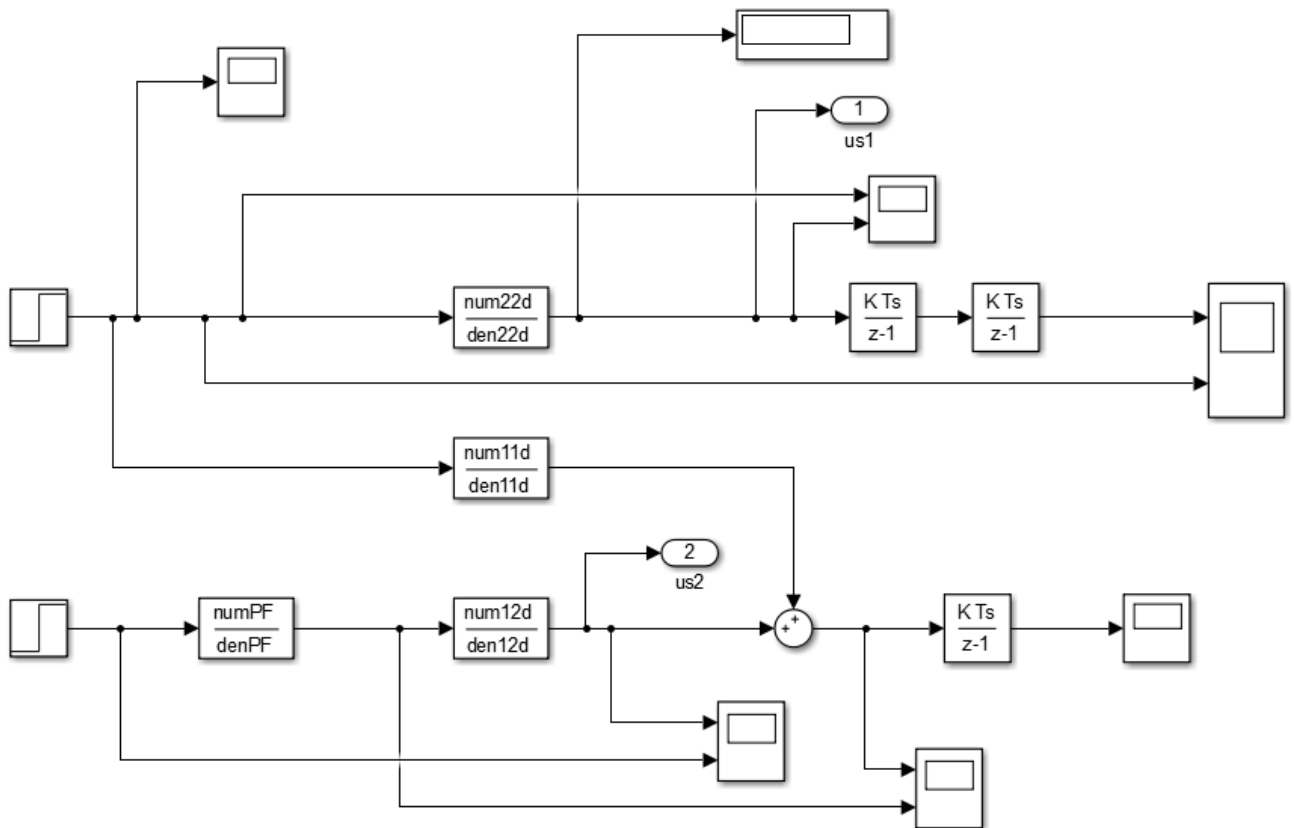


Figure 3.19 Discrete-time optimal motion cueing algorithm Simulink model

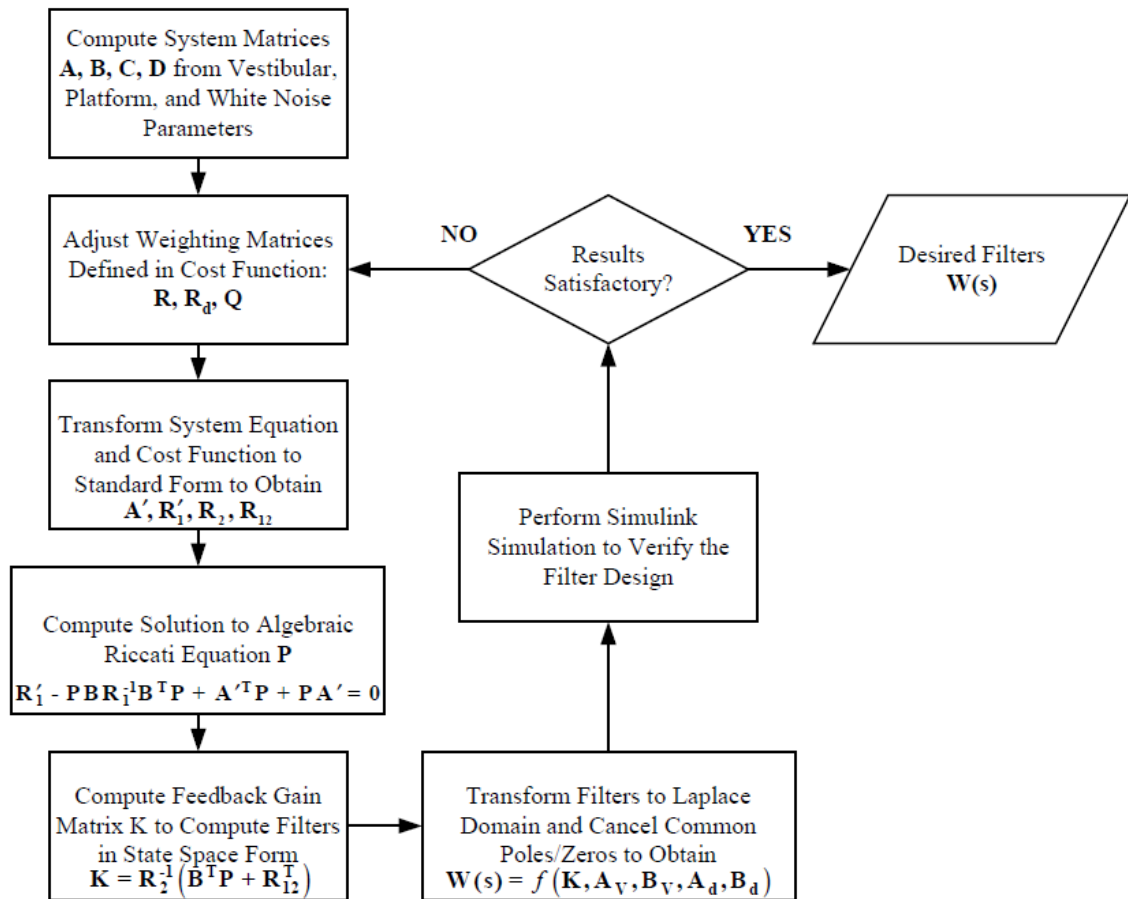


Figure 3.20 Optimal washout filter solution procedure

### 3.4 System Integration

#### 3.4.1 Integration of 6 DOF Low Payload Motion Platform

Flight simulator system platform construction was completed with Amega Inc.'s collaboration. Elements that platform has are as follows:

- Platform motion system
- Controller unit
- IO data processing cards unit
- Mechanical and software switching elements
- Simulation control computer and motion control computer
- Command control
- Force sensor and IMU sensor

General representation of the platform is as shown in Figure 3.21.



Figure 3.21 6 DOF low payload flight simulator system

#### 3.4.1.1 Platform Motion System

Platform is driven by high capacity 6 AC servo motors with ball screws. Motors and pistons are as shown in Figure 3.22. In order to provide connection between platform and both ground and top plate 2 shaft clutches for each piston were used. Stroke length for each piston is 570 mm.

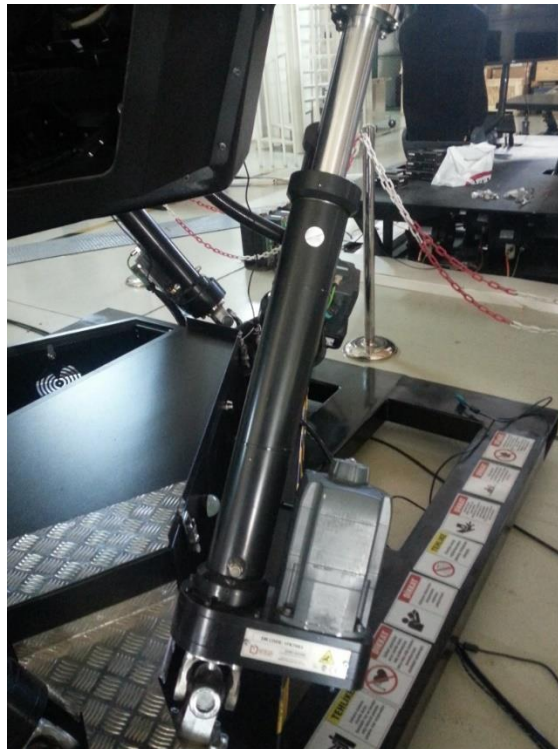


Figure 3.22 Motor and piston

### 3.4.1.2 Motion Controller Units

Platform motion commands are controlled by controller with drivers. Representation of controller group in the platform is as shown in Figure 3.23.



Figure 3.23 Controller group

In terms of their task, drivers and controllers are placed between motion control computer and drive system. During simulation motion information that is sent by the pilot with command control is transmitted to the aircraft model via controllers. Each driver drives 2 different leg of the platform. In order to prevent any possibility of conflict different IP addresses are defined as shown in Table 3.4.

Table 3.4 Defined IP addresses

	IP Addresses
Simulation Control Computer	127.0.0.1
Controller	192.168.214.1
Motion Control Computer	192.168.214.20

### 3.4.1.3 Mechanical and Software Switching Elements

During motion in order to ensure that pistons are not taking off from their place 2 different switching methods are used as mechanical and software switching elements. Mechanical switching method is provided by switching elements that are placed on two endpoints of pistons as shown in Figure 3.24. Software switching elements are integrated into both Simulink model as saturation blocks and controller.

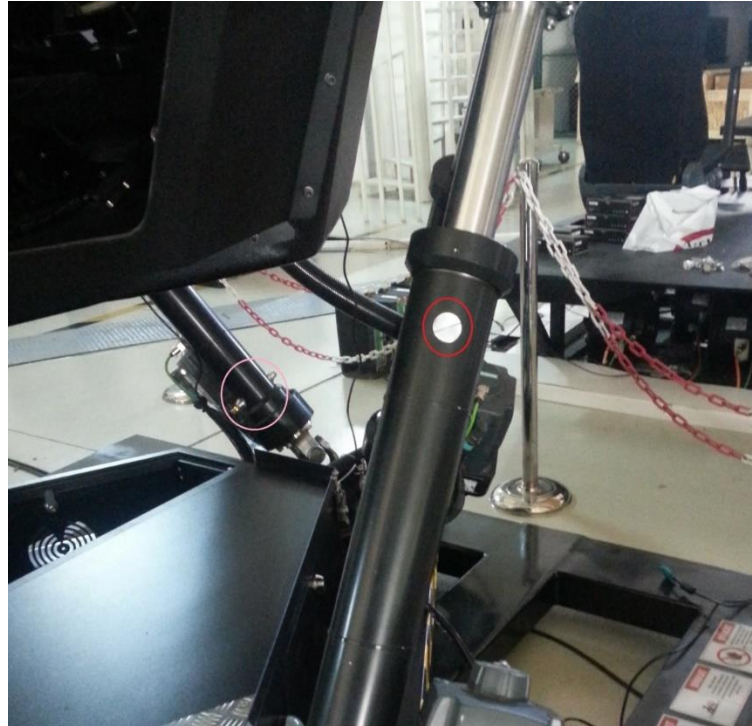


Figure 3.24 Mechanical switching elements

#### **3.4.1.4 Computer Systems**

2 computer are used in the platform. Motion control computer and simulation control computer.

#### **3.4.1.5 Command Control, Pedal and Screen Systems**

In order to provide a realistic flight, command control elements and pedal are as shown in Figure 3.25. 3 LCD TVs were used and combined with an angle of  $120^\circ$  to obtain image.





Figure 3.25 Command controls and pedals

Specification of command control unit:

- 3D magnetic sensor
- 19 active buttons with 16 bit resolution in joystick
- 17 active buttons with 14 bit resolution in throttle
- USB connection

Specification of pedals:

- Pedals have adjustable system shows resistance effect
- Pedals have ability to move translationally and obtain yaw motion
- For braking, pushing onto pedals in the vertical direction is necessary.

#### 3.4.1.6 Force and Acceleration (IMU) Sensors

In order to measure any acceleration value of the platform physically during flight, IMU sensor is integrated on the platform. Firstly IMU sensor was mounted to the head alignment in order to work on human perception as shown in Figure 3.26. Coordinates of central point of top plate and acceleration levels of human motion perception are measured by using the transfer functions of vestibular system via IMU sensor.



Figure 3.26 Acceleration (IMU) sensor

Force sensor is placed under the pilot seat and external forces pilot exposed to are measured via this sensor. Sensor is shown in Figure 3.27.



Figure 3.27 Force sensor

#### **3.4.1.7 IO Data Processing Cards Compartment**

Siemens IO data processing cards were used to provide data exchange between force sensor, IMU sensor and EEG sensor with the system. Cards were mounted into a till placed over the back plate of screens and connection has been made ready. Cards are as shown in Figure 3.28.



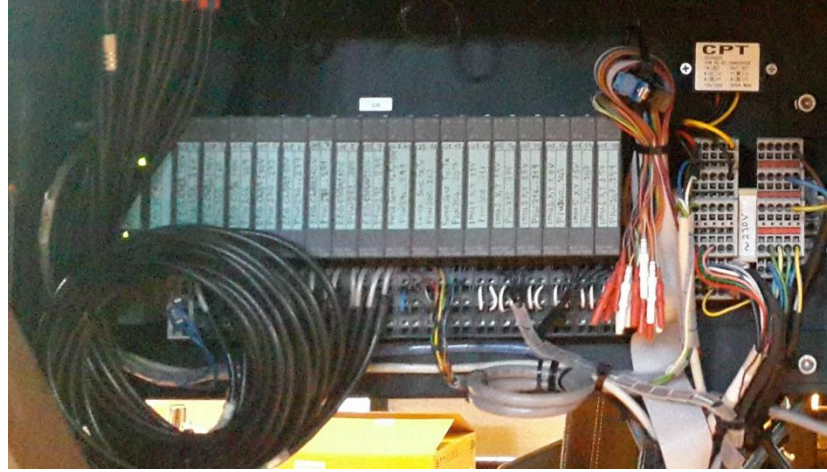


Figure 3.28 IO data processing cards

### 3.4.2 Frequency Responses of the Platform

Frequency responses of the platform were measured in decibels by applying inputs at certain frequencies to the platform. During measurement, replacement of  $\pm 3^\circ \sin(\omega t)$  (mm) was applied to the system through translational directions and angular rotations by supplying the system with frequency change of 0.1 – 3.5 Hz interval. Maximum amplitude values obtained as a result of these measurements are as shown in Table 3.5. Results of system responses in decibel are shown in Table 3.6.

Table 3.5 Frequency responses of the platform

	Frequencies – $\omega$ (Hz)										
	0,1	0,2	0,3	0,4	0,5	1	1,5	2	2,5	3	3,5
<b>X (mm)</b>	19,99	19,99	19,99	19,99	19,99	20	20,02	20,08	20,02	12,18	9,75
<b>Y (mm)</b>	19,98	19,99	19,99	19,99	19,99	20	20,02	20,08	20,14	19,75	14,52
<b>Z (mm)</b>	19,99	19,99	19,99	19,99	19,98	19,99	20,07	20,07	17,26	11,45	8,62
<b>Pitch (deg)</b>	2,998	2,998	2,998	2,998	2,998	2,998	2,933	1,329	1,707	0,96	0,96
<b>Roll (deg)</b>	2,998	2,998	2,998	2,998	2,998	2,999	2,982	1,823	1,103	0,751	0,896
<b>Yaw (deg)</b>	2,998	2,998	2,998	2,998	2,998	2,999	3,003	2,74	1,783	1,18	1,553

Table 3.6 Frequency responses in decibel

	(DB)										
	0,1	0,2	0,3	0,4	0,5	1	1,5	2	2,5	3	3,5
<b>x</b>	- 0,0043	-0,0043	-0,0043	-0,0043	- 0,0043	0	0,0087	0,0347	0,0087	- 4,3077	- 6,2405
<b>y</b>	- 0,0087	-0,0043	-0,0043	-0,0043	- 0,0043	0	0,0087	0,0347	0,0606	- 0,1093	- 2,7813
<b>z</b>	- 0,0043	-0,0043	-0,0043	-0,0043	- 0,0087	- 0,0043	0,0303	0,0303	- 1,2798	- 4,8445	- 7,3105
<b>Pitch</b>	- 0,0058	-0,0058	-0,0058	-0,0058	- 0,0058	- 0,0058	- 0,1962	- 7,0719	- 4,8978	-9,897	-9,897
<b>Roll</b>	- 0,0058	-0,0058	-0,0058	-0,0058	- 0,0058	- 0,0029	- 0,0523	- 4,3267	- 8,6909	-12,03	- 10,496
<b>Yaw</b>	- 0,0058	-0,0058	-0,0058	-0,0058	- 0,0058	- 0,0029	0,0087	- 0,7874	- 4,5194	- 8,1048	-5,719

According to JAR standards results are convenient.

## CHAPTER 4

---

### RESULTS AND DISCUSSION

Previously, we implemented classical washout filter on a small scale desktop platform. After a significant optimization was applied, a full scale flight simulator mechanical structure has been constructed and classical washout filter was implemented. The system is on a 6x6 parallel manipulator and is driven by AC motors. The classical washout filters were implemented on Matlab – Simulink environment with Real-Time Windows Target. This algorithm has first and second order high pass filters for high frequency components and first order low pass filters for low frequency components of aircraft flight variables. Also, all software implementations are established on Matlab. Flight data are obtained from virtual flight software with 5 Hz sampling rate. In this section, validations of motion cueing algorithm for all motions, results that obtained from Boeing 777-300ER and Sikorsky flights are going to be shown.

Results of Boeing 777-300ER are as follows:



Figure 4.1 Platform motion during takeoff

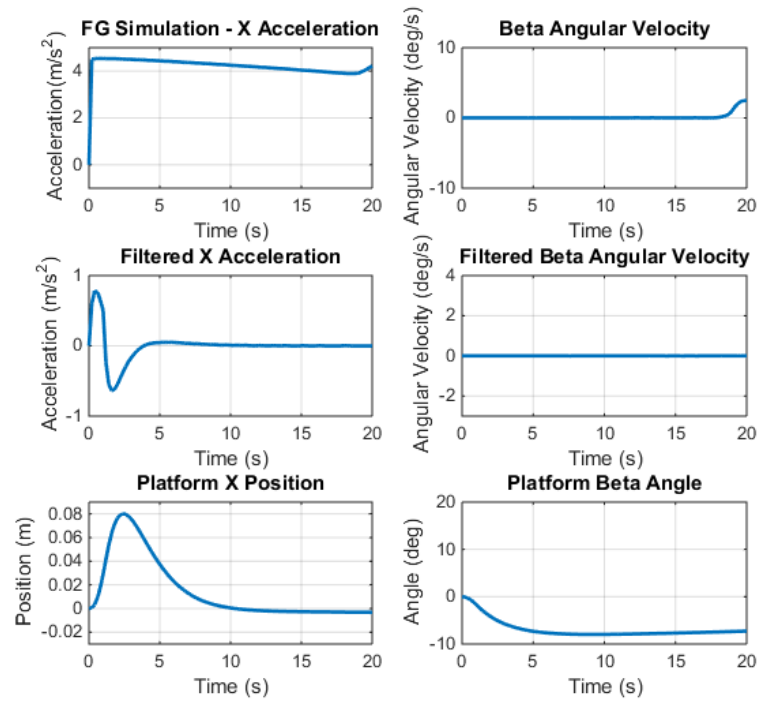


Figure 4.2 Acceleration through x- axis, filtered x- acceleration and platform x- position



Figure 4.3 Roll Motion about counter clockwise direction



Figure 4.4 Roll Motion about clockwise direction

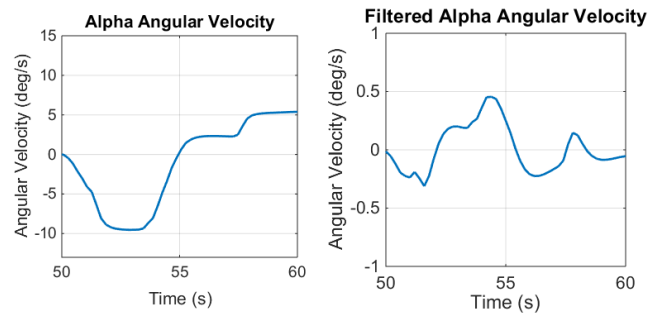


Figure 4.5 Incoming and filtered Roll velocity of the aircraft

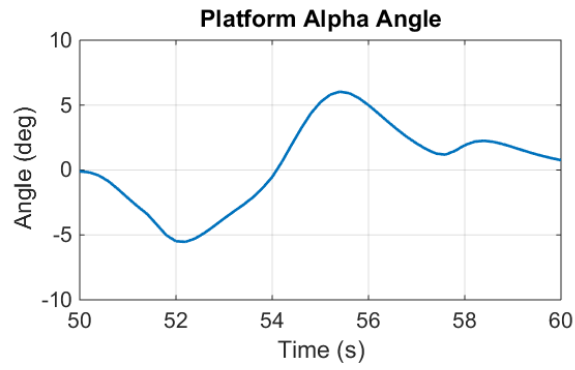


Figure 4.6 Performance of the platform for roll motion



Figure 4.7 Pitch motion of the aircraft through negative pitch direction



Figure 4.8 Pitch motion of the aircraft through positive pitch direction

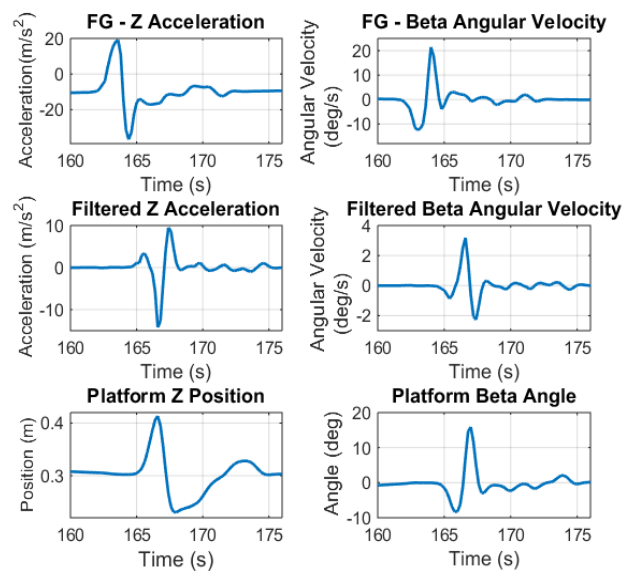


Figure 4.9 Pitch velocity, filtered pitch velocity and platform pitch angle

Results of Sikorsky type helicopter on the low payload full scale flight simulator system structure are as follows:

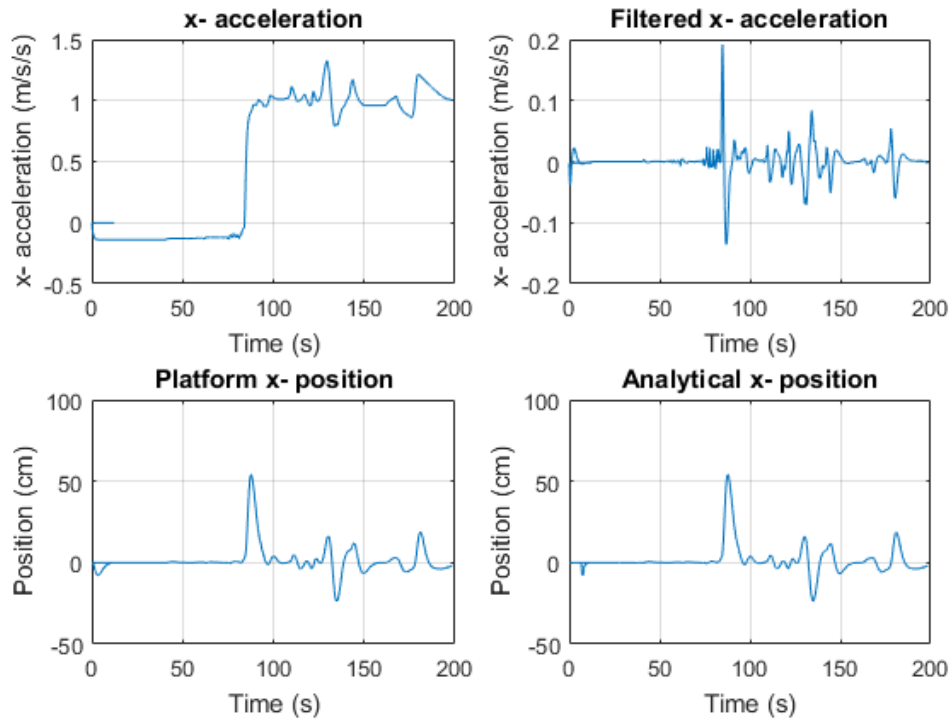


Figure 4.10 Helicopter motion through x- axis and results

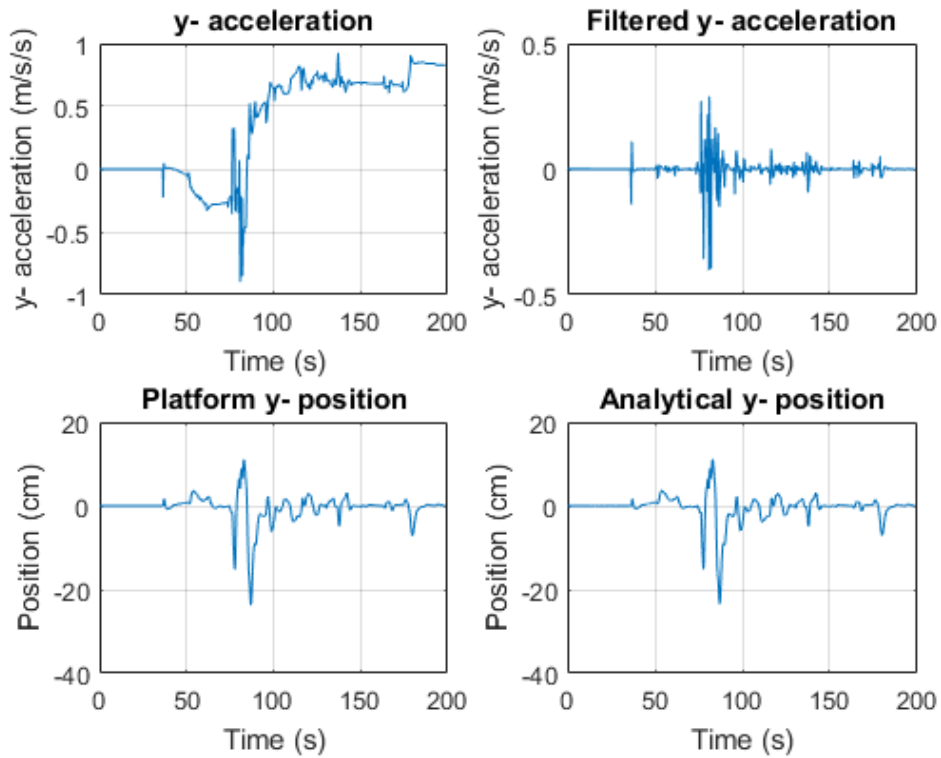


Figure 4.11 Helicopter motion through y- axis and results

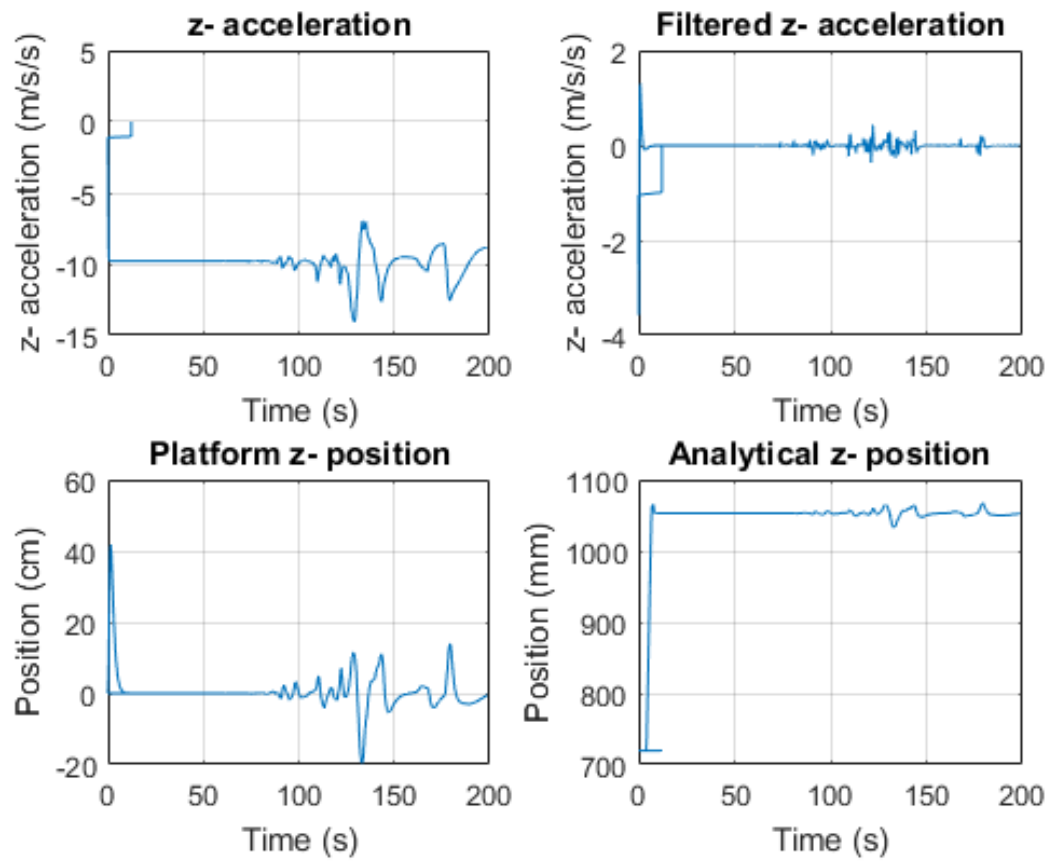
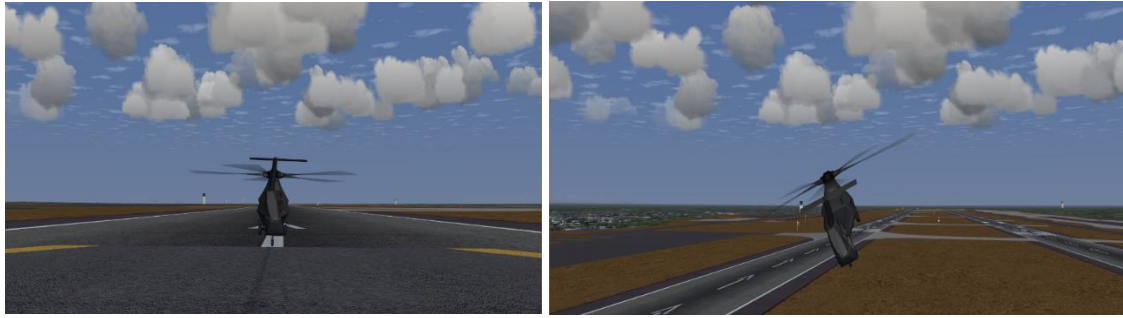


Figure 4.12 Helicopter motion through z- axis and results

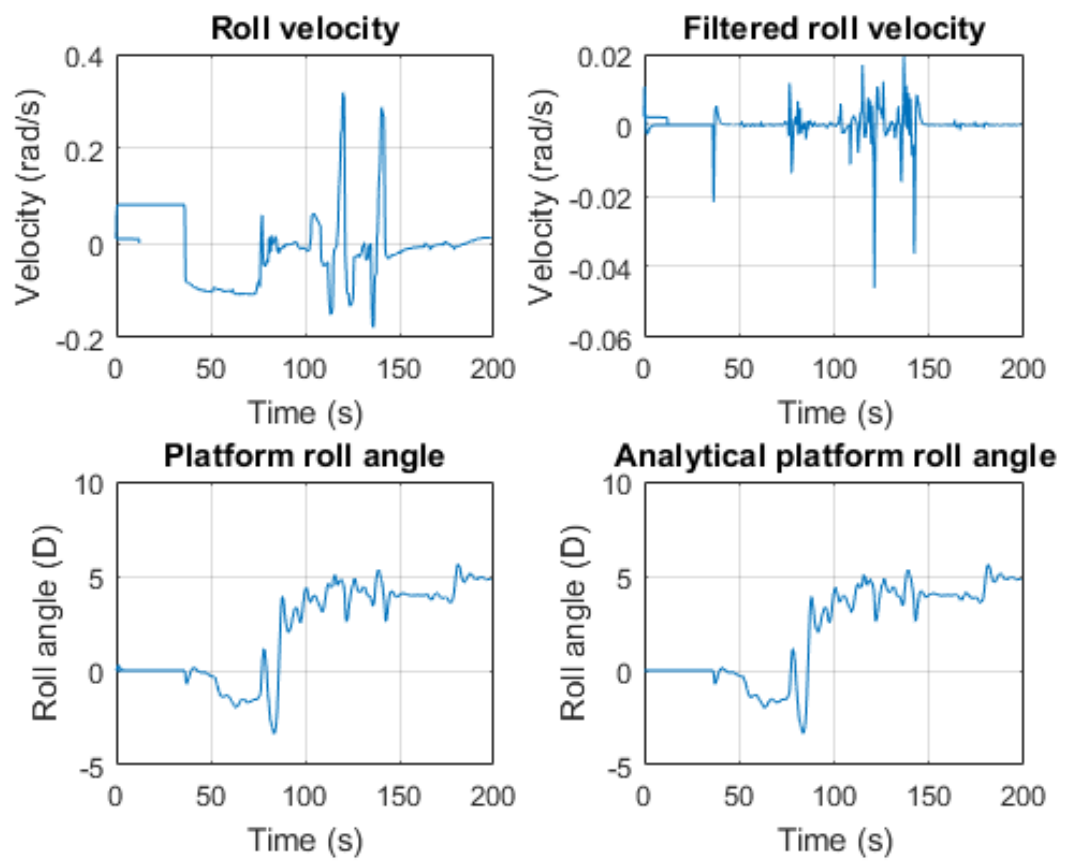
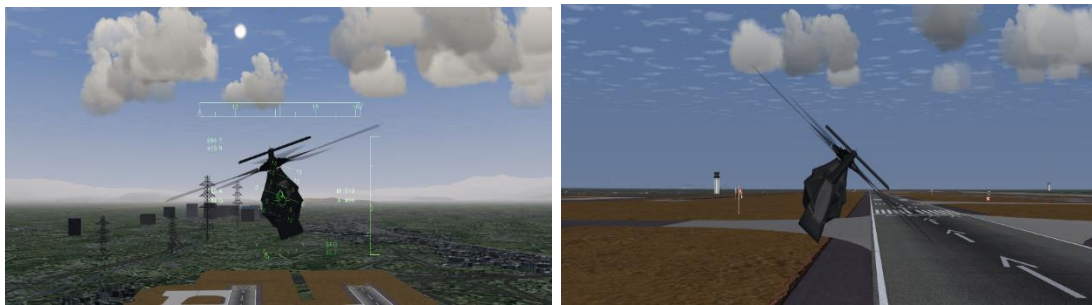


Figure 4.13 Results for roll motion of the helicopter



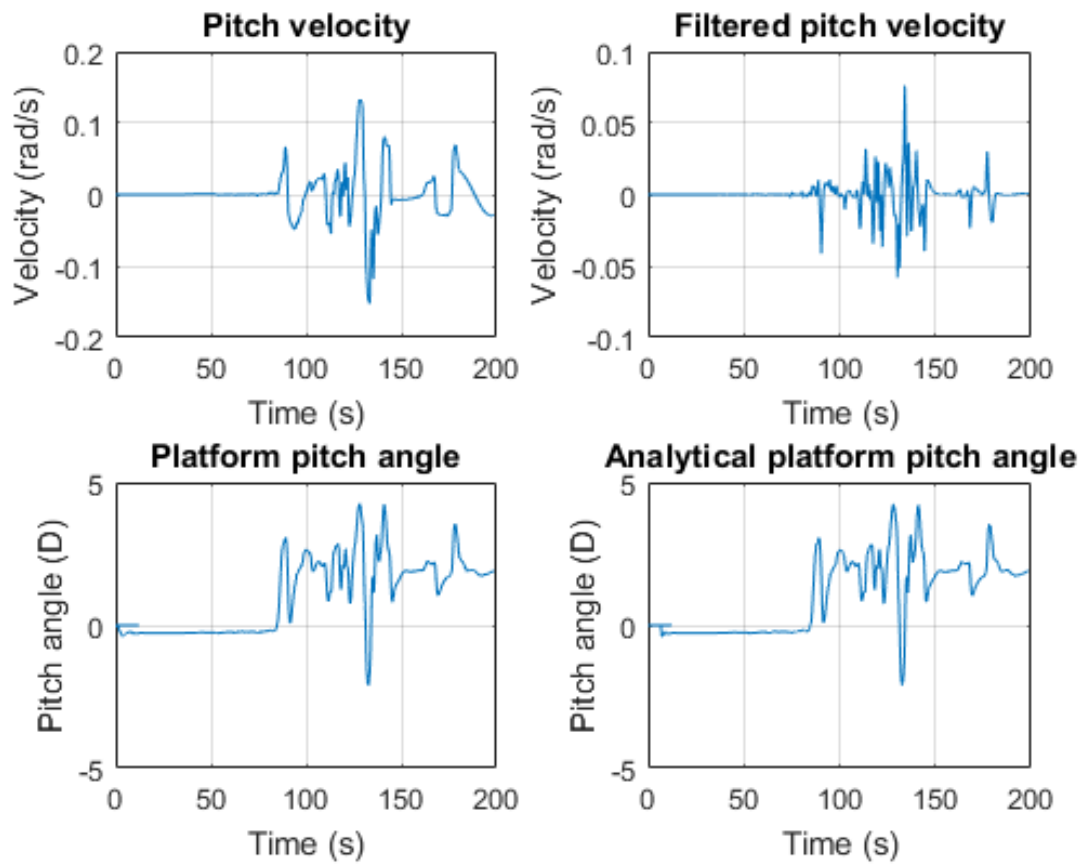
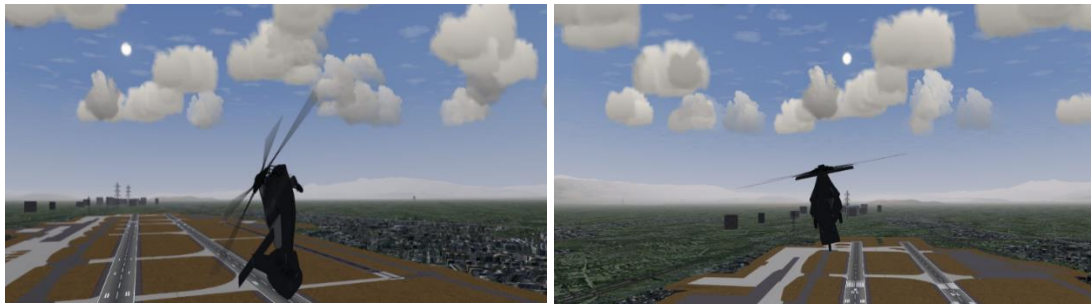


Figure 4.14 Results for pitch motion of the helicopter

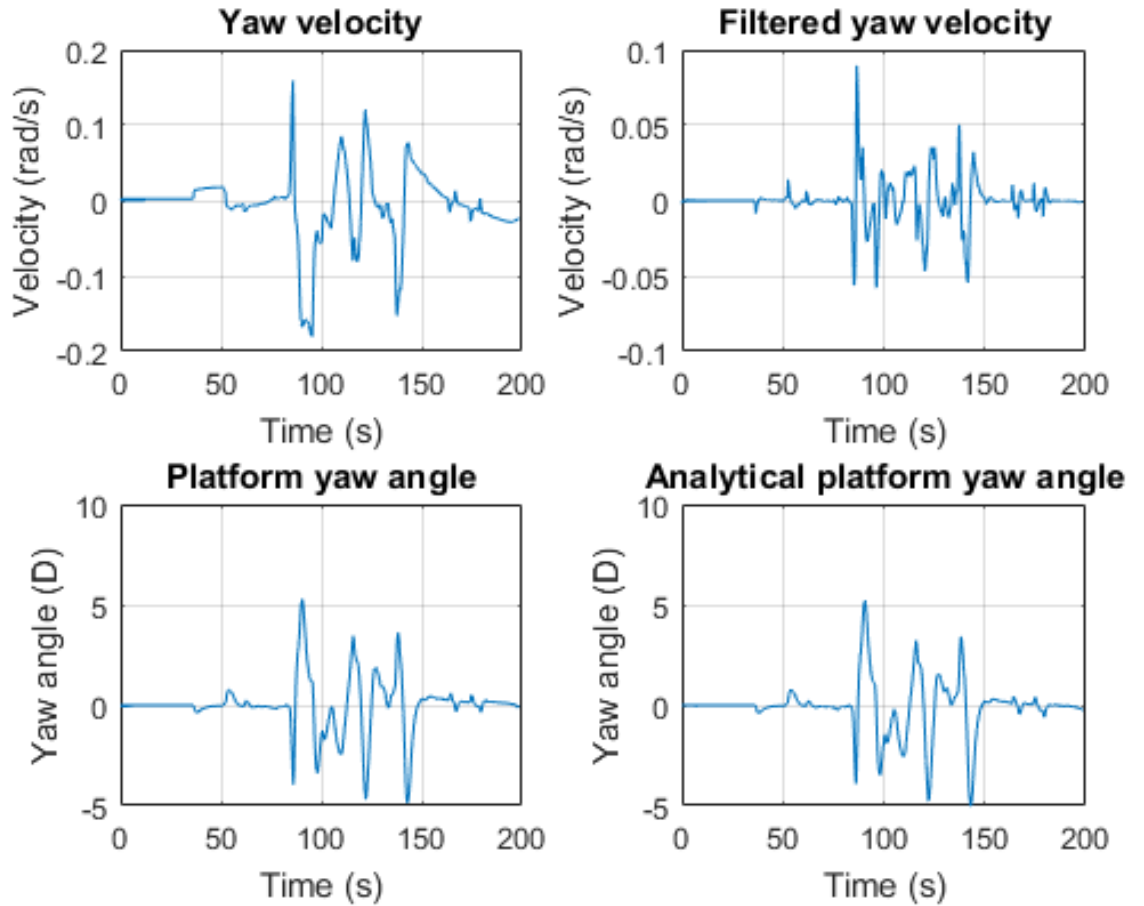
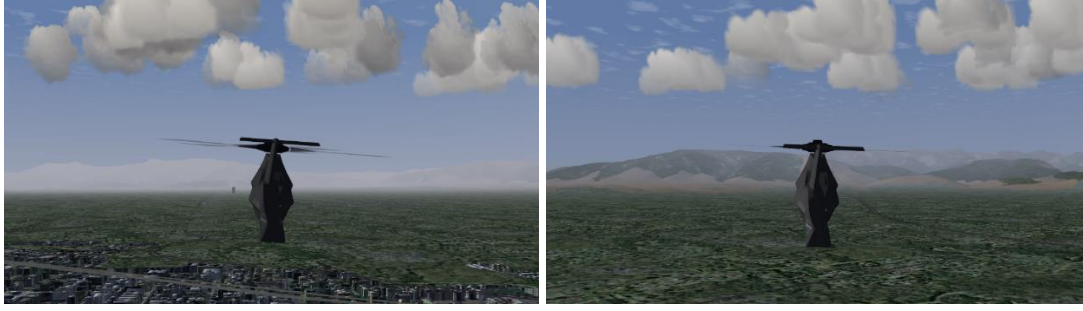


Figure 4.15 Results for yaw motion of the helicopter

Figure 4.1 shows that during takeoff, aircraft performed a translational acceleration of  $4 \text{ m/s}^2$  through x-axis. Thus the platform moved for the length of approximately 0.08 meter through +x direction. It also performed a rotational motion of  $8^\circ$  about y-axis because of the tilt effect. Figure 4.2 represents related results.

Figure 4.3 shows the roll motion of the aircraft about x- axis and through counter-clockwise direction at the angle of  $5^\circ$  and Figure 4.4 shows the roll motion of the aircraft about x- axis and through clockwise direction at the alpha angle of  $7^\circ$ .

Incoming roll velocity data from FlightGear and filtered roll velocity can be seen in Figure 4.5. Platform alpha angle obtained by integrating filtered roll velocity twice can be seen in Figure 4.6. Washout condition can also be seen in these figures.

Figure 4.7 and Figure 4.8 shows the pitch motion of the aircraft through negative and positive directions at the angles of approximately  $13^\circ$  through positive direction and  $9^\circ$  through negative direction as can be seen in Figure 4.9.

Figure 4.10 shows the incoming translational acceleration data, filtered translational acceleration, related position of the platform and analytical position of the platform of Sikorsky type helicopter through x- axis.

Similarly Figure 4.11 shows the incoming translational acceleration data, filtered translational acceleration, related position of the platform and analytical position of the platform of Sikorsky type helicopter through y- axis.

Helicopter motion during takeoff and hover is as shown in Figure 4.12. Related incoming datas, filtered acceleration and positions can be seen in Figure.

The results of the helicopter for roll motion are as shown in Figure 4.13. and similarly Figure 4.14 and Figure 4.15 represent the pitch and yaw motion of the helicopter about y- and z- axes respectively.

## REFERENCES

- 
- [1] N. J. Garrett and M. C. Best, (2012). "Evaluation of a New Body-Sideslip-Based Driving Simulator Motion Cueing Algorithm", *Journal of Automobile Engineering*, 1433-1445.
  - [2] H. Elloumi, M. Bordier and N. Maïzi (2005). "Optimal Control Scheme For A Driving Simulator", In *Proceedings of the Second International Conference on Informatics in Control, Automation and Robotics*, 14-17 September 2005, Barcelone, 40-47.
  - [3] Z. Lei and J. Hongzhou, (2007). "PC Based High Quality and Low Cost Flight Simulator", *International Conference on Automation and Logistics*, Jinan, China, 1017-1022.
  - [4] A. Hamish and J. Jamson, (2010). *Motion Cueing in Driving Simulators for Research Applications* The University of Leeds Institute for Transport Studies, PhD Thesis, England.
  - [5] G. L. Zacharias, (1978). *Motion Cue Models for Pilot-Vehicle Analysis*, AMRL – TR – 78 – 2, Wright – Patterson Air Force Base, Ohio.
  - [6] R. J. Telban, W. Wu and F.M. Cardullo, (2000). "Motion Cueing Algorithm Development: Initial Investigation And Redesign of the Algorithms", *National Aeronautics and Space Administration, Langley Research Center*; CR – 2000 – 209863.
  - [7] A. Capustiac, B. Hesse, D. Schramm and D. Banabic, (2011). "A Human Centered Control Strategy for a Driving Simulator", *International Journal of Mechanical & Mechatronics Engineering IJMME-IJENS*, 11(1):37-44.
  - [8] J.A. Houck, R. J. Telban and F.M. Cardullo, (2005). "Motion Cueing Algorithm Development: Human – Centered Linear and Nonlinear Approaches", *Technical Report. CR-2005-213747*, Nasa.
  - [9] Chang L.-H., Liao, C.-S., and Chieng, W.-H., (2009). "Optimal Motion Cueing for 5 – DOF Motion Simulations via a 3 – DOF Motion Simulator", *Control Engineering Practice*, 17: 170 – 184.
  - [10] F. Colombet, D. Paillot, F. Merienne and A. Kemeny, (2011). "Visual scale factor for speed perception", *Journal of Computing and Information Science in Engineering, American Society of Mechanical Engineers (ASME)*, 11(4):1-6.
  - [11] B. D. Correia Augusto and R. J. Leal Loureiro, (2009). *Motion Cueing in the Chalmers Driving Simulator: A Model Predictive Control Approach*, Master of Science Thesis, Chalmers University Of Technology, Department of Signals And Systems, Report No. EX057.

- [12] L. Reid and A. Nahon, (1990). "Simulator Motion-Drive Algorithms: A Designer's Perspective", The University of Toronto, 13(2):356-362.
- [13] M. Aminzadeh and A. Mahmoodi, (2012). "Optimal Motion Cueing Algorithm Using Motion System Kinematics", European Journal of Control 18(4):363 – 375.
- [14] L. D. Reid and M. A. Nahon, (1985). Flight simulation motion-base drive algorithms: Part 1 – Developing and Testing the Equations, UTIAS Report No. 296, CN ISSN 0082-5255, December.
- [15] R. Sivan, J. Ish-Shalom, and J.-K. Huang, (1982). "An Optimal Control Approach to the Design of Moving Flight Simulators", Systems, Man and Cybernetics, IEEE Transactions on Systems, 12:818-827.
- [16] R. A. Stela, C. Miklos and L. C. Lihaciu (2014). "Adaptive Control of Pantograph – Catenary Interactions Force", WSEAS Transactions on Systems, 13:708-719.
- [17] Y. Yang and S. Zheng, (2011). "Motion Drive Algorithm for Flight Simulator Based on the Stewart Platform Kinematics", School of Mechatronics Engineering, Harbin Institute of Technology, Harbin, 150080, Key Engineering Materials 460-461:642-647, China.
- [18] G. Reymond and A. Kemeny, (2000). "Motion Cueing in the Renault Driving Simulator", International Journal of Vehicle Mechanics and Mobility, 249–259.
- [19] H. Sehammer, M. Fischer and G. Palmkvist, (2010). "Motion Cueing for 3-6 and 8 Degrees of Freedom Motion Systems", Proceedings of the Driving Simulation – Conference Europe, Swedish National Road and Transport Research Institute, 9-10 September 2010, France, 121-134.
- [20] Z. Fang and G. Reymond, (2011). "Performance Identification and Compensation of Simulator Motion Cueing Delays", RENAULT, Technical Center for Simulation, TCR AVA 0 13, 1, 11, France.
- [21] M. Dagdelen, G. Reymond, A. Kemeny, M. Bordier, and N. Maïzi, (2009). "Model-based predictive motion cueing strategy for vehicle driving simulators", Control Engineering Practice, 17(9): 995-1003.
- [22] M.L.Saidi, A. Debbah, H.Arioui, M.S. Kermiche, and H.A. Abbassi, (2006). "Predictive control of motion platform in driving simulator", Asian journal of information technology, 5(2):133–138.
- [23] L.Nehaoua, H. Arioui, S. Espie and H. Mohellebi, (2006). "Motion Cueing Algorithms for small Driving Simulator", IEEE International Conference in Robotics and Automation (ICRA06), 28 September 2009, Orlando, 3189-3194.
- [24] C. Carlota and S. Chá, (2014). "An Existence Result for Non-Convex Optimal Control Problems", WSEAS Transactions on Systems and Control, 9:687-697.
- [25] P. Chunping, L. Y. Lu, J. Liang and G. Yongjun, (2012). "A Time Varying Washout Approach for Flight Simulation Hexapod Motion System", IEEE, Aviation University of Air Force Changchun, 260-263, China.
- [26] Chin-I Huang, Li-Chen and Fu, (2008). "Senseless Maneuver Optimal Washout Filter Design with Human Vestibular Based (HVB) for VR-based Motion Simulator", 17<sup>th</sup> IFAC World Congress (IFAC'08), 14755-14760, Korea.

- [27] A. M. Yousef, (2011). “Model Predictive Control Approach Based Load Frequency Controller”, WSEAS Transactions on Systems and Control, 7(6):265-275.
- [28] I. A. Qaisi and A. Traechtler, (2012). “Human in the Loop; Optimal Control of Driving Simulators and New Motion Quality Criterion”, University of Paderborn, 2235-2240, Germany.
- [29] Volkaner, B., Sozen, S. N., Altuntas, H., Kaya, E. E., and Omurlu, V. E., (2015). “Experimental Motion Cueing Studies Employing Desktop Flight Simulation System”, Proceedings of the 14<sup>th</sup> International Conference on Circuits, Systems, Electronics, Control & Signal Processing CSECS ‘15, 20-22 May 2015, Konya, 24 – 33.
- [30] S.H. Chen and L.D. Fu, (2010). “An Optimal Washout Filter Design for a Motion Platform with Senseless and Angular Scaling Maneuvers”, Proceedings of the American Control Conference, USA, 4295 – 4300.

## **APPENDIX-A**

---

# CALIBRATION PROCESS OF CLASSICAL MOTION CUEING ALGORITHM

Table A.1 Calibration parameters of washout filters

Parameter	Definition
$w_{0x}$	Natural frequency of 1 <sup>st</sup> order high-pass filter for x- axis ( $\omega$ )
$Q_x$	Natural frequency of 2 <sup>st</sup> order high-pass filter for x- axis ( $\omega$ )
$p$	Damping ratio of 2 <sup>nd</sup> order high-pass filter for x-, y- and z- axes ( $\zeta$ )
$w_{0y}$	Natural frequency of 1 <sup>st</sup> order high-pass filter for y- axis ( $\omega$ )
$Q_y$	Natural frequency of 2 <sup>st</sup> order high-pass filter for y- axis ( $\omega$ )
$w_{0z}$	Natural frequency of 1 <sup>st</sup> order high-pass filter for z- axis ( $\omega$ )
$Q_z$	Natural frequency of 2 <sup>st</sup> order high-pass filter for z- axis ( $\omega$ )
$w_{roll}$	Natural frequency of 2 <sup>st</sup> order high-pass filter for roll motion ( $\omega$ )
$w_{pitch}$	Natural frequency of 2 <sup>st</sup> order high-pass filter for pitch motion ( $\omega$ )
$w_{yaw}$	Natural frequency of 2 <sup>st</sup> order high-pass filter for yaw motion ( $\omega$ )
$k$	Damping ratio for roll and pitch motion ( $\zeta$ )
$m$	Damping ratio for yaw motion ( $\zeta$ )
$w_{lx}$	Natural frequency of 2 <sup>st</sup> order high-pass filter for tilt coordination effect about x- axis ( $\omega$ )
$w_{ly}$	Natural frequency of 2 <sup>st</sup> order high-pass filter for tilt coordination effect about y- axis ( $\omega$ )

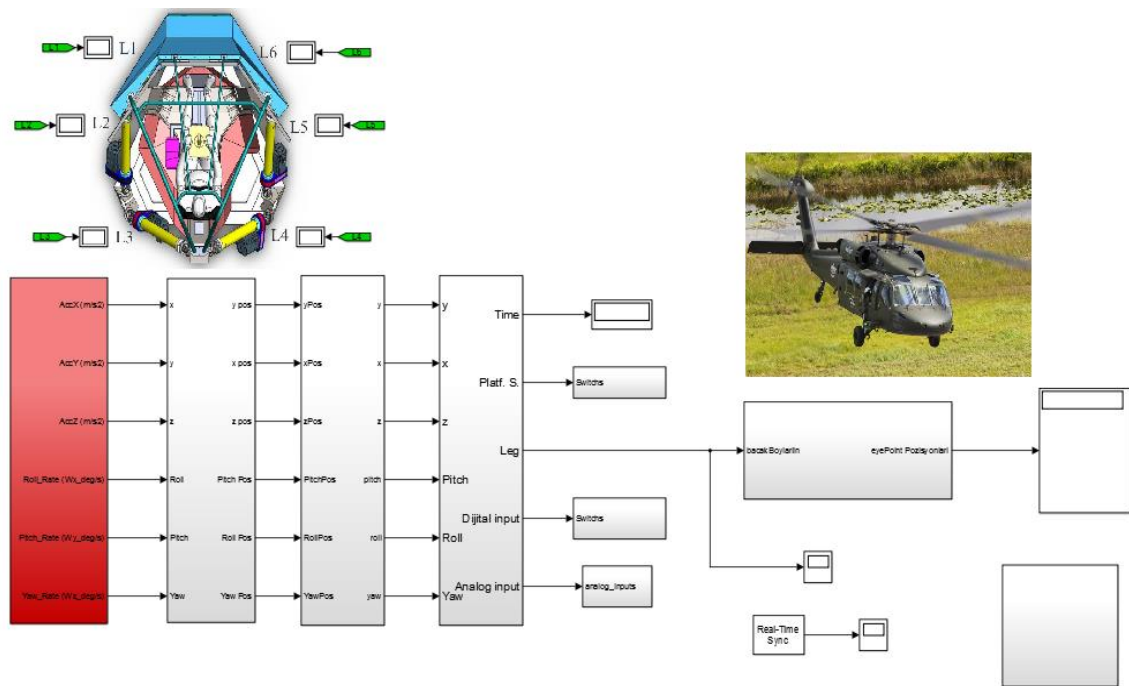


Figure A.1 Simulink block in which received datas from aircraft are scaled

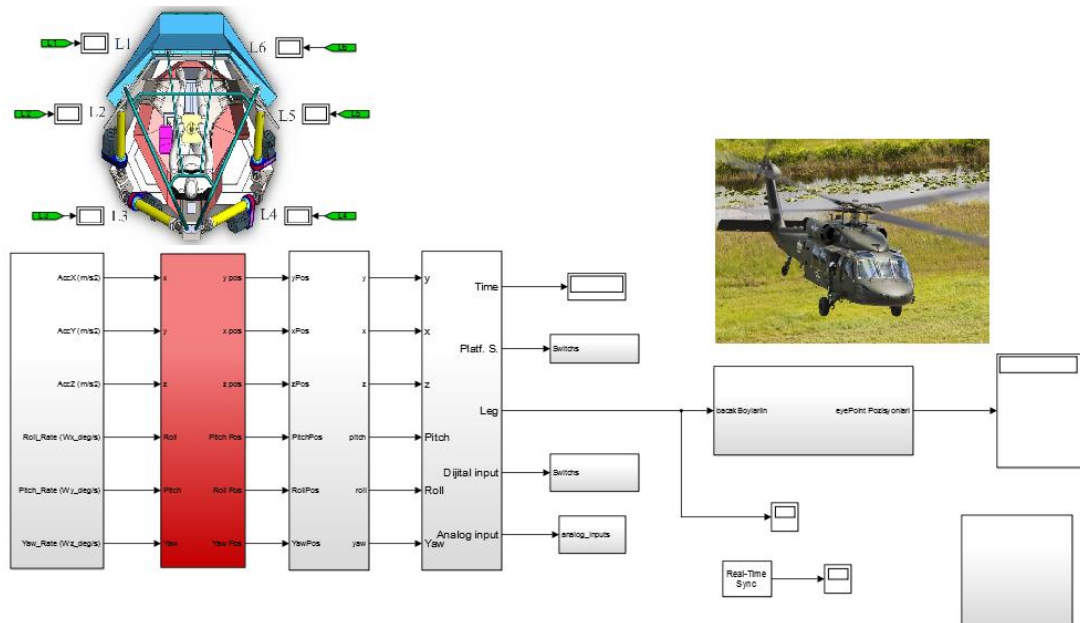


Figure A.2 Representation of Simulink block containing washout filter



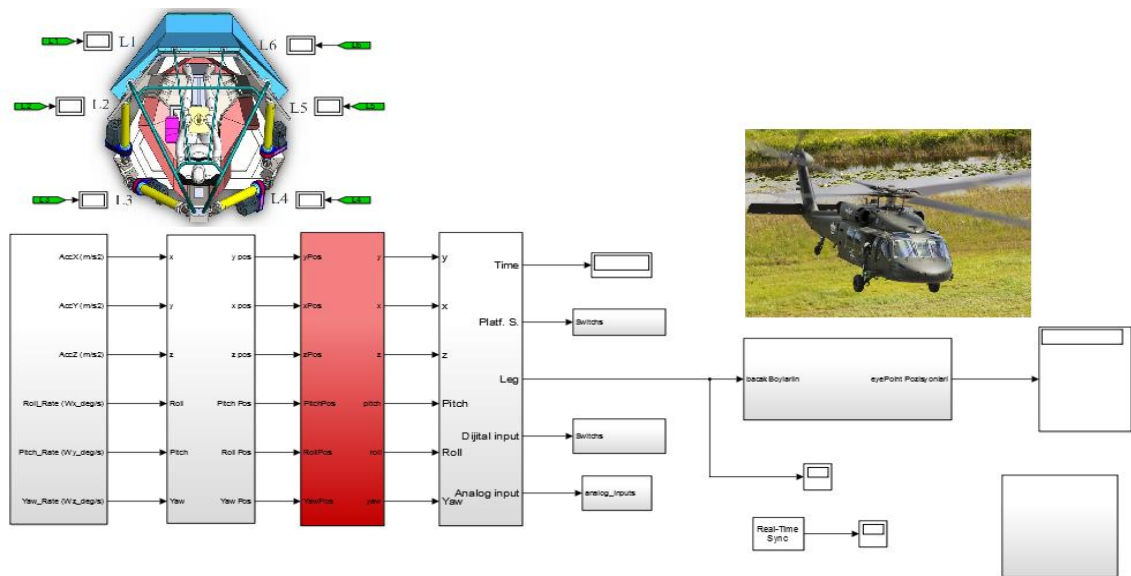


Figure A.3 Representation of “conversion” block including saturation values in Simulink

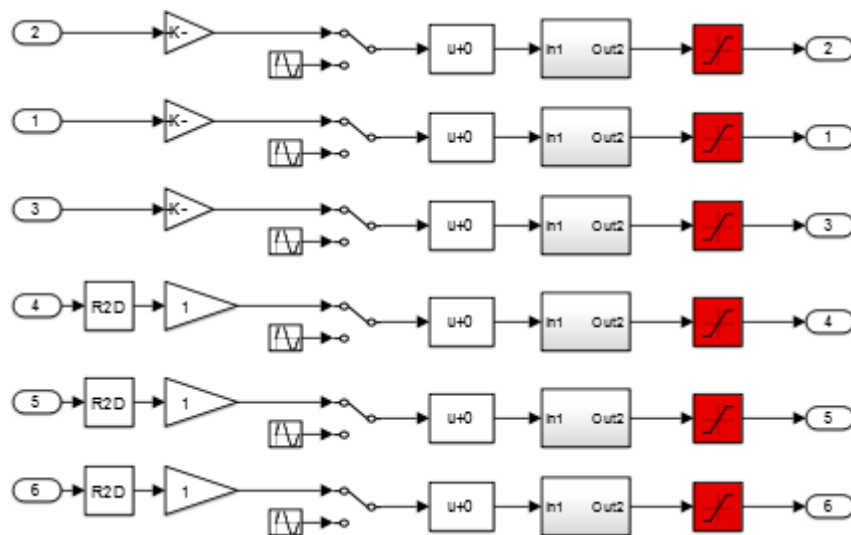


Figure A.4 Interior structure of “conversion” block (saturation – red blocks)

## APPENDIX – B

---

### FORWARD KINEMATIC M. FILE

M. file belongs to forward kinematic analysis is as follows:

```
function [iter,xDf,xD]= forwardKinematic(L,P,B,xDi)
xD=xDi;
Rp = 0.0923;
Rb = 0.0932;
err= 0.000001;
ind=0;
iter= 0;
R=zeros(3,6);
f=zeros(1,6);
L1=0;
J=zeros(6,6);
%%
L=L/10000;
B=B/10000;
P=P/10000;
while ind~=6
    R11=cos(xD(4))*cos(xD(5));
    R12=cos(xD(4))*sin(xD(5))*sin(xD(6))-
sin(xD(4))*cos(xD(6));
    R13=cos(xD(4))*sin(xD(5))*cos(xD(6))+sin(xD(4))*sin(xD(6));
    R21=sin(xD(4))*cos(xD(5));
```

```

R22=sin(xD(4))*sin(xD(5))*sin(xD(6))+cos(xD(4))*cos(xD(6));

R23=sin(xD(4))*sin(xD(5))*cos(xD(6))-
cos(xD(4))*sin(xD(6));

R31=-sin(xD(5));

R32=cos(xD(5))*sin(xD(6));

R33=cos(xD(5))*cos(xD(6));

R=[R11 R12 R13; R21 R22 R23; R31 R32 R33];

for i=1:6

    f(i)=xD(1)*xD(1)+ xD(2)*xD(2)+
xD(3)*xD(3)+Rp*Rp+Rb*Rb+2*(R(1,1)*P(i,1)+R(1,2)*P(i,2))*(xD
(1)-B(i,1))+ 2*(R(2,1)*P(i,1)+R(2,2)*P(i,2))*(xD(2)-
B(i,2))+ 2*(R(3,1)*P(i,1)+R(3,2)*P(i,2))*xD(3)-
2*(xD(1)*B(i,1)+xD(2)*B(i,2))-L(i)*L(i);

    L1=sqrt(xD(1)*xD(1)+
xD(2)*xD(2)+xD(3)*xD(3)+Rp*Rp+Rb*Rb+2*(R(1,1)*P(i,1)+R(1,2)
)*P(i,2))*(xD(1)-
B(i,1))+2*(R(2,1)*P(i,1)+R(2,2)*P(i,2))*(xD(2)-
B(i,2))+2*(R(3,1)*P(i,1)+R(3,2)*P(i,2))*xD(3)-
2*(xD(1)*B(i,1)+xD(2)*B(i,2)));

    J(i,1)=[2*xD(1)+2*(R(1,1)*P(i,1)+R(1,2)*P(i,2))-
2*B(i,1)]*(1/(2*L1));

    J(i,2)=[2*xD(2)+2*(R(2,1)*P(i,1)+R(2,2)*P(i,2))-
2*B(i,2)]*(1/(2*L1));

    J(i,3)=[2*xD(3)+2*(R(3,1)*P(i,1)+R(3,2)*P(i,2))]*(1/(2*L1))
;

    J(i,4)=[2*(xD(1)-B(i,1))*(-
sin(xD(4))*cos(xD(5))*P(i,1)+(-
sin(xD(4))*sin(xD(5))*sin(xD(6))-
cos(xD(4))*cos(xD(6)))*P(i,2))+2*(xD(2)-
B(i,2))*(cos(xD(4))*cos(xD(5))*P(i,1)+(cos(xD(4))*sin(xD(5)
)*sin(xD(6))-sin(xD(4))*cos(xD(6)))*P(i,2))]*(1/(2*L1));

    J(i,5)=[2*(xD(1)-B(i,1))*(-
cos(xD(4))*sin(xD(5))*P(i,1)+cos(xD(4))*cos(xD(5))*sin(xD(6

```

```

)) * P(i, 2)) + 2 * (xD(2) - B(i, 2)) * (-
sin(xD(4)) * sin(xD(5)) * P(i, 1) + sin(xD(4)) * cos(xD(5)) * sin(xD(6)
)) * P(i, 2)) + 2 * xD(3) * (-cos(xD(5)) * P(i, 1) -
sin(xD(5)) * sin(xD(6)) * P(i, 2)) ] * (1 / (2 * L1));

J(i, 6) = [2 * (xD(1) -
B(i, 1)) * (cos(xD(4)) * sin(xD(5)) * cos(xD(6)) + sin(xD(4)) * sin(xD
(6))) * P(i, 2) + 2 * (xD(2) -
B(i, 2)) * (sin(xD(4)) * sin(xD(5)) * cos(xD(6)) -
cos(xD(4)) * sin(xD(6))) * P(i, 2) + 2 * xD(3) * cos(xD(5)) * cos(xD(6))
* P(i, 2)] * (1 / (2 * L1));

end

diffJ = inv(J);

xD(:) = xD(:) - diffJ * f(:);

for i = 1:6

f(i) = xD(1) * xD(1) + xD(2) * xD(2) + xD(3) * xD(3) + Rp * Rp + Rb * Rb + 2 * (R(1
, 1) * P(i, 1) + R(1, 2) * P(i, 2)) * (xD(1) -
B(i, 1)) + 2 * (R(2, 1) * P(i, 1) + R(2, 2) * P(i, 2)) * (xD(2) -
B(i, 2)) + 2 * (R(3, 1) * P(i, 1) + R(3, 2) * P(i, 2)) * xD(3) -
2 * (xD(1) * B(i, 1) + xD(2) * B(i, 2)) - L(i) * L(i);

end

ind = 0;

for k = 1:6

if (abs(f(k)) < err)

ind = ind + 1;

end

end

iter = iter + 1;

if iter >= 200

break

end

end

xDf = [xD(1) * 10000; xD(2) * 10000; xD(3) * 10000; xD(4) * (180/pi) * (-
1); xD(5) * (180/pi) * (-1); xD(6) * (180/pi) * (-1)];

end

```

### LONGITUDINAL MODE OPTIMAL MOTION CUEING ALGORITHM DESIGN

The MATLAB m. file is as follows:

```
% Longitudinal Mode
```

#### Vestibular System Model

```
tau_a = 80
tau_1 = 5.73
tau_2 = 0.005
tau_l = 0.06
G_scc = 28.6479
g = 9.81
Rsz = .559017
gamma1 = 1
gamma2 = pi
samp = 0.001

T0 = 1/(tau_a*tau_1*tau_2)
T1 = (tau_a+tau_1+tau_2)/(tau_a*tau_1*tau_2)
T2 = ((tau_1*tau_2)+tau_a*(tau_1+tau_2))/(tau_a*tau_1*tau_2)
T3 = G_scc/tau_2
T4 = G_scc*tau_l/tau_2

A_scc = [-T2 1 0;-T1 0 1;-T0 0 0]
B_scc = [T3-T2*T4 0;-T1*T4 0;-T0*T4 0]
C_scc = [1 0 0]
D_scc = [T4 0]

A0 = 1/tau_l % 1/tau_l
B0 = 1/tau_l % 1/tau_l
B1 = 1/tau_2 % 1/tau_2
K_oto = 0.4
K_oto_prime = K_oto*tau_1*tau_2/tau_l
G_oto = 4.7059*B1

a = B0 + B1
b = B0*B1
c = B0 + B1 - A0
d = g + Rsz*B0*B1
```

```

e = g*A0
f = 1
h = A0

A_oto = [0 1 0 0 0;-b -a 1 0 0;0 0 0 0 0;0 0 0 0 1;0 0 0 -b -a] % 5x5
B_oto = [c 0;d-a*c 0;e 0; 0 f;0 h-a*f] % 5*2
C_oto = [1 0 0 1 0] % 1x5
D_oto = [-G_oto*K_oto_prime 0] % 1x2

A_ves = [A_scc zeros(3,5);zeros(5,3) A_oto]; % 8x8
B_ves = [B_scc;B_oto]; % 8x2
C_ves = [C_scc zeros(1,5);zeros(1,3) C_oto]; % 2x8
D_ves = [D_scc;D_oto]; % 2x2

Ad = [0 1 0 0;0 0 1 0;0 0 0 0;0 0 0 0];
Bd = [0 0;0 0;0 1;1 0];

An = [-gamma1 0;0 -gamma2];
Bn = [gamma1;gamma2];

A = [A_ves zeros(8,4) -B_ves;zeros(4,8) Ad zeros(4,2);zeros(2,8) zeros(2,4) An]; %
14x14
B = [B_ves;Bd;zeros(2,2)]; %
14x2
C = [C_ves zeros(2,4) -D_ves;zeros(4,8) eye(4) zeros(4,2)]; %
6x14
D = [D_ves;zeros(4,2)]; %
6x2
H = [zeros(12,1);Bn]; %
14x1

```

## Optimal Controller Equations

```

R = ones(2);
Rd = [1 0 0 0;0 4 0 0;0 0 1 0;0 0 0 250];
Qe = [1 1;1 10];
G = [Qe, zeros(2,4);zeros(4,2) Rd]; % 6x6

R1 = C'*G*C; % 14x14
R12 = C'*G*D; % 14x2
R2 = R + D'*G*D; % 2x2
A_prime = A - (B*inv(R2)*R12'); % 14x14
R1_prime = R1 - (R12*inv(R2)*R12'); % 14x14

```

## Computing Algebraic Riccati Equation

```

L = B*inv(R2)*B';

n=size(A_prime,1);

Z=[A_prime -L
-R1_prime -A_prime'];

```

```

[U1,S1]=schur(Z);
[U,S]=ordschur(U1,S1,'lhp');

P=U(n+1:end,1:n)*U(1:n,1:n)^-1; % 14x14

K = inv(R2)*(B'*P+R12'); % 2x14

K1 = [K(1,1) K(1,2) K(1,3) K(1,4) K(1,5) K(1,6) K(1,7) K(1,8)
      K(2,1) K(2,2) K(2,3) K(2,4) K(2,5) K(2,6) K(2,7) K(2,8)]; % 2x8

K2 = [K(1,9) K(1,10) K(1,11) K(1,12);K(2,9) K(2,10) K(2,11) K(2,12)]; % 2x4

K3 = [K(1,13) K(1,14);K(2,13) K(2,14)]; % 2x2

A1 = [A_ves - B_ves*K1 -B_ves*K2;-Bd*K1 Ad - Bd*K2]; % 12x12
B1 = [-B_ves*(eye(2) + K3);-Bd*K3]; % 12x12
C1 = [K1 K2]; % 2x12
D1 = -K3; % 2x2

sysv = ss(A1, B1, C1, D1);

[b1,a1] = ss2tf(A1, B1, C1, D1,1); % b1 2x13 a1 1x13

w11 = tf(b1(1,:),a1);
w21 = tf(b1(2,:),a1);

[b2,a2] = ss2tf(A1, B1, C1, D1,2); % b2 2x13 a2 1x13

w12 = tf(b2(1,:),a2);
w22 = tf(b2(2,:),a2);

[num11, den11] = tfdata(w11,'v');
[num12, den12] = tfdata(w12,'v');
[num21, den21] = tfdata(w21,'v');
[num22, den22] = tfdata(w22,'v');

w11d = c2d(ss(w11), samp, 'tustin');
w21d = c2d(ss(w21), samp, 'tustin');
w12d = c2d(ss(w12), samp, 'tustin');
w22d = c2d(ss(w22), samp, 'tustin');

[num11d, den11d] = tfdata(w11d,'v');
[num12d, den12d] = tfdata(w12d,'v');
[num21d, den21d] = tfdata(w21d,'v');
[num22d, den22d] = tfdata(w22d,'v');

nump = [T4 T3 0 0];
denp = [1 T2 T1 T0];

syys = tf(nump,denp);
syysp = c2d(syys, samp, 'tustin');

[numdp, dendp] = tfdata(syysp);

```

```
numPF = cell2mat(numdp);  
denPF = cell2mat(dendp);
```



## APPENDIX – D

### JOINT AVIATION REQUIREMENTS (JAR) STANDARDS

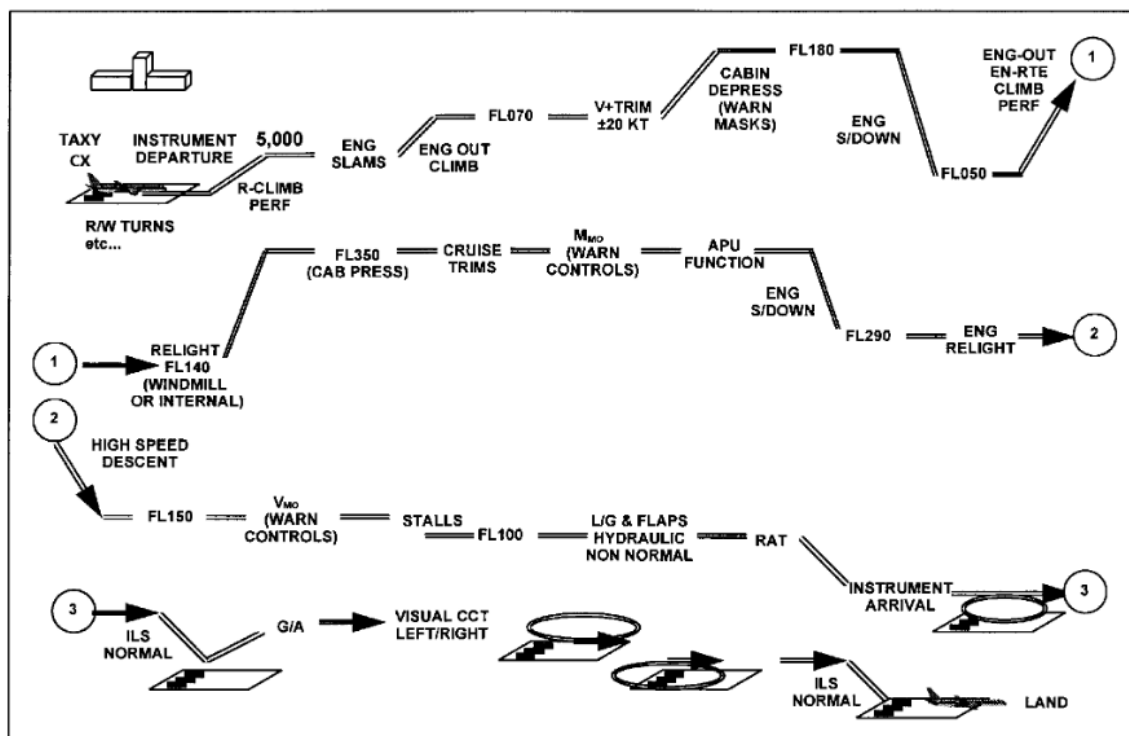


Figure D.1 Typical test profile for 2 hours

## D.1 Flight Simulator standards

SIMULATOR STANDARDS		SIMULATOR LEVEL				COMMENTS
		A	B	C	D	
a.	Cockpit, a full scale replica of the aeroplane simulated. Direction of movement of controls and switches identical to that in the aeroplane. The cockpit, for simulator purposes, consists of all that space forward of a cross section of the fuselage at the most extreme aft setting of the pilots' seats. Additional required crew member duty stations and those required bulkheads aft of the pilots' seats are also considered part of the cockpit and should replicate the aeroplane.	✓	✓	✓	✓	
b.	Circuit breakers that affect procedures and/or result in observable cockpit indications properly located and functionally accurate.	✓	✓	✓	✓	
c.	Effect of aerodynamic changes for various combinations of drag and thrust normally encountered in flight corresponding to actual flight conditions, including the effect of change in aeroplane attitude, thrust, drag, altitude, temperature, gross weight, centre of gravity location, and configuration.	✓	✓	✓	✓	For level 'A' generic ground handling, flare and touchdown effect are acceptable.
d.	All relevant instrument indications involved in the simulation of the applicable aeroplane automatically respond to control movement by a crew member or induced disturbance to the simulated aeroplane; e.g., turbulence or windshear.	✓	✓	✓	✓	Numerical values should be presented in accordance with ICAO Annex 5.
e.	Communications, Navigation, and Caution and Warning equipment corresponding to that installed in the applicant's aeroplane with operation within the tolerances prescribed for the applicable airborne equipment.	✓	✓	✓	✓	

SIMULATOR STANDARDS	SIMULATOR LEVEL				COMMENTS
	A	B	C	D	
f. In addition to the flight crewmember stations, two suitable seats for the Instructor/Delegated Examiner and Authority inspector. The Authority will consider options to this standard based on unique cockpit configurations. These seats should provide adequate vision to the pilot's panel and forward windows. Observer seats need not represent those found in the aeroplane but should be equipped with similar positive restraint devices.	✓	✓	✓	✓	
g. Simulator systems should simulate applicable aeroplane system operation, both on the ground and in flight. Systems should be operative to the extent that normal, abnormal, and emergency operating procedures appropriate to the simulator application can be accomplished.	✓	✓	✓	✓	
h. Instructor controls to enable the operator to control all required system variables and insert abnormal or emergency conditions into the aeroplane systems.	✓	✓	✓	✓	
i. Control forces and control travel which correspond to that of the replicated aeroplane. Control forces should react in the same manner as in the aeroplane under the same flight conditions.	✓	✓	✓	✓	
j. Significant cockpit sounds which result from pilot actions corresponding to those of the aeroplane.	✓	✓	✓	✓	
k. Sound of precipitation, windshield wipers, and other significant aeroplane noises perceptible to the pilot during normal operations and the sound of a crash when the simulator is landed in excess of limitations.			✓	✓	Statement of Compliance required.

SIMULATOR STANDARDS	SIMULATOR LEVEL				COMMENTS
	A	B	C	D	
l. Realistic amplitude and frequency of cockpit noises and sounds, precipitation, windshield wipers, and airframe sounds. The sounds should be coordinated with the weather representations required by paragraph 3.3 – 4.f (page 2-C-57).				✓	Tests required for noises and sounds that originate from the aeroplane or aeroplane systems.  See paragraph 3.3 – 5.c (page 2-C-59).
m. Ground handling and aerodynamic programming to include:  (1) Ground Effect – for example: roundout, flare, and touchdown. This requires data on lift, drag, pitching moment, trim, and power Ground Effect.  (2) Ground reaction – reaction of the aeroplane upon contact with the runway during landing to include strut deflections, tyre friction, side forces, and other appropriate data, such as weight and speed, necessary to identify the flight condition and configuration.  (3) Ground handling characteristics – steering inputs to include crosswind, braking, thrust reversing, deceleration and turning radius.	✓	✓	✓	✓	Statement of Compliance required. Tests required.  For Level 'A' simulators, ground handling generically represented to the extent that allows turns within the confines of the R/W and adequate control on the landing and roll-out from a X-wind landing.
n. Windshear models which provide training in the specific skills required for recognition of windshear phenomena and execution of recovery manoeuvres. Such models should be representative of measured or accident derived winds, but may include simplifications which ensure repeatable encounters. For example, models may consist of independent variable winds in multiple simultaneous components. Wind models should be available for the following critical phases of flight:  (1) Prior to take-off rotation (2) At lift-off (3) During initial climb (4) Short final approach			✓	✓	Tests required. See paragraph 3.3 – 2.h (page 2-C-50).

SIMULATOR STANDARDS		SIMULATOR LEVEL				COMMENTS
		A	B	C	D	
n.	<p>(continued)</p> <p>The FAA Windshear Training Aid, wind models from the Royal Aerospace Establishment (RAE), the Joint Airport Weather Studies (JAWS) Project or other recognized sources may be implemented and should be supported and properly referenced in the QTG.</p> <p>Wind models from alternative sources may also be used if supported by aeroplane related data and such data is properly supported and referenced in the QTG. Use of alternative data should be coordinated with the Authority prior to submittal of the QTG for Approval.</p>					
o.	Representative crosswinds and instructor controls for wind speed and direction.	✓	✓	✓	✓	
p.	<p>Representative stopping and directional control forces for at least the following runway conditions based on aeroplane related data:</p> <p>(1) Dry (2) Wet (3) Icy (4) Patchy Wet (5) Patchy Icy (6) Wet on Rubber Residue in Touchdown Zone.</p>			✓	✓	<p>Statement of Compliance required.</p> <p>Objective Tests required for (1), (2), (3), Subjective check for (4), (5), (6).</p> <p>See paragraph 3.3 – 1.e (pages 2–C–38 and 39).</p>
q.	Representative brake and tyre failure dynamics (including antiskid) and decreased brake efficiency due to brake temperatures based on aeroplane related data.			✓	✓	<p>Statement of Compliance required.</p> <p>Tests required for decreased braking efficiency due to brake temperature.</p> <p>See paragraph 3.3 – 2.g (page 2–C–50).</p>
r.	A means for quickly and effectively testing simulator programming and hardware. This may include an automated system which could be used for conducting at least a portion of the tests in the QTG.			✓	✓	Statement of Compliance required.

SIMULATOR STANDARDS	SIMULATOR LEVEL				COMMENTS
	A	B	C	D	
s. Simulator computer capacity, accuracy, resolution, and dynamic response sufficient for the qualification level sought.	✓	✓	✓	✓	Statement of Compliance required.
<p>t. Control feel dynamics which replicate the aeroplane simulated.</p> <p>Free Response of the controls should match that of the aeroplane within the tolerance given in paragraph 3. Initial and Upgrade evaluations will include control Free Response (column, wheel, pedal) measurements recorded at the controls. The measured responses should correspond to those of the aeroplane in take-off, cruise, and landing configurations.</p> <p>1) For aeroplanes with Irreversible Control Systems, measurements may be obtained on the ground if proper pitot static inputs are provided to represent conditions typical of those encountered in flight. Engineering validation or aeroplane manufacturer rationale will be submitted as justification to ground test or omit a configuration.</p> <p>(2) For simulators requiring static and dynamic tests at the controls, special test fixtures will not be required during initial evaluation if the operator's QTG shows both text fixture results and alternate test method results such as computer data plots, which were obtained concurrently. Repetition of the alternate method during initial evaluation may then satisfy this requirement.</p>			✓	✓	Tests required. See paragraph 3.3 – 2.b (1) to (3) (page 2-C-42) and paragraph 3.4.1 (pages 2-C-60 to 62).

SIMULATOR STANDARDS	SIMULATOR LEVEL				COMMENTS
	A	B	C	D	
<p>u. Relative response of the visual system, cockpit instruments and initial motion system response should be coupled closely to provide integrated sensory cues. These systems should respond to abrupt pitch, roll, and yaw inputs at the pilot's position within the permissible delay, but not before the time, when the aeroplane would respond under the same conditions. Visual scene changes from steady state disturbance should occur within the system dynamic response limit but not before the resultant motion onset. The test to determine compliance with these requirements should include simultaneously recording the analog output from the pilot's control column, wheel, and pedals, the output from the accelerometer attached to the motion system platform located at an acceptable location near the pilot's seats, the output signal to the visual system display (including visual system analog delays), and the output signal to the pilot's attitude indicator or an equivalent test approved by the Authority. The test results in a comparison of a recording of the simulator's response with actual aeroplane response data in the take-off, cruise, and landing configuration.</p> <p>The intent is to verify that the simulator system Transport Delays or time lags are less than the permissible delay and that the motion and visual cues relate to actual aeroplane responses. For aeroplane response, acceleration in the appropriate rotational axis is preferred.</p> <p>As an alternative, a Transport Delay test may be used to demonstrate that the simulator system does not exceed the permissible delay.</p>	✓	✓	✓	✓	<p>Test required. See paragraph 3.3 – 4.a (page 2–C–53).</p> <p>For Level 'A' &amp; 'B' simulators the maximum permissible delay is 300 milliseconds.</p> <p>For Level 'C' and 'D' simulators the maximum permissible delay is 150 milliseconds.</p>

SIMULATOR STANDARDS	SIMULATOR LEVEL				COMMENTS
	A	B	C	D	
<p>u. (continued)</p> <p>This test should measure all the delay encountered by a step signal migrating from the pilot's control through the control loading electronics and interfacing through all the simulation software modules in the correct order, using a handshaking protocol, finally through the normal output interfaces to the motion system, to the visual system and instrument displays. A recordable start time for the test should be provided by a pilot flight control input. The test mode should permit normal computation time to be consumed and should not alter the flow of information through the hardware/software system.</p> <p>The Transport Delay of the system is then the time between control input and the individual hardware responses. It need only be measured once in each axis.</p>					
<p>v. Aerodynamic modelling which, for aeroplanes issued an original type certificate after June 1980, includes low altitude level flight Ground Effect, Mach effect at high altitude, effects of airframe icing, normal and reverse dynamic thrust effect on control surfaces, aeroelastic representations, and representations of non-linearities due to sideslip based on aeroplane Flight Test Data provided by the manufacturer.</p>				✓	<p>Statement of Compliance required.</p> <p>Paragraph 2.1 m (page 2-C-21) and paragraph 3.4.2 (page 2-C-62) provide information on Ground Effect. Mach effect, aeroelastic representations, and non-linearities due to sideslip are normally included in the simulator aerodynamic model, but the Statement of Compliance should address each of them. Separate tests for thrust effects and a Statement of Compliance and demonstration of icing effects are required.</p>
<p>w. Aerodynamic and ground reaction modelling for the effects of reverse thrust on directional control.</p>		✓	✓	✓	<p>Statement of Compliance required.</p> <p>Test required. See paragraph 3.3 – 2.e.(7) (page 2-C-49).</p>



SIMULATOR STANDARDS	SIMULATOR LEVEL				COMMENTS
	A	B	C	D	
x. Self-testing for simulator hardware and programming to determine compliance with the simulator performance tests as prescribed in paragraph 3. Evidence of testing should include simulator number, date, time, conditions, tolerances, and the appropriate dependent variables portrayed in comparison with the aeroplane standard. Automatic flagging of 'out-of-tolerance' situations is encouraged.			✓	✓	Statement of Compliance required.
y. Timely permanent update of simulator hardware and programming subsequent to aeroplane modification.	✓	✓	✓	✓	For Level 'A': sufficient for the qualification level sought.
z. Daily preflight documentation either in the daily log or in a location easily accessible for review.	✓	✓	✓	✓	

## D.2 Motion System standards

SIMULATOR STANDARDS		SIMULATOR LEVEL				COMMENTS
		A	B	C	D	
a.	Motion cues perceived by the pilot representative of the aeroplane e.g. touchdown cues should be a function of the simulated rate of descent.	✓	✓	✓	✓	
b.	<p>A motion system:</p> <p>(1) Providing sufficient cueing which may be of a generic nature to accomplish the required tasks.</p> <p>(2) Having a minimum of 3 degrees of freedom (pitch, roll &amp; heave).</p> <p>(3) Which produces cues at least equivalent to those of a six-degrees-of-freedom synergistic platform motion system.</p>	✓	✓	✓	✓	Statement of Compliance required. Tests required.
c.	A means of recording the motion response time for comparison with aeroplane data.	✓	✓	✓	✓	See Para 2.1.u. (page 2-C-24) of this Part.
d.	<p>Special effects programming to include:</p> <p>(1) Runway rumble, oleo deflections, effects of groundspeed and uneven runway characteristics.</p> <p>(2) Buffets on the ground due to spoiler/speedbrake extension and thrust reversal.</p> <p>(3) Bumps after lift-off of nose and main gear.</p> <p>(4) Buffet during extension and retraction of landing gear.</p> <p>(5) Buffet in the air due to flap and spoiler/speedbrake extension.</p> <p>(6) Stall buffet to, but not necessarily beyond, the certified stall speed, <math>V_{st}</math>.</p>	✓	✓	✓	✓	For Level 'A': may be of a generic nature sufficient to accomplish the required tasks.

SIMULATOR STANDARDS	SIMULATOR LEVEL				COMMENTS
	A	B	C	D	
d. (continued) (7) Representative touchdown cues for main and nose gear. (8) Nosewheel scuffing and thrust effect with brakes set. (9) Mach buffet.					
e. Characteristic buffet motions that result from operation of the aeroplane (for example, high-speed buffet, extended landing gear, flaps, nosewheel scuffing, stall) which can be sensed at the flight deck. The simulator should be programmed and instrumented in such a manner that the characteristic buffet modes can be measured and compared with aeroplane data. Aeroplane data are also required to define flight deck motions when the aeroplane is subjected to atmospheric disturbances. General purpose disturbance models that approximate demonstrable Flight Test Data are acceptable. Tests with recorded results which allow the comparison of relative amplitudes versus frequency are required.				✓	Statement of Compliance required. Tests required. See paragraph 3.3 – 3.e (page 2-C-53).

### D.3 Visual System standards

SIMULATOR STANDARDS	SIMULATOR LEVEL				COMMENTS
	A	B	C	D	
a. Visual system capable of meeting all the standards of this paragraph and paragraphs 3 & 4 (Validation and Functions and Subjective Tests) as applicable to the level of qualification requested by the applicant.	✓	✓	✓	✓	
b. Continuous minimum collimated visual field-of-view of:  – 45 degrees horizontal and 30 degrees vertical field of view simultaneously for each pilot.  – 75 degrees horizontal and 30 degrees vertical per pilot seat. Both pilot seat visual systems should be able to be operated simultaneously.	✓	✓	✓	✓	Wide angle systems providing cross cockpit viewing should provide a minimum of 150 degrees horizontal field of view; 75 degrees per pilot seat operated simultaneously. 30 degrees vertical field of view may be insufficient for paragraph 3.3 – 4.d (page 2–C–55).
c. A means of recording the visual response time for visual systems in paragraph 3.3 – 4.a (page 2–C–53).	✓	✓	✓	✓	
d. Verification of visual ground segment and visual scene content at a decision height on landing approach. The QTG should contain appropriate calculations and a drawing showing the pertinent data used to establish the aeroplane location and visual ground segment. Such data should include, but is not limited to:  (1) Airport and runway used. (2) Glide slope transmitter location for the specified runway. (3) Position of the glide slope receiver antenna relative to the aeroplane main landing wheels. (4) Approach and runway light intensity settings. (5) Aeroplane pitch angle.	✓	✓	✓	✓	See paragraph 3.3 – 4.c (page 2–C–55).

SIMULATOR STANDARDS	SIMULATOR LEVEL				COMMENTS
	A	B	C	D	
d. (continued) The above parameters should be presented for the aeroplane in the landing configuration and a main wheel height of 30 m (100 ft) above the touchdown zone. The visual ground segment and scene content should be determined for a runway visual range of 350 m (1 200 ft).					
e. Visual cues to assess sink rate and depth perception during take-off and landing.	✓	✓	✓	✓	For Level 'A' , Visual cueing sufficient to support changes in approach path by using runway perspective.
f. Test procedures to quickly confirm visual system colour, RVR, focus, intensity, level horizon, and attitude as compared with the simulated attitude indicator.			✓	✓	Statement of Compliance required. Tests required. See paragraph 3.3 – 4.b (page 2–C–54).
g. Dusk scene to enable identification of a visible horizon and typical terrain characteristics such as fields, roads, bodies of water.			✓	✓	Statement of Compliance required. Tests required. See paragraph 3.3 – 4.e (page 2–C–56).
h. A minimum of ten levels of occulting. This capability should be demonstrated by a visual model through each channel.			✓	✓	Statement of Compliance required. Tests required. See paragraph 3.3 – 4.b (page 2–C–54).
i. Surface resolution will be demonstrated by a test pattern of objects shown to occupy a visual angle of 3 arc minutes in the visual scene from the pilot's eyepoint. This should be confirmed by calculations in the statement of compliance.			✓	✓	See paragraph 3.3 – 4.b (page 2–C–54). Where a night/dusk system is used on a Level C simulator, this test does not apply.
j. Lightpoint size – not greater than 6 arc minutes measured in a test pattern consisting of a single row of lightpoints reduced in length until modulation is just discernible, a row of 40 lights will form a 4 degree angle or less.			✓	✓	See paragraph 3.3 – 4.b (page 2–C–54). This is equivalent to a light point resolution of 3 arc minutes.
k. Lightpoint contrast ratio – not less than 25:1 when a square of at least 1 degree filled with lightpoints (i.e. lightpoint modulation is just discernible) is compared with the adjacent background.			✓	✓	

		SIMULATOR LEVEL				COMMENTS
		A	B	C	D	
I.	<p>Daylight, dusk, and night visual scenes with sufficient scene content to recognise airport, the terrain, and major landmarks around the airport and to successfully accomplish a visual landing. The daylight visual scene should be part of a total daylight cockpit environment which at least represents the amount of light in the cockpit on an overcast day. Daylight visual system is defined as a visual system capable of producing, as a minimum, full colour presentations, scene content comparable in detail with that produced by 4 000 edges or 1 000 surfaces for daylight and 4 000 light points for night and dusk scenes, 20 cd/m<sup>2</sup> (6 ft-Lamberts) of light measured at the pilot's eye position (Highlight Brightness), and a display which is free of apparent quantization and other distracting visual effects while the simulator is in motion. The simulator cockpit ambient lighting should be dynamically consistent with the visual scene displayed. For daylight scenes, such ambient lighting should not 'washout' the displayed visual scene nor fall below 17 cd/m<sup>2</sup> (5 foot-Lamberts) of light as reflected from an approach plate at knee height at the pilot's station. [ ] All brightness and resolution requirements should be validated by an Objective Test and will be retested at least yearly. Testing may be accomplished more frequently if there are indications that the performance is degrading on an accelerated basis.</p> <p>Compliance of the brightness capability may be demonstrated with a test pattern of white light using a spot photometer.</p>				✓	<p>Statement of Compliance required.</p> <p>Tests required.</p> <p>See paragraph 3.3 – 4.b (page 2–C–54).</p> <p>The ambient lighting should provide an even level of illumination which is not distracting to the pilot.</p>

SIMULATOR STANDARDS	SIMULATOR LEVEL				COMMENTS
	A	B	C	D	
<p>I. (continued)</p> <p>(1) Contrast Ratio. A raster drawn test pattern filling the entire visual scene (three or more channels) should consist of a matrix of black and white squares no larger than 10 degrees and no smaller than 5 degrees per channel with a white square in the centre of each channel.</p> <p>Measurement should be made on the centre bright square for each channel using a 1 degree spot photometer. This value should have a minimum brightness of 7 cd/m<sup>2</sup> (2 foot-Lamberts). Measure any adjacent dark squares. The contrast ratio is the bright square value divided by the dark square value. Minimum test contrast ratio result is 5:1.</p> <p>NOTE: During contrast ratio testing, cockpit ambient light levels should be maintained as required in paragraph I above.</p> <p>(2) Highlight Brightness Test. Maintaining the full test pattern described in paragraph (1) above, superimpose a highlight on the centre white square of each channel and measure the brightness using the 1 degree spot photometer. Lightpoints are not acceptable. Use of calligraphic capabilities to enhance raster brightness is acceptable.</p>				✓	

## CURRICULUM VITAE

---

### PERSONAL INFORMATION

**Name Surname** : Şemsettin Numan SÖZEN  
**Date of birth and place** : 16.05.1990 - Adıyaman  
**Foreign Languages** : English, German, Italian  
**E-mail** : sozennuman@gmail.com

### EDUCATION

Degree	Department	University	Date of Graduation
Master	Mechatronics Engineering	Yıldız Technical University	2016
Undergraduate	Mechatronics Engineering	Gaziosmanpaşa University	2014
High School	Adıyaman Anatolian High School		2008

### WORK EXPERIENCE

Year	Corporation/Institute	Enrollment
2011	Gaziosmanpaşa University	2011
2012	DİMES INC.	2012
2013	Indesit Company	2013
2014	Yıldız Technical University	2016



## **PUBLISHERMENTS**

### **Papers**

1. Volkaner, B., Sozen, S., N., Omurlu, V., E.. “Realization of a Desktop Flight Simulation System for Motion Cueing Studies,” International Journal of Advanced Robotic Systems 13 (85), 1-14.

### **Conference Papers**

1. 4. Ulusal Mekatronik Mühendisliği Öğrenci Kongresi (MEMÖK) – SABİT MIKNATISLI DC SERVO MOTORUN HIZ KONTROLÜ İÇİN PID KONTROLÖR TASARIMI.
2. 5. Ulusal Mekatronik Mühendisliği Öğrenci Kongresi (MEMÖK) – İNDÜKSİYON ISITMA PRENSİBİ İLE ÇALIŞAN ANİ SU ISITICISININ TASARIMI VE KONTROLÜ.
3. Volkaner, B., Sozen, S. N., Altuntas, H., Kaya, E. E., Omurlu, V. E., “Experimental Motion Cueing Studies Employing Desktop Flight Simulation System”. Proceedings of the 14th International Conference on Circuits, Systems, Electronics, Control & Signal Processing (CSECS ‘ 15), (pp. 24 –33). WSEAS Press, Konya, 2015.
4. Volkaner, B., Sözen, S., N., Altuntaş, H., Kaya, E., E., Ömürlü, V., E.. “Hareket Algı Çalışmaları için Masaüstü Uçuş Simülatörü” Otomatik Kontrol Türk Milli Komitesi Ulusal Toplantısı (TOK), (2015).

### **Projects**

1. Tokat Solar Car Team (TOGATA) – A member of Mechatronics Design Team and the Pilot of the car – 2011.
2. Tokat Solar Car Team (TOGATA) – A member of Mechatronics Design Team – 2012.
3. Sound Tracking Robot.
4. Line Tracking Robot with PID controller.
5. Crescent - wheel robot.
6. Project team leader of CAMPUS newspaper.
7. Organizer of ISO Quality Management Education.
8. Volunteer and Moderator of Young Guru Academy (YGA).