Developing a Supervisory Safety Algorithm for Obstacle Avoidance in an Autonomous Golf Cart

Shreejith Ravikumar, Sohan Nair

***Abstract* – A supervisory safety algorithm is proposed in this paper for obstacle avoidance for low speed autonomous vehicle. This supervisory controller alters the control inputs only when there is a possibility of collision because of failure of lower level navigation controller or else remains dormant. A Barrier function control method is used for the supervisory controller and a quadratic cost function was used to have control input from supervisory controller as close to navigation controller as possible.**

I. Introduction

Safety is a major concern while developing algorithms for navigation of autonomous vehicles. There are many algorithms which have been developed for path planning for moving the vehicle from an initial point to target point such as Model Predictive control [1][2], fuzzy logic[3] and the motion primitive method[4]. But these methods usually consume a lot of computational time and effort and often cannot guarantee safety. As such there is a need for a higher-level supervisory controller which can diagnose the collision and can navigate around when the lower level controller fails to mitigate the collision.

In this project we focus on developing supervisory obstacle avoidance algorithm for a low speed autonomous vehicle (Golf Cart). Low speed autonomous vehicles have a wide range of applications like golf carts, airport transportation, urban street commute, museum tour guide and vehicles for people with disability. Two assumptions are taken for developing this supervisory algorithm for low speed autonomous vehicle: 1) there are no lane boundaries and obstacle avoidance is the only objective 2) the obstacles are stationary.

The project is divided in two phases. In the first phase we try to develop and simulate the supervisory control on the MATLAB/Simulink. This simulation would be used to verify the controller. In second phase the control algorithm would be implemented on the Autonomous Golf Cart. This Golf Cart would be used to validate the controller.

II. Literature Review

Several researchers are investigating supervisory control algorithm for low speed autonomous vehicles. There are lot of advancements in the controllers for navigation in terms of motion planning. These controllers take a lot of computational power because of object identification and image processing. This creates a need for a higher-level controller concentrating on the safety aspect in terms of obstacle avoidance when there is failure from the lower level navigation controller.

[5] presents a way to use Barrier Function for avoiding pedestrian avoidance for low speed autonomous vehicle. The method uses a polar algorithm to compute the avoidable set using polar properties of polytopes. A mixed integer programming (MIP) method was used to solve the barrier control algorithm. The model uses a complex approach and would require solving a complex mixed integer program which might take lot of computation power.

[6] presents a non-linear control approach to tackle both navigation problem and minimize the possibility of collision. The vehicle is modelled using egocentric co-ordinates as it makes the control easier providing human like motion. The navigation controller uses Lyapunov function control. A zeroing barrier function is proposed to avoid collision. A quadratic programming approach is used to satisfy both soft and hard constraints from Lyapunov and barrier functions. This controller is validated both in simulations and on an actual robot. This approach uses a combined navigation and supervisory control. As such this supervisory control cannot be used with any navigation controller.

The golf cart is a slow speed autonomous vehicle. A zeroing barrier function is developed in this project to be implemented in golf cart to act as a supervisory controller. This controller would be first verified by simulation using MATLAB and this would then be further validated on the golf cart.

III. Approach and System Model

The vehicle is equipped with two sensors: 1) LIDAR – For getting data on depth perception and 2) Global positioning system (GPS) sensor for obtaining current coordinates of vehicle. The vehicle is modelled in the GPS sensor reference frame. The data of the obstacle's coordinates are obtained from the LIDAR. In case of multiple obstacles, the obstacle with highest possibility of collision is only considered. The reference frame used for system modelling is shown in figure 1.

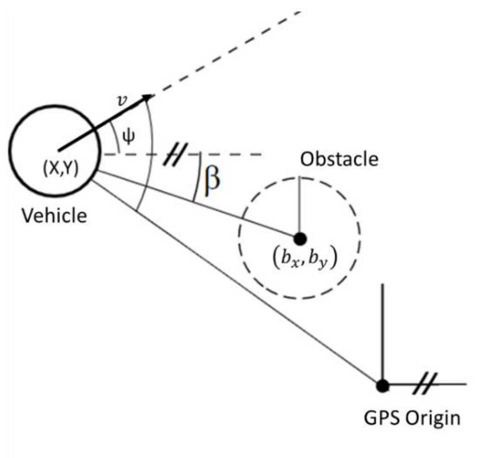

Figure 1: Reference co-ordinate system used for modelling the vehicle

Co-ordinate The dynamics of vehicle is modelled using unicycle model represented as

$$\begin{bmatrix}\dot{X}\\ \dot{Y}\\ \dot{v}\\ \dot{\psi}\end{bmatrix} = \begin{bmatrix} v\cos\psi \\ v\sin\psi \\ a \\ r\end{bmatrix} \qquad (1)$$

The control inputs to the system are the acceleration and the change in yaw rate.

$$u = \begin{bmatrix} a \\ r \end{bmatrix} \tag{2}$$

The two control inputs can be replicated in the actual golf cart trough wheel motor torque and steering motor torque.

A. Barrier Function

Control Barrier Function can be used to enforce collision safety constraint in a logical way [7][8]. The Barrier Function activates when the obstacle approaches certain threshold otherwise leaves the lower level control unaffected. We select a barrier function which manipulates control input to drive the vehicle away from obstacles as the obstacle approaches. The safe set is described as following:

$$C = x \in \mathbb{R}^3 | z_b \geq z_{safe} \tag{3}$$

$$\delta C = x \in \mathbb{R}^3 | z_b = z_{safe} \tag{4}$$

$$Int(C) = x \in \mathbb{R}^3 | z_b > z_{safe} \tag{5}$$

The barrier function is a zeroing function h(x) such that $\dot{h}(x) \leq \gamma / h(x)$. The zeroing barrier function is as follows:

$$p1 = (b_x - X - v\, T_s \cos\psi) \tag{6}$$

$$p2 = (b_Y - Y - v\, T_s \sin\psi) \tag{7}$$

$$h(x) = \sqrt{p_1^2 + p_2^2} \tag{8}$$

The h(x) function value is calculated for all the obstacles from the LIDAR sensor. The obstacle with lowest h(x) value is only considered for control barrier function.

Derivative of h(x) this function is given as follows:

$$\dot{h}(x) = \frac{1}{h(x)}[-v(p_1 cos\psi + p_2 sin\psi) + rvT_s(p_1 sin\psi - p_2 cos\psi) - aT_s(p_1 cos\psi + p_2 sin\psi)] \tag{9}$$

According to [7] the barrier function B is a control barrier function if

$$\inf_{u \in U} [\dot{B} - \gamma / B] \leq 0 \tag{10}$$

Hence, by satisfying $\dot{B} \leq \gamma / B$, barrier function can keep the robot from leaving the safe set. The Barrier function was chosen as:

$$B = \frac{1}{(h(x) - z_{safe})} \tag{11}$$

Derivative for this is given as

$$\dot{B} = -\frac{\dot{h}(x)}{(h(x) - z_{safe})^2} \tag{12}$$

B. Controller Definition

The control system consists of lower level navigation controller and the higher-level supervisory control. The objective of the supervisory control is to keep the overall control input ass close the navigation control as possible while avoiding the collision with the obstacles. A standard quadratic cost function can easily realize this which can be defined as follows:

$$minimize\ J = \left|\left|u - u_0\right|\right|_Q^2 \tag{12}$$

$$s.t. \quad \dot{B} \leq \gamma / B$$

$$v_{min} \leq v \leq v_{max}$$

$$a_{min} \leq a \leq a_{max}$$

$$r_{min} \leq r \leq r_{max}$$

This ensures that the vehicle never leaves the safe zone and the final control inputs are as close to the navigation controller inputs as possible.

C. Nomenclature

(X, Y)	Co-ordinates of the location of vehicle(m)
v	Velocity of the vehicle(m/s)
a_0	Acceleration input from navigation controller(m/s^2)
a	Final acceleration input from controller(m/s^2)
ψ	Yaw angle of the vehicle(rad)
r_0	Yaw rate input of the vehicle from navigation controller (rad/s)
r	Final Yaw rate of the input of vehicle from controller (rad/s)
C	Safe set
z_b	Distance of the obstacle from the vehicle(m)
z_{safe}	Safe distance between vehicle and obstacle(m)
(b_x, b_y)	Co-ordinates of the location of obstacle
Ts	Sample time of the controller (s)
γ	Positive controller constant
Q	Cost function weight matrix

IV. Simulation Setup

The simulation setup consists of a vehicle simulator and a controller. The general layout of simulation is shown in figure 2.

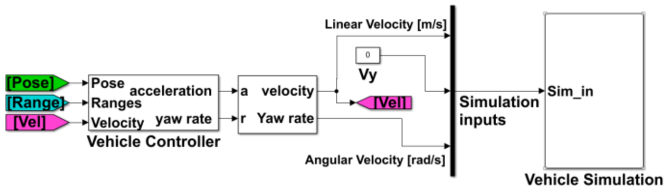

Figure 2: Simulink layout of vehicle control simulation

A. Vehicle Simulation Model

MATLAB Robotics Simulation Toolbox was used for simulation of the sensors and the motion of the vehicle. The toolbox can simulate vehicle motion if velocity and heading angle change was given to it as input. It can also simulate the behavior of LIDAR sensor during the vehicle motion. A built-in virtual Map was used to generate obstacles for the vehicle. The simulation layout is shown in figure 3.

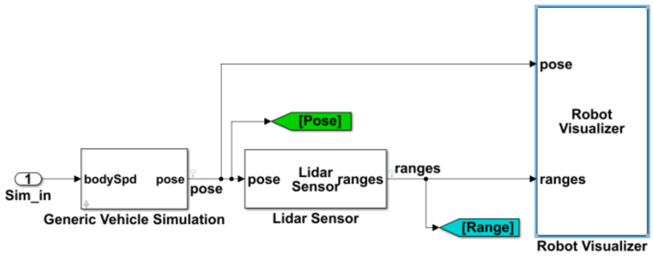

Figure 3: Vehicle motion simulation model layout

The Lidar sensor gets current location and posture of vehicle as input and gives distance of the object and angle of the object with respect to vehicle heading direction for various scan angles. There are 101 scan angles are distributed evenly from -pi/2 to pi/2 from the vehicle heading angle. The velocity and heading angle rate are computed from the acceleration and velocity input from the Higher-level controller. The figure 4 illustrates LIDAR angles and vehicle position.

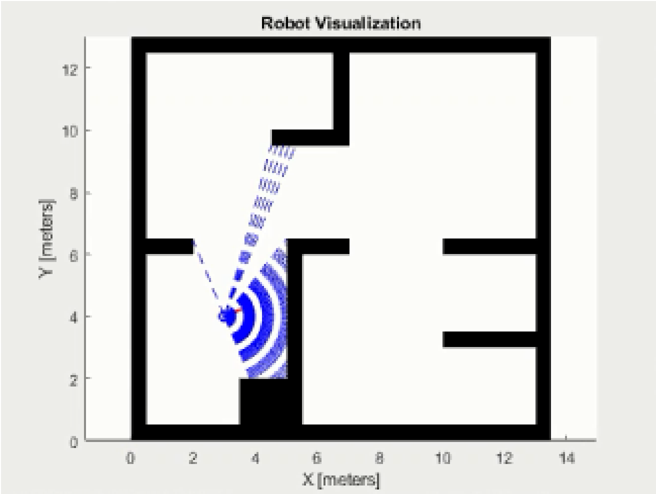

Figure 4: The black zones are boundaries or obstacles. The blue dashed lines are the scan angles from the lidar. The blue circle is the vehicle position. Blue solid curve is vehicle path.

B. Controller Design

The main controller consists of lower level navigation controller and the higher-level supervisory controller. The controller takes input from vehicle pose, vehicle motion and LIDAR sensor data as input and gives out vehicle acceleration and yaw rate as the output.

The supervisory controller takes input from the lower level navigation controller to get the acceleration and yaw rate and as well as the vehicle motion and posture data from the sensors. The controller gives final acceleration and yaw rate as the output, which is sent to the vehicle simulation. The layout of the vehicle control is shown in figure 5.

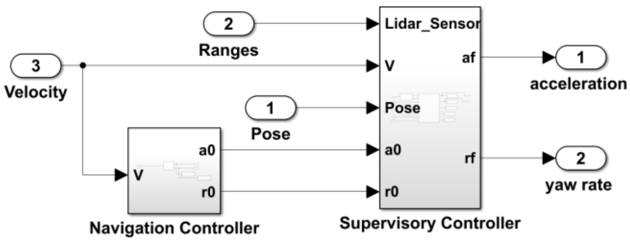

Figure 5: Vehicle controller layout with navigation and supervisory controller.

V. Simulation Model and Results

For current simulation, the vehicle navigation controller was considered as PI controller trying to match the velocity of the vehicle to 2 m/s and the yaw rate to be zero. Various simulation parameters are given in the table 1.

Table 1: Simulation parameter values

Parameters	Value
z_{safe}	0.8
Initial X position	3
Initial Y position	4
Initial Yaw	-Pi/4, 0, Pi/8(Three runs)
Initial velocity	2
(a_{min}, a_{max})	(-4,4)
(r_{min}, r_{max})	(-5,5)
(v_{min}, v_{max})	(-3.2,3.2)
γ	$5.15*10^7$
Q	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

The motion of the vehicle under supervisory control is shown in figure. The vehicle is simulated for 5s and the vehicle is successfully able to avoid the obstacles even when the navigation controller was directing the vehicle towards the obstacle. This can be clearly seen in the figure 6 where the blue solid curve indicates vehicle motion. Figure 7 shows the change in barrier function and derivative of barrier function as the vehicle comes close to obstacle. Figure 8 shows the change in acceleration and the yaw rates imposed by the supervisory controller. The acceleration and yaw rate values are not altered when the vehicle is away from the obstacle but as the vehicle approaches the obstacle the acceleration values drive the vehicle away from the obstacle as shown in figure 8. The same is shown for two more situation in figures 9-14.

VI. Conclusion

Literature review was done on various Barrier function-based obstacle avoidance system implemented in various autonomous vehicles. A functional Supervisory Barrier function was identified and implemented. The Barrier function was simulated on a PI navigation controller for verifying the controller. The Barrier function controller was successful in avoiding the obstacles when the navigation controller was heading the vehicle towards an obstacle.

VII. Future Work

The computer simulations to verify the supervisory control has been achieved using PI controller as a navigation controller. The controller must be tested for different possible types of navigation controllers. The project can move to the second phase, where this controller would be implemented on the Golf cart platform. The Golf cart uses dSPACE platform for its Microautobox control hardware. The controller must be translated to dSPACE friendly blocks to generate C-code for flashing in the hardware. One of the major challenges is to reduce the computation time required for the quadratic cost function.

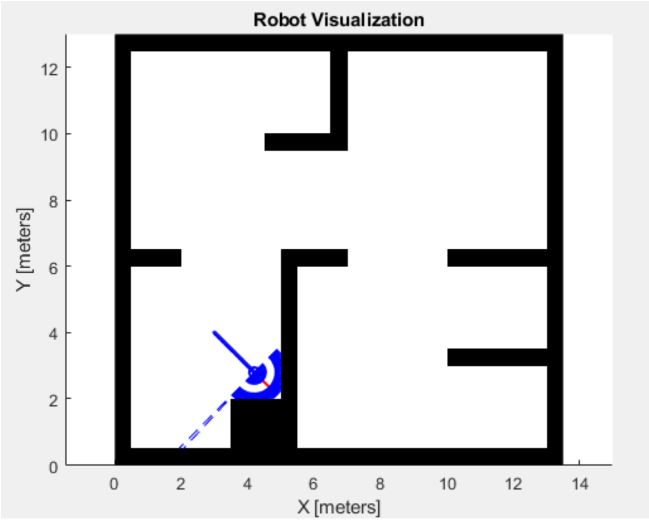

Figure 6: The motion of vehicle using vehicle controller for yaw angle = -pi/4

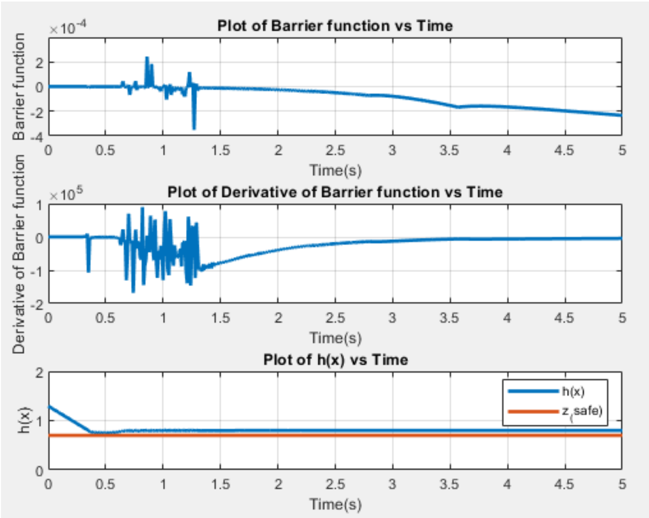

Figure 7: Barrier function plots for yaw rate = -pi/4

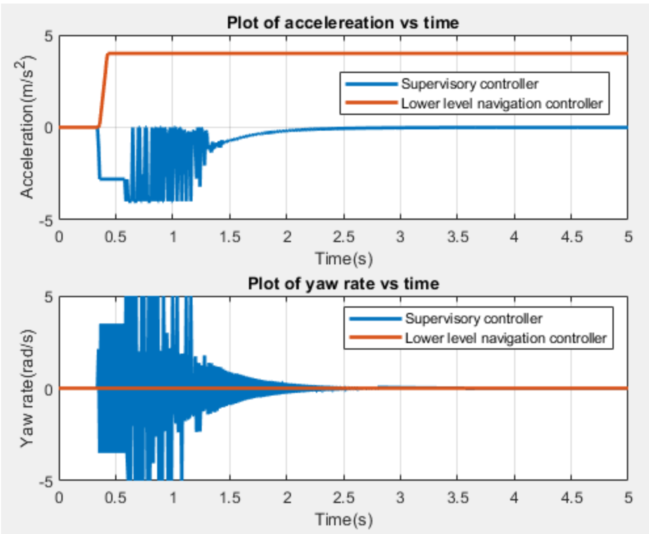

Figure 8: Acceleration values and yaw rate plots for yaw rate = -pi/4

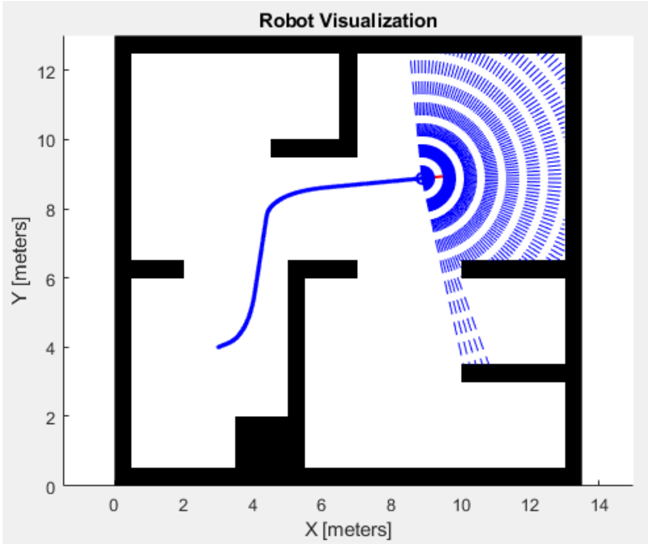

Figure 9: The motion of vehicle using vehicle controller for yaw angle = pi/8

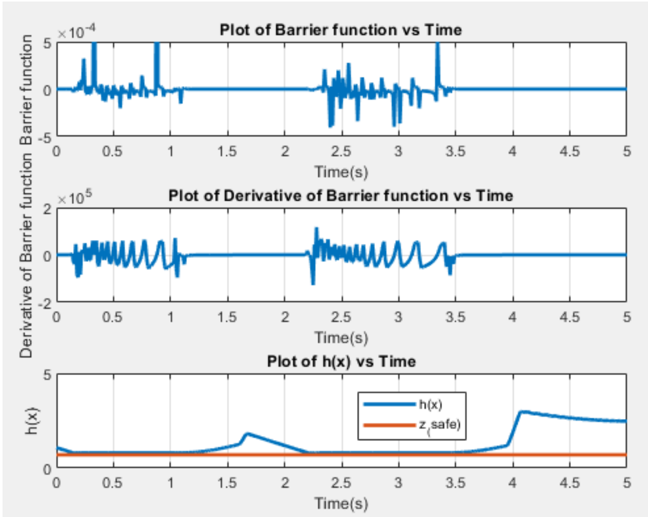

Figure 10: Barrier function plots for yaw rate = pi/8

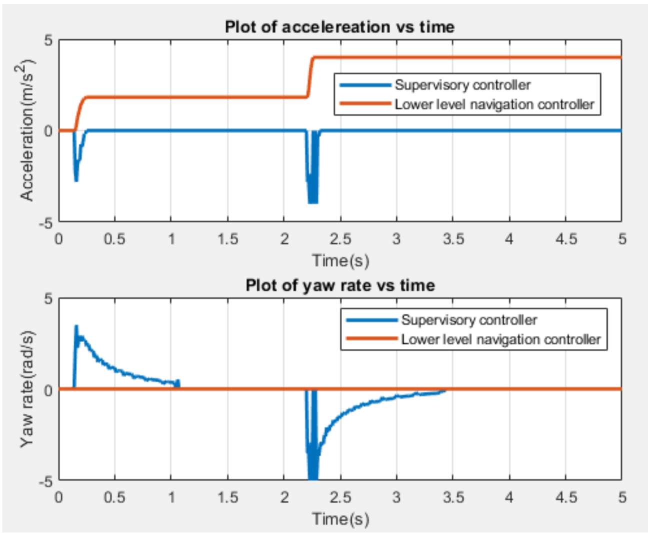

Figure 11: Acceleration values and yaw rate plots for yaw rate = pi/8

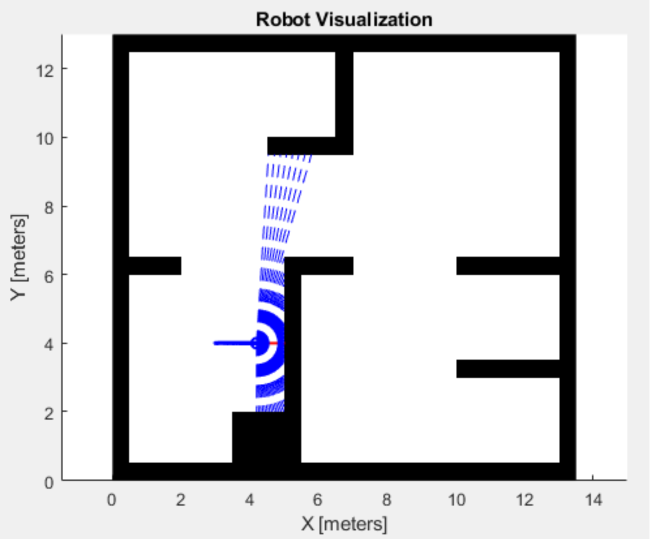

Figure 12: The motion of vehicle using vehicle controller for yaw angle = 0

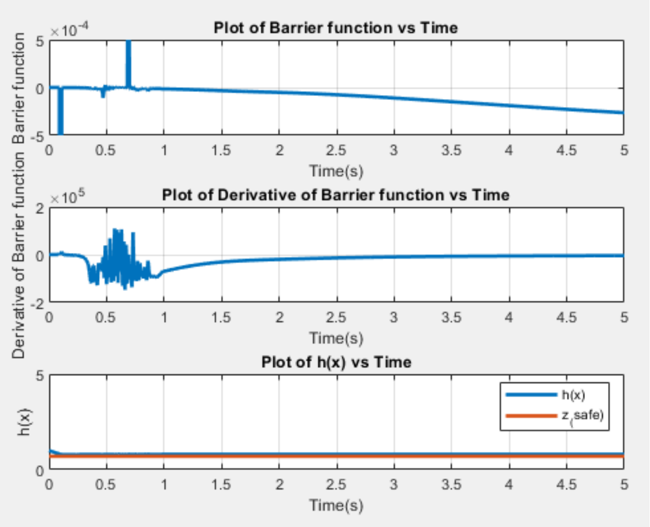

Figure 13: Barrier function plots for yaw rate = 0

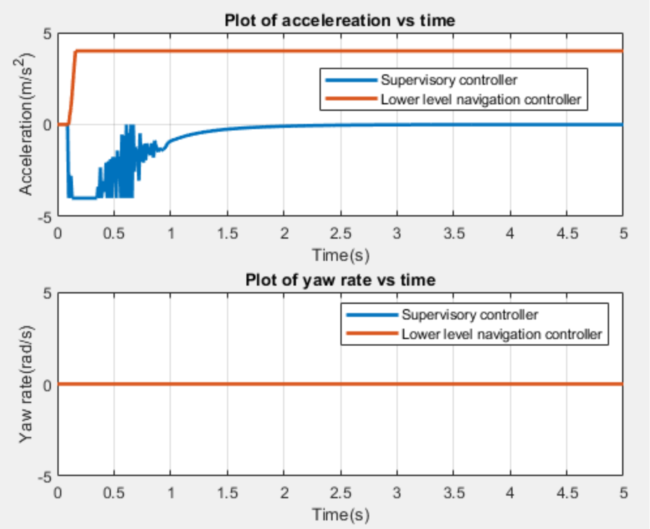

Figure 14: Acceleration values and yaw rate plots for yaw rate = 0

VIII. References

[1] A. Gray, Y. Gao, T. Lin, J. K. Hedrick, H. E. Tseng and F. Borrelli, "Predictive control for agile semi-autonomous ground vehicles using motion primitives," 2012 American Control Conference (ACC), Montreal, QC, 2012, pp. 4239-4244, doi: 10.1109/ACC.2012.6315303.

[2] Y. Yoon, J. Shin, H.J. Kim, Y. Park and S. Sastry, "Model-predictive active steering and obstacle avoidance for autonomous ground vehicles," 2009 Control Engineering Practice, 17(7), pp.741-750.

[3] D. Fernandez Llorca et al., "Autonomous Pedestrian Collision Avoidance Using a Fuzzy Steering Controller," in IEEE Transactions on Intelligent Transportation Systems, vol. 12, no. 2, pp. 390-401, June 2011, doi: 10.1109/TITS.2010.2091272.

[4] E. Frazzoli, M. A. Dahleh and E. Feron, "Real-time motion planning for agile autonomous vehicles," Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148), Arlington, VA, USA, 2001, pp. 43-49 vol.1, doi: 10.1109/ACC.2001.945511.

[5] N. Arora, I. Buckley, Varun Murali, "Evaluation of Safe Control Policies for Mobile Robots in Dynamic Environments," Segway Robot in Dynamic Environments, 2016.

[6] A. D. Ames, J. W. Grizzle and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," 53rd IEEE Conference on Decision and Control, Los Angeles, CA, 2014, pp. 6271-6278, doi: 10.1109/CDC.2014.7040372.

[7] A. D. Ames, K. Galloway, K. Sreenath and J. W. Grizzle, "Rapidly Exponentially Stabilizing Control Lyapunov Functions and Hybrid Zero Dynamics," in IEEE Transactions on Automatic Control, vol. 59, no. 4, pp. 876-891, April 2014, doi: 10.1109/TAC.2014.2299335.

[8] Yuxiao Chen, H. Peng and J. Grizzle, "Correct by construction design of autonomous vehicles through a barrier function method," 2017 American Control Conference (ACC), Seattle, WA, 2017, pp. 4926-4931, doi: 10.23919/ACC.2017.7963718.