April 26, 2017

Mohamed Al-Janabi

Shirwa Haji

Functional Specification

**Real Time Performance and Error Correction**

**of Middlesex 6DOF Platform (The Chair)**

**CCE4999**

[**1. Abstract**](#_sa94r7skwnb0) **2**

[**2. Background**](#_lcgv66tcu6v2) **2**

[**3. Introduction**](#_6uhgdl9dbydx) **4**

[**4. Functionalities / Usage**](#_xyvvnfrzd7tw) **5**

[4.1. Capturing Real Time Data](#_19dvqs4fk4bl) 5

[4.1.1. Initializing the Sensor Server](#_hh6x6eqyi0xo) 6

[4.1.2. Troubleshooting](#_c30t9e8nf6rb) 7

[4.2. Capturing Commanded Data](#_v2x0jbvrlvh) 8

[4.2.1. Initializing](#_uozhg88biu4f) 8

[4.2.2. Troubleshooting](#_geq4byk086r6) 9

[4.3. Measuring Accuracy of the System](#_q2kkd1q3prb6) 9

[4.3.1. Initializing the Monitor](#_cwad5sytdbcq) 11

[4.3.2. Understanding Output Data](#_66xwt4wtypv0) 11

[4.3.3. Increasing Accuracy](#_9mgqdtimdhr7) 12

[4.3.4. Troubleshooting](#_jd47hog3uss2) 12

[4.4. Viewing Stored Data](#_64o071as73mz) 13

[**5. Calibration**](#_b8o0gfnepna4) **13**

[5.1. Procedure](#_m3cwi2en97vt) 13

[**6. Troubleshooting**](#_vi73ylgwmz0g) **15**

[6.1. Actuator Fault](#_bl5wruq0m4ra) 15

[6.2. Wiring Issue](#_bqaj9zgri49y) 15

[6.3. Power Supply Issue](#_283e63dyc00) 15

[6.4. Sensor Configuration Issue](#_8fjvbvvmmx0x) 15

[6.5. Device Unresponsiveness](#_kq27ehcq3fn0) 16

[6.6. Intermodule Connectivity / Network Issue](#_d1bq3x6p1rkx) 16

[6.7. Calibration Issue](#_ai4m1sm5wb5a) 16

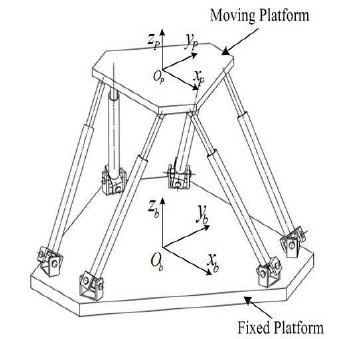
[**7. References**](#_wu2hkxv8nmnr) **17**

# 1. Abstract

This document describe the functional specifications for the hardware and software system that monitors the real time performance of the Middlesex 6DOF platform (The Chair) and also provides the capability to enhance the accuracy by error correction of existing software that controls the platform. The specifications define what the system does, the user-system relationships and how the end user will interact and use the system. This document acts as guide for system architecture and development. It also provides troubleshooting details including possible cause of errors and their solutions for various functionalities of the system.

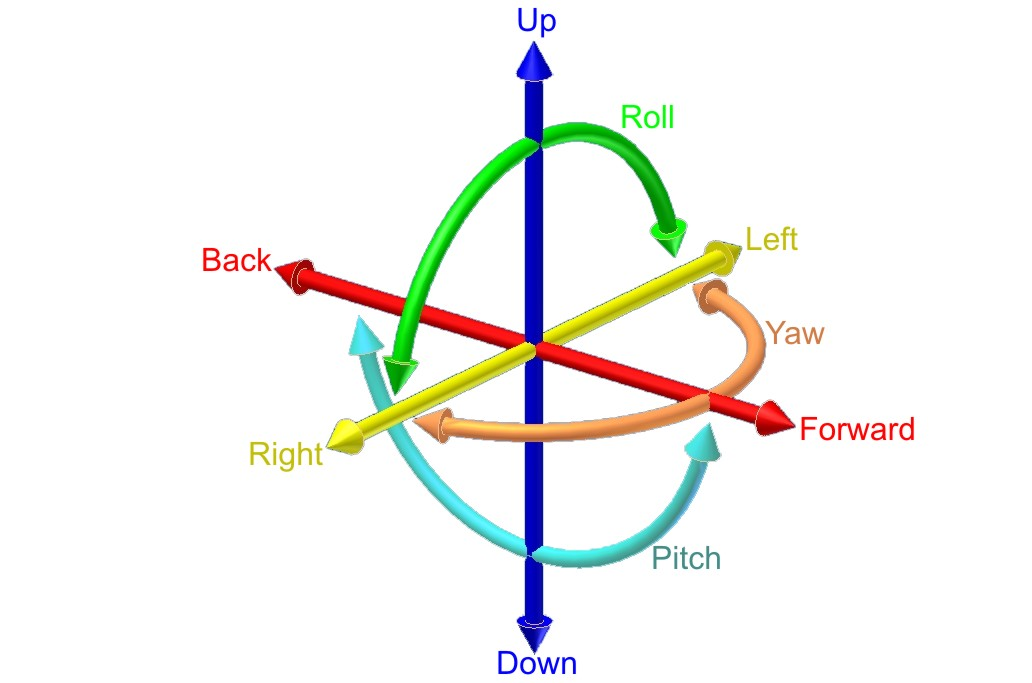
# 2. Background

The Chair is basically a 6DOF motion platform built using a robotic configuration called Stewart platform. It has two platforms, Fixed Platform and a Moving Platform which are attached together with six prismatic actuators, commonly hydraulic jacks or electric actuators. The moving platform can move in six degrees of freedom using six prismatic actuators as shown in the picture below.



**Figure 1. Stewart Platform [1]**

The six degrees include three linear movements x, y, z and three rotations yaw, pitch, roll.



**Figure 2. Six Degrees of Freedom [2]**

The actual picture of the chair is shown below.

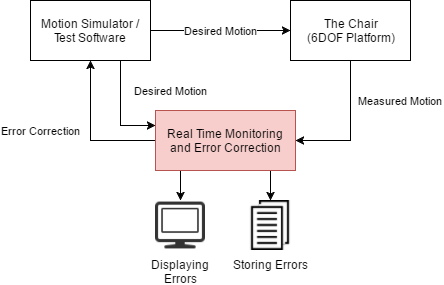


**Figure 3. The Chair**

The actuation of the chair is done by using 6 hydraulic jacks as seen in the picture.

# 3. Introduction

The purpose of this project is to design and develop a complete system including hardware and software that monitors the real-time performance of the Middlesex 6DOF Platform and provides error correction to the existing motion simulating software that controls the platform. The system will measure the actual movements of the platform by using a set of sensors and will compare the measured movements with the desired movements that are generated by the motion simulators. The Test Command Server that generates the dummy commands to test the functionality of the system is also the part of this project.



**Figure 4.. Block Diagram**

The system can be subdivided into 3 main functional blocks:

1. **Sensor Server**

The Sensor Server will read the actual position, both linear and angular motion, of the platform in real time and would make the data available for other modules to use it.

1. **Test Command Server**

The Test Command Server will generate some random commands for the desired motion that would test the system in all possible direction i.e. all linear and rotational directions.

1. **Monitor**

The Monitor will receive and compare between the actual motion and the desired motion and would display and store the errors in a meaningful way. It would also generate feedback data that can be used by the motion simulating softwares to minimize the errors.

Desired accuracy for static translational measurements is to the nearest millimeter and for angular measurements is to the nearest 1/2 degree. The orientation of the chair is updated with a delay of at least 2 seconds.

# 4. Functionalities / Usage

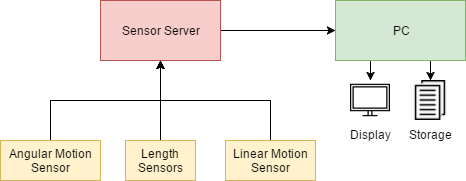
The system can be categorised in five functionalities that it provides:

1. Capturing Real Time Data
2. Capturing Commanded Data
3. Viewing Error
4. Viewing Stored Data
5. Unit Testing and Verification

The details of the above-mentioned functions of the system are described below:

## 4.1. Capturing Real Time Data

The capturing of the actual motion of the platform is handled by a module called Sensor Server. The Sensor Server will read the actual position and movement of the platform in real-time, at least every 30 msec, using a controller and a set of motion sensors. It will also display the real time data and store it in a file as comma separated values.



**Figure 5. Block Diagram of Sensor Server**

There are a total of 13 parameters that are be measured by the Sensor Server i.e.

**1x Unix Time (t)**

Unix time (also known as POSIX time or epoch time) is a system for describing instants in time, defined as the number of seconds that have elapsed since 00:00:00 Coordinated Universal Time (UTC), Thursday, 1 January 1970,[1][note 1] not counting leap seconds. [3]

For example, the time **04/18/2017 @ 5:38pm (UTC)** would be stored as **1492537088**

**6x Prismatic Arm Length (L1-L6)**

As there are six arms used in the platform, so the Sensor Server would measure and store the actual length of each arm individually in **mm**.

**3x Linear Motion (ax, ay, az)**

The linear motion of all the three axes would be stored in **m/s^2.**

**3x Angular Motion (gx, gy, gz)**

The angular motion of three axes would be stored in **rad/s^2**

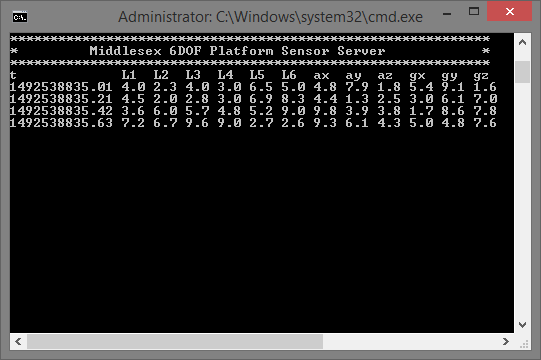
The format for displaying the actual data is illustrated below.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **t** | **L1** | **L2** | **L3** | **L4** | **L5** | **L6** | **ax** | **ay** | **az** | **gx** | **gy** | **gz** |
| 0 | 10 | 12 | 9 | 13 | 10 | 13 | 12 | 12 | 34 | 50 | 23 | 45 |
| **...** |  |  |  |  |  |  |  |  |  |  |  |  |

### 4.1.1. Initializing the Sensor Server

To start the Sensor Server, make sure that all sensors are connected to it and it is powered ON. Connect the Sensor Server with the PC and run the program SensorServer.exe.

Once the program is initialized successfully, it will immediately start displaying the sensor data according to the format described above. Once the program is stopped, it will dump all the data in a new file created in the same folder where the program is located. Make the platform move using Motion Simulator or Test Command Server and see the actual data change in real time.

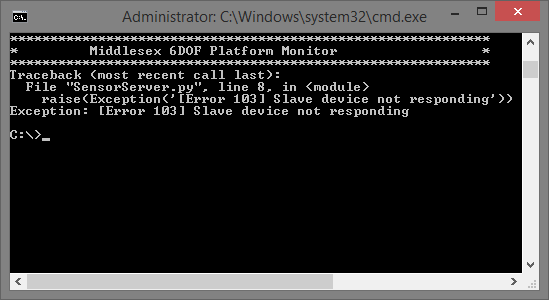


**Figure 6. Sensor Server - Real Time Data (Dummy)**

### 4.1.2. Troubleshooting

In case there is a problem in capturing or storing the real-time data, an error message will be generated on the display.

An example of the error message being displayed is shown below.



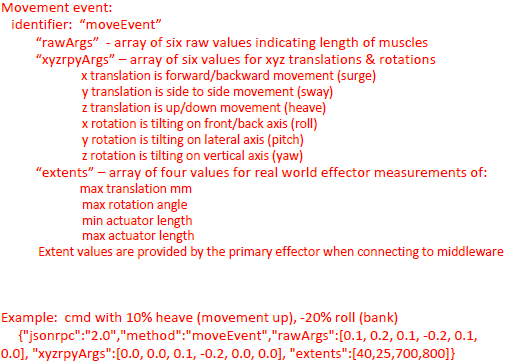
**Figure 7. Error format example**

Common error messages or problems and their solution are listed below.

|  |  |  |  |
| --- | --- | --- | --- |
| **Error Code** | **Error Message / Problem** | **Possible Cause** | **Solution** |
| 100 | No such port as xx | The PC is not able to detect the Sensor Server | Unplug and reconnect the Sensor Server with the PC |
| 101 | Could not connect with port | The Sensor Server is being used by another application in the PC | Close all other programs and try restarting the PC |
| 102 | Cannot write file | The storage file is being used by another program | Close all other programs that might be using the output file |
| - | Some of the actual values are not changing or stuck at zero | The corresponding sensor is disconnected from the Sensor Server | Verify all the sensors are connected to the Sensor Server and restart the program |
| - | Inconsistent sensor values or garbage values | The sensor is loosely connected or not configured properly | Replug the sensor and reconfigure its settings for e.g. baud rate. |
| 103 | Slave Device Not responding | Any of the slave device is either stuck or not connected with the PC | Replug all the slave devices and restart them |
| 104 | No Data Received from Slave Device | The Slave device is connected but the communication protocol is not configured | Make sure that the communication protocol and rate of transfer is synchronized between master and slave |

## 4.2. Capturing Commanded Data

The commanded data is served by the Middleware or by the Test Command Server. The module Monitor connects to either the Middleware or the Test Command Server to get the commanded data. The commanded data is received in a json format as described below.



### 4.2.1. Initializing

To start capturing the commanded data, start the program called CommandedData. Once the program is started successfully, it will connect with the Middleware or Test Command Server and display the commanded data on screen.

The data will be displayed as it is received i.e. in a json format as shown below.

{

“jsonrpc”: “20”,

“Method”: “moveEvent”,

“rawArgs”: [0.1, 0.2, 0.1, -0.2, 0.2, 0.3],

“xyzrpyArgs”: [0, 0, 0, 0, 0, 0],

“Extents”: [40, 20, 700, 800]

}

### 4.2.2. Troubleshooting

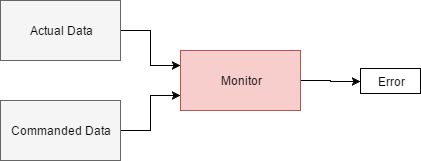
In case there is a problem in viewing the commanded data, an error message will be generated on the display.

Common error messages or problems and their solution are listed below.

|  |  |  |  |
| --- | --- | --- | --- |
| **Error Code** | **Error Message / Problem** | **Possible Cause** | **Solution** |
| 200 | Could not open port xx | The PC is not able to connect with the Middleware or Test Command Server | Make sure that the Middleware or the Test Command Server is running and the port number is configured properly |

## 4.3. Measuring Accuracy of the System

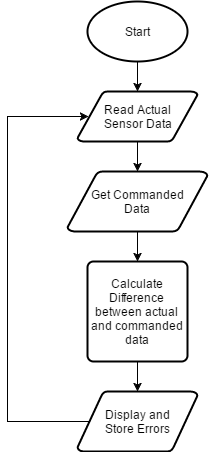
This main functionality of the system is to compare the actual data with the commanded data and calculate the error. This is handled by the module called Monitor. It connects with the Middleware and the Sensor Server, gets both the actual data and the commanded data and calculates the difference between them.



**Figure 8. Block Diagram for Monitor**

The format for displaying and storing the error data is same as the actual data except that it is a difference from the commanded data.

A high-level flow of the module Monitor is depicted below using a flow chart.

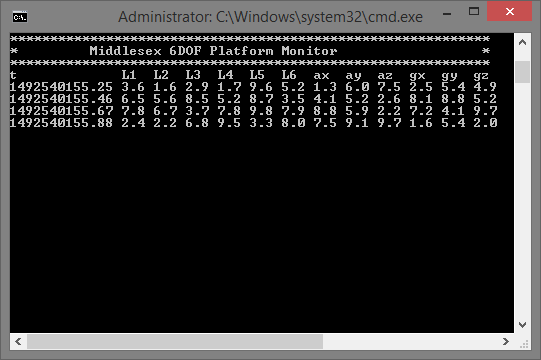


**Figure 9. Monitor Flow Chart**

### 4.3.1. Initializing the Monitor

To start calculating and viewing the error data, start the program called Monitor. Once it has initialized successfully, it will start capturing the actual data and the commanded data and then calculates the error in real time.

A sample output of what the program would look like if everything is working is shown below.



**Figure 10. Monitor - Displaying Errors in Actuator Lengths, Linear and Angular Motion**

### 4.3.2. Understanding Output Data

Each row displays the difference of 12 parameters between the commanded value and the measured value along with the timestamp t at which the error occurred. The closer theses values to zero, the better will be the accuracy of the system.

The 12 parameters and their units are described below.

|  |  |  |
| --- | --- | --- |
| **Parameter Symbol** | **Description** | **Unit** |
| t | timestamp | seconds |
| L1 | Error in actuator 1 | mm |
| L2 | Error in actuator 2 | mm |
| L3 | Error in actuator 3 | mm |
| L4 | Error in actuator 4 | mm |
| L5 | Error in actuator 5 | mm |
| L6 | Error in actuator 6 | mm |
| ax | Translational error in x-axis | mm |
| ay | Translational error in y-axis | mm |
| az | Translational error in z-axis | mm |
| gx | Rotational error in x-axis | deg |
| gy | Rotational error in y-axis | deg |
| gz | Rotational error in z-axis | deg |

### 4.3.3. Increasing Accuracy

Once the accuracy of the system is measured, it can be increased by minimizing the errors. The errors can be categorized in the following two categories.

1. **Actuator Length Errors**

These errors indicate the difference between the required length and the actual length of the actuators. These kind of errors occur can occur due to several reasons which are discussed below.

One possible reason for these errors is that the sensors, that are used to measure the actual length of the actuators, are not properly fixed in place. This could cause inconsistent values and hence large errors values. To minimize these errors, make sure that all the six sensors for measuring the actuator lengths are securely fixed in place.

Another reason for these kind of errors could be overloading the system with more weight than it can handle. To avoid such a situation, verify the loading limit of the actuators via their specification document, and make sure to keep the load under this limit.

1. **Linear and Angular Motion Errors**

These errors occur when there is a difference between the desired final motion of the chair and the actual motion including both linear and angular motion.

To minimize these kind of errors, make sure that the motion sensor is perfectly aligned and secured with the chair, otherwise it would generate persistent errors.

### 4.3.4. Troubleshooting

In case there is a problem in calculating the difference between the commanded data and the actual data, an error message will be generated on the display.

Common error messages or problems and their solution are listed below.

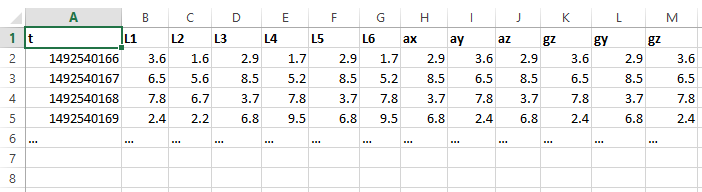
|  |  |  |  |
| --- | --- | --- | --- |
| **Error Code** | **Error Message / Problem** | **Possible Cause** | **Solution** |
| 300 | Data format mismatch | The data received from the Middleware or the Sensor Server got corrupted | Recheck all the connections and restart the program |
| 301 | Cannot Connect with the Middleware / Test Client | The Middleware is not running or the ports are not configured properly | Make sure the Middleware is running and the port numbers are synchronised |

## 4.4. Viewing Stored Data

The data stored by Monitor is in a simple comma separated values format. The file generated in the same folder where the program Monitor is located. This file can be easily opened with MS Excel or even in text editor. The format for storing the data is the same as described previously.

Filename format: stored-data-yyyy-mm-dd-HH-MM.csv

E.g stored-data-2017-04-20-20-34.csv



**Figure 11. Displaying Stored Data - Error Data**

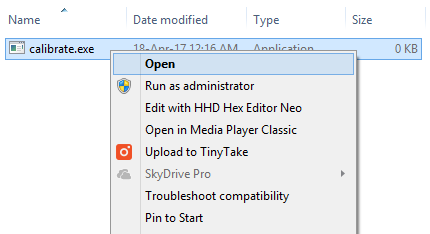
# 5. Calibration

The system has to be calibrated every time before using it whenever the external conditions change. The response of the system gets affected by the weight of the person sitting on the chair and also due to the minute differences in the response of the actuators. The position of all the sensors cannot be precisely the same so calibration is required to make the software adjust itself with the external conditions.

## 5.1. Procedure

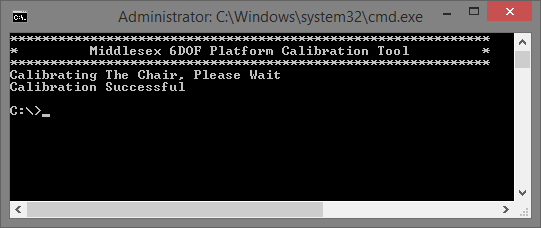
The steps for calibrating the chair for a specific external conditions are listed below.

1. Make sure all the sensors are attached and securely fixed in their places.
2. Turn on the Sensor Server and make sure it is connected with the PC.
3. Remove any load from the chair and make sure that it is levelled with the ground.
4. Load the chair with the required weight for which you want to calibrate the system. For example, if you want to calibrate the system for a person whose weight is 60 Kg, load the chair with 60 Kg or let that person sit on the chair without holding anything else.
5. Locate the file calibrate.exe on the PC and execute it.



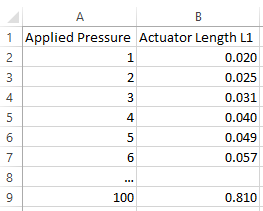
**Figure 12. Executing calibrate.exe**

1. Once executed successfully, it would indicate that the calibration was done successfully as shown below



**Figure 13. Calibration Successful Output**

1. The calibration tool will slowly move the chair by changing the actuator pressure in small increments while reading the measured length of the actuator. Once the calibration is finished, it would generate a csv file for each actuator containing a table that shows the relationship between applied pressure and actual length. This calibration data would be then used by the middleware to move the actuators precisely.



**Figure 13. Calibration Data for Actuator 1 (Dummy Data) - calibration\_actuator\_1.csv**

1. In case the calibration is failed, please see the section 6 for common possible issues and their solutions.

# 6. Troubleshooting

Some of the likely issues that may be encountered are discussed below.

## 6.1. Actuator Fault

It is possible that one or few of the actuators starts lagging and does not respond as the others. This could happen due to wear or due to physical damage to the actuator.

Re-run the calibration to resolve minor differences between the actuators.

## 6.2. Wiring Issue

As the platform moves with high acceleration and makes jerks so it is likely that the wiring of the sensor or the controller goes loose. To avoid such a situation, keeps the wires and the modules tightly secured with their location.

## 6.3. Power Supply Issue

Using long wires to provide power can result in voltage drop that is much significant with low voltage systems. If the power lights of the modules starts to get dim during operation, it's time to use thicker wires for supplying power.

## 6.4. Sensor Configuration Issue

Each sensor has a specific location to which it belongs. If you see that moving the chair in one direction does not change its corresponding sensor value, re-verify the location of all the sensors.

## 6.5. Device Unresponsiveness

If any of the slave or master device gets stuck or does not send data, try restarting the device. If it does not help, reprogram the device to make sure that the device is functional.

## 6.6. Intermodule Connectivity / Network Issue

There are several software modules that communicate with each other using sockets. If any of the module is unable to communicate with other modules, verify the port numbers for both the server and the client. If the problem is still there, try closing all programs and restart the PC.

## 6.7. Calibration Issue

If the system is not as responsive as before after changing the environment or changing the weight of the person sitting on the chair, try calibrating the software again as mentioned in section 4.0.

# 7. References

|  |  |
| --- | --- |
| [1] | <https://www.researchgate.net/figure/264725216_fig1_Fig-1-Generalised-Stewart-platform-manipulator-motion-consists-of-longitudinal> |
| [2] | <https://en.wikipedia.org/wiki/Six_degrees_of_freedom> |
| [3] | <https://en.wikipedia.org/wiki/Unix_time> |
|  |  |