Carrom Playing Robot

Team ID

80

Team Name

Here for Beer!!

Team Members

- Udayan Joshi
- Pranav Kulkarni
- Saurabh Kolambe
- Anand Naik

Link for Project Abstract

https://docs.google.com/document/d/1_X91nt21NaHV6yF4IyeWEEpfHFuQ524Hx2wxHQo82OA/edit

INTRODUCTION

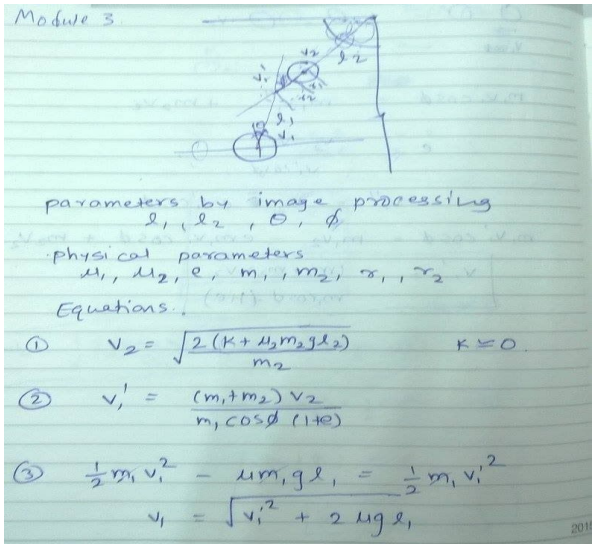
Our project is to build a carrom playing bot,which will attempt shots pocket coins at a basic level. We have attached a camera above the carrom board which will take image of the carrom board before playing shot. We have used Raspberry-Pi 2 which compares the image taken by the camera before each shot with a reference image and thus determines the coordinates of the coin in term of pixels in the image. The data obtained from the images is used to determines

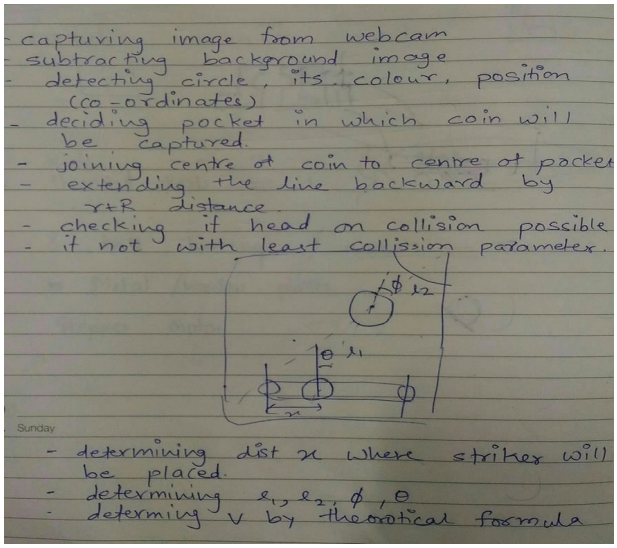
position of striker, angle and speed of the shot to capture the coin. And finally the execution of actual shot is done using combination of steppers and dc motor controlled by the Raspberry-Pi.

MODULES :-

1. ALGORITHM

The image shows final results of theoretical calculations using Newtonian Mechanics.





The image shows the basic algorithm developed on paper.

2. MECHANICAL PART

1. Horizontal mechanism

We have used a Timer belt mechanism for horizontal movement of the slider. Initially we planned on making a lead-screw mechanism, but

due to low pitch of lead-screw, the horizontal movement was slow, so we changed the mechanism. Our new mechanism has a stepper motor to drive a timer belt over two pulleys. The ends of the timer belt are fixed to the rotational module so that the rotational module slides with the timer belt.

Problems faced:-

1. Rods get easily rusted in monsoon weather.
2. Rods also sag due to weight attached on them.
3. Stepper motor also stalls sometimes because of heavy load.

2. Rotational mechanism

We have used another stepper motor to rotate the entire striking mechanism. We didn't face much problems in this module. The slider on which the rotational mechanism is attached is not exactly vertical due to unbalanced center of mass.(We can't balance the centre of mass by attacking extra mass as it will cause problems in horizontal movement like sagging)

Problems faced:-

1. Unbalanced CoM
2. Proper alignment of shaft of stepper motor with centre of strip on which striker is placed is tough

3. Striking mechanism

We used a DC motor to hit the striker. We attached a arm to the shaft of the DC motor which rotates and eventually hits the striker.The main problem in this module was the speed of the DC motor. We had to increase the length of arm to get more speed at the tip as compared to an arm of smaller length, due to which we had to change the position of the DC motor with respect to the stepper motor used in

rotational module. We had to change the height of both DC and Stepper motor by shifting the whole mechanism up.

Problems faced:-

1. Speed of DC motor is insufficient to make a decent shot.We have used a DC motor with 1000 rpm but still we cannot make rebound shots.
2. DC motor vibrates heavily due to unbalanced weight of arm. We had to attach extra mass at the other end to balance CoM.

3. CODING IN PYTHON

Python is a real easy programming language for processing huge data in form of images. OpenCV is used to process image in python. Using image processing we can detect the circles and their centres in the image and also can distinguish them as coin, striker, pocket by comparing their radii. We build a python code to process the coordinates of coins obtained by image processing to calculate required parameters like speed, position and angle at which striker should be placed to take a decent shot. All the parameters returned by this python code are fed to stepper motors which aligns the bot at required position on the striking strip.

Python tutorial playlist
https://www.youtube.com/playlist?list=PLEA1FEF17E1E5C0DA
(This is one of the many good python 2.7 playlists on youtube. This playlist is extremely beginner friendly and can be watched even by those who are starting to code.)

Byte of Python free PDF book
http://www.ibiblio.org/swaroopch/byteofpython/files/120/byteofpython_120.pdf
(A good book)

Problems faced:-
Installation of different libraries.

4. IMAGE PROCESSING

As name suggest, it processes on the image and returns required parameters. In our code we used IP to compare the images captured by by camera before each shot with the reference image of a blank carrom board. Then we subtract this two images to highlight the extra objects in the new image i.e. coins. We can measure the radius of the coins in terms of pixels in the image and then converting it to normal scale using appropriate multiplier. We also detects the circles in the reference image i.e pockets. After getting coordinates of coin and pockets data is fed to python code which then determines most appropriate shot and calculates required parameters for the same.

Background subtraction

(These are simple member functions of openCV library.They are really easy to implement)

Colour detection

(Applying different color filters and detecting colors.)

Python program for detecting circles

(Detection of circles with their centres and radius)

Hough's circle transform

(Houghs Circle Transform is the only readymade transform for detecting circles efficiently. However , there are 10 different arguments for this function and we have choose values correctly)

Co-ordinates

(There are predefined co-ordinates in any image. They are a bit different from our normal cartesian co-ordinates. The only difference is that the y-axis is inverted)

5. RASPBERRY PI

(1)Flashing up the SD Card

https://www.raspberrypi.org/documentation/installation/

(2)Booting up RPi for first time and installing OpenCV 3.0.0

https://www.youtube.com/watch?v=6j-Wy9j0TCs

(The video shows both-First Boot Up of RPi as well as installing OpenCV 3.0.0 on RPi.

It is strongly advised to follow this video step by step)

(OpenCV 3.1.0 is unstable and hence, not used.)

(3)GPIO Pins of RPi

These pins can be used to connect some external hardware to RPi.

It is important to understand the numbering scheme of GPIO Pins.

BCM Numbering Scheme of GPIO Pins

https://www.raspberrypi.org/documentation/usage/gpio-plus-and-raspi2/

GPIO library of python for RPi-This library provides very easy access to GPIO pins of RPi.

Basics of how to use GPIO Pins on RPi can be learnt from this set of 9 videos.

https://www.youtube.com/watch?v=OR5h0UnMcUE

(4) RPi 2 doesn't have an inbuilt wifi adapter.So it is preferable to use RPi 3.One needs to buy a suitable usb wifi adapter for RPi 2(if one wants to control RPi remotely when accessories like keyboard, mouse, monitor are not available.)

(5) Connecting to RPi remotely(using internet)

Using SSH

https://www.raspberrypi.org/documentation/remote-access/ssh/

There is no need to install any additional software for Ubuntu Users as they can use SSH

directly from command line.
Windows Users need to install Putty and it is recommended to install XMing and XMing Fonts.

To get RPi desktop on laptop one can use Tightvnc software
https://www.raspberrypi.org/documentation/remote-access/ssh/

A4988-DRIVER FOR STEPPER MOTOR

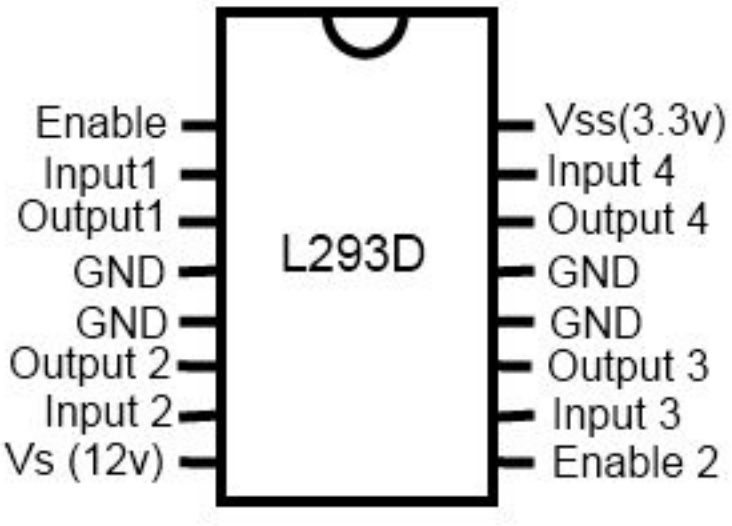
The link below gives complete information about the driver-Pinout and how to use it with a four wire stepper motor.
It also gives a code to use stepper motor with arduino.
Basic principle of this driver is that the motor goes forward one step when the input a step pin changes from LOW to HIGH.

http://www.instructables.com/id/Drive-a-Stepper-Motor-with-an-Arduino-and-a-A4988-/

It can be easily extended for RPi.

L293D-DRIVER FOR DC MOTOR

The Pinout Diagram is self-sufficient



For some more summarised information see link below
http://www.engineersgarage.com/electronic-components/l293d-motor-driver-ic

PLAN OF ACTION:-

Week 1:

Things planned:
Install python and openCV and other required libraries.
Learning functions and syntax of python and openCV.
Finding algorithm for calculating required parameters like angle ,position of striker.
Work on Mechanical modules.
Things accomplished:
Installed all required libraries and tried basic codes.
Developed algorithm for finding speed angle and position.

Week 2:

Things planned:
Controls and functions of raspberry pi will be learnt.
Raspberry-pi and stepper motor interface understanding.
Electrical connections will be decided.
Work on Mechanical arrangement will be continued.
Things accomplished:
Learnt the interface between stepper and R-Pi
Controlled stepper motors using R-Pi

Week 3:

Things planned:
Basic code is developed and debugging starts.
Electrical connections will be decided.
All the mechanical components should be ready.
Integration of the modules starts.
Things accomplished:
Had to change sliding mechanism from leadscrew to timerbelt.
Mechanical components were ready individually..
Tested horizontal and rotational motion with Stepper.

Week 4:

Things planned:
Finish integrating the modules and testing starts.
If everything works as per expectation for one piece and time permits, we will modify the the algorithm for more
Pieces.
Things accomplished:
Integration of mechanical parts.
Tested whole mechanical part and solved stepper motor faults.
Tested striking mechanism.

Week 5:

Things planned:
We will keep last week as buffer and do thorough testing and improvisation.
Things accomplished:
Integrated all modules.
Tried running codes for different positions of coins.
Captured videos of bot taking actual shot.

WORKING:-

- Capturing image of blank carrom board
- Detecting coordinates of pockets and getting the perspective view of the carrom board
- Capturing image of carrom board with coin.
- Getting perspective transform of the image same as that of the blank image.
- Feeding this two images to R-Pi for actual IP.
- R-Pi will subtract this two images to highlight the coin and then detect the coordinates of the coins in the new subtracted image.
- The python code will takes this coordinates as input and process them to find the speed position and angle of easier and optimum shot.
- The returned values from the above code are given to other code which drives the stepper and dc motors.
- Stepper motors drive the bot to the desired position and align the striking part in direction of hitting.
- Stiker is placed manully at the position marked by the bot.
- DC motor is then enabled which will rotate the hitting arm which will eventually hit the striker.
- The path of bot is traced back to bring the bot at its original position again.
- Codes used in the whole working are uploaded on github link.

Components

Mechanical part

Metal rods
Ball bearings
Support rods
Wooden plank
Nuts and bolts/screws
Angles and other small stuff

Electronic part

Raspberry-Pi
WiFi dongle

Stepper (4kg)
Stepper (4.2kg)
Stepper motor driver
DC motor (1000rpm)
DC motor driver
12V DC adapter
Jumper/normal wires

Other small stuff

CONTACT

1. Udayan Joshi
   14udayanjoshi@gmail.com
   7028462894

2. Pranav Kulkarni
   pranav.dk212@gmail.com
   8805540900

3. Saurabh kolambe
   saurabhkolambe@gmail.com
   7066446686

4. Anand Naik
   anandnaik1711@gmail.com
   8828290447