# Experimental Motion Cueing Studies Employing Desktop Flight Simulation System

Berkay Volkaner, S. Numan Sozen, Hakan Altuntas, E. Ebru Kaya, V. Emre Omurlu
Mechatronic Engineering Department
Yildiz Technical University
Barbaros Bulv., Yildiz Camp., A309, 34340
Turkey
bvolkaner, nmnsozen, hakanaltuntas1,eceebrukaya, eomurlu@gmail.com

*Abstract:* - Parallel manipulators are frequently used in flight simulator applications along with a motion cueing algorithm in order to create an unbounded motion feeling in a limited workspace. Translational accelerations and angular velocities of simulated environment obtained from FlightGear are processed via "motion cueing" algorithm and these acceleration and velocity information are transmitted to the manipulator as set points of each leg. Major issue in flight simulators is that the simulation has infinite space and the manipulator has a limited one. "Washout Filter" used in motion cueing algorithm eliminates this issue. In this study, motion cueing algorithm is used in a 6 DoF desktop parallel manipulator and functionality of the algorithm is confirmed with tests performed in Simulink real time environment.

*Key-Words:* - Stewart Platform, Motion Cueing, Washout Filter, Flight Simulator

## 1 Introduction

Flight simulators are useful for studies related to vehicle design, tests, and simulator systems are elements that allow us to understand the pilot behavior as close as to reality. Several universities and industrial laboratories are performing researches for new generation prototypes and vehicle dynamic models, nowadays.

Simulators create a realistic flight feeling using feedback of the motion information [1] and are completely reliable, realizable and can be implemented. It is proposed to increase this reality by improving the software, hardware and physical abilities of simulators. To get closer to the real flight feeling, it is important to employ a robust motion cueing system [2].

Simulators are generally processes of "man in the loop" and include an intensive code algorithm. For instance, in a Level D simulator, there are numerous code arrays and they are expensive software systems. Because of their complex structure and high cost, simulator design is quite a difficult task [3].

In simulator systems, examining which factor causes the differences between real and virtual data is needed. These factors are generally separated into 3 categories; human motion perception (vestibular) system, the problems that may occur on the visual flight simulation and the sound system. Problems, possible to occur in the motion perception applications, are as follows:

- What does the human perception depend on in real time applications?
- How is it possible to improve human perception by changing the parameters of these factors? [4].

Human motion perception system constitutes the basis of motion perception studies. Motion is sensed by a human via "vestibular system" located in inner ear and consists of semi-circular canals for sensing angular motion and otoliths for sensing translational motion. Mathematical models of vestibular system was firstly used in [5] and simplified transfer function models were presented in [6, 7]. Thus, simulation studies became possible to perform [8].

Visual scaling creates differences in image perception feeling. The optimal level visual scaling affects the image perception positively. In a study about this, changes in visual effect was examined under different visual scale factors [9].

There are four basic algorithms used for motion cueing. Classical Algorithm, Optimal Algorithm, Adaptive Algorithm and Model-Predictive Algorithm. Classical algorithm was found in NASA Ames Research Center by Conrad & Schmidt in 1969 and shortly after that rotating coordinate filter was created by Conrad, Schmidt & Douviller in 1973. Generally, classical algorithm consists of combination of high and low pass filters. Corner frequency and damping ratio are initially set. While examining the acceleration effect, integrating the gravitational effect into the acceleration change in the vertical direction is also an important issue. Firstly, translational and rotational accelerations are scaled and rotational accelerations are passed through high pass filters. To return the simulator to its initial position, second order and third order filters may be used. If the effects are scaled and limited then just second order filter usage is sufficient [10].

Because of workspace limitations, adaptive algorithm was developed by Parrish, Dieudonne, Bowles and Martin in 1975 and this new algorithm is implemented in frequency domain to eliminate the problem. In this algorithm, it was provided that the coefficients are different [15] and motion perception is considered as tracking problem [16]. the acceleration that is obtained from the simulator should be tracked as soon as possible by the platform [17].

By applying adaptive filter in classical algorithm, "adaptive classical motion cueing algorithm" was developed. Adaptive gain is obtained by minimizing a cost function [11] and the parameter which is being changed here is high-pass filter gain. The purpose of this algorithm is to minimize the cost function. Defusing wrong clues at the actuator extensions and providing a smooth simulation environment is the advantage of the adaptive algorithm but due to the structure of the algorithm which may vary, there are sometimes gaps in motion perception.

In [20], "developed adaptive algorithm" structure was also examined with the developing of adaptive algorithm and this structure was implemented as an alternative method to keep the simulator in limits of workspace. This structure has single DoF motion limit calculator block alternatively. This block calculates the kinematics and the central point coordinates of the platform together.

Filter parameters like corner frequency, damping ratio, gain, cost and adaptive step size are adjusted according to the balance of re-creation of translational motion in limited workspace. Initial cost values are determined based on experience.

After classical filter design, implementing trial-and-error method is sufficient to determine necessary parameters [12].

There are several studies comparing usage of classical algorithm on 3, 6 and 8 degrees of freedom (DoF) simulation platforms. On 3 DoF platforms, the translational feedback of the system was quite good. On 6 DoF platforms, motion were felt more intensively because of motion freedom. On 8 DoF platforms, similar performance of 6 DoF platforms were observed [13].

From the comparison between platforms that have different DoF, following results were obtained.

- Rotation effect must be created as little as possible.
- If the rotation coordination effect is unavoidable, head of pilot must be taken as the reference point.
- Small washout corner frequency must be chosen.
- Filter parameters must be chosen so as to decrease the washout signal levels.

In order to handle the phase delay between the simulation and the platform, there should also be a software that is flexible and effective in terms of cost. There are also methods of compensation available (i.e. fuzzy logic system, adaptive model and nonlinear solving) but most of these methods require closed-loop control [14].

Model-predictive algorithm which is a novel motion cueing algorithm approach is presented in [18].

In some cases, motion cueing algorithms give good results but in some situations due to the motion, actuators may reach the maximum limits and this may disrupt the system for which adaptive algorithm has nonlinear filters [19].

To reduce the cost function, optimal control theory including human vestibular system model was implemented by Sivan, Ishsalom & Huang in 1982. Linear motion perception model was suggested by Hosman & Van Der Vaart in 1981. In this algorithm, Riccati equation is adjusted in real time by using Newton Raphson method. The main problem is to assign the W parameter (in transfer function) which is used to obtain desired position of platform and it minimizes motion perception error. It is observed that the optimal motion cueing algorithm is more effective than the classical and adaptive algorithm [21].

Pouliot suggested two approaches in terms of comparison of motions [22]:

- Comparison of simulator and air vehicle in terms of angular velocity and some forces that produced.
- Comparison of angular velocity with jerk.

In suggested approach:

- Comparison of translational acceleration or angular velocity of simulator with vehicle data.
- Comparison of rotational acceleration or angular velocity with vehicle data.

Studies on the motion perception are based on linear filtering and optimal control. Recently, studies on model-predictive algorithm have been performed and they have been associated with optimal control and it was observed that motion-predictive control algorithm is more effective than classical algorithm under similar conditions. In model-predictive algorithm, the acceleration obtained from simulation is converted to the acceleration that was sensed by motion perception system. This acceleration value is used as a reference value for operation of model-predictive algorithm. Then model-predictive control signals are obtained and they are transferred to the system. In literature, model-predictive control algorithm was designed by separating into two parts. Actuator lengths were used instead of Cartesian quantities so as to keep the platform inside maximum limits of workspace [23].

In this study, a desktop flight simulation system is introduced in following section with its components including flight simulation software, the manipulator and the motion cueing system. Human motion perception is summarized. Inverse/forward kinematics of the system and their real time solution is explained. Communication problem solution between the simulation and the platform is addressed and lastly, the results from the hardware is discussed.

## 2 Flight Simulators and Components

Parallel manipulator used as simulator has 6 DoF including 3 translational and 3 rotational motion which was firstly designed in 1965 by D. Stewart mounting top and bottom plates and connecting with six parallel legs that are moving linearly, [24].

Flight simulator components are as follows:

A. Simulated vehicle
- Visual feedback system
- Vehicle control system

B. Motion simulator
- Actuators and drivers
- Sensors

C. Motion perception software
- Motion cueing algorithm
- Human perception model

### 2.1 Flight simulation software

Simulators have visual interface on which the simulation of aircraft motions is performed. In this study Flightgear flight simulation software is used. Flightgear is an open-source software that has several aircraft models. Commands given by the pilot through motion controller are transferred to the platform and tracked visually. To transfer the pilot commands into the simulation software, an advanced flight vehicle control device (Thrustmaster Hotas Warthog) is used in this study, Fig.2(b).

### 2.2 Motion Simulator

The system is being structured on a 6x6 parallel desktop manipulator and is driven by linear DC motors (Linmot PS01-23X80 / 0150-1201). Linear motors have the motion capacity of 110 mm and multi-axis motion control of the motors is performed via Linmot E210-VF model motor driver.

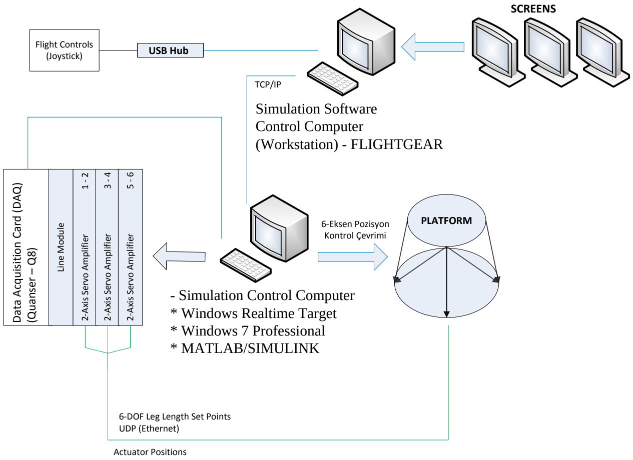
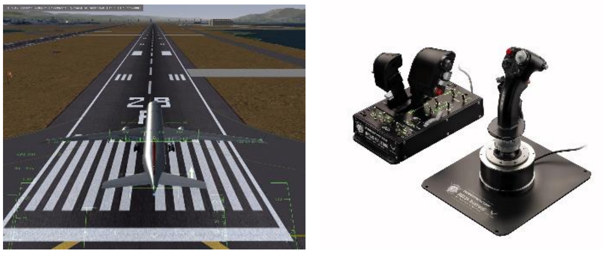

SCREENS
Flight Controls (Joystick)
USB Hub
TCP/IP
Simulation Software Control Computer (Workstation) - FLIGHTGEAR
Data Acquisition Card (DAQ) (Quanser – Q8)
Line Module
2-Axis Servo Amplifier
1 - 2
2-Axis Servo Amplifier
3 - 4
2-Axis Servo Amplifier
5 - 6
6-Eksen Pozisyon Kontrol Çevrimi
PLATFORM
- Simulation Control Computer
* Windows Realtime Target
* Windows 7 Professional
* MATLAB/SIMULINK
6-DOF Leg Length Set Points
UDP (Ethernet)
Actuator Positions

Fig. 1 Simulator system block diagram



(a) (b)

Fig. 2 (a) Flight simulation software, (b) Joystick (Thrustmaster Hotas Warthog)

Table 1 Parameters of 6x6 Parallel Manipulator Mechanism

| | | |
|---|---|---|
| Top plate weight | $M_u$ | 1.387 kg |
| Motor shaft weight | $m_u$ | 0.135 kg |
| Motor body weight | $m_d$ | 0.44 kg |
| Top plate radius | $r_p$ | 0.15 m |
| Bottom plate radius | $r_b$ | 0.175 m |
| Damping ratios on joints | $c_f$ | 0.03 Nm.s/deg |
| Motor inductance | $L_a$ | 1.5 mH |
| Motor resistance | $R_a$ | 10 ohm |
| Motor force coefficient | $K_t$ | 11 N/A |

Because the internal control mechanisms of motor drivers are not active in force mode, this mode is used on the platform system. Position data are obtained via internal linear encoders of the motors and sensitivity of 10 µm is preferred.

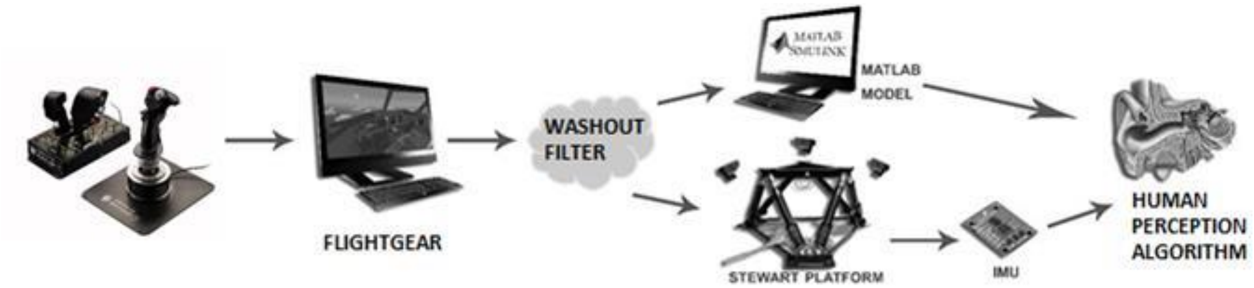


Fig. 3 Representation of experimental system for motion perception based on sub-components

## 2.3 Motion Cueing

In flight simulators, all linear actuators are driven simultaneously in order to take the 6 DoF (x, y, z – roll, pitch, yaw) parallel manipulator to the desired position. Required positions of actuators are recalculated at each cycle and position error of the legs is kept at minimum level. The major issue in flight simulators is that the simulation software has infinite space and platform has a limited one. “Washout Filter” used in Motion Cueing Algorithm eliminates this issue. Motion cueing algorithm constraints position data that platform receives, and most importantly, attempts to move the platform to its initial position under the driver’s perception of motion level.

Motion perception algorithm that consists of high-pass and low-pass filters is applied in order to reduce the translational acceleration and angular velocity values obtained from Flightgear flight simulation software via communication protocol to an acceptable limits of the platform workspace. Translational accelerations are filtered by high-frequency data and noise using 3rd order (1st order high-pass washout filter and 2ns order low-pass washout filter) and angular velocities are filtered from high-frequency data and noise by 2nd order washout filter.

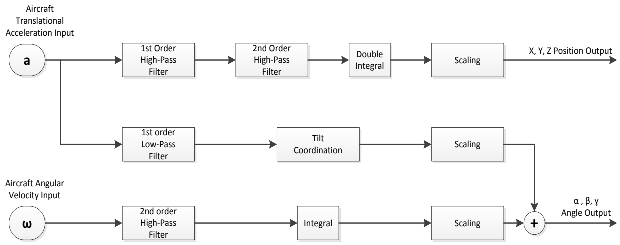

Aircraft Translational Acceleration Input
a
1st Order High-Pass Filter
2nd Order High-Pass Filter
Double Integral
Scaling
X, Y, Z Position Output
1st order Low-Pass Filter
Tilt Coordination
Scaling
Aircraft Angular Velocity Input
ω
2nd order High-Pass Filter
Integral
Scaling
+
α , β, γ Angle Output

Fig. 4 Motion perception structure for flight simulator

As shown in Fig. 4, filtered signals are transmitted to motor drivers by taking the double integrals of translational acceleration inputs and single integral of angular velocity inputs and by creating position information via inverse kinematics of the platform.

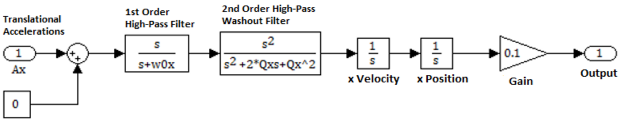

Translational Accelerations
1
Ax
+
+
0
1st Order High-Pass Filter
s
s+w0x
2nd Order High-Pass Washout Filter
s2
s2 +2*Qxs+Qx^2
1
s
x Velocity
1
s
x Position
0.1
Gain
1
Output

Fig. 5 Motion cueing realization for x-axis motion

Platform velocity is kept lower than human perception level during washout process. Thus the pilot does not feel the washout motion presence. Combination of movements creates a real flight feeling. Classical algorithm has a simple structure but due to its constant parameters it cannot improve itself.

### 2.3.1 Human Motion Perception

Human perception system constitutes the basis of human perception studies. Transfer functions between sensed angular velocity and the input angular velocities at each semi-circular canal with stimulus acceleration are as follows and $\tau_1, \tau_2, \tau_a$ , $\tau_L$ are the time constants and $G_{sc}$ is the gain of the system.

$$W_{S_i}(s) = \frac{\widehat{\omega}_i(s)}{\omega_i(s)} = \frac{\mathrm{G}_{sc}\tau_1\tau_a s^2(1+\tau_L s)}{(1+\tau_a s)(1+\tau_1 s)(1+\tau_2 s)} \qquad (1)$$

In the transfer function between translational acceleration sensed in ear and stimulus acceleration, $1/\tau_L$ is named as $a_0$, $1/\tau_1$ is named as $b_0$ and $1/\tau_2$ is named as $b_1$ by using time constant coefficients of otoliths. $K_{oto}$ is the gain.

$$W_{O_i}(s) = \frac{\hat{a}_i(s)}{a_i(s)} = \frac{\mathrm{K}_{oto}(s+a_0)}{(s+b_0)(s+b_1)} \qquad (2)$$

Index $i$ shows the rotational motions in Euler angles (Roll – Pitch – Yaw).

A human detects the accelerations and angular motion through vestibular system in inner ear. Because of the insensitivity in low speeds, motion perception algorithms simulates the acceleration of the platform and attempts to return the platform to its initial position at these unsensed low speeds. When high frequency motion signals move the platform, low frequency motion signals are transformed into angular motion. The major issue in this motion is the loss of low-frequency content of the acceleration. To eliminate this issue, the idea of tilt coordination was developed. The gravitational acceleration is $g = 9.78033\ m/s^2$ and this is the z component of the gravity vector. If an amount of roll motion is given as input to the simulator, x component of gravity vector can be adjusted to the level of pilot motion perception.

Table 2 Human Motion Perception System Coefficients

| | Chen[24] | | | Houck ve Telban [19] | | |
|---|---|---|---|---|---|---|
| | 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 | - |
| $K_{oto}(s^{-1})$ | 1,50 | 1,50 | 1,50 | 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 |

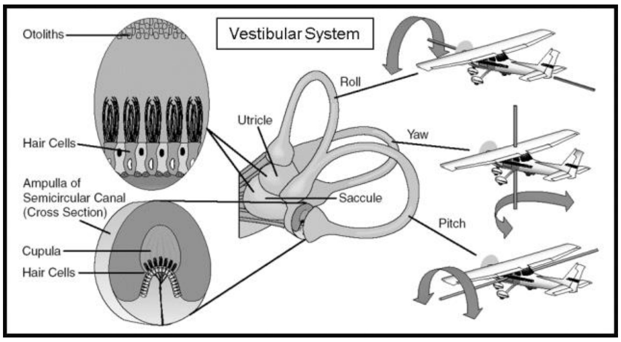


Fig. 6 Relationship between vestibular system and human perception

## 3 Kinematic Analysis of 6 DoF Parallel Mechanism

As in serial robots, kinematics analysis of parallel mechanisms may also examined in terms of inverse and forward kinematics. Inverse kinematics analysis allows to find the leg lengths give the position and rotation of the system $(L_1, L_2, L_3, L_4, L_5, L_6)$ when the position and rotation of top plate (x, y, z, α, β, γ) is known. Forward kinematics is the finding of the position and rotation of the top plate with respect to the fixed bottom platform when the leg lengths are known. Forward kinematics analysis is important in terms of all conditions platform may have in specific leg lengths. The contrast between serial and parallel robots continues its existence also here. When it is required to follow a specific trajectory in real time, it is necessary to solve the inverse kinematics in real time. According to several studies, it can be said that the forward kinematics solution of the parallel robots is not singular that is there are several available configurations for a leg length. In inverse kinematics, solution is singular which means that one position and one rotation of top plate may only occurs with one input.

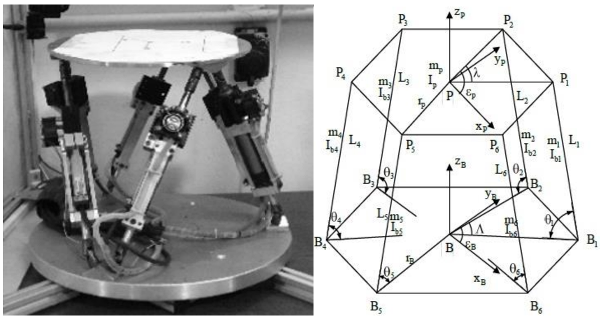


Fig. 7 Experimental system and modelling parallel manipulator

### 3.1 Inverse kinematics analysis

Inverse kinematics analysis for parallel mechanisms is easier than forward kinematics to solve. Inverse kinematics of a robot system can be used to plan a specified rotation [tk_dk – 2]. Leg lengths that provide any specified position of center of gravity of top plate or bottom plate used as end-effector.

{P} demonstrates the axes of center of gravity of the top plate and {B} demonstrates the axes of center of gravity of the bottom plate. Λ angles can be defined as angles between forthrights drawn from the origin of axes {B}to successive two corners of the fixed plate and λ's are the angles between forthrights drawn from the origin of axes {P} to successive two corners of the moving plate.

$$
\begin{aligned}
{}^{P}\vec{p}_i &= \left\{\vec{p}_{ix}\,, \vec{p}_{iy}\,, \vec{p}_{iz}\right\}^T \\
&= \left\{r_p.\cos(\lambda_i)\,, r_p.\sin(\lambda_i)\,, 0\right\}^T \quad (1)
\end{aligned}
$$

$$
\begin{aligned}
{}^{P}\vec{b}_i &= \left\{\vec{b}_{ix}\,, \vec{b}_{iy}\,, \vec{b}_{iz}\right\}^T \\
&= \{r_b.\cos(\Lambda_i)\,, r_b.\sin(\Lambda_i)\,, 0\}^T \quad (2)
\end{aligned}
$$

$$
{}^{B}\vec{S}_i =- {}^{B}\vec{b}_i + {}^{B}\vec{t} + {}^{B}\vec{p}_i \qquad (3)
$$

Position of points $P_i$ are defined by (1) equation and position of $B_i$ points are defined by (2) equation. Leg lengths can be obtained by calculating the magnitude of the $\vec{S}_i$ vector and the open form of this vector is shown in (3) equation. $\vec{t}$ is the translational vector here. Rotation of the moving platform {B} with respect to the coordinate system is as follows:

$$ {}^{B}\vec{p}_i = {}^{B}_{P}\mathbf{R}\,{}^{P}\vec{p}_i \qquad (4)$$

$$ l_i = \sqrt{S_{ix}^2 + S_{iy}^2 + S_{iz}^2} \qquad (5)$$

**R** is the rotation matrix of the top plate with respect to {B} axes. By integrating (3) into (5), leg lengths are being calculated according to translation and rotation of top plate. After related regulations, the final equation is obtained as follows:

$$ l^2 = x^2 + y^2 + z^2 + {r_p}^2 + {r_b}^2 + \ldots $$

$$ 2\left(r_{11}p_{ix} + r_{12}p_{iy}\right)(x - b_{ix})2\left(r_{21}p_{ix} + r_{22}p_{iy}\right)\ldots $$

$$ \left(y - b_{iy}\right) + 2\left(r_{31}p_{ix} + r_{32}p_{iy}\right)(z) \ldots \qquad (6)$$

$$ -2\left(xb_{ix} + yb_{iy}\right) = f $$

### 3.2 Forward kinematics analysis

There are several approaches upon forward kinematics solution in literature. In this study, Newton–Raphson method which is one of numerical analysis method is used. In this method, an equation in the form of F(x) = 0 can be solved as follows:

In this equation $x_0$ is considered as starting point and if F(x) has a derivative at $x_0$, the tangent of the function at related point can be expressed as follows:

$$ Y = \dot{F}(x_0).(x - x_0) + F(x_0) \qquad (6)$$

Next $x_1$ can be found by replacing $x_1$ with x in the equation and zero with *Y*.

$$ 0 = \dot{F}(x_0).(x_1 - x_0) + F(x_0) \qquad (7)$$

$$ x_1 = x_0 - \frac{F(x_0)}{\dot{F}(x_0)} \qquad (8)$$

$x_1$ in (8) equation will be closer to the solution. To achieve more absolute solution in same method, iterations are applied and solutions of $x_1$, $x_2$, $x_3$ ... $x_n$ can be obtained. When the desired solution sensitivity is reached iterations are stopped.

### 3.3 Implementation of forward kinematics

For the forward kinematics solution of Stewart platform mechanism, Newton-Raphson method is applied as follows:

$$ \mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \qquad (9)$$

R is the rotation matrix and the equation that will give leg length is defined with (10) equation. In (10) equation, x, y, z are translation values of the moving platform. $r_p$ is radius of the circle passing through the joint points of top plate, $r_b$ is radius of the circle passing through the joint points of bottom plate. $p_{ix}$ , $p_{iy}$ , $p_{iz}$ are coordinates of junctions to top plate. $b_{ix}$, $b_{iy}$, $b_{iz}$ are coordinates of junctions to bottom plate. When unknowns are taken as (x, y, z, α, β, γ) and $l_i$ is passed to the right side of the equation and the equation is equal to zero, the function that will be used in Newton-Raphson is obtained.

$$ F_i(X) = x^2 + y^2 + z^2 + {r_p}^2 + {r_b}^2 + 2\left(r_{11}p_{ix} + r_{12}p_{iy}\right)(x - b_{ix}) + \qquad (10)$$

$$ 2\left(r_{21}p_{ix} + r_{22}p_{iy}\right)\left(y - b_{iy}\right) + $$

$$ 2\left(r_{31}p_{ix} + r_{32}p_{iy}\right)(z) - 2\left(xb_{ix} + yb_{iy}\right) - {l_i}^2 = 0 $$

$$ \mathbf{X} = x, y, z, \alpha, \beta, \gamma \qquad i = 1,2,..6 $$

Because of the function contains more than one variable, the derivative of the function used in Newton-Raphson can be found by taking partial derivative of each variable as follows:

$$ \mathbf{J}_{(ij)} = \frac{dF_i}{dX_j}(X) \quad \forall\{i,j\}\,\epsilon\,\{1,..6\} \qquad (11)$$

Then unknowns are found by the following equation:

$$ \mathbf{X}^{(n+1)} = \mathbf{X}^{(n)} - \left[\mathbf{J} \cdot \left(\mathbf{X}^{(n)}\right)\right]^{-1} \cdot F\left(\mathbf{X}^{(n)}\right) \qquad (12)$$

As initial conditions, following values should be given:

$$ \mathbf{X}^{(0)} = (x_0, y_0, z_0, \alpha_0, \beta_0, \gamma_0) \qquad (13)$$

Iterations are applied to approach the equation to correct result. When approaching the solution, to stop the iterations, an $\varepsilon$ value should be defined as follows:

$$ |F_i(x)| \leq \varepsilon \qquad (14)$$

When (14) equation is confirmed iteration is stopped and last x values are considered as the solution.

The flowchart of forward kinematics analysis is as follows:

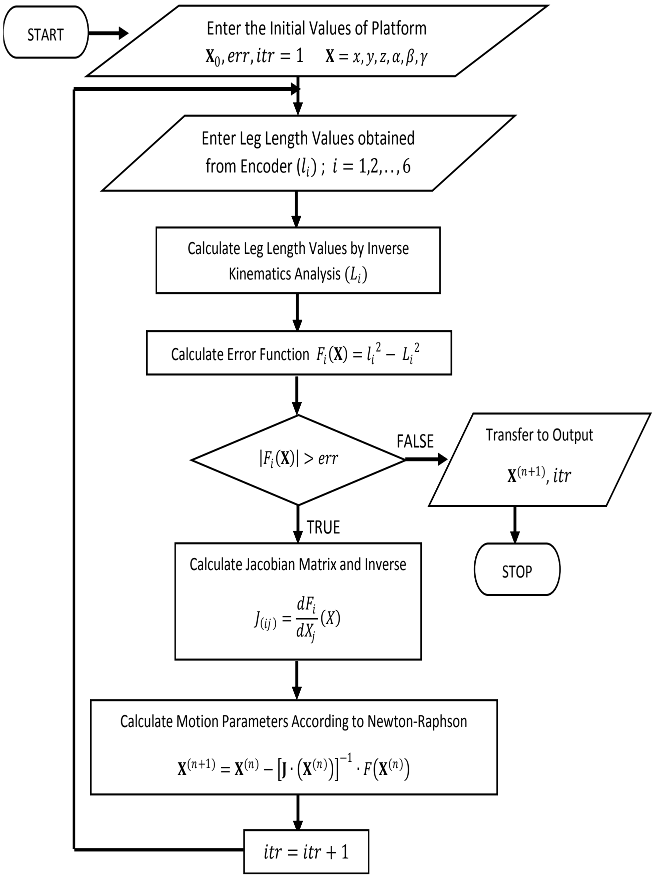

START
Enter the Initial Values of Platform
$\mathbf{X}_0, err, itr = 1 \quad \mathbf{X} = x, y, z, \alpha, \beta, \gamma$
Enter Leg Length Values obtained from Encoder $(l_i)$ ; $i = 1,2,..,6$
Calculate Leg Length Values by Inverse Kinematics Analysis $(L_i)$
Calculate Error Function $F_i(\mathbf{X}) = l_i^2 - L_i^2$
$|F_i(\mathbf{X})| > err$
FALSE
TRUE
Transfer to Output
$\mathbf{X}^{(n+1)}, itr$
STOP
Calculate Jacobian Matrix and Inverse
$J_{(ij)} = \frac{dF_i}{dX_j}(X)$
Calculate Motion Parameters According to Newton-Raphson
$\mathbf{X}^{(n+1)} = \mathbf{X}^{(n)} - [\mathbf{J} \cdot (\mathbf{X}^{(n)})]^{-1} \cdot F(\mathbf{X}^{(n)})$
$itr = itr + 1$

Fig. 8 Flowchart of forward kinematics analysis

## 4. COMMUNICATION, SOFTWARE AND CONTROL

Flight data is fed to the simulator and data are transmitted into the motion control computer for further processing.

### 4.1 Communication and Software

Mathematical models that convert the joystick controls to the motion on the aircraft have run into the visual simulation software. These mathematical models produce motion parameters with analysis. Basically, these motion parameters are transferred via these mathematical models and the platform is supplied. “Generic Protocol” is used on the input/output unit in the visual simulation software. Parameters produced by the mathematical models in flight simulation software via Generic Protocol are transmitted as packages and delivered into the platform control computer via TCP/IP protocol. During this process, data to be transmitted are defined by creating an XML script file. Thus, 6 parameters including translational accelerations of aircraft at pilot position and angular velocities, are transmitted into an external port.

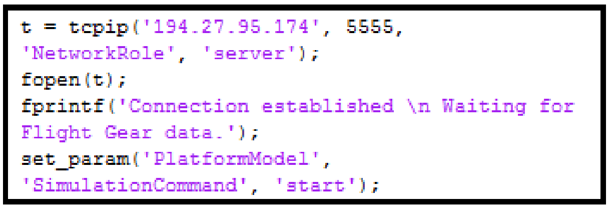
Created XML script should be kept into the setup folders of visual simulation software and should be run here with software simultaneously. The script is as follows:

```
<PropertyList>
 <generic>
  <output>
   <line_separator>newline</line_separator>
   <var_separator> : </var_separator>
   <chunk>
    <name>zaman (sec)</name>
    <type>float</type>
    <format>%.2f</format>
    <node>/sim/time/elapsed-sec</node>
   </chunk>
   <chunk>
    <name>roll angular velocity (deg/sec)</name>
    <type>float</type>
    <format>%.2f</format>
    <node>/orientation/roll-rate-degps</node>
    <factor>0.3048</factor> <!--feet to meter -->…
 </chunk>
```

```
generic=socket,out,120,174.26.78.124,5555,tcp,mypr
otocol.xml
```

Fig. 9 XML script file and last expression

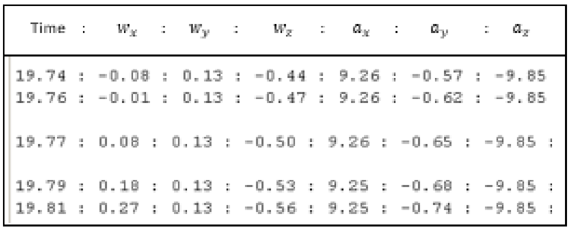
It is determined that sample frequency of 240 Hz is sufficient for simulation. Thus, with respect to sampling frequency > 2 · $f_{max}$ due to the Nyquist theorem, visual simulation software frequency was calculated as 120 Hz. Data obtained from the simulation are transmitted to the defined port of motion control computer via TCP/IP protocol. To receive incoming data in Matlab environment, a server in an m-File is created and it is provided to observe incoming data.



```
t = tcpip('194.27.95.174', 5555,
'NetworkRole', 'server');
fopen(t);
fprintf('Connection established \n Waiting for
Flight Gear data.');
set_param('PlatformModel',
'SimulationCommand', 'start');
```

Fig. 10 TCP/IP Protocol connection codes

Table 3 Flight Gear sample data output



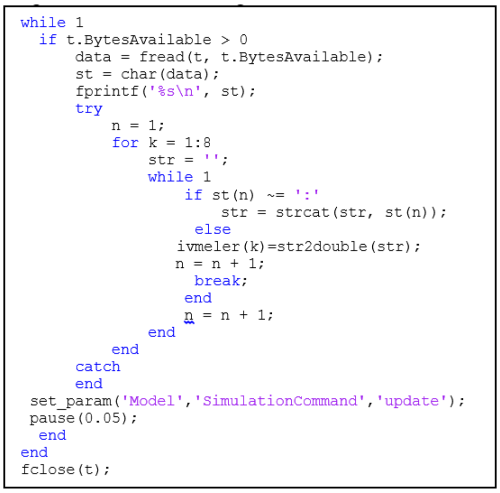
```
 Time  :   Wx   :  Wy   :   Wz   :  ax   :   ay   :  az

19.74 : -0.08 : 0.13 : -0.44 : 9.26 : -0.57 : -9.85
19.76 : -0.01 : 0.13 : -0.47 : 9.26 : -0.62 : -9.85

19.77 : 0.08 : 0.13 : -0.50 : 9.26 : -0.65 : -9.85 :

19.79 : 0.18 : 0.13 : -0.53 : 9.25 : -0.68 : -9.85 :
19.81 : 0.27 : 0.13 : -0.56 : 9.25 : -0.74 : -9.85 :
```

At this point, incoming data should be transformed into a processable format in Matlab environment. This process is shown in Figure 11. Incoming data obtained as a result of the transformation process are firstly extracted then are assigned into an array by making the transformation process. Data of this array are transferred into the simulator model in Simulink environment to be given to the algorithm input in real time as aircraft motion parameters.



```
while 1
  if t.BytesAvailable > 0
      data = fread(t, t.BytesAvailable);
      st = char(data);
      fprintf('%s\n', st);
      try
          n = 1;
          for k = 1:8
              str = '';
              while 1
                  if st(n) ~= ':'
                      str = strcat(str, st(n));
                   else
                 ivmeler(k)=str2double(str);
                 n = n + 1;
                   break;
                  end
                  n = n + 1;
              end
          end
      catch
      end
 set_param('Model','SimulationCommand','update');
 pause(0.05);
  end
end
fclose(t);
```

Fig. 11 Format change code

### 4.2 Control

Platform control consists of two stages. Linear actuator driver control and platform mathematical model control. For control of linear actuators via driver configurations, the algorithm shown in Figure 13 is used. The output of the PID controller shown in Figure 12 is seen as analog input. This input is included in the algorithm with a current gain. This gain parameter is assigned as catalog value of 3A (Ampere) depending on supply voltage. This input relationship is as shown in Figure 14.

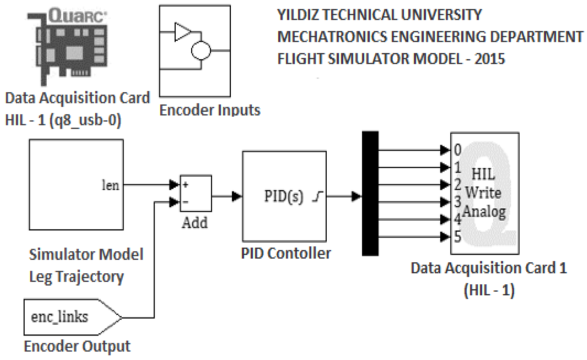


Fig. 12 Simulink model of the simulator

## 5 Conclusion

Simulation tests were performed on the desktop platform system. Boeing 777-300ER model aircraft is prepared for departure from a specified airport. Acceleration effect of the aircraft during departure is simulated as an amount of motion in the x-axis direction (Figure 15-a).

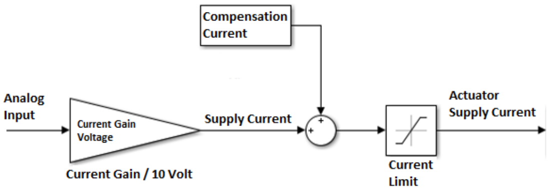


Fig. 13 Linear actuator driver control model

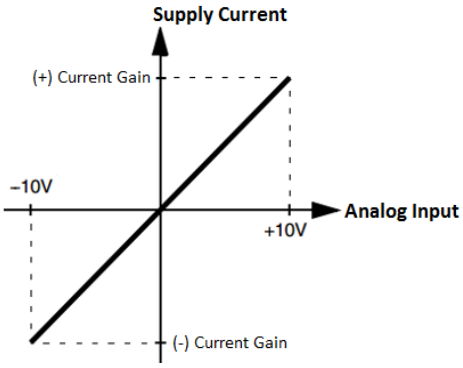


Fig. 14 Relationship between analog input and supply current

Table 4 Platform mathematical model controller coefficients

| PID CONTROLLER COEFFICIENTS | | | |
|---|---|---|---|
| $K_p$ | $K_I$ | $K_D$ | *Filter* |
| 600 | 23 | 8 | 115 |

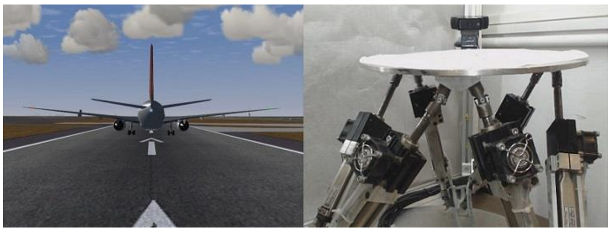


Fig. 15-a Motion at 20. Second

The aircraft moved through x axis with 4 m/s$^2$ acceleration and the platform moved 0.08 meter to the x direction. Also platform rolled about y axis with 8°, because of tilt the effect. The results are shown in Figure 15-b.

In second case, aircraft performed rolling motion. Aircraft moved about x axis through CCW direction -5° alpha angle (Fig. 16-a). Then, aircraft moved about clockwise direction with 7° alpha angle (Fig. 16-b). Results are shown in Figure 16-c.

In third case, aircraft dived for 10° about y axis (Fig 17-a.) and lifted nose 15° about y axis (Fig 17-b.). Results can be shown in Figure 17-c. In this motion, platform moved through z axis for perceived g-force of aircraft. Platform moved -0.05 meters when aircraft dived. Secondly, platform moved 0.1 meters to z direction when aircraft lifted nose.

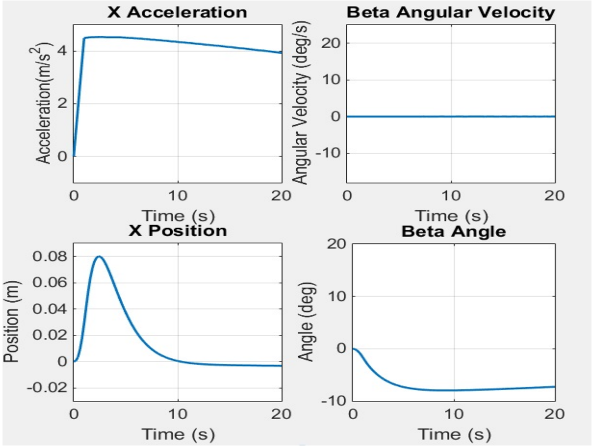


Fig. 15-b Aircraft acceleration and platform motion.

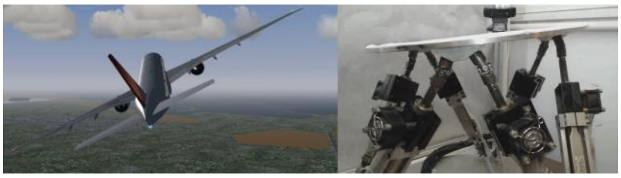


Fig. 16-a 52. second of flight, rolling about counter clockwise.

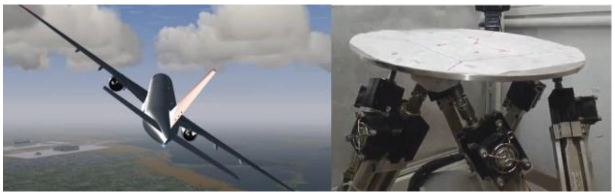


Fig. 16-b Motion at 56. second

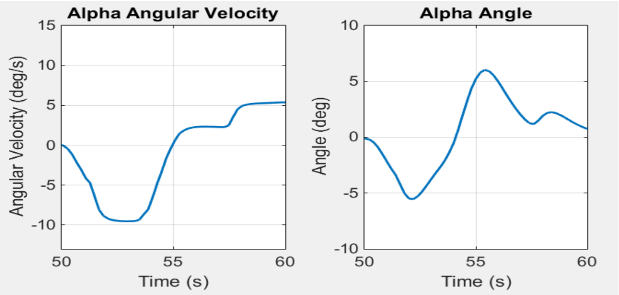


Fig. 16-c Angular velocity of aircraft and rolling performance of platform.

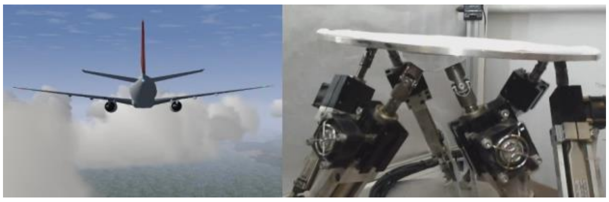


Fig. 17-a Motion at 167. second

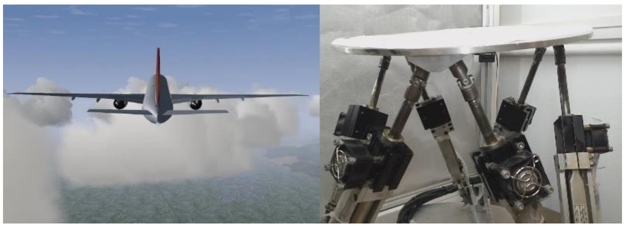


Fig. 17-b Motion at 169. second

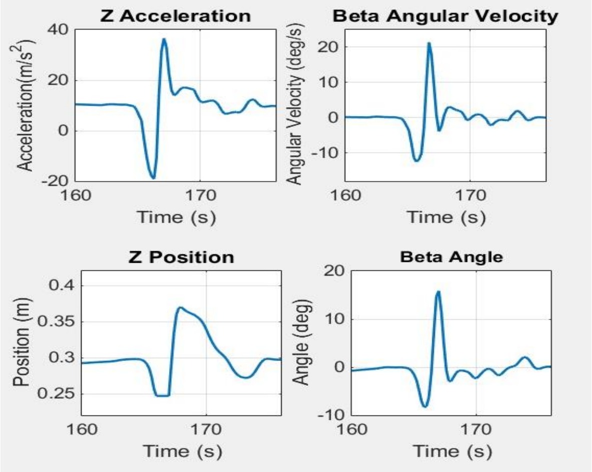


Fig. 17-c Diving and lifting nose accelerations of aircraft and motions of platform

The results of the simulation on desktop simulator are considered to be satisfactory. As a further analysis, pilot perception of motion is to be analyzed using the vestibular model provided and forward kinematics of the platform. Since the results obtained are based on classical washout algorithm, further investigation is to be performed using extensive algorithms like adaptive, optimal and model predictive washout structures.

*Acknowledgement*

This work is supported by the Ministry of Science, Industry and Technology of Turkey under grant 0563.STZ.2013-2. Provided information in the article does not reflect the idea of the Ministry.

*References:*

[1] H. Sehammer, M. Fischer "Motion Cueing for 3-6 and 8 Degrees of Freedom Motion Systems", Swedish National Road and Transport Research Institute, 581 95 Linköping, Sweden, 2010.

[2] M. Aminzadeh, A. Mahmoodi, "Optimal MC Algorithm Using Motion System Kinematics" European Journal of Control, 2012.

[3] Z. Lei, J. Hongzhou, "PC Based High Quality and Low Cost Flight Simulator" International Conference on Automation and Logistics, Jinan, China, August 18 - 21, 2007.

[4] A. Hamish, J. Jamson, “Motion Cueing in Driving Simulators for Research Applications” The University of Leeds Institute for Transport Studies, November 2010.

[5] G. L. Zacharias, "Motion Cue Models for Pilot-Vehicle Analysis" Bolt Beranek and Newman Inc Cambrıdge Ma Contr. Syst. dept, May 78.

[6] R. J. Telban, W. Wu, F.M. Cardullo, *Motion cueing algorithm development* [microform] : *initial investigation and redesign of the algorithms*, National Aeronautics and Space Administration, Langley Research Center; National Technical Information Service, 2000.

[7] T. Robert, C. Frank, and H. Jacob, "A Nonlinear, Human-Centered Approach to Motion Cueing with a Neurocomputing Solver," in AIAA Modelling and Simulation Technologies Conference and Exhibit, ed: American Institute of Aeronautics and Astronautics, 2002.

[8] B. Ü. Hastanesi, *Vestibüler Sistem ve Bozuklukları,* 2012.

[9] F. Colombet, D. Paillot, “Visual Scale Factor for Speed Perception”, Journal of Computing and Information Science in Engineering, December 2011.

[10] L. Reid, A. Nahon, “Simulator Motion-Drive Algorithms: A Designer's Perspective”, The University of Toronto, VOL. 13, NO. 2 , 1990.

[11] M. A. Nahon and L. D. Reid, "Simulator motion-drive algorithms - A designer's perspective," *Journal of Guidance Control and Dynamics*, vol. 13, pp. 356-362, 1990.

[12] G. Reymond, A. Kemeny, “Motion Cueing in the Renault Driving Simulator”, International Journal of Vehicle Mechanics and Mobility, August 2000.

[13] H. Sehammer, M. Fischer “Motion Cueing for 3-6 and 8 Degrees of Freedom Motion Systems”, Swedish National Road and Transport Research Institute, 581 95 Linköping, Sweden, 2010.

[14] Z. Fang, G. Reymond, “Performance Identification and Compensation of Simulator Motion Cueing Delays”, RENAULT, Technical Center for Simulation, TCR AVA 0 13, 1, Avenue du golf – 78288 Guyancourt, France, Vol. 11, December 2011.

[15] M. Aminzadeh, A. Mahmoodi, “Optimal MC Algorithm Using Motion System Kinematics” European Journal of Control, 2012.

[16] L. D. Reid and M. A. Nahon, *Flight simulation motion-base drive algorithms* [Toronto, Ont.]: Institute for Aerospace Studies, University of Toronto, 1985.

[17] R. Sivan, J. Ish-Shalom, and J.-K. Huang, "An Optimal Control Approach to the Design of Moving Flight Simulators," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 12, pp. 818-827, 1982.

[18] M. Dagdelen, G. Reymond, A. Kemeny, M. Bordier, and N. Maïzi, "Model-based predictive motion cueing strategy for vehicle driving simulators," *Control Engineering Practice*, vol. 17, pp. 995-1003, 9, 2009.

[19] L. Nehaoua, H. Arioui, S. Espie and H. Mohellebi, “Motion Cueing Algorithms for Small Driving Simulator”, IEEE International Conference in Robotics and Automation (ICRA06), 2006.

[20] Y. Yang, S. Zheng, “Motion Drive Algorithm for Flight Simulator Based on the Stewart Platform Kinematics”, School of Mechatronics engineering, Harbin Institute of Technology, Harbin, 150080, China, Key Engineering Materials Vols. 460-461 (2011) pp 642-647, 2011.

[21] P. Chunping Aviation University of Air Force Changchun, China, “A Time Varying Washout Approach for Flight Simulation Hexapod Motion System” 2012.

[22] I. A. Qaisi, A. Traechtler, “Human in the Loop; Optimal Control of Driving Simulators and New Motion Quality Criterion”, University of Paderborn, Paderborn, Germany, 2012.

[23] M. Baseggio, A. Beghi, “An MPC Approach to the Design of Motion Cueing Algorithms for Driving Simulators”, 14th International IEEE Conference on Intelligent Transportation Systems Washington, DC, USA. October 5-7, 2011.

[24] V.E. Ömürlü, I. Yildiz, " Parallel self-tuning fuzzy PD+ PD controller for a Stewart–Gough platform-based spatial joystick" Arabian Journal for Science and Engineering, vol. 37/7, pp. 2089-2102, 2012.