



# Capstone Project Specification and Plan

Robotic Air Hockey System  
2018.02.23

David Eelman - 6365316  
Stanislav Rashevskyi - 7028178  
Thomas Abdallah - 7141518

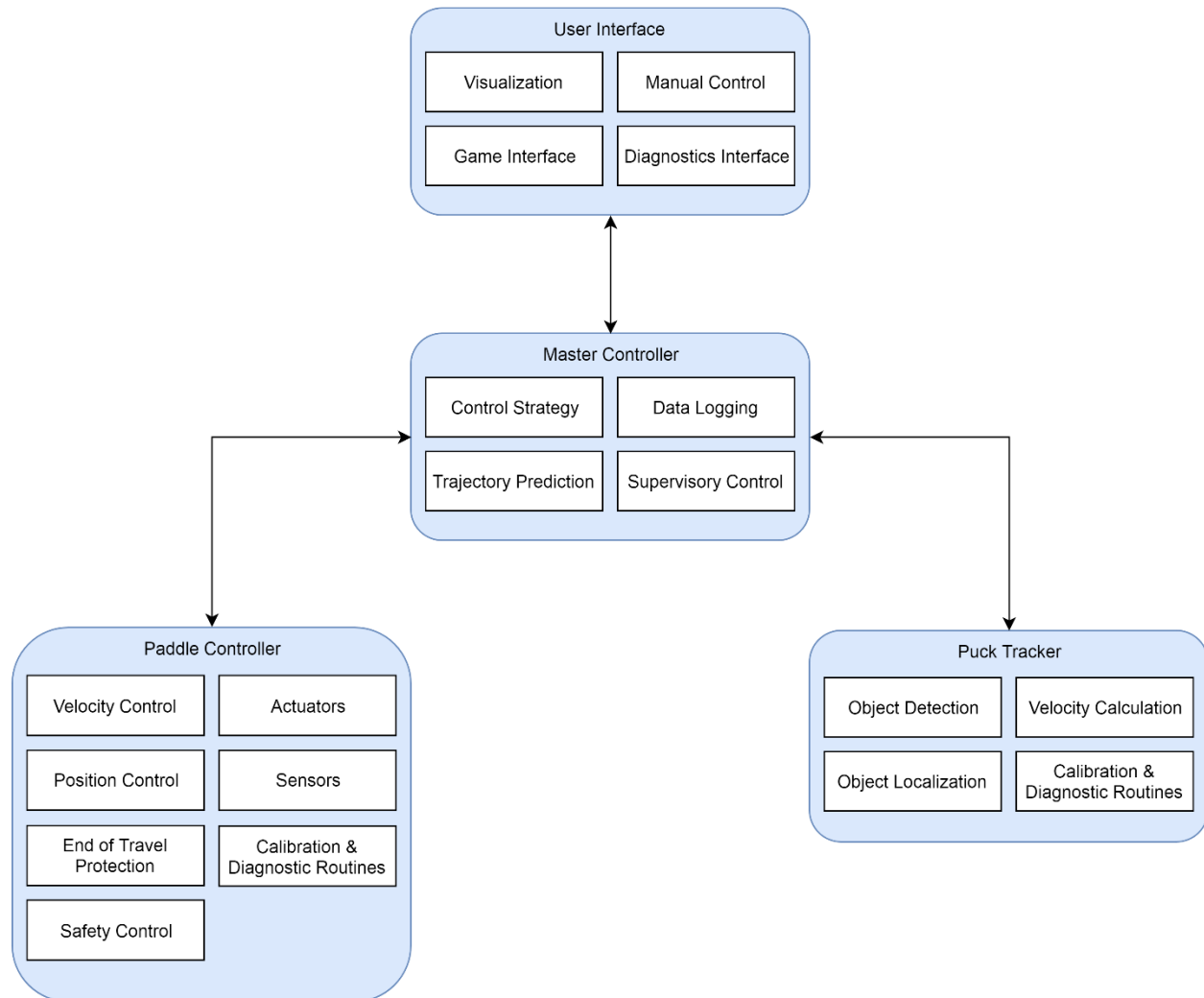
Conestoga College  
Electronic Systems Engineering  
Capstone Project I - EECE74125  
Semester 7, Class of 2018



	0
Complete System Diagram and Description	2
System Level Design Specification	3
Design Task 1: Master Controller	4
Design Task 2: Paddle Controller	5
Design Task 3: Puck Tracker	6
Design Task 4: User Interface	6
Project Schedule	7
Appendix A - Risk Charter	8
Appendix B - Gantt Chart	0
References	0

## Complete System Diagram and Description

The Air Hockey Robot (AHR) shall be comprised of four major modules as shown in Figure 1. The Master Controller will be the central interface for all other modules and is primarily responsible for control strategy. The Paddle Controller is the primary hardware interface responsible for safely controlling the AHR electromechanical system. The Puck Tracker will be responsible for tracking the position and velocity of the air hockey puck on the playing surface. The User Interface will allow users to interact with the AHR system.



*Figure 1 - Air Hockey Robot Block Diagram*

The Master Controller, Puck Tracker, and User Interface will be software of our own design running on and interfacing with off-the-shelf hardware. The Paddle Controller shall be software of our own design implemented on an embedded target of our own design which interfaces with off-the-shelf sensors and actuators. The Paddle Controller mechanical system shall be based on open-source designs with some design modifications to improve manufacturability and achieve our system level performance specifications. The team shall work collaboratively on the system level design. Each module shall have a lead developer who takes final responsibility for the design and implementation of the module. All group

members may contribute to the design, implementation, and review of all modules. Stanislav shall lead development of the Master Controller, Thomas shall lead development of the Paddle Controller, and David shall lead development of the Puck Tracker (Semester 7) and User Interface (Semester 8). Appendix A includes the risk charter developed during the proposal phase of the project which includes risk management for each identified risk. Overall, we believe the risk for this project is low as we have developed our work plan to address high risk challenges early on.

## System Level Design Specification

MuSCoW specifications for the AHR system are provided below. Table 1 provides risk and effort ratings for each submodule based on these specifications.

- 1.0 The AHR must be capable of detecting the position of a 2.5" diameter puck travelling at up to 10m/s to within +/-25mm on a 5.5'x2.5' air hockey table in the ESE 4<sup>th</sup> year classroom regardless of outside lighting conditions.
- 2.0 The AHR must be capable of controlling the position of air hockey paddle in two dimensions covering one half of the air hockey table playing surface.
- 3.0 The AHR must be capable of tracking the position of the paddle to within +/-5mm.
- 4.0 The AHR must provide a mechanism to locate the home position of the paddle.
- 5.0 The AHR should be capable of accelerating the paddle to a speed of 10m/s over a distance of 2.75'
- 6.0 The AHR must shield users from potentially dangerous elements such as rotating mechanical components or pinch points.
- 7.0 The AHR must include a mechanism to detect when the opposing player reaches over centre ice and stop the movement of the air hockey paddle within 100ms.
- 8.0 The AHR could include a mechanism to detect the presence of foreign objects in the path of the air hockey paddle.
- 9.0 The AHR must include a mechanism to stop the paddle from making contact with the boards of the air hockey table.
- 10.0 The AHR could include mechanism to adjust the difficulty.
- 11.0 The AHR must include a defensive control strategy and an offensive control strategy.
- 12.0 The AHR must include a visual user interface with a mechanism for the user to provide input.
- 13.0 The AHR could include a web interface or mobile application.
- 14.0 The AHR should provide a mechanism to visualize the position and trajectory of the air hockey puck.
- 15.0 The AHR should provide an interface to display the game score and time clock.
- 16.0 The AHR could automatically keep track of goals scored.
- 17.0 The AHR could include sound effects.
- 18.0 The AHR should provide an interface to allow the user to change control mode.
- 19.0 The AHR should provide an interface to allow the user to change difficulty.
- 20.0 The AHR should provide an interface to display debugging data and access diagnostics and calibration routines.
- 21.0 The AHR won't support voice control.

- 22.0 The AHR must include an Emergency Stop button mounted in a central location that disables the entire system.
- 23.0 The AHR power supply must operate on 120VAC standard North American electrical power.
- 24.0 The AHR could provide an interface to allow users to manually control the air hockey paddle.
- 25.0 The AHR could provide a mechanism to detect game violations such as paddle crossing centre ice, puck leaving table, paddle covering puck, etc.
- 26.0 The AHR won't support multiple pucks on the playing surface.

*Table 1: Risk and Effort Ratings for each Sub-Module*

Module	Risk (Low, Medium, High)	Effort (Low, Medium, High)
UI – Visualization	Medium	Medium
UI – Manual Control	Low	Low
UI – Game Interface	Low	Medium
UI – Diagnostics Interface	Low	Medium
MC – Control Strategy	High	High
MC – Data Logging	Low	Low
MC – Trajectory Prediction	Medium	Low
MC – Supervisory Control	Low	Low
PC – Position Control	Medium	Low
PC – Velocity Control	Low	Low
PC – Actuators	High	High
PC – Sensors	Low	Medium
PC – End of Travel Protection	Low	Low
PC – Calibration & Diagnostics	Low	Medium
PC – Safety Control	Low	Medium
PT – Object Detection	Low	Low
PT – Velocity Calculation	Medium	Medium
PT – Object Localization	Medium	Medium
PT – Calibration & Diagnostics	Low	Medium

## Design Task 1: Master Controller

The Master Controller development will start early during Semester 7, and will continue throughout Semester 8. The Master Controller will be responsible for game strategy control, trajectory predictions, data logging and supervisory control. It will be implemented using off-the-shelf Linux PC. The final hardware decision will be made during the design phase. The Master Controller will primarily communicate with the Puck Tracker to identify velocity and trajectory of the puck on the playing surface. The Master Controller will primarily communicate with the Paddle Controller to get the paddle position, and set the paddle travel path to intercept the puck. The Master Controller will primarily communicate with the User Interface to display game score and clock, game strategy decisions (puck trajectory predictions, paddle movement, etc.), provide settings menu, have diagnostics interface, allow manual control of the paddle.

By the end of Semester 7 the Master Controller should be capable making defensive decisions based on data coming from the Paddle Controller and the Puck Tracker, and sending the final travel path

to the Paddle Controller. By the end of Semester 8, when the User Interface block is introduced, the Master Controller shall be feature complete with the final software implemented.

The development of the Master Controller software, hardware/electronics selection, and system level design will build on Stanislav's knowledge and skills gained in the ESE program, mostly in courses like Embedded Software and Hardware, Operating Systems, Data Communication, Programming, etc. Experience and techniques from prior ESE project courses will accelerate the development of the Master Controller.

Attention to detail will be required in all aspects of the system design and implementation to ensure real-time operation of a complex robotic system with constant data transfer between various system blocks. The Master Controller software and electronics will require access to some IDE tools (e.g. Sublime, Freescale CodeWarrior, etc.). Stanislav might need to improve his existing scripting skills in Python for the software development of the Master Controller.

The primary risks for the Master Controller development is the managing scope creep, real-time object tracking problems, and inexperience with advanced control algorithms. The Risk Charter in Appendix A outlines these risks, their potential impact on the project objectives, and proposed risk management plans.


## Design Task 2: Paddle Controller

The Paddle Controller will begin development very early during Semester 7 using small, low-power DC motors driven by Semester 4 HCS12 project boards. This will enable early experimentation with the mechanical system design at lower speeds which will help inform our final design. At this time, it is anticipated that a scaled-up version of the jrobot's mechanical system design [1] can feasibly meet our system level performance targets. After the system level design is complete, a custom embedded controller will be designed and manufactured to control the actuators and interface with any sensors and communication networks the Paddle Controller will require. Initial development of the Paddle Controller software will be accomplished with Semester 4 HCS12 project boards and shall be ported to the Paddle Controller hardware once it is complete.

By the end of Semester 7 the Paddle Controller should be capable of controlling the air hockey paddle position, although not necessarily using the final hardware design or being capable of meeting the system performance targets. By the end of Semester 8 the Paddle Controller shall be feature complete with the final hardware design implemented.

The development of the Paddle Controller mechanical system will draw on Thomas' skills and experience in Mechanical Engineering. New knowledge may be required to successfully implement a mechanical system that can operate at high speeds with large accelerations. The Paddle Controller mechanical system may require 3D printing and machine shop resources. The Paddle Controller electronics hardware will require PCB manufacturing and assembly resources.

The development of the Paddle Controller software, electronics, and system level design will build on knowledge and skills gained in the ESE program, specifically the Semester 4, 5, and 6 projects. Special consideration will be required in all aspects of the system design and implementation to ensure



the safe operation of a high-speed robotic system that will operate in close proximity to humans. The Paddle Controller software and electronics will require access to design tools such as Altium Designer and an IDE such as Freescale CodeWarrior.

The primary risks for Paddle Controller development are the mechanical system design and integration, and overall system sizing. The Risk Charter in Appendix A outlines these risks, their potential impact on the project objectives, and proposed risk management plans.

## Design Task 3: Puck Tracker

The Puck Tracker development will begin early in Semester 7 and is scheduled to be complete early in Semester 8. Experimentation with object tracking has begun using a webcam, a Linux system, and OpenCV, an open source computer vision library. During the design phase, an object detection method will be selected and further prototyping will begin. Once the target object can be reliably tracked on the playing surface, development of object localization techniques will commence. Object localization will provide relevant feedback to the Master Controller about the puck location relative to the playing surface. At this time, the velocity calculations from the Puck Tracker will also be required by the Master Controller in order to predict trajectory of the puck and devise a control strategy to intercept the puck. All of the functionality mentioned above will be implemented by the end of Semester 7, giving enough time in Semester 8 to develop more complex control strategies for the AHR. Early in Semester 8, diagnostic and calibration routines will be implemented to optimize the performance of the Puck Tracker.

The development of the Puck Tracker software will build on knowledge and skills gained in the ESE program, specifically the physics, math, and programming courses. Developing the Puck Tracker module will also require learning how to accurately detect objects in motion. While an object tracking method has not yet been confirmed, open source libraries will likely be utilized to decrease development time. A debugging interface such as the python command line will be required to develop the module.

The primary risks for the Puck Tracker software development relate to problems with real-time object tracking. The accuracy of the object tracking software will be essential for the trajectory prediction control strategies. The Risk Charter in Appendix A outlines these risks, their potential impact on the project objectives, and proposed risk management plans.

## Design Task 4: User Interface

The User Interface development will begin late in Semester 7 and continue until the end of Semester 8. The preliminary interface will be implemented using a display and a command line for system bring-up tasks. This command line interface will include a method to manually control the Puck Tracker module using a keyboard. Once a design decision has been made for a platform on which to implement the Master Controller, an off-the-shelf display unit will be selected to interface with the entire system. This display unit may have a touchscreen or a keyboard for user input. After a display unit has been selected, research will be commenced to determine what GUI implementation best suits the chosen hardware. The actual GUI development will begin early in Semester 8 once the 3 other modules have sufficient functionality. Diagnostic display will be the first focus for User Interface and as the AHR



system progresses a game interface will be developed. A visualization of the Master Controller's trajectory prediction will be implemented for demonstration to technical and non-technical audiences. By the end of Semester 8 the User Interface will be completed and fully integrated into the AHR.

The development of the User Interface software will build on knowledge and skills gained in the ESE program, specifically the Semester 6 courses and projects. The development will also draw on David's skills and experience attained in past UI implementations. His familiarity with common GUI methods such as HTML and PyQt may be useful for completing the User Interface for the AHR. New knowledge may be required to successfully implement a User Interface that can run on the chosen platform for the Master Controller.

The primary risks for the User Interface development are inexperience with HMI design/implementation and scope creep. The Risk Charter in Appendix A outlines these risks, their potential impact on the project objectives, and proposed risk management plans.

## Project Schedule

The schedule in Figure 2 is a high level representation of the project development and milestones. Appendix B includes a detailed Gantt chart.

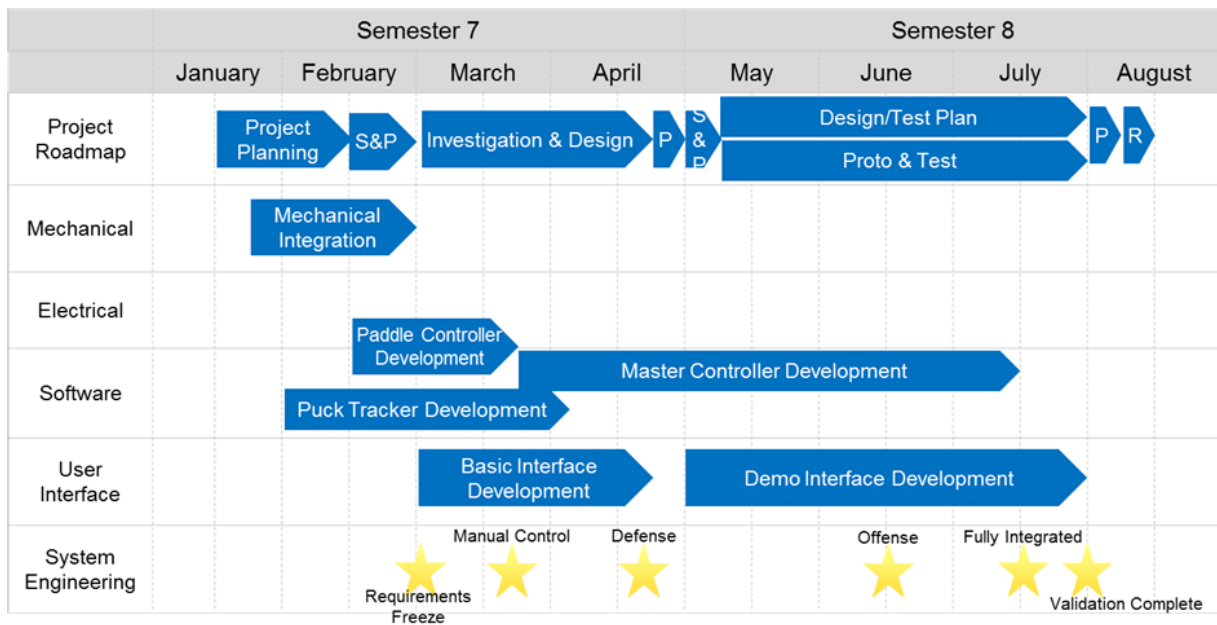


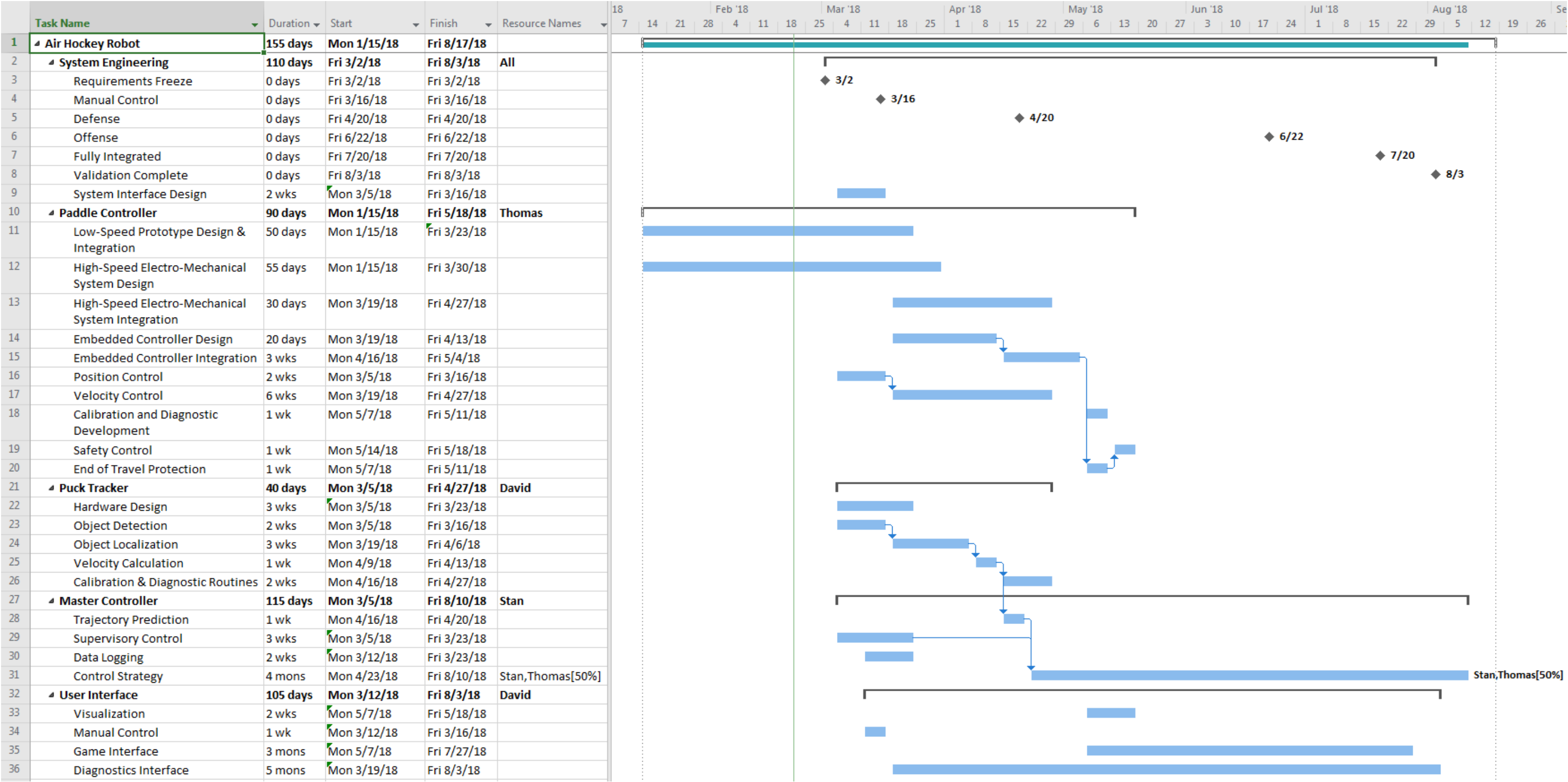
Figure 2 - High Level Project Schedule



## Appendix A - Risk Charter

Risk (Priority Highest to Lowest)	Category	Impact on Project Objectives	Risk Management Plan
<b>1. Mechanical design/integration problems</b>	Technical	Unable to control robot motion	-Start mechanical prototyping early  -Leverage group members Mechanical Engineering experience
<b>2. Real-time object tracking problems</b>	Technical	Unable to automate robot motion	-Start object tracking prototyping early  -Leverage proven open-source object tracking solutions
<b>3. Security of project in shared classroom</b>	External	Lack of lab workspace availability.  Potential damage to project hardware.	-Utilize dedicated ESE lab space  -Advocate for continued support of ESE dedicated lab space
<b>4. Catastrophic loss of data</b>	Organizational, external, technical	Schedule delays.	-Utilize source control for all project materials  -Manually back up all data once per week
<b>5. Managing scope creep</b>	Organizational, project management	Schedule delays.  Lack of focus on core features.	-Strictly define scope of project during planning phase  -Additional features shall only be implemented after 100% completion of core project features
<b>6. System sizing incorrect</b>	Technical, performance	Lower than desired system performance.	-Use system level performance requirements to drive component design  -Define system level performance requirements based on real-world data
<b>7. Inexperience with HMI design &amp; implementation</b>	Technical	Less relatable demonstration.  Worse user experience.  Difficult to debug.	-Define user interface features early (see: Managing scope creep)  -Start HMI prototyping early  -Utilize popular GUI implementation solutions

Appendix B - Gantt Chart



## References

- [1] "Air Hockey Robot (a 3D printer hack)", jjrobots, 2018. [Online]. Available: <https://www.jjrobots.com/air-hockey-robot-a-3d-printer-hack/>. [Accessed: 08- Feb- 2018].