**System Design Document**

**For**

NASA Vestibular Chair

Team member:

Brandon Boyle-Fagan

Matthaeus Gebauer

Miles Osborne

Dylan Prothro

Noah Reid Kent

Wilson

|  |  |
| --- | --- |
| Version/Author | Date |
| 1 | 10/6/2022 |
|  |  |
|  |  |
|  |  |

TABLE OF cONTENT

[1 INTRODUCTION 3](#_Toc116055615)

[1.1 Purpose and Scope 3](#_Toc116055616)

[1.2 Project Executive Summary 3](#_Toc116055617)

[1.2.1 System Overview 3](#_Toc116055618)

[1.2.2 Design Constraints 3](#_Toc116055619)

[1.2.3 Future Contingencies 3](#_Toc116055620)

[1.3 Document Organization 3](#_Toc116055621)

[1.4 Project References 4](#_Toc116055622)

[2 SYSTEM ARCHITECTURE 4](#_Toc116055623)

[2.1 System Hardware Architecture 4](#_Toc116055624)

[2.2 System Software Architecture 4](#_Toc116055625)

[2.3 Internal Communications Architecture 4](#_Toc116055626)

[3 HUMAN-MACHINE INTERFACE 4](#_Toc116055627)

[3.1 Inputs 4](#_Toc116055628)

[3.2 Outputs 5](#_Toc116055629)

[4 DETAILED DESIGN 5](#_Toc116055630)

[4.1 Hardware Detailed Design 5](#_Toc116055631)

[4.2 Software Detailed Design 6](#_Toc116055632)

[5 EXTERNAL INTERFACES 6](#_Toc116055633)

[5.1 Interface Architecture 6](#_Toc116055634)

[5.2 Interface Detailed Design 7](#_Toc116055635)

[6 SYSTEM INTEGRITY CONTROLS 7](#_Toc116055636)

System Design Document

# INTRODUCTION

## Purpose and Scope

The goal of the NASA Vestibular Chair project is to restore the basic hardware functionality to the chair and provide it with a new controller. This consists of ensuring the system can reach a specified RPM, hold that specified RPM for a set duration and allow the servos to gradually slow down to idle. If time permits, we will begin integrating more modern software and hardware tools to improve the "quality of life" features of the chair. This includes a web interface, custom test profiles/sequences, and ability to read and store sensor data from the chair.

## Project Executive Summary

This section provides an overview of the NASA Vestibular Chair project from a macro perspective, showing the framework with which the system design was conceived.

### System Overview

The NASA Vestibular Chair system consists of the hardware component of the chair itself, which is connected to a controller composed of a motor controller and other components to be able to handle digital and analog inputs for the chair motor. An additional web interface is planned to be added to the controller to allow for a more precise measurement of the input for the chair. The hardware components of the chair consist of a tachometer, motor, and the actual chair itself as well as some other pins set up for other once used analog measurements.

### Design Constraints

The development team hopes to maintain the most amount of the original internal hardware as possible. With this in mind, one of the major constraints in design of the controller of the chair and its interface is to have the new components work with the older technology present in the chair.

### Future Contingencies

Future contingencies include the inclusion of a digital part of the controller to account for any analog issues, with the intent to keep an analog and digital component to control the inputs of the chair.

## Document Organization

This System Design Document is organized into six sections. The introduction section explaining the basics of the design, followed by the document going over the system architecture, the human-machine interface, the detailed design, and external interfaces.

## Project References

*No references at this time.*

# SYSTEM ARCHITECTURE

The system uses an interface of hardware and software to control the movement of the NASA Vestibular Chair, as well as measure it’s speed and other readings.

## System Hardware Architecture

The composition of the hardware is represented by

* NASA chair
  + The NASA chair consists of inner hardware such as its motor and sensors already installed in the device such as a tachometer, which will be used to measure the speed of the chair during operation. The plan is to keep the sensors that are already in the chair if they are still usable, however it is also on the table to replace them if necessary.
* Controller
  + The controller will consist of an internal motor controller as well as components to allow it to receive data from the chair, such as serial ports. The current plan is to use High-Power Simple Motor Controller G2 24v12 from Polulu brand. This motor controller has, upon initial inspection, seemed like one of the best choices for the overall scope of the project.

## System Software Architecture

The software is planned to be

## Internal Communications Architecture

In this section, describe the overall communications within the system; for example, LANs, buses, etc. Include the communications architecture(s) being implemented, such as X.25*,* Token Ring, etc. Provide a diagram depicting the communications path(s) between the system and subsystem modules. If appropriate, use subsections to address each architecture being employed.

**Note:** The diagrams should map to the FRD context diagrams.

# HUMAN-MACHINE INTERFACE

Human input will be necessary for the system as the input for the chair is needed to increase its voltage, and by extension its speed. This interaction will cause the user to be able to adjust the chair’s speed gradually through the controller device.

## Inputs

This section is a description of the input media used by the operator for providing information to the system; show a mapping to the high-level data flows described in Section 1 .2.1, System Overview. For example, data entry screens, optical character readers, bar scanners, etc. If appropriate, the input record types, file structures, and database structures provided in Section 3, File and Database Design, may be referenced. Include data element definitions, or refer to the data dictionary.

Provide the layout of all input data screens or graphical user interfaces (GUTs) (for example, windows). Provide a graphic representation of each interface. Define all data elements associated with each screen or GUI, or reference the data dictionary.

This section should contain edit criteria for the data elements, including specific values, range of values, mandatory/optional, alphanumeric values, and length. Also address data entry controls to prevent edit bypassing.

Discuss the miscellaneous messages associated with operator inputs, including the following:

* Copies of form(s) if the input data are keyed or scanned for data entry from printed forms
* Description of any access restrictions or security considerations
* Each transaction name, code, and definition, if the system is a transaction-based processing system

The inputs of the system will be given from the controller to the chair to allow movement of the chair or set up of testing cases for the voltage to be applied without needing to be adjusted by the user. These inputs are user generated, as the individual in control of the device sets the voltage that the chair is being given. The sensors used will also count as input, including things such as the tachometer and accelerometer.

## Outputs

The output response from the chair should be the tachometer reading to feed into the controller to control the speed via its feedback. This will allow the system to avoid reaching a speed that outpaces the scope of the controller. The other output is that given from the chair itself, which is its actual rotation given the input voltage supplied by the controller. The feedback from the chair should also be able to communicate the direction it is rotating to the controller.

# DETAILED DESIGN

## Hardware Detailed Design

A hardware component is the lowest level of design granularity in the system.

* The current power input requirements determined for the chair are an input of 24 DC voltage.
* Serial port connections
* Processor requirements (speed and functionality)
* Graphical representation depicting the number of hardware items (for example, monitors, printers, servers, I/O devices), and the relative positioning of the components to each other
* Cable type(s) and length(s)
* User controller will have components to receive serial data as well as adjust the voltage for the chair.

## Software Detailed Design

A software module is the lowest level of design granularity in the system. Depending on the software development approach, there may be one or more modules per system. This section should provide enough detailed information about logic and data necessary to completely write source code for all modules in the system (and/or integrate COTS software programs).

If there are many modules or if the module documentation is extensive, place it in an appendix or reference a separate document. Add additional diagrams and information, if necessary, to describe each module, its functionality, and its hierarchy. Industry-standard module specification practices should be followed. Include the following information in the detailed module designs:

* A narrative description of each module, its function(s), the conditions under which it is used (called or scheduled for execution), its overall processing, logic, interfaces to other modules, interfaces to external systems, security requirements, etc.; explain any algorithms used by the module in detail
* For COTS packages, specify any call routines or bridging programs to integrate the package with the system and/or other COTS packages (for example, Dynamic Link Libraries)
* Data elements, record structures, and file structures associated with module input and output
* Graphical representation of the module processing, logic, flow of control, and algorithms, using an accepted diagramming approach (for example, structure charts, action diagrams, flowcharts, etc.)
* Data entry and data output graphics; define or reference associated data elements; if the project is large and complex or if the detailed module designs will be incorporated into a separate document, then it may be appropriate to repeat the screen information in this section
* Report layout

# EXTERNAL INTERFACES

The current external interface is the idea/plan to add a web based component for the controller. This is not currently within the scope of the basics of the system; however, it is something that is planned to be attempted time permitting.

## Interface Architecture

In this section, describe the interface(s) between the system being developed and other systems; for example, batch transfers, queries, etc. Include the interface architecture(s) being implemented, such as wide area networks, gateways, etc. Provide a diagram depicting the communications path(s) between this system and each of the other systems, which should map to the context diagrams in Section 1.2.1. If appropriate, use subsections to address each interface being implemented.

## Interface Detailed Design

For each system that provides information exchange with the system under development, there is a requirement for rules governing the interface. This section should provide enough detailed information about the interface requirements to correctly format, transmit, and/or receive data across the interface. Include the following information in the detailed design for each interface (as appropriate):

* The data format requirements; if there is a need to reformat data before they are transmitted or after incoming data is received, tools and/or methods for the reformat process should be defined
* Specifications for hand-shaking protocols between the two systems; include the content and format of the information to be included in the hand-shake messages, the timing for exchanging these messages, and the steps to be taken when errors are identified
* Format(s) for error reports exchanged between the systems; should address the disposition of error reports; for example, retained in a file, sent to a printer, flag/alarm sent to the operator, etc.
* Graphical representation of the connectivity between systems, showing the direction of data flow
* Query and response descriptions

If a formal Interface Control Document (ICD) exists for a given interface, the information can be copied, or the ICD can be referenced in this section.

# SYSTEM INTEGRITY CONTROLS

Sensitive systems use information for which the loss, misuse, modification of, or   
unauthorized access to that information could affect the conduct of State programs, or the   
privacy to which individuals are entitled.  
Developers of sensitive State systems are required to develop specifications for the   
following minimum levels of control:  
• Internal security to restrict access of critical data items to only those access types   
required by users  
• Audit procedures to meet control, reporting, and retention period requirements for   
operational and management reports  
• Application audit trails to dynamically audit retrieval access to designated critical   
data  
• Standard Tables to be used or requested for validating data fields  
• Verification processes for additions, deletions, or updates of critical data  
Ability to identify all audit information by user identification, network terminal   
identification, date, time, and data accessed or changed.