

Simulation of Moving Obstacle Avoidance for Auto Guided Land Vehicle

Mohammed Sahib Mahdi Altaei ¹, Laith A. Al-Ani ² and Qudama Khamis ³

¹ Department of computers, college of science, Al-Nahrain university, Baghdad-Iraq.

² Department of physics, college of science, Al-Nahrain university, Baghdad-Iraq.

³ Department of physics, college of science, University of Al- Anbar, Baghdad-Iraq.

Corresponding Author: qudama.khamees@gmail.com

Abstract

This paper concerned with simulating the behavior of designed control system that tries to govern robot to safely passing a path containing moving obstacle ahead. The used robot carried some specific sensors are used to sense the existence of obstacles along the moving path, these sensors are IR sensor that used to detect the obstacle and visual sensor that used to measure the size and the distance for the obstacle. In order to overcome the obstacle, the controller assumes new transient away point at the far side of the obstacle, and guides the robot to pass through that point. The position of such transient point is depending on the size and direction of the obstacle. Then, whenever the robot close to the transient point, the controller guides the robot to identify the intended path again. This algorithm enables the robot to move far away from the moving obstacle and then back it into planned path. [DOI: [10.22401/ANJS.21.4.07](https://doi.org/10.22401/ANJS.21.4.07)]

Keyword: IR sensor; Visual sensor; Obstacle avoidance; Mobile robot.

Introduction

Robotics is widely used in many industries due to the high level of performance and reliability. A mobile robot needs locomotion mechanisms that enable it to move unbounded throughout its environment. But there are a large variety of possible ways to move, and so the selection of a robot's approach to locomotion is an important aspect of mobile robot design [1]. All mobile robots have some kinds of features to obstacle avoidance, where they have been developed to help the human to perform the hard jobs in life; different types of robots are found according to its task. Commercial and industrial robots are now in widespread use performing jobs more cheaply or with greater accuracy and reliability than humans [2]. Designing autonomous robot requires the integration of many sensors and actuators according to their task. Obstacle detection is primary requirement for any autonomous robot. The robot acquires information from its surrounding through sensors mounted on the robot [3]. Various types of sensors can be used for obstacle avoiding. Methods of obstacle avoiding are distinct according to the use of sensor. Some robots use single sensing device to detect the object. But some other robots use multiple sensing devices. [4].

Navigation of a robot without outside help is one of the most important aspects of developing a fully functioning autonomous mobile robot. Computer simulations are recognized and used widely as valuable design and testing tools. The benefits of a robot simulation are numerous. They can be used to develop control algorithms for a robot and verify the correct operation of the robot. Initial development and testing can be accomplished in simulation, making the robot available for other use [5]. Autonomous navigation is associated to the availability of external sensors that capture information of the environment through visual images or distance or proximity measurements. The most common sensors are distance sensors (such as, infrared "IR", ultrasonic, or laser) capable of detecting obstacles and of measuring the distance to barriers close to the robot path[6]. The problem of robotic guidance has attracted a lot of research. The applications field was still searching about an accurate method that can be used to improve the guidance results. The most interested researches besides our contribution are briefly explained in the following subsections:

Several studies have been published in the field of interest; there are a great deal of focus was granted to obstacle avoidance for mobile

robot and simulation for auto guide land vehicle to avoid obstacles. Sachin Modi in 2002 presented an obstacle avoidance methods for an autonomous guided vehicle, where obstacle avoidance may be divided into two parts, obstacle detection and avoidance control. These include fixed mounting of sonar sensors, a rotating sonar sensor and a laser scanner. The systems were tested at the International Ground Robotics Competition. This test bed system provides experimental evaluation of the tradeoffs among the systems in terms of resolution, range and computation speed as well as mounting arrangements [7]. Aye A. N. in 2008 established a software implementation of obstacle detection and avoidance system for wheeled mobile robot. This system consists of infrared sensors and microcontroller. Three infrared sensors are used for left, front and right. The input signal is received from sensor circuit and operated according to the received sensor signal. The infrared sensor reading is taken and processed to avoid the obstacles. The desired goal of this system is to avoid obstacles along its path and determining the distance [8]. Rusu C. G. in 2010 developed a fuzzy obstacle avoidance system for an autonomous mobile robot using IR detection sensors. They present implemented control architecture for behavior based mobile robot. The mobile robot is able to interact with an unknown environment using a reactive strategy determined by sensory information. A rule-based fuzzy controller with reactive behavior was implemented and tested on a two wheels mobile robot equipped with infrared sensors to perform collision-free navigation [9]. Matteo S. in 2014 presented an algorithm allows the robot to avoid moving obstacles and to reach the assigned goal is proposed. A dynamical system (DS) modulation matrix is calculated using the distance from the obstacles and their velocity, without the need of an analytical representation of the obstacles. This matrix modulates a generic first order DS, used to generate the desired path, saving the equilibrium points of the modulated system [10]. Jiechao L. in 2017 presented a model predictive control-based obstacle avoidance algorithm for autonomous ground vehicles at high speed in unstructured environments.

Obstacles are detected using a planar light detection and ranging sensor. Simulation results show that the proposed algorithm is capable of safely exploiting the dynamic limits of the vehicle while navigating the vehicle through sensed obstacles of different size and number. The proposed variable speed formulation can significantly improve performance by allowing navigation of obstacle fields that would otherwise not be cleared with steering control alone [11].

The significance of our work falls in flexible treatment of robot guidance to overtake moveable obstacles facing the robot along the intended path.

This increases the understanding of obstacle avoidance for robot control and the applications of autonomous guided vehicle technology. The used novel algorithm is capable to control the robot to avoid obstacles at high speed taking into account dynamical safety constraints through a simultaneous optimization of reference speed and steering angle.

The problem of determining the position of the robot is very important point due to its relations with the complexity of the used computational method. In presence, the linear positioning relations given in equations (1) are used to determine the position of the robot[12]:

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} 91 & 0 \\ 0 & 112 \end{bmatrix} \begin{bmatrix} X_o \\ Y_o \end{bmatrix} = \begin{bmatrix} L\phi - L\phi_o \\ L\theta - L\theta_o \end{bmatrix} \dots\dots\dots (1)$$

where $L\phi_o$ and $L\theta_o$ represent the longitude and latitude respectively of the reference point which can be chosen as a midpoint within the planned guidance path. The 2-D cartesian space that determines the position of the vehicle is described by the two plane axes X and Y to define the head of the vehicle on the surface of the Earth with respect to reference point as shown in Fig.(1). The instantaneous components of the velocity could be described by the spherical coordinates as follows[13]:

$$V_x = V \text{Cos}(\text{Head})$$

$$V_x \approx \frac{\Delta x}{\Delta t} = V \text{Cos}(\text{Head}) \dots\dots\dots (2)$$

Where Δx and Δy are represent the positioning deflections respectively during one short time interval Δt , $Head$ are the head angle of the car.

Equations (1) can be utilized to determine the positioning deflection due to the car motion, which leads to determine the deflections of the car head as follows [14]:

$$\Delta Head = \tan^{-1}\left(\frac{\Delta y}{\Delta x}\right) \dots\dots\dots (3)$$

where, $\Delta Head$ is the deflection of the head.

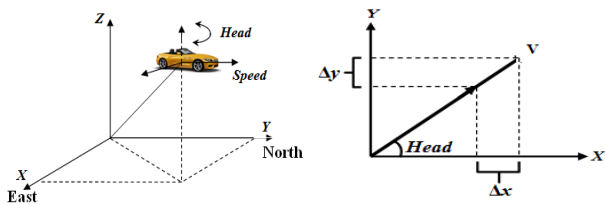


Fig (1): The directional head of the robotic car.

When using land vehicle like a car, the guidance parameters describe the guidance tools of the car and its relation with navigation parameters; they include the angular descriptors of the wheels rotation. When the vehicle needs to be turned, the guidance parameter that is responsible for the turn is the wheel rotation (R), which is determined by the range -45 to 45° direction. Whilst, the guidance parameter that responsible on changing the speed is the revolution per minute (RPM) of the engine. The control parameters cause specific changes in the guidance parameters. Whereas, each guidance parameter has a direct effect on the corresponding control parameter. Therefore, the control tools can guide the car to specific guidance states according to the linear proportional command model as follows[15]:

$$\Delta C_i = k_{2i} \Delta G_i \dots\dots\dots (4)$$

where ΔC_i and ΔG_i are the deflections of control and guidance parameters, k_{2i} is the proportional factor, and i is an index takes the values (1 or 2), it specifies the control and guidance parameters according to Table (1) shown below.

Proposed Guidance Method

The generic structure of the proposed obstacle avoidance robotic system is shown in Fig.(2). The proposed method contains three main stages; they are *Setting Stage*, *Control Module*, and *Obstacle Avoidance Module*. The setting stage is a preparing stage in which the initial values of the used parameters are determined; these initial values are either determined by used sensors or determined by assigning useful initial values.

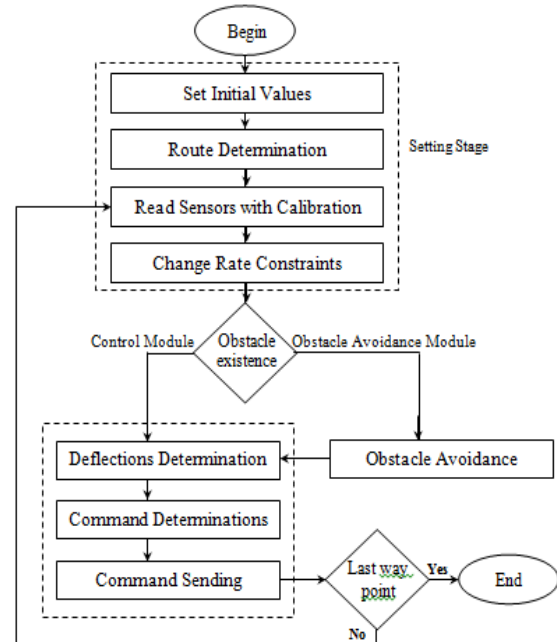


Fig.(2): Blok diagram of proposed control and obstacle avoidance method.

Initially, the sensors on board of the vehicle are used to sense the condition parameters of that vehicle, while the navigation and guidance parameters are set to be zeros. The controller is based on the concept of action reaction that mentioned before; it aims at controlling the auto guided car during its mission. The idea is implemented by introducing a specific action according to accounted determinations, and then the expected reaction will navigate the car toward the desired situation. Whereas, the obstacle avoidance stage is based on the feedback of the IR sensor that detect an obstacle on the path of the vehicle and then suggest an effective method for dealing with obstacle that ahead the robotic vehicle.

Path Specification

The map carries information about the geographical features of the guidance region. This information besides the desired conditions are put in a dedicated *database* established for a certain guidance trip. The database has a set of records, each record includes the intended attitude of the car at certain route segment connecting a pair of successive way points. Table (1) shows an example of planned attitude registered as records in the database.

Table (1)
Input database of an intended situations.

Way Points	Position		Speed (km/s)	Radius of arrival (m)
	Longitude (km)	Latitude (km)		
1	2×91	2×112	0.1	0.25
2	5×91	4×112	0.125	0.25
3	6×91	8×112	0.10	0.25
4	12×91	3×112	0.13	0.25

The last column in Table (1) helps the navigator to switch to the next way point. The process of switching is done by comparing the distance *RemDis* between the current position (x_{Act}, y_{Act}) on the actual route and the target way point (x_2, y_2) with the radius of arrival of the target way point. If *RemDis* is less than the radius of arrival then the car is considered arrived to the end of current route segment (reach the area of the target way point). Therefore, the auto-navigator should switch with the next way point within the planned route. The pointed circle, shown in Fig.(3), represents the area of the arrival of the way point, and its radius is the radius of arrival of that way point.

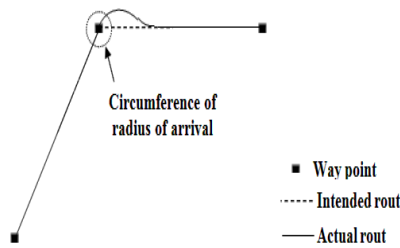


Fig.(3): The radius of arrival.

Obstacle Avoidance Module

This module is operated only when the controller detects an obstacle along the actual path. It depends on the feedback of both IR sensor and visual sensor. The IR sensor tells the distance between the car and the obstacle, while the visual sensor tells the size of the obstacle and its motion direction. Actually, the visual sensor is a simple camera gives a picture of dark obstacle in bright environment, which is used to estimate the length of the obstacle (L_{Img}) and converts it into true length (L_{Obs}) that measured by meters using the width of the image (W_{Img}), scale factor (S_F) and (W_{Real}) that related to the field of view (FOV) of the used camera shown in Fig.(4), as follows [14]:

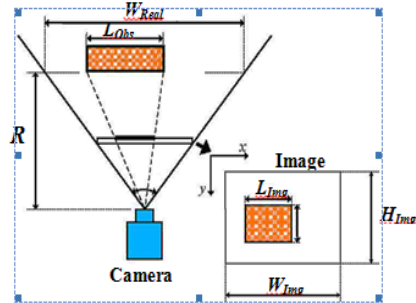


Fig.(4): Obstacle information given by visual sensor.

$$L_{Obs} = \frac{L_{Img}}{W_{Img}} \times S_F \dots\dots\dots (5)$$

$$S_F = \frac{W_{Real}}{W_{Img}} \dots\dots\dots (6)$$

Also, the successive image captured by the visual sensor enables to determine the direction of the obstacle motion, the two differences between two successive images showed that the direction of the obstacle is biased toward the right or the left of the car. The changes that happen in the two captured images are detected by computing the absolute difference between the current image and the next one. The result of the difference is an image which contains expanded dark region except the region of the moving obstacle that appeared bright.

Fig.(5 and 8) shows the difference image, in which the dark regions refer to the fact that the difference between the corresponding regions in the two images is very small approach to zero, due to the same background

appearing on the two frames whereas the difference of the region in which the obstacle moves, is greater in comparison with the dark region. Such regions appeared relatively lighter.

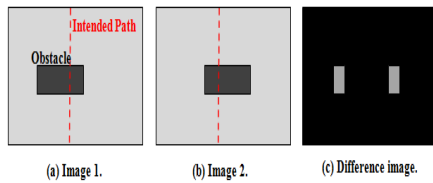


Fig.(5): Difference image computation.

The direction of obstacle motion is determined by computing the vector of pixels weight (V_w) two times, where V_w is one dimensional array of length equal to the width of image, each element in the V_w represents the average value of the pixels in the difference image as Fig.(6) shows.

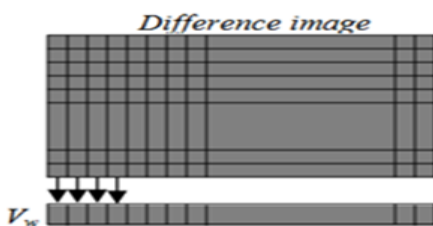


Fig.(6): computation of V_w .

The behaviors of such curves contain two characteristics peaks as shown in Fig.(7), each peak refers to the location of the obstacle in one image.

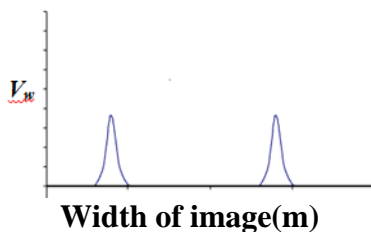


Fig.(7): V_w versus width of the image.

The direction of obstacle motion (D_r : right or left) is determined using localization (L_p) of these peaks, whereas the speed of the obstacle (S_{Obs}) is determined by dividing the average absolute differences of the two peaks localizations in the two images by the average time ($2s$) which requires to move the obstacle from the position (1) in the first image up to position (2) in the second image as shown in Fig.(8).

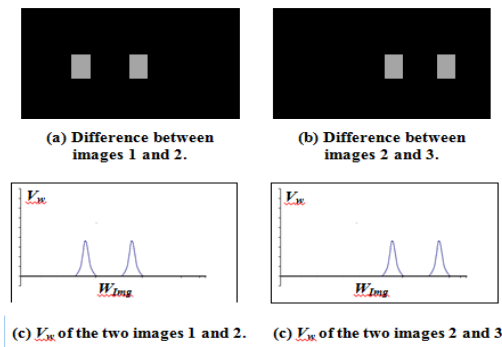


Fig.(8): Difference image computation.

Thus, the direction and speed of the obstacle are determined as follows:

If $L_{p,1} > L_{p,2}$ then the obstacle directed toward left.

Else If $L_{p,2} > L_{p,1}$ then the obstacle directed toward right: Otherwise the obstacle is stopped

$$S_{Obs} = \frac{\sum_{i=1}^2 |L_{p,i+1} - L_{p,i}|}{2} \dots\dots\dots (7)$$

In such case, the controller creates a transient way point at location in front of the obstacle and far away from the obstacle edge by a safety distance (D_s), and then switches the control process to this point. Thus, the transient point will be the new destination that the controller drives the car into. Thus, the controlling events may be changed according to the new change as follows: if the obstacle is directed toward the right of the car, then the transient way point will be at the left while the car may be speeded down when the obstacle moves in slow speed, otherwise the car remains in its same speed when the obstacle moves in relatively higher speed and the remaining distance between the car and the obstacle is enough make the required change.

In this research, it is assumed that the obstacle moves horizontally in the direction that is perpendicular to the attended path. Therefore, the IR sensor feeds a binary reading: True refers to the existence of the obstacle while False refer to the absence of obstacle. Actually, false IR sensing may also refers to absence of the obstacle along the sensing range of the IR sensor, where the IR sensing may be changed into true when the car becomes closer to the obstacle and the obstacle being in the range of the sensor. In case of true IR sensing, the IR sensor feeds the distance (R) between the car and obstacle.

Results and Discussion

The implementation require firstly to feed the data table of the guidance parameters given in Table (2). Table (3) presents the guidance specifications that could be setup for calibrating and ensuring smooth behavior in the change of the guidance parameters. These specifications are usually determined depending on the car capabilities.

Table (2)
Data table of the guidance trip.

Way point	Position		Speed (km/h)	Radius of arrival (m)
	Longitude (km)	Latitude (km)		
1	10×91	20×112	20	5
2	20×91	50×112	80	5
3	15×91	80×112	40	5
4	80×91	90×112	70	5
5	95×91	55×112	60	5
6	70×91	25×112	30	5
7	60×91	70×112	100	5
8	85×91	5×112	90	5
9	40×91	10×112	70	5
10	27×91	47×112	80	5

Table (3)
Specifications of the considered parameters.

Parameter	Min. value	Max. value	Max. change
Speed (km/h)	0	100	1
RPM (rpm)	1000	8000	70
Throttle (unit)	1000	8000	70
Turn Degree)	-180	180	3.6
Rotation Degree)	-45	45	0.9
Steering Degree)	-90	90	1.8
Head (Degree)	-180	180	3.6

The response coefficients is set to be one, the radius of arrival is set 5 m, and the maximum change of each parameter is equal to one percent (1%) of the range of any parameter, this percentage is called Fraction factor since it may be useful to control the dynamic response. It has been noticed that the setting of the fraction factor at 1% provides least overshoot and more stability for the navigation and guidance parameters. Fig.(9) shows the actual and intended route of the simulator.

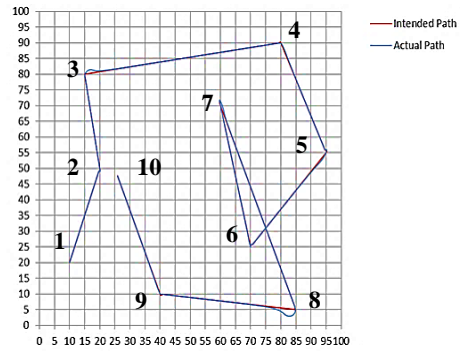


Fig.(9): *Intended and actual paths of the considered ten way points.*

Condition and Control Parameters Results

In the experiment, the condition parameters versus guidance and navigation parameters were recorded and analyzed. Table(2) presents the changes in route segment toward the second way point. By consider just the speed condition parameter and note its behavior after switching to the second way point (i.e., this happens after passing through the first way point) that predetermined in the guidance plan, then the value of the intended speed will be 80 km/h and the actual speed became 20 km/h is the intended speed of last route segment). The determined change of speed has been allowed to reach maximum positive value of the corresponding guidance parameter (maximum, $IntRPM=60$), the positive sign comes from the positive desired change of the speed (i.e., the simulator was positively accelerated) while the actual speed was still zero (i.e., $ActRPM=0$) since the guidance (at the way point switching instance) was still at level state. Therefore, the deflection of the RPM become a large positive value. Already, the deflection in the speed is restricted by the maximum amount of change in the speed, therefore, the current value of the actual change of the speed . This small change in the speed makes fewer changes in the throttle by an amount proportional to the amount of the attended deflection in the speed. Thus, the throttle that has a previous value equal to zero will change its state to be equal to 1, this leads to an increase in the RPM and then actual speed, and the simulator has shown positive accelerated behavior. As a result, the state of the speed was changing with an associative increase in the throttle. Fig.(10) shows the behavior of the actual speed when the simulator follows the intended path given

in the considered ten waypoints of Table (1). It is shown that the actual speed rise from the initial zero value up to the intended speed of value 20 km/h at about 40s, which is an acceptable acceleration for adopted type of robotic car. Then, the speed remains at the intended value until reaching the second waypoint where the speed is changed into the intended speed of 80 km/h and then other intended values of other way points. It is noticeable that the overshoot occurs when the speed becomes closer to the intended value in the two cases of speed up and speed down, which has small value and it is quickly corrected.

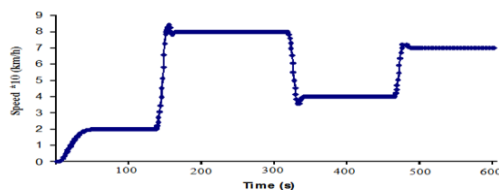


Fig.(10): Behavior of the simulator's speed through the considered eight waypoints.

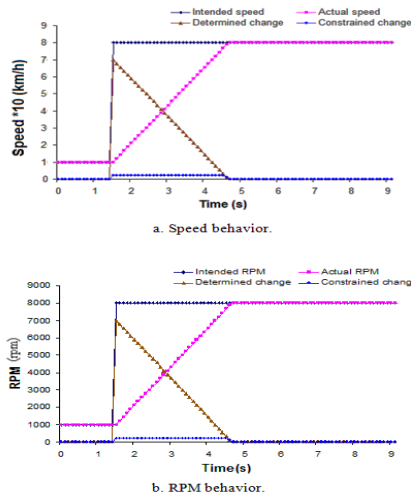


Fig.(11): Simulated speed and RPM at speedup state.

Figures (11-13) show the behaviors of the actual and the intended speed besides their deflections and constrained deflection for one considered intended path segment. The behavior of the RPM and throttle associated with the intended speed variation are given below:

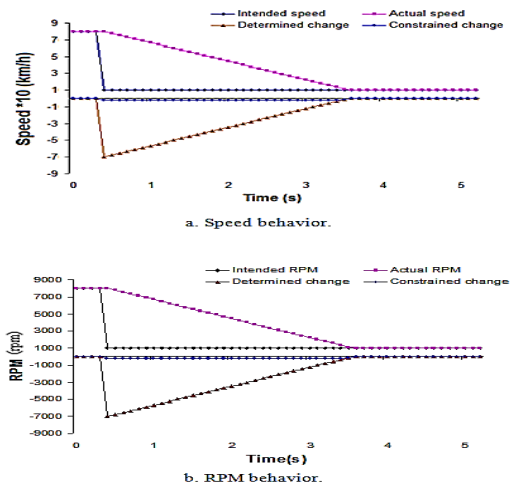


Fig.(12): Simulated speed and RPM at slowdown state.

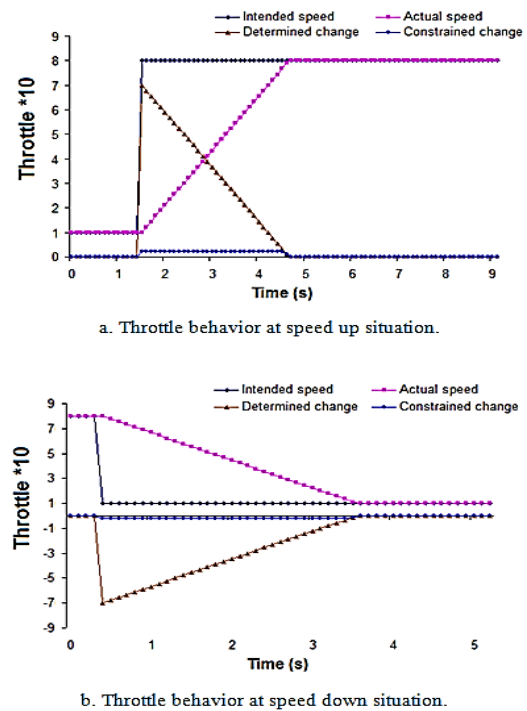


Fig.(13): Simulated throttle behaviors at different situations.

At intervals of head changing; the intended steering remained constant along all the correction intervals, while the actual steering changed periodically toward the desired value due to the continuous correction. The deflection of the steering had reached the maximum value at first correction intervals (after the switching instance) and it was decreased gradually.

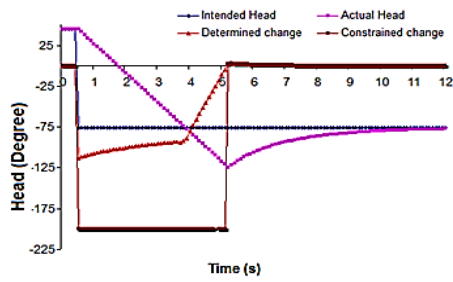
Since the intended turn is proportional to the deflection of the steering, its same behavior of change was similar to that for the deflection of the steering (i.e., it was gradually decreased). The actual turn that was in normal

state (i.e., its angle equal to zero) was continually raised toward the intended. The deflection of the turn earned large value at the first correction intervals and then it decreased smoothly due to the decrease in the deflection of the steering. The positive turn deflection had caused the steering to move continuously toward the positive direction. These changes in steering status caused gradual increase in the steering that computed by the simulator. The predicted deflection of the turn was straight line as well as the determined deflection of the turn was greater than the amount of the maximum allowed change in the turn, for such cases it was clipped to be equal to the maximum allowable value. This led to make the simulator actual turn increased by a rate proportional to the maximum change of the turn. The deflection of the determined turn remained straight until it reaches the case where the deflection of the turn became less than the maximum allowed change in the turn, and then it was gradually decayed at each correction interval.

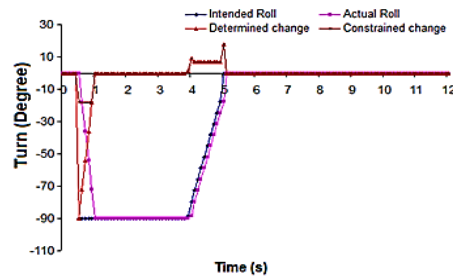
As a result, the deflection in the steering was decreased, the intended turn was decreased, and the actual turn was increased. At the same time, both the deflections in the steering and turn have been decreased sequentially due to continual correction in the steering. Also, the determined actual steering (by the simulator) was increased following the intended steering, till reaching the transient state. The transient state occurs when the actual turn value reach a maximum allowed value, in such cases the deflection of the turn becomes zero. This state remain until the intended turn becomes equal or close to the actual turn, in this state the actual turn shows an intentment to reverse its behavior.

After the transient point, the correction of the steering is still continuing, the intended steering remains fixed as it was, the actual steering remains continually increase, and the deflection of the steering has decreased till reaching the value zero. On the other hand, the decrease in the turn deflection continues and becomes a negative value because of the retreat of the intended turn from the actual turn, i.e. the intended turn becomes smaller than the actual one. The intervals characterized by the negative change of the turn led the

steering decrease back to its normal state, such state occurs when both the actual turn and the intended turn become zero. Also, one can notice that the change in the predicted deflection of the turn is little and it takes a long time to become zero. Figures (14 and 15) illustrate the corresponding behaviors of the guidance parameters due to the deflections that happened in the navigation parameters for different cases. The navigation parameters are also shown in these figures. An interesting behavior has been noticed; it is the uniform increase in the speed behavior, which is due to the uniform acceleration imposed by the simulator (car) during the simulation. In the following paragraphs the behaviors of both the navigation and guidance parameters (except the speed and RPM) are described for the two cases; before and after the reversion state. Before the transient state, the intended navigation parameter remained as it is along the time of corrections; the actual navigation parameter was gradually corrected toward the intended value. The changes in the navigation parameters behaved in inverse manner with the behavior of the actual navigation parameter. While the behaviors of the intended guidance parameters were similar to behaviors of the changes of the corresponding navigation parameters. The actual guidance parameters follow the intended, and the behavior of deflection of the guidance parameters (before the transient state) is linear as long as the determined deflection of the guidance parameters is greater than the amount of the maximum changes in the guidance parameters. The deflection in the guidance parameters decreases, and begins to decay slowly until it reaches zero at the transient state. After the transient point, the change in guidance parameter continues in same way. The actual guidance parameter changes its course of change from increasing/decreasing or decreasing increasing (i.e., reverse its behavior).

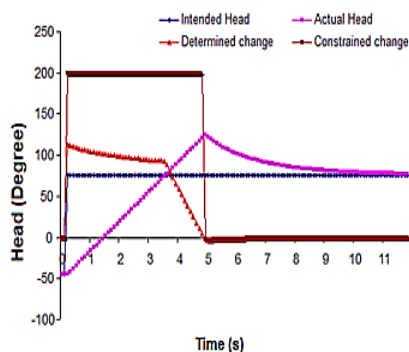


a. Head behavior.

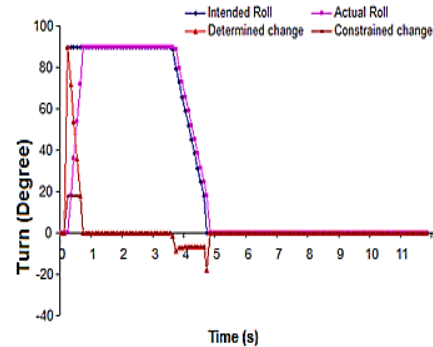


b. Turn behavior.

Fig.(14): Simulated head and turn at left turn state.



a. Head behavior.



b. Turn behavior.

Fig.(15): Simulated head and turn at right turn state.

Control Result Analysis

Fig.(16) describe the behavior of the guidance parameters and its effect on the navigation parameters when the response coefficients are taken to be equal to one. The behaviors become faster when the corresponding response coefficients are taken greater than one, and they are slower when the response coefficient is taken less than one.

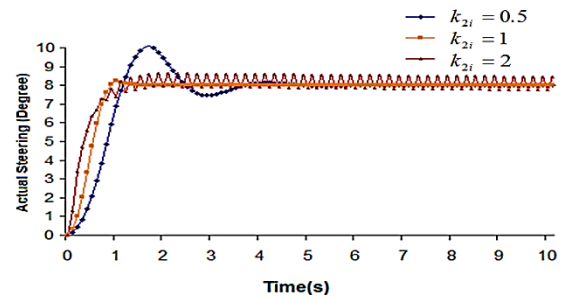


Fig.(16): Simulated actual steering behavior for different values of k_{2i} .

Obstacle Avoidance Result

In case when the vehicle faces moving obstacle, the created transient point may lies at the right or left of the obstacle depending on the vehicle and obstacle situations. These transient points are found in a distance is greater than the length of the obstacle by an amount is equal to the assumed safety distance. It is shown that the simulator was able to change its direction toward the transient point effectively. When the simulator sense the existence of obstacle ahead, the intended path is changed toward the transient point, and then the actual path that follow the intended one is newly directed toward the transient point. This required changing the head of the simulator by an angle is proportional to the amount of deflecting the intended path toward the transient point as shown in Figure (17 and 18)

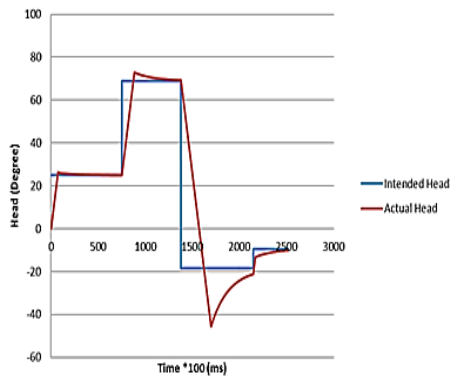


Fig.(17): Intended and actual head variation when switching with the transient point.

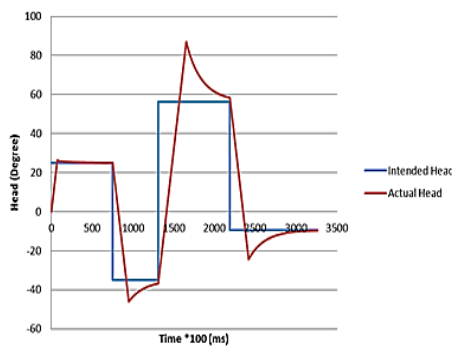


Fig.(18): Intended and actual head variation.

when back to switch with the next way point. When the simulator close the transient point, the simulator back to switch with the next way point. Such that, the simulator was shown back to identify the original intended path. It is shown that the behavior of the simulator from the previous way point to the transient point is identical to that of the transient point to the next way point, this refers to the stability of the simulator against the intended head variation. It is noticeable that the head correction when the simulator directed to the right direction consumes a time is longer than that of the left deflection. This is due to the position of the obstacle was closer to the simulator at the case of the right deflection. This indicates the correct behavior of the simulator against the faced obstacle, and proves the correct way of the assumption and solution of the present research.

Conclusions

The successful implementation of moving obstacle avoidance ensures the efficiency of the proposed method to guide the vehicle safely. More practices show that the best value

of the Fraction factor is 0.01, which is useful to control the dynamic response of the controller that providing least overshoot and more stability for the navigation and guidance parameters. The deflections earned large value at the first correction intervals and then it decreased smoothly. The behavior of the deflection is linear as long as the determined deflection is greater than the amount of the maximum changes, which is decreasing gradually until reaching a zero value at the transient state. The rate of change of the actual steering depends directly upon the actual speed. The overshoot usually occurs when a large deflection in the head of a turn associated with high speed car is needed to occur. The safety distance is proportional to the size of the faced obstacle. The simulator is able to change its direction toward the transient point effectively when facing obstacle ahead.

References

- [1] Dhuha Abdullah Amer, “Barrier Avoidance for Indoor Mobile Robot”, MSc. Thesis submitted to Computer Science department, College of Science, Baghdad University, Iraq, 2014.
- [2] Roland Siegwart, Illah R. Nourbakhsh, “Introduction Autonomous Mobile Robots”, A Bradford Book ,The MIT Press Cambridge, Massachusetts London, England, 2004.
- [3] Calin.G.Rusu, Iulian.T.Birou, “Obstacle Avoidance Fuzzy System for Mobile Robot with IR Sensors”, proceeding of the 10th International Conference on DEVELOPMENT AND APPLICATION SYSTEMS, Suceava, Romania, May 27-29, 2010.
- [4] Aye Aye Nwe, Wai Phyo Aung and Yin Mon Myint, “Software Implementation of Obstacle Detection and Avoidance System for Wheeled Mobile Robot”, World Academy of Science, Engineering and Technology 42, 2008.
- [5] Arturo L Rankin “Development of Path Tracking Software for an Autonomous Steered-Wheeled Robotic Vehicle and Its Trailer” UMI Microform 9801142 Copyright, United States Code, 1997.
- [6] 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.
- [7] Sachin Modi, “*Comparison of three obstacle avoidance methods for an autonomous guided vehicle*”, MSc. thesis submitted to the Division of Graduate Studies and Advanced Research of the University of Cincinnati, 2002.
- [8] Calin.G.Rusu, Iulian.T.Birou, “*Obstacle Avoidance Fuzzy System for Mobile Robot with IR Sensors*”, proceeding of the 10th International Conference on DEVELOPMENT AND APPLICATION SYSTEMS, Suceava, Romania, May 27-29, 2010.
- [9] Matteo Saveriano and Dongheui Lee, “*Distance based Dynamical System Modulation for Reactive Avoidance of Moving Obstacles*”, IEEE International Conference on Robotics & Automation (ICRA) Hong Kong Convention and Exhibition Center, Hong Kong, China, May 31 - June 7, 2014.
- [10] Jiechao Liu, Paramsothy Jayakumar, Jeffrey L. Stein, and Tulga Ersal, “*Combined Speed and Steering Control in High Speed Autonomous Ground Vehicles for Obstacle Avoidance Using Model Predictive Control*”, IEEE Transactions on Vehicular Technology algorithms, Citation information: DOI 10.1109/TVT. 2707076, 2017.
- [11] Hofmann-Wellenhof, Bernhard, Klaus Legat, and Manfred Wieser. Navigation: principles of positioning and guidance. Springer Science & Business Media, 2011.
- [12] Kleppner, Daniel, and Robert Kolenkow. An introduction to mechanics. Cambridge University Press, 2013.
- [13] Junkins, John L., and Hanspeter Schaub. Analytical mechanics of space systems. American Institute of Aeronautics and Astronautics, 2009.
- [14] Faraway, Julian J. Extending the linear model with R: generalized linear, mixed effects and nonparametric regression models. Vol. 124. CRC press, 2016.
- [15] Zhang, Hong, and James P. Ostrowski. “Visual motion planning for mobile robots”. IEEE Transactions on Robotics and Automation 18.2: 199-208, (2002).