EVELOPMEN .E ROBOT

SYED MUSA KAZIM BIN SYED ZAINAL ABIDIN

Report submitted in partial fulfillment of the requirements
for the award of Bachelor of Mechatronics Engineering

Faculty of Manufacturing Engineering

University Malaysia Pahang

SEPTEMBER 2013

SUPERVISOR'S DECLARATION

We hereby declare that we have checked this project report and in our opinion this project is satisfactory in terms of scope and quality for the award of the degree of Bachelor of Mechatronic Engineering with Manufacturing.

Signature : ____________________

Name of Supervisor : PROF. DR. ZAHARI TAHA

Position: DEAN FACULTY OF MANUFACTURING

Date : SEPTEMBER 2013

Signature : ____________________

Name of Panel : DR. RAZLAN BIN YUSOFF

Position: VICE DEAN FACULTY OF MANUFACTURING

Date : SEPTEMBER 2013

STUDENT'S DECLARATION

I hereby declare that the work in this report entitled "Modelling, Simulation and Experimental Varification of Variable Parameter of Susupension Spring" is the results of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted for award of other degree.

Signature : ______________________

Name : SYED MUSA KAZIM BIN SYED ZAINAL ABIDIN

ID Number : FB09030

Date : SEPTEMBER 2013

Dedicated to my beloved father, Syed Zainal Abidin Bin Syed Hussin,
mother, Sharifah Hazlina Binti Syed Rashid, brothers and sisters,
and my supervisor, Prof. Dr. Zahari Bin Taha

ACKNOWLEDGEMENTS

First and foremost, I am very grateful to the almighty Allah S.W.T. for giving me strength and opportunity to accomplish my Final Year Project.

I would like to take this opportunity to express gratitude to my dedicated supervisor, Prof. Dr. Zahari Bin Taha for guiding this project at every stage with clarity and that priceless gift of getting things done by sharing his valuable ideas as well as share his knowledge. Without his support, and advice, this project will not be a success.

I also want to thank to all of my friends who have been in my ups and down and help me a lot to make this project a success. And last but not least, I want to give my special thanks to my parents who have given me support in every way, without them I won,t be able to finish this project.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1. Project Background

Wheel mobile robots (WMRs) have been famous since over the past three decades in research and development of mobile robot. This is due to its advantages over legged mobile robot which are wheel mobile robot having no stability problem and quicker in smooth surface compared to legged-robot. Furthermore, wheel mobile robot is more energy efficient than legged-robot and also simpler in design parts thus it is widely applied in the industry.

The development of mobile robot has benefit human in many ways. The advantage of robot that benefit human very much is that robot does not feel tired even after days of doing the same task it can do work for a very long time without rest, robot also can do repetitive task with appropriate speed and with high accuracy and precision. Mobile robot is a robot that can move freely in its environments to do its tasks. However, the limitation of mobile robot is mobile robot depending on the capacity of power source from the batteries. When the batteries have drained out of power, the robot cannot continue its current operation until the batteries is recharged

There are many researches of making cars that has no driver that able to guide itself to a destination desired by the passenger or also known as autonomous car. Autonomous car or an autonomous mobile robot is a robot that that can perform its given task at unknown environments without continuous guidance from human. This research has already been prototyped and there is possibility that this technology to be implemented in the future. One of the most popular driverless car projects is Google driverless car.

This research is a first step taken in order to make a competition to join them making one of this project applicable in the future. As the first step this robot will make a journey on the road base on line following type of robot. This robot will follow the road line as a guide. This robot is a differential drive robot which is the simplest type of robot and suitable for experiment and collecting valuable data. Moreover, this robot will be equipped with a photovoltaic cell or solar panel which able the robot to harvest solar energy to charge the batteries.

1.2. Project Objective

The objective of this project is to build an autonomous line following mobile robot that has the ability to harvest solar energy from the sun to recharge its batteries. The robot can operate 24 hours where in the day the batteries will be charged by the solar and in the night that robot will used the stored energy.

1.3. Problem Statement

The robot has to follow the line on the road as the path of the robot while harvesting solar energy from the sun. Mobile robot is a robot that can move freely in its environments to do its tasks. However, the limitation of mobile robot is the mobile robot is depending on the capacity of power source from the batteries. When the batteries have drained out of power, the robot cannot continue the current operation until the batteries is recharged. The solution to this problem is to attach a secondary power source to the mobile robot so that the batteries will be constantly being charge while the robot is doing its task. Thus, this will extend the operating duration of the mobile robot. The secondary power source will be the solar.

The problem to this project is to develop a line following mobile robot that is using solar power as the secondary power source so that the mobile robot can operate for a very long time without the batteries gone out of power.

1.4. Project Scope

The scopes of the project are as follow:

1. Making a line following robot that follow line on the road
2. Harvesting solar energy from the sun while the robot on its mission
3. Study the effective way of harvesting solar power
4. Study the mobile robot kinematics, dynamics and controller
5. To develop a mobile robot line following program using image processing method

CHAPTER 2

LITERATURE REVIEW

2.1. Introduction

A mobile robot is a machine that has the ability to move around in its area of operation on its own. An autonomous mobile robot is a mobile robot that can solve its own problem while accomplishing its goal. The navigation path planning is one of the most crucial parts of developing a mobile robot that has the ability to move on its own. Line following mobile robot can move by following the line as the path of the robot. Limitation of mobile robot is it is limited by the capacity of energy it carries in the batteries. In this chapter, we will look deeply into what are relevant to the project.

2.2. Mobile Robot – Wheeled Mobile Robot

Robotics is a revolution of industrial technology. The term of robotics means a different aperception and understanding to different people. People imagine that the robot as Optimus Prime and Bumble-Bee from the Transformer movies but in reality, robot can be in other formed of many shapes and type as long as it fulfills the characteristics of a robot.

The term “robot” was firstly used in early 1920 by a Czechoslovakian dramatist; Karel Capek in his play entitles “Rossum's universal Robots”. ‘Robota’ is the original term of robot that is widely used since it has been introduced to the public. ‘Robota’ is a Czech word which means 'salve laborer'.

Mobile Robot is a robot that has the ability to move on its own in an environment (ground motion ability). Mobile robot is not fixed to a physical surface or location. (Riza Muhida et al. 2008). Just like any other robot, it has the ability to do repeated tasks without feel tired with accuracy and precision. Mobile robot is different than the industrial robot, where the industrial robots usually have jointed arm, gripper assembly or any other specialized devices to complete its task. Mobile robot does not have any power cord connected like the industrial robot does, hence, that give the advantage to the mobile robot to move freely.

Wheeled mobile Robots have been by far the most popular locomotion mechanism in mobile robotics. It can achieve very good efficiencies and also its simple mechanical implementation (Roland Siegwart, Illah R. Nourbakhsh). The problems of wheeled robot that we should focus on are traction and stability, maneuverability and control. Table 1 below shows the types of wheels arrangements used in mobile robot.

2.3. Kinematics and Dynamics of Differential Drive Mobile Robot

Robotics is a very interesting and popular field of knowledge. There are many people have tried to build an autonomous robot for research, educational purpose or even hobby. What is interesting about autonomous mobile robot is it can move on its own and solve its own problem using the artificial intelligent that is provided to the robot. We can program the robot as we desired.

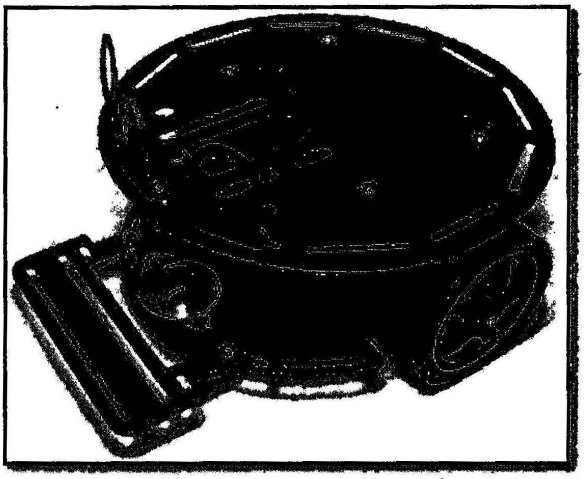


Figure 1: Example of differential drive robot

The simplest and most practical type of mobile robot is a wheel mobile robot that uses a drive mechanism known as differential drive. It is a drive mechanism which control each wheels independently. Each wheel can have a different velocity and even rotates forward or backward to perform its motion. When the robot make a turn either left or right there is a point known as ICC (*Instantaneous Center of Curvature*).

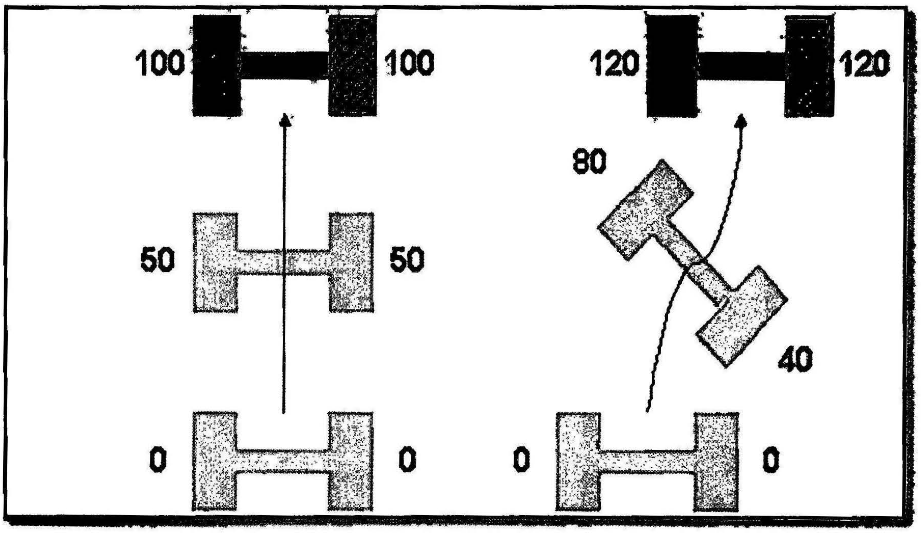


Figure 2: Example on how the differential drive robot control

Differential drive is a way of controlling the wheels where it is independently actuating the left and right wheels either to move forward or to the left or the right. The speed of each wheel is independently controlled to suit the direction of the robot. The method of controlling the robot by controlling the velocity of both wheels, we can make different trajectories as we desired for example if we want to make turn to the right then the left wheel must be faster than the right wheel. By knowing the rate of rotation of ω about the ICC we can find the left and right wheels.

$$V_R = \omega(R + L/2)$$

(1)

$$V_L = \omega(R - L/2)$$

L	= the distance between the two wheels
ω	= the rate of rotation about ICC
V_R	= Velocity of right wheel
V_L	= Velocity of left wheel
R	= Distance between center of robot and the ICC

From above equation we can solve for R and ω:

$$R = \frac{L}{2}\frac{V_L + V_R}{V_R - V_L} \qquad (2)$$

$$\omega = \frac{V_R - V_L}{L} \qquad (3)$$

There are three cases of controlling the wheels independently:

1. If $V_L = V_R$, then we have forward linear motion in a straight line. R is equal to infinite, and there is no rotation about ICC, thus ω is zero.
2. If $V_L = -V_R$, then R is equal to zero and the robot rotates in place at center of robot.
3. If $V_L = 0$, then we have rotation about the left wheel and vice versa. R will be half in this case.

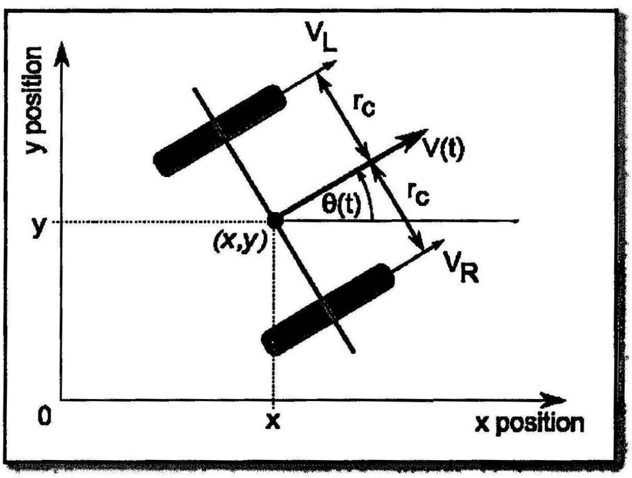


Figure 3: Example of mobile robot configuration

2.3.1. Forward Kinematics

Forward kinematics equation of a differential drive robot is used to find the position of the robot after given the velocity of each wheels. Through this we can predict the trajectories the robot makes by varying the wheels velocity. By manipulating the control parameters of V_L and V_R we can have a different orientations and position of the robot to move.

Taking Figure 3 as example we assume the robot is at position (x, y) and is making an angle Θ. using the equation above with V_L and V_R, the location of ICC can be found.

$$ICC = [x - R\sin(\theta),\ y + R\cos(\theta)] \tag{4}$$

Position of robot at time t + δt

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} \cos(\omega\delta t) & -\sin(\omega\delta t) & 0 \\ \sin(\omega\delta t) & \cos(\omega\delta t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - ICCx \\ y - ICCy \\ \theta \end{bmatrix} + \begin{bmatrix} ICC_x \\ ICCy \\ \omega\delta t \end{bmatrix} \tag{5}$$

The equation shows the robot rotating motion about ICC at a distance R and the angular velocity of ω. In other words to explain that motion, the position of ICC is translated to the origin coordinate and rotating in angular of ω about the origin and translated back to ICC.

2.3.2. Inverse Kinematics

Inverse kinematic is use to find the velocity of each wheel when given the desired position of Y_d, Y_d and Θ_d. It is a question on how we control the wheel to reach the desired position. Most articles I read, the inverse kinematics of differential drive robot is a based on path planning strategy of rotating the robot in place and moving in straight line. This is because differential drive robot has what we called non-holonomic constraints, for example it cannot move sideways directly. Thus a path planning is important in inverse kinematic, where we need to find the velocity that can get the robot to the desired position. The most common strategy is to rotate the robot heading to the goal and making the robot move in straight.

For the case of robot moving in straight line, the equation is:

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} x + V\cos(\theta)\delta t \\ y + V\sin(\theta)\,\delta t \\ \theta \end{bmatrix} \tag{6}$$

For the case of robot rotates in place, the equation is:

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} x \\ y \\ \theta + \frac{2V\delta t}{L} \end{bmatrix} \tag{7}$$

From these equations we can create navigation strategy of moving the robot to the goal, by rotating the robot in place heading to the goal and moving forward. The forward kinematics is to determine the position and orientation of the mobile robot, given the angular velocity of each wheel. From the Fig. 1, the forward kinematics is as follows:

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix} = \begin{bmatrix} -\frac{r\sin(\theta)}{2} & -\frac{r\sin(\theta)}{2} \\ \frac{r\cos(\theta)}{2} & \frac{r\cos(\theta)}{2} \\ \frac{r}{b} & -\frac{r}{b} \end{bmatrix} \begin{bmatrix}\omega_R\\ \omega_L\end{bmatrix} \qquad (8)$$

From the above equation we can find the inverse kinematics of the mobile robot by finding the angular velocity of each wheel.

$$\begin{bmatrix}\omega_R\\ \omega_L\end{bmatrix} = \begin{bmatrix} -\frac{2}{r\sin(\theta)} & -\frac{1}{r\cos(\theta)} & \frac{b}{2r} \\ -\frac{2}{r\sin(\theta)} & -\frac{1}{r\cos(\theta)} & -\frac{b}{2r} \end{bmatrix} \begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix} \qquad (9)$$

2.3.3. Dynamics

The dynamic motion equation can be derived using Euler- Lagrange formulation (Modelling of Mobile robot dynamics,Edouard Ivanjk, Toni Petrini, Ivan Petrovi):

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = Q_i, \qquad (10)$$

L = the kinetic difference, T

U = potential energy

qi = generalized coordinate

Qi = generalized force acts on the mechanical system

This dynamic of mobile robot is assume that the robot only move on a plane surface, thus the potential energy of the robot is equal to zero (U = 0). The total kinetic energy of the mobile robot is:

$$T = T_t + T_r + T_{rwr} \tag{11}$$

Where,

Tt = kinetic energy of mobile robot translation

Tr = kinetic energy of mobile robot rotation

Trwr = kinetic energy of rotation of wheels and rotor

$$T_t = \frac{1}{2} m V_c^{\ 2} = \frac{1}{2} m(x_c^2 + y_c^2) \tag{12}$$

$$T_r = \frac{1}{2} IA\dot{\theta}^2 \tag{13}$$

$$T_{rwr} = \frac{1}{2} I_o \dot{\theta}_R^{\ 2} + \frac{1}{2} I_o \dot{\theta}_L^{\ 2} \tag{14}$$

Where,

m = mass of the mobile robot (kg)

Vc = Velocity of mobile robot

IA = moment of inertia of mobile robot

Io = moment of inertia of wheel and rotor

2.4. Line following robot

For mobile robots, the ability to navigate in its environment is very important. Mobile robot navigation based on lines, landmarks and sign have been widely implemented around the world (A.H. Ismail, 2009). Line following robot can be useful and practical to do some task. There are many technique of image processing to make a line following robot. Most of the line following program will use black and white to differentiate the lines and the background.

Mobile robot navigation based on lines, landmarks and signs have been widely implemented around the globe. The main goal of these works is to develop a mobile robot with the capability of navigating through a predefined path or towards a set destination using a line, landmark or sign as a point of reference. The image is treated as a Cartesian plane and points from the image are taken to be coordinates on the Cartesian and tangent calculations on those coordinates would produce the desired trajectory of the robot.

Most of the literature reviewed here deal with images that are monotone or black and white as to reduce computational complexity. Research however, uses color signs to assist with the robot navigation while simultaneously using an adaptive sub-sampling method to avoid having to process each pixel in the image individually.

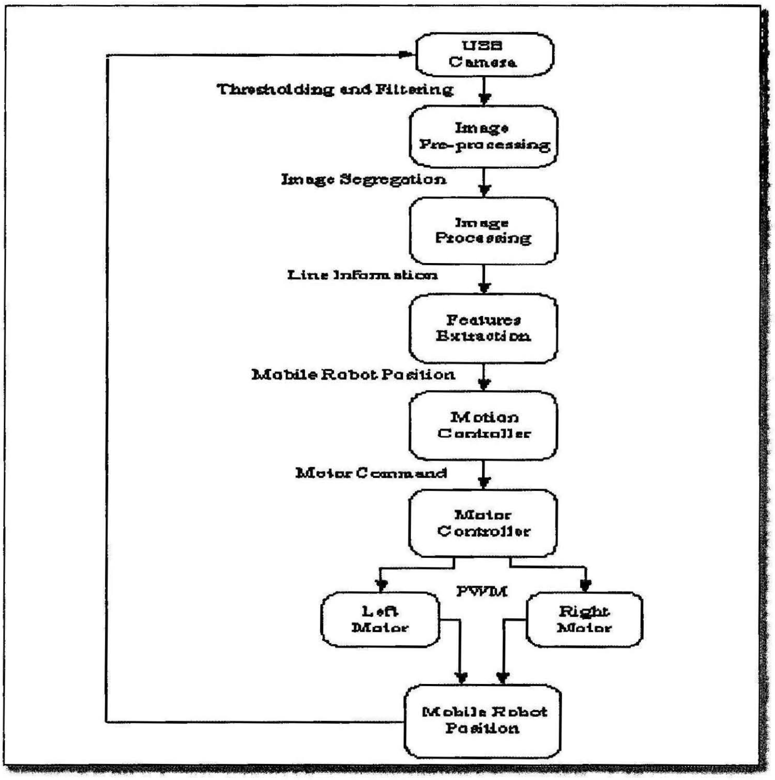


Figure 4: Block Diagram of line following Mobile Robot proposed by (A.H. Ismail, 2009)

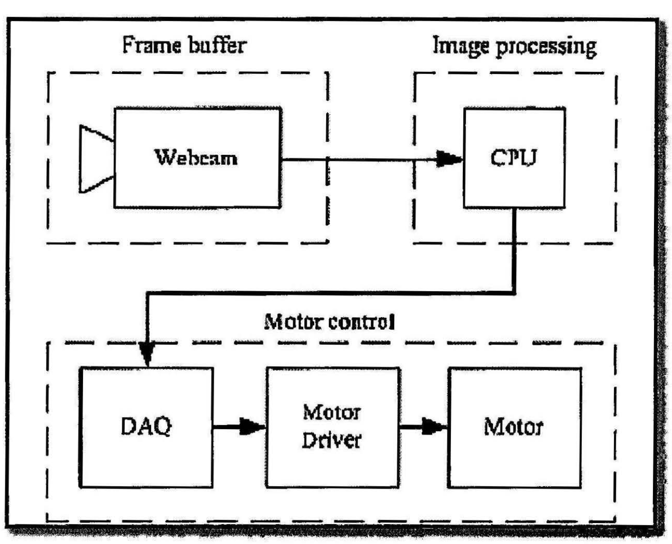


Figure 5: Mobile Robot configuration System by (A.H. Ismail, 2009)

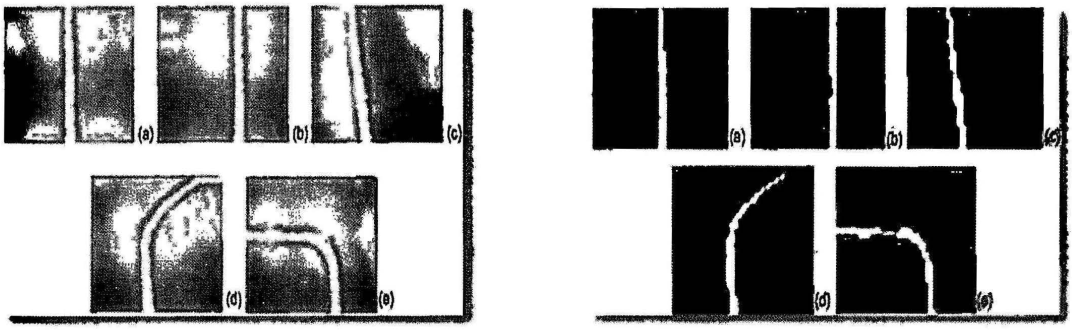


Figure 6: On the left is the typical guided lines viewed by the robot's camera, on the right is the segmented lines from the background.

In the sample shown in Figure 6 it can be seen that the image captured is of a straight white line. By segregating the image into blocks, shown in Figure 6, it is identified that all of the bright pixels, which indicates the line, are concentrated in the blocks which are central to the image.

$$W = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

This matrix is obtained by calculating the number of white pixels in each block and checking if that number exceeds 20% of the total number of pixels in that block. Any block which has less than that amount of white pixels is considered to have an insignificant amount of white pixels within it and is considered to have a value of 0. Blocks with more than that amount of white pixels are considered to have a value of 1.

$$x=\left(5-\sum_{i=1}^{5}w_{i3}\right)\left[0.1\left(\left(\sum_{i=1}^{5}\sum_{j=1}^{2}w_{ij}\right)-\left(\sum_{i=1}^{5}\sum_{j=4}^{5}w_{ij}\right)\right)\right]$$

The value x is useful to be used for mobile robot control system.

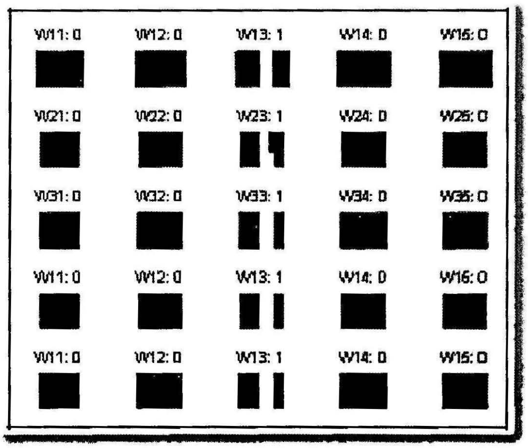


Figure 7: Sample of segregated image

2.5. Solar charger MPPT

Solar power harvesting is a matured technology of harvesting free and renewable energy. Solar energy harvesting through photo-voltaic conversion provides the highest power density of harvesting technology (Aman Kansal et al.2009). This make it's a reasonable and suitable for a mobile robot application due to it is mobility and can provide enough energy to the robot by using right approached of energy management.

POWER DENSITIES OF HARVESTING TECHNOLOGIES

Harvesting technology	Power density
Solar cells (outdoors at noon)	$15mW/cm^3$
Piezoelectric (shoe inserts)	$330\mu W/cm^3$
Vibration (small microwave oven)	$116\mu W/cm^3$
Thermoelectric (10°C gradient)	$40\mu W/cm^3$
Acoustic noise (100dB)	$960nW/cm^3$

Table 1: Power density comparison

The design of the solar energy harvesting module does have some tradeoffs due to interaction of several factors such as the characteristic of the solar cells, chemistry and capacity of the batteries used, power supply requirements and power management features of the embedded system (Vijay et al. 2009).

Energy harnessed from the photo-voltaic cells does not have a stable current output thus; it is not efficient to be used to power some systems. The solution to this problem is by having a Maximum Power Point Tracking (MPPT) solar charge controller. By operating at the solar panel's maximum power point (MPP) and by intelligently drawing the power from the panel, energy can be successfully harnessed to power a desired system. (Rotar Dan, 2011).

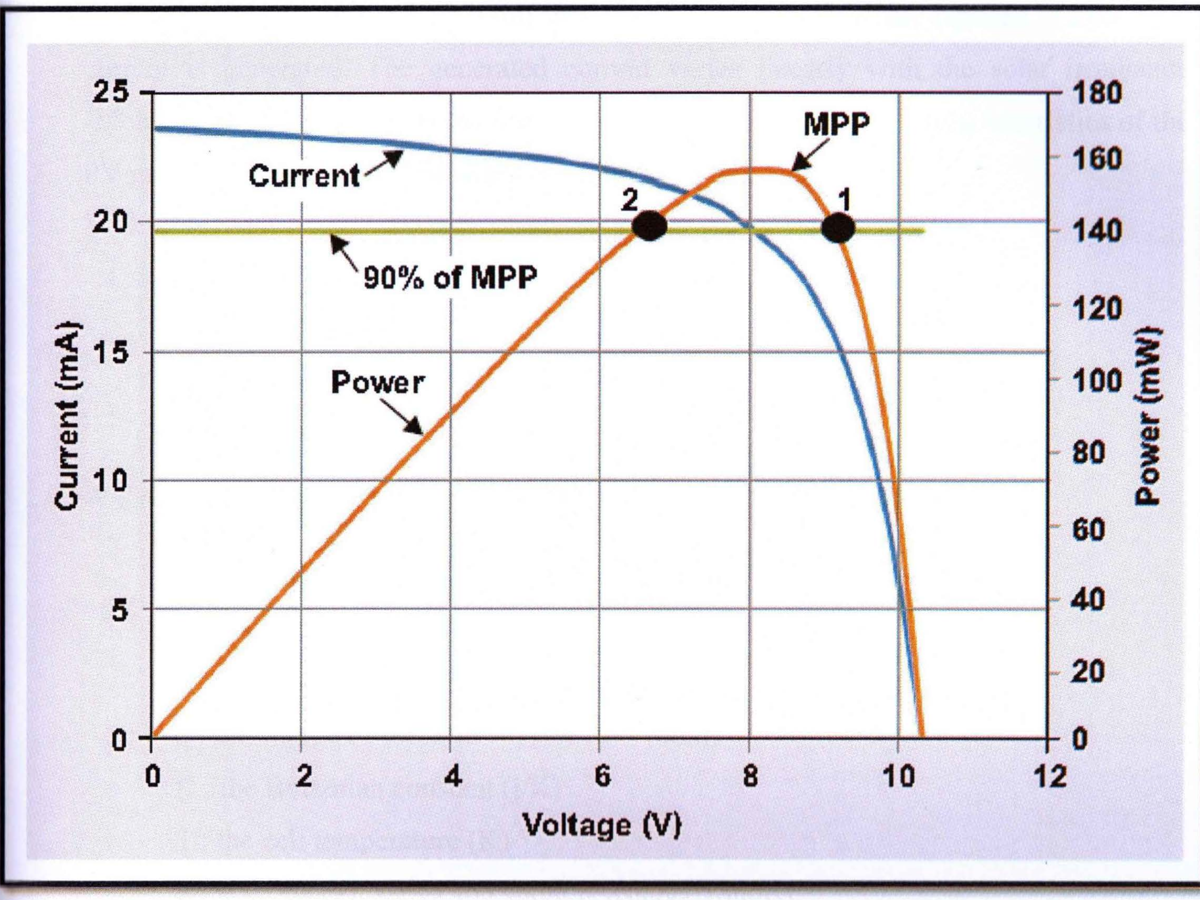


Figure 8: illustrate the particular solar panel's output current and output power versus its output voltage.

There are many researches to improve the efficiency of the MPPT solar charge controller, one of the approaches is through improving the existing Maximum Power Point Tracking (MPPT) Algorithm. There are my people tried to track MPP mathematically but has never get an accurate as tracking it dynamically. The most well-known and implemented dynamic MPPT algorithm is perturb and observe (P&O). P&O is simply perturbs the current by delta Amps and then observes the effect. If the power goes up, the direction of delta was towards the MPP, if the power goes down the direction was away from the MPP (Mohamed Azab, 2009).

A solar cell basically is a p-n semiconductor junction. When exposed to light, a dc current is generated. The generated current varies linearly with the solar irradiance (Mohamed Azab, 2009). The basic equation that describes the (I-V) characteristics of the PV model is given by the following equation:

$$I = I_L - I_o\left(e^{\frac{q(V + IR_s)}{kT}} - 1\right) - \frac{V + IR_s}{R_{sh}} \qquad (15)$$

Where:

I : the cell current (A)

I_L : the light generated current (A)

I_o : the diode saturation current

q : the charge of electron = 1.6×10^{-19} (coul)

K : the Boltzman constant (j/K)

T : the cell temperature (K)

Rs, R_{sh} : the cell series and shunt resistance (ohms)

V : the cell output voltage (V)

CHAPTER 3

METHODOLOGY

3.1. Robot Construction

The construction of the robot is divided in two parts electronics and mechanical. The electronics part consist the MPPT charger to collect the energy from sunlight and convert it to a useable electricity to charge the batteries. The mechanical part has cube shape aluminum as the chassis, two hub motor 48V, and squared L shape steel to hold the hub motor in place. Most of the material is provided and some material is not available in the lab so I have to order it from the supplier.

Figure 9: Before the wheel is attached

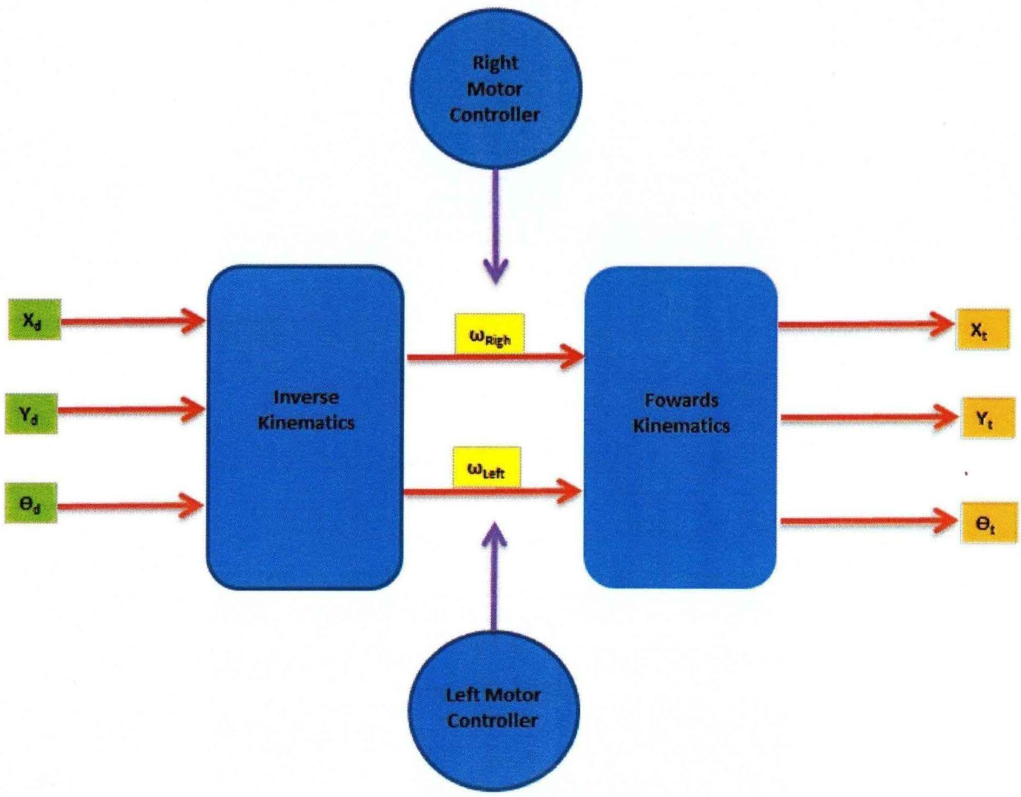
Figure 10: After Wheels is attached

After the attachment of the wheel the robot motions are tested by moving both wheel and the robot motion is observed. The observation of the robot motion is the robot move in a curve but good enough to make a straight motion with a little tune later and the robot can move without encoder to read wheel speed but it is required in future to perfect the robot motion and path planning.

3.2. Kinematics simulation and Dynamics Modeling

The kinematics and dynamics modeling is important as the data that we can learn about the robot and prepare the robot before program and run it in real situation. From the data we know what to adjust when we test it on the road for real simulation.

The kinematics of the mobile robot was easy to do but the inverse kinematics was quite tricky. I refer to many articles and website to learn how to derive it. Most of the journals show that the solution is to create a path planning where robot rotates in place facing the goal and move in straight motion. Dynamic of the mobile robot was harder to understand so I just find a related article/journals that is can use the dynamics of the robot and modify.



From the journal I read, this is what I came out with. The use of inverse kinematics is to get the speed needed to get to the position and forwards kinematics is to get the position when we have the speed. From that there are motor controllers that get the input from the inverse kinematics and give out output to control the wheel at desired speed.

3.3. Line following program(RoboRealm and Arduino)

Line following is common model of robot but using camera to detect line on the road and give command to the robot is quite challenging. Getting the line on the road is way difficult than detecting line on the controlled floor or a taped line. Using RoboRealm to detect the line on the road is one of the easiest software to learn and I can use it with Arduino to process the image processed to detect line and send it to Arduino with serial communication to move the wheels.

The kinematics equations used are:

- $x = \frac{(VR+VL)L}{(VR-VL)2} \sin \frac{(VR-VL)t}{L}$ (16)
- $y = \frac{(VR+VL)L}{(VR-VL)2} \cos \left[\frac{(VR-VL)t}{L} - \frac{(VR+VL)L}{(VR-VL)2}\right]$ (17)
- $\theta = cos \frac{x}{\sqrt{(x2+y2)}}$ (18)

The equation used to get the inverse kinematics has two methods by create a path planning of rotates towards goal and move in straight line and by knowing the robot set speed in x and y direction, we can find the angular velocity of the wheel

Equations used to find the inverse kinematics is:

- Move Straight

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} x + v\,cos(\theta)\delta t \\ y + v\,sin(\theta)\delta t \\ \theta \end{bmatrix} \tag{19}$$

- Rotates in place

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} x \\ y \\ \theta + 2v\delta t/l \end{bmatrix} \tag{20}$$

The other method is by knowing the velocity of x, y and Θ

- $$\begin{bmatrix} \omega R \\ \omega L \end{bmatrix} = \begin{bmatrix} -\frac{2}{r\sin(\theta)} & -\frac{1}{r\cos(\theta)} & \frac{b}{2r} \\ -\frac{2}{r\sin(\theta)} & -\frac{1}{r\cos(\theta)} & -\frac{b}{2r} \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \tag{21}$$

3.4. Assemble the MPPT to the batteries to charge it under the sunlight

The charging will not be fast but the charge stored in the batteries can hold long enough for the robot to operate its task. The MPPT charging with the low current and the maximum output from the solar panel is around 36 volt.

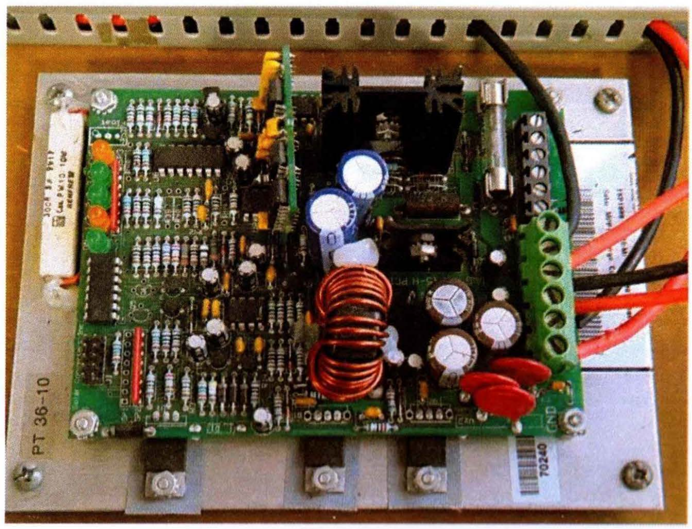


Figure 11: Maximum Power Point Tracking (MPPT)

Figure 12: Solar Panel

CHAPTER 4

RESULTS AND DISCUSSION

4.1. Introduction

This chapter shows all the results obtained from this study. The results are represented in graphs, table of result and figures. The kinematics and dynamics are obtained from several journals taken as reference. The kinematics simulation is simulated using Excel. PID simulation is created using MatLab/Simulink. Image processing to detect the line is processed using RoboRealm and Arduino is used to control the wheels after receiving processed data from RoboRealm. The ability of robot to recharging the batteries comes from the MPPT and Solar Panel attached to the robot.

4.2. Kinematics simulation

The kinematics equation used to create simulation for this robot is:

$$x = \frac{(V_R+V_L)L}{(V_R-V_L)2} \sin\left[\frac{(V_R-V_L)t}{L}\right] \quad (22)$$

$$y = \frac{(V_R+V_L)L}{(V_R-V_L)2} \cos\left[\frac{(V_R-V_L)t}{L} - \frac{(V_R+V_L)L}{(V_R-V_L)2}\right] \quad (23)$$

$$\theta = \cos\frac{x}{\sqrt{x^2+y^2}} \quad (24)$$

Where;

L = distance between two wheels

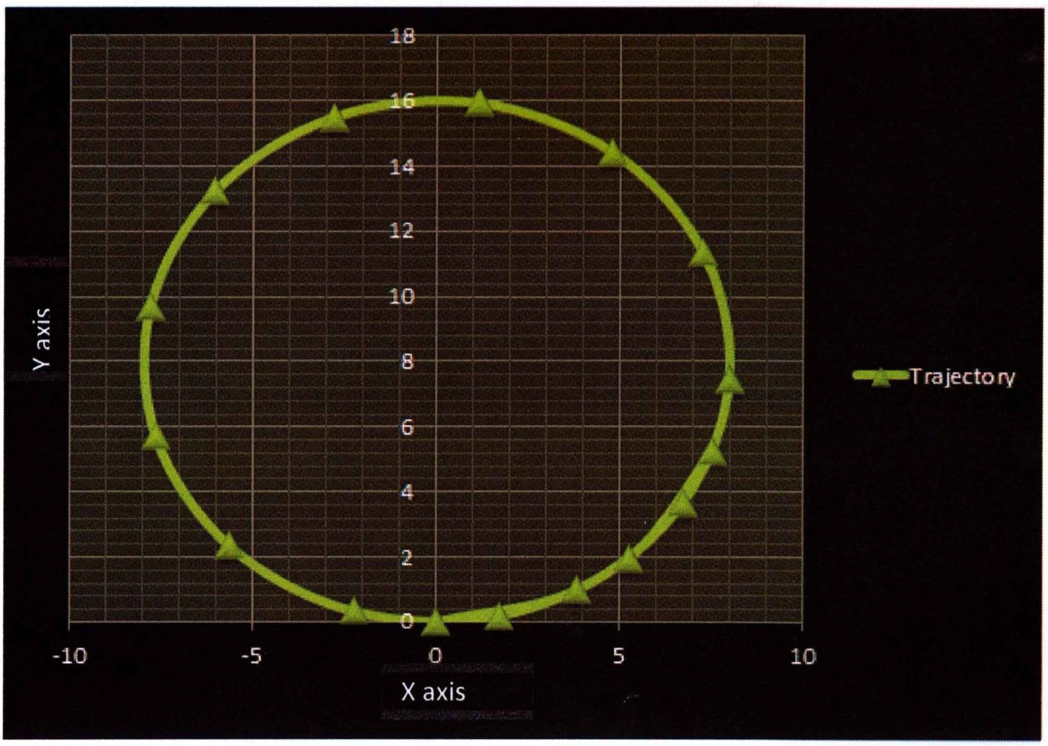
V_R = velocity of right wheel

V_L = velocity of left wheel

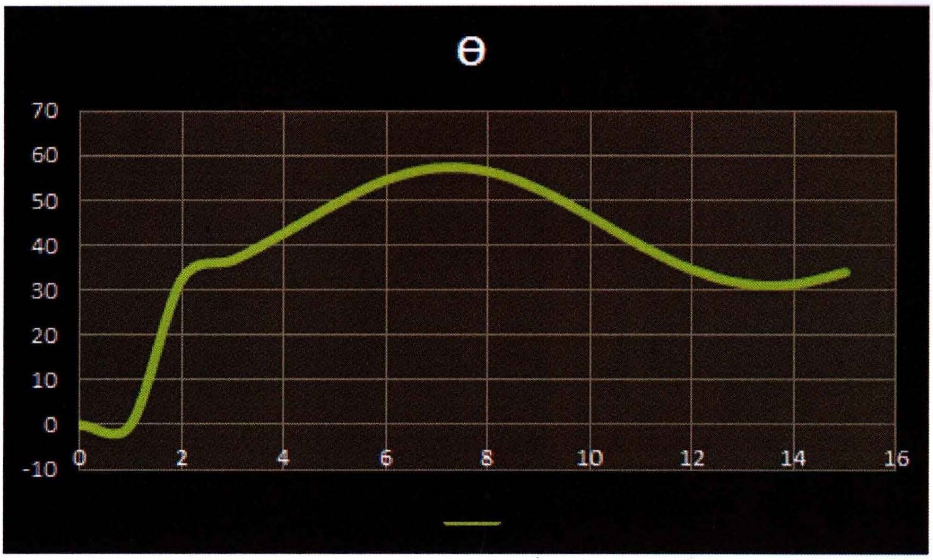
By using this equation the trajectory of the robot can be predicted and the trajectory can be planning to suit the operation. Therefore, we have the knowledge how the robot will move with the velocity affected on the left and right wheels.

	Forward Kinematics							
				Input		Trajectory		
Time (s)	xi	yi	θi	vl	vr	x	y	θ
0	0	0	0	5	3	0	0	0
1	0	0	0	5	3	3.835404	0.97934	32.44066
2	3.835404	0.97934	32.44066	5	3	6.731768	3.677582	36.61272
3	6.731768	3.677582	36.61272	5	3	7.97996	7.434102	42.63076
4	7.97996	7.434102	42.63076	5	3	7.274379	11.32917	49.13418
5	7.274379	11.32917	49.13418	5	3	4.787777	14.40915	54.47089
6	4.787777	14.40915	54.47089	5	3	1.12896	15.91994	57.15249
7	1.12896	15.91994	57.15249	5	3	-2.80627	15.49165	56.388
8	-2.80627	15.49165	56.388	5	3	-6.05442	13.22915	52.40578
9	-6.05442	13.22915	52.40578	5	3	-7.82024	9.686366	46.35814
10	-7.82024	9.686366	46.35814	5	3	-7.67139	5.730703	39.87132
11	-7.67139	5.730703	39.87132	5	3	-5.64432	2.330642	34.5145
12	-5.64432	2.330642	34.5145	5	3	-2.23532	0.318638	31.43797
13	-2.23532	0.318638	31.43797	5	3	1.72096	0.187299	31.23953
14	1.72096	0.187299	31.23953	5	3	5.255893	1.968782	33.9561
15	5.255893	1.968782	33.9561	5	3	7.504	5.226917	39.06496

Table 2: shows the trajectory of the robot in tabulate form



Graph 1: shows the trajectory of the robot



Graph 2: Shows the angle the robot makes during the motion

4.3. PID simulation of mobile robot

The figure below is to simulate the control of each wheel of robot with PID controller. The figure show left and right wheel control. Left and right wheel is set to different desired velocity.

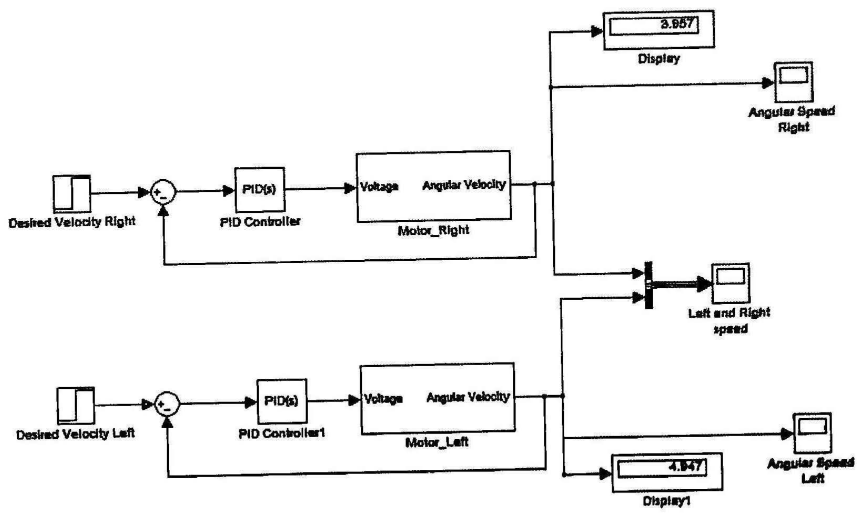


Figure 13: PID controller to control left and right wheels

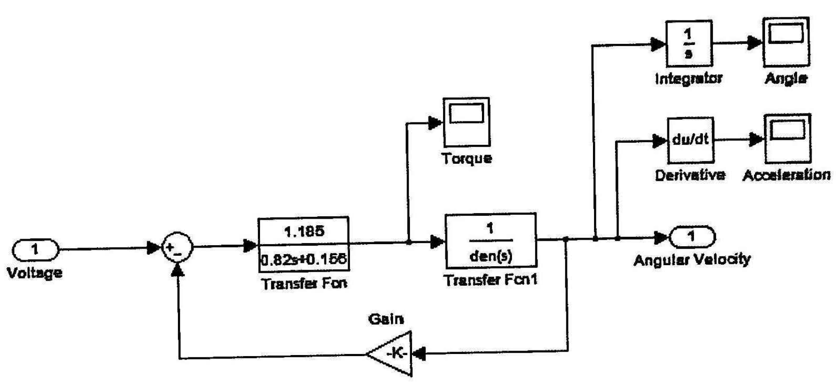


Figure 14: Motor block diagram

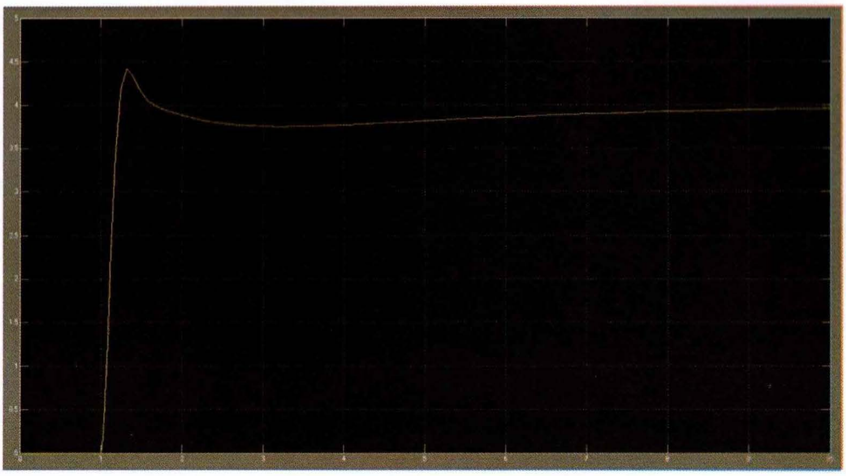


Figure 15: Angular velocity responds to achieved desired velocity

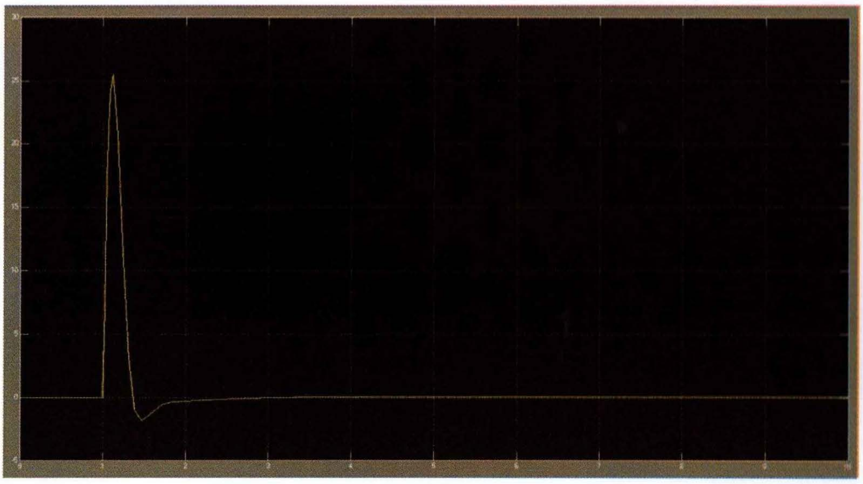


Figure 16: Acceleration vs Time

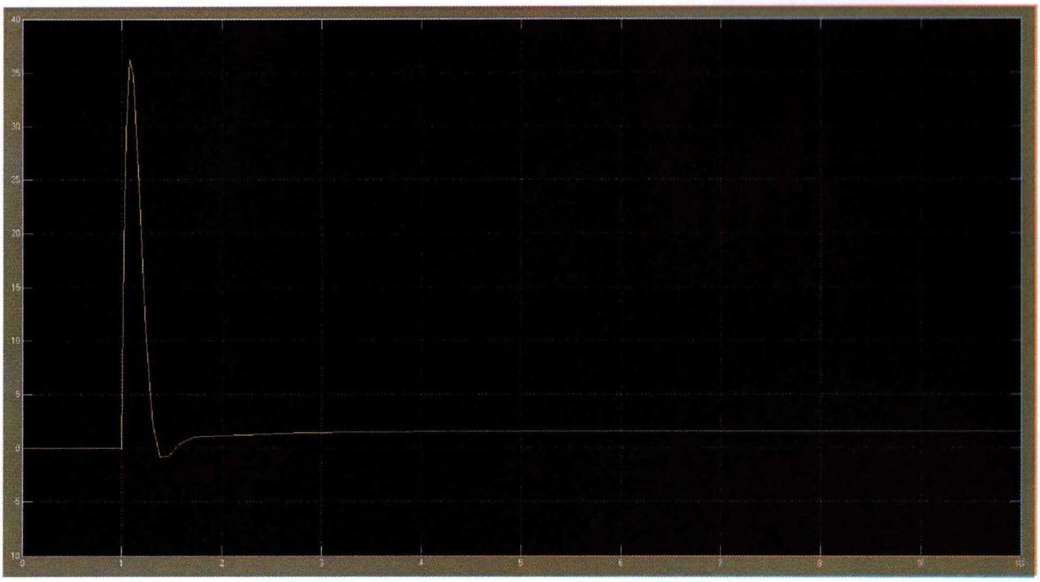


Figure 17: Torque vs Time

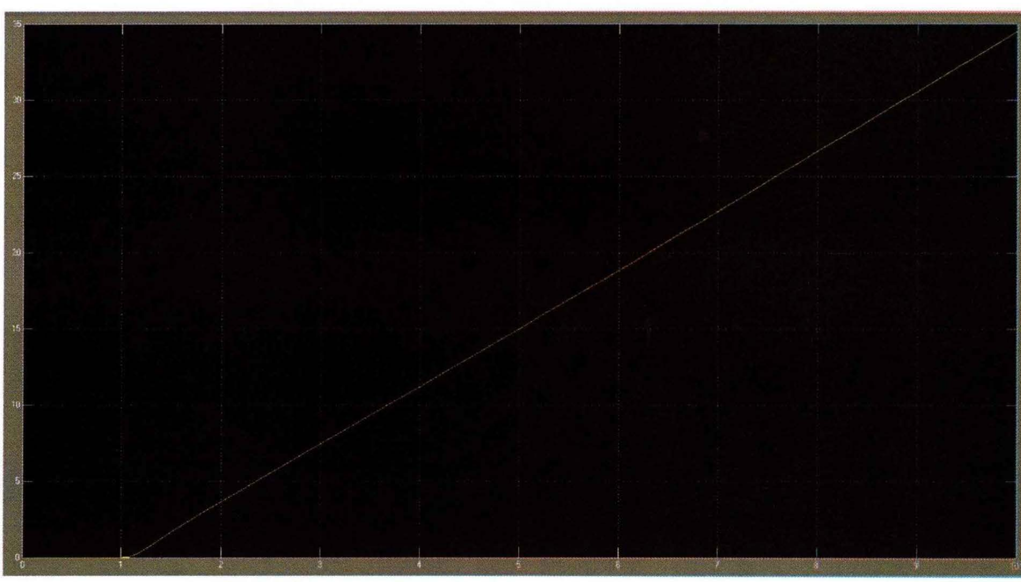


Figure 18: Angel vs Time

The graph shown were the results obtained from the PID controller of the wheel controller to achieve the desired input. Figure A shows that PID tried to quickly achieve the desired velocity with an acceptable overshoot. The respond of the controller is very fast of 0.2222 sec. Figure B shows the acceleration of the wheel make to achieve the desired velocity and the respond is very fast as shown same as the Figure E torque of the wheel is very high thus having no problem to move the robot weighted about 100kg. Figure C show the angle the motor make versus time.

4.4. Image processing to detect line on the road

The image processing of this robot is created to follow the line on the road. The line on the road is a trail like line. Therefore, the approached I use is to track one to one the line in sequence to guide the robot through the line. The line is detected through following method using RoboRealm:

1. First from the original image, the image is processed to the grey scale to get the black and white effect more visible. Thus easier to trace the lines.

Figure 19: Original

Figure 20: Grey scale

2. Then, the image is given blurred effect to remove the noise in the image and the blurred image is subtracted to the original image to get the detail back.
3. Flood effect is added to the current image to flood the background of the image so that the image we want to detect is pop out.
4. Blob effect is applied to the image to make the line detectable to find the center of gravity of the line which it will be used to calculate the distance between center of image and line.
5. VBScript is program to get the center of gravity value and using serial communication module in the RoboRealm to send the data to the Arduino.

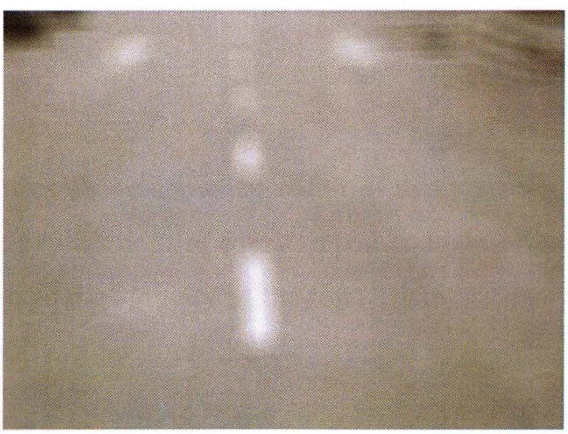


Figure 21: Blurred

Figure 22: Original image – the blurred

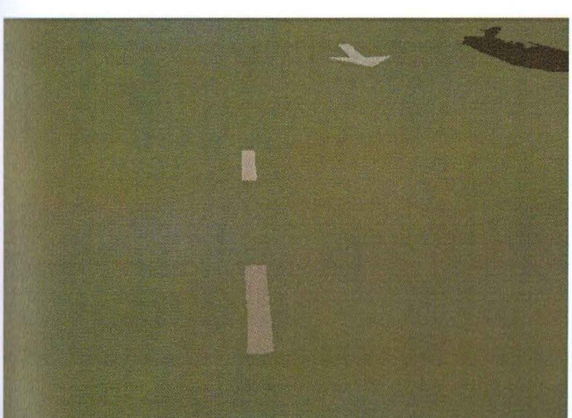


Figure 23: Flood image

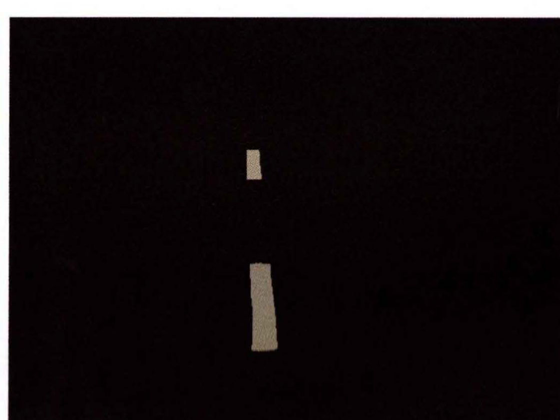


Figure 24: Blob filter

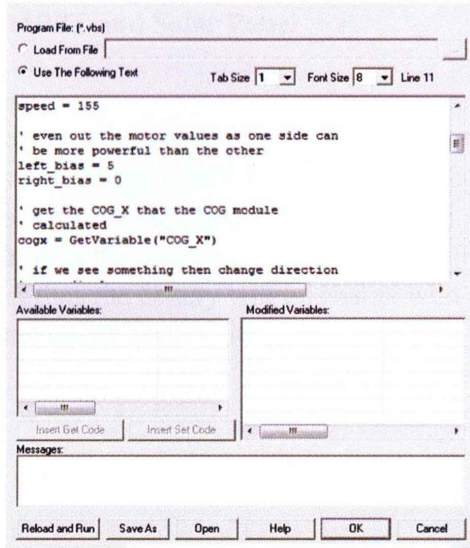


Figure 25: VBScript

Figures above show the result of every step of image processing to detect the line on the road. The explanations of every step already described above. The image processing above is to obtain the center of gravity of the line. There are many other steps that can detect the line but I use this to suit the environment.

4.5. Charging from the MPPT and Solar Panel

The MPPT is used to give out a stable and usable current that is able to recharge the batteries. This MPPT is set to charge a 48V batteries connected in series or a single battery. The charging time varies with the supplied current from the solar panel if the sun supply stable and continuous sun rays the charging will be better. This MPPT is set to charge the battery at optimum current thus the battery will be charge with a quality current and result in longer discharging time and longer battery life.

Figure 26: MPPT used to charge

The charge left in the batteries will be indicated by the LEDs on the MPPT board.

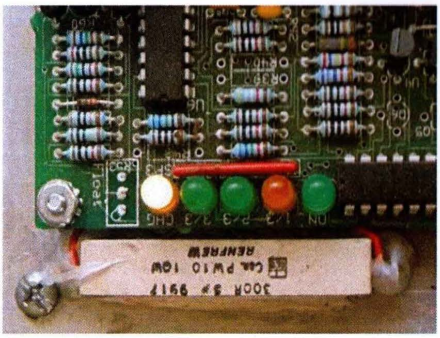


Figure 27: Charging with a very low

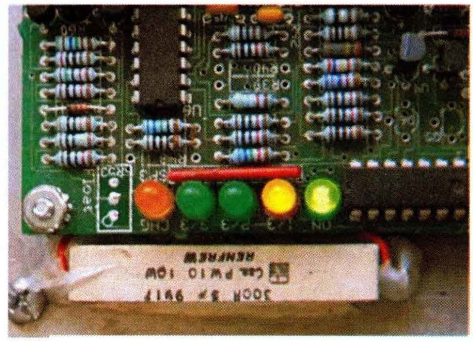


Figure 28: 1/3 of charge level in

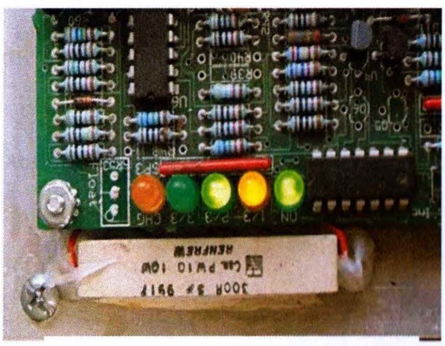


Figure 29: 2/3 of charge level

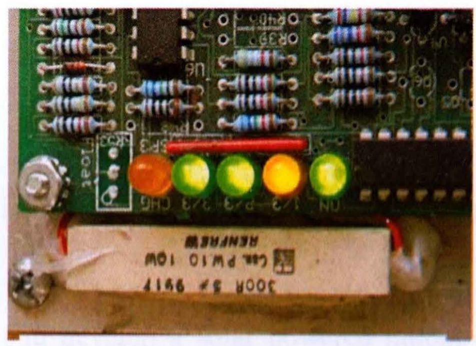


Figure 30: The batteries is fully

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1. Conclusion

In this research, the kinematics simulation and dynamics of the mobile robot is successfully created as for simulation of the motion of the mobile robot and also to calculate energy used to move the robot. This robot used PID controller and the PID controller of this robot is successfully simulated and used to have a smooth motion of the mobile robot while following the line. The robot use RoboRealm to process the image to detect line and successfully detect the line on the road. Last but not least the MPPT charger and solar panel attached to the mobile robot successfully recharging the batteries.

Thus the objective of this project to develop a solar powered mobile robot is achieved by following the methodologies proposed in this research.

5.2. Recommendation

From the results that have been obtained from the previous chapter, there are some recommendations to future works that can improve this project to another level:

1. To make this robot a lane following robot as it will be more car like operation as for now the robot only follow line on the road.
2. To have a sun tracking solar panel to track the sun and this will increase efficiency of harvesting energy from the sun as the sun is not always perpendicular to the solar panel.
3. To have obstacles avoiding ability thus more sensors will have to be added to the robot. This will help to avoid any collision that can harm the robot or even harm the environment.

REFERENCES

1. M. Egerstedt, X. Hu, and A. Stotsky, Control of Mobile Robot Platforms Using a Virtual vehicle Approach, IEEE Ttansaction on Automatics Control Vol. 46. No.11, November 2001.
2. D.Sivaraj, A.Kandswamy, V.RajaSekar, Design and Development of Vision Based AVCS, European Journal of Scientific Research, ISSN 1450-216X Vol.67 No.2 (2012), pp. 309-323, EUROJournals Publishing, Inc.2012.
3. Farhan A. Salem, Dynamic and Kinematic Models and Control for Differential Drive Mobile, International Journal of Current Engineering and Technology,Vol.3 No.2(June 2013)
4. Jose de Jesus Rubio, Victor Aquino, Maricela Figuroa, Inverse Kinematics of a Mobile robot, Received: 8 September 2011 / Accepted: 11 January 2012 / Published online: 3 February 2012 Springer Verlag London Limited 2012
5. Maung Than Zaw, Kinematics and Dynamic Analysis of Mobile Robot, National University of Singapore 2003.
6. Julius Maximilian Kinematics of a car-like mobile robot, Mobile robot basic – differential Drive,Universität Würzburg Institute of Informatics VII, Updated 22 March 2003
7. Edouard Ivanjko, Tini Petrinic, Ivan Petrovic, Modelling of mobile robot Dynamic, edouard.ivanjko@fer.hr
8. Jasmin Velagic, Bakir Lacevic and Nedim Osmic, Nonlinear Motion Control of Mobile Robot Dynamics Model, University of Sarajevo.
9. Sourabh Battacharya, Rafael Murrieta, Seth Hutchinson, Path Planning for a Differential Drive Robot: Minimal Length Path-A Geometric Approach, Intelligent Robots and Systems September 28 ~ October 2,2004, Sandal, Japan
10. Dongyoung Chwa,. Tracking Control of Differential Drive Wheeled Mobile Robot Using a Back Stepping-Like Feedback Linearization.
11. (SWR 26 Sept 2011) PID Control : A Brief Introduciton and Guide, Using Arduino,

APPENDICES

Arduino Program

```
/////////////////////////////////////////////////////////////////////
Declaration of
/////////////////////////////////////////////////////////////////////
int Left_motor = 12;
int Right_motor = 8;
int COG_X;
int Integral = 10;
int kP = 11.4361931207227;
int kI = 3.01749484734774;
int kD = 8.81556874700663;
int Last;
//int SetPt = 320;
//int Actual = COG_X;


void setup()
{
   Serial.begin(9600);
   pinMode(Right_motor, OUTPUT); //RIGHT
   pinMode(Left_motor, OUTPUT); //LEFT
   pinMode(3, OUTPUT);
   pinMode(5,OUTPUT);

}
```

```
 void loop()
{
  if (Serial.available() > 0){
    COG_X = Serial.parseInt();
//////////////////////////////////////////////////////////////////////////////////
PID Control
//////////////////////////////////////////////////////////////////////////////////
  int Error = 320 - COG_X;
  //int P = Error*kP;
  //int I = Integral*kI;
  //int D = (Last - Actual)*kD;

  //int Drive = P+I+D; //Drive as motor Speed

  //Drive = Error;
  //if (abs(Drive > 255)){
  // Drive = 255;
 // } else {
  // Drive = Drive;
  //}
//////////////////////////////////////////////////////////

    //if (255 < COG_X < 385){
    //  analogWrite(3, Drive);
    //  analogWrite(5, Drive);
    //  digitalWrite(Right_motor, LOW);
    //  digitalWrite(Left_motor, LOW);
    //  Serial.println("FOWARD");
    //  Serial.println(Drive);
    //}
    if (COG_X >= 325){
     //analogWrite(Right_Speed, 0);
      analogWrite(5, abs(Error));
      digitalWrite(Right_motor, HIGH);
      digitalWrite(Left_motor, LOW);
      Serial.println("KANAN");
      Serial.println(Error);
    }
```

```
    if(COG_X <= 315) {
      analogWrite(3, abs(Error));
     //analogWrite(Left_Speed, 0);
      digitalWrite(Left_motor, HIGH);
      digitalWrite(Right_motor, LOW);
      Serial.println("KIRI");
      Serial.println(Error);
    }
    if( Error == 0)
     {
       analogWrite(3, abs(50));
       analogWrite(5, abs(50));
      digitalWrite(Left_motor, HIGH);
      digitalWrite(Right_motor, HIGH);
      Serial.println("STOP");
      Serial.println(Error);
    }

  }
}
```

RoboRealm VBScript

```
COG_X = GetVariable ("COG_X")

SetVariable "COG_X", COG_X
```

Circuit Connection with Arduino and Motor Controller