**Final Project**

**Voice Controllable Mobile Arm Platform**

**Requirement Spec and Schedule**

**Lab Team: Tianshu Bao**

**Shawn Stern**

**Dan Tran**

**Lab Section: EE 478**

**Date: May 14, 2013**

TABLE OF CONTENTS

[1 TABLE OF FIGURES 2](#_Toc356469284)

[2 REQUIREMENT SPECIFICATION 3](#_Toc356469285)

[2.1 System Description 3](#_Toc356469286)

[2.2 Specification of External Environment 3](#_Toc356469287)

[2.3 System Input and Output Specification 3](#_Toc356469288)

[2.3.1 System Inputs 3](#_Toc356469289)

[2.3.2 System Outputs 4](#_Toc356469290)

[2.4 User Interface 4](#_Toc356469291)

[2.5 Use Cases 5](#_Toc356469292)

[2.6 System Functional Specification 7](#_Toc356469293)

[2.7 Operating Specifications 7](#_Toc356469294)

[2.8 Reliability and Safety Specification 7](#_Toc356469295)

[3 SCHEDULE 8](#_Toc356469296)

# TABLE OF FIGURES

[Figure 1 - User app splash screen 5](#_Toc356469297)

[Figure 2 - User app interface 5](#_Toc356469298)

[Figure 3 - Use Case Diagram 6](#_Toc356469299)

# REQUIREMENT SPECIFICATION

## System Description

This specification describes a user-controlled, multi-wheeled robot with a rotating arm for grabbing nearby objects. The robot is to be able to move at reasonable speeds through its environment, with the ability to quickly turn in any direction, and to move forward and backwards in addition to braking. The robot is to be able to detect immediate hazards and obstacles in its forward path and halt, and provide feedback to the user informing them of the event. The robot is to be able to grasp small, light objects on the ground or in the robot’s immediate vicinity for relocation. All robot functions are to be controllable using an android phone application with controls for the motion of the robot, in addition to controls for the arm mechanism itself. The robot is to also be able to accept voice commands as a substitute for direct, physical interaction with the android interface. The robot is to be low cost, flexible, easily reproducible, and intuitive to use.

## Specification of External Environment

The system is to operate in an environment such as a home, ideally with minimal clutter on the floor surfaces on which it will be operating. Room temperature and typical in-house lighting are the expected environment.

The unit will run off of a 9V battery pack attached to the robot chassis for easy swapping in the event of a dead battery.

## System Input and Output Specification

### System Inputs

The system will accept the following external inputs:

* Analog pulse from the distance sensor, 5V with a pulse width of 0uS to 38ms

The system will accept the following commands by means of button presses on the android application screen:

* Move forward (move continuously while button is pressed)
* Move backward (move continuously while button is pressed)
* Turn right (move continuously while button is pressed)
* Turn left (move continuously while button is pressed)
* Brake (Halts ALL robot movement)
* Rotate arm left (move continuously while button is pressed)
* Rotate arm right (move continuously while button is pressed)
* Rotate arm up (move continuously while button is pressed)
* Rotate arm down (move continuously while button is pressed)
* Grasp/Release (toggle button)
* Receive Voice Command

The following voice commands will be usable after pressing the “Receive Voice Command” button on the application:

* Go forward (robot moves until told otherwise)
* Go back (robot moves until told otherwise)
* Turn left (robot moves until told otherwise)
* Turn right (robot moves until told otherwise)
* Brake (robot halts ALL motion)
* Arm left ‘x’ degrees
* Arm right ‘x’ degrees
* Arm up ‘x’ degrees
* Arm down ‘x’ degrees
* Grasp
* Release

### System Outputs

The system will report the following information on the user interface via serial interface over Bluetooth:

* Current direction of travel (forward, back, turning, braked)
* Arm status (rising, lowering, grasped, open)
* Distance to nearest large object in front of the robot in cm
  + Range from 0 to 4000 ± 1cm

## User Interface

The user will be able to continuously control the robot and query its sensors for data using a simple android touch interface or vocal commands. The following buttons are available:

Robot Direction:

* Directional arrows. Robot moves (relative to its current orientation) in accordance with the held arrow until it is released.
* Brake. Robot halts all motion.

Arm Control:

* Directional arrows to control the orientation and elevation of the grasper at the end of the arm. Arm moves as long as the appropriate button is depressed or until the arm reaches the limits of its rotation.
* Grasp/Release toggle. Pressing the button causes the grasper to switch to the opposite state (from grasping to releasing the object, and vice versa)

Activate Voice Control:

* Press button to launch voice prompt to interpret voice commands.

Measurements and robot status information will be displayed in the top left corner of the android interface as text through a serial interface over Bluetooth.

The android interface will appear as follows: upon loading the application, the user will be presented with a splash screen while the Bluetooth pairing and initialization of the app occur. The application interface itself will have an upper set of buttons for directional control with a central button for sending voice commands. The lower set of buttons will operate the arm assembly, with the center button serving as the grasp/release toggle.

|  |  |
| --- | --- |
| E:\Dropbox\EE\EE 478\478 Spring 2013\Final Project\Deliverables\Splash Screen.png  Figure - User app splash screen | E:\Dropbox\EE\EE 478\478 Spring 2013\Final Project\Deliverables\Interface_edited.png  Figure - User app interface |

The field where “Voice command” is listed in the location where textual status information will be displayed.

## Use Cases

The cases are given in the below diagram:



Figure - Use Case Diagram

***Control Robot***

The robot chassis will move in accordance with user input. If the Forward button is held down, the robot will continue to move forward in a straight line until the button is released or if an object is detected as too close (robot will automatically brake and the only available movements will be rotation or moving backwards). This system is controlled by the upper set of arrows in the top half of the android interface.

Exceptions: If terrain is too cluttered, the robot may not be able to maintain true, straight motion as its path will be disrupted by debris. If Bluetooth connection is lost or Arduino loses power, error will occur.

***Control Arm***

The arm assembly will move in accordance with user input. If the user holds the Down button, the grasping end of the arm will lower itself closer to the ground until either the user releases the button, or if the grasping end reaches the limit of its lowering range. The grasping end of the arm can be commanded to either close its grip or release it. This system is controlled by the lower set of arrows and central button in the bottom half of the android interface.

Exceptions: If the object is too heavy or too large, the arm assembly is not guaranteed to be able to properly grasp the object, or to be able to lift it without potentially destabilizing the robot. If Bluetooth connection is lost or Arduino loses power, error will occur.

***Voice Control***

The user interface will support the conversion from spoken words in English to their respective commands as needed to control the robot and/or arm assembly as appropriate. For example, the user may instruct the arm to “Arm up 20 degrees” at which point the robot arm will rise such that the difference from its previous position and new position is 20 degrees, within tolerances. The mode may be activated by pressing the speaker icon in the center of the user display.

Exceptions: If user does not speak clearly with proper enunciation, their statement may be misinterpreted. If Bluetooth connection is lost or Arduino loses power, error will occur.

***Measure Distance***

The robot will return the distance from the front section of the robot to the nearest object in front of it that is at least .5sq ft in dimension. The button for this function is still under design consideration.

Exceptions: If an object is smaller than .5sq ft. there may be an error in detecting the object correctly. If Bluetooth connection is lost or Arduino loses power, error will occur.

## System Functional Specification

The system is intended to operate and move around in the environment based on user input, with the input from the distance sensor as a safeguard from collisions. A measurement system will be implemented to convert the output pulse of the distance sensor into a user-accessible distance value, selectable as either inches or cm. This distance value will also be used directly by the robot for safety reasons to halt robot movement if it is in danger of a head-on collision.

The user will be able to select either continuous motion of the robot (moves until stopped) or motion that only lasts as long as the respective button is being held down.

## Operating Specifications

The system shall operate in a typical household environment:

Temperature Range 40 – 120F

Humidity up to 90%

Power 9V battery

The system shall operate for a minimum of 5 hours on a fully charged 9V battery

## Reliability and Safety Specification

MTBF: Minimum 2,500 hours

# SCHEDULE

