IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 7.39 ⁜ Vol. 11, Issue 4, April 2022

DOI: 10.17148/IJARCCE.2022.114188

AUTOMATED GUIDED VEHICLE USING ARTIFICIAL INTELLIGENCE

Koli vaishnavi[1], Sheetal A. Wadhai[2]

Department of Computer Engineering, Universal College of Engineering and Research, Pune[1,2]

Abstract: The design of a mobile robot is a challenging task. A truly autonomous robot must be able to sense its environment and react appropriately. This issue becomes even more important if the environment is varying. One of the goals in robotics is to mdow robots with the ability to move and operate autonomously in an environment with unknown, perhaps moving obstacles.

Keywords: Information retrieval, Page ranking, Evaluation of information retrieval system.

1. INTRODUCTION:

Software is the soul of this research work. Without this is difficult to even imagine for the functionality of the system. The development of software tools as per the hardware requirement is the always desired with any software used for the control of the hardware. They work smilar to the device drivers for the overall system.

2. SOFTWARE SELECTIONS AND COMPATIBILITY ANALYSIS

Robotics research includes developing complex real-time software for testing theories and algorithms. Functions performed by the software include data acquisition, motor control, processing of sensor signals, perception of objects and navigation through the environment. The software can be written in any language, which supports low- level programming. The software selection depends on requirement for the operation. There may be three modes ⁃ of operation and as per that there are availability of the software as tabulated

3. DETAILS OF AVAILABLE SOFTWARE

SN	Mode of Operation	Software	Operating System
01	Stand alone	C, C++, Basic	MS-Dos (any version)
02	Network Based	C, Java, VB	Win 95 / Win –98
03	Internet Based	Java (Applet Servlet Communication)	Win 95 / Win 98

Developing the user interface in C, C++ is best suitable for standalone and network based operation. Looking to the dual feature provided by the language all the software has been developed in C language. The chronically order of development of software is detailed in the following section.

4. FLOWCHART AND ALGORITHM

Software in the heart of this systems develops Al in the symem. The total performan of the system depends on the sillective Suncang of the softwars. Various des were developed based on the Flow charts and algorithms so Falfill the objective of Aumated Guided Vehicle Using Artificial Intelligence. Their logis and
⁃ phase of development is detailed in the following sertion under various subheadings using Flow chasts and other necesary details.

5. PATH PLANNING

The main objective of this software is to present a working solution for Autonomous Path Finding System [59] to be implemented in a vehicle for its navigation. The Auto Controlled Ground Vehicle

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

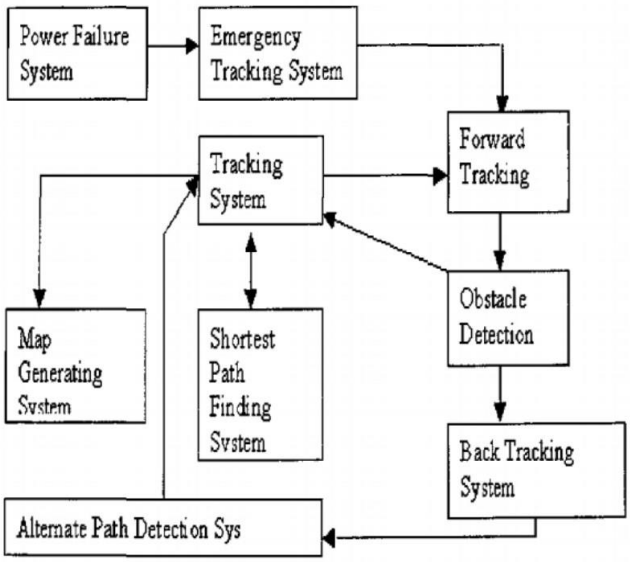
International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 7.39 ⁜ Vol. 11, Issue 4, April 2022

DOI: 10.17148/IJARCCE.2022.114188

• should operate in two modes, Path Recording & Path Tracking [60]. In Path Recording mode a human driver drives along the chosen path, recorded in the Computer memory. In the Path Tracking mode the computer assumes control over propulsion & steering. The vehicle then automatically travels for . the successful execution of the project along a memorized path according to the proposed plan.

The execution of the plan starts with the determination of the shortest path between the given source & destination using Dijkstra algorithm [61, 62]. There are several other algorithm available for it such as Floyd Warshall's algorithm [63] and Bellman Ford's algorithm [64] with their relative merits, But the Dijkstra is the best suited for the shortest Path determination. Tracking is done on the derived path decided by Dijkstra algorithm with the provision of planned/unplanned obstacle detection & avoidance. The logic of the software is depicted in the Flowchart and fig



FLOWCHART 3.1 PATH PLANNING AND NAVIGATION OF THE SYSTEM

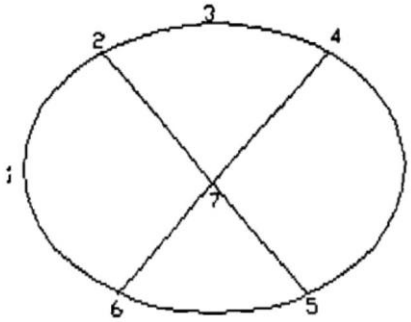


FIG. 3.1 TYPICAL MAP OF PATH TO TRAVERSE

Fig. shows the typical map of the proposed path. The map is implemented using bi-directional graph. The distance between any two node i.e. source and destination is termed as its weight / cost in between those two nodes. The non availability of path in-between any two nodes is achieved practically by increasing the nodal distance between those two nodes.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

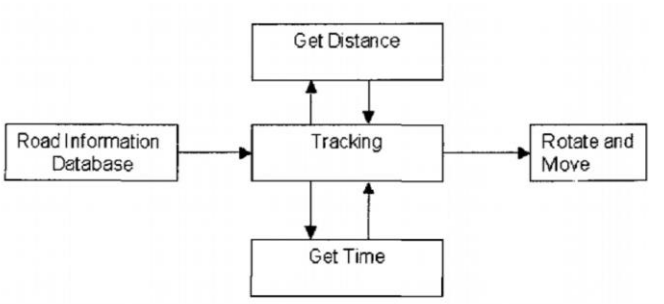
Impact Factor 7.39 ⁜ Vol. 11, Issue 4, April 2022

DOI: 10.17148/IJARCCE.2022.114188

6.AUTONOMOUS TRACKING USING ROAD INFORMATION

Based on the testing of the above software in the lab environment. Several facts have been observed and have been concluded to traverse safely in any environment. The knowledge of the environment is must. As, The total control is based on the PC. Once the vehicle is started nothing can be done except to Stop it in emergency. The Autonomous Tracking system [65] is based on it.

The only open problem, which stays in robot motion planning, is the trajectory planning during the motion in a curved path. To overcome it the overall map of the path is tried to break in smallest straight-line segments. The length of the straight segment depends upon the dimension of the vehicle smaller the dimension sharper the turns can be made for the same type of steering mechanism available in a given system. Thus the qualities of robotized processes depend on the solution of this problem. Through this software, an attempt is made towards the development of automatic trajectory planning using road information. A sophisticated robot was tested for this purpose having similar features of the on road vehicles. The different road conditions and the tracking result over this path are reported as shown in following Flowchart and fig.



FLOWCHART 3.2 ALGORITHM FOR TRACKING THE ROAD

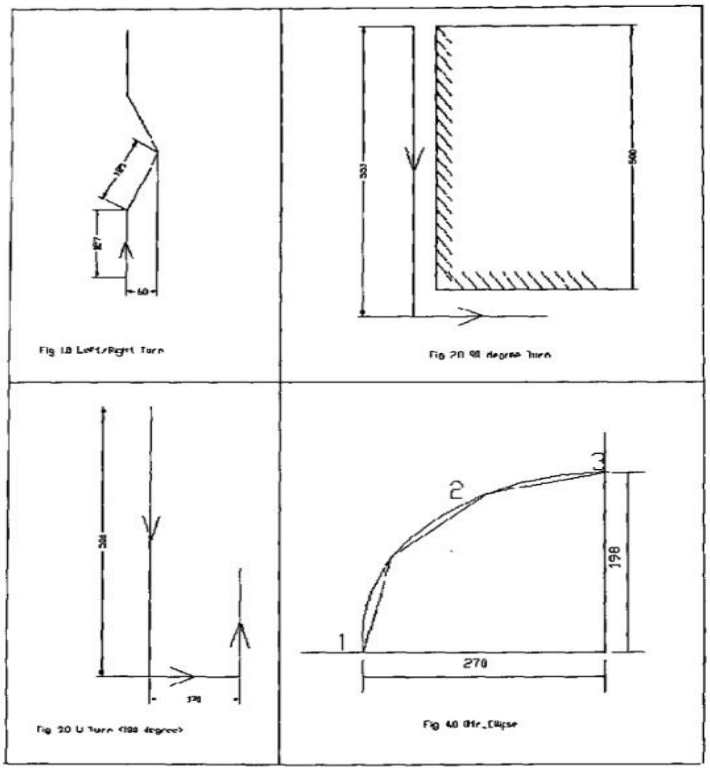


FIG 3.2 RESULTS OF AUTONOMOUS TRACKING USING ROAD INFORMATION

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 7.39 ⁜ Vol. 11, Issue 4, April 2022

DOI: 10.17148/IJARCCE.2022.114188

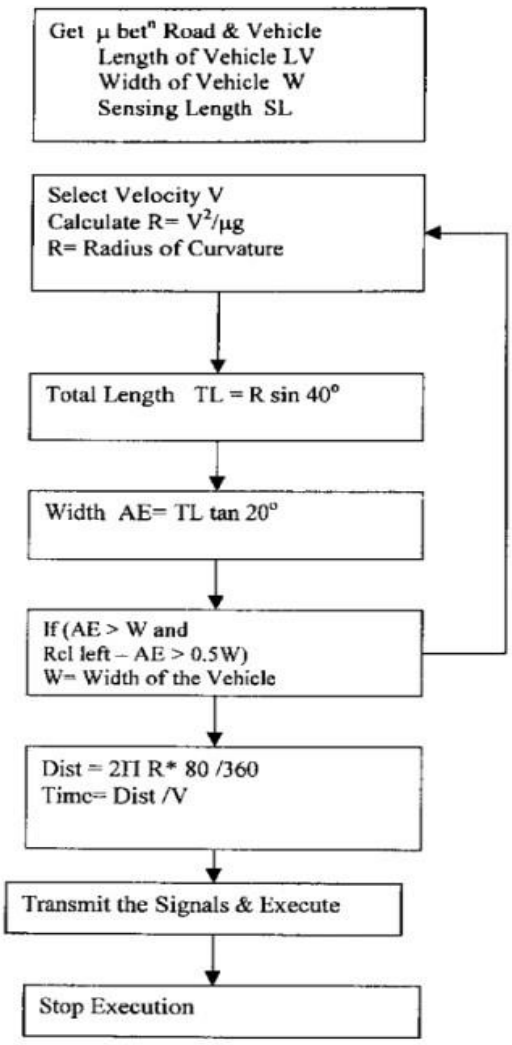
7.OBSTACLE AVOIDANCE

The mission of Autonomous Navigation cannot be achieved unless the vehicle has capability to detect, locate and avoid the static and dynamic obstacle safely during the navigation. During the obstacle avoidance the following parameter has to be controlled dynamically i.e. 1) Velocity, 2) Rate of Steering as well as location of the object and its shape has to be estimated.

Software has been developed to avoid the obstacle using 1) velocity control method [66] and other 2) Velocity and rate of steering both [67, 68].

Velocity Control Method

The logic of the software is shown in the following Flowchart 3.3 and 3.4. The two methods have been depicted there i) is used to select the safe value of velocity, ii) is used to calculate the safe value of velocity.
In the selection of safe value of velocity it is calculated whether the vehicle is capable to steer with this value of velocity without overturning when it follows a curved path during turning or overtaking.
In the second case, for a given conditions of the environment safe value of velocity is calculated and then rest of the operation is carried out. In both the cases necessary signals are generated to get the desired velocity of the vehicle.



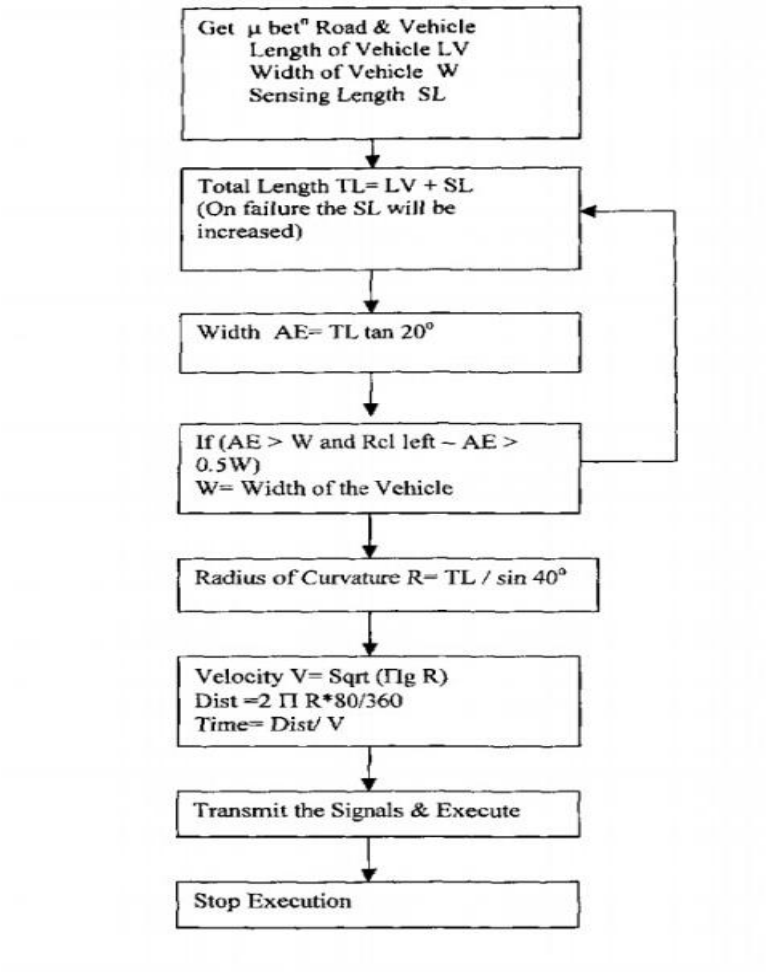
FLOWCHART 3.3 TO SELECT SAFE VELOCITY

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 7.39 ⁜ Vol. 11, Issue 4, April 2022

DOI: 10.17148/IJARCCE.2022.114188



FLOWCHART 3.4 TO CALCULATE THE SAFE VALUE OF VELOCITY

Velocity and Steering Rate both

For the obstacle avoidance control of velocity is alone not capable but at the same time it is also necessary to control the rate of steering also. The problem of steering control is not a simple problem. Because it is not simple geometric calculation based on the obstacle location. But during the steering the vehicle moves continuously towards the obstacle, therefore the problem has been discussed in depth.

Development of the Algorithm

The algorithm was developed based on the results of the simulated environment. In the very beginning the car like mobile was tested to avoid the static obstacle of width 10 cm as shown in Fig. 3.3a. The other parameter considered for the analysis is Rate of steering 10 deg/sec,
Vehicle velocity 10 cm/sec
Safe distance PQ = 28 cm, Diagonal Distance PR = 30.5 cm.
Total width RQ = 12.5 cm,
Time required to cover this distance 3.05sec. Theoretical angle of rotation 23.2 deg. Maximum steering of the vehicle within that time 30.5 deg.

Fig. 3.3a shows the simulated environment for the same. The curve A is the theoretical plot of the conditions but the B is the actual implementation of the same. Curve B shows that the vehicle is going to collide with the obstacle because it is not following the path A whereas it continues on path B, which is the actual trace of the vehicle movement during its travel. From this test it is clear that the vehicle has to steer with a larger angle as compared to the theoretically calculated angle. Because it has got the two motions altogether 1) Forward motion 2) Rotary motion. As a result of that the actual path followed by the system is B not that of A. Using this fact keeping the velocity and distance fixed if the vehicle has to be steered with larger angle within that time then the rate of steering will also be increased and possibly it may reach the destination.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 7.39 ⁜ Vol. 11, Issue 4, April 2022

DOI: 10.17148/IJARCCE.2022.114188

To analyze the problem with necessary detail an arbitrary case is studied as shown in Fig. 3.3b. in which the vehicle has to be steered by 50 deg. The curve A shows the theoretical path, whereas curve B, C and D shows the actual path traced by the vehicle at steering rate of 10, 15 and 20 deg /sec respectively.

Curve B is practical implementation of A, which is not acceptable in any case. For C and D, steering rate is 15 and 20 deg/sec respectively, but the result is not satisfactory. From these simulated results it can be ensured that the practical implementation of the theoretically calculated angle of turn will be more and its exact value has to be determined. Thus the question arises that what value of steering angle will satisfy the prevention of accident. For this, an algorithm is developed looking to the behavior of the above curves itself. During the calculation the Constraints that has to be considered are Horizontal and Vertical offset i.e. the vehicle should be steered till it crosses the obstacle width but before it reaches to the vertical limit otherwise it will collide with the obstacle.

To analyze the curve is broken into several small segments of straight lines representing each degree rotation of the vehicle as shown in Fig. 3.3c. The time required to rotate by single degree multiplied by the velocity gives the distance traveled within that period. From the geometry the horizontal offset (Of) and vertical offset (Of) can be calculated.

Mathematical Model Representation

The offset can be represented as

Of = V*t sin (0)

Where t is the time required to turn by 0 deg angle. Similarly

Of Of+V*t sin(20)

In this way if we go on summing till the total horizontal offset is reached we can get actual angle to be turned.

1.c.

Of hn Of hn-1 +Vt sin(ne)

With the similar calculation the vertical offset can also be calculated as

Of vn Of vn-1 +Vt cos(0)

8.RESULTS OF THE FINAL ANALYSIS

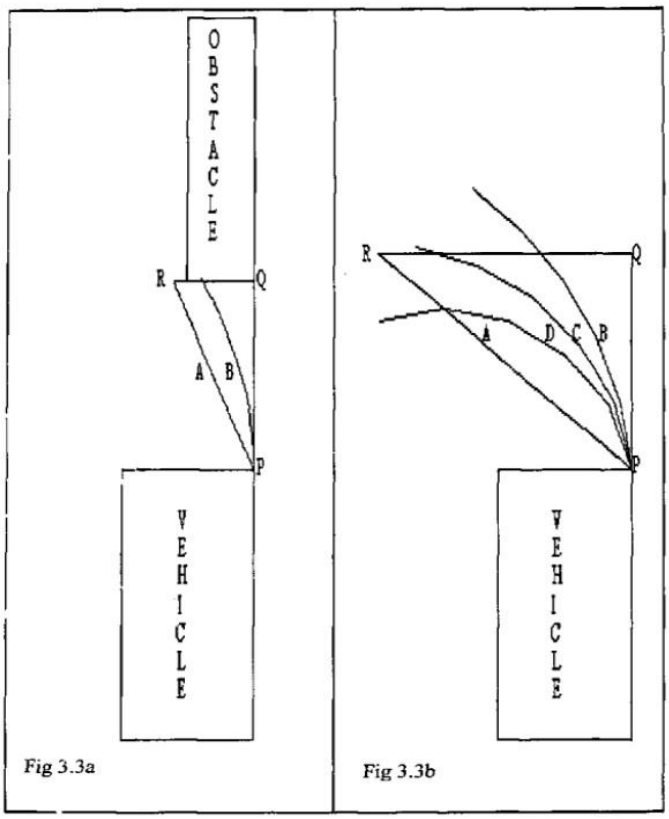
Sn	Obstacle Width, cm	Theoretical Angle, deg	Actual Angle, deg	End Offset, cm	Remarks
01	7.5	8.4°	38°	3.3	Net Deviation from actual path is zero in each case. And also the turn is safe.
02	12.5	13.8°	50°	3.9	
03	17.5	19°	60°	4.4	
04	22.5	23°	69°	4.8	

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 7.39 ⁜ Vol. 11, Issue 4, April 2022

DOI: 10.17148/IJARCCE.2022.114188



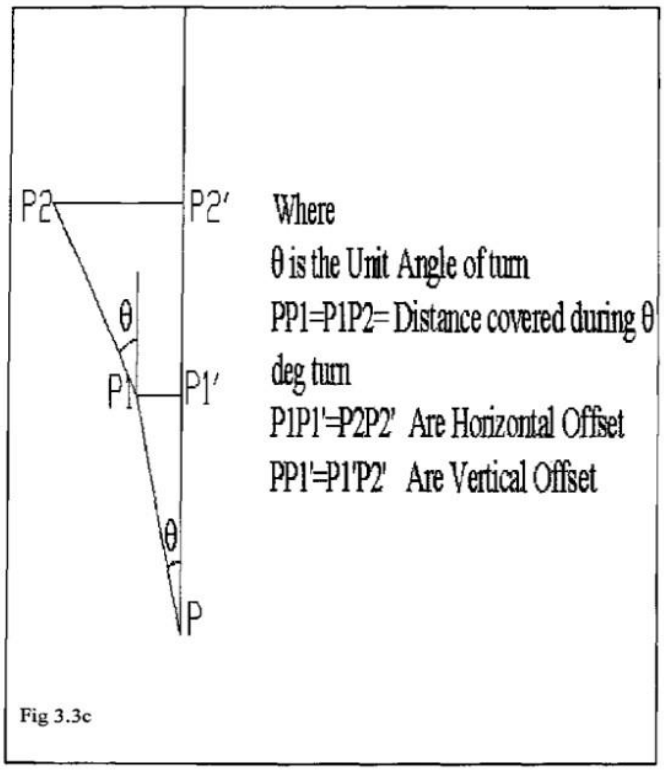


FIG. 3.3 ANALYSIS BASED ON THE CONTROL OF VELOCITY AND STEERING RATE

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 7.39 ⁜ Vol. 11, Issue 4, April 2022

DOI: 10.17148/IJARCCE.2022.114188

8.DEVELOPMENT OF SYSTEM FOR AGV

Passing through the above phases of development i.e. development of the sub systems in part finally all the systems are merged together and the final system has been developed for the AGV [69]. The logic of the integration and implement is shown in Fig.

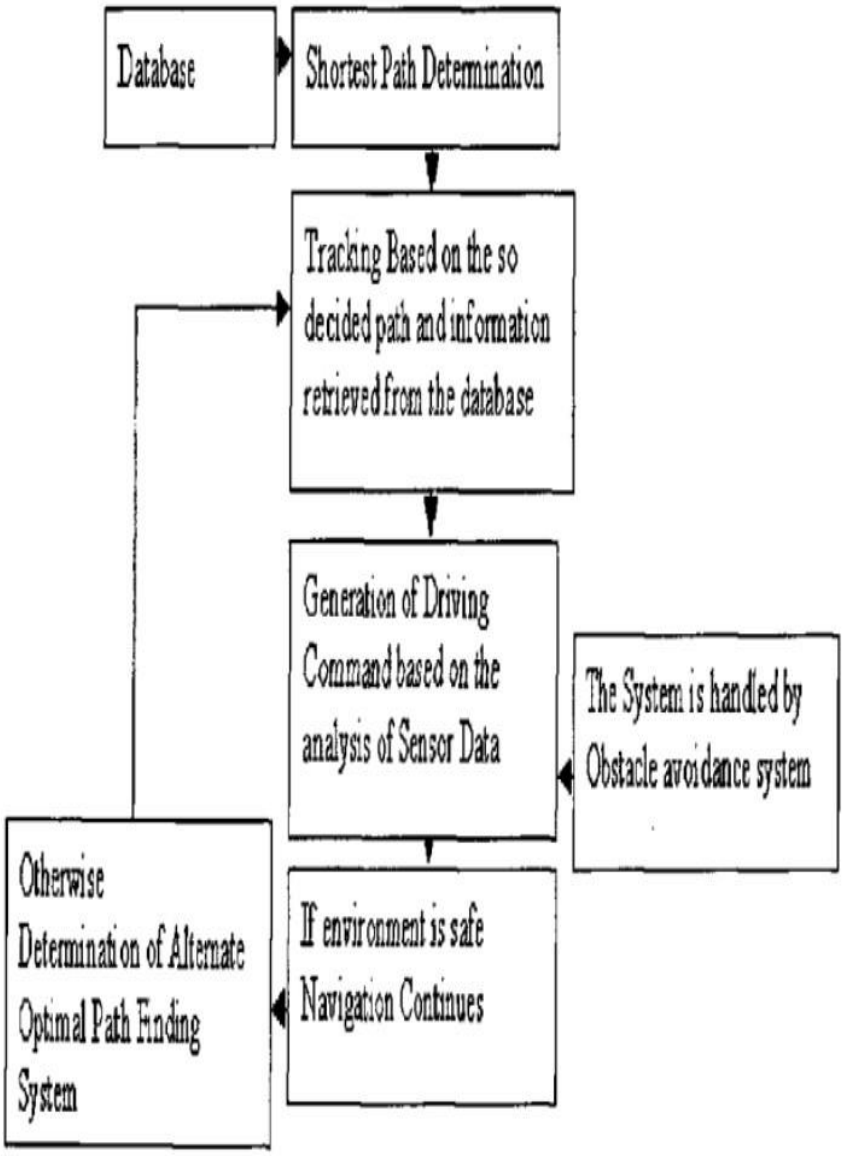


FIG. 3.5 ARCHITECTURE OF TRACING SYSTEM

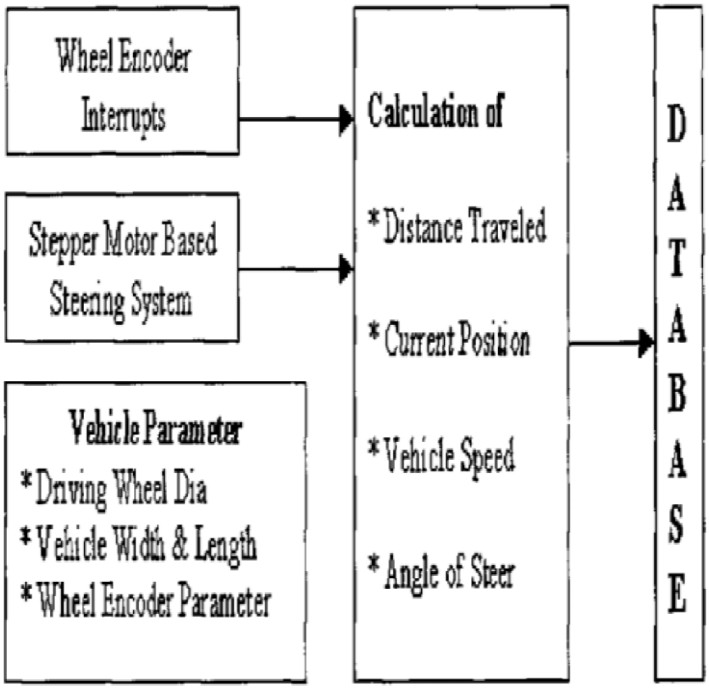
The development of the software is not limited to the above system. But there are several unlisted software has been developed which are essential for the component testing e.g. control of unipolar stepper motor through PC is one of those similarly during design and fabrication of the electronic hardware system and its tuning with mechanical system several software has been developed for on the spot testing of the system. The detailed executable files along with the catalogue and its purpose is available in the CD enclosed here with and the same can be referred through the appendix 5.0-23.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 7.39 ⁜ Vol. 11, Issue 4, April 2022

DOI: 10.17148/IJARCCE.2022.114188



Finally, a detailed software is developed based on the integration of the above components. It integrates the various functionality of 43 functions scattered over the thirteen header files. Development of the component in parts and its setting to work in a integrated way help in faster development and quick debugging. The details of the header files along with the functions and variables it support are tabulated in Table 3.3.

9.DETAILS OF THE FINAL SOFTWARE

SN	Name of Header File	Description
01	PIN_DEF	Has the definition of all the data pins used
02	VARIABLE	Has the definition and declaration of variables
03	HEADER	Has the declaration of all the header files
04	FN_DCL	Had the declaration of functions used in various files
05	INTRO	Has information regarding the software and preliminary status of the hardware. It has the following function definitions: 1) int Hardware_lock() 2) int Login() 3) int Init_Graph() 4) void Introduction() 5) void Data_Display() 6) void Count_Down().
06	OBSTACLE	Has obstacle avoidance related functions: 1) int Get_Obstacle(int steps) 2) int Detect_obstacle(int dir_opt, int steps, float start_angle, float end_angle) 3) int Overcome_obstacle(float cat_pos) 4) void Stepper_Mov(int dir_opt,int steps) 5) void TrRx_Test().

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 7.39 ⁜ Vol. 11, Issue 4, April 2022

DOI: 10.17148/IJARCCE.2022.114188

07	RELAY	Has all necessary function related to the working and testing of all the relays used in the system. The listing of functions are: 1) void Reset(int start, int end) 2) void OnFF(int FFno) 3) void OffFF(int FFno) 4) void Active_High(int start, int end) 5) void Cylcic_Process(int start, int end) 6)void Reverse_High(int start, int end).
08	GENERAL	Has general purpose supporting functions: 1) void Initialize() 2) void Exit() 3) void Show_Status() 4) void Print_Line(char c) 5) void Beep(void) 6) void Inprocess(void) 7) void Process_complete(void) 8) void Text_Print(char str[],int bkclr,int tclr) 9) void Msg(void) 10) void Delay_Setting() 11) void Extra() 12) void Headlight(int opt).
09	TRACTION	Has functions responsible for the traction of the vehicle listed as follows: 1) void Speed_Selector(int Speed_no) 2) void Scl_Mov_Direction(int dir) 3) void Show_cur_dir() 4) void Steer(int FF_n0) 5) void Fwd_Bak() 6) void Brake(int FFno) 7) void Fwd_motion() 8) void Bkd_Motion().
10	MOVE	It integrates the all the function necessary for the movement of the vehicle in independent way contains only one functions namely " void Mov_Vehicle() ".
11	CAL_STR	It calculates the necessary steering angle depending upon the location and geometry of the obstacle using the function " int Cal_Rot(int str_rate, float lr_cls, float safe_vel, int safe_dist, int obst_width, int obst_length)".
12	RECORDER	It contains functions required for the path recording and retracing of the same. The functions are: 1) void Path_Recording() 2) void Path_Following() 3) void Control_Display() 4) void Command_display().
13	Menu	Has necessary functionality to make the program menu driven using a function " int Get_Option().

10.REFERENCES

1 Agraw al Himanshu, Zadgaonkar Dr. A. S., "Autonomous Path Finding System", National Conference on Control, Communication and Information Systems (CCIS 2004), Goa College of Engineering, pp 68-70, January 2004.

2 Thomos Hellstrom, "Autonomous Navigation for Forest Machines", a Pre Study, Department of Comp Sc, Umea University, Sw eden, pp 3 5 .

3 R. S. Salarai, "Data Structure & Algorithm Theroy, Design & Implementaion Using C", Khanna Publication, New Delhi, 2002, pp 292-297.

4 http://www.ece.northwestern.edu/~guanghui/Transportation/spt/section3_ 1.html.