

A Design of the Obstacle Avoidance Software for Solar Autonomic Trolley

Dong Zhi-Ming^{*}, Guo Li-Xia and Song Le-Peng

School of Electrical & Information Engineering, Chongqing University of Science and Technology, Chongqing, 401331, P.R. China

Abstract: A four-wheeled autonomic trolley of obstacle avoidance is controlled by way of front-wheel steering and back-wheel driving. Three pairs of ultrasonic sensors are, respectively, installed on the left front, front and right front of the trolley in order to measure the distances between obstacles and the trolley. Its control strategy is derived from the concept of path planning for mobile robot based on fuzzy logic. The reference variables are the current location coordinates, X Y, and tp(the steering angle to target point), thus to find out the output steering angle (hp) of the front wheel.

Keywords: Autonomic trolley, Obstacle avoidance, Fuzzy logic.

1. INTRODUCTION

Global path planning and local obstacle avoidance are combined to highlight the mobile robot autonomy. Path planning aims to seek a required non-collision path between a given starting point and a destination point [1]. Practically, real-time sensors are used to determine the environmental conditions, where the sensors pass status information that is received by the processor. By using the pre-programmed instructions to generate control commands, the robot is enabled to move freely at the desired speed, the direction and the trajectory.

2. AVOIDANCE STRATEGIES

In this design, the reference on the obstacle avoidance for the trolley is based on the idea of the fuzzy logic of path planning for the mobile robot [2]. There are three pairs of ultrasonic sensors that are placed for measuring the distances to the obstacle in three directions (left front dL, front dF, right front dR).

2.1. Computing the Position and Orientation

For a mobile robot, determining its position and orientation requires to find: the heading angle θ , the angular direction tp of the robot to the destination and the steering angle hp. The key for the trolley to avoid obstacles is to get the output angle hp of the front wheel. The conventional avoidance method is basically based on the calculation to track the curves or traces, thus enabling the trolley to avoid obstacles. Comparatively, the advantage of the proposed path planning idea based on fuzzy logic is derived from certain artificial experience

[3], which requires no mathematical calculations. Rather, the fuzzy control rules established [4] have the reference variables that are taken as distances to the obstacles in three directions and the steering angle of the trolley to the target point tp, leading to the output of angle hp.

Accordingly, assuming at the initial moment: $\theta = 0$, $tp = 0$, $hp = 0$. That is, the mobile robot may have its heading direction initially placed as the direction toward the target point. In the process of walking, the mobile robot may have a deflection in order to avoid obstacles, hence θ , tp , hp all will change. It is specified:

① If the target point is right in front of the robot: $tp > 0$, otherwise $tp < 0$;

② If the robot turns right: $hp > 0$, left $hp < 0$;

Then, the mobile robot is given a steering angle relative to the coordinate system as the algebraic sum of a number of steering angles, that is

$$h_p = \sum_{i=1}^n h_{pi} \quad (1)$$

Set the target point on a straight line along which the mobile robot walks for a distance L at the initial direction, then its position and orientation follows the computing diagram as shown in Fig. (1).

From Fig. (1), it is calculated as:

$$X = \sum_{i=1}^n V_i \sinh_{pi} t_i \quad (2)$$

$$Y = \sum_{i=1}^n V_i \cosh_{pi} t_i \quad (3)$$

$$t_p = \arctan \frac{X}{L - Y} \quad (4)$$

^{*}Address correspondence to this author at the School of Electrical & Information Engineering, Chongqing University of Science and Technology, Chongqing, 401331, P.R China; Tel: 023-65023987
E-mail: zhmdong@cqust.edu.cn

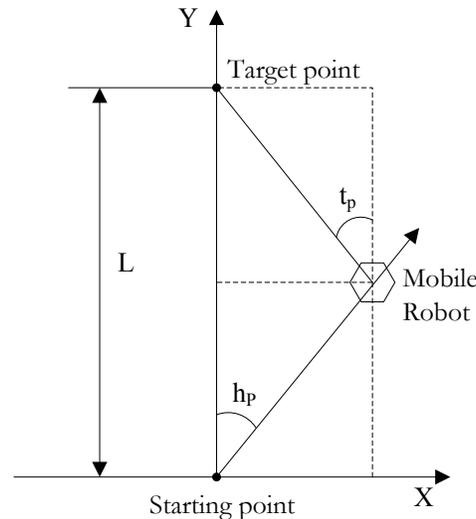


Fig. (1). Schematic for Computing the Position and Orientation of the Mobile Robot.

Therefore, X , Y , tp may be used to determine the current location of the mobile robot, and the direction to walk in the next step.

2.2. Avoidance Realization

□ Definition of fuzzy linguistic variables of dL , dF , dR , tp , hp .

For dL , dF , dR , their fuzzy language variables are $\{TNear, Near, Far\} = \{\text{"too near"}, \text{"near"}, \text{"far"}\}$, with the domain $0 \sim 6m$; the fuzzy language variable for the direction angle tp is set as $\{LB, LS, Z, RS, RB\} = \{\text{"left-big"}, \text{"left-middle"}, \text{"zero"}, \text{"right-middle"}, \text{"right-big"}\}$, with the domain $(-180^\circ, 180^\circ)$; hp as the output is given its fuzzy variable language as $\{TLB, TLS, TZ, TRS, TRB\} = \{\text{"turn-left-big"}, \text{"turn-left-small"}, \text{"zero"}, \text{"turn-right-small"}, \text{"turn-right-big"}\}$, with the domain $(-30^\circ, 30^\circ)$.

□ Build-up of knowledge regular library

The path planning for obstacle avoidance of a mobile robot is based on the sensor-acquired dL , dR , dF data, combined with tp , hp .

In the environment where the robot is away from obstacles or obstacles do not exist, the following five rules are established:

R1 : If dL is Far, dF is Far, dR is Far, tp is LB, then hp is TLB ;

R2 : If dL is Far, dF is Far, dR is Far, tp is LS, then hp is TLS ;

R3 : If dL is Far, dF is Far, dR is Far, tp is Z, then hp is TZ ;

R4 : If dL is Far, dF is Far, dR is Far, tp is RS, then hp is TRS ;

R5 : If dL is Far, dF is Far, dR is Far, tp is RB, then hp is TRB ;

As the robot detected obstacles in the left front, the following five rules may be established:

R6 : If dL is Near, dF is Far, dR is Far, tp is LB, then hp is TZ ;

R7 : If dL is Near, dF is Far, dR is Far, tp is LS, then hp is TZ ;

R8 : If dL is Near, dF is Far, dR is Far, tp is Z, then hp is TZ ;

R9 : If dL is Near, dF is Far, dR is Far, tp is RS, then hp is TRS ;

R10 : If dL is Near, dF is Far, dR is Far, tp is RB, then hp is TRB ;

Similarly, control rules may be established to cover a variety of other simple or complex distribution of obstacles, such as the case of an obstacle present in the right front, or the obstacles found in both left front and right front.

In actual programming, the range of values is actually considered in terms of each fuzzy language, thus being able to achieve the avoidance goal.

In this work, the fuzzy languages had their actually derived values as shown in Table 1.

Note that this design is not strictly subject to fuzzy control, for no fuzzy decision, anti-blur and some other operations are made. Rather, hp is simply attained by means of control rules.

3. SOFTWARE DESIGN

The main program has a flow chart as shown in Fig. (2).

Subroutine for the control rules is mainly designed to get the steering angle of the front wheel, and the purpose of backward motion is to make sure that the trolley is able to

Table 1. Actually derived values of fuzzy languages.

tp	LB	LS	Z	RS	RB
	(-90, -45)	(-45, 0)	0	(0, 45)	(45, 90)
hp	TLB	TLS	TZ	TRS	TRB
	-30	-15	0	15	30
dL, dF, dR	TNear		Near		Far
	(0, 30)		(30, 80)		(80, +∞)

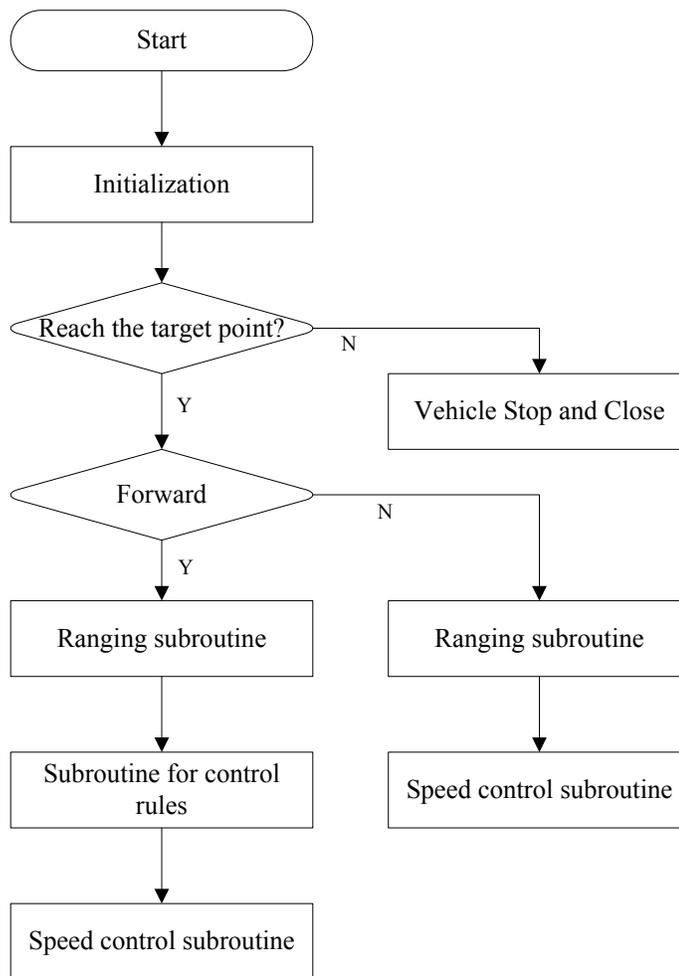


Fig. (2). Flow Chart of the Main Program.

retreat to a safe distance. So the steering angle of the front wheel is no longer involved in the retreat process.

3.1. Ranging Subroutine

This is so called an assignment subroutine, which works on such an idea: If an interruption occurs in the corresponding periods, then the actual measured value is assigned. Otherwise, a larger specific value is assigned [5, 6].

Ranging subroutine is mainly used to get the distances to obstacles on three directions, providing a reference volume for the trolley to avoid obstacles. Ultrasonic detection is circuited in three directions for measurements, with the flow chart as shown in Fig. (3).

After opening the ultrasonic, 20us latency is needed before the receiving device works. This consideration is to prevent the ultrasonic waves received that are mistaken as the

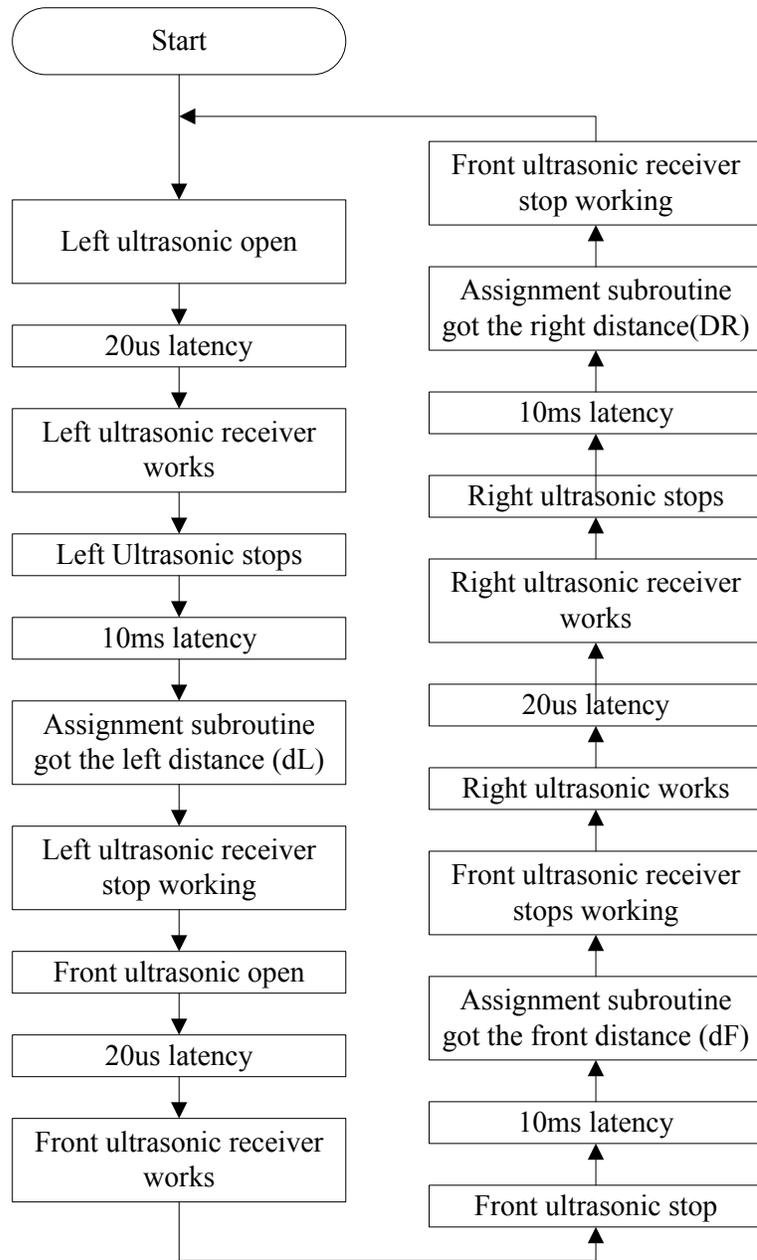


Fig. (3). Ranging Subroutine.

returned ultrasonic waves from obstacles. Ultrasonic transmitter would stop in every several signals. 10ms ultrasonic wave can be transmitted in a distance of $340 \times 100 \times 10 \times 10 = 3400000$ cm = 34000m. That is, an obstacle in a distance of 170cm from the robot is measured. If no interruption occurred during this time, then the obstacle may be considered to be far away. After getting the distances in each direction, the ultrasonic receiving in that direction is ceased.

3.2. Subroutine for Control Rules

This subroutine mainly aims to obtain the output angle hp of the front wheel according to the distance variables in three

directions and the steering angle to the target point. According to the control rule, a simple programming can be realized using the if statements. Provided such variables and the steering angle are both able to satisfy certain conditions, it is possible to output a corresponding front wheel angle hp [7]. Subsequently, the steering control subroutine is called in order to control the motor.

```

void controlrule( )/* Control rules generated upon
fuzzy control */
{ if (dl>80 && df>80 && dr>80) // No obstacles in all
directions
    
```

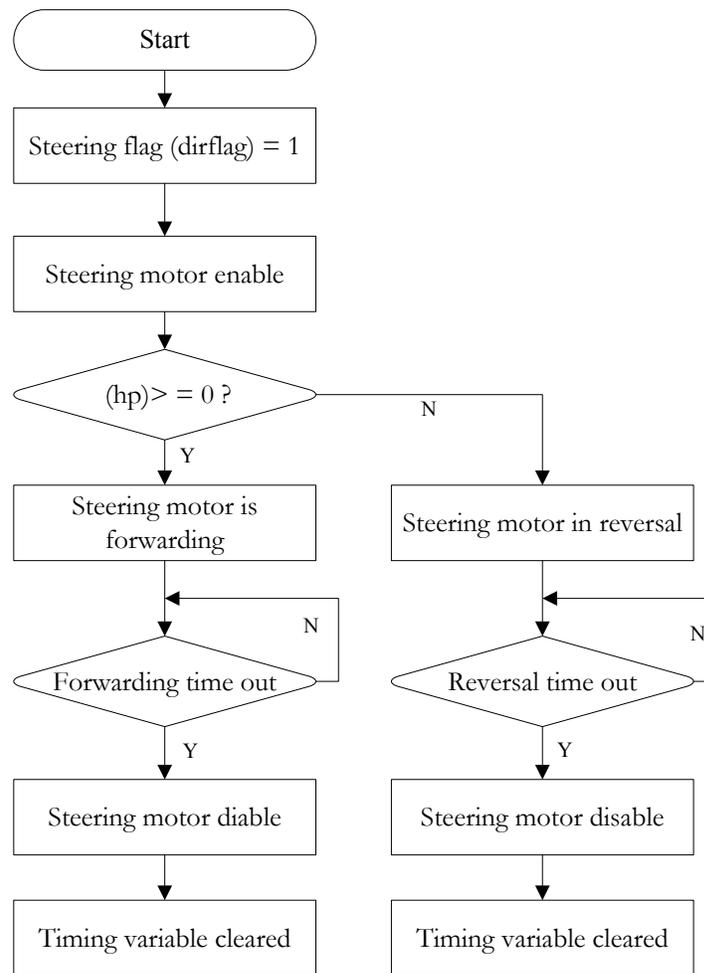


Fig. (4). Flow Chart of the Steering Control Subroutine.

```

{ if(-90<tp<=-45)
hp=-30;
if(-45<tp<0)
hp=-15;
dircontrol();
}
}

```

3.3. Subroutine for Steering Control

The steering angle hp of the front wheel can be obtained through the subroutine control rules, while the role of the steering control is how to control the motor to achieve the output of the hp value. Motor speed is nr/min , and its rotation time is $(60/n)$ s. The rotation time required for the motor to get the angle of rotation hp can be determined by observing the proportion of the output angle hp accounting for the part of 360° , where the time value is displayed as the timing variable. As the motor starts to rotate, the timing action starts. By the time when the timing condition on the time variable

is met, the motor is stopped immediately. The program has a flow chart as shown in Fig. (4).

3.4. Subroutine for Speed Control

By switching the high and low level at the enabling end of the motor, the duty cycle may be obtained to control the speed of the trolley. According to the rotational speed and the radius of the wheel, the robot speed is figured out to get the forward and backward speeds in opposite directions. Forward speed is positive, and backward speed is negative. The program flow chart is shown in Fig. (5).

The product of velocity and the corresponding time provides the displacement of the trolley. The speed may either be positive or negative, so is the displacement. Thus, the algebraic value of the trolley displacement is determined. The calculated displacement components in the X and Y directions lead to the current position of the trolley.

When the trolley is closer to the obstacle, deceleration occurs. Being far away from the obstacle, the trolley accelerates or maintains the original fast speed. In a certain direction, where it is too close to an obstacle, the trolley moves in

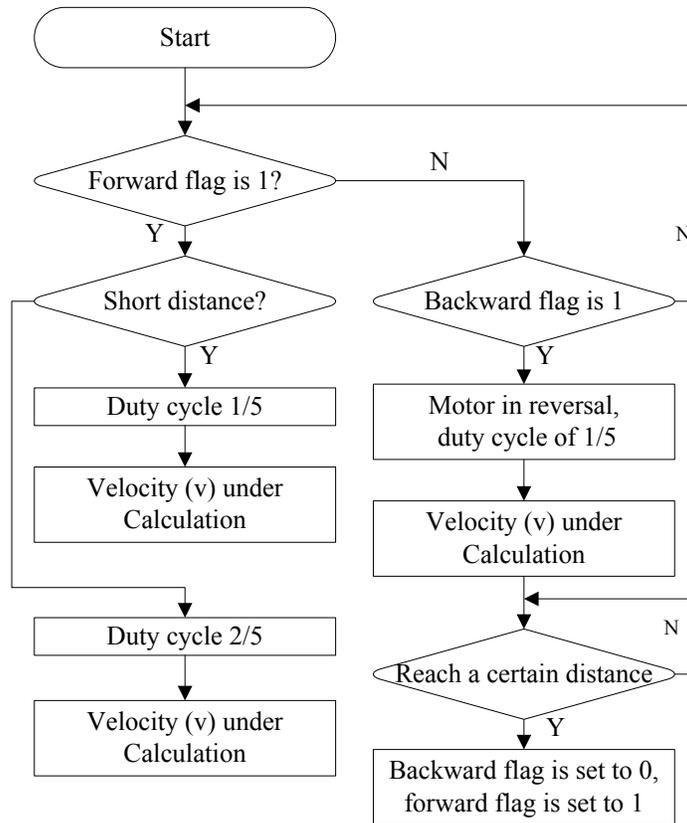


Fig. (5). Flow Chart of the Speed Control Subroutine.

a smaller reverse speed. After a certain safe distance is retreated, the trolley continues to move forward.

3.5 External Interrupt Program

This program is only taken to calculate the distance between obstacles and the trolley. After getting the readout value, the timer is cleared. Subsequently, the timer value is multiplied by ultrasonic velocity to obtain the distance to the trolley. Whenever there is an interruption, calculation is given thereon.

Ultrasonic wave has its propagation velocity that is affected by temperature, so not taken into account in this design. To be precise, the speed may be added with a temperature adjusting module.

3.6 Timer Interrupt Routine

This is a routine that is mainly set to calculate the trolley's current location. It works on the basis of calculus, where the motion of the trolley is divided into a plurality of small processes. In such a short time period, the speed may be considered constant, thus calculating the travel distance of each segment and its components in the X and Y directions. The result from each calculation is added to the last value, and you can get the current position of the robot. Based on the target position and the present position, the steering angle from the trolley to the target point is then obtained.

Additionally, the timer interrupt program is also responsible for processing some timing variables for timing, calling the display subroutine and other operations.

3.7 Display Subroutine

This routine is used to display the distances to an obstacle from three directions alternatively, each display for 1s. In the case of being too close or closer, the actual distance is displayed; when the distance is too far, the symbol Far is displayed.

CONCLUSION

In this work, a more detailed design is given to the software for the self-avoidance trolley. Depending on different speed values, the trolley is able to get its current location information. Then following the fuzzy control rule, the front-wheel output angle is straight-forwardly found, thus being able to avoid obstacles. To add it, the data on the distance to the obstacle can be visually checked in three directions. This design is proven to be effective for the trolley in self-avoidance of obstacles.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

Declared none.

REFERENCES

- [1] X. Zhang, "Robot Technology & Application," *Research on the Terrestrial Mobile Robotics: Status and Trend*, no. 2, pp. 19-23, 2005.
- [2] M. Shen, "A Study of Wheeled Mobile Robots on Navigation Control and Path Planning," *Northwestern Polytechnical University*, pp. 32-33, 2006.
- [3] Y. Song, "Studies of Mobile Robots on the Path Planning Method based on Recurrent Neural Networks," *Shandong University*, (In Chinese), pp. 43-44, 2008.
- [4] J. Liu, *Intelligent Control*, Beijing: Science Press, pp. 118-120, 2005.
- [5] Y. Zhu, "The path planning algorithm based on ultrasonic sensors ranging for the edgewise going cleaner," *Robotics and Applications*, no. 2, pp. 16-18, 2002.
- [6] Y. Peng, "A design of the mobile robot control system based on AT89S52," *Journal of the Nanchang Institute of Aeronautical Technology*, vol. 20, no. 12, pp. 67-69, 2006.
- [7] S. Zhang, "4-wheel universal mobile robot: motion analysis and motion control," In: *Chinese Association of Automation East China 17th Annual Conference Proceedings*, pp. 446-452, 2004.

Received: September 16, 2014

Revised: December 23, 2014

Accepted: December 31, 2014

© Zhi-Ming et al.; Licensee Bentham Open.

This is an open access article licensed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/4.0/>) which permits unrestricted, non-commercial use, distribution and reproduction in any medium, provided the work is properly cited.