Archit Enterprises
Team 20
Lab number 1
Risk Analysis
February 6th, 2013
Version 1.0

By signing below, each group member approves of this document and contributed fairly to its completion.

Raymond Tang, Andrew McMillion, Archit Rupakhetee, Tyler Lenig

On our honors, as students of the University of Virginia, we have neither received nor given unauthorized aid on this assignment.

Raymond Tang, Andrew McMillion, Archit Rupakhetee, Tyler Lenig

Contents

Abstract 5
Risk Analysis 6
Controlling the Movement of the Robot 6
Communicating between the Robot Computer and the Base Computer 6
Robot Computer Capabilities — how much software will you be able to run onboard? 7
Data Transmission Speed and Latency — can the link cope with communication protocol's requirements? 7
Sensor Detection Capabilities 7
Conclusion 8

Abstract

This document details possible risks that may arise with any software we implement for our robotics system.

Risk Analysis

Controlling the Movement of the Robot

Problem	Details	Resolution
Motors being calibrated to different settings	Each of the motors likely have different sensitivity and rotate at different speeds even when the inputs are the exact same. Making it difficult to have the robot travel in a straight path.	API allows you to individually set motor speed so we can find the difference in each motors speed and account for the discrepancies.
Wheels become easily uneven	The wheels for the robot tend to easily become misaligned, causing the robot to overturn or under turn as well as create difficulties traveling in a straight path.	Before each run with the robot check to make sure the wheels are even.

Communicating between the Robot Computer and the Base Computer

Problem	Details	Resolution
Robot may sometimes not receive signals	Signals send to the robot are not confirmed so there is a possibility that the robot may not properly receive commands.	Create a confirmation system where the robot is able to indicate or return a signal that confirms that the previous command was properly received. (Note: Bluetooth may already account for this)

Robot Computer Capabilities — how much software will you be able to run onboard?

Problem	Details	Resolution
Physical memory limit	The NXT brick has a limit to how much program we can save in the actual robot.	Unfortunately there is no way for us to increase the actual size in the NXT brick. The best we can do is write optimal code. This includes things such as reusing code as much as possible and writing algorithms in their shortest form. We can also save space by having only basic commands and codes in the NXT and signaling complex commands from the base station.

Data Transmission Speed and Latency — can the link cope with communication protocol's requirements?

Problem	Details	Resolution
Robots response speed to our commands	We need to know how fast the robot reaction time to our commands, how fast the NXT brick is able to process code and how fast we can respond or give consecutive commands.	Testing the robot for each of the conditions and record findings for future references.

Sensor Detection Capabilities

Problem	Details	Resolution
Microphone picking up unnecessary noise from the room	The microphone may pick up normal noise around the room as an instruction and cause error in our programs.	The API allows for selecting a range for the sensitivity of the microphone. We will test and search for the appropriate threshold. If we are unable to find a proper threshold we may resort to muffling the microphone with a cover.

Conclusion

We have come up with a few risks here, however it is not entirely inclusive of all possible risks. We will adapt our software to the situation after evaluating the new risks and solutions with the team.