Obstacle Avoidance System using 2D Lidar for Autonomous Vehicle

Danai Phaoharuhansa[1,a)] Natasit Supperm[1] Kunanont Raksawid[1] Chanin Kampeewichean[1]

Abstract: This paper aims to develop autonomous driving vehicle based on robot operating system (ROS). ROS nodes are defined relative to model-based control and each node computes as the same period. For control system based on ROS, the structure of the feedback loop is designed based on the block diagram, which is not convenient for the complex system. The conceptual design of SysML may be more useful than block diagram. The feedback control with obstacle avoidance system is extracted to subsystems such as obstacle avoidance system, trajectory tracking, and motion control system. The communication between the subsystem is defined as subscription and publication in ROS. MATLAB/Simulink and Arduino IDE are used to develop ROS code. The obstacle avoidance node and trajectory tracking nodes subscribed the necessary data and they publish the desired velocity and the desired steering angle. The motion control node consists of velocity control and steering control, which are programmed to Arduino board and communicate with other nodes through UART communication. Gaussian potential function is applied to detect the obstacle and define the direction to avoid the obstacle. The prototype vehicle is developed as small vehicle and it is used to perform by the algorithms. The experiment result presents that the vehicle can detect the obstacle and avoid the obstacle successfully. The maximum of detection range is varied to observe the behavior and the steering angle is bounded +/- 20 degrees. As summary, the avoidance performance depends on the radius of detection, steering mechanisms, and communication speed between the nodes, which effects to sampling rate of control system.

Keywords: Obstacle avoidance system, Robot operating system, ROS node, MATLAB/Simulink, SysML

1. Introduction

Robot operating system (ROS) is one of open source platform for develop mobile robot and autonomous vehicle[1], [2], [3], which is familiar with computer engineering and it may not concern about dynamic of vehicle. The comparison of control system based on ROS [1], [2], [3] and conventional control system [4], [5], [6], [7], [8] is that ROS consists of several nodes and each node performs only one task. The published data of each node is shared in the core and the data will be used by data subscription. Then, the operation of each node is parallel computing with others.

By the way, conventional control system is well known as feedback control system and it is usually constructed based on block diagram so that the operation will perform as serial computing process that sensing, computing, and acting, respectively.

Then, autonomous vehicle system becomes complex system and subsystems are performed in the same time. control system based on ROS may be useful than conventional control system. The processor should be organized the resource and it should concern about the sampling rate of sensor. The sampling rate is very important for conventional control as presented in the paper [9] because the sampling rate of sensor will response the precision of feedback control too.

To design the control system, System Modeling Language (SysML) [10] is applied to separate the task of the control system and the communication between the nodes is defined relative to the sampling rate. Therefore, this paper will introduce the obstacle avoidance system, which constructs based on ROS by MATLAB/Simulink and Arduino IDE. ROS nodes are designed relative to the modeling language, and then the system is demonstrates in experiment.

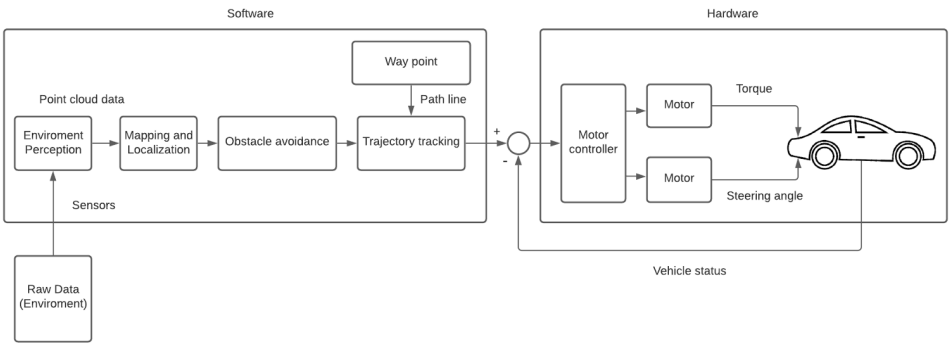
Fig. 1: Prototype vehicle is used in experiment.

2. Control System

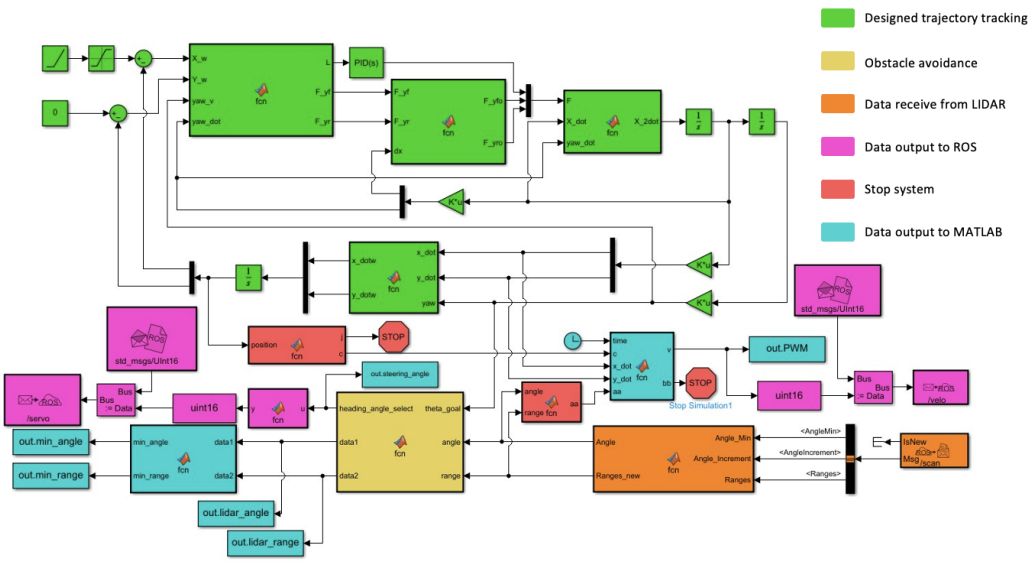
The prototype vehicle is developed as shown in Figure 1. It contains mini computer, embedded board, motor controller for steering and powertrain mechanisms. 2D-Lidar is used to detect the obstacle, range of which is less than 8 meters.

The overall control system is designed as feedback control system and the system is separated to two parts as shown in Figure

[1] Department of Mechanical Engineering, King Mongkut's University of Technology Thonburi, Bangkok, 10140, Thailand
[a)] danai.pha@kmutt.ac.th



(a) Block diagram of control system.



(b) ROS node design in MATLAB/Simulink.

Fig. 2: Block diagram of control system is represented by ROS node in MATLAB/Simulink

2a. Software part includes the mapping and localization, obstacle avoidance system, and trajectory tracking system. 2D Lidar and IMU nodes provide raw data such as point cloud of environment detection and yaw of vehicle, respectively. Obstacle avoidance aims to revise the trajectory to goal position for avoiding the obstacle if the obstacle is detected. It received data from every node in order to know the status of vehicle, environment, and obstacle. Then, desired speed (v_d) and new desired direction (φ) are compute and they are published to hardware.

Hardware part obtains as vehicle, which includes the position controller for steering system and the speed controller for powertrain system. Then, the hardware requires the desired condition from software part such as speed and steering angle.

To redesign the control system, the paper [10] suggested to design the system based on task and operation so that ROS nodes are organized as Figure 3 that the system includes 5 nodes such as way point node, obstacle avoidance node, vehicle node, 2D Lidar node, and IMU node.

2D Lidar node and IMU node are grouped as sensor nodes, which measure raw data. The way point node, the obstacle avoidance node are constructed by MATLAB/Simulink as shown in Figure 2b, Simulink model is deployed to ROS code for operation. ROS message is defined to subscribe data from sensor nodes and publish desire condition to vehicle node by Rosserial. Vehicle node contains the motion control system for speed and steering control.

2.1 Hardware

As above sentences, the system requires a computer, a vehicle, 2D lidar, and IMU sensor. Specification of computer is that Intel core i5 processor, 8 GB of Ram, and SSD 256GB. ROS Noetic distribution and MATLAB/Simulink 2021 are installed in the computer and connect to hardware by USB interface and UART communication. 2D lidar is YDlidar X4, the detection range is from 0.12 to 10 meters and scan frequency is fixed at 12 Hz. It communicates with computer at 115,200 bit/sec. IMU sensor is MPU9250 and the data output frequency is up to 500 Hz. Vehicle is controlled by ATmega2560, which with computer at 115,200 bit/sec.

2.2 Obstacle Detection Algorithms

To detect the obstacle and define the position of obstacle, 2D Lidar is used to survey the area as shown in Figure 4. Lidar transmits the laser around the sensor and it will measure distance from

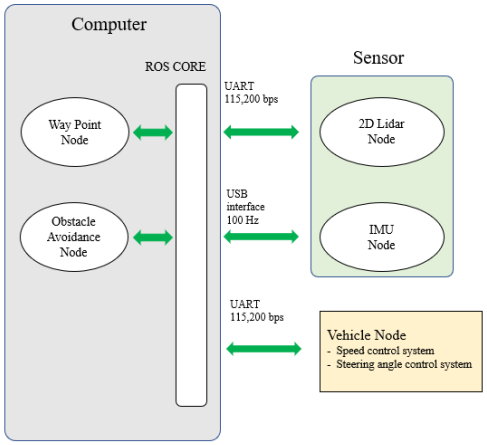


Fig. 3: ROS node design for demonstration system.

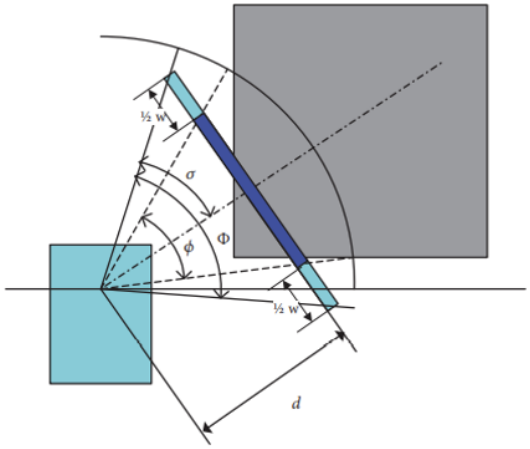


Fig. 4: Obstacle detection using 2D Lidar.

sensor. The data will be sent as array of detection angle (θ) and distance (d) and the size of array is $k \times 2$, which can be used to calculate the position of obstacle ($P_{o,x}$, $P_{o,y}$) in XY coordinate that

$$P_{o,x} = d_{\theta,k} cos(\theta_k) \tag{1}$$

$$P_{o,y} = d_{\theta,k} sin(\theta_k) \tag{2}$$

where, $d_{\theta,i}$ denotes distance from sensor to object at θ_k angle. θ_k is angle of 2D Lidar at k step.

The angle of obstacle width (ϕ) denotes the range of laser scan, which detect the obstacle is defined as

$$\phi = \theta_{max} - \theta_{min} \tag{3}$$

Then, the direction from Lidar to obstacle is given as

$$\phi_{ob} = \frac{\theta_{max} - \theta_{min}}{2} \tag{4}$$

where, θ_{min}, θ_{max} denote the minimum and maximum scanning angle of Lidar, which found the obstacle.

To compensate the width of vehicle during the steering, the angle of obstacle width will be included the width of vehicle (W_v) that

$$\Phi = 2\text{atan}\frac{d_{av}\tan(\frac{\phi}{2}) + \frac{W_v}{2}}{d_{av}} \tag{5}$$

where, d_{av} denotes average distance from Lidar to obstacle.

2.3 Obstacle Avoidance Algorithms

To design the obstacle avoidance system, Gaussian potential field is used to compute the avoidance direction, which is presented by the paper [6]. The authors developed and simulated the motion in 2D planar. The field equation is summation of repulsive field equation (f_{re}) and attractive field equation (f_{at}) and the minimum of total field equation indicates the avoidance angle, which is send to vehicle to perform the steering angle. It is given as

$$f_{to} = f_{re}(\phi_{ob}, \theta, \Phi) + f_{at}(\theta_g, \theta) \tag{6}$$

The repulsive field equation is given as

$$f_{re} = \sum_{k=1}^{n} A_k \exp(-\frac{(\phi_{ob} - \theta_k)^2}{0.5\Phi^2}) \tag{7}$$

$$A_k = (d_{max} - d_{av})\exp(\frac{1}{2}) \tag{8}$$

and the attractive field equation is expressed as

$$f_{at} = \varphi|\theta_g - \theta_k| \tag{9}$$

where, k is step of sampling data of Lidar scan and n is maximum among all lines of scanning laser. A_k is repulsive coefficient. θ_g denotes the angle of direction from Lidar to goal. φ is yaw angle.

3. Experiment Result

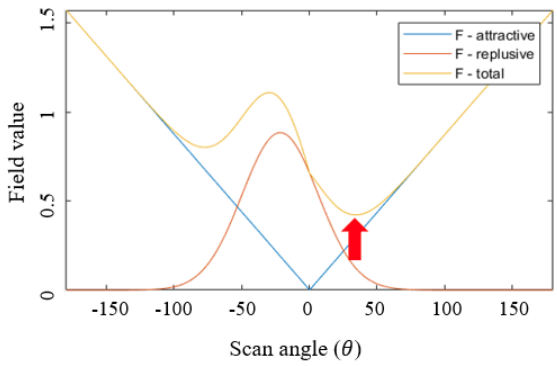
In this study, the speed of vehicle is fixed as constant speed at 0.5 m/s. The vehicle starts at initial position and the obstacle is in front of the vehicle at 2.50 meters as shown in Figure 5. The original trajectory is from (0,0) to (4,0). It is revised by obstacle avoidance system to avoid the obstacle by setting the detection length.

After the raw data from 2D lidar is input to ROS node in computer, the input frequency is reduced to 10-11 Hz by communication. Then, obstacle avoidance will calculate the total field equation as shown in Figure 5a, it presents the field values versus scanning angle (θ) that red and blue lines present repulsive and attractive values from Equation (7) and (9), respectively. The summation of both values is total field value, which is shown as yellow line. The minimum of the total field value means to avoidance angle, which is pointed by the red arrow. It will be published to the vehicle by UART communication at 115,200 bit/sec as shown in Figure 5b. The frequency of transmitting data is 0.15 Hz or 6.42 times per sec.

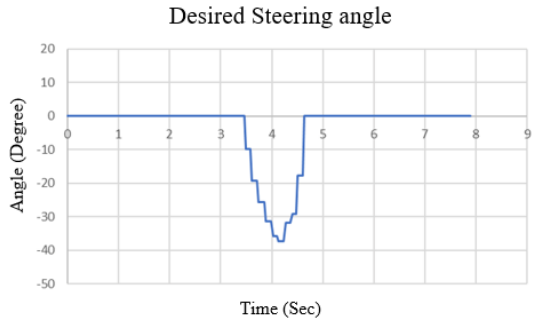
When the obstacle is placed in front of vehicle at 2.50 meters as shown in Figure 5c, the obstacle detection is set at 1.50, 1.75, and 2.00 meters for avoiding the disturbance from outdoor light and then, the results in XY coordinate system are shown in Figure 5c as red, blue, and pink solid lines, respectively. The dashed line represented the starting position of turning, which the desired steering position is changed from straight line.

For obstacle detection range is set at 2.00 meters, the motion is shown as pink line and the vehicle starts to turn at (0, 0.46) position and the desired steering is highest at 57.99 degree. The vehicle can start the turning immediately when it detected the vehicle.

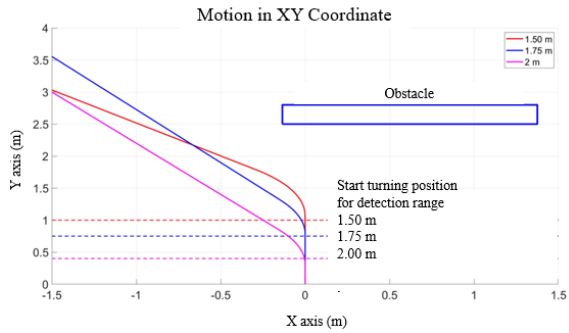
By the way, the obstacle detection range is set at 1.50 meters, the motion is shown as red line and the vehicle starts to turn at



(a) Example of the result of total field equation to calculate the avoidance angle.



(b) Example of desired steering result is published to vehicle node.



(c) Graph shows the position of obstacle and vehicle during the avoidance.

Fig. 5: Experiment result of the obstacle avoidance.

(0, 1.1) position and the desired steering is highest at 61.31 degree. The control system should be detected the obstacle at (0, 1.0) position but it has delay time before the control publish the message to avoid the obstacle. However, the vehicle can avoid the obstacle successfully but the motion of vehicle can not be turned hasty because the limitation of steering mechanism so that it may be cause of accident.

4. Conclusion

As summary for control system based on ROS, the prototype vehicle is performed by position control for steering system and speed control for powertrain system. ROS node is designed based on task, and then every nodes are performed as parallel computing and the result showed that the vehicle can avoid the obstacle successfully.

By designing the communication, the sampling control system should be operated as multi-rate control system and the sampling rate should be similar with 12 Hz, which is sampling frequency of 2D lidar but the communication PC and microcontroller board in the vehicle is transmitted via UART communication at 115,200 bit/sec. The sampling rate of desired condition has significant reduction from 10-11 Hz to 6.4 Hz, which may occur by communication speed of sensor and the process time of ROS node. However, the vehicle can avoid the obstacle. The vehicle has an opportunity to crash because the obstacle when the speed is increased and the bottleneck of communication has still occurrence.

Therefore, the control performance should improve the control performance by changing the communication protocol and process period in ROS. The other communication in the actual vehicle system such as CAN communication, which can be applied for the communication between PC and other equipments such as microcontroller and 2D Lidar. J1939 standard for CAN communication is usually used in commercial vehicle and bitrate is up to 1 Mbps.

References

[1] B Abhishek, S Gautham, D Varun Rufus Raj Samuel, K Keshav, U.P. Vignesh, Shyam R Nair: ROS based stereo vision system for autonomous vehicle, *Proceedings of 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)*, DOI: 10.1109/ICPCSI.2017.8392121 (2017).

[2] Kento Hasegawa, Kazunari Takasaki, Makoto Nishizawa, Ryota Ishikawa, Kazushi Kawamura, Nozomu Togawa: Implementation of a ROS-Based Autonomous Vehicle on an FPGA Board, *Proceedings of 2019 International Conference on Field-Programmable Technology (ICFPT)*, DOI: 10.1109/ICFPT47387.2019.00092 (2019).

[3] W. Eric Wong, Yue Chen, Dennis Kengo Oka, Martin Soukup, German Ros, Min Xie, Zijiang Yang, Mohammad Zulkernine, Ming J. Zuo: Panel on Safety, Security, and Reliability of Autonomous Vehicles, *Proceeding of 2020 Asia-Pacific International Symposium on Advanced Reliability and Maintenance Modeling (APARM)*, DOI: 10.1109/APARM49247.2020.9209498 (2020).

[4] Park J-M, Kim D-W, Yoon Y-S, Kim HJ, Yi K-S.: Obstacle avoidance of autonomous vehicles based on model predictive control, *Proceedings of the Institution of Mechanical Engineers*, Part D: *Journal of Automobile Engineering*. 2009;223(12):1499-1516. DOI: 10.1243/09544070JAUTO1149 (2009).

[5] Madhavan T.R.;Adharsh M.: Obstacle Detection and Obstacle Avoidance Algorithm based on 2-D RPLiDAR, *Proceeding of 2019 International Conference on Computer Communication and Informatics (ICCCI)*, DOI: 10.1109/ICCCI.2019.8821803 (2019).

[6] Jang-Ho Cho ,1 Dong-Sung Pae ,1 Myo-Taeg Lim ,1 and Tae-Koo Kang: A Real-Time Obstacle Avoidance Method for Autonomous Vehicles Using an Obstacle-Dependent Gaussian Potential Field, *Journal of Advanced Transportation*, Volume 2018, Article ID 5041401, DOI: 110.1155/2018/5041401 (2018).

[7] Deepali Ghorpade, Anuradha D. Thakare, Sunil Doiphode: Obstacle Detection and Avoidance Algorithm for Autonomous Mobile Robot using 2D LiDAR, *Proceeding of 2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA)*, DOI: 10.1109/ICCUBEA.2017.8463846 (2018).

[8] Siddharth Bansal, Fatemeh Alimardani, John S. Baras: Model-Based Systems Engineering Applied to the Trajectory Planning for Autonomous Vehicles, *Proceeding of 2018 IEEE International Systems Engineering Symposium (ISSE)*, DOI: 10.1109/SysEng.2018.8544452 (2018).

[9] Danai Phaoharuhansa, Akira Shimada: Multi-rate discrete control for omni-directional robot in intelligent space, *Proceeding of SICE Annual Conference 2014*, DOI: 10.1109/SysEng.2018.8544452 (2014).

[10] Danai Phaoharuhansa, Akira Shimada: An approach to SysML and Simulink based motion controller design for inverted pendulum robots, *Proceeding of SICE Annual Conference 2011*, 2011.