Beethoven Bot

Oliver Chang

3/20/2014

University of Florida

Department of Electrical and Computer Engineering

EEL 4665-IMDL-Written Report 2

Instructors: A. Antonio Arroyo, Eric M. Schwartz

TAs: Josh Weaver, Andy Gray, Nick Cox, Daniel Frank

Contents

Abstract

Beethoven Bot is a mobile autonomous transcribing robot. As a casual pianist, one of the most frustrating aspects of playing piano is finding music sheets to recent songs. Most sheet music available for purchase consists of classical music or old classic songs. Finding recent or popular songs that are heard in random situations is a difficult task as the music may not be readily available or already transcribed online. In order to solve this issue, the Beethoven Bot was designed.

This autonomous robot is designed to listen to audio tones through a microphone, detect which notes have been played, and transcribe the notes onto staff paper. The audio will be received through a Bluetooth microphone. The note is then determined through frequency analysis of the recorded music. The robot then finds the location of the note in the treble clef section. The note is then written at the correct location and the robot proceeds to the next note.

Introduction

As a child I grew up taking classes to learn how to play piano. I greatly enjoyed playing music however all of the music that I played and had was classical pieces. I desired to played music featured in current movies, played in radios, or heard in video games. The issue was that the sheet music at musical shops was mostly classical songs and finding sheet music for these recent pieces were far and few between. In order to overcome this problem, the Beethoven Bot was created. The original Beethoven Bot was an open-loop, differential drive platform with a drawing mechanism at the center powered by a 12 V battery and controlled through a microcontroller and small computer. Since the initial Beethoven Bot's electret microphone had extremely low range of sound detection, the new version consists of using a Bluetooth microphone.

This report will explain the design of both the mechanical and digital aspects of the Beethoven Bot. There are two main brains to this robot. The Raspberry Pi will listen to the audio tones and analyze the sound input to calculate the notes played through fast fourier transform. The Arduino ATMEGA 2560 receives the notes and moves the robot accordingly and activates the drawing mechanism to transcribe the note.

Integrated System

The Beethoven Bot uses a Raspberry Pi for high level decisions and an Arduino ATMEGA 2560 for low level decision making. The bump switches, ultrasonic sensors, and camera provide inputs to be incorporated into the controller's computations. The melody will be taken in from the microphone analyzed with the raspberry pi to be sent to the Arduino for transcription.

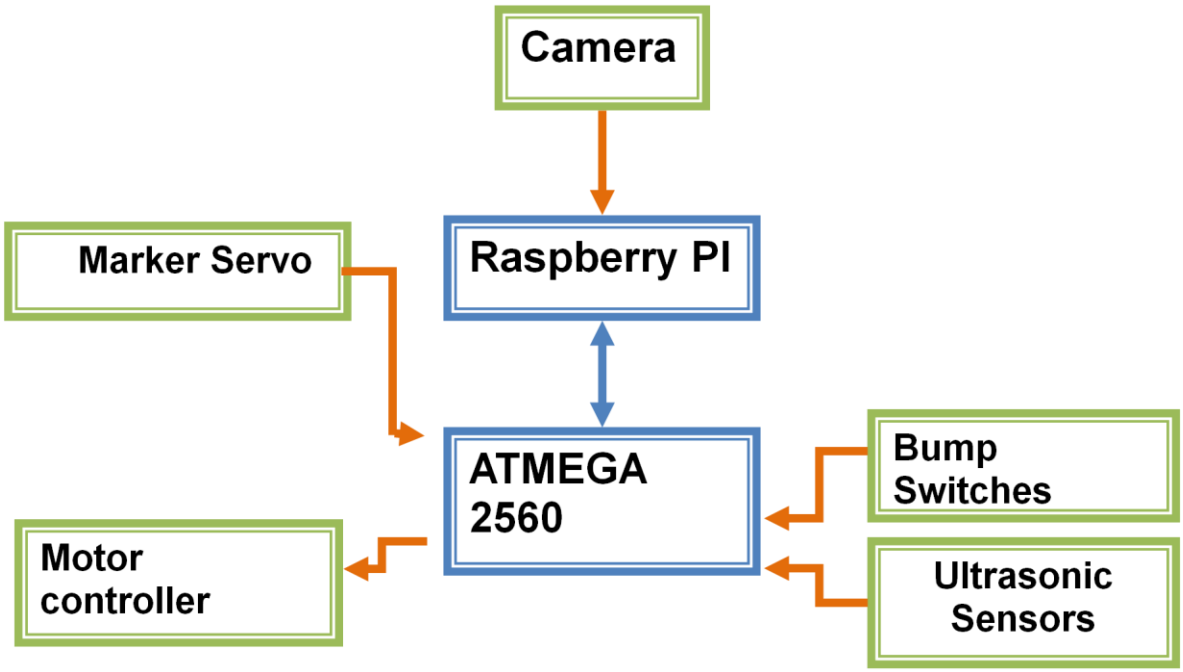

Figure 1: Beethoven Bot Block Diagram

Mobile Platform

A two wheel differential drive system will be utilized along with a caster for stability. The robot is designed to be relatively small is relation to the potential area in which it is able to draw. The robot has a writing mechanism, which will hold the marker down in place while drawing, but lift it to prevent undesired markings. The motors are mounted to metal mounts placed underneath the robot. In order to maintain proper drawing formation, the caster must lift the platform to be level with the height at the wheels. The platform is made with airplane grade wood machined in lab. The components are bolted to the base platform with the exception of the Raspberry Pi and camera which will be bolted to the raised platform to give the camera the proper view point.

Actuation

Two Pololu stepper motors drive the robot. The motors are differentially driven to allow turning in-place. The stepper motors do not have an encoder, but are however accurate enough to drive in a straight line without much error. Two Pololu motor controllers provide the H-bridges to drive the motors forward or backwards by the given steps. The Arduino transmits to the motor controllers using pulse width modulation.

For a writing mechanism, a stamp marker is pressed down through a weight and lifted through a servo, which will rotate up to 90 degrees to lift the marker above the floor surface. This action is also completed through the use of pulse width modulation.

Sensors

Collision detection is implemented through the use of three bump sensors. These three bump sensor are placed at the front, left and right side of the robot since the robot's main movements will be forward and rotating right or left to reach the next line in the drawing grid. 5 volts power the switches and connect to an interrupt through the switches with a pull down resistor. The switch arms are extended into paddles to increase their range of detection. The three ultrasonic sensors allow distance measurements to locate and determine the boundaries of the drawing area. These sensors additionally function for object avoidance when an object breaks the threshold distance of the robot's path, the robot will wait until the object is removed.

Through the use of a webcam and OpenCV, the lines of the sheet music is be recognized and after determining what note is heard, the robot moves to the appropriate vertical location on the sheet and write a quarter note. Then the robot moves to the next location for a quarter note to be written. The Raspberry Pi camera is mounted to see downward onto the staff sheet and transmit to the Raspberry Pi information through thresholding the seen lines.

Behaviors

The Beethoven Bot has two modes of operation. The first being a listening mode, in which the robot will listen for a specific note and then begin reading the melody that it hears. The music heard is then calculated on the Raspberry Pi through FFT and the notes are sent serially to the Arduino to send the robot to the second mode. The second mode is the drawing mode in which the drawing area is determined and the drawing begins through the use of the marker servo mechanism. The robot will go from section to section of the sheet music to write the notes one at a time.

Experimental Layout and Results

Points of experimentation include the ability of the robot to drive straight through feedback from the ultrasonic sensors, how well the camera can see the music lines and how accurate the note placements are. If the motors are not able to maintain a straight line or a 90 degree turn, a gyro may be added for improved accuracy. One option to write the notes is to stamp the note down or to draw the note with a marker and circular motion. Initial testing has proven that using an electret microphone is not possible in this operation due to the limited hearing range of the microphone. This was tested by computing the frequency of generated tones into the Arduino and moves the speaker farther away until the sound was lost, both to the Arduino and an oscilloscope. Thus a Bluetooth microphone and dongle were purchased to add

greater flexibility and range to the robot. This additionally removes much of the noise that would be added to the omni-hearing electret microphones.

Conclusion

The realistic goal of this project is to create a robot that will be able to transcribe simple tone melodies onto sheet music. The limitations that follow this work are the complexity of the melody to be transcribed, how well the music is heard and transcribed through the Raspberry Pi, and how well the robot performs in locating the note position. The robot is able to perform object avoidance through the use of the ultrasonic sensors, however there is little need for the robot to avoid objects expect at the point in which the robot moves to the next note location. The motor controller and power circuits were designed as PCBs machined on campus to create a cleaner, more professional design.

The sound analysis as the special system waits until a sound above the noise threshold is heard before beginning to record the music. After which the music is saved as a wav file and analyzed in python to determine the frequencies present through FFT. The noise values are disregarded and the frequency is averaged and the note is determined from that averaged frequency. Current issues include random spikes of noise may trigger the recording along with the current code which calculates the value of only one note and the wav file varies in length.

.

Documentation

- ATMEGA 2560 Datasheet: http://arduino.cc/en/Main/arduinoBoardMega2560
- Raspberry Pi: http://www.raspberrypi.org/
- Pololu Motorcontroller: http://www.pololu.com/product/2133
- Project Website: http://ochang.weebly.com/

Appendices

- Code can be found in project website: http://ochang.weebly.com/