ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 3, September 2014

Robust Method for Robotic Guidance Using Line Following Method

Dr. Mohammed Sahib, Dr. Loay Kadom, Dhuha Abdullah
Al-Nahrain University, Baghdad, Iraq

Abstract*— The motivation we address in current research is to establish a robotic control system to follow a line was drawn on the ground. Thus, the design of such system required to be consisted of two parts; hardware and software. The hardware part is stand for the robot, while the software part is playing the role of controller. The robot consists of robot body, which is a toy car including on its board a visual sensor to see the planned path. Other robot's parts are just communication and control devices. On the other side, the controller is software was established to receive captured images from the camera carried on the robot board, and then these images are processed to issue the guidance decisions. The adopted image processing makes the resulted decisions to be more accurate. Later, the controller converts the decisions into commands, which are sent to the robot as wireless signals. The robot receives and implements these commands. Frequently, the robot senses its environment and sends data to the controller and the controller issue its commands for the robot, and so the control process is. Frequent experiments and analysis showed that the behavior of the robot was acceptable and it was enjoying with the ability to follow the planned path.

Index Terms—**Mobile Robot, Line Following, Robotic vision, Visual Sensor.**

I. INTRODUCTION

Robotics is an interdisciplinary field involving diverse disciplines such as physics, mechanical design, statics and dynamics, electronics, control theory, sensors, vision; signal processing, computer programming, artificial intelligence (AI), and manufacturing [1]. Mobile robots have the capability to move around in their environment and are not fixed to one physical location [2]. It is a mechanical device capable of moving in an environment with a certain degree of autonomy. Autonomous navigation is associated to the availability of external sensors that capture information of the environment through visual images or distance or proximity measurements [27]. There are a great deal of focus was granted to line following for mobile robot, the most interested literature is given by Gini G. and Marchi A. in 2002 how developed an idea of a simple autonomous agent relying only on vision information using single camera for integrated mobile robot navigation. The results indicate that the use of simple reactive strategies reduces the risk of failures [3]. Ono et al. in 2004 had been proposed a framework for designing and implementing a mobile robot control program for navigation that is easily expandable and portable to other robotic platforms. The robot and the control system were presented and analyzed in the experiments. The robot's performance demonstrates the feasibility of a multi-agent approach [4]. Ghanavati M. and Ghanbarzadeh A. in 2010 present project guidance and control of underwater robot (including three engines and attached propeller) has been investigated by Fuzzy control. This robot may be used in sea or swimming pool environment for finding goal point and locate in desired direction [11]. In this paper, we used a series of image processing; each process prepares the data to be more useful for the next one. The collision work of them gives more accurate results, which make the guidance more robust.

II. PROPOSED ROBOTIC SYSTEM

The proposed robotic system consists of two separated parts: hardware and software as shown in Figur1. The hardware part represent the designed robot, which is a set of robot contents that includes car body and other devices related to the control equipments, and signal transmission or receiving. The software part is the control system, which is the controller, is the line following routine. The line following routine is responsible on routing the robot according to the detected path, and then making the suitable decisions that issued to the robot. It is important to mention that both the controller and robot are operated according to closed loop control system technique. The input data of line following routine in the controller are received from the visual sensor carried in the robot. Such data help the controller to detect the intended path of robot motion and determine the decisions needed for correct and safe navigation through the environment.

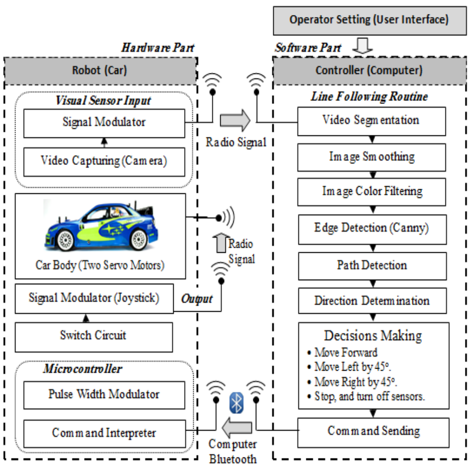


Fig 1. Block diagram of the proposed closed loop Robotic system design

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 3, September 2014

III. HARDWARE PART

The hardware part is stand for mobile robot. The robot is responsible for sensing its current situation and obeys to the controller orders. Robot sensing help the controller to predict the desired movement of the mobile robot toward the destination point. In addition to the robot body, the robot consists of some important parts; each of which is used to perform a specific function. These parts are discussed and explained in the following sections:

(i). Robot Body

It is useful to make the robot body is a remote controlled game car with wireless remote controller (joystick). This is because it is available and can be adaptive to be suited with the proposed system. The remote controller contain electric circuit contain control points for each function expected by the remote controller buttons these control points are connected to the positive pole of the power which supplies form batteries by pressing the buttons the electrical circuit is closed the current flows to the transmitter. A transmitter is an electronic circuit which transforms electric power from a battery into a radio frequency alternating current by means an electronic oscillator circuit that generate the radio frequency signal. This make to reverse direction millions of times per second. The modulator circuit will add the information to be transmitted to the carrier wave produced by the oscillator. This is done by varying some aspect of the carrier wave. In an amplitude modulation (AM) transmitter the amplitude of the carrier wave is varied in proportion to the signal. Then, the generated radio waves are applied to the antenna; the antenna radiates radio waves when excited by alternating current . The problem in such case is the difference of operating voltages found between the joystick and microcontroller, the joystick operates with a voltage of 3v whereas the microcontroller operates with 9v. To overcome such problem, the joystick should be adapted to be connected with the microcontroller in a case that makes both of them operate with its required voltage. Therefore, a switch electronic circuit was suggested to be assigned with the joystick in between the joystick and microcontroller. Such that, the signal comes from the microcontroller is used to operate just the switch circuit, and then the switch will provide the joystick with its operating voltage given by 6v battery as shown in Figure 2. The switch circuit was designed manually in the laboratory using two transistors of type (NPN, BC 547) and 8 resistances of (1kΩ) amount, other requirements are given in Table 1. Therefore, the 9v voltage signal that input to the switch is only employed to activate the circuit's transistor, which leads to close the circuit of joystick and 3v battery.

In addition, the used car composed of plastic body, two motors connected to rear wheels, communication system (receiver), and power from 6v battery. The motors are connected to an electrical circuit that supplied by power form batteries when receiving radio waves carrying the function in which step to move from the joystick to the car. A receiver converts signals from an antenna to a usable form. It uses electronic filters to separate a wanted radio frequency (17MHz) signal, the electronic amplifier increases the level suitable for further processing, and finally recovers the desired information through demodulation and decoding, and converts them into electric signal directed to the motors of the wheels to perform the motion orders comes from the controller. In addition to remote controller and switch circuit, robot body contains a microcontroller, which contains a specific analog and digital chips were used to fix the Bluetooth. The wireless video camera is also fixed on the robot body. The electric power of the video camera is independent of that of the robot, that because of their voltage are different between each other. This necessitate including another battery inside of the robot for operating the camera. It should be mentioned that the weight of all robot components should be in amount is appropriate to the force moment of robot motion that supplied by the actuators; i.e. motors.

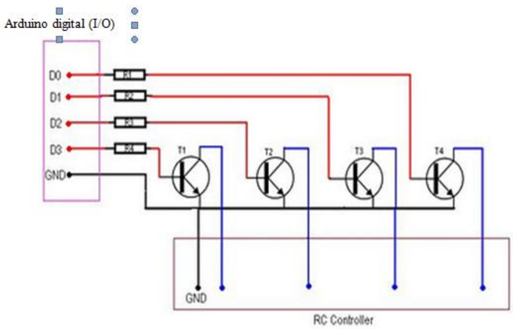


Fig. 2: The Electronic Switch Circuit for Four Bins

Table1: Switch Circuit Components

Component	Value	Quantity
Resistance (R)	1 (kΩ)	8 (one per line)
Transistor (T)	BC 547(NPN)	4 (one per line)
Ribbon wire	6 (cm)	10 (line)
Batteries	1.5 v	2

(ii). Visual Sensor

The used visual sensor is a mini surveillance camera of type: closed circuit television (CCTV) micro camera provided with one type of power cable and antenna as shown in Figure (3-a). The power cable receives the electric voltage from a local 12v battery installed on the robot. Such camera has high performance, good effect, high quality and good service specifications. The captured video is transmitted using the antenna of the transmission set found in the camera into remote receiving set by a wireless connection. Usually, the receiver set shown in Figure (3-b) is installed on the controller; it receives the video data in a form of analog audio-video radio signal (AV) through a radio band of frequency 2.4 GHz. The receiver set in the controller converts the analog signal comes from the camera into digital encoded data. The digital audio and video recoder

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 3, September 2014

(DVR) shown in Figure (3-c) is connected with the receiving set by audio and video cables; it is responsible on formatting the digital encoded data to be as video sequential frames, which transmitted into controller using the USB port. The frame rate of the used camera is adjustable, it may be 15frame/sec as maximum or can be set to be less than that. Figure3 shows the used CCTV camera and its receiving set.

(iii). Microcontroller

Arduino Leonardo microcontroller is used in current research, it is a single-board microcontroller designed to make the process of using electronics in multidisciplinary projects more accessible. It consists of hardware and software parts, the hardware is a simple open source board designed around an 8bit Atmel AVR microcontroller. Whereas the software is a standard programming language compiler and a boot loader that executes on the microcontroller. Arduino is an open-source electronics prototyping platform that can be used to develop interactive objects, taking inputs from a variety of switches or sensors, and controlling a variety of lights, motors, and other physical outputs. It has 20 digital input/output pins (of which 7 can be used as PWM outputs and 12 as analog inputs), a 16 MHz crystal oscillator, a micro USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with AC-to-DC adapter or battery to be operated [25]. where the Bluetooth card is fixed directly to the microcontroller Whereas, the digital output pins are employed to control robot directions. Arduino gets 9-12v electric power from the same battery of the robot. In addition, Arduino has a RAM to store the controlling software. When Arduino get started, the designed controlling software is loaded to the Arduino; the Arduino processor will interpret the loaded software and computing the data, and then distribute them into its corresponding pins.

(iv). Bluetooth Set

The Bluetooth is a communication set provides a data transform for short distance about 1-100m with low power consume 3.3-5v. It is commonly used between carrier devices and peripherals. In current project, it is used to transfer the commands from controller to the robot .Bluetooth set connect to the microcontroller board by wiring GND of Bluetooth module to the GND of microcontroller, Vcc of Bluetooth set to the 3.3v of microcontroller, RX of Bluetooth set to TX of microcontroller, and TX of Bluetooth set to the RX of microcontroller.

IV. SOFTWARE PART

The software sub-system is responsible on making the correct decisions that enable the mobile robot to achieve its destination. This is carried out by analyzing the video images received from the visual sensor. The video images are picturing the path in front of robot, which can be detected using image processing methods, which can be employed to yield decisions needed to guide the robot toward the detected path. Such that, it is intended to detect the robot's path by image processing. Line following routine is the complement part of the visual sensor to make the visual system of the robot. The visual sensor is a video camera facing the front path of the robot; the camera plays the role of the eye that picture the environment of the robot, while the line following module is the brain of the robot that interpret such picture. The used camera is capable to capture a video stream of 15frame/sec to be processed in purpose of getting useful information related to the planned path of the robot. Five stages are adopted to extract the planned path from the video stream; they are discussed in the following sub sections:

(i). Video Segmentation

The camera contains a wireless transmitter device in the robot used to send the video stream into the line following module found in the controller. To credit efficient and fast processing for the received video signal, it is just 5frame/sec are included in the processing and the remaining frames are neglected. The included frames are taken during equal times along the time interval. Thus, the control interval will be equal to five parts of second; 0.2 sec. Each individual frame is stand for an input still image for the next stage as shown in Figure3.

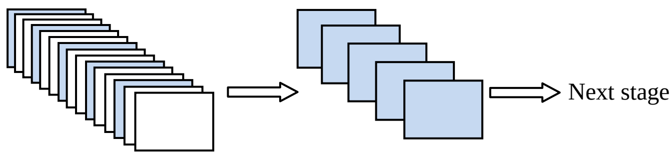


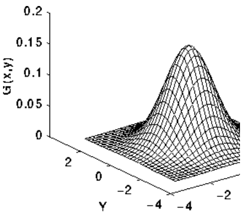
Fig. 3: Video segmentation.

(ii). Image Smoothing

The two dimensional (2-D) convolution operator based Gaussian smoothing is used to remove fine details and noise may found in the images resulted from the video segmentation stage. This smoothing is identical to the use of mean filter with a Gaussian kernel of bell-shaped hump. Such kernel has useful properties; it is noise sensitive and sharp edges conserved. To credit such results, the Gaussian smoothing is applied on the image with a large standard deviation, which an optimal smoothing filter for edge detection under the criteria used to derive the Canny edge detector [Gonzalez and R. Woods]. The 2-D isotropic (i.e. circularly symmetric) Gaussian has the form:

$$G(x,y)=\frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}} \qquad \dots (1)$$

Where, σ is the standard deviation of the distribution. It assumed that the distribution has a mean of zero (i.e. it is centered on the line x=0). This distribution is illustrated in Figure 4.



$\frac{1}{273}$ ×

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

(a) Gaussian distribution. (b) Corresponding mask.

Fig. 4: The 2-D Gaussian distribution with mean (0,0)and σ =1

The idea of Gaussian smoothing is to use this 2-D distribution as a `point-spread' function, and this is achieved by convolution. Since the image is stored as a collection of discrete pixels, it needs to produce a discrete approximation to the Gaussian function before performing the convolution. Figure.4 shows a suitable integer-valued convolution kernel that approximates a Gaussian with σ of 1.The 273 is the sum of all the values in the mask.

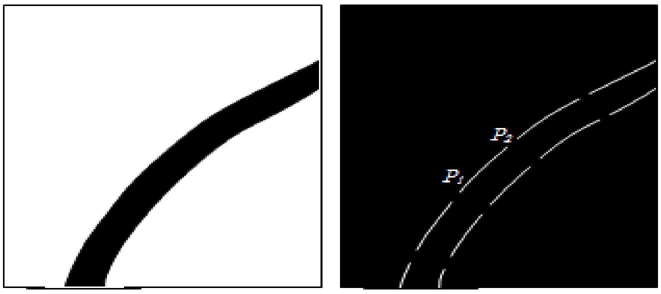
(iii). Image Color Filtering

To find out the planned path in the images resulted from the smoothing stage, it is should first be recognize its color from background color. In IDM programming environment, one can determine the minimum value (MinC) and maximum value (MaxC) for each color band in order to exclude all colors that do not appear in the image. This useful to detect any object in the image based on its chromatic features. It is well known that the attended bands are the three basic colors; red, green, and blue, which gets a value through a range of 0-255 for each band. The included colors are related to the capabilities of the camera and its starting setting. Practically, it is useful to fix the minimum values for the three bands to be 10, while the maximum values are 180. Such values make the edge detection algorithm to detect only the edges beside black color of the planned path line. Also, such process can exclude one or two band from the image when the MaxC is set to be equal to MinC for same band. This is useful when the color of the followed line be different of black.

(iv). Edge Detection

Canny edge detector is applied on the color filtered image. Such method is employed to highlight regions with high spatial derivatives, and then track these regions to suppress any pixel that is not at the maximum. Later, the gradient array is further reduced by hysteresis. Hysteresis is used to track along the remaining pixels that have not been suppressed. Hysteresis uses two thresholds and if the magnitude is below the first threshold, it is set to zero (made a non-edge). If the magnitude is above the high threshold, it is made an edge. And if the magnitude is between the two thresholds, then it is set to zero unless there is a path from this pixel to a pixel with a gradient above T2. The following steps explain the application of canny edge detector:

- Find the edge strength by taking the gradient of the image by Sobel operator given in eq. (1).
- The direction of the edge is computed using the gradient in the x and y directions given in eq. (2).
- Relate the edge direction to a direction that can be traced in neighbored pixels.
- Apply the non-maximum suppression.
- Eliminate breaks in the detected edges by the Hysteresis.



(a):Input image. (b): Output image.

Fig.5: Resulted Image by applying Canny Edge Detection Method.

The output of edge detection is a binary image contains black background, and bright edges. The edges are represented as a set of line segments, the position of each line segment is defined by two points P1(x,y) and P2(x,y), as shown in Figure5.

(v). Path Detection

The ground of the robot environment is chosen to be almost bright region containing a dark planned path. The use of such environment make the image resulted from Canny edge detection method is black containing few discrete lines represent the edge segments of the planned path. Therefore, the detected planned path is a collection of these edge segments. Since each line segment is defined by two points P1(x,y) and P2(x,y), the planned path can be stored in a structured vector (Path) contains a number of values is equal to the double value of the number of segments. Each element in this vector is composed of two components; one for x and another for y. The odd arrangements of the elements in such vector are belong to the first point P1 of the segment, whereas the even arrangements are belong to the later point P2 of the segment. Then, the detected planned path is used in the line following stage when the robot needs to decide what the next motion is.

(vi). Decision Making

Decision making depends on the output of the line following routine. The line following algorithm assumes that the robot located at the middle distance along the width (w=640 pixel) of the image, one can compute the average (xa) of all x2 values of all points Pi (i=1 to n; n is the number of line segments in the current image) that resulted from the edge detection stage. The deflection (Df) is the difference between xa and the half value of the width.

$$D_f = x_a - \left(\frac{w}{2}\right) \quad \ldots (2)$$

If absolute value of Df is less than a threshold amount (Dth) then the robot should be directed forward, but when Df is negative and less than (-Dth) then the robot should be turn left, and when Df is positive and greater than (Dth) then the robot should be turn right. This deflection will be taken in account to make decisions of robot motion as presented in Table 2. Such decisions are the output of current module, which are suspended decisions. Algorithm (1) shows how applying the line following technique to estimate the robot motion.

Table 2: Line following decisions based on Df determination

Path average (x_a)	Path deflection (D_f)	Decision
$(w/2)-D_{th} < x_a < (w/2)+D_{th}$	$\|D_f\| < D_{th}$	Forward
$x_a > (w/2)+ D_{th}$	$D_f > D_{th}$	Right
$x_a < (w/2)+ D_{th}$	$D_f < - D_{th}$	Left

Two fuzzy commands are used to control the left wheel and right wheel respectively. Combined these commands, the robot movement can be described by the following five fuzzy sets: (Forward, Backward, Right, Left, Stop), as presented in Table 3.

Table. 3: Types of robot movement.

Right motor	Left motor	Robot movement
Forward	Forward	Forward
Backward	Backward	Backward
Backward	Forward	Right
Forward	Backward	Left
Stop	Stop	Stop

At this time, the decision based on line following is used to generate the motion commands. The controller is now become ready to send the issued commands to the robot by Bluetooth set.

Algorithm (1) Line following

Input: Video frames

Output: ML

Procedure:

Step0: Open camera, Set image resolution (W*H).

Step1: Open serial port (for Bluetooth set), Set Color: C, MinC and MaxC.

Step2: For each time interval 0.2 sec; do the following steps

Step3: Get new frame

Step4: Perform smoothing by Gaussian smooth

Step5: if C is color (R,G,B) then

Get colored image filter on specific color (C)

Else

Get gray image

Step6: Apply Canny edge detection on the image

Step7: Get lines (array of line Segment2Ds)

Step8: Sum all X2 of P2 for each line, and find Xa

Step9: calculate Df =|(w/2)-Xa|

Step10: IF there is no line then

Stop Robot moving

Else

If |Df|<Dth then /*(Xa) >=(w\2 - Dth) and (Xa)<= (w\2 + Dth) */

Set ML= forward. /*ML= movement based line following */

Else if Df>Dth then /*(Xa)< (w\2 - Dth)*/

Set ML=right.

Else if Df<-Dth then /*(Xa)> (w\2 + Dth)*/

Set ML=left.

Else

Set ML=Stop.

End if

Step11: Send ML

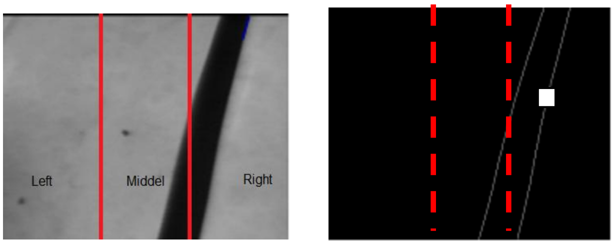
Step12: End for

V. RESULTS ANALYSIS

The line following method was based on dividing the width of each frame into three regions: left region starting from 0 to (half frame width-Dth), then the middle region extended up to (half frame width+Dth) and third right region end at the frame width. By assuming Dth=40, these regions are numerically

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 3, September 2014

determined as:0-280, 280-360, and 360-640 respectively. Fig.(6-a) shows these regions, whereas Fig.(6-b) shows same frame after applying Canny edge detection, the result is a set of line segments in binary image, the computed average path distance (Xa) is referred as small bright square ling in the right region since the planned path is directing toward the right.



(a) Original frame's regions: (Left, Middle, and Right) **(b) Canny edge detection, the line and x_a are shown.**

Fig.6: Canny edge detection results refers to that the detected path is directed to the right

The images illustrated in Figure 6 shows the ability of canny method to detect different directions of planned path. It is shown the detected boundaries in Figures (6-b) are thin, and each line segment is very close to next one, they are seem to be continuous. The accuracy of the canny method in determining the planned path make the estimated position of the path average to be accurate too.

(i) Path Detection versus Deflection

Practically, it is found that the amount of deflection threshold Dth affect the result of path detection when the frame rate is set at a specific test values (e.g. 4 or 5 fps). By considering the test map mentioned before, the robot was asked to return a Boolean value is true when it detected the panned path, and False when there was no path detected. This test repeated four times, each for different deflection threshold (Dth), the adopted Dth are (20, 40, 60 and 80). The behavior of the robot was measured according to the returned Boolean value as given in Table 4. The frame rate less than 4fps gave True return values at all turns with different deflection threshold Dth, while the frame rate greater than 5fps are excluded since it leads to late the robot mission, this fact is explained in the next sections. It was noticeable that the robot enjoying a great ability to detect the planned path at different situations, it do not detect the path only at the second and third turn when the frame rate is 4fps and Dth is 80, this is due to the wide middle region makes no deflection to be occurred since the position of (Xa) lies inside the middle region at the second turn. This make the robot directed forward till passing the second turn, the sharp rounding of the second turn causes the planned path to be lost from the robot vision quickly, which lead to no sighting the third turn, so the returned situation was false for both two turns. By getting less Dth, the middle region was narrower; the robot became keen to catch the planned path. At the case of frame rate=5fps, the situation of losing the planned path was appeared at the wide middle region again. Whereas, the narrower middle region when Dth equal to 40 or 20 pixel, all the returned values were True since the planned path at the SECOND turn was estimated at the left region, which make the robot to be turned toward the left. Time by time, the robot followed the planned path and could be known its way well. Therefore, one can conclude that so larger Dth will not generate an adequate response synchronized between line and location, while smaller Dth a more complicated movement correction will be the dominant events. According to such analysis, one can decide that the deflection threshold Dth=40 pixel are the best value that enable the robot to follow the planned path correctly.

Turn No.	Frame rate=4 fps				Frame rate=5 fps			
	Dth =80	Dth =60	Dth =40	Dth =20	Dth =80	Dth =60	Dth =40	Dth =20
Turn (1)	True	True	True	True	False	True	True	True
Turn (2)	False	True	True	True	True	True	True	True
Turn (3)	False	True	True	True	True	False	True	True
Turn (4)	True	True	True	True	False	False	True	True
Turn (5)	True	True	True	True	False	True	True	True

Table. 4: Path detection results with respect to the variation of Dth and frame rate

(ii) Mission Time versus Frame Rate

The mission time estimated for the robot to complete its round from beginning point to the end destination point was measured for each considered value of frame rate as given in Table 5. It was found that the time is decreasing nonlinearly with increasing Dth as shown in Figure 7. To ensure such behavior, different values of Dth was considered.The analysis of the robot behavior according to the information that extended horizontally at the default value of Dth (i.e. 40pixel) in Figure 7 shows that the mission time consumed short temporal amount of about 2.16sec at frame rate is 4fps,in comparison with other frame rates. The variation of mission time in such case is expected and exponentially increasing. The frame rate values less than 4fps make the robot late to reach its destination due to it processed the current frame in time and wait for reaching the next frame, the wait time is lost without any action. While the frame rate values greater than 4fps are make the controller take more time to process information of current frame that are same as that found in previous frame. This was slightly late the robot motion, which showed little increasing in the behavior of mission time. Therefore, one can conclude that the best value of frame rate that enables the controller to analyze the captured image in time with less deflection is the middle value (i.e. 4fps).

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 3, September 2014

TABLE.5: MISSION TIME (IN SEC) VERSUS FRAME RATE AT THE CONSIDERED DTH.

Frame rate (fps)	Dth=20 pixel	Dth=40 pixel	Dth=60 pixel	Dth=80 pixel
1	14.59	12.42	11.27	6.27
2	8.25	5.39	6.51	4.35
3	4.72	2.64	2.39	3.17
4	4.01	2.16	2.06	1.56
5	2.29	1.09	0.59	0.49
6	4.37	1.62	1.44	1.88
7	6.62	2.03	1.72	2.31
8	7.82	2.17	2.51	3.49

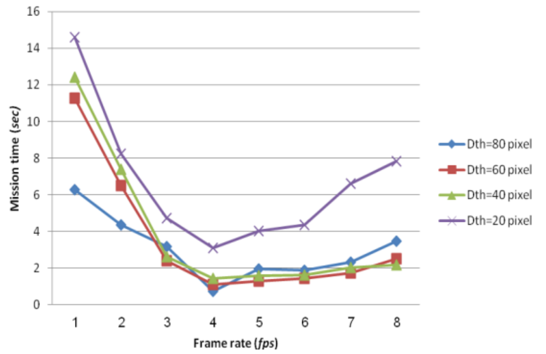


Fig.7: Mission time versus frame rate for different values of D_{th}.

Whereas, the analysis of the robot behavior according to the information that extended vertically at frame rate=4fps in Figure 7 shows that the mission time decreases unequally with equal increases of Dth, this is due to less decision making and therefore less computations were needed in the case of great Dth since the planned path appeared on head, i.e. the current case is forward and the next is forward too. But, when the considered deflection Dth became greater, the middle region in the robot view became narrower and planned path became outside the middle region, which the case that requires for the robot to do some other things such as left/right turn. The turn is curried out when reversing the direction of moving one wheel, which leads to slow the overall motion of the robot and leads to late its reaching into the destination point. This ensures that the proper value of Dth is 40pixel and the best value of frame rate is 4fps.

(iii) Response versus Mission Time

The mobility characteristic of the robot depends on its motor kind, which determines the motion response of the robot. In the present work, the used motor is DC electric motor (toy motor) rotated by 360o, the use of this kind does not give the chance to control the angular velocity. Instead, the available method to control the time rate of motor revolutions is the time duration (or, period) only, which is the second face of displacement when the speed assumed to be constant. Therefore, there was a need to determine the best period of the motor that corresponding to optimal response of the robot. Figure 8 presents the mission time of the robot at different periods with fixing default values of frame rate and deflection threshold.

It should be mentioned that the time period less than 50ms caused a fluctuated behavior for the robot when follow the planned path. In spite of the fluctuated behavior, the robot was still over the planned path and the resultant direction was toward the destination. Thus, such periods are excluded in the analytical study of the response. Whereas, the period values greater than 50ms that shown in Figure 8 more stability, and the stability increases by increasing the motor period.

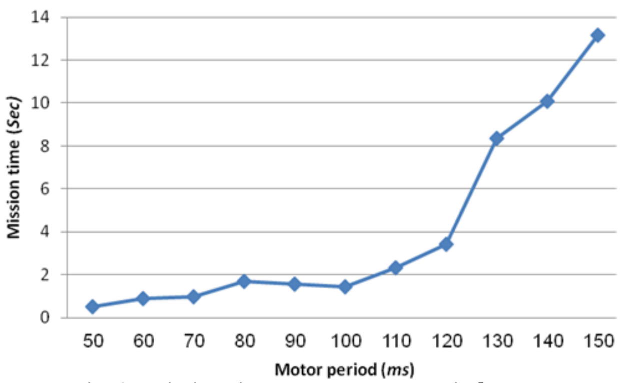


Fig. 8: Mission time versus motor period.

It is noticed that the mission time is exponentially increases as a function to the motor period, which indicates the small increase in the period causes greater increase in the mission time that lead to late the reach of the robot. In all the considered periods, the robot success to reach its destination, the amount of late was increases each time. The least time period displayed acceptable behavior of the robot and the presentation was enjoying. While greater time period displayed very slow behavior of the robot, which the case that similar to periodic waiting state happen to the robot behavior. Therefore, the best value of the motor period is 50ms since showed fast and stable behavior of the robot at suitable frame rate (i.e. 4fps).

(iv) Actual Path Results

The actual path results describe the behavior of the robot when follow the planned path. The implementation of line following method on the considered map by setting the

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 3, September 2014

suitable values mentioned in previous analysis make the robot travels along the planned path well. For more explanation, different cases of robot situations were considered. In these cases, there was few deflections occurred in the actual path of the robot from the planned path were noticed. In order to examine the robot behavior, both planned and actual paths are recorded at different frame rate and motor period, the results of such considered cases and the specifications of each case are given in Figures (9-12), these figures show the behavioral performance of the robot when follow the planned path, which is the actual path. The case of 2fps was considered two times; the first was very slow and takes a long time 21minute, then it was repeated by increasing the motor period to be 100ms to consume less time of about 5:39, which ensure that the motor period is related to the response of the robot.

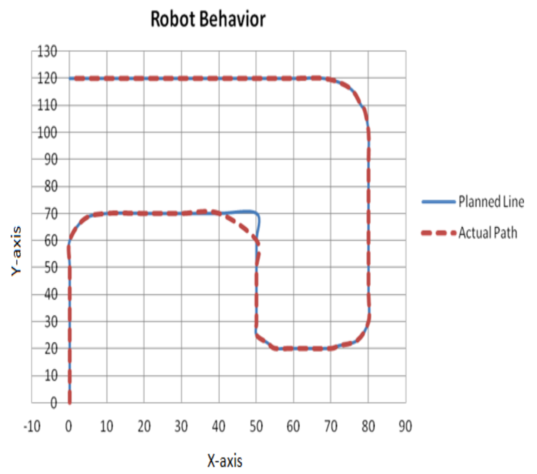


Fig. 9: Robot behavior of line following when frame rate is 2fps, motor period is 50ms and mission time is 21:09 minutes.

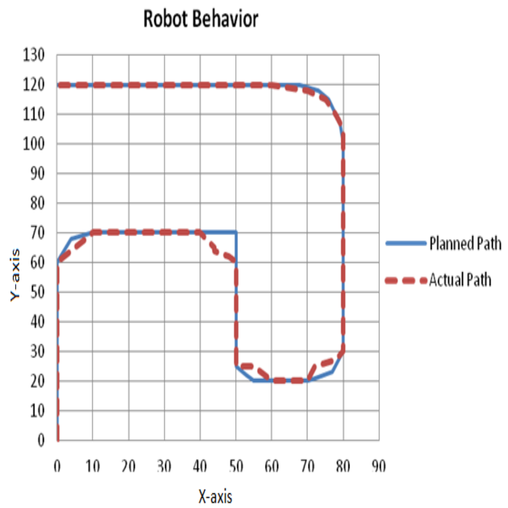


Fig. 10: Robot behavior of line following when frame rate is 2fps, motor period is 100ms and mission time is 5:39 minutes.

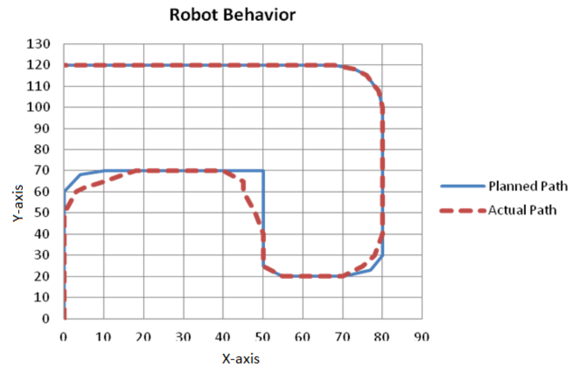


Fig. 11: Robot behavior of line following when frame rate is 4fps,motor period is 50ms and mission time is 1:43 minutes.

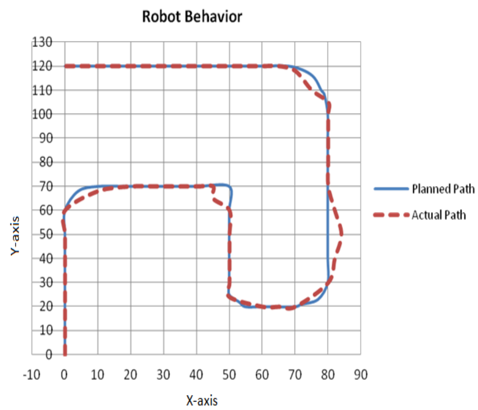


Fig.12: Robot behavior of line following when frame rate is 6fps,motor period is 50ms and mission time is 1:09 minutes.

It is shown that the actual path was matching the planned path at the straight forward directions. There was small deflections occurred at turns, where the actual path was deflected from the planned path, this deflection is non uniform and cannot be expected. The measurements indicates that the maximum deflection in all considered cases was 5cm, which a small regardless value in comparison with the width of the robot. In addition, one can noticed that the deflection was happen at one side with respect to the planned path, which means there is no fluctuations are occurred in the actual path. This refers to high stability of the robot to follow the intended trajectory. Furthermore, the consideration of the actual path with the variation of frame rate and motor period are shown in same figures. The deflection at the first turn was minimum in case (1) where the frame rate is 2fps and motor period is 50ms. This is due to fast computations were occurred for less frame rate, i.e. less input data. Whereas, the deflection of the actual path was greater at the sharp second turn in comparison with other turns. Also, the frame rate 6fps with motor period 50ms in case (4) showed some overshot in the actual path at the turns. The maximum amount of the occurred overshot was about 4cm, which is regardless since the robot still walking over the planned path. The occurred overshot is due to higher value of motor speed that leads to the faster motion of the wheels, which make the motion continuity push the robot toward the direction of motion, and finally affect the control process. The shortness happen in the actual path in comparison with the planned path is due to deflecting robot motion far away from the planned path at turns. Since there is no restriction for the robot to reach the turn, the deflection was beginning from a distance before reaching the turn. This distance is related to the angle of tilting the camera and field of view of the camera, i.e. at these angles, the planned path appears viewed to the robot. By measuring the actual path length for the Four considered cases given in figures (9-12), it is found they are: 352.58cm, 350.3cm, 346.5cm, and 352.88cm respectively. No matter how much the deflection was occurred since the robot match the planned path any way. But, it is important to measure the amount of matching. Such that, one can compute the matching percent in each case using the following empirical equation:

$$M_A = \frac{L_M}{L_P} \times 100\% \qquad \ldots (3)$$

Where, MA is matching percent of the robot path, LM is the length of the actual path parts that match the planned path measured in centimetre, and LP (366 cm) is the length of the planned path measured in centimetre, then the deflection percent can be computed as follows:

$$D_A = 1 - M_A \qquad \ldots (4)$$

According to these equations, the measured LM for four considered cases was: 349cm, 322cm, 313cm, and 301cm respectively. Whereas the corresponding computed matching percentages were about 95.3%, 87.9%, 85.5, and 82.2% respectively, which refer to deflection percentages of about 4.7%, 12.1%, 14.5%, and 17.8%. These quantitative results indicate that the behavior of the robot in second case is the best since the matching percent is high and mission time is little. Also, the third and fourth cases are acceptable too, while the first case is undesirable because a long time was required to reach the destination.

VI. CONCLUSIONS

The proposed robotic system showed acceptable behavior when applying the line following method; it showed encourage results when tested with different situations of turns. The best value of the deflection threshold Dth in the line following method was 40pixels. The values greater than that, were fast the robot actions, while less values showed uncontrolled motion. The useful frame rate values in the line following method were 2fps up to 6fps. The controller response was related to the motor period, which was useful to be set at 50ms. For getting fast response, 100ms was very useful.

REFERENCES

[1] Misra and Kate N. Nagaraj,"Security in Wireless Ad Hoc Networks", in Book The Handbook of Ad-Hoc Wireless Networks (Chapter 30), CRC Press LLC, 2003.

[2] Midong Yhou, Zygmunt J. Hass, Securing Ad-Hoc Networks, IEEE Networks Special Issue on Network Security, November/December 1999.

[3] Data Integrity, from Wikipedia, the free encyclopedia,[Online]http://en.wikipedia.org/wiki/Data_integrity.

[4] Z. Hu, C. Peter, S. Johnson, Ariadne: A Secure On-demand Routing Protocol for Ad Hoc Networks, in Proceedings of ACM MOBICOM'02, 1986.

[5] R. Sonzgiri, M. Dehill, V. P. Levine, C. Shields, E. M. Belding-Royer," A Secure Routing Protocol for Ad-Hoc Networks, in Proceedings of ICNP'02, 1978.

[6] Y. Hu, A. Perrig, D. Johnson, "Packet Leashes: A Defense against Wormhole Attacks in Wireless Ad-Hoc Networks", in Proceedings of IEEE INFOCOM'03, 1965.

[7] Frank L.Lewis, Darren M.Dawson, Chaouki T.Abdallah "Robot Manipulator Control Theory and Practice". Second Edition, by Marcel Dekker, 2004.

[8] Lee, C.H. "Recurrent neuro fuzzy control design for tracking of mobile robots via hybrid algorithm", Expert Systems With Applications, 2009.

[9] Vamsi Mohan Peri, "Fuzzy Logic Controller for an Autonomous Mobile Robot", Master thesis of Science in Electrical Engineering at the Cleveland State University, MAY, 2005.

[10] Giuseppina Gini, Alberto Marchi, "Indoor robot navigation with single camera vision", proceeding of the 2nd international on pattern recognition information system, Ciudad Real, Spain, 2002.

[11] Yuki Ono, Hajime Uchiyama and Walter D. Potter, "A mobile robot for corridor navigation: a multi agent approach", ACM press, Hunsvile, Alabama, USA, 2004.

[12] M. Ghanavati and A. Ghanbarzadeh, "Control and Guidance of an Underwater Robot via Fuzzy Control Method", proceeding in Int J Advanced, Design and Manufacturing Technology,Vol. 4/ No. 1/ Desember-2010, ISSN: 2008-1421.

[13] WWW.ARDUINO.cc.URL:http://arduino/arduinoBoardLeonardo/.2013.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 3, September 2014

AUTHOR BIOGRAPHY

First Author Assist .prof. Dr. Mohammed Sahib Mahdi Altaei has B.sc. From Al-Nahrain University/dept. Of Physics, Msc. And P.hd.from Al-Nahrain University in digital image processing and works in Al-Nahrain University Dept. of computer Science.

Second Author Assist .prof. Dr. Loay Kadom Abood has B.sc. from University of Baghdad and Msc. And P.hd.from University of Baghdad in digital image processing and works in University of Baghdad Dept. of computer Science.

Third Author Dhuha Abdullah Amer has B.sc from Al-Nahrain University/dept. of computer science and Study M.sc. in University of Baghdad/dept. of computer science.