PROJECT REVIEW REPORT ON

"Swarm Robots In A Closed Loop Visual Odometry System By Using Li –Fi"

Submitted in partial fulfillment of the requirements of the degree of
B.Tech. (Electronics Engineering)

By

Kewal Shah (121060027)

Dhiraj Patil (121060028)

Anshuman Singh (121060040)

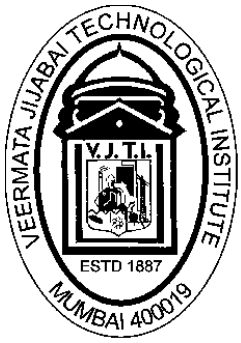
Udit Patadia (121060058)

Nilay Sheth (121060065)

Rahul Solanki (121090022)

(2015-2016)
Under the guidance of

Dr. Faruk Kazi



DEPARTMENT OF ELECTRICAL ENGINEERING
VEERMATA JIJABAI TECHNOLOGICAL INSTITUTE

(Autonomous Institute Affiliated to University of Mumbai)

Mumbai 400 019

(2015 - 2016)

ACKNOWLEDGEMENT

It gives us great pleasure to present this report, a written testimonial to a fruitful endeavour. We take this opportunity to thank our project guide **Dr. Faruk Kazi,** VJTI, working with whom is a delightful and enlightening experience.

We would also like to express our sincere thanks to **Dr. R. N. Awale**, Head of the Electrical Engineering Department, for his constant encouragement and support.

We thank Society of Robotics and Automation (SRA, VJTI) for letting us use the space and resources. All the robots and circuits presented in this review were fabricated at SRA.

Extending thanks to ABU Robocon for all the experiences with embedded systems and robot designing, which helped us to prototype quicker, think faster, materialize our plans better.

ABSTRACT

Motivated by the looming radio frequency (RF) spectrum crisis, this project aims at demonstrating that optical wireless communication (OWC) has now reached a state where it can demonstrate that it is a viable and matured solution to this fundamental problem. Li-Fi stands for light fidelity. It is a technique used for transmission of data at very high speeds through light, which transfers data by varying its intensity, frequency at unperceivable rates. Using this technology, we have proposed to implement Swarm Robots in a Closed Loop Visual Odometry System. The closed loop is established with an overhead camera, mocking the GPS in the swarm environment, providing the local odometry values. Pose estimation data on the robots is done with Augmented Reality tags, and this data is fed-back to the robots via VLC. The project report introduces the differences between Li-Fi, Wi-Fi, and VLC. It elaborately describes the modular approach used to exploit off-the-shelf components to modularize Li-Fi and Overhead localization. The closed loop to control each swarm robot with simplex communication is highlighted. Problems faced while prototyping and overcoming them in later versions have been also described.

DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources.

We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea / data / fact / source in our submission.

We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

CERTIFICATE

This is to certify that **Kewal Shah, Dhiraj Patil, Anshuman Singh, Udit Patadia, Nilay Sheth, Rahul Solanki** students of **B.Tech. Electronics** have completed the project review report entitled, **"Swarm Robots In A Closed Loop Visual Odometry System By Using Li –Fi"** to our satisfaction.

Dr Faruk Kazi

Guide

Dr. R. N. Awale

Head

Department of Electrical Engineering

INDEX

Sr No.	Topic	Page No.
I	List of Figures and Tables	
II	Introduction i) LiFi ii) Visual Odometry iii) Closed Loop control	1
III	Hardware Implementation i) LiFi Receiver ii) LiFi Transmitter	5
IV	Setting up the environment i) Swarm bot ii) Dealing with Approach strategies iii) Dynamic Path Planning iv) Swarming Algorithms	12
V	Software Implementation i) Algorithms tested a) Rev. A b) Rev. B c) Rev. C ii) Packet composition iii) Path Planning algorithms	15
VI	Conclusion	26
VII	References	27

I. LIST OF FIGURES and TABLES

Figures

Figure No.	***Description***	***Page no.***
1	*VLC = Illumination + Communication*	*1*
2	*Communicatopon frequency spectrum*	*1*
3	*Wavelength of different spectrums*	*1*
4	*Modularized diagram explaining the Closed Loop Li-Fi control*	*4*
5	*PhotoDiode v/s PhotoTransistor*	*6*
6	*TEMT6000*	*6*
7	*TEMD6200*	*7*
8	*Photocurrent v/s Luminance of TEMD6200*	*7*
9	*Choosing the compatible Transmitting and Receiving Diodes*	*10*
10	*Transmitting Phosphor LED*	*12*
11	*Mechanical CNC CAD of the Robot*	*13*
12	*Electrical CAD of the robot*	*13*
13	*Checking for Li-Fi packets*	*14*
14	*Testing the robot using CAMshift*	*14*

Tables

Table No.	***Description***	***Page no.***
1	*Comparison between Li-Fi and Wi-Fi*	*2*
2	*Characteristics of TEMD6200 from Vishay*	*7*
3	*TEMT6000 v/s TEMD6200*	*8*
4	*Transmitter Rev. A v/s Rev. B*	*11*
5	*Algorithms for Robot tracking*	*15*
6	*Serial packet compositions*	*24*
7	*Path planning Algorithms comparisons*	*25*

II. INTRODUCTION

The project aims to establish closed loop which controls swarm robots, via LiFi. The data to establish closed loop is done via visual odometery. Each concept i.e. (i) Li-Fi, (ii) Visual Odometry, (iii) Closed loop swarming is discussed in this section.

(i) Introduction to Li-Fi: Li-Fi stands for Light Fidelity and is gaining recognition due to a large number of benefits. It exploits the bandwidth of visible spectrum for communication and hence is based on LEDs for the transfer of data. Variation in the intensity or wavelength of light provides us binary 1 and binary 0 of data which transmits information wirelessly through Visible Light Communication (VLC). Fig. 1 illustrates the basic idea behind this concept. Also the spectrum diagram in Fig. 2 highlights the Bandwidth of Li-Fi communications is 10,000 times greater than the current radio communication bandwidths existing [1]. Hence Li-Fi is ideal for high density wireless data coverage inside confined area.

Fig.1: VLC= Illumination + Communication

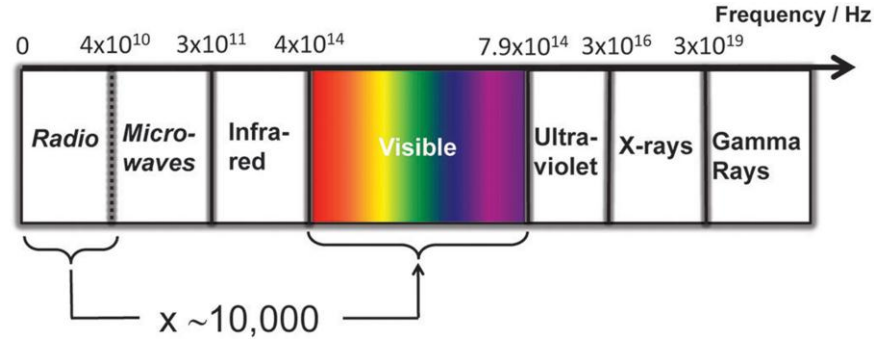


Fig. 2: Communication Spectrum

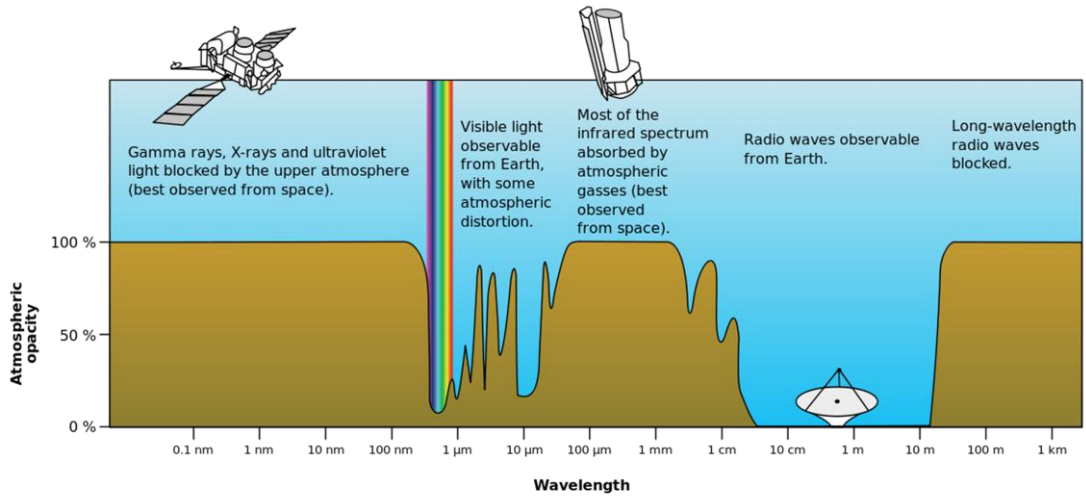


Fig. 3: Wavelength distribution of different spectrums.

The differences between Li-Fi, Wi-Fi and VLC are discussed in *Table1*. The main difference when talking about Wi-Fi and VLC is that VLC means that the communications are duplex. It ensures a minimum simplex communication via light, until a particular distance.

Table 1 : LiFi vs Wifi	**LiFi(Light Fidelity)**	**WiFi(Wireless Fidelity)**
Interference	Do not have any interference issues similar to radio frequency waves.	Will have interference issues from nearby access points(routers)
Technology	Present IrDA compliant devices	WLAN 802.11a/b/g/n/ac/ad standard compliant devices
Applications	Used in airlines, undersea explorations, operation theaters in the hospitals, office and home premises for data transfer and internet browsing [2]	Used for internet browsing with the help of wifi kiosks or wifi hotspots
Merits(advantages)	Interference is less, can pass through salty sea water, works in dense region	Interference is more, cannot pass through sea water, works in less dense region
Privacy	In LiFi, light is blocked by the walls and hence will provide more secure data transfer	In WiFi, RF signal cannot be blocked by the walls and hence need to employ techniques to achieve secure data transfer.
Data transfer speed	About 1 Gbps	WLAN-11n offers 150Mbps, About 1-2 Gbps
Frequency of operation	10 thousand times frequency spectrum of the radio	2.4GHz, 4.9GHz and 5GHz
Data density	Works in high dense environment	Works in less dense environment due to interference related issues
Coverage distance	About 10 meters	About 32 meters (WLAN 802.11b/11g), vary based on transmit power and antenna type
System components	Lamp driver, LED bulb and photo detector will make up complete LiFi system.	requires routers to be installed, subscriber devices(laptops,PDAs,desktops) are referred as stations

Methodology elaborated:

In this report, we prototyped a **simplex Li-Fi** communication system which is highlighted in the hardware section below. The system was used to communicate to the Robot to parse various instructions given to it. The location of the robot was refreshed using an efficient tracking algorithm, using an **overhead camera**. The camera acts as localized GPS for the robots while the computer calculates the **heading errors** of the robot to go to the goal and sends out instructions to the robot to update it serially.

(ii) Visual Odometry: Odometry [6] is the use of data from motion sensors to estimate change in position over time, hence an important aspect in robotics. The robots need closed loop control to go from one position to other since open loop techniques are largely inaccurate while doing so.

Traditionally closed loop is established on the robot by using:

- Wheel encoders- Counts are missed while processing and hence not accurate
- Obstacle sensors –data not enough to provide the localized co-ordinates of the robot
- 2D laser sensors–too expensive & require outer periphery wall to calculate relative distance

Visual odometry in turn is efficient, cheap and only requires one overhead camera and a image processing platform. By using Visual odometry robots could now:

- The robots could be sent from A to B efficiently.
- Robots could be assigned IDs on markers and Pose estimation data could be directly acquired.

(iii) Closed Loop Swarming: The odometry data is processed on the computer and sent out serially via Li-Fi. The swarm robot with the correct ID number accepts the errors and corrects itself to reach the goal. The errors are updated every frame and are sent out serially. The Li-Fi packets are broadcasted at 19.2 kbps via the repeater Arduino. Robot swarming [4] [5] is implemented in this report to explore the scope of one to many Li-Fi communication in dynamic and noisy environments. However the closed loop feedback is unidirectional i.e. from mock GPS to robot. The robot cannot send back any data on the reverse channel to the primary source

station. Hence this system is described as Unbalanced Asynchronous Unidirectional communication.

Process of the entire methodology in a while(1) loop

i) Overhead camera sends live feed to the image processing platform (PC)
ii) The tracking algorithm update the local co-ordinate of each robot
iii) Orientation errors are calculated and the error angle is updated [6]
iv) Packets are composed with the Swarm ID and the corresponding error
v) Packet strings are sent out serially to repeater micro-controller
vi) Repeater micro-controller sends out the last updated packet continuously
vii) The packet is decoded after processing on Swarm Robots
viii) The bot take in the error after checking the packet ID
ix) The errors are mapped directly via P controllers to the motors
x) The robot refreshes motor speeds and the entire process repeats.

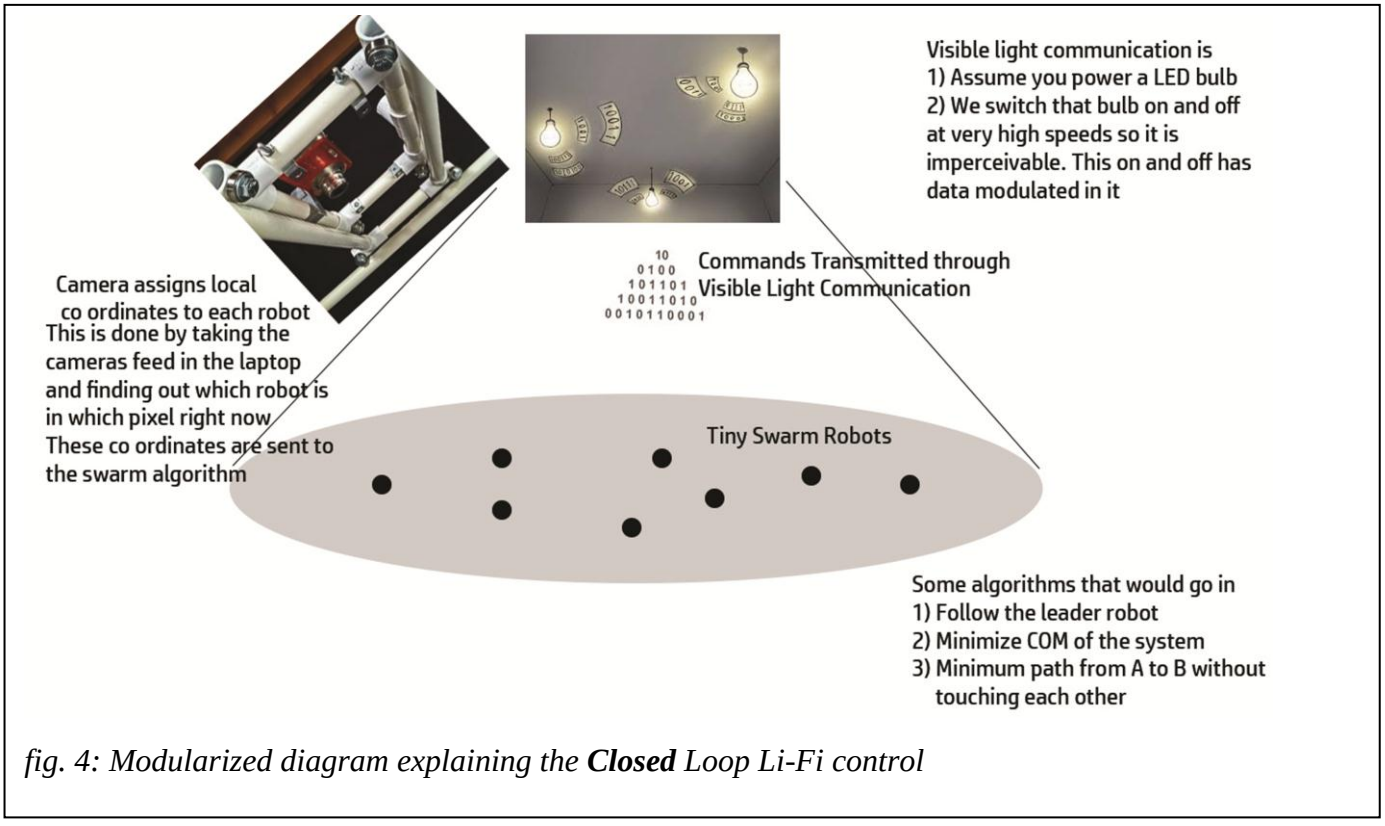


*fig. 4: Modularized diagram explaining the **Closed** Loop Li-Fi control*

III. HARDWARE IMPLEMENTATION

Hardware implementation on this modularized system involves working on three stages:
(i) Li-Fi receiver (ii) Li-Fi transmitter (iii) Swarm bot

(i) Li-Fi Receiver:

The initial tests on the Li-Fi system were explored using a TEMT6000 phototransistor from Vishay Semiconductors. The Ambient Light sensor [16] was capable of giving signals ranging from Analog zero to Analog 5V depending on the luminosity of the surroundings. LiFi receiver was supposed to deal with:[9][11]

1) Amplification
2) Noise-filtering
3) Thresholding
4) Without loading effects on any end
5) Compatible with Arduino signal noise margin levels

Testing the TEMT6000 [15]

1) Sensor was directly plugged into the UART pin of the micro-controller and the transmitting LED was blinked at 1200 bps.
2) The characters on the UART bus arrived, but with noise at times. The setup involved transmitting white LED and TEMT6000 pointing at each other without any gap between them.
3) Gradually the distance was increased and the signals fell beyond recognition on the UART noise margins of AtMega328. To correct this, thresholding was done to bring back the signal.
4) The next step involved filtering the ambient noise (Tubelights and Sunlights). This noise imposed a DC voltage on the readings. This DC was filtered through high pass filter.
5) The phototransistor being slow could not be used beyond 2400bps i.e. baud rate of 300 characters in a second. The packet contained x and y coordinate of robot along with start and stop characters, this means the packet size was around 30bytes. If 10 robots were used, then bot is updated every one second. This delay is too large to refresh for closed loop control. This wasn't feasible since the error by this time would become too large to be corrected and hence robot would never reach its destination. The reason for slow response of the phototransistor was its junction capacitance. The transistor took time to discharge to zero.
6) Hence we chose a component with lesser junction capacitance. We chose to shift to photodiodes.

Phototransistor v/s Photodiode [3]

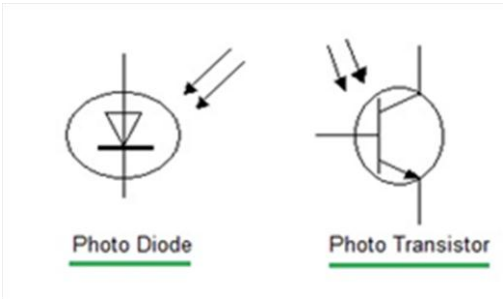


Fig. 5: PhotoDiode v/s PhotoTransistor

Photo diode consists of a normal p-n junction housed in a small enclosure with a transparent window through which light can fall inside. A photo diode is operated in reverse bias in which leakage current increases in proportion to the amount of light falling on the junction. This is result of light energy which breaks the bonds in the crystal lattice of the semiconductor producing electrons and holes. This effect is similar to photo voltaic cell.

Photo transistor [13] can resemble as photo diode giving current amplification due to transistor action. Few of these devices are molded in transparent plastic cases with convex lenses. This convex lens focuses light on the transistor. As a result of this, extra minority carriers are liberated at the reverse biased CB junction (Collector to Base). This generated leakage current is later amplified. If used in this way, connection to base terminal is not needed. Hence many of the photo transistors do not have a base lead. However, Photodiodes are much faster than phototransistors (nanoseconds vs. microseconds)

Gain: Phototransistors have a higher gain. Photodiodes require an amplifier to use.
Temperature Response: Photodiodes vary lesser with temperature

1) A photodiode is a special diode whose resistance varies with light
2) The photodiode also responds to changes in light intensity very quickly
3) An external voltage source must be connected to the photodiode, since it is a passive device
4) Any current flowing through the device will increase as the light increases because the resistance of the photodiode decreases.

TEMT6000

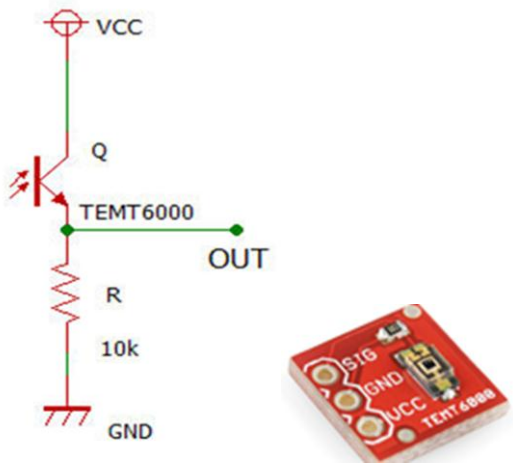


Fig. 6: TEMT6000

A phototransistor is a light-sensitive transistor. A common type of phototransistor, called a photobipolar transistor, is in essence a bipolar transistor encased in a transparent case so that light can reach the base–collector junction.

TEMD6200[15]

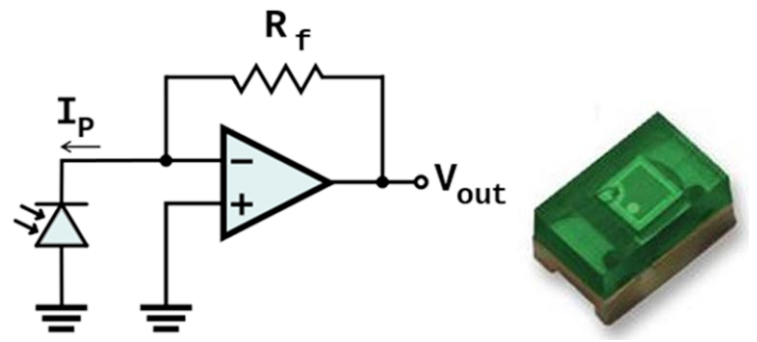


Fig. 7: TEMD6200

TEMD6200FX01 is a high speed and high sensitive PIN photodiode in a miniature flat plastic package. It is spectral sensitivity is closely matched to the human eye [15].
The small reverse current is converted to voltage using trans impedance amplifier, the output of amplifier is given to notch filter, and is further filtered by active devices and hence output signal is received.

BASIC CHARACTERISTICS (T_{amb} = 25 °C, unless otherwise specified)

PARAMETER	TEST CONDITION	SYMBOL	MIN.	TYP.	MAX.	UNIT
Breakdown voltage	I_R = 100 µA, E = 0 lx	$V_{(BR)}$	16			V
Reverse dark current	V_R = 10 V, E = 0 lx	I_{ro}		0.1	5	nA
Diode capacitance	V_R = 0 V, f = 1 MHz, E = 0 lx	C_D		60		pF
	V_R = 5 V, f = 1 MHz, E = 0 lx	C_D		24		pF
Reverse light current	E_e = 1 mW/cm², λ = 550 nm, V_R = 5 V	I_{ra}		1		µA
	E_V = 100 lx, CIE illuminant A	I_{ra}	0.03	0.04	0.09	µA
Angle of half sensitivity		φ		± 60		deg
Wavelength of peak sensitivity		λ_p		540		nm
Range of spectral bandwidth		$\lambda_{0.5}$		430 to 610		nm
Rise time	U_R = 5 V, R_L = 50 Ω, TLMW3300	t_r		150		ns
Fall time	U_R = 5 V, R_L = 50 Ω, TLMW3300	t_f		150		ns

Table no 2: Characteristics of TEMD6200 from Vishay

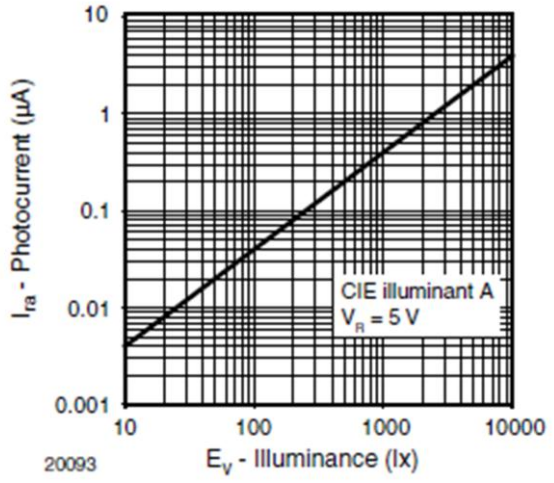


Fig no 8: Photocurrent v/s Luminance of TEMD6200

Table 3: TEMT6000 v/s TEMD6200 [15]			
Rev.A: TEMT6000- transistor		Rev.B: TEMD6200- diode	
Advantage	Disadvantage	Advantage	Disadvantage
No trans-impedance reqd. to convert photocurrent to voltage	High junction capacitance, Slow discharge rates- slows baud	Low junction capacitance, Fast discharges, Faster baud	Trans-impedance stage reqd. to convert photocurrent to voltage

Trans-impedance amplifier:

In electronics, a transimpedance amplifier (TIA) [13] is a current-to-voltage converter,most often implemented using an operational amplifier. Current-to-voltage converters are used with sensors that have a current response that is more linear than the voltage response. This is the case with photodiodes where it is not uncommon for the current response to have better than 1% linearity over a wide range of light input. The transimpedance amplifier presents low impedance to the photodiode and isolates it from the output voltage of the operational amplifier. In its simplest form a transimpedance amplifier has just a large valued feedback resistor, R_f. The gain of the amplifier is set by this resistor and because the amplifier is in an inverting configuration, has a value of $-R_f$. There are several different configurations of transimpedance amplifiers, each suited to a particular application [13]. The one factor they all have in common is the requirement to convert the low-level current of a sensor to a voltage. The gain, bandwidth, as well as current and voltage offsets change with different types of sensors, requiring different configurations of transimpedance amplifiers.

Notch Filter:

In signal processing, a band-stop filter or band-rejection filter is a filter that passes most frequencies unaltered, but attenuates those in a specific range to very low levels. It is the opposite of a band-pass filter [14]. A notch filter is a band-stop filter with a narrow stopband (high Q factor).Narrow notch filters (optical) are used live sound reproduction (public address systems, or PA systems) and in instrument amplifiers (especially amplifiers or preamplifiers for acoustic instruments such as acoustic guitar, mandolin, bass instrument amplifier, etc.) to reduce or prevent audio feedback, while having little noticeable effect on the rest of the frequency spectrum. Other names include 'band limit filter', 'T-notch filter', 'band-elimination filter', and 'band-reject filter'. Typically, the width of the stop band is 1 to 2 decades (that is, the highest frequency attenuated is 10 to 100 times the lowest frequency attenuated). Active notch filters

have been used in the past for applications like elimination of 50- and 60-Hz hum components. They have proven to be somewhat problematic from the standpoints of centre frequency (f0) tuning, stability, and repeatability.

Filter Circuit:

The filter circuit is used for removing DC signals from the environmental light. Bandpass filters are designed around the Receiver circuit to block Low frequency DC noise using High Pass filters and High frequency motor noise/harmonic noise/environmental noise using Low pass filters. The filters severely attenuate the noise at their 3db cut-off and hence isolate the cascaded circuits from the distortion noises [14]. On the way to complete a flawless Li-Fi system, we encountered an issue as soon as we paired the Li-Fi circuits on the swarm bots. The motors were cutting the power lines at a frequency of 500Hz and the on and off for the PWM on power lines was distorting the Li-Fi circuits with exactly the same pulses. The PWM distortion was superimposed on the UART data and it caused problems while thresholding the final data before reaching the arduino. Hence the receiver circuit was coupled with a huge decoupling capacitor across its power lines to prevent these dips.

One major drawback to working with motors is the large amounts of electrical noise they produce. This noise can interfere with the sensors and can even impair the microcontroller by causing voltage dips on your regulated power line. Large enough voltage dips can corrupt the data in microcontroller registers or cause the microcontroller to reset.

The main source of motor noise is the commutator brushes, which can bounce as the motor shaft rotates. This bouncing, when coupled with the inductance of the motor coils and motor leads, can lead to a lot of noise on your power line and can even induce noise in nearby lines.

There are several precautions can be taken up to help reduce the effects of motor noise on the system:

1) Solder capacitors across the motor terminals. Capacitors are usually the most effective way to suppress motor noise, and at least one capacitor should be soldered across the motor terminals. Typically anywhere from one to three 0.1 µF ceramic capacitors, soldered as close to the motor casing as possible can be used. For applications that require bidirectional motor control, it is very important that polarized capacitors are not used.

- If single capacitor is used, it should be directly soldered across the motor terminals.
- For greater noise suppression, two capacitors can be soldered to the motor; one from each motor terminal to the motor case. For the greatest noise suppression, all three capacitors can be soldered.

2) Motors and Power leads should be kept as short as possible. Noise can be decreased by twisting the motor leads so they spiral around each other.

3) Motor power and signal wires must be isolated and kept away from each other. Noisy motors induce considerable spikes in the power rails and signals are distorted.

4) Decoupling capacitors can be placed (also known as “bypass capacitors”) across power and ground near any electronics that must be isolated from noise. Closer to electronics (uC) the better and higher the capacitance, the better. It is recommended to use electrolytic capacitors of at least several hundred µF.

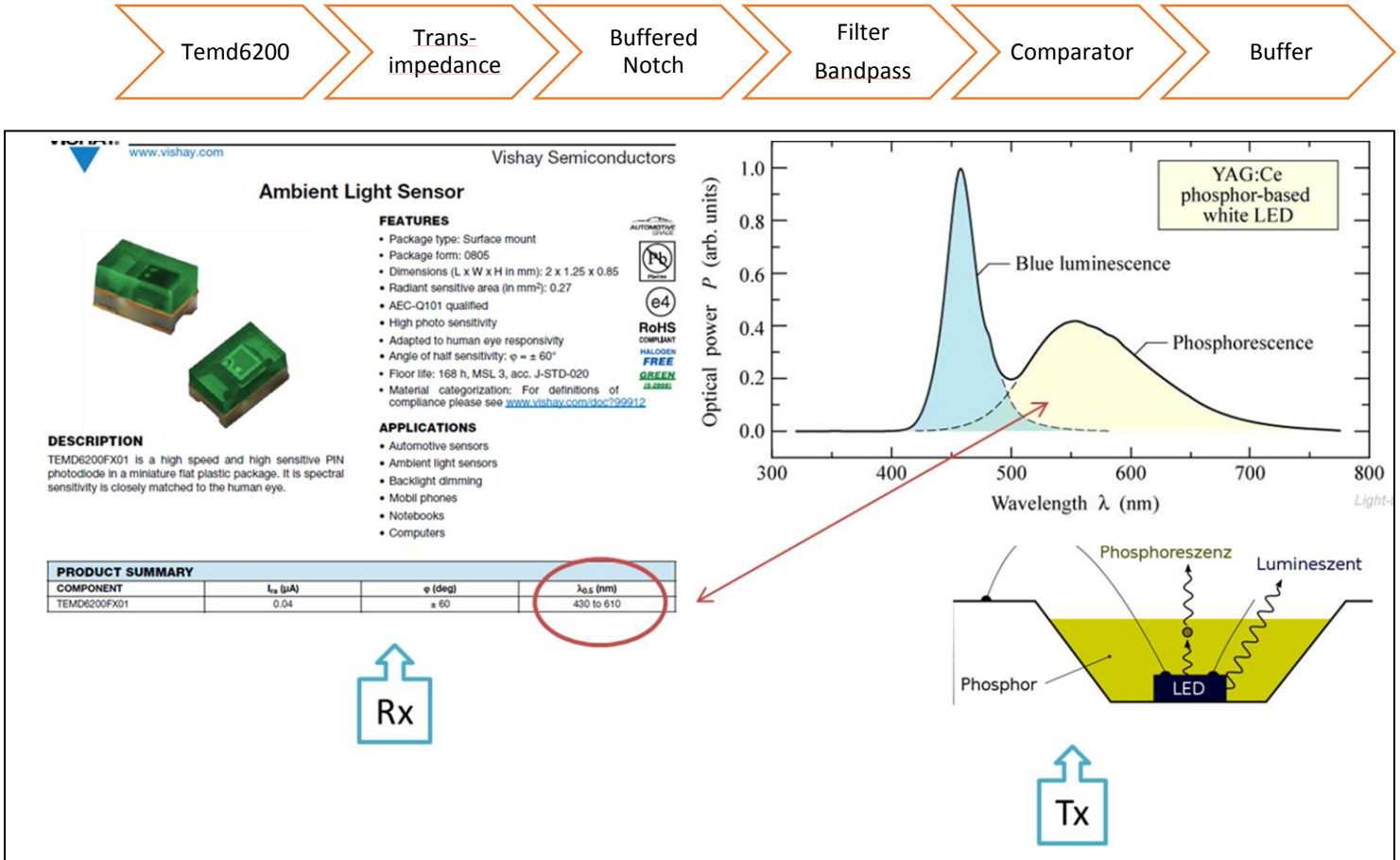


Fig. 9: Choosing the compatible Transmitting and Receiving Diodes

Bottlenecks on the Baud rates were imposed because of –

1) Op-Amp's Slew-rates [13]
   Increased baud rates on the receiver's end matched the slew rates on the opamp and it became difficult to threshold in the later cascaded op amp stages, which imposed a problem while speeding up the system.
   Problems while removing bottlenecks on the Op-Amps– If the Op-Amps perceived faster – it could also perceive high frequency noise, which required more filter cascade. At the current level we restricted the baud at 9.6k

2) Processing Power on the Rx end
   The Receiver end after thresholding sends the UART signals to the Atmega328 [7]. The Atmega is supposed to not only parse the UART packets but also the PID subroutines for every input going to the input. Increased baud rates could make the Atmega skip UART packets.

(ii) Li-Fi transmitter:

Li-Fi transmitter [9] [11] is a microcontroller based switching device which cuts the Light at high baud rates as instructed by the UART packets. Apart from just switching, the transmitter stores the last updated packet from the computer to the Atmega328 temporarily. The last packet is continuously unless a new packet arrives from the computer. This keeps the receiver busy in parsing the packets and hence it stays synchronized [10]. Also, the link is continuously occupied and corruption of data due to random millivolt signals does not happen. If the receiver is not flooded with packets, it might threshold millivolt signals due to noise and give garbage in the receiver buffer.

Two transmitters were prototyped during the duration of testing the robots:

Table 4.	***Camera rev A***	***Camera rev B***
Resolution	*VGA: 480x480*	*HD : 1024x768*
Features	*Flash/no focus no IR filter (easy cam-shift and brightest point)*	*Auto Focus (easy marker detection) Algo2*
	Tx LEDs rev A	***Tx LEDs rev B***
Power	*36V 7W*	*27V 20W*
Type	*Phosphor*	*Phosphor*
Features	*W/O Heat SINK*	*With heat sink*

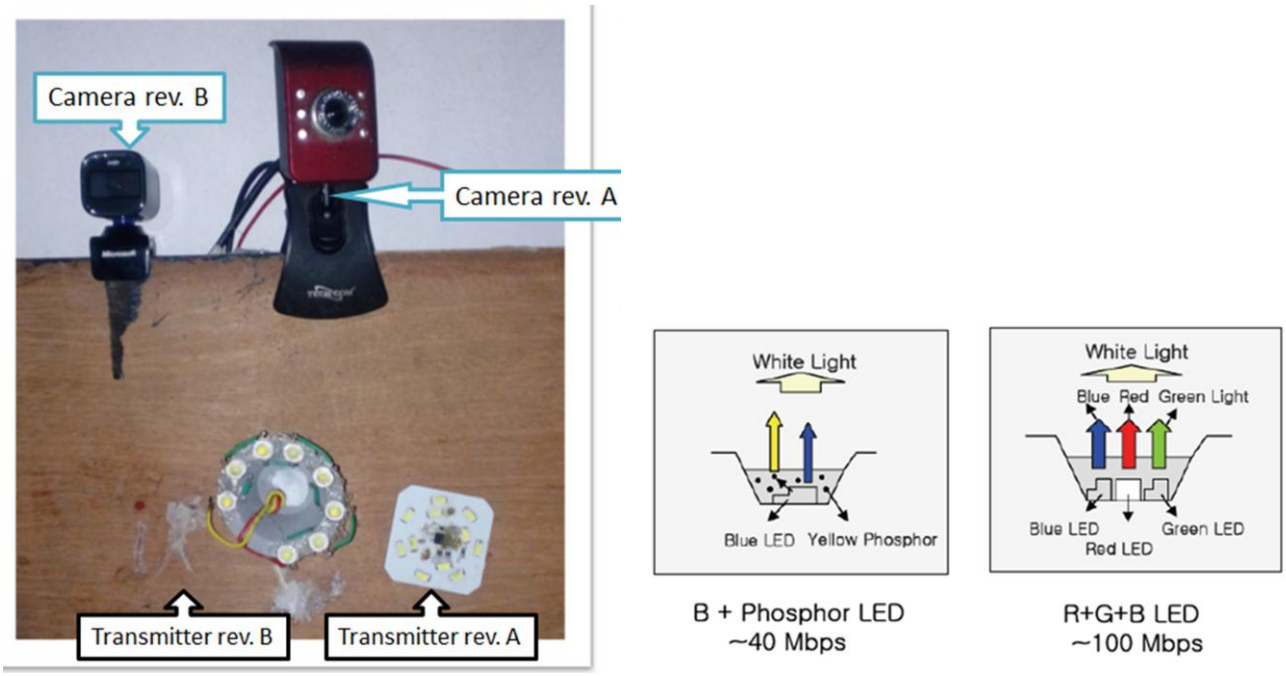


Fig. 10: Transmitting Phosphor LED reference [11]

IV. SETTING UP THE ENVIRONMENT

Setting up environment involved integrating all the modules together to finally bring together the exact goal of the project. A setup was done with Transmitter mounted at 1 meter above the workspace. Camera and Transmitting bulb with Repeater microcontroller was mounted overhead. While the robots below were flashed with the boot-loader [8].

The robots were fabricated using a CNC machine to make sure the motors are mounted at exact angles for equal traction on both ends. The weight distribution had to be taken care of, because the motors ran without a gearbox and unequal distributions could directly reflect on the robots dynamics. Each robot had two shields: The two shields were mounted on plastic studs to prevent shorting out while mounting. One shield was the AtMega 328 with the Motor driver (L293D) and the other one was the LiFi –RX shield. The footprints of the electrical CAD for the shields were made to match the chassis. A castor wheel was coupled in the front of the robot, for weight distribution so it doesn't topple while thrusting forward. Rubber grips on the motors shaft provide enough traction to prevent slips while traversing the workspace. [4] [5]

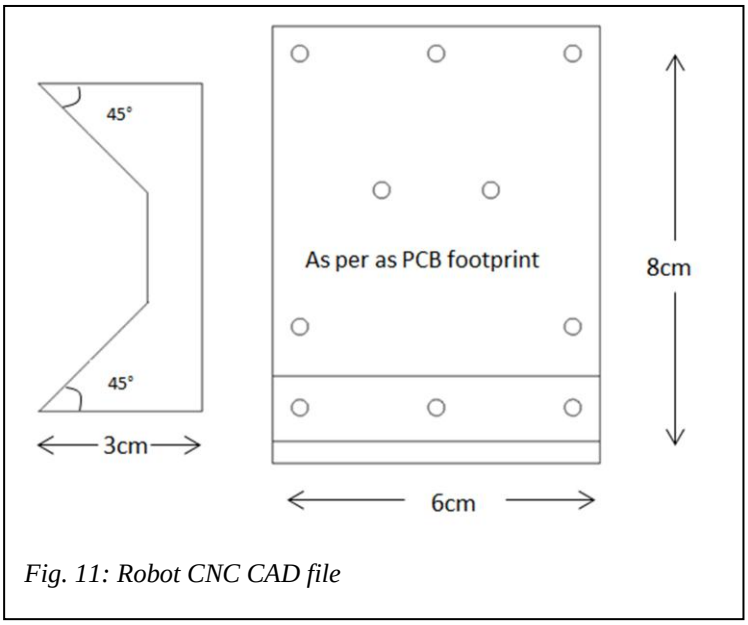


Fig. 11: Robot CNC CAD file

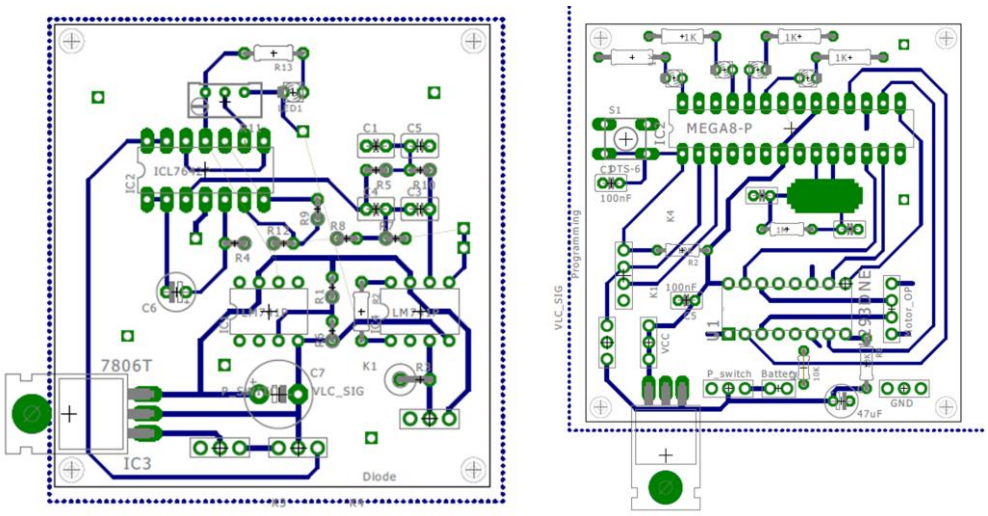
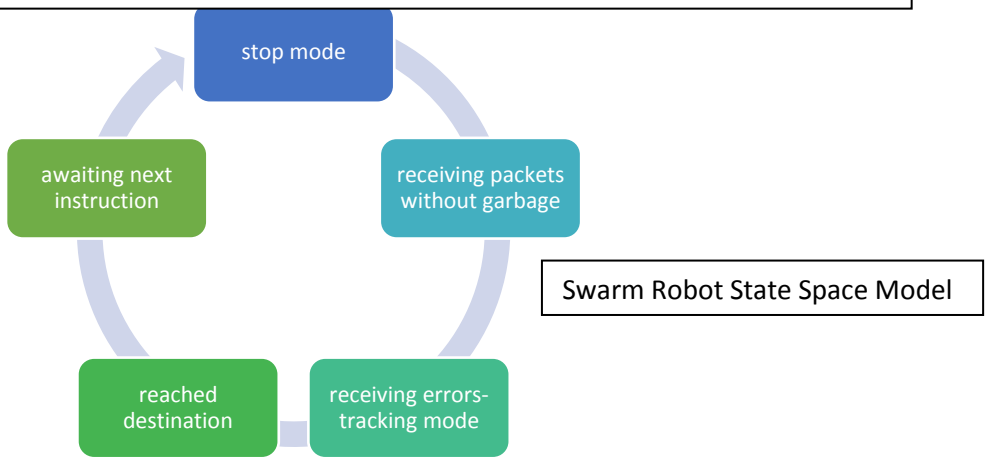


Fig. 12 : Li-Fi receiver shield (left) and Atmega 328 with L293D (right)



Swarm Robot State Space Model

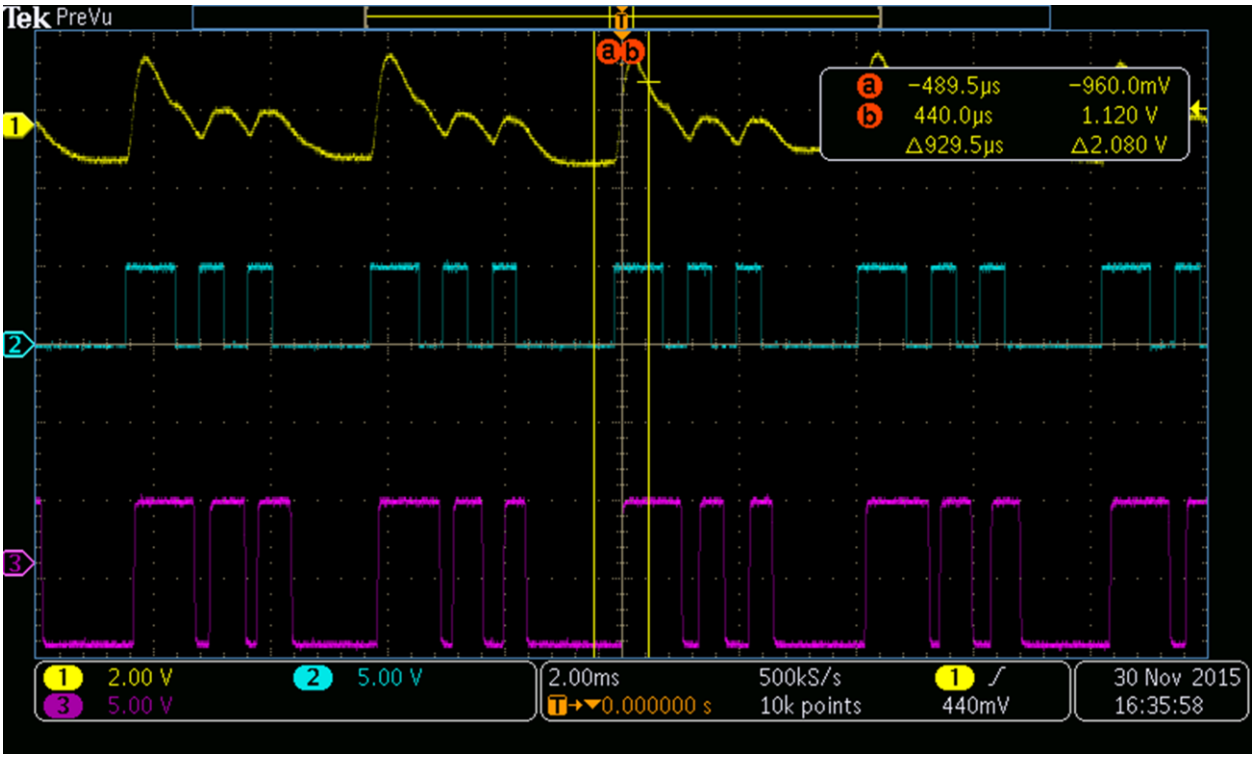


Fig. 13: Checking for LiFi packets

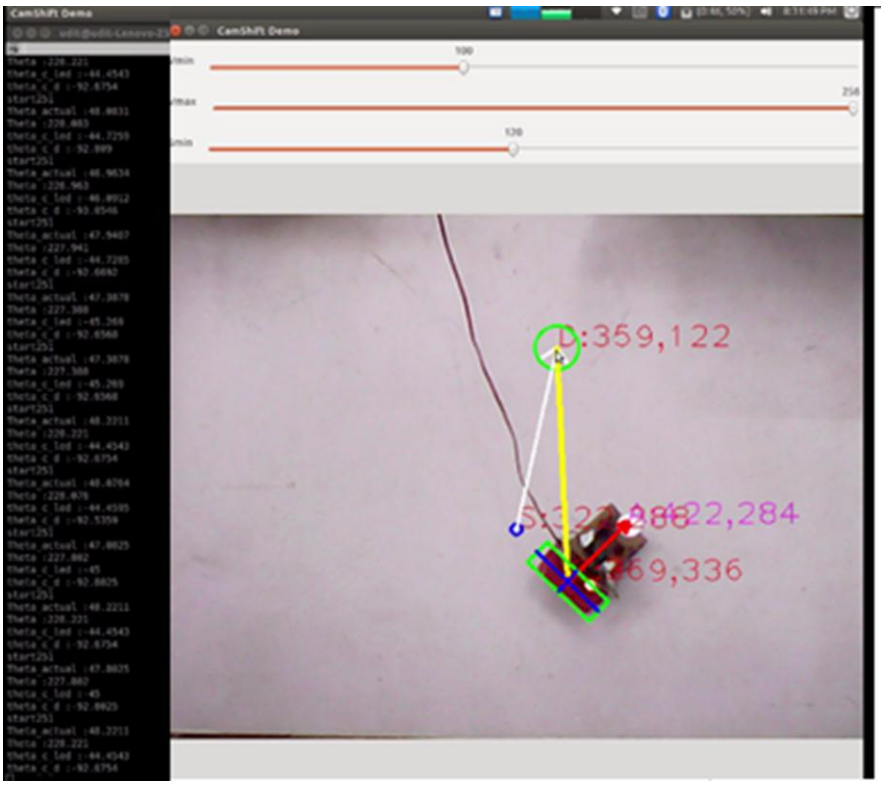


Fig. 14: Testing the robot using CAMshift

Errors were sent in to GotoGoal P controllers which minimize the orientation error of each robot [6]

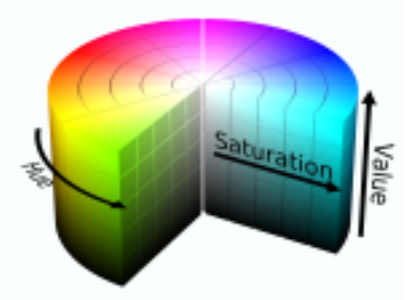
IV. SOFTWARE IMPLEMENTATION

(i) Algorithms tested

Table 5- Algorithms	**Rev A**	**Rev B**	**Rev C**
Detection	HSV thresholding	Camshift	arUco
Orientation info	NA	IR LED, Angle of CM and brightest point to give angle of robot	Pose estimation Edges of marker gives out angle of robot
Threading	CPU	Threaded to GPU	CPU
FPS	15	38	25
Error to destination	$Theta_{previouswithgoal}$ - $theta_{currentwithgoal}$	$Theta_{bot}$- $theta_{goal}$	$Theta_{bot}$- $theta_{goal}$
Efficiency	Least	Better	Best
Robot IDs	NA	IR LED of the destination robot is turned on	ARuco hamming codes
Multi-robot control	Not possible	Complex	Simple but requires Camera caliberation
Flowchart	1. Threshold via HSV 2. Convert to binary workspace 3. Erode & Dilate 4. cv::findContours 5. Moments[10] = centre of mass 6. Plot lines and select goal points	1. Backprojection with histogram normailization 2. meanShift to cluster contour 3. Track contour	1. Adaptive Thresholding 2. Findcontours 3. Polygonal approximation of 4 corners 4. Refine corners using subpixel interpolation 5. Frontal perspective of marker by homography 6. Otsu thresholding and hamming decode

*a) **Rev. A** described in detail:*

HSV is a three value format for describing a color with the properties “hue”, “saturation” and “value”. The first property “Hue” is given as an angle from 0° to 360° of a color wheel where 0° is pure red, 120° is pure green and 240° is pure blue. For example, purple would be half way between blue and red, i.e. 300°. The other two properties are a little harder to describe but we can think of “saturation” as saying how strong or pale the color is, and “value” says how bright or dark the color is. In OpenCV the HSV format is stored as 3 unsigned 8 bit values. For saturation and value the ranges 0-255 are used. For the hue component we also have a maximum range of 0 to 255 but hue is a value from 0 to 360 so OpenCV stores the hue as half the angle, i.e. range 0 to 180.



Color Thresholding:

Thresholding is a fundamental image processing technique whereby we replace each pixel in an image with a “yes” or “no” value depending on whether that pixel meets some criteria. We effectively create a black and white version of the original image where “white”, i.e. value 255, means “yes” and “black” (0) means “no”.

OpenCV has the function cv::findContours() that finds all the “contours” or edges of an image as an array of points. In fact it creates an array of these arrays of points, each set of points represents a distinct region in the image.

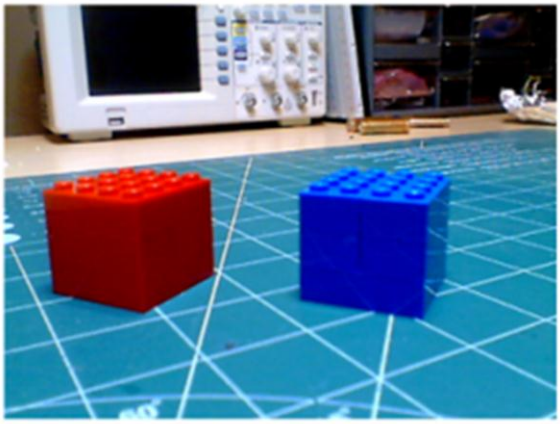
cv::inRange() can be used for thresholding

```
cv::Scalar   min(hueMinValue, satMinValue, volMinValue);
cv::Scalar   max(hueMaxValue, satMaxValue, volMaxValue);
cv::Mat threshold_frame;
cv::inRange( hsv_frame, min, max, threshold_frame);
```

The cv::inRange() operation simply compares each HSV value in the frame and replaces it with the value 0 if it is outside the min/max values or with 255 if it is inside the range.

Here is an example:

```
cv::Scalar   min(220/2, 0, 0);
cv::Scalar   max(260/2, 255, 255);
cv::Mat threshold_frame;
cv::inRange( hsv_frame, min, max, threshold_frame);
```



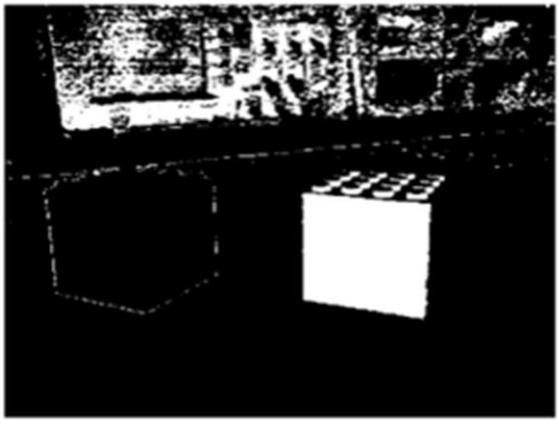


Fig. 15: HSV Thresholding example

Error of the robot without orientation: The HSV thresholding only indicated the center of robot without any orientation data. The current error the robot has to follow is compared to the previous origin position error. This method had a single flaw that the robot might be on the same line as the destination, but it might be facing elsewhere- still the error to the destination is updated as zero. This way the method is inefficient and robot might not reach its goal.

(Courtesy : Kyle Hounslow Channel on OpenCV object tracking)

*b) **REV. B** Described in Detail*

Continuously Adaptive Mean SHIFT

Courtesy: By ERIC | *Published: SEPTEMBER 18, 2013*

CamShift is a tracking algorithm, which is based on MeanShift algorithm, camShift does nothing but meanShift in every single frame of a video, and records the result.
CamShift algorithm includes these three parts:

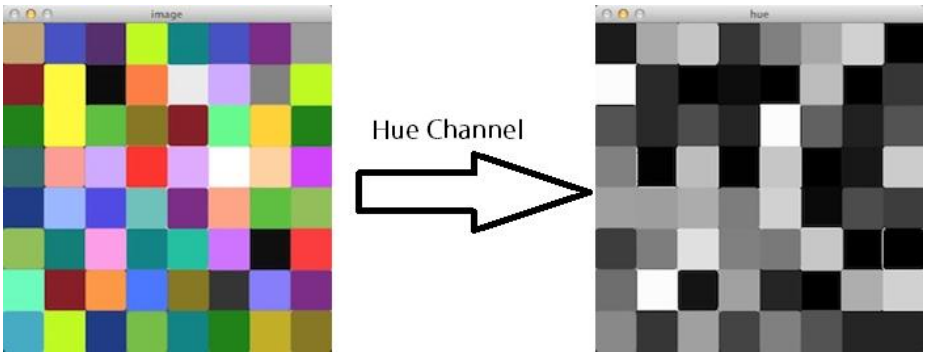
1. Back Projection
2. MeanShift
3. Track

1. Back Projection.
Back projection is a method which using the histogram of an image to show up the probabilities of colors may appear in each pixel

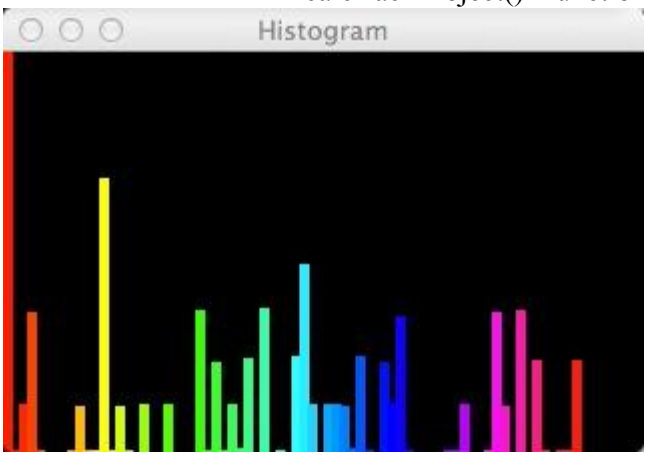
example:

```
1 cvtColor(image, hsv, CV_BGR2HSV);
2 int ch[]={0,0};
3 hue.create(hsv.size(), hsv.depth());
4 mixChannels(&hsv, 1, &hue, 1, ch, 1);
5 calcHist(&hue, 1, 0, Mat(), hist, 1, &hsize, &phranges);
6 normalize(hist, hist, 0,255, CV_MINMAX);
7 calcBackProject( &hue, 1, 0, hist, backproj, &phranges, 1,true );
```

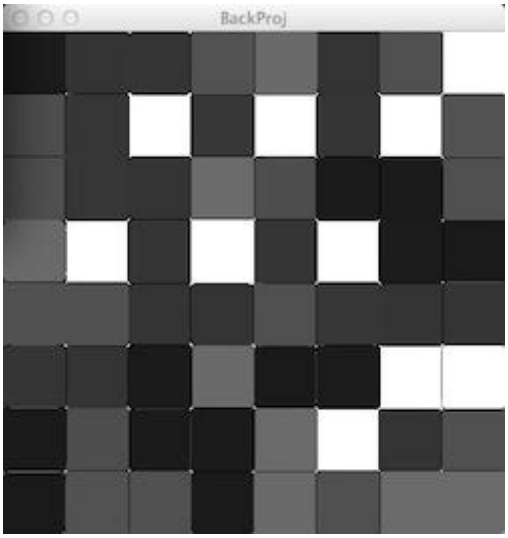
First the picture space is converted to HSV space. Secondly, the H channel is split, as a single grayscale image, and after getting its histogram, it is normalized. Thirdly, “calcBackProject()” function is used to calculate the back projection of the image.



Histogram output :



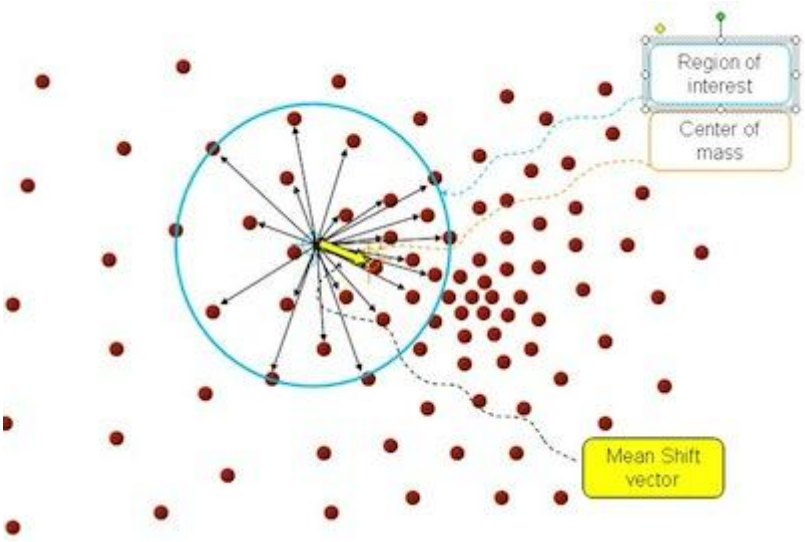
"calcBackProject()" function actually calculates the weight of each color in the whole picture using histogram, and changes the value of each pixel to the weight of its color in whole picture. For instance, if one pixel's color is, say yellow, and the color yellow's weight in this picture is 20%, that is, there are 20% of pixels' color in the whole picture is this kind of yellow, we change this pixel's value from yellow to 0.2 (or 0.2*255 if using integer), by doing this method to all pixels, the back projection picture can be obtained.



2. MeanShift

MeanShift is nothing but an algorithm which finding modes in a set of data samples representing an underlying probability density function (PDF) in R^N. It is a nonparametric clustering technique which does not require prior knowledge of the number of clusters, and does not constrain the shape of the clusters.

Imagine we are in a space, and the dimension of this space is d, (of course, d maybe bigger than 2) and there are a lot of points in this space, what we are going to do is clustering these points. We now make a sphere which center is any of the points, and radius is, say, h. Because we are in a high-dimensional space, so what we just made is a high-dimensional sphere. By now, every single point inside this space can be seen as a vector (directed line), and the sum (normalized) of these vectors is called mean shift. By this mean shift vector, the current mass center can be found.



In c++, the current mass center can be calculated like this:

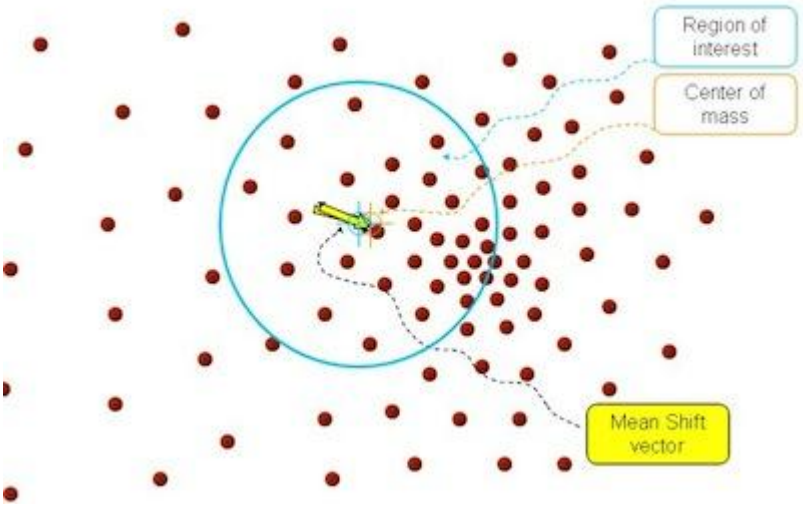
```
01 M00 = 0.0;
02 M10 = 0.0;
03 M01 = 0.0;
04 for(int i = 0; i < probmap.height; i++){
05     for(int j = 0; j < probmap.width; j++){
06         M00 += probmap.at(i, j);
07         M10 += i * probmap.at(i, j);
08         M01 += j * probmap.at(i, j);
09     }
10 }
11 Center_x = M01 / M00;
12 Center_y = M10 / M00;
```

Above is the core of meanShift algorithm, so the whole algorithm is:

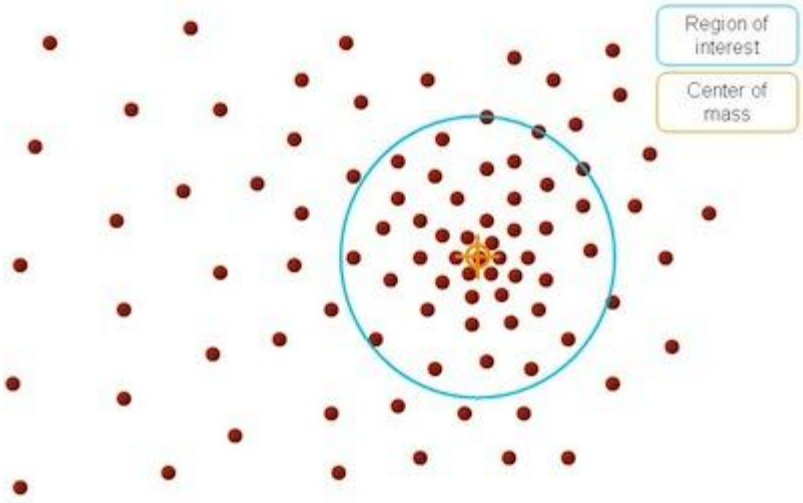
a. Initialize the sphere, including the center and radius.

b. Calculate the current mass center.

c. Move the sphere's center to mass center.



d. repeat step b and c, until converge, that is, current mass center after calculate, is the same point with center of sphere.



In OpenCV, meanShift function is something like this:

```
1 CV_IMPL int
2    cvMeanShift( const void* imgProb, CvRect windowIn,
3                 CvTermCriteria criteria, CvConnectedComp* comp );
```

In which, imgProb is 2D object probability distribution, which is the result of back projection above; windowIn is CvRect of window initial size; numIters means that if the algorithm iterates this many times, stop, that prevents in some cases the function just repeats and repeats; windowOut is the location, height and width of converged window.

3. Track

The last step is tracking, if there is a video, or frames captured by a web camera, meanShift algorithm can be used on every single frame, and the initial window of each frame is just the output window of the prior frame.

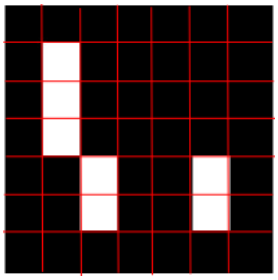
By using OpenCV camshift() function, we can get a RotatedRect, which is defined in OpenCV like:

```
01 class RotatedRect
02 {
03 public:
04     // constructors
05     RotatedRect();
06     RotatedRect(const Point2f& _center, const Size2f& _size, float _angle);
07     RotatedRect(const CvBox2D& box);
08     // returns minimal up-right rectangle that contains the rotated rectangle
09     Rect boundingRect() const;
10     // backward conversion to CvBox2D
11     operator CvBox2D() const;
12     // mass center of the rectangle
13     Point2f center;
14     // size
15     Size2f size;
16     // rotation angle in degrees
17     float angle;
18 };
```

This can also get an angle of the rectangle, which means we can track the orientation of the target, this is a very useful feature

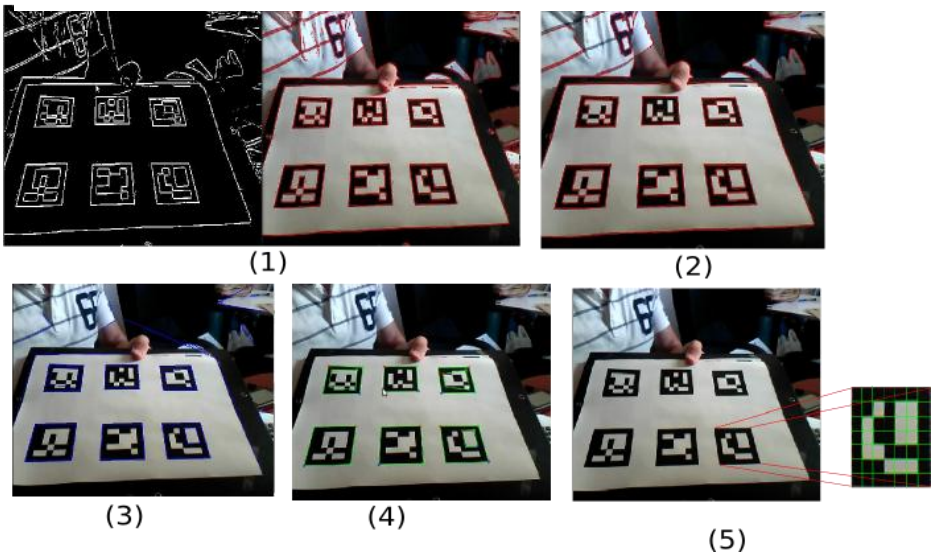
c) ***REV. C*** *Described in Detail*

Marker coding



Each marker has an internal code given by 5 words of 5 bits each. The codification employed is a slight modification of the Hamming Code. In total, each word has only 2 bits of information out of the 5 bits employed. The other 3 are employed for error detection. As a consequence, we can have up to 1024 diferent ids.

The main difference with the Hamming Code is that the first bit (parity of bits 3 and 5) is inverted. So, the id 0 (which in hamming code is 00000) becomes 10000 in our codification. The idea is to prevent a completely black rectangle from being a valid marker id with the goal of reducing the likelihood of false positives with objects of the environment.



Detection Process in ArUco of Rev C.

The marker detection process of ArUco is as follows:

- Apply an Adaptive Thresholding so as to obtain borders (Figure 1)
- Find contours. After that, not only the real markers are detected but also a lot of undesired borders. The rest of the process aims to filter out unwanted borders.
    - Remove borders with an small number of points (Figure 2)
    - Polygonal approximation of contour and keep the concave contours with exactly 4 corners (i.e., rectangles) (Figure 3)
- Sort corners in anti-clockwise direction.
- Remove too close rectangles. This is required because the adaptive threshold normally detects the internal and external part of the marker's border. At this stage, we keep the most external border. (Figure 4)
- Marker Identification

- Remove the proyection perspective so as to obtain a frontal view of the rectangle area using an homography (Figure 5)
- Threshold the area using Otsu. Otsu's algorithms assume a bimodal distribution and finds the threshold that maximizes the extra-class variance while keeping a low intra-class variance.
- Identification of the internal code. If it is a marker, then it has an internal code. The marker is divided in a 6x6 grid, of which the internal 5x5 cells contain id information. The rest correspond to the external black border. Here, we first check that the external black border is present. Afterwards, we read the internal 5x5 cells and check if they provide a valid code (it might be required to rotate the code to get the valid one).

- For the valid markers, refine corners using subpixel interpolation
- Finally, if camera parameters are provided, it is computed the extrinsics of the markers to the camera.

(ii) Serial Packets and Parsing [10]

Table 6	*Header*	*Instruction*	*Robot ID*	*Error*	*Trailer*
Length	5 bytes	2 bytes	3 bytes	3 bytes	4 bytes
Description	"start" Start of frame	"30" : go "25" : stop	arUco robot ID	Heading error Offset by 180	"stop" End of frame
Example of typical frame	Start	30	768	200	stop

The frame composed: "start30768200stop" indicates

Robot no **768** should go in **Dynamic mode**

With an error of **200**-180=20

Pwmright= fwd+20

Pwmleft=fwd-20

If fwd=80

Motorright will go forward with a speed of 100

MotorLeft will go forward with a speed of 60

And robot will differentially take a left.

Error Refresh Rate per robot

- The error carried in the frame is 3/(5+2+3+3+4)th part of the frame.
- The baud rate currently is 19.2kbps
- The frames are sent out sequentially to N robots.
- Error refresh rate will hence be $\frac{(19200 \times 3)}{(N \times 17)}$

Packets are UART encoded. They are simple bitstreams of the ASCII values of the characters being sent from the computer via LiFi. The packet “O” “K” looks like this:

The ASCII value of “O” and “K” is simply surrounded by start and stop sync bits [10]

(iii) Path planning Algorithms

Table 7	Advantages	Disadvantages
A* algorithm	Heuristic analysis and solution	Works only on Static frame CPU Intensive
Diversion Node	No load on CPU Dynamic frame	Obstacle avoidance not guaranteed Not the shortest path to goal
Real time force field calculations	Shortest path to goal Not very CPU intensive Dynamic frame	Shortest path with obstacle avoidance

Real time force field calculations

$$F = Q_r \left(C \frac{r_t}{||r_t||^2} + Q_r \sum_{i=1}^{K-1} \frac{r_i}{||r_i||^2} + \sum_{i=1}^{L} B_i \right)$$

Q_r is the robot potential, C is the target potential, r_t is a vector from the target to the robot, r_i is a vector from the moving robot to the i-th robot and B_i is the i-th barrier term. Each robot, barrier and target is assigned a constant potential, which corresponds to coefficients Q_r, B_r (which will be discussed soon) and C. Also, K is the number of robots on the testbed and L is the number of boundaries in the environment. The first two terms are simply a direct attraction or repulsion inversely proportional to distance. However, the boundary term is a bit different because we want the robot to go around the boundary instead of directly away from it. As shown in Figure 4.1, the robot feels a force perpendicular to the barrier instead of directly repulsing it. The robot only detects boundaries in the semicircle where the robot is facing, so this ensures the robot moves around the boundary.

$$B_i = B_r \left(\frac{r_B^{\perp}}{||r_B||^2} \right)$$

V. CONCLUSION

We successfully explored the **scope of Li-Fi communications** by demonstrating that unintelligent swarm systems could be easily co-ordinated by a single light source while also establishing a closed loop over the dynamics of the robot. The off the shelf components used while designing a **19.2 kbps Li-Fi system** demonstrate the project can be replicated easily at a Do it Yourself level. Wi-Fi can be less cluttered and Li-Fi bandwidths can be exploited for nominal data transfers. Applications can range from Indoor Localization to Headlights on Self Driving cars.

The ease with which the augmented reality markers were used as compared to our initial method of tracking with camshaft and brightest point on the screen was explored.

The feasibility of establishing a **closed loop system** on dumb- swarm robots which only have motors on them and no kind of feedback was explored using our tracking algorithm. The **mock-GPS** recognized the ARuco markers at a large FPS and motors on the robots were updated with a nominal delay. The mock GPS system was setup using an overhead camera and the GPS co-ordinates were sent over serial communication to the Li-Fi transmitting module.

VI. REFERENCES

Inspiration

[1] http://spectrum.ieee.org/telecom/internet/lifi-gets-ready-to-compete-with-wifi/

[2] https://standards.ieee.org/findstds/standard/802.15.7-2011.html on the infrastructure level implementation of Lifi

[3]https://www.ted.com/talks/harald_haas_wireless_data_from_every_light_bulb

Swarm Robots References

[4] Michael Rubenstein on "Kilobot: A low cost scalable robot system for collective behaviors" at "Robotics and Automation (ICRA), 2012 IEEE International Conference"

[5] Bin Feng, Yuan Gao on "Development of Strategy Software Algorithm Simulators for Multi-Agent System in Dynamic Environments Technology and Communication" -2013 from Vaasan Ammattikorkeakoulu VAMK, University of Applied Sciences

[6**]** Paril Jain, "Odometry and motion planning for omni drive robots" Computational Intelligence on Power, Energy and Controls with their impact on Humanity (CIPECH), Nov 2014

[7] Datasheet Atmel co-operation Atmega 328 datasheet for PWM and USART

[8] Standalone Arduino reference - https://www.arduino.cc/en/Main/Standalone

LiFi References

[9] Harald Haas on "Principles of LED Light Communications Towards Networked Li-Fi" published with "Cambridge University Press" in 2015 being editor of IEEE Transactions on Communications and IEEE Journal of Light wave Technologies

[10] Stefan Mangold, "Linux Light Bulbs: Enabling Internet Protocol Connectivity for Light Bulb Networks" at "Workshop on Visible Light Communication Systems (VLCS)" on September 7, 2015

[11] Stefan Mangold, "Using Consumer LED Light Bulbs for Low-Cost Visible Light Communication Systems" at "Workshop on Visible Light Communication Systems (VLCS)" on September 7, 2014

[12] Stefan Mangold, " LED-to-LED Visible Light Communication Networks" at "ACM International Symposium on Mobile Ad Hoc Networking and Computing (ACM MobiHoc)" on August 1, 2013

[13] Luis Orozco on "Optimizing Precision Photodiode Sensor Circuit Design" published in "Application notes of Analog Devices, Inc" for Transimpedance stage in Receivers

[14] Bruce Carter on “Filter Design in Thirty Seconds” published in “High Performance Analog Texas Instruments”

[15] Datasheet: TEMD6200FX01 by Vishay Semiconductors

[16] Ambient Light Sensors - Circuit and Window Design Vishay – filtering amplifying and window size

[17] http://www.lifi.eng.ed.ac.uk/li-fi-research-publications