Robocon 2016

Technical Documentation

Electrical Team

Robotics Club, IIT Delhi

Acknowledgement

Team IIT-Delhi for Robocon 2016 sincerely acknowledge all the support provided by the Director of IIT Delhi, Prof. K. Gupta, and his administrative team, without which the participation in Robocon 2016 would not have been possible. We hope that the present Director, Dr. V Ramgopal Rao and the other administrative staffs will continue supporting our activities. The support of Prof. S. K. Gupta, Dean of Students, and Prof. Jyoti kumar, President (BRCA) in purchasing different components while fabricating the robots, and allowing us to work in the club room in SAC are also highly acknowledged.

The team members would also like to thank all our faculty members, especially Prof. S. K. Saha, Prof. Sunil Jha, Prof. Kolin Paul, Prof. J. P. Khatait and Jaitly sir of mechatronics Lab for taking time out of their busy schedules for guiding and mentoring us through this journey. Assistant Registrars of the Accounts department, Custom department and Purchase Section are also thanked for speedy actions in Robocon related matters. There is an endless list of people from different departments and sections of this institute without whose supports and blessings the participation in Robocon 2016 would not have been possible.

The help and guidance that we got in the later stages from our Robotics Club alumni is also very much appreciated. We would also like to thank all those involved in the manufacturing of the robot, both in the Central workshop of IIT Delhi and Mr. Parminder whose workshop and services were utilized extensively.

Finally, the efforts of all those involved in the compilation and documentation of this report is also acknowledged. This report has been compiled by Nishant Agarwal with valuable inputs from Amal George, Nilesh Kumar Jha, Rahul Kumar, Mohit Kumar, Pranjal Kumar, Gurkeerat Singh Bajwa and suggestions from Rishabh Agarwal, Varan Gupta, Vaibhav Gupta, Rishabjit Singh and Jyotirmoy Ray.

Abstract

The annual Robocon competition is an international robotics competition whose national round is held in Pune and is organized by Doordarshan (DD) and Asian Broadcasting Union (ABU). The theme for Robocon 2016 is 'Chai-Yo: Clean Energy Recharging the World' and the problem statement is to manufacture two robots named as 'Hybrid Robot' and 'Eco Robot'. The Eco Robot doesn't have an actuator to drive itself and obtains its driving force indirectly from the Hybrid Robot. Eco Robot carrying a propeller, should run through the arena to reach its final destination Wind Turbine Station (WTS) after manoeuvring through various slopes and hills, where the Hybrid Robot will pick up the propeller and climb a Pole to place the propeller on top. The robots must follow a stringent set of rules regarding the dimensions, weight and other aspects. The winner of the national round of Robocon will represent India in the International Robocon to be held in Bangkok.

Following the relative success of the 2015 IITD Robocon team, the 2016 Electrical team has focused its energies to the development of better, newer and more robust systems for the different components and mechanisms used in the robots. Looking at the problems faced by past teams from IITD, the teams have been able to implement several new design strategies and features for this year.

The following document contains the chronological design process that each electrical subgroup followed to arrive at a complete subsystem design. Then the document covers the integration of the subsystems into a complete robot and then the fabrication of the both the robots. Finally, the testing and final results of the two robots are presented. This document fully chronicles the steps that were followed by the Electrical design team from the initial group formation to the final testing of the 2016 robots.

Team Members

Mentors

- Himanshu Patel (2012ME10672)
- Nupur Kumari (2012MT50608)
- Rahul Kumar (2012BB52034)
- Rishabh Agarwal (2012ME10710)

Members

- Aditya Jain (2014ME20707)
- Aditya Kumar Chaurasiya (2014ME20708)
- Amal George (2014PH
- Amogh Gupta (2014ME20712)
- Ayush Shukla (2013EE10442)
- Gaurav Milankumar Sardal (2014ME20726)
- Gurkeerat Singh Bajwa
- Jyotirmoy Ray (2013ME10675)
- Kritarth
- Mohit Kumar
- Nalin B (2013ME10652)
- Nikhil Gupta (2014CS50462)
- Nilesh Kumar Jha (2014EE10523)
- Nishant Agarwal (2014EE10464)
- Pranjal Kumar (2014MT10355)
- Pranjal Maheshwari (2013CS50702)
- Purnendu Tripathi (2014ME20749)
- Qaim Abbas (2014ME10357)
- Rahul Kumar (2014EE10478)
- Rishabjit Singh (2013EE10488)
- Rithul P
- Rohit Patel (2014ME20757)
- Sanket Chaudhary (2014CH10129)
- Saurabh Sinha (2013ME10720)
- Shreyansh Gattani
- Vaibhav Gupta (2013ME10735)

- Varan Gupta (2013ME10737)

Contents

List of Figures

List of Tables

Chapter 1 Overview

The electrical team of the Robocon 2016 was responsible for designing, building and implementing all the electrical circuits used in the two robots, in addition to writing the software running on the onboard microcontrollers.

This year, importance was also laid on refining the old systems by using new components. These include new microcontrollers, different connectors, better sensors and better voltage convertors.

1.1 Theme

The motif of the Robocon 2016 theme is **"Clean Energy Recharging the World"**. Each team consists of two robots: **one Eco Robot** and **one Hybrid Robot**. Eco Robot doesn't have an actuator to drive itself. The driving force of Eco Robot is obtained indirectly from Hybrid Robot; for example, wind force, magnetic force, etc., or from the game field structure, gravity force, etc.

Eco Robot carrying Wind Turbine Propeller departs from "Eco Robot Start Zone". It runs along three zones; "3 Slopes and Hills", "River", "Down Hill", and aims for "Wind Turbine Station" by receiving driving energy from Hybrid Robot.

After Eco Robot reaches at "Wind Turbine Station", Hybrid Robot gets Wind Turbine Propeller from Eco Robot. Then Hybrid Robot climbs Wind Turbine Pole and assembles Wind Turbine Propeller on Wind Turbine Engine attached on top of Wind Turbine Pole. The team that successfully assembles Wind Turbine Propeller earlier is the winner of the game. This type of winning is called "**Chai-Yo**".

1.2 The Team

The electrical team comprised of one 3rd year student and seven 2nd year students.

Team Members:

1. Varan Gupta (Coordinator)

2. Vaibhav Gupta

3. Rishabjit Singh

4. Pranjal Maheshwari

5. Gurkeerat Singh Bajwa

6. Mohit Kumar

7. Nilesh Kumar Jha

8. Nishant Agarwal

9. Pranjal Kumar

10. Rahul Kumar

11. Amal George

Our faculty advisers were Prof. Sunil Jha and Mr. Dharmendra Jaitly.

1.3 Electrical Requirement for the robots

- 2 robots need to be built: one Hybrid Robot and one Eco Robot
- Eco Robot is allowed to use only one actuator to steer heading direction of the robot
- Hybrid Robot can be either semi-autonomous or fully autonomous robot
- In semi-autonomous mode, an operator is allowed to operate the robot for all tasks except Wind Turbine assembly task, using a wired controller.
- Communication between robots is not allowed

- The potential difference between any two points on the robot must be less than 24 Volts
- Only laser beams classified less dangerous than Class 2 lasers can be used
- An emergency switch must be present in both of the robots
- Fuses or circuit breakers have to be used in order to prevent damage in case of a short circuit

1.4 Team Management

1.4.1 Meetings

Meetings were held twice a week during the initial design phase. In the first meeting after the release of the problem statement, the members were grouped into four teams. Each team had to come up with its own design for the two robots. These designs were improved upon in each subsequent meeting. Feasibility of the designs was verified by performing the requisite calculations and analysis. After two months of brainstorming, the final design was formulated by combining the best elements from each team's design.

Once the final design was conceptualised, meetings were held more regularly, almost on a daily basis. In these meetings, our work was focused on arranging the required electrical components and designing the circuits while the efforts of the mechanical team were concentrated on fabricating the design.

1.4.2 Discussion Group and Email

A Google group was made on which all the important announcements were posted which reached each member through Gmail. In each meeting, a different member was selected who had to post the minutes of that meeting on the group. Discussions on the practicality of various designs were also held in this group.

1.4.3 Cloud Storage

An online folder was maintained on Google Drive where all the circuit designs, Arduino codes and datasheets of the various components were available for the group members.

1.4.4 Inventory

Two members of the team was assigned to maintain the inventory of all the components available in the club. The inventory was an online spreadsheet stored on the Google Drive which contained a list of all the parts available along with their quantities.

1.4.5 Practice

From the beginning, it was clear that this year's competition was all about completing the task in minimum possible time and required a lot of practice for the operators. Practice sessions were held regularly every day and night on the game field assembled at the first floor of the Student Activity Center.

These sessions started immediately after the completion of manufacturing part in the end of January and continued till the end of February. A specific document was maintained to record all the necessary data during these test runs. This included total time taken for completing all the tasks, the number of retries required, violations by the operator etc. By analysing the recorded data, mistakes were pointed out to the operators and strategies were devised for use in the competition.

1.5 Subsystems Approach

The problem statement of Robocon 2016 demanded the integration of multiple systems. These subsystems were identified to be the following:

- Drive
- Hybrid Force mechanism
- Gliding Arm mechanism
- Propeller Exchange Mechanism
- Line Following and Pole grabbing in automatic operation
- Pole Climbing Mechanism
- Remote controller for Manual operation
- Eco Robot Steering Mechanism

The members were each allotted one of the above subsystems and were tasked with implementing that particular subsystem.

1.5.1 Drive

A 3-wheel differential drive was selected for the Hybrid Robot because it provided better control and could easily negotiate turns. In addition to being quick enough to move, the drive system also needed to be agile and able to follow curved lines. For the drive, Maxon Motors were used in the Hybrid bot. Since the Eco bot was allowed to have only one steering motor and had to be light weight, a light weight, encoded servo motor, Rhino motor was used.

//TODO Pranjal

1.5.2 Hybrid Force mechanism

Different Force Mechanism were discussed some involved hovering of eco bot which were dropped after clarification of rules. Magnetic Force and Wind Force were our final options. Magnetic Force System was heavier and costlier than Wind Force. Also it involved mechanical exactness which was not possible. So we finally decided to go with Wind Force. For POC we bought EDF's from Hobbyking. They were able to give the required amount of force.

1.5.3 Gliding Arm mechanism

Gliding Arm needed to be sturdy and lightweight, because of these constraints magnets were rejected. Gliding arm created was 1000mm long and 600mm extending from the body. Calculations were done to make sure that the slider can hold the required weight.(figure and calculations to be added).

1.5.4 Propeller Exchange Mechanism

//I have no idea paste it from mech report.

1.5.5 Line Following and Pole grabbing in autonomous mode

This portion was first not that extensively discussed during POC. We used IMU as an additional input to drive the bot in straight line autonomously. Pole grabbing was discussed a lot and a variety of mechanisms came forward. After calculations piston mechanism was selected because it was easier to automate and it met the required normal force.(Calculations)

1.5.6 Pole Climbing Mechanism in autonomous mode

To climb 2m pole with a 20kg load required a lot of force and torque. After calculations appropriate motors were used to drive the wheels. Heating was observed in the circuit, it was solved by adding inductors in series. The horizontal rotation of bot could not be solved electrically.

1.5.7 Remote controller for Manual operation

As it was necessary for the hybrid robot to be controlled through wired, controller in manual mode, it was decided that Sony Dual Shock 3 controllers would be used for this purpose.

1.5.8 Eco Robot Steering Mechanism

In order to the decrease the dependence on the rider, several methods were tried to detect the shuttle for automatic actuation of the rackets. These included - using a 2D array of lasers, using multiple ultrasonic sensors or using image processing with the help of a camera. Though a certain level of detection was achieved using lasers and ultrasonic sensors, the system was not reliable enough to be implemented for the competition.

2. Microcontrollers

2.1 Introduction

Micro-controllers are the brains in the embedded systems. It is through the micro-controller that the coordination between various sub-systems in a complex system is achieved. Hence selection of most suitable micro-controller and having an efficient procedure for reliable and fast coordination and computation, are necessary for a machine to perform optimally to its objective.

2.2 Selection of Micro-controllers

Arduino micro-controllers are the de Facto standard at Robotics Club IIT Delhi, primary reason being the better hardware support. But this year saw a major shift in this as a wide variety of microcontrollers were worked upon and compared based on performance, ease of coding and circuit implementation. The major change was shift towards newer and faster ARM processor based microcontrollers like BeagleBone, Arduino Due, Rasberry Pi and Arduino Mega from the common AVR processor based Arduino Uno etc. Since the requirements of each robot was very different, we analyse them separately.

2.3 Eco Robot

Since the bot was fully automated it required more processing and the encoder algorithms needed trigonometric functions, which were very slow on the previously used Arduino Mega. Thus the switch to faster Arduino Due and BeagleBone was considered vital. But of these had drawbacks that neither of them worked on 5V, while all sensors like encoders required 5V supply. Arduino Due works on 3.3V whereas Beaglebone works on 1.8V. Debugging BeagleBone proved to be a major drawback while selecting the microcontrollers. Finally we zeroed down on Arduino Due, due to the ease of working on Due and its similarity to the Arduino platform.

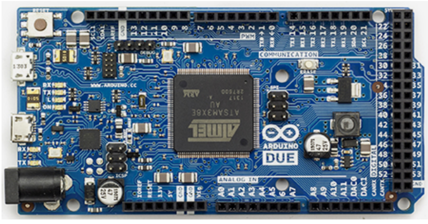


FIGURE 2.1 ARDUINO DUE

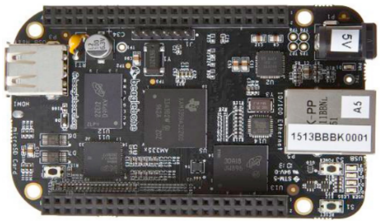


FIGURE 2.2 BEAGLEBONE BLACK 1

Parameter	Arduino Mega	Arduino Due	BeagleBone
Microcontroller	ATmega2560	AT91SAM3X8E	AM3358 ARM Cortex-A8
Operating Voltage	5V	3.3V	3.3V
Clock Speed	16 MHz	84 MHz	1GHz
SRAM	8 kB	96 KB	512MB DDR3
Flash Memory	256kB	512 KB	2GB
Analog Input Pins	16	12	7(max input - 1.8V)
Digital I/O Pins	54 (PWM - 15)	54 (PWM - 12)	65(3.3V)
Analog Output Pins	0	2 (DAC)	
Supported interfaces	4x UART, 1X I2C, 1x SPI	4x UART, 2X I2C, 1x SPI	4x UART, 8x PWM, LCD, GPMC, MMC1, 2x SPI, 2x I2C, A/D Converter, 2xCAN Bus, 4 Timers

TABLE 2.1 COMPARISON OF MICROCONTROLLERS 1

2.4 Hybrid Robot

As compared to Eco Robot the Hybrid Robot had more number of sensors and systems, but lesser computational requirements. Although the computational capacity of Raspberry Pi is much larger compared to an Arduino, Arduino is more user friendly for projects which are more about electronics than computation.

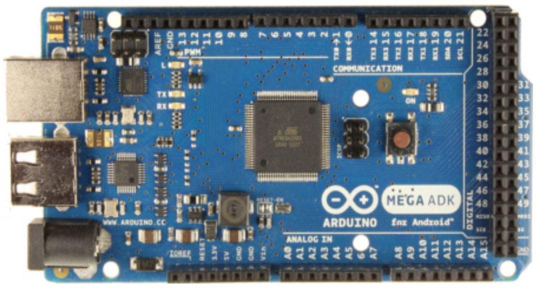


Figure 2.3 Arduino Mega ADK

Arduino provides its own open-source API in which the controller can be programmed easily on C-language based syntax and direct access to input and output pins via pin numbers. Almost all kinds of simple electronic components (eg: sensors, motors) can be operated directly using the pins provided on Arduino boards. Further, it also provides a well-documented and exhaustive collection of libraries for more complex components and protocols like LCD, keypad, stepper motor, SD card, Ethernet, Wi_, USB, SPI communications, etc. Also it has increased number of pins available for connecting other systems. Thus Arduino Mega was chosen as the sole microcontroller.

Chapter 3
Motors

The camera eco robot consisted of various subsystems-

3.1 Eco Steering Motor-

3.1.1 Rhino Motor-

Since the eco bot could have only one steering actuator, the only solution was to have servo style steering mechanism on the front wheel (similar to the auto-rickshaw). It was initially proposed for the actuator to be servo motor, but we later switched to Rhino motor (RMCS-220X). Rhino is an encoded DC motor with inbuilt micro-controller with various interfaces like Serial and I2c to communicate with the same.

We started with the Serial communication but we soon realized that Serial without feedback could abruptly trigger a random signals and render the motor unstable. A possible solution was to increase the delay between 2 consecutive commands to the rhino. However by the end we switched to I2C communication since it had feedback, but there were 2 major drawbacks, first we cannot use I2c in interrupt routines since clock is suspended (So we could not stop it in emergency situations) and secondly the execution stops if the rhino does not respond. We overcame first by checking for interrupt condition in each loop, the second was ensured by reducing cutoff time, anyways it didn't mean any sense in continuing if rhino does not respond.

The motor used for steering the robot was Rhino Motor. The angle to be rotated was transmitted to the motor via Serial Communication in the form of Strings separated by carriage return. As we were not allowed to have an extra actuator for braking, we used Paul and Rachet mechanism which allows the wheel to rotate in only one direction. Hence whenever we needed to apply brakes to the eco, we simply rotated the wheel by 180 degrees.

3.1.2 Hall Effect Sensor

Some other sensors were also used in the initial stage, for example, Hall Effect sensor was used to find the zero index of rhino motor, since the motor has a relative sense of its internal encoder, thus the position for the microcontroller was reset to 0 at start, irrespective where it actually was. This had to be corrected. A magnet was attached to the motor mounting and zero position was detected through its detection, this solution, however, failed since the error was significant and non-repeatable. We later used the same idea as electronic limit switch so that the motor, in case of any problem, did not move more than a certain degree and disturb the other connecting wires.

3.2 Hybrid Drive

3.3 Pole Climbing

3.4 Slider

3.5 Propeller Picking

Chapter 4
Motor Driver

4.1 Sabertooth

4.1.1 Introduction

To control the motors, we had two choices - one was to create a driver circuit ourselves or use a ready-made module or device for it like Sabertooth by Dimension Engineering. It was decided to go for the Sabertooth as they were more reliable, robust and energy efficient than any drier circuit we were going to make.

4.2 Selection of Sabertooth

Sabertooth can control 2 motors at a time as long as they are below a specific current rating.

S. No.	Motor	No. of Motors	Amperage per motor	Sabertooth Selected	No. of Sabertooth
1.	Maxon Motor	2	~6	2*32	1
2.	Banebot Motor	2	~20	2*32	2

TABLE 4.1 SELECTION OF SABERTOOTH 1

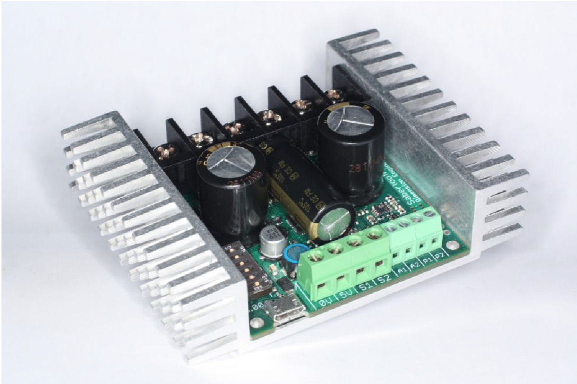


Figure 4.1 Sabertooth 2x32 1

4.3 Setting up Sabertooth

Sabertooth communicates with the arduino using a TTL type Serial communication at 5V. The communication was done at max possible baud rate available to ensure quicker response. For Sabertooth 2x32, we decided to use official library to control it.

Sabertooth 2x32, gave us a new capability to map our input signal to duty cycle of the motor and ramping of motors using a software named 'DEScribe' available on the official site. It was used to account for motor imperfections and ramping to ensure better control over the bot.

4.2. Kangaroo

Kangaroo is self-tuning device and used in the conjugation with Sabertooth motor driver. It is capable of reading two quadrature encoder (or potentiometer) to control the speed and position of one or two motors.

A Windows Software called DEScribe is available to customize, tune and operate the device. There are two channels in kangaroo one for each motor, input pins to get the feedback from the encoders (AMT113Q). Before being able to do anything with kangaroo, we have to follow a tuning procedure. DEScribe software and button on the device are two option to tune it. LED on the device shows the status depending upon the number of blinks.

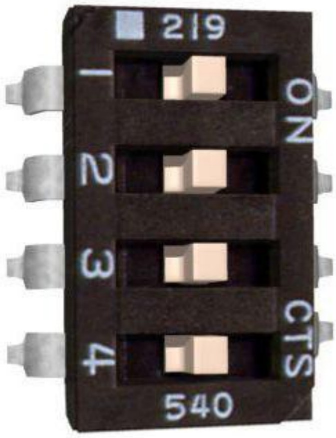
OFF setting		ON setting
1 Off: Analog input. Connect 0-5V analog signals to the S1 and S2 inputs.		**1 On: Digital input** Connect TTL serial, TX to S1 and RX to S2, or R/C servo signals to S1 and S2
2 Off: Analog feedback Connect a 0-5V signal to Feedback Input A		**2 On: Quadrature feedback** Connect an encoder to Feedback Inputs A and B
3 Off: Velocity control Motor speed and direction are controlled by the input signal		**3 On: Position control** Motor position is controlled by the input signal
4 off: Mixed mode The outputs are mixed together for differential drive mobile robots		**4 On: Independent mode** The outputs are independent. S1 controls motor 1 and S2 controls motor 2.

TABLE 4.1 DIP SWITCH SETTINGS 1

4.2.1 Tuning procedure using button:

Step 1: Press and hold the Autotune button for at least one second. The LED will begin to blink 1 blink, followed by a pause. Release the Autotune button.

Step 2: To confirm the teach tune mode which corresponds to number one, press Autotune button once. The LED will continue to blinking but more rapidly and dimmer.

Step 3: In teach tune mode, you must physically move the system to teach the kangaroo the acceptable travel range for tuning. The motors will not be powered at this point. To provide range follow this -> rotate wheel to some x (as u wish but should be more than or equal to two) revolutions forward and then 2x revolution backward then again x revolution forward. Remember that you must define the range for both channels, if you have motors attached to both channels 1 and 2.

Step 4: To confirm the range and begin the teach tune cycle, click and release the Autotune button again. The LED will begin a countdown blink, first slowly and then rapidly. After a delay of 10 seconds, the tune cycle will start. Your system will tune within provided range. The more will be the range more time it will take to tune and tuning will be more precise. When the tune cycle has finished successfully, the motors will stop and the LED will turn on solid. After tuning, you must power the device off and back on to use the new tune. If the LED is instead blinking, it means that the tune was not successful. The number of blinks corresponds to the error number that caused the tune to fail. One of the most common error is control error (corresponds to 3 blinks) and occur when feedback from encoders is lost.

DIP switches 1 and 3 can be changed without redoing the tune. If switches 2 and 4 are different from the setting recorded during the tune, the kangaroo will not operate the motors until the switch is changed back or the system is tuned again.

Since this device was new to club, we faced many difficulties using it. Making connectors for the encoders (AMT 113Q) to provide feedback to kangaroo was really time consuming. Proper wire was not used to make the connectors and many connectors got damaged due to excessive current flow in the ground wire of the connector.

One problem with the AMT encoder we got to know very late is that its resolution kept changing randomly, later we found this is because of old firmware so we updated the latest firmware in all the encoders we had using software AMT viewpoint.

Chapter 5
Force Mechanism

5.1 EDF

The Electric Ducted Fan bought from Hobbyking was used to push the eco bot through the hills and slopes is a BLDC motor attached to the specific ESC driver. The company had put up a video demonstrating the power of the fan in the Robocon home page, which grabbed our notice. This was proved true once we got the fan. Special safety net was used to cover the fan as there was an accident during the initial trials. The ESC driver uses servo commands to control the speed of the propeller. Two 12V Lipo batteries were used in series for powering up the ESC.

FIGURE 5.1 EDF 1

5.2 Slider

The Hybrid slider carrying the BLDC motors for driving eco bots was automated using distance sensors which kept its distance from ground constant. PID was used as the controlling algorithm to move the slider (Banebot motor 12V). We switched from ultrasonic to laser sensor, PEPPERL + FUCHS VDM28 -15L in an early stage. Whose output is a constant current, varying from 4 mA to 20 mA. The distance could be set from anything between a 20 cms to around 15m and the output was noise free current depending on the distance and the response time is of the order of nano seconds which is far less than that of the Arduino Mega. I read the output by driving the output current through a 220ohm resistor and measuring it against a high impedance input. We used 220ohm since in that case the maximum voltage would be 220 * 20mA = 4.4V, which is below the 5V maximum limit of the Analog in pins of Arduino Mega. The same distance sensor was used to determine whether the bot had climbed the right height to put the propeller in place.

Chapter 6
Line Following Mechanism

6.1 OPT Sensors

6.1.1 OPT 101

Various sensors to facilitate line following were used, we sticked with OPT101 sensor which is an industrial grade light intensity sensor. These sensors have been tested well over the years and have given reliable results. Since the problem statement required us to detect white line over coloured surfaces, RGB LEDs needed to be used along with OPT101 sensors.

6.1.2 RGB LED

The PCB for Line following used RGB led along with OPT101 sensors. We then switched between Red and Blue LEDs for differently coloured regions, controlling them with MosFETs. In the red regions we used blue LEDs such that we get maximum contrast between white line and red region. The captured intensity was then processed via the analog In pins and PID mechanism (with maximum error limit) was used as the steering mechanism on the Rhino motor.

6.2 TCS Colour Sensor

We also used TCS3200 and TCS 34725 (colour sensors). TCS3200 is interrupt based and sends a 50% duty cycle, frequency proportional to the light intensity received by the sensor, waiting for a certain fixed duration to count the number of ticks received gave a measure and TCS34725 is I2c based. Since rhino already used the I2c port, we had to skip TCS34725. So finally we did not use any of the colour sensors.

6.3 Hybrid Line Following

This year, line following was an obvious choice for path traversing of Hybrid robot. PCBs and Opt101 sensors for this purpose were available in club, so we had head start in the sense how the arrangement would look like.

6.3.1 Challenges-

But when we started testing in December, these problems turned out to be considerable-

(1) Detection- Colour surrounding white line to be followed was grey, so difficult and imprecise detection of the line.

(2) Calibration (a) - 8 Opt101 sensors were to be used and there individual outputs on a single colour were mismatched.

(3) Calibration (b) - Small exposure of sensors to ambient light would give drastically wrong output.

(4) Mechanical Constraint- Arrangement of Opts on PCB was equidistant. PCB was 15 cm long, and mechanical constraints of robot did not allow more than 12 cm.

(5) Linear error output- As white line moves sensor to sensor, was not effective to bring robot back on center.

(6) Large number of Opts- If automation was to be done throughout all the path, robot had to have 3 PCBs or sensor arrays because of mechanical constraints, two at sides, 1 at center. So number of Opt sensors 8X3 = 24, which is too much.

(7) Algorithm- For line following keeping sensor array at sides, no algorithm was available at that time for such a heavy robot as Hybrid.

6.3.2 Overcoming Challenges

Here we discuss these problems and how we solved them or could not solve them.

(1) Detection- Arduino reads analog inputs in resolution of 0 – 1023 (5 V). In case of grey colour surrounding white line, this made detection harder, because transition of an Opt from white to grey or grey to white gave output

difference only 200 (~1 V) in best case. We had to somehow magnify this window. Opt can give output as big as 10 V, if supply given 10 V and external gain impedance of 2 MΩ [*1]. With increased supply, output difference between grey and white line came out to be 2 V (~400 analog input on Arduino), which is fairly detectable. But Arduino can't take input of more than 5 V. So we had to bring output of Opt down to 5 V without loss of the magnification. We did it with help of amplifiers and negative supplies -

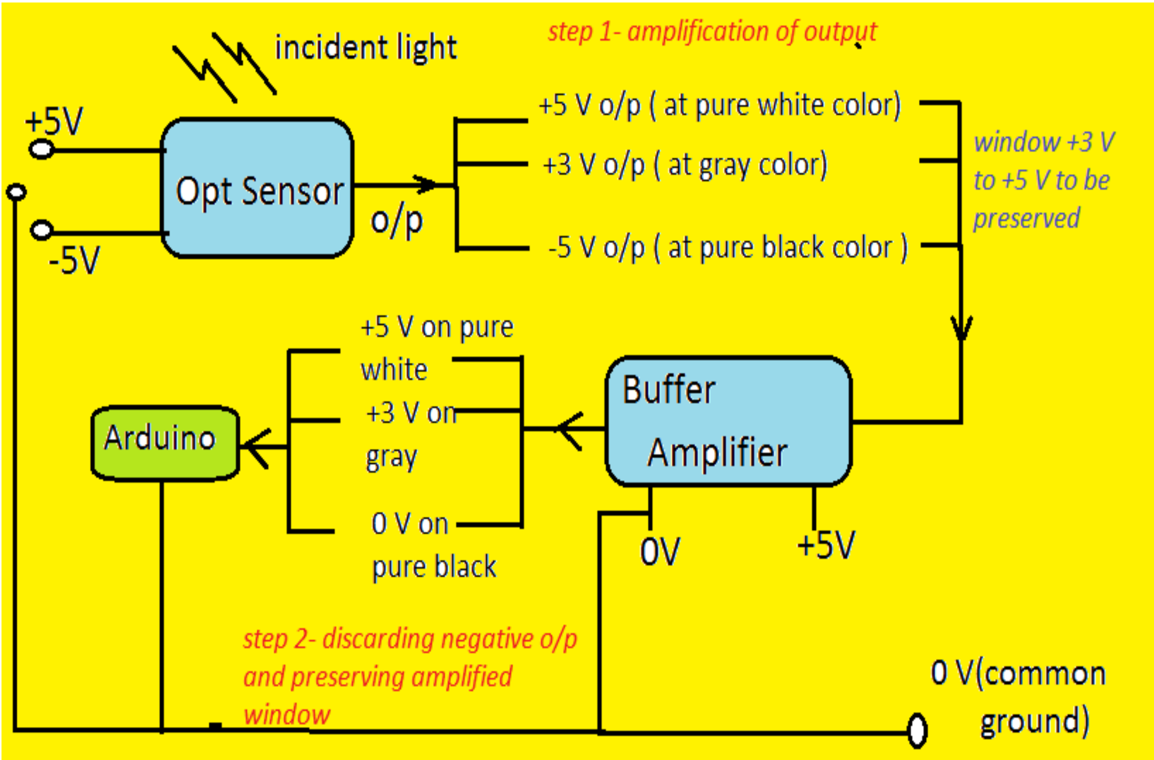


FIGURE 6.1 LINE FOLLOWING CIRCUIT 1

(2) Calibration (a) – Mapping of one all Opt outputs was done from there minimum on black and maximum on white to 0 – 5 V in order to solve mismatched outputs on the same colour. This method is time consuming, because in new ambient light conditions or on replacement of any opt, we had to get minimum and maximum values of Opts on black and white colour again. Later, we developed code to remove need of black colour and to reduce time taken in process. In new method, we sweep the array on white

line again and again, code records average values of individual Opt sensor on white line and grey colour. Then in between the grey value and white value we define threshold value, above which, we can assume sensor is on line, and below which we assume sensor is in grey region.

(3) Calibration (b)- To discard effects of ambient light, we could not make any algorithm. So we covered the array tightly with black rubber sheet and a layer of black cloth to isolate from external light. Also, we isolated all Opt sensors from each other and from LEDs using small box shapes to discard all interference with an Opt. This way, we eventually removed effect of external lights.

(4) Mechanical constraints- Initial testing was done on large PCBs. But new were designed which had small gap between Opts and total length of ~ 12 cm.

(5) Linear error output- As line moves from sensor to sensor, we found that linearly increasing and decreasing the error (distance from centre) was not working. Because actually, distance from centre is not the error in differential drive, but it is angle that drive makes with line (differential drive can't move sideways, linear error would work in Swerve and Holonomic drive because they can move sideways). To solve this, we gave weights to different Opts, to make error nonlinear. For example- (from left to right) -20, -10, -5, -1, +1, +5, +10, +20 . These values were found from repetitive testing.

(6) Big numbers of opts- 24 opts make calibration and maintenance an issue. We removed the central 2 Opts from PCBs, because there most important job was to tell whether line is in centre. We could tell this by finding outputs of other Opts, and making sure line is not out of the PCB itself! So Opts came to number 18. Further, we tried to make such algorithm so that line remains between 2^{nd} and 7^{th} Opt in PCB, this way we could remove 1^{st} and 8^{th} Opts too, making total Opts 12, 4 each.

(7) Algorithms for line following with sensor array at sides- Our mechanical constraints were such that if we want to make the robot autonomous, line would be near left wheel or right one. There was not algorithm available for this configuration. Till end of January, we had made and algorithm with help extensive testing and we were available to follow line at sides of robot. But to secure expensive motors, ramping was applied to drive, and with ramping,

our algorithm didn't work. Finally, for Robocon, manual operating was selected and line following at side of robot was not used.

6.3.3 Other Problems

More problems on new mechanical system– A new mechanical chassis was made, because the previous one was heavy, and space for fixed sensor array PCB in centre of robot was not there in the new design by mistake. Also, because of less time with new mechanical system, practice was given priority. So we could not test line following properly on new system. Finally, few days before Robocon, we made a design in which the sensor array PCB was not fixed on its place, instead, when robot would reach the pole, PCB would slide under the robot. This way the space problem for PCB was solved.

Line Following in Robocon 2016 – We could not implement line following properly on new mechanical design and robot was slightly unstable and slightly oscillating with what we had done. Oscillations could lead to collision with the boundary of the field and loss of points, or robot could be damaged. So finally, line following was totally dismissed and in final automation part of statement, encoded automation was applied.

Chapter 7
Encoding

7.1 Introduction

Since the path to be followed was available exactly months before the competition, it was a wise idea to encode the complete path in the microcontroller and use encoded wheel to find the distance travelled and the position of eco robot. In order to follow a predefined path, the Encoders are used to measure the distance travelled from the starting position and using this data as an input required PWM corresponding to the Velocity and Angular Velocity is given at the output at each instance on the path. Thus following the required predefined path.

7.2 Algorithm

Using the encoder as primary input and giving the corresponding radius of curvature of the path as the output the position turning angle data is stored in the bot. In the turns where the time for detection of turns is minimum for line following sensor this helps in following the predefined path. The error or deviation from the path is minimized when the robot is executing the straight line (see Path Planning) using the help of line tracking sensors where ever there is line and IMU during the river phase.

7.3 Path Planning

The points through which the robot has to travel is joined by straight line. The corners thus formed is replaced by symmetric polynomials, these polynomials are generated by tacking the toppling limit of the robot into consideration. Thus throughout the path we get the information of turning radius at each position i.e. Radius of curvature of path w.r.t the position is extracted. In the case of eco bot the

Radius of curvature was controlled by the rhino motor, so the radius of curvature of path is modified into turning angle of rhino.

7.4 Using IMU (Inertial Mass Unit)

IMU (Razor, 9DOF) (for AHRS) was used to measure the yaw angle and it was also used (using P controller) to assist movement in the river region (zig-zag line where line following was not a good option, given that following it was optional) and also to restore orientation specially when we switched from encoder control to line following with the risk of the eco-bot going out of the defined line bounds by the end of encoder control.

7.5 Limitations

The encoder had to be reset after travelling about 6-7 meters (Hill-3), this was due to the accumulation of incremental error in the encoder due to the error correction from line sensor array. Also the theoretical use of path travelling algorithm had certain drawbacks such as the dependence of turning angle on the speed of the bot, Continuous to discrete mapping of turning angle. The former was tackled using a high speed rhino. The second problem could be solved by considering the least count of rhino and encoder.

Chapter 8
Computer Vision

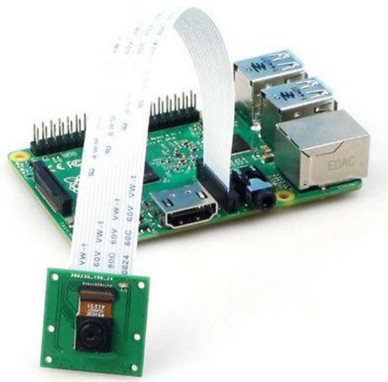
8.1 Introduction

Apart from line following using line detection sensors and encoders we worked parallelly on line following using computer vision techniques. This was required as a little physical change in the field or the lighting conditions should not affect the performance of Eco Robot, hence camera imaging was chosen. For the purpose of computer vision it was decided to use Raspberry Pi2 as the microcontroller and Pi Cam as the primary camera.

8.2 Microcontroller

We used Raspberry Pi 2 Model B as a microcontroller for the camera eco bot. As this robot is based on Image processing, it was very important to use a microcontroller with sufficiently high clock speed in order to obtain a good frame rate, even though the amount of data to be processed (the pixels of an image) are too high.

The raspberry pi 2 has a 4 core processor of 900 MHz each and 1GB RAM which provides sufficient processing speed for our problem. But frame rate of the video stream of the camera depends not only on the speed of processing of each frame, but also on the speed of communication between camera and microcontroller. Raspberry pi cam communicates at a faster rate than a simple USB camera as it has more channels for communication. Using this combination of Raspberry pi cam and Raspberry pi 2, we were able to achieve a frame rate of around 50fps which is a



decent rate for updating the angle of rhino.

FIGURE 8.1 RASPBERRY PI WITH PI CAMERA 2 1

8.3 Raspberry Pi Cam

For detecting the white line in the field, we used simple segmentation techniques based on the RGB values. However, as the pi cam has an in built feature of automatic brightness control while capturing a video stream, the RGB values of the white line varied greatly according to which region it belongs to. For example: the white line in the river region has a slight yellowish tinge as compared to other regions (when seen through camera). This tinge becomes significant during dull ambient lighting. To prevent such fluctuations due to change in region (or change in ambient conditions), we attached an artificial light source in the form of three parallel bright LED strips which illuminated the area of the field directly in front of the robot. Due to this artificial light source, this erratic behaviour of white colour was prevented and hence line segmentation was perfect.

8.4 Line Following

8.4.1 Line Detection Algorithm

An important feature of this algorithm is that it does not colour segment the entire image. However it segments one of the top row of the image using the threshold RGB values of ideal white strip. It calculates the starting point and end point of each white patch obtained in that row and hence its thickness. If the thickness of this patch is sufficient, then by using the start point of this patch, then the angle made by the line joining middle of patch with middle of bottommost row and the vertical line is fed into the rhino motor and we wait for the next frame. Examples:-

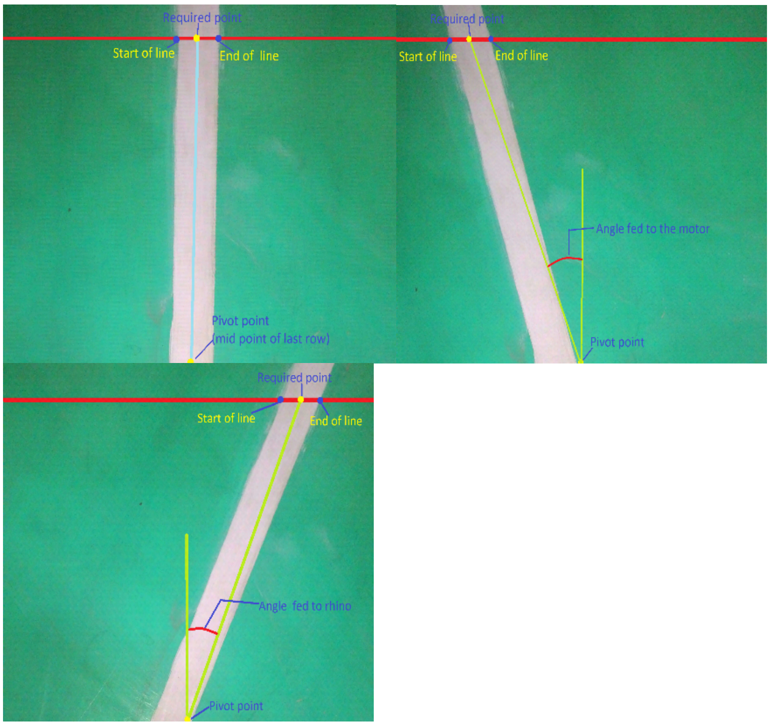


FIGURE 8.2 (A) STRAIGHT MOTION **FIGURE 8.2 (B) LEFT TURN** **FIGURE 8.2 (C) RIGHT TURN**

Fig 8.2 Processing of an image

In case an optimal white patch is not obtained till a particular row number, then an angle of around 65 degrees is fed into the rhino motor. This is for the case of 105 degree turn before river section.

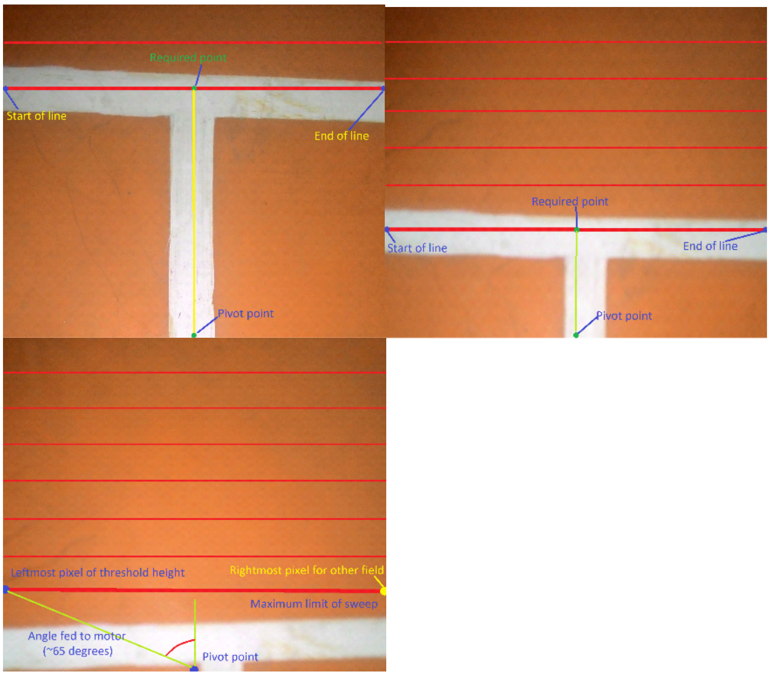


FIGURE 8.3 PHASES OF 105 DEGREE TURN

8.4.2 Navigation on Downward Slope

As the spline was an inclined region where the robot had to cover under the influence of gravitational force, its speed can't be controlled by the hybrid robot. So, without any braking the bot would achieve a lot of speed and reach the turbine station much before the hybrid robot which we did not want (as our propeller picking mechanism required hybrid robot to reach turbine station before eco). So for this we had to repeatedly rotate rhino by 180 degrees after a time of around 50 frames. The detection of spline region was done by using average RGB values of the row of pixels which we are sweeping.

8.4.3 Navigation in River Region

An ideal river (zigzag) has turns at exactly 90 degrees to each other. But as the camera was tilted by almost 25 degrees to the horizontal, a skewed image of the river was seen to the camera. Due to this, the angle of intersection of lines in the river section was acute. So, the above mentioned line following algorithm (row-wise sweeping which works for obtuse angles) may fail. Due to this, a combination of vertical and horizontal pixel sweeps had to be done.

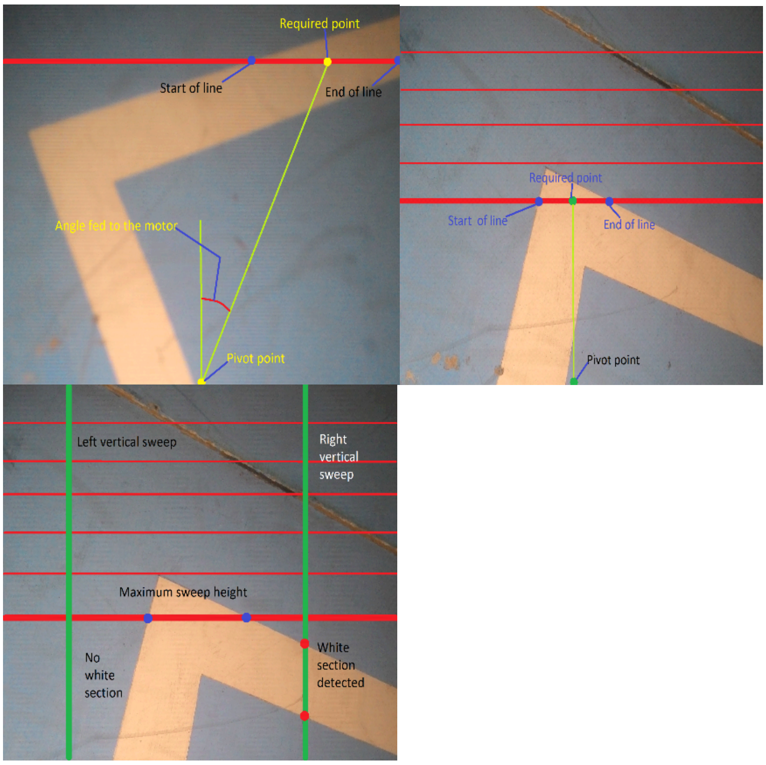


FIG 8.4 STANDARD IMAGE PROCESSING IN RIVER

Chapter 9 Propeller Exchange Mechanism

9.1 Propeller placing in eco bot

Earlier we were using Rhino motor (same motor which is used in Eco bot) for propeller picking but later we found it unsuitable due to motor calibration issues and heavy vibrations when load applied so finally we decided to use Servo motor for this particular task. According to propeller picking mechanism the propeller should be placed horizontally. Major aim of propeller placing stand designing was to make picking of propeller easy. And the placing should be light weight. The design should ensure that propeller should not fall down during game. This design is purely mechanical design. The holder was made up of circlip of appropriate diameter that can hold the spherical part of the propeller.

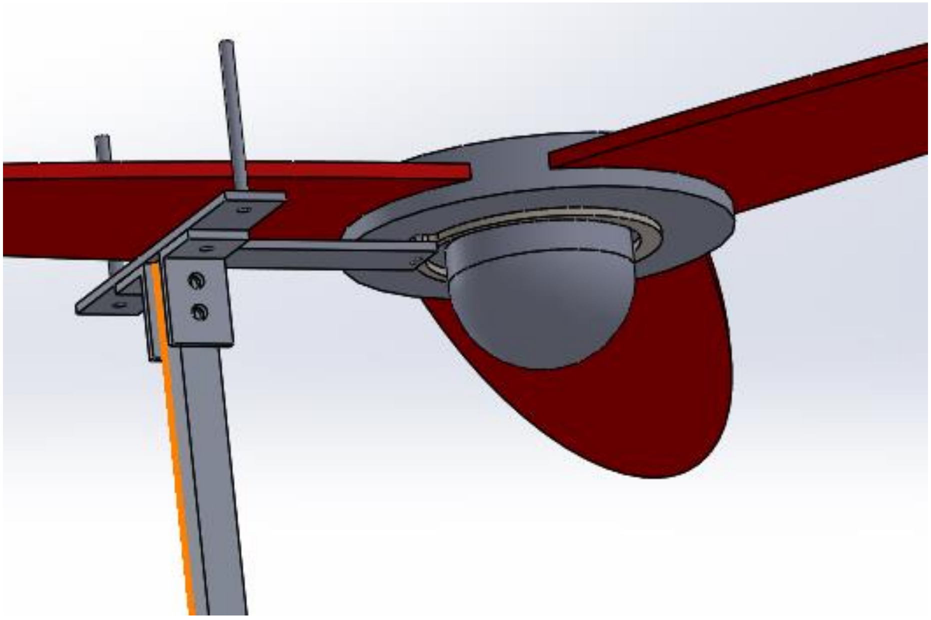


FIGURE 9.1 PROPELLER PLACING ON ECO BOT

9.2 Propeller picking mechanism in Hybrid bot

The picking of the propeller from eco bot by the Hybrid bot is based on the design of the picker which is in U shape made up of acrylic material. The U shaped picker helped in diametrical alignment of the spherical extrusion of the propeller with the picker. This ensures the stability of the propeller on the picker and maintain stability while taking back the Hybrid bot back to the automatic start zone. And also stabilise it during the movement of hybrid bot from Automatic start zone to the pole and till it finally placing at the Wind Turbine Station.

This design uses the servo based mechanism which is initially at 90 degree position and at the time exchange it comes to 0 degree position and aligned below the propeller, after exchange it comes to 45 degree position and finally changes the configuration at the time of placing the propeller at the wind turbine station. The torque which is coming is very low as the mass of the propeller was very low. During the propeller picking, motion of servo motor is controlled by PS3 controller whereas during propeller placing on wind turbine station, its motion is controlled by automatic command in code.

Chapter 10
Pole Climbing Mechanism

10.1 Pole Detection

After reaching the automatic start zone our first objective was to reach the pole. But there we need feedback mechanism to detect the pole. Initial approach was to use the ultrasonic sensor to detect the pole and Quadrature Encoder (AMT113Q) to get the distance traversed by Hybrid bot towards the pole. Later we switched to Laser sensor from Pepperl+Fuchs due to poor performance of ultrasonic sensor.

Figure 10.1 Pepperl+Fuchs Laser Sensor

10.2 Pole Grabbing

After reaching the pole next task of Hybrid is to grab the pole and climb the pole. For this we have compact piston on which 8 bearing system is mounted (driving) which are actuated by the air pressure between the range 2.0-3.5 bar. Also Hybrid has two belt coupled motor driven wheels. The pole is grabbed such that there is minimum distance between the axis of the pole and axis of the Hybrid bot. Hybrid enters the pole and then this compact piston is allowed to open, this action grabs the pole.

10.3 Pole Climbing

After grabbing the pole the hybrid is to climb the pole. For this there are two motor driven wheel having rubber attached to provide friction which are belt coupled and eight driving bearing which provide enough normal force to climb the pole.

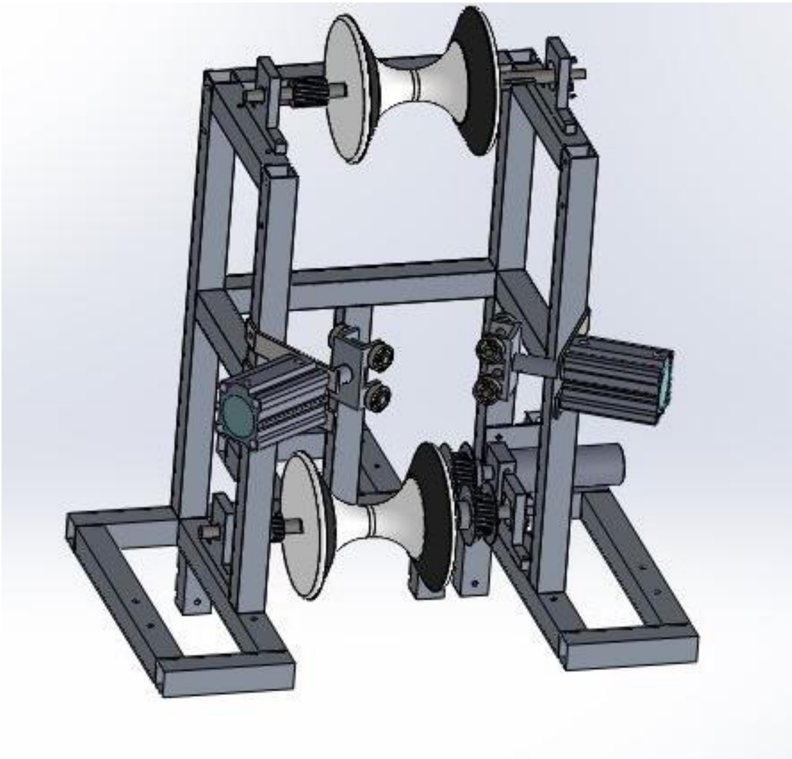


FIG 10.2 POLE CLIMBING DESIGN HYBRID BOT

10.4 Stopping

We have to climb the pole of about 2m. In order to stop at the right position we need the feedback mechanism. So we used the banner reading of the laser which gives the values for the slider mechanism and calibrated our value to stop after reaching the Wind Turbine Station. We used 24 V BaneBot with sabertooth and kangaroo along with 2 pistons with 2-3 bar pressure each.

10.5 Placing Propeller

After reaching the Wind Turbine Station (WTS) the next task is to place the propeller at the WTS. This task is easily done as the servo is at 45 degree carrying the propeller. To place it just we have to turn servo to 90 degree and propeller got placed to wind turbine station as it has magnet which got attached to magnetic plate. In order to ensure that propeller is placed we turn servo back to 45 degree position.

Chapter 11
Voltage Regulation

11.1 Turnigy 12V to 5V/6V Converter

We also used the highly efficient 12 to 5V power convertor from Turnigy, since the HC7805 were highly prone to failures from noise and had much less current ratings. We learnt using wheels with electromagnetic brakes, however dropped the idea since we could not find a reliable way to generate power to drive it (The competition constraints restricted the use of pre-stored energy, battery etc, for more than one actuator).

11.2 //Todo Nilesh

Chapter 12
Wire Selection

12.1 Introduction

This year we used both Maxon motros as well as BaneBot motors in the Hybrid robot which had much higher current rating then any of the motors previously used in the club. So, it was decided to do all the necessary calculations for the wires and choose them according to need. This year we used better and thin insulation enamel wire. Since no motor apart from steering motor was used in Eco robot, it had low current usage and hence no specific wire selection was required. Hence we have decided to analyse only Hybrid robot.

12.2 Hybrid Robot

Hybrid bot had following devices:

- 2 Maxon Motors
- 1 Servo Motor
- 2 BaneBot Motors, 1 each for Slider and Pole Climbing
- For 2 Electric Duct Fans
- Arduino Mega 2560
- Sensors-
  1. Line following Sensor Array
  2. Ultrasonic Sensor
  3. Laser Sensor
  4. Encoders
  5. Pneumatic pressure Sensor
  6. Voltage Sensor

S. No.	Connection end points		Optimum Current Rating (A)	Designed Current Rating (A)	Wire Size (AWG)
1.					

//TODO Pranjal

12.3 Guideline for Wire Selection

The length of the wire include the wire length from one end point to other and back from second end point to first end point. This table is for 12V circuit, so to use it with 24V, we convert our current rating to the table's current rating keeping the power (*V * I*) through the wire same.

Amps @12 Volts	LENGTH OF WIRE (Centimeters) American Wire Gauge (AWG)						
	100	150	200	300	450	600	750
0 to 1	18	18	18	18	18	18	18
1.5	18	18	18	18	18	18	18
2	18	18	18	18	18	18	18
3	18	18	18	18	18	18	18
4	18	18	18	18	18	18	18
5	18	18	18	18	18	18	18
6	18	18	18	18	18	18	16
7	18	18	18	18	18	18	16
8	18	18	18	18	18	16	16
10	18	18	18	18	16	16	14
11	18	18	18	18	16	16	14
12	18	18	18	18	16	16	14
15	18	18	18	18	14	14	12
18	18	18	16	16	14	14	12
20	18	18	16	16	14	12	10
22	18	18	16	14	12	12	10
24	18	18	16	14	12	12	10
30	18	16	14	12	10	10	10
36	16	14	14	12	10	10	10
40	16	14	12	12	10	10	8
50	16	14	12	10	10	10	8
100	12	12	10	10	6	6	4
150	10	10	8	8	4	4	2
200	10	8	8	6	4	4	2

Figure 12.1: Wire Selection Criteria

Chapter 13
PCB Designing

13.1 Introduction

In previous few years, we had quite a problem with the our designed PCBs. We faced quite a bit of problems from overcurrent and excessive heating of the boards due to improper design. Last year we used IPC-2221 guideline to design our circuit boards.

The problems of overcurrent and excessive heating were solved. So this year too we used the same guidelines. This year due duality of softwares and ideas led to a few problems. We introduced eagle software to design our PCBs, it was efficient but due to lack of proper libraries we reverted back to use of diptrace. Two PCBs for each bot were created one for beaglebone and other for arduino microcontrollers. This led to confusions which made the whole process time consuming and prone to errors. As something to learn from this Robocon electrical designing should be finalised in the POC itself.

13.2 Power Board

Power board was used only in Hybrid Bot. It was made separately because of high current requirements in it making it more prone to defaults while using.Maximum design current-( ) from where to where.

//Add a photo(Nishant)

13.3 Hybrid Main Board

It had digital signal on it with maximum current of 500mA and we had copper thickness of 0.05mm.As we wanted maximum temperature rise of 10◦C at the ambient condition of 40◦C, we got the required trace width to be 0.08mm. Due to manufacturing restrictions and as a factor of safety, we designed the PCB with trace width of 0.2mm(For Power lines). Hybrid Main Board was created for beaglebone too which had a few extra voltage regulators for analog signals. Learning from previous years, this year PCBs were economically efficient.

13.4 Eco Main Board

Similar to Hybrid Main Board It had digital and analog signal on it with maximum current of 500mA and we had copper thickness of 0.05mm.As we wanted maximum temperature rise of 10◦C at the ambient condition of 40◦C, we got the required trace width to be 0.08mm. Due to manufacturing restrictions and as a factor of safety, we designed the PCB with trace width of 0.2mm(For Power Lines). The odd shape of the PCB was to accommodate the Rhino motor on the Eco bot.

13.5 Line Following Board

This year we required line following board for both the bots. In order to achieve that a generic board with all the connections and components was created afterwards it was placed to our requirements in both the bots. Boards created with Eagle had some problems because Error checking was not simultaneous in it and also parameters set were wrong. Due to this all final PCBs were created in DipTrace.