Mobile Autonomous Vehicle for Research in Intelligent Control (MAVRIC)

Preliminary Project Design Document

Chad Condon* ccondon@uw.edu
Keenan Fejeran* kfejeran@uw.edu
Ben Foster* benf94@uw.edu
Caleb Horst* calebjh@uw.edu

March 17, 2015

*Institute of Technology, University of Washington Tacoma

Contents

List of Figures

List of Tables

1 Problem Statement

Professor Mobus has a robotics platform for graduate research. Time and previous projects have created a need for updating and expanding its communication and control systems. The platform needs an on-board embedded Linux computer that can transceive data and control signals and execute Professor Mobus's learning algorithm. An ARM microcontroller will be used to interface with sensor arrays and actuators. Sensors are to be implemented to mimic the senses of a typical mammal. Existing power systems will be adapted to supply new and retained components with an aim to support uninterrupted 3+ hour experiments. Component selection, software choices, and design decisions need to be made with extensibility and the Bachelor of Science in Computer Engineering and Systems (CES) curriculum in mind.

2 Background

Professor Mobus directs graduate research at the University of Washington Tacoma's Institute of Technology under the banner of his Adaptive Agents Laboratory[1]. The lab is dedicated to the study of adaptive processes in artificial agents. Professor Mobus's Adaptrode learning mechanism[**?**] is an implementation of an artificial neural network which emulates animalian learning in a dynamic environment.

Professor Moubus's Mobile Autonomous Vehicle for Research in Intelligent Control (MAVRIC)[**?**] is the robotics platform used for testing of these adaptive agents. The MAVRIC began as a commercial ActivMedia Pioneer 2 platform. The platform maneuvered with two independently motorized wheels. It was powered by an on-board bay loaded with 1–3 12 V rechargeable batteries. Stimuli to the MAVRIC were delivered from nearby objects to a sonar array, from radio controlled lights to four light sensors, and from radio controlled speakers to a microphone. Processing was done remotely from a computer base station.

Last year, a team attempted to modernize and extend the MAVRIC's sensing and computing platforms.[**?**] The team succeeded in interfacing the motors and the sonar array. However, the team used two Arduino Mega microcontroller boards for discrete input and output processing and a Raspberry Pi single board computer (SBC) for higher level processing. Wiring was done using multiple breadboards and impermanent jumpers. The client was generally dissatisfied with the product.

Moving forward, a more intentional, engineering driven design is expected. Professor Mobus would prefer architectures that fall in line with the Institute of Technology's curricula (e.g. ARM Cortex-M). This platform is intended for continued graduate research into adaptive agents and artificial intelligence.

The team was motivated to pursue this project due to its larger scope and complexity. There is a strong interest in robotics, as well as graduate research,

[1] `http://faculty.washington.edu/gmobus/AdaptiveAgents/`

among our team. Compared to the others proposed, this project presented much more interesting problems and opportunities.

3 Design Plan

Design began with analysis of the artifact's characteristics. These characteristics were identified and categorized as objectives, constraints, functions, and means. These are shown in Table 1. The following subsections describe each of these categories more thoroughly.

3.1 Objectives

The following objectives were identified for the MAVRIC design. Measurable units and point values are assigned to each. Note that the points awarded for each objective fall within the range of 0–10. Any larger and smaller scores are mapped to 10 and 0 respectively.

3.1.1 Low Cost

The budget for the project is to be reasonably minimized. The client proposed a reasonable budget estimate of $200. The unit assigned to measure cost is dollar amount (c). This metric is weighted by the point assigning formula

$$p = 400 - \frac{c}{30}. \tag{1}$$

3.1.2 Small Operating System

In order to provide for future extension of the project and to ensure system responsiveness, the size of the operating system is to be minimized. The size of the operating system is measured in gigabytes occupied on the file system (f). This metric is weighted by the point assigning formula

$$p = 12 - 4f. \tag{2}$$

3.1.3 Long Battery Life

The MAVRIC platform is intended for use in experiments lasting as long as three hours. As such the operation time for the robot on a single charged battery is to be maximized. Hours are the unit used to measure to time the robot functions on a single charged battery (t). Points for this objective are awarded by the formula

$$p = 5t - 12.5. \tag{3}$$

Characteristic	**Objective**	**Constraint**	**Functionality**	**Means**
High resolution audio sensing	✓			
Long battery life	✓			
Low cost	✓			
Small operating system	✓			
Wide range audio sensing	✓			
"Curriculum compatible" microcontroller and computer				✓
Uses existing sonar array				✓
Linux operating system		✓		✓
Compatible with Adaptrode brain software		✓		
Battery level monitor		✓	✓	
Drive motor control		✓	✓	
Senses distances to forward obstacles		✓	✓	
Wireless data transmission		✓	✓	
Wireless stop switch		✓	✓	
Cliff detection			✓	
Touch sensor			✓	
Gradient sensing to mimic smell			✓	
Inertial sensing			✓	
Interfaces with camera			✓	
Internal cooling			✓	
Internal temperature sensing			✓	
Sound output			✓	
Stereoscopic audio sensing			✓	

Table 1: Characteristics

3.1.4 Wide Range Audio Sensing

Audio sensing on the MAVRIC is intended to emulate an animal's hearing. As such, it should detect as much of the audible frequency range as feasible. The frequency range is gauged by octaves as determined by the maximum and minimum detectable frequency ($f_{\max}$, $f_{\min}$). Points for this objective are awarded by the formula

$$p = \log_2 \left(\frac{f_{\max}}{f_{\min}} \right). \tag{4}$$

3.1.5 High Resolution Audio Sensing

The resolution of audio sensing is a measure of the granularity of the audio input. Sound within discrete frequency bands will be measured. As such, the number of these bands (n) is to maximized towards an ideal of 10. Points for this objective are awarded by the formula

$$p = 2n - 10. \tag{5}$$

3.2 Constraints

The following constraints were identified for the MAVRIC design. Some of these constraints were also identified as either functions (3.3) or means.

3.2.1 Battery Level Monitor

A battery level monitor needs to be implemented in order to mimic "hunger" and to also keep track of how much power is left during an experiment.

3.2.2 Compatible with Adaptrode Brain Software

Graduate research students will be developing Adaptrode brain software for this robot. The software is written in C++ and optimized for a Linux environment. Our hardware and software systems must be compatible with the Adaptrode brain software.

3.2.3 Drive Motor Control

The robot should be able to drive the motors in order to move about. This will be done with varying power so the vehicle can move at different speeds and turn at varying angles.

3.2.4 Linux Operating System

The single-board computer (SBC) on the MAVRIC must be running a distribution of Linux.

3.2.5 Senses Distances to Forward Obstacles

There is already an existing Sonar array mounted on the robot that is connected to a driver and functions properly. Therefore there is no need to replace this sonar array or fabricate our own.

3.2.6 Wireless Data Transmission

The MAVRIC collects data from its sensors during experiments. These data need to be sent wirelessly to the base station.

3.2.7 Wireless Stop Switch

A stop switch located on the base station needs to be able to halt all processes on the robot at any time during an experiment. This is to prevent any damage that may be caused to the vehicle or surrounding environment and this is a way to halt the experiment if there is any unpredicted behavior.

3.3 Functions

The following functions of the MAVRIC were identified. Some of these were also recognized as constraints discussed previously in 3.2. Performance specifications for each of the functions were created. Candidate means for each of these functions were also generated and are compared in Table 2.

3.3.1 Battery Level Monitor

The MAVRIC will be able to gauge the charge level of its battery. During experiments, this data will serve as an analog to hunger.

3.3.2 Drive Motor Control

The MAVRIC will maneuver using two independently driven DC motors. Control signals for these motors will be passed from the brain software.

3.3.3 Gradient Sensing to Mimic Smell

In order to simulate smell, the MAVRIC will have continuous sensor input representing discrete "odor" signals. These signals are to be generated from beacons in the environment such that signal readings increase with proximity to the beacon.

3.3.4 Inertial Sensing

In order to simulate an animal's vestibular system, inertial sensing will be implemented. Readings from the three lateral or rotational axes are acceptable.

3.3.5 Interfaces with Camera

To provide for extension of the MAVRIC's capabilities and future research into computer vision, a color video camera is to be installed. The camera will not be utilized in the current scope of this project.

3.3.6 Internal Cooling

As a mammal does by sweating, the MAVRIC will be able to cool itself. The ActivMedia Pioneer 2 platform has an internal fan which may be utilized.

3.3.7 Internal Temperature Sensing

The MAVRIC will monitor its internal temperature. This data will be provided to the brain software for interpretation.

3.3.8 Senses Distances to Forward Obstacles

The MAVRIC will sense physical obstacles within a short range of its front. This sense is to emulate an animal's antennae. The ActivMedia Pioneer 2 platform includes a sonar array which may be utilized.

3.3.9 Sound Output

The MAVRIC will be capable of generating different sounds. The ActivMedia Pioneer 2 platform includes a small speaker which may be utilized.

3.3.10 Stereoscopic Audio Sensing

To emulate hearing, the MAVRIC will have two audio sensors which will be used to create level readings within discrete audible frequency bands. These signal filters will need to be developed and implemented.

3.3.11 Wireless Data Transmission

The Adaptrode brain software interprets data and generates signals through "in-slots" and "out-slots." During experiments, these data will be transmitted wirelessly to a computer base station for logging and analysis.

3.3.12 Wireless Stop Switch

Functionality for remotely pausing operating of the robot will be provided. This will allow for a "kill switch" for service to the robot or environment during an ongoing experiment.

Functions \ Means	1	2	3	4
Battery Level Monitor	Voltage divider to ADC			
Drive Motor Control	Pololu drivers	Alternate driver		
Gradient Sensing To Mimic Smell	Filtered light-to-frequency sensors	Filtered photoresistors	Filtered audio	Bluetooth
Inertial Sensing	Accelerometer	Gyro	Dual package	
Internal Cooling	Motor driver and fan	Op-amp and fan	Transistor amp and fan	
Internal Temperature Sensing	Thermistor and voltage divider	Tiva internal sensor	Thermocouple	
Senses Distances to Forward Obstacles	Microcontroller driven sonar	Raspberry Pi driven sonar		
Sound Output	SBC output with amplifier	single frequency tones from microcontroller		
Stereoscopic Audio Sensing	hardware filters	SDFT	FFT	Goetzel algorithm
Wireless Data Transmission	XBee	Wi-Fi		
Wireless Stop Switch	integrated into data transmission	exclusive interface		
Interfaces with Camera	Raspberry Pi camera	COTS camera		
Touch Sensor	button bumpers	pressure pads	flex sensors	custom sensor
Cliff Detection	ultrasonic to main microcontroller	IR to main microcontroller	ultrasonic to slave microcontroller	IR to main microcontroller
Interchip Communication	I^2C	SPI	UART	

Table 2: Morphological Chart

4 Design Research

The product being created is a specialized research platform, specifically outfitted with sensors and actuators for our clients desires. The chassis being retrofitted was originally a Pioneer 2 robot platform from ActivMedia. Other research platforms are available from Activrobots, such as the newer Pioneer III[2] and PeopleBot[3]. These platforms also have extra sensor arrays, however they do not have as much customizability as needed to implement all the desired functions. Furthermore, these packages come at a significant price, while the current project recycles resources to minimize cost.

The Washington company CoroWare also produces a number of robotics platforms for research.[4] However, cost ranges from $400[5] to $19,999[6], well above the acceptable budget for the entire project. All of the platforms offered would also require significant adjustments and extension to meet the client's needs.

5 Design Alternatives

5.1 Interchip Communication

The interchip communication (between the microcontroller and the SBC) has been identified as the project's *high risk item*. The entire success of the project rests on effective, timely data transfer between these two processing platforms. The bulk of sensor data and actuator control will be driven by the microcontroller, but high level control will be done in Professor Mobus's brain software as installed on the SBC.

5.1.1 Universal Asynchronous Receiver/Transmitter

Serial communication using universal asynchronous receiver/transmitter (UART) is perhaps the simplest approach. However, because we chose to use the XBee for wireless data transmission, the serial resource needed to use UART is unavailable.

5.1.2 Serial Peripheral Interface

Serial peripheral interface (SPI) is a solution that uses $3+N$ wires for synchronous communication between a master and N slaves. Research suggests that SPI is less of an industry standard compared to I^2C. In addition SPI requires more resources to expand upon if more slave nodes were desired. These aspects cause an otherwise practical solution to score poorly amongst other choices.

[2] http://activrobots.com/ResearchRobots/PioneerP3DX.aspx
[3] http://activrobots.com/ResearchRobots/PeopleBot.aspx
[4] http://www.corobot.net/products/
[5] http://www.corobot.net/spark/
[6] http://www.corobot.net/corobot-pro/

5.2 Base Station Communication

Communication between the robot and the base station was a relatively easy decision to make since we were provided an Xbee and found that it fit our needs quite nicely. One other option was considered for sending data back and forth between the robot and the base station.

5.2.1 Wi-Fi

We considered plugging a USB Wi-Fi adapter into the Raspberry Pi and having it send data to the base station. That way we would have use of the serial port on the Raspberry Pi for UART interchip communication. Wi-Fi would also require reconfiguration for any given computer base station and create a compatiblity requirement between the two network adapters. The XBee, however, uses a similar means of communication and, while the Xbee is using the only serial port on the Raspberry Pi, it is still simpler to use than a Wi-Fi communication device and it serves its purpose effectively without any recurring configuration.

5.3 Audio Input

Multiple options were considered based on their fulfillment of the objectives of a wide total range of frequencies detectable and a large number of sub-ranges measured.

5.3.1 Hardware Solutions

A hardware solution would minimize the processing load of this filter at the cost of extensibility and configurability.

Hardware Band-Pass Filters The advantages provided by the use of hardware band-pass filters (as shown in Figure 1) include its extensibility. By adding hardware, many ranges, bands and frequencies are possible. However, limited analog inputs on a microcontroller would require additional considerations or multiplexers or other hardware, tuning frequency ranges properly may be difficult or unfeasible, quantity of hardware necessary would make extension less elegant and wiring prone to error.

External Chips (MSGEQ7) A discrete component such as this would provide an easy interface. An IC is a small package which saves space. This specific package also requires only minimal tuning. This solution would also require minimal processor time on either the microcontroller or the SBC. However, this solution would not provide for extensibility and only provides seven frequency bands.

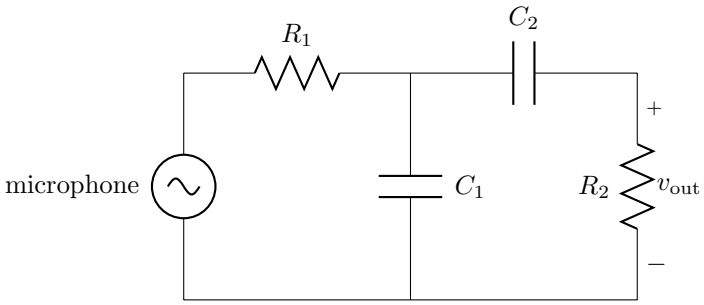

Figure 1: Band-Pass Filter

5.3.2 Software Solutions

A software solution would allow for extensibility at the cost of a significant computational overhead.

Fast Fourier Transform A fast Fourier Transform (FFT) would provide a degree of extensibility and configurability in that it allows for arbitrarily defined bands. However, its use would require a large amount of high speed sampling requiring higher processor use. Also, the team possesses limited practice in signal processing.

Sliding Discrete Fourier Transform A sliding discrete-time Fourier transform (SDFT) provides all of the functionality of the FFT at a smaller scale. However, it suffers from similar drawbacks as well.

Goertzel Algorithm The Goertzel algorithm provides a more numerically efficient approach to the Fourier transform. However this efficiency advantage is only at small numbers of frequencies. Also, the algorithm only measures specific frequencies, not bands.

6 Preliminary Design

Of the design choices discussed in Sections 4 and 5, choices were made based on the parameters discussed in 3 and based on the feasibility of completing them within the alloted time and skill set of the team. Figure 3 shows the preliminary electrical hardware design.

The main computational platforms selected are the Texas Instruments Tiva C Series TM4C1294 LaunchPad[7] and the Raspberry Pi B+[8]. The Tiva was selected because it is the platform used in the CES curriculum and thus best satisifies curriculum compatibility. The Raspberry Pi B+ was chosen for its known compatibility with existing software and its improvements to the previous version's power and data busses. The new Raspberry Pi 2 B was not chosen because documentation and support for the upgraded ARMv7 processor are still immature.

Alternatives to the Raspberry Pi, such as the BeagleBone Black[9] and the CubieTruck[10], were also considered. However they did not offer advantages significant enough to justify a change in platform.

For interboard communication, the I^2C protocol was selected. This choice was deemed most appropriate because of its minimal wiring, high speed, and hardware support from both boards.

Several hardware components from the existing platform are being retained. The sonar array completely satisfies the obstacle detection functionality. Its use will also reduce engineering time and cost. The original motors in the ActivMedia Pioneer 2 chassis are also being kept as they perform adequately. The stock battery bay satisfies the power requirements for the system. The motor drivers installed by last year's team[11] , the original fan, and the speaker are all likewise being used.

Steroscopic audio sensing will be implemented using electret microphones[12] and the MSGEQ7 graphic equalizer display filter[13]. Though this solution only provides seven bands of resolution, the MSGEQ7 covers sufficient frequency range and off-boards all signal processing. Given the timing requirements of the microcontroller's software and the limited analog input capabilities of the Raspberry Pi, this compromise is deemed acceptable. This configuration requires minimal calibration, processing time, and physical space in the system.

Touch sensing will be implemented with flex sensors mounted offset from the perimeter of the chassis. The sensors operate as a variable resistor and will be configured with a voltage divider for input into the microcontroller's analog-to-digital converter (ADC). This layout is shown in Figure 3. This design will create a continuous output signal as opposed to the binary output of a simple switch. This is more compatible with the client's anticipated use. Four

[7] http://www.ti.com/ww/en/launchpad/launchpads-connected-ek-tm4c1294xl.html
[8] http://www.raspberrypi.org/products/model-b-plus/
[9] http://beagleboard.org/BLACK
[10] http://cubieboard.org/model/cb3/
[11] https://www.pololu.com/product/1451
[12] https://www.adafruit.com/products/1063
[13] https://www.sparkfun.com/datasheets/Components/General/MSGEQ7.pdf

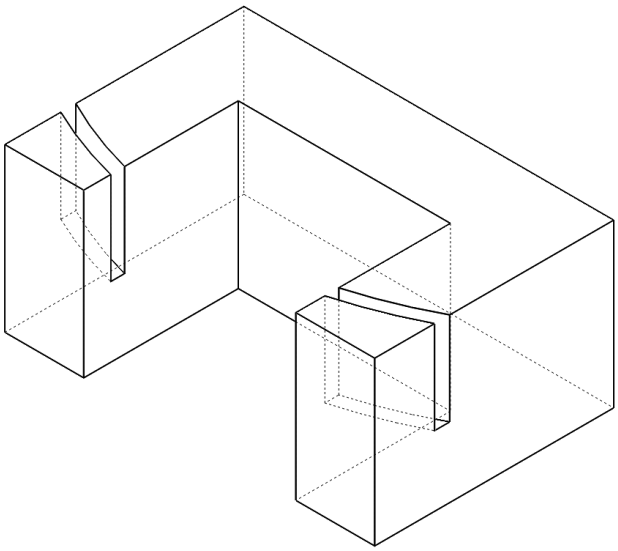

Figure 2: Flex Sensor Bumper Mount

sensors will be installed: one on each side and two in the rear. Front touch sensors are deemed redundant with the sonar array and have been excluded. Mounting hardware for the flex sensors will be modeled with SolidWorks and fabricated with an additive 3D printer at a local maker space. Figure 2 shows a design for the sensor mount.

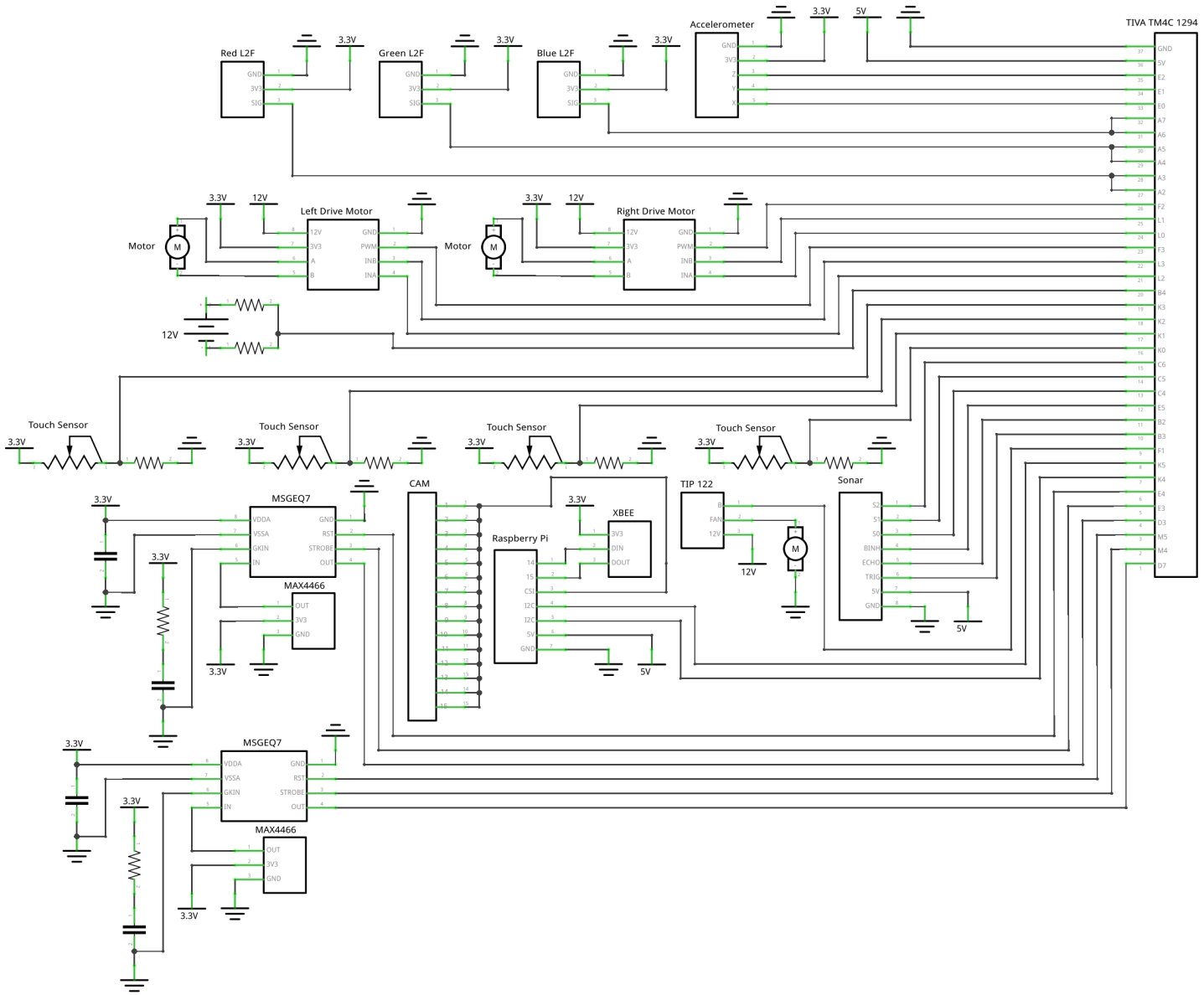

Figure 3: Electrical Schematic

7 Testing

Testing will be completed at every functional level of design. Unit testing will be done as each of the sub-components is fabricated or obtained to ensure anticipated behavior under our specified operating ranges.

After system components are completed and tested, subsystems (such as those shown in Figure 3) will be assembled from them. Integration testing will be done to ensure the expected interface. Likely flaws and failures will be identified and design revision will be required. The following test scripts demonstrate the type of unit testing that will be conducted.

Figure 4 shows the plan for hardware testing of the sonar array that has been developed. This testing is under way. Table 5 shows the data collected in a single trial. These data are graphed in Figure 5. Further trials will increase the fidelity of the time-distance relationship and may correct for some experimental inconsistencies.

S_2	S_1	S_0	**Sensor**
0	0	0	1
0	0	1	0
0	1	0	3
0	1	1	2
1	0	0	5
1	0	1	4
1	1	0	7
1	1	1	6

Table 3: Sonar Select Signals

S_2	S_1	S_0	**Sensor**	**Object distance** (cm)	**Trial**	**Time** (ms)
0	0	0	1	10.0	1	—
...	...	...	...	...	...	...
1	1	1	6	40.0	8	—

Table 4: Sonar Test Data Format

Figure 6 shows the plan developed for testing the sonar driver software.

1. Using DC power supply at 5 V
   (a) Supply to sonar array
      i. HIGH to 5 V
      ii. GND to ground
   (b) Apply select signals to S_2, S_1, S_0 as shown in Table 3
2. Using function generator
   (a) Apply a 3.3 V 560 ms pulse at 10 Hz to $TRIG$
   (b) Apply a 3.3 V 560 ms pulse at 10 Hz to 570 ms out of phase to $BINH$
3. Using oscilloscope
   (a) Apply digital channel 0 probe to $TRIG$
   (b) Apply digital channel 1 probe to $ECHO$
4. Position selected sonar sensor on ledge to prevent surface interference
5. Using a ruler and object with a flat vertical surface
   (a) Place the object such that its surface is orthogonal to the sensor
   (b) Distance the object beginning at 10.0 cm and incrementing by 1.0 cm until 40.0 cm
   (c) Record the distance
6. Using the oscilloscope
   (a) Measure and record the time between the rising edges on channel 0 and channel 1
7. Repeat 8 times for each sensor and populate data as shown in Table 4

Figure 4: Sonar Array Test Script

Distance (cm)	**Time** (ms)							
Sensor	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
10.0	0.790	0.736	0.796	0.788	0.956	0.964	0.956	0.950
11.0	0.800	0.744	0.792	0.792	0.956	0.960	0.948	0.950
12.0	0.800	0.742	0.800	0.792	0.960	0.956	0.948	0.950
13.0	0.800	0.736	0.812	0.800	0.960	0.956	0.968	0.970
14.0	0.830	0.758	0.804	0.832	0.968	0.964	0.972	0.970
15.0	0.880	0.828	0.880	0.888	0.972	0.972	1.03	1.04
16.0	0.930	0.884	0.932	0.948	1.03	1.01	1.09	1.09
17.0	0.990	0.940	0.996	1.01	1.09	1.09	1.14	1.15
18.0	1.05	1.01	1.05	1.07	1.14	1.14	1.21	1.20
19.0	1.11	1.07	1.11	1.13	1.20	1.20	1.26	1.26
20.0	1.19	1.13	1.18	1.19	1.27	1.25	1.32	1.31
21.0	1.24	1.19	1.33	1.26	1.32	1.32	1.37	1.37
22.0	1.29	1.25	1.39	1.32	1.38	1.37	1.43	1.43
23.0	1.36	1.31	1.45	1.38	1.44	1.43	1.49	1.49
24.0	1.41	1.37	1.51	1.44	1.50	1.49	1.54	1.55
25.0	1.46	1.48	1.57	1.49	1.55	1.55	1.61	1.61
26.0	1.53	1.54	1.63	1.55	1.61	1.60	1.66	1.66
27.0	1.60	1.60	1.68	1.62	1.67	1.66	1.73	1.72
28.0	1.65	1.68	1.75	1.68	1.73	1.72	1.79	1.78
29.0	1.72	1.74	1.81	1.74	1.79	1.78	1.84	1.84
30.0	1.78	1.80	1.86	1.80	1.84	1.83	1.89	1.89
31.0	1.84	1.84	1.92	1.86	1.90	1.89	1.95	1.95
32.0	1.90	1.92	1.98	1.92	1.96	1.95	2.01	2.01
33.0	1.95	1.98	2.04	1.99	2.02	2.01	2.07	2.07
34.0	2.02	2.02	2.10	2.04	2.09	2.08	2.12	2.14
35.0	2.08	2.10	2.16	2.11	2.14	2.13	2.18	2.20
36.0	2.15	2.16	2.22	2.18	2.20	2.19	2.24	2.26
37.0	2.20	2.22	2.27	2.23	2.26	2.25	2.30	2.32
38.0	2.26	2.28	2.33	2.29	2.32	2.31	2.34	2.37
39.0	2.33	2.34	2.39	2.35	2.37	2.37	2.42	2.43
40.0	2.38	2.40	2.44	2.40	2.43	2.42	2.47	2.49

Table 5: Sonar Test Data

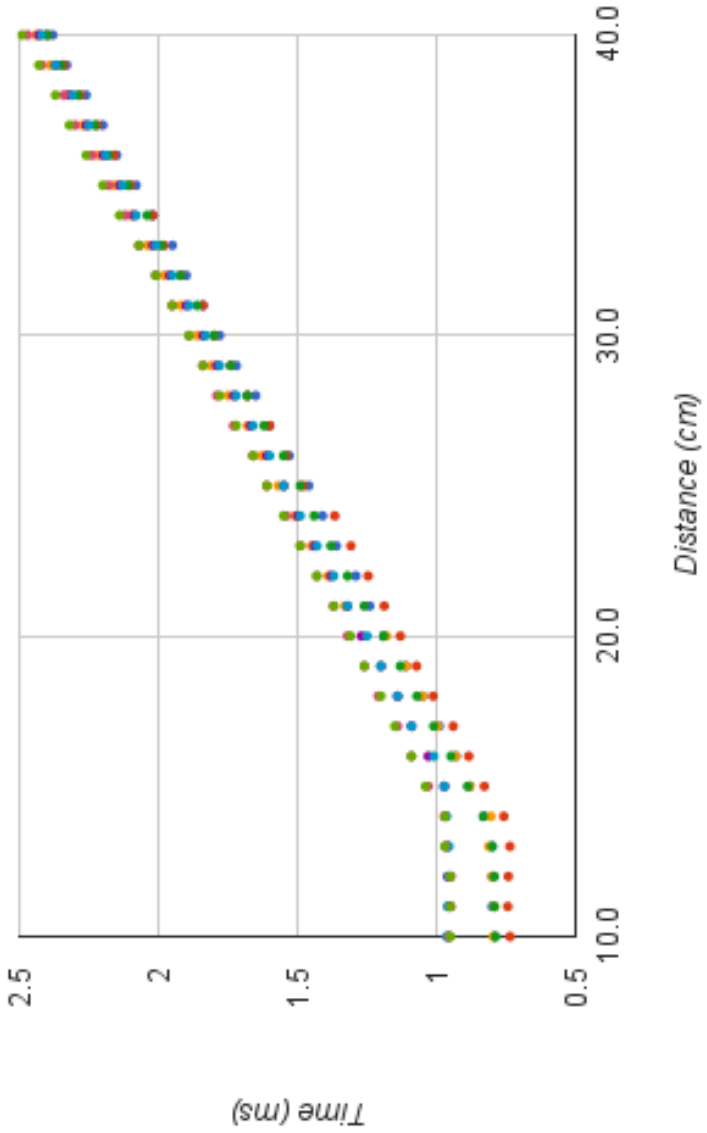

Figure 5: Sonar Test Results

1. Connect a computer running Code Composer Studio via USB to the debug port of the Tiva board.
2. In the code that integrates with sonar driver
   (a) Send select line signals and print values
   (b) Activate trigger signal and print current time
   (c) Activate inhibit signal and print current time
   (d) Make sure select line is sending correct value according to specified order.
   (e) Ensure the trigger and inhibit signals are activated 0.57 ms apart.
   (f) Iterate through each select line value as shown in Table 3
3. Connect sonar array to Tiva board GPIO
4. Position selected sonar sensor on ledge to prevent surface interference
5. Using a ruler and object with a flat vertical surface
   (a) Place the object such that its surface is orthogonal to the sensor
   (b) Distance the object beginning at 10.0 cm and incrementing by 1.0 cm until 40.0 cm
   (c) Record the distance
6. In the code that integrates with the sonar driver
   (a) Send select line signal and print value
   (b) Activate trigger and inhibit signals as described above
   (c) Wait for echo interrupt and print resulting value
   (d) Calculate distance from value and compare to actual distance from sonar.
   (e) Iterate through each select line value.
7. Repeat for each incremental distance.
8. Repeat 8 times for each sensor.

Figure 6: Sonar Driver Test Script

8 Considerations

Because our project is specifically the renovation of a single robot platform, many of the standard design considerations are out of project scope. (For instance, we do not consider the price viability of using the Raspberry Pi for mass production). That being said there are still some important topics to consider as listed here.

8.1 Design for Manufacturing

In order to reduce the manufacturing cost in our project, many of the pre-existing components are being retained. Particular examples include the chassis, motors, wheels, and sonar array. These modules have been tested to perform to project requirements and show no reason to be replaced.

8.2 Design for Assembly

Use of the pre-existing chassis simplifies the design as there are already points to install sensors, actuators and power supplies. In fact, because the chassis is already equipped with motors, the only assembly will be done to replace prototyping breadboards with permanent, soldered circuit boards. This will be done by hand as a single run labor task. Where practical, "commercial-off-the-shelf" components have been chosen to reduce complexity in assembly and cost.

8.3 Environmental

The main environmental concern for our robot is potential pollution from the energy source. In our current design, we use a rechargable lead-acid battery. The recharable nature allows us to reduce waste generated by single use batteries. In addition lead-acid battery recycling is one of the most successful recycling programs in the world. Retailers that sell batteries collect used batteries for recycling, as required by most state laws. As such, at the end of the MAVRIC project life cycle, the battery should be taken to a local retailer for recycling.

8.4 Ethical

It is important to remember that our project is specifically to update the hardware inside the MAVRIC system, and has only a limited relationship with the driving software (which, as an intelligent agent, has a legion of ethical considerations). That being said, we can consider the ethical implications of the actuators and sensors we render to the robot.

8.4.1 Privacy

An autonomous agent has potential to invade privacy as it collects data. If used as intended, inside a controlled experiemental lab space, there should be

no personal data for collection that could cause ethical issues. With its limited sensor space, we conclude this to be true of all sensors onboard MAVRIC.

8.4.2 Hazards

The robot actuators are a pair of DC motors capable of driving at moderate speeds. In order to prevent harmful collisions, we implement a remote stop mechanism, or "kill switch." If used in the proper lab space with the stop switch, operational hazards should be minimized.

8.4.3 Economic

As progress continues in computational intelligence, the potential effects on the labor market are severe. Just as the agricultural revolution reduced the manpower required for food and the industrial revolution in many ways commoditized labor, this technological revolution may remove the need for many positions in today's economy. As machines have supplanted many laborers, the economy has moved to value "knowledge workers." As computational intelligence matures, decision-making that once only humans could do, may be overtaken by machines. Obviously, this would be very disruptive to the labor market.

9 Project Management

A work breakdown structure (shown in Figure 7) was developed to identify and clarify the phases of the project. A more granular breakdown of the tasks, dependencies, and deadlines outlined in the work breakdown structure (Figure 7) was developed into a Gantt chart. Figure 8 shows the Gantt chart.

The budget for the project is outlined in the Bill of Materials shown in Table 6. An estimated total cost of $229.13 has been identified. According to the metrics assigned by equation 1, this budget earns 6 of 10 points.

Item		**Count**	**Cost per Item**	**Cost**
MAX4466	electret microphone amplifier	2	16.00	32.00
MSGEQ7	graphic equalizer display filter	2	4.95	9.90
TM4C1294 LaunchPad	microcontroller board	1	19.99	19.99
Raspberry Pi B+	single board computer	1	29.99	29.99
ADXL335	triple axis accelerometer	1	14.99	14.99
TSL235	light-to-frequency converter	3	—	—
ATtiny85	slave microcontroller	1	—	—
HC-SR04	distance sensor	1	5.00	5.00
MCC7805	5 V positive voltage regulator	1	—	—
XBee Pro 900 RPSMA	wireless transceiver	2	—	—
M891	leaded NTC thermistor	2	—	—
ActivMedia Pioner 2	chassis	1	—	—
PS-1270	battery	2	—	—
GM9236E132	12 V DC motor	2	—	—
VNH5019A	motor controller	2	—	—
Wombat (PTH)	prototyping board	4	9.95	39.80
Raspberry Pi Camera Board		1	25.00	25.00
FS7548	flex sensor	4	12.95	51.80
TIP122	NPN Darlington transistor	1	0.66	0.66
various resistors, capacitors, wire, etc.		—	—	—
			Total	229.13

Items without a cost are on hand.

Table 6: Bill of Materials

1. Preliminary Work
    (a) Examine problem statement
    (b) Research
    (c) Client interview
2. Zoning In
    (a) Finalize problem statement
    (b) Create full list of objectives and constraints
    (c) Create metrics
    (d) Assign points for metrics
3. Brainstorming design alternatives
    (a) Create morphological chart
    (b) Create candidate designs
4. Design Choice
    (a) Compare design metrics
    (b) Communication with client
    (c) Preliminary testing
5. Prototyping
    (a) Gather materials
    (b) Determine sources and tools
    (c) Delegate responsibility
        i. Building
        ii. Programming
        iii. Testing
        iv. Integration
        v. Documentation
6. Final Report
7. Colloquium Presentation
8. Project Management Activities
    (a) Weekly Meetings
    (b) Progress Reports
    (c) Scheduling
    (d) Source Control

Figure 7: Work Breakdown Structure

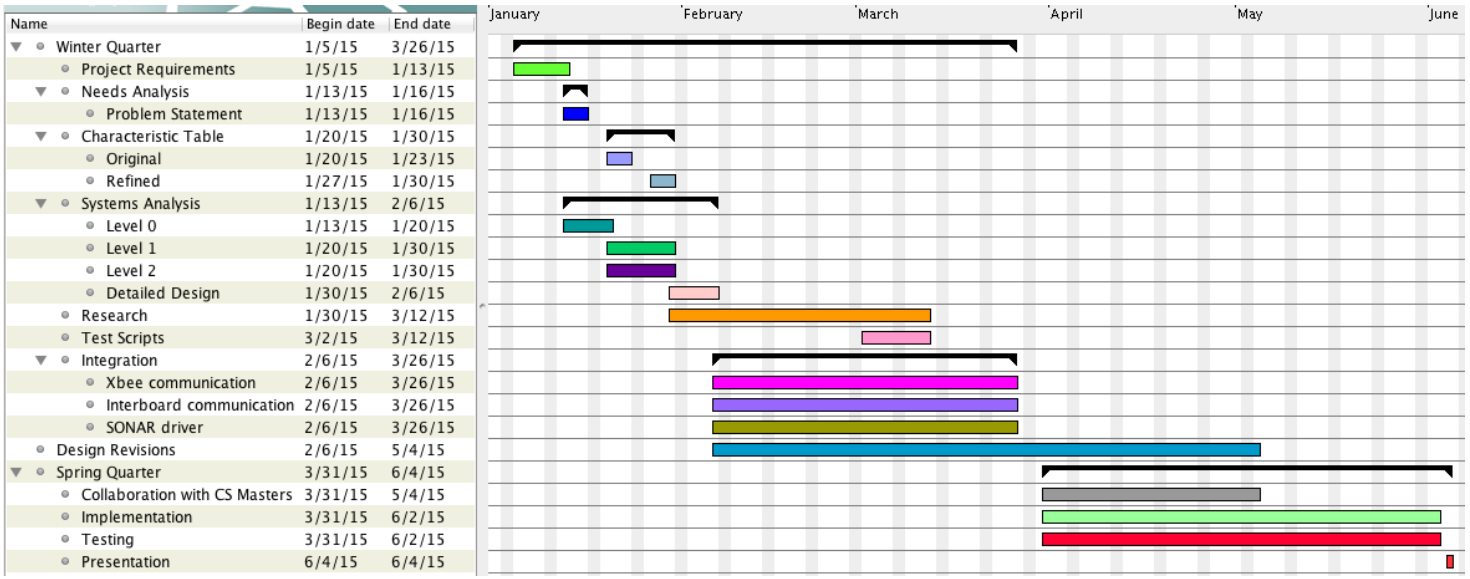

Figure 8: Schedule

10 Research

10.1 Library Search Results

Several published works were researched for relevance to our project.

10.1.1 "A Systems Software Architecture For Training Neural Fuzzy Neural And Genetic Computational Intelligent Networks"[?]

This paper describes a systems approach to neural networking. This is similar to the application for which our platform is being developed. The article in fact references one of Professor Mobus?s presentations from 1994 on the subject. Though not directly applicable to the scope of our project, this article provides context for the system in which our platform will be performing.

10.1.2 "Design of remote data monitoring and recording system Based on ARM"[?]

This article describes the design and implementation of a remote sensor system. The system shares many characteristics of our own project. FreeRTOS is implemented on the ARM architecture to poll sensors and then packetize and transmit data. The SPI protocol is used. Though the targets are different than our own (FAT file system instead of POSIX computer), the fundamental problem of collecting and sending data using FreeRTOS is the same.

10.1.3 "Interrupt aware Queue Implementation for Energy Efficient Multitasking Systems based on Cortex-M3 Architecture" [?]

This article addresses means to optimize power efficiency when implementing a FreeRTOS application on an ARM Cortex-M system. Specifically, a message queue methodology is presented for inter-task communication and resource sharing. This is relevant to our application in that many tasks must run concurrently on our Cortex-M4 microcontroller and they will need to communicate effectively and efficiently on our mobile, battery-powered platform.

10.2 Other Research

The client has written several papers outlining the MAVRIC project history which are hosted on his website `http://faculty.washington.edu/gmobus/AdaptiveAgents/publications.html`. These works provide an historical background on the development of the MAVRIC platform as well as context for its intended experimental research applications.

The documentation from last year's senior design team was also reviewed.[?] This provided insight into the types of problems faced in extending the existing platform and working with legacy hardware components. Specifically, the sonar array presented difficulty for the team. This review allowed for anticipation and mitigation of similar difficulties.

Shane Kwon, graduate of the University of Washington Tacoma's Master of Science in Computer Science and Systems program, shared his research on the MAVRIC.[**?**] This work thoroughly explains the underlying structure of the Adaptrode brain software which will drive the platform. This provides guidance on designing the software interface created for the Single Board Computer.

11 Concluding Remarks

11.1 Vehicles for Continued Learning

The MAVRIC project, as expected of a robotic platform, involves many different aspects of engineering. Inside, there are several avenues for continued learning.

11.1.1 FFT's

Optimal project implementation would utilize software signal processing through an FFT or a derivative algorithm. Due to time constraints, we have opted to utilize a hardware chip that does this for us. We would nevertheless like to pursue this option and field of study further, even after the project is over, to gain mastery that would enable us to do our own signal processing in later projects.

11.1.2 Interboard Communication

Communication between distinct components is a pervasive technical challenge. In the MAVRIC project, we had to research the different options and choose one based on our requirements. Mastery of the different communication methods is an interesting and practical endeavor.

11.1.3 Artificial Agents and Neural Networks

The area of artificial intelligence (AI) is a vast topic with many interesting aspects. The papers that have been written about MAVRIC encourages further exploration of neural networks and the current state of AI.

11.2 Contemporary Issues

Autonomous agents like the MAVRIC robot are rapidly entering our daily lives. Self driving cars[14] and aerial delivery drones[15] are just two examples of intelligent vehicles that will need to operate in a dynamic environment as MAVRIC does. Learning behavior will undoubtedly help these agents accomplish their missions.

[14] `https://plus.google.com/+GoogleSelfDrivingCars/videos`
[15] `http://www.amazon.com/b?node=8037720011`