



Team Smarker Final Project Report

Supervisor: Professor Amr Elkady

Author:
Harshavardhan Srinivas
Ting-yu Yeh

March 21st, 2019



COEN 315: Web Architecture & Protocols

Introduction	2
Abstract	2
Summary	2
Project Objective	2
Purpose	2
Background	3
Design	3
Hardware	3
C. Software	8
1. Firmware design and Implementation	8
2. Front End Design and Implementation	9
3. Back End Design and Implementation	9
4. Exposed API list	9
D. User Interface	10
Result	11
Conclusion	11
Appendices	13
Hardware/Software Inventory	13
Hardware Details of all hardware used, model numbers, pictures, references, etc.	13
Mobile App What software packages/frameworks/databases/APIs etc. was used	13
Backend & Web App What software packages/frameworks/databases/APIs etc. was used	13
Execution Instructions	14
Hardware setup	14
Front end	14
Backend	14



Introduction

Abstract

This paper describes an intelligent lightweight drawing robot which allows remote writing. The robot proposes a small-scale agile bot that can climb on smooth magnetic vertical surfaces. In the local environment, the participant decides on an arbitrary character to transmit. Our robot placed in the remote environment writes the word on a whiteboard in real time. Due to its compact design, a high degree of miniaturization is possible while writing. It has onboard power, computing, and wireless communication, which allow for semi-autonomous operation. Various aspects of a functioning prototype design and performance are discussed in detail, including the theory of mobile robotics and bot control. The robot is intended for a variety of fields, such as education and entertainment.

Keywords: Human-Robot interaction, Mobile Robotics, Miniature Robotics, Mechatronics, Remote Writing.

Summary

Based on the movement of wall-climbing rolling bots, the architecture of our robot was designed and built, including the adhesion mechanism, the mechanical architecture and the anti-toppling mechanism. Magnetic adhesion mechanism and tracked locomotive mechanism were employed in our wall-navigating robot. The robot has the main advantage of being compact and mechanically simple.

Project Objective

Purpose

Mobile robots with the ability to navigate on horizontal and vertical surfaces have the potential to solve a multitude of problems. One notable example is the Sky Cleaner [1], a climbing robot for glass-wall cleaning. In general, climbing robots use one of three types of attachment mechanisms:



magnetic attraction [2], vacuum suction [3], or grasping mechanism [4], [5]. Each of these mechanisms has its own set of merits and demerits. Suction-based robots rely on a complete seal with the surface, making cracked or non-smooth surfaces problematic. Similarly, clawed and grasping robots cannot climb smooth surfaces. To avoid these drawbacks, the robot presented in this paper is designed to ultimately utilize a magnetism-based adhesion mechanism.

Background

This paper presents Smarker, a semi-autonomous miniature robot theoretically capable of navigating on any smooth magnetic surface of any orientation. The robot is actuated by two stepper motors, each controlling one wheel, and one servo motor to control the pen/marker attached to it. Smarker is able to make smooth turns, and is able to withstand its own weight in entirety with ease using Buckyballs [6]. The next section of this paper will describe the rationale behind our architecture, followed by the design and implementation of our robot.

Design

The Smarker system consists of multiple building blocks, including physical robots (Smarker Robot), mobile and web interface (Smarker UI), and a cloud infrastructure (Smarker Alert).

Hardware

The main feature of the Smarker robot is the ability to travel on magnetic surfaces without restriction. Rare earth magnets are chosen as the key components for their high strength-weight ratio. Each 5-mm-diameter buckyball could exert roughly 200-gram-force and only weighs about 1 gram (Figure 1). Each Smarker robot is installed with 12 buckyballs to ensure static and dynamic equilibriums under various configurations.



Figure 1: Rare-earth magnetic buckyballs

The embodiment of a Smarker robot consists of a bottom plate, a motor housing base, a servo fixture plate, and a marker fixture. The bottom plate, shown in Figure 2, comprises 12 compartments for hosting buckyballs. The cross-section profile of the compartments are designed with leeway for buckyballs to rotate freely while the Smarker robot is in motion. The bottom plate also comprises 6 extruded pins engaging bores on a baseplate, shown in Figure 3. The baseplate, also known as the motor housing base, consists of a circular plate which adheres to the bottom plate and 2 motor brackets designed to fit stepper motors. The drive shaft of each stepper motor is coupled to a customized wheel with corresponding key-hole, shown in Figure 4. To ensure no-slip assumption of the Smarker robot under any circumstances, the wheels are mounted with high-friction tires made of #83 rubber bands. The width of wheels are designed to optimized performance of different control maneuvers.

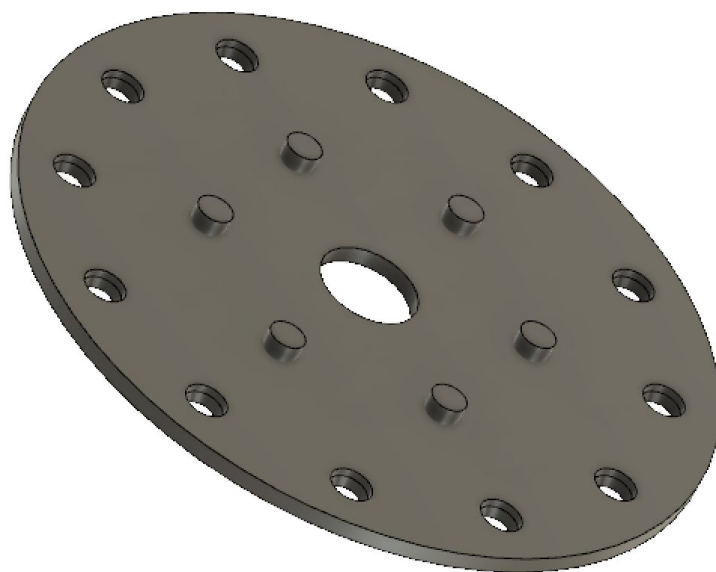


Figure 2: Bottom plate of Smarker Robot

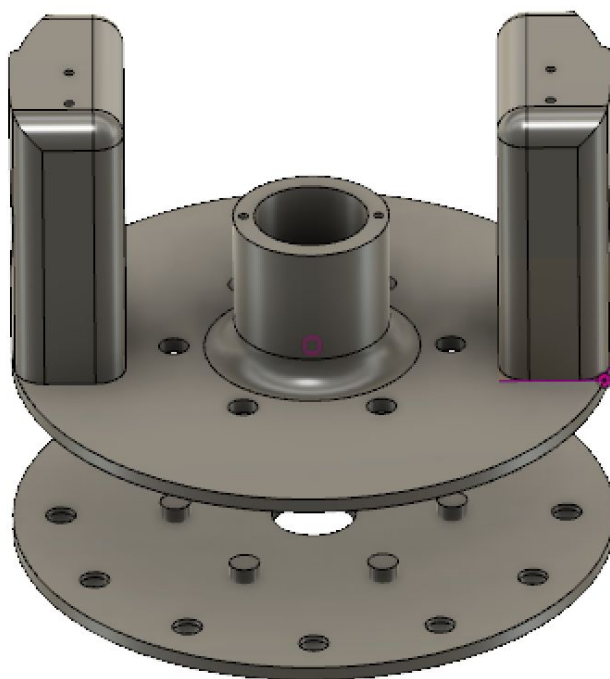


Figure 3: Assembly of bottom plate and baseplate

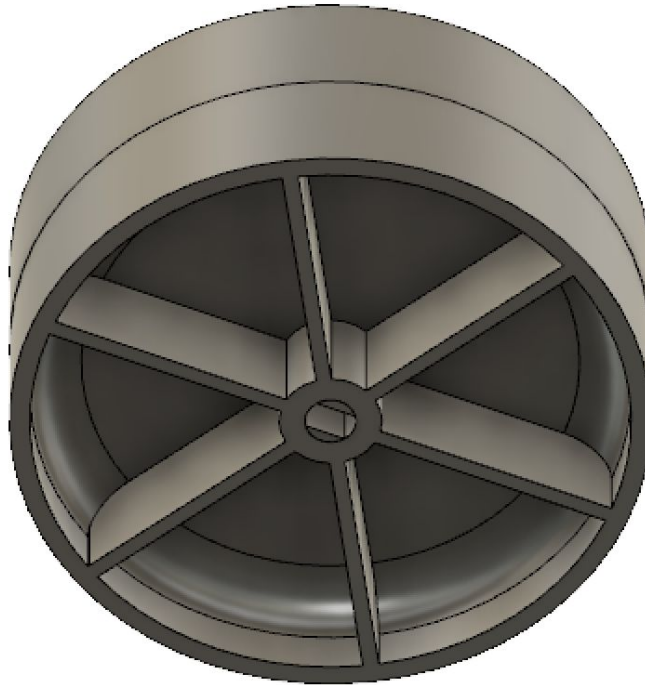


Figure 4: Wheel of Smarker Robot

The top of the motor brackets are drilled and tapped to install m2 screws, connecting a servo fixture plate, shown in Figure 5. The servo fixture plate is used for two purposes, to guide a linear motion of a installed marker and to mount a servo motor to control the linear motion. The fixture plate comprises of two pin holes around the central bore to accommodate connecting rods from a movable marker fixture, shown in Figure 6. Another embodiment of this mechanism could make use of linear bearings to facilitate smooth motion. The movable marker fixture is attached to the servo motor through slot-pin joint which converts rotary to linear motion.

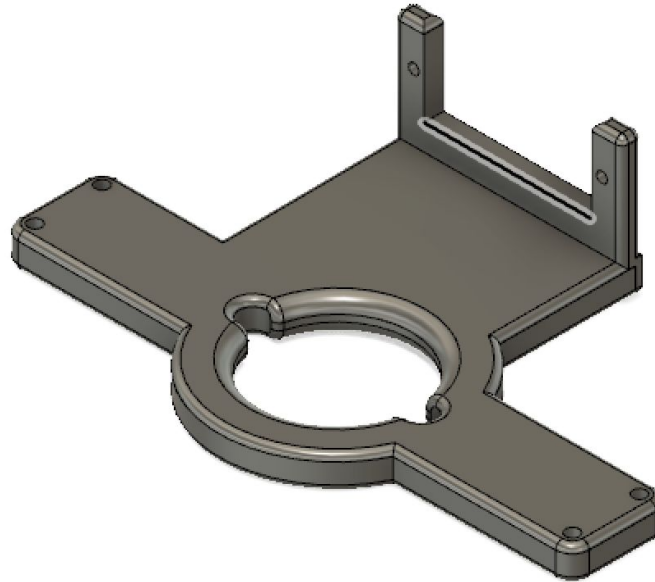


Figure 5: Servo fixture plate of Smarker Robot

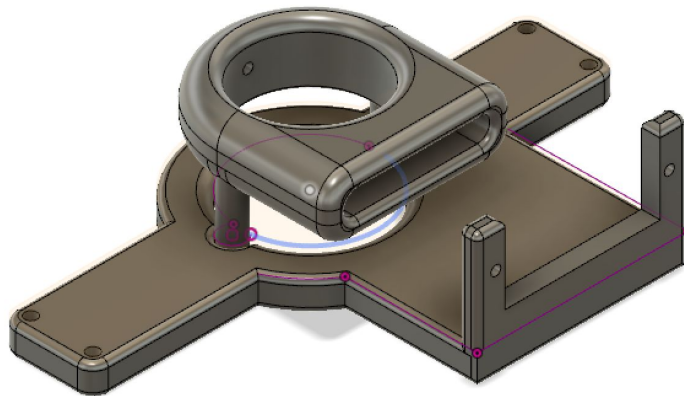


Figure 6: Assembly of servo fixture plate and movable marker fixture

The electrical components are selected to ensure smooth operation while minimizing the total weight. Stepper motor model 28BYJ-48-5V is used to drive the wheels. It has a gear ratio of 64:1 with a



rated voltage of 5V, consuming 500 mA of current when driven with ULN2003 chips. The ULN2003 chip encapsulates high-voltage-high-current Darlington transistor arrays, which are specifically designed to drive stepper motors. In addition, flyback diodes are included to eliminate voltage spikes when inductive loads are connected.

To control the movable marker fixture, servo motor MG90S is used to engage and disengage marker from the writing surface. This model is lightweight with high output torque which consumes roughly 300 mA under 5V. The metal gearing also adds strength and provides wear resistance.

C. Software

1. Firmware design and Implementation

To drive the Smarker Robot along a predefined path consists of waypoints, a kinematic model has to be implemented to generate wheel control. Since the Smarker Robot is a two-wheel differential drive robot, it is also a non-holonomic system. Due to the nature of non-holonomic systems, there exists infinite solutions to control the Smarker Robot from each waypoint to the next. Therefore, the control of the wheels requires further investigation in path planning optimization. Currently, the firmware algorithm calculates the derivatives of given path and generate wheel angular velocities based on the information and other inputs, including operating speed and mechanical measurements.

We used AccelStepper[7], an open source library, to control the wheels smoothly. The library provides interface to control stepper motor behaviours with various parameters, including speed and acceleration. In addition, the Smarker Robot is driven with full step control to maximize the torque. To ensure the synchronization between the two wheels, MultiStepper library is also implemented to coordinate wheel motions. MultiStepper is an interface to package multiple AccelStepper into a single object. The purpose is to drive multiple stepper motors proportionally along the path.



COEN 315: Web Architecture & Protocols

2. Front End Design and Implementation

For our front-end, we have designed an HTML5 Canvas based model that can record and transmit user's inputs in the form of drawings by converting them into coordinates.

3. Back End Design and Implementation

For programming our backend, we utilized Arduino IDE and integrated the scripts with our ESP32 chip. The key role of our backend was to send through the gradient of coordinates as inputs to our motors attached to our chipset in such a way that our bot moves smoothly. In addition to our on-board chipset, we've also built a Django-based cloud API that performs real-time monitoring on our robot.

4. Exposed API list

No other external APIs were utilized in the development of our system. However, certain external libraries were used to perform the integration between Arduino, our chipset, motors and our servo. We utilized libraries AccelStepper.h, MultiStepper.h and Servo.h to do the same.



D. User Interface

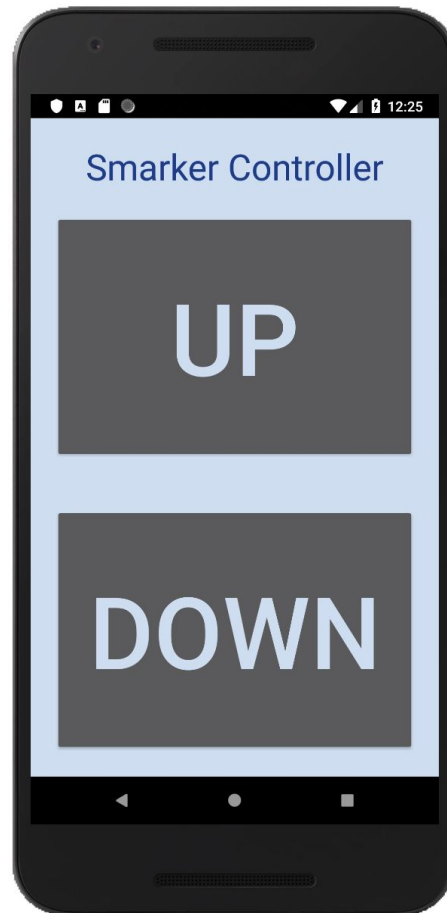


Figure 7: Mobile user interface



Figure 8: Webapp user interface

Result

This paper gave the details of the proposed remote writing robot that can navigate on horizontal and vertical surfaces by using the concept of magnetic adhesion. By static and dynamic force analysis of the robot, design parameters about magnetism and motor torque are obtained. Finally, the embedded system, control architecture, wireless networking, and web-based architecture was designed.

Conclusion

A remote writing robot design was analyzed and implemented. Design criteria were established to guide our magnetic adhesive power selection. Finally, a semi-autonomous prototype was designed



and fabricated, which is able to climb on magnetic smooth surfaces of various orientations. The system allows a user to select by thought arbitrary characters and send them to our robot, which receives the command via the network and writes the characters on the whiteboard in real-time. Future research should explore the viability of the system in terms of usability, path planning, position error, battery power saving.

References

- [1] Houxiang Zhang et al., "Sky Cleaner 3: a real pneumatic climbing robot for glass-wall cleaning", IEEE Robotics & Automation Magazine, Volume: 13 , Issue: 1 , March 2006.
- [2] J. Grieco, M. Prieto, M. Armada, and P. Gonzalez de Santos, "A six-legged climbing robot for high payloads," in Proc. Int. Conf. Control Appl., 1998, pp. 446–450.
- [3] S. Ryu, J. Park, S. Ryew, and H. Choi, "Self-contained wall-climbing robot with closed link mechanism," in Proc. Int. Conf. Intell. Robots Syst., 2001, vol. 2, pp. 839–844.
- [4] S. Kim, A. T. Asbeck, M. R. Cutkosky, and W. R. Provancher, "Spinybot II: Climbing hard walls with compliant microspines," in Proc. Int. Conf. Adv. Robot., 2005, pp. 601–606.
- [5] T. Bretl, S. Rock, and J.-C. Latombe, "Motion planning for a three-limbed climbing robot in vertical natural terrain," in Proc. Int. Conf. Robot. Autom., 2003, pp. 2947–2953.
- [6] Neodymium magnet toys, From Wikipedia, the free encyclopedia
https://en.wikipedia.org/wiki/Neodymium_magnet_toys
- [7] AccelStepper and MultiStepper library
<https://www.airspayce.com/mikem/arduino/AccelStepper/classAccelStepper.html>



Appendices

Hardware/Software Inventory

Hardware Details of all hardware used, model numbers, pictures, references, etc.

Table: Bill of materials

Part Name	Description	Quantity	Price
PLA	Prusa PLA	170 g	Free (Maker Lab)
Stepper Motor	28BYJ-48-5V	2	\$4.95 (Adafruit)
Breakout Board	ULN2003	2	\$1.95 (ebay)
Servo Motor	MG90S	1	\$9.95 (Adafruit)
Rubber Bands	#84	1 pack (45 ct)	\$4.50 (Staples)
Bucky Balls	5 mm	1 pack (125 ct)	\$19.68 (Online)

Mobile App What software packages/frameworks/databases/APIs etc. was used

HTML, CSS, Javascript, PhoneGap, Native Android

Backend & Web App What software packages/frameworks/databases/APIs etc. was used

Django==2.1.7

pytz==2018.9



COEN 315: Web Architecture & Protocols

Execution Instructions

Hardware setup

Print the STL files in the attachment and tap corresponding holes with M3x0.5 and M4x0.7 taps. M2 holes can be self-tapped with machine screws if the parts are printed in PLA. Final parts are assembled with M2, M3, and M4 screws. The bottom plate and baseplate are assembled through press fit. The wheels are connected to motor shafts through press fit as well, and the motors should fit loosely in the brackets, secured with M4 screws.

Front end

Connect to the ESP32 onboard server and access the UI through regular browser.

Backend

Deploy the backend Django API behind an Nginx instance using gunicorn and supervisor.

Follow the below commands to set up the services:

```
$ pip install -r requirements.txt
$ python manage.py migrate
$ python manage.py runserver
```