

MECHTRON 4TB6 -

System Requirements

Dhruv Aggarwal

Gurkarmjit Kooner

Joseph Moolaseril

Auda Rab

Ahmed Belal

# 

# 

# **Purpose**

The purpose of this document is to outline the system requirements for a small-scale racing simulator robot. It will describe the expectations that the Next Generation Solutions team has for the implementation of this system and will be referenced throughout this project’s development.

# **Objective**

The objective of the project is to build a mechanical system which will interact with a racing simulator. The system will be capable of adjusting the orientation of a platform to mirror the orientation of a car within the computer game. These movements include positional changes and angular rotations. The project will involve the application of control systems and real time systems knowledge.

# **Scope**

The intent of the project is to have a development team of 5 Mechatronics engineering students apply engineering principles and theory to create a miniature physical video game simulator. The system will be able to translate a user’s real time in-game decisions into mechanical movements that a person in a car would feel. The system interfaces with a computer which will be running the video game for simulation.

In scope

1. Adjusting the physical system based on video game outputs
2. Communicating between the simulator software and a microcontroller
3. Communicating between a video game and simulator software
4. At the end of operation the system will move to home position
5. The platform will simulate forces in the x,y,and z directions
6. The platform will simulate rotation in the x,y, and z directions

Out of scope

1. Running videogames on a phone application
2. Creating a full scale (human sized) simulator
3. Creating a videogame or the simulation software
4. Creating a stand alone system for the simulation

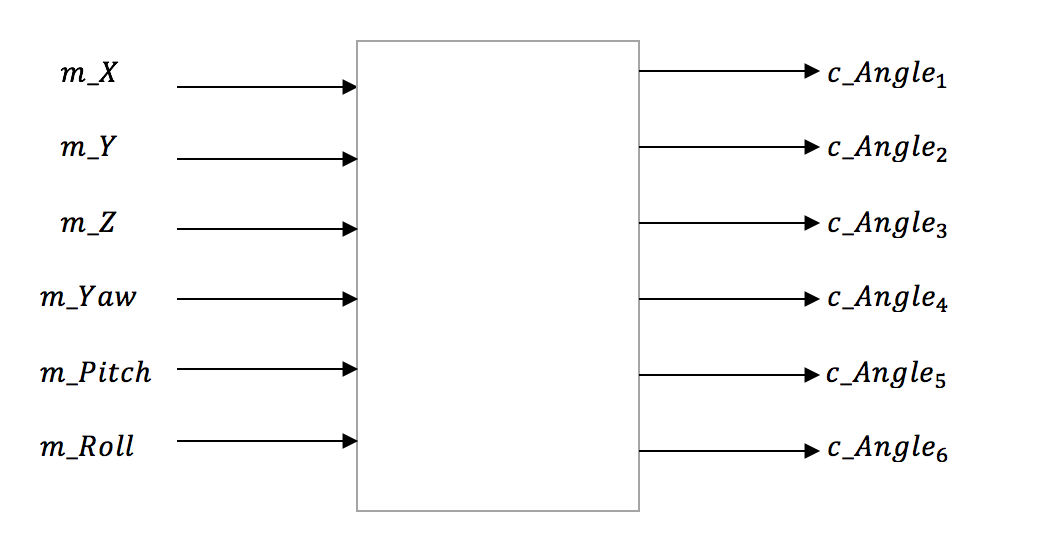
# 

# 

# **Standard Notation**

|  |  |
| --- | --- |
| **Notation** | **Description** |
| Video Game | The racing computer game that will be used to simulate a race car. |
| Platform | The flat surface which will be rotating/translating in sync with the car within the video game. |
| Mechanical System | The set of mechanical joints and parts which will drive the motion of the platform. |
| Simulation Software | The software which will extract the values of motion from the video game. |

# **Context Diagram**



|  |  |  |
| --- | --- | --- |
| Variable | Unit | Description |
| m\_X | Newtons | Force on a person in a car in the x-axis (latitudinal direction) |
| m\_Y | Newtons | Force on a person in a car in the y-axis (longitudinal direction) |
| m\_Z | Newtons | Force on a person in a car in the z-axis (vertical direction) |
| m\_Pitch | Degrees | Rotation of the vehicle in the x-axis |
| m\_Roll | Degrees | Rotation of the vehicle in the y-axis |
| m\_Yaw | Degrees | Rotation of the vehicle in the z-axis |

|  |  |  |
| --- | --- | --- |
| Variable | Unit | Description |
| c\_angle1 | Degrees | Translation of the platform in the x direction |
| c\_angle2 | Degrees | Translation of the platform in the y direction |
| c\_angle3 | Degrees | Translation of the platform in the z direction |
| c\_angle4 | Degrees | Rotation of the platform in the x direction |
| c\_angle5 | Degrees | Rotation of the platform in the y direction |
| c\_angle6 | Degrees | Rotation of the platform in the z direction |

|  |  |  |
| --- | --- | --- |
| Variable | Unit | Description |
| k\_X\_Max  k\_X\_Min | cm | Upper and lower limits for translation in the x-axis |
| k\_Y\_Max  k\_Y\_Min | cm | Upper and lower limits for translation in the y-axis |
| k\_Z\_Max  k\_Z\_Min | cm | Upper and lower limits for translation in the z-axis |
| k\_Pitch\_Max  k\_Pitch\_Min | Degrees | Upper and lower limits for rotation about the x-axis |
| k\_Roll\_Max  k\_Roll\_Min | Degrees | Upper and lower limits for rotation about the y-axis |
| k\_Yaw\_Max  k\_Yaw\_Min | Degrees | Upper and lower limits for rotation about the z-axis |

# **Behaviour Overview**

This concept was formulated from group discussion and once formalized, pitched to Dr. Wassyng. After discussion, this project was approved for the Computing and Software Department Capstone Project. Over the course of this document the system will be referred to as an autonomous system and the object being adjusted will be referred to as a car. The car imitates the forces being carried out in the real-time video game being played.

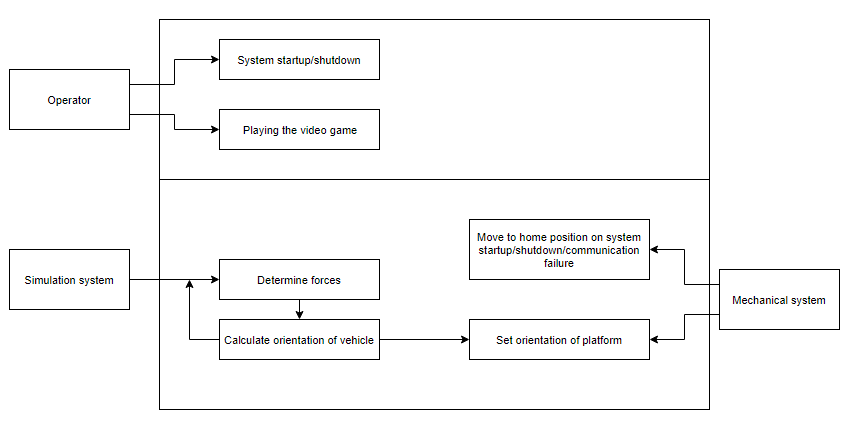
# 

# 

# **Required Behaviour Description**

# 

|  |  |
| --- | --- |
| **State** | **Description** |
| Race Started | The system is turned on and the user is currently playing a race within the video game on the computer. |
| Determine Forces | The simulator will extract force values based on user inputs in the game being played. For each movement that the player makes with their vehicle, a corresponding force is determined by the simulator. |
| Send Orientation Values | The force values determined by the simulator are sent from the computer to the mechanical system. |
| Set mechanical system orientation to the received values | If all orientation values (translation in x/y/z directions, pitch, roll, and yaw) are within the mechanical system limits, set the mechanical system orientation to match the received values. |
| Set mechanical system orientation to the limits | If any of the orientation values (translation in x/y/z directions, pitch, roll, and yaw) exceed the mechanical system limits, set that value to its limit, as it is not physically possible to go beyond this. |



* The rectangles within the square represent the system's reactions to differing external factors.
* The top rectangle represents user-determined states:
  + The user is responsible for powering up and shutting down the system.
  + The user is also responsible for playing the game which drives the system
* The bottom rectangle represents the system’s reactions to external factors not controlled by the user

# **Normal Operation**

## Description

In a normal scenario, the system must be able to communicate with the simulation software to determine the current orientation of the car within the video game. After receiving this orientation, the mechanical system will adjust itself to match the orientation it receives from the software.

## Normal Use Cases

### System is on and the user is participating in a race within the video game

When the user is racing within the video game and the simulation software is successfully communicating with the mechanical system, the mechanical system will move a platform based on the user’s inputs that affect the in-game vehicle’s movements.

### System is on, but the user is NOT participating in a race within the video game

When the user is not actively racing within the video game (for example, if the race has ended, the video game is paused, or if the user is in the main menu of the game) the mechanical system will not communicate with the simulation software and will return to a flat resting position until a race is started.

## Undesired Event Handling

### The system cannot communicate with the simulation software

If the system cannot communicate with the simulation software, the platform will return to a flat resting position until it can reestablish the communication.

# **Requirements**

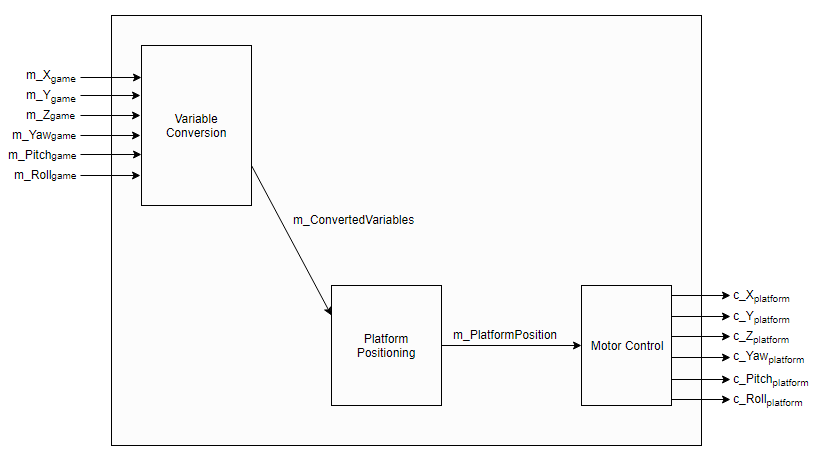
## Functional Requirements

1. The system shall accurately simulate forces in x,y, and z directions.
2. The system shall accurately simulate rotations about x,y, and z.
3. The system will move the platform based on the data passed from the simulation software.
4. The data passed from the simulation software must be converted to angles
5. The simulation software must be compatible with the game selected
6. The simulation software must be able to extract the desired variables
7. The system will operate in real time
8. The simulation software will communicate with the microcontroller
9. The laptop must be able to run the simulation software and the game at the same time
10. When the system is turned off it will return to the home position

## Non-Functional requirements

1. The device must be portable.
2. The system must be ready by the agreed upon deadline date
3. The device shall not exceed the allotted budget
4. The device must be easy to use.

## Functional Diagram



­­

List of Requirements that are likely to change

* The system will be portable not including the laptop

The dimensions of the system is a requirement that can potentially change due to possible changes in the system design.

List of Requirements that are not going to change

* The system shall accurately simulate forces in x,y, and z directions.
* The system shall accurately simulate rotations about x,y, and z.
* The system will move the platform based on the data passed from the simulation software
* The data passed from the simulation software must be converted to angles
* The simulation software must be compatible with the game selected
* The simulation software must be able to extract the desired variables
* The system will operate in real time
* The simulation software will communicate with the microcontroller
* The laptop must be able to run the simulation software and the game at the same time
* When the system is turned off it will return to the home position

The above functional requirements will not change as these are the fundamental requirements for the system to behave as required