Stewart Platform Project

Deliverable One – Conceptual Project Design

SWE 6823 – Embedded Systems

Author, and sole group member:

Marion Garrett Sisk
KSU ID: 000942002

February 21, 2021

Table of Contents

Project Description **2**

Hardware Architecture **3**

Hardware Implementation Tools 5

Software Architecture **5**

Software Implementation Tools 6

Preliminary Bill of Materials **6**

Deliverables **7**

Schedule and Timeline **8**

References **9**

Project Description

This project centers around the design, and development of a custom Stewart Platform. A Stewart Platform is a stable platform that enables motion with six degrees of freedom within the physical limits of the system. This project will encompass the design and implementation of both the hardware and software aspects of the system using systems and software engineering practices to ensure the process is adequately documented, and traceable. The use of novel methodologies and domain specific techniques will be commonplace.

The project will have several deliverables, as defined by the project requirements provided on the course's D2L home page. These are discussed in detail in the *Deliverables* section of this document.

The primary technical goal of this project is to create a platform that can maintain a fully stable position, given both translational and rotational motion in three dimensions. As part of a cursory search regarding common implementations of this type of system, the concept of the Stewart Platform was found. There have been many examples of Stewart Platforms produced, both in an industrial environment, as well as an amateur/hobbyist environment. For instance, Stewart Platforms are often used in high-fidelity flight simulators to provide for actual motion within the simulator itself. In figure 1 below, a large example is shown. It is a flight simulator in use by the German airline, Lufthansa which utilizes a Stewart platform for translational and rotational motion.

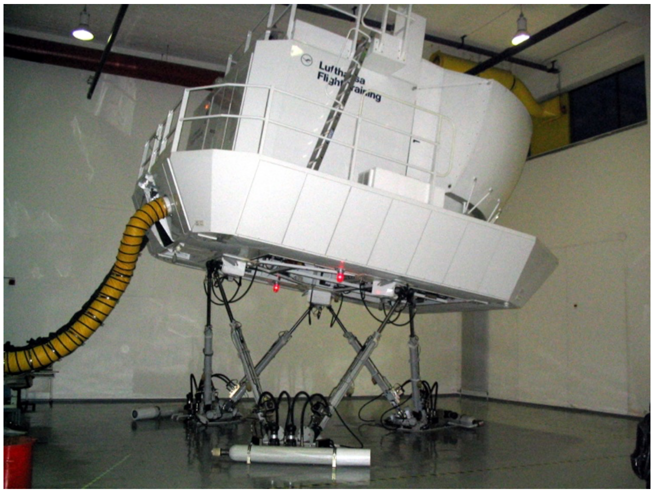

Figure 1: Stewart Platform supporting flight simulator.

By Ethan Arnold, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=8864535

This project, however, is much smaller in scale. It will use inexpensive, off the shelf components, as well as custom 3D printed structural elements. These will help keep the costs low, but also require some creativity when implementing the details. Additional details regarding the hardware and software architectures are given in their respective sections within this document.

Hardware Architecture

For the platform to provide for a stable platform that will absorb/counter any perturbation to the base of the system, it must have a means to measure the movement of the platform itself. For this function, an inertial measurement unit (IMU) will be utilized. There are several small IMU integrated circuits available for any combination of microcontroller units (MCU's) and embedded platforms. That being said, the MCU environment must also be selected. For this, the Arduino MEGA2560 will be utilized as the MCU. This particular MCU has been chosen for several reasons, the first of which is that I currently have several of these development boards on hand already that have been used in other projects in some capacity. Additionally, the Arduino programming environment has a rich support community and wide support for all manner of sensors and protocols through the use of libraries, IMU's being among these. A specific IMU has not been chosen as yet.

Utilizing an Arduino MEGA and an IMU will provide all the input and logical capability needed for a feedback system, but it must also use actuators in order to affect the physical world. The project will include the use of rotational actuators and linkages, instead of the linear actuators seen in Figure 1. This will reduce the cost substantially, at the expense of some slightly more complicated kinematics. Rotational actuators include stepper motors or servo motors. Given the need for precision over power, servo motors would be a better choice. They have the added benefit of being cheaper at smaller sizes as well. There is also a large selection of linkage components that are commonly used with servo motors. These have the luxury of having very mature library support within the Arduino environment, but also low cost and easily obtainable off-the-shelf (OTS) controllers and other required components. In Figure 2 the hardware components are arranged with the proper information shown being transmitted between them. In this diagram, the IMU, MCU, and servo systems are shown interacting with external perturbations. For the platform to remain stable, these must all act to minimize the perturbations. Some details regarding the logic behind this process is given in the software architecture section.

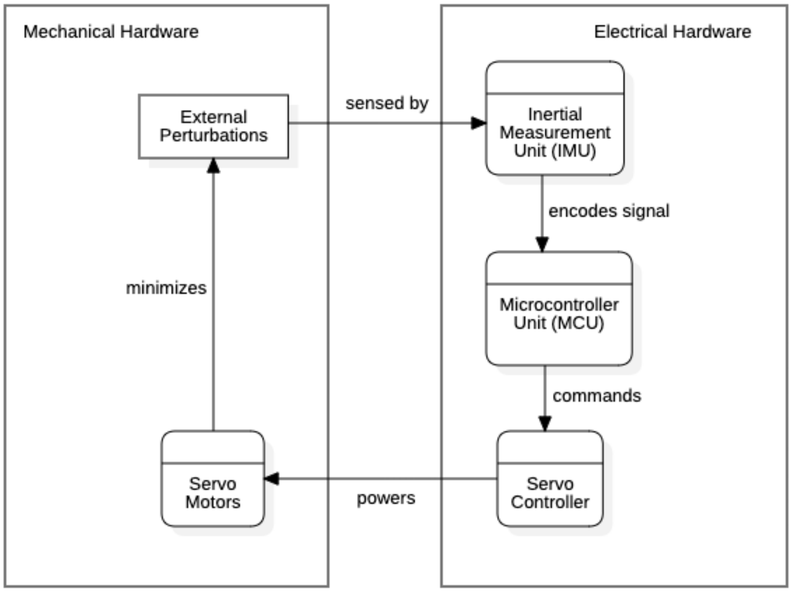

Figure 2: Top-Level Hardware Architecture Diagram. Note that the software logic occurs exclusively in the MCU.

Lastly, there must be some kind of frame in order to hold everything together. For this, the use of SolidWorks and a Form3 3D printer will permit the design and fabrication of a bespoke solution to properly hold all of the electronic, and mechanical components as an integrated device. The detail design behind this element has not been completed as yet. However, assets like dimensional drawings of Arduino boards and servos have been collected and will help expedite the completion of this portion of the design. The entire system will be powered by a DC power supply and be able to run standalone under its own internal control and power source. An AC adapter will be chosen, as having integrated batteries will add considerably to the complexity of the whole system.

Hardware Implementation Tools

All necessary hardware tools are currently on-hand, so an actual device being constructed is within the scope of this project and is a part of the schedule. Hardware implementation tools will include the following (this is currently a non-exhaustive list):

- Form3 3D Printer and associated resin
- SolidWorks 3D Modeling Software
- SolidWorks PDM Version Control
- KiCAD Electrical Design Software
- Standard electrical hand tools
- Molex connectors and associated crimp tools
- Soldering iron, rosin-core solder, and prototyping boards

Software Architecture

As stated in the previous section, all of the control logic is contained within the microcontroller unit (MCU). Since the Arduino MEGA 2560 has been chosen as the MCU for use in this project, the software will be developed within the Arduino IDE environment. There also exists the possibility of completing this control logic using MATLAB and Simulink's Embedded Coder feature as there is sufficiently robust support in Simulink for MCU boards within the Arduino environment. For a continuous system like the Stewart Platform, and coupled with other control system tools within the MATLAB ecosystem, more advanced models could be developed with relative ease. Both approaches are still being considered.

The way in which the software is to be architected centers around the separation of sensing, control, and actuation logic. The diagram given in Figure 3 shows the high-level software architecture diagram. Specific implementation details have yet to be completed.

Software Implementation Tools

Tools used to complete the software implementation portion of the project include, but are not limited to, the items on the following list:

- Arduino IDE
- MATLAB/Simulink and Embedded Coder
- StarUML
- GitHub Version Control

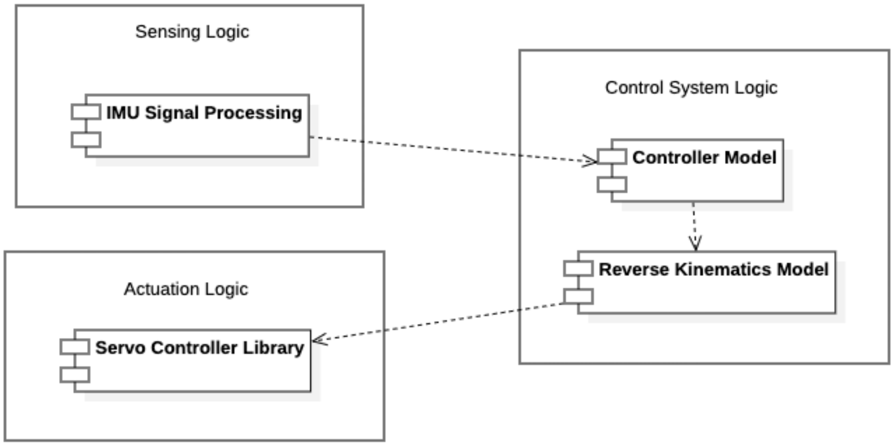

Figure 3: High-Level Software Architecture Diagram

Preliminary Bill of Materials

A preliminary bill of materials is presented below, however, until the hardware detail design has been completed in SolidWorks, a comprehensive list that includes fasteners, cabling, and all required motion linkages, will be missing. The list presented encompasses the high-level components necessary to implement the hardware and software architectures described previously in this document.

Quantity	Part Name	Subsystem
6	MicroRC Servo Motor	Mechanical
TBD	Associated Servo Motion Linkages	Mechanical
TBD	Screws and other Fasteners	Mechanical
1	Form3 Gray Resin Cartridge	Mechanical
1	Arduino MEGA 2560 MCU Board	Electrical
1	Servo Controller Arduino Shield	Electrical
1	Arduino Compatible IMU Board	Electrical
1	AC-DC Adapter	Electrical
TBD	Cabling and Associated Connectors	Electrical

Figure 4: Preliminary Bill of Materials

Deliverables

The deliverables associated with this project will parallel those required of the course itself. They are listed below with brief descriptions for each one. Each of these is then associated with a milestone in the next section, Schedule and Timeline.

- Deliverable 0
  - An initial statement with a project proposal
- Deliverable 1
  - The initial conceptual design and project description (this document)
- Deliverable 2
  - The finalized detail design and system integration
- Deliverable 3
  - A full initial demonstration of Stewart Platform
- Deliverable 4
  - Comprehensive report and Demonstration

Schedule and Timeline

Task #	Description
1.0	**Project Planning**
1.1	Set Up Jira Project
1.2	Create Project Roadmap
1.3	Write Project Proposal
1.4	Planning Milestone Review
1.5	Submit Project Proposal **(Deliverable 0)**
2.0	**Research and Resource Collection**
2.1	Choose appropriate MCU
2.2	Research kinematics transformations
2.3	Collect potential suppliers
2.4	Resource Milestone Review
3.0	**Software System Architecture**
3.1	Define System Architecture **(Deliverable 1)**
3.2	Create System Models
3.3	Software Architecture Review
4.0	**Mechanical System Architecture**
4.1	Define Mechanical Architecture **(Deliverable 1)**
4.2	Define Electrical Architecture **(Deliverable 1)**
4.3	Model Main Assembly
4.4	Model Electrical Systems
4.5	Mechanical Architecture Review
5.0	**Prototype Assembly**
5.1	Fabricate Parts
5.2	Fabricate Wiring Harness(es)
5.3	Fabricate Electrical Components
5.4	Total System Integration **(Deliverable 2)**
5.5	Wire Management and Organization
5.6	Assembly Milestone Review
6.0	**Software Implementation**
6.1	Interface Implementation
6.2	Kinematics Implementation
6.3	Unit Testing **(Deliverable 2)**
7.0	**Testing and Validation**
7.1	Commanded Tests **(Deliverable 3)**
7.2	Reactive Kinematics Tests
7.3	Saturation Tests
8.0	**Final Demonstration and Presentation**
8.1	Presentation Creation
8.2	Record Demonstration and Presentation **(Deliverable 4)**
8.3	Compile Project Report **(Deliverable 4)**

References

Introduction: PID Controller Design;

https://ctms.engin.umich.edu/CTMS/index.php?example=Introduction&section=ControlPID

Stewart Platform;

https://en.wikipedia.org/wiki/Stewart_platform

Many linked blog posts with tag “Stewart Platform”;

https://hackaday.com/tag/stewart-platform/

Stewart Platform;

https://www.instructables.com/Stewart-Platform/