Navigating roundabouts and unprotected turns in autonomous driving

Mahir Gulzar[*]
Institute of Computer Science
University of Tartu
Tartu, Estonia 51009
`FirstName.LastName@ut.ee`

Yar Muhammad
Department of Computer Science, School of Physics, Engineering & Computer Science
University of Hertfordshire
Hatfield, AL10 9AB, United Kingdom
`y.muhammad@herts.ac.uk`

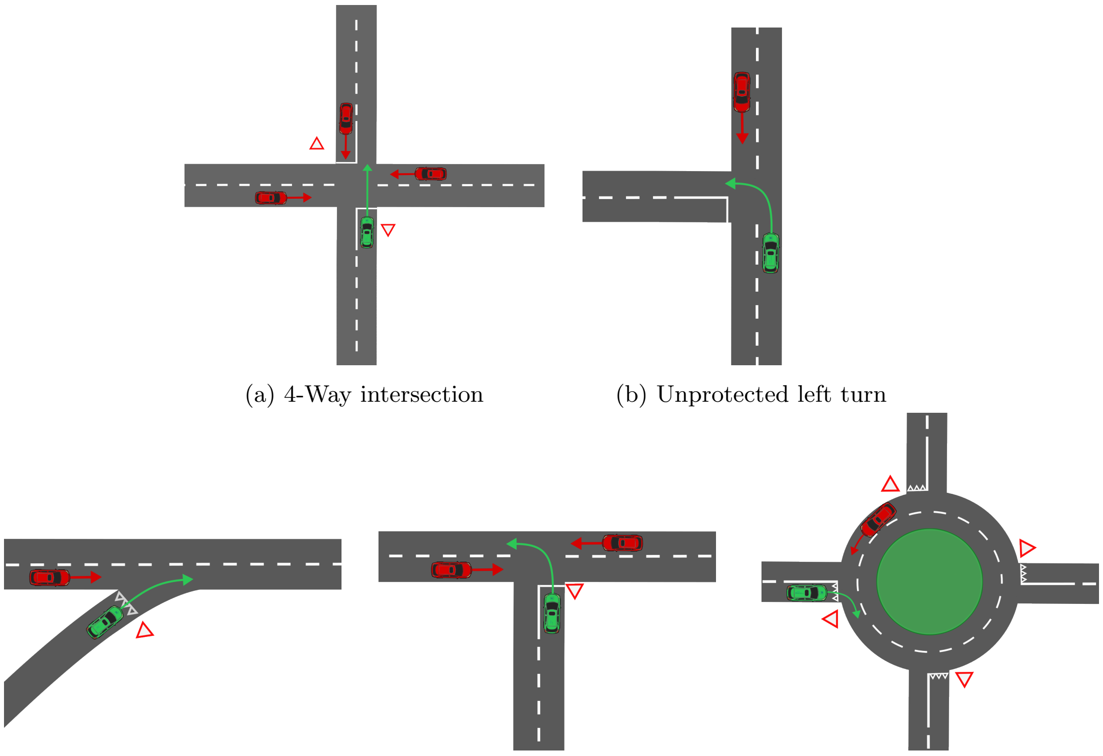
Naveed Muhammad
Institute of Computer Science
University of Tartu
Tartu, Estonia 51009
`FirstName.LastName@ut.ee`

Abstract

The development of a fully autonomous driving vehicle (AV) requires various traffic situations to be handled efficiently. One of the most common driving manoeuvres which an AV experiences in daily traffic is giving way (yielding) to other traffic participants. In this paper, we propose a simple yet efficient method of yielding that doesn't query yielding areas of interests from map API making it hassle free to use without having to rely on digitized yielding areas. We incorporated our method into one of the well-known open-source autonomy stacks called Autoware. The proposed method makes use of high-definition (HD) map elements including lanes and stoplines for filtering vehicles which participate in yielding decision making. Our method estimates future collisions of filtered vehicles of interest with AV's planned trajectory and outputs a binary yielding decision for ego vehicle. Our method covers different yielding areas including a roundabout and an unprotected turn. We tested and evaluated the decision making of our method on various simulated scenarios and afterwards successful real-world tests were conducted using an in-house AV. An in-depth analysis of our approach shows that the proposed yielding solution works reasonably well i.e. 87% successful yielding area navigation ratio on real data.

Keywords: autonomous driving, autonomous navigation, yielding, give-way areas, Autoware

[*]Mahir Gulzar is currently doing PhD in Computer Science from University of Tartu, Estonia. His research interests include context aware behaviour modelling and motion planning in autonomous driving. He is also working as research engineer in Autonomous Driving Lab of Institute of Computer Science, University of Tartu, Estonia. Visit https://adl.cs.ut.ee for more information about his research and the lab.



(a) 4-Way intersection (b) Unprotected left turn

(c) High-way merge (d) T-junction (e) Roundabout

Figure 1: Illustration of various give-way areas. Here AV is denoted by green color which does not have the right of way

1 Introduction

Research in autonomous vehicle industry has been very active since the past decade specifically after DARPA Grand and Urban challenges (Buehler et al., 2009). Many automotive AV manufacturers have raised the bars over SAE autonomy levels (SAE Levels of Autonomy, 2021). In addition to Tesla and Waymo, companies such as Cruise, Argo.AI are developing fully autonomous driving systems strictly following level 4 autonomy (Singh and Saini, 2021). For an AV to co-exist with other manually driven vehicles in real-life traffic, it has to understand the traffic rules and efficiently navigate different traffic scenarios such that it closely mimics a normal human driver. One of the common challenging traffic scenarios that an AV should handle is navigating yielding or give way signs. According to Vienna Convention on Road Signs and Signals, the give way sign falls under the umbrella of priority signs and is categorized as priority B-1 (Convention on Road Signs and Signals, 2006). Some of the most notable yielding areas we encounter in daily traffic include are shown in Figure 1. Here Figure 1a shows a 4-way intersection scenario in which the protected horizontal lane has priority over AV's (green vehicle) lane. Figure 1b shows an unprotected left turn scenario where there is no yield sign placed but by traffic regulations AV has to give way to the oncoming traffic. Figure 1c and Figure 1d are examples of high-way lane merge and T-junction. Lastly, Figure 1e shows a very common roundabout scenario where a yielding sign is placed on all entry points and any vehicle outside the roundabout should yield for traffic inside the roundabout unless there is a safe time interval for low priority vehicle to navigate without interrupting the flow of roundabout traffic. In addition to above illustrated yielding intersections, there are various unmarked yielding intersections e.g. an unprotected u-turn.

The idea of this research is to enhance automated driving systems with the capability of navigating yielding areas without driver intervention. In order to build this functionality we turn towards open-source autonomy stacks. One of the most notable and modular open-source autonomy stacks is Autoware (Kato, 2017) by Autoware Foundation. Companies such Apex.AI and Tier.IV contribute to this stack. This stack is also supported by a decent number of AV developers community. According to statistics, many research institutions and AV companies including startups use Autoware's open-source solution (Carballo et al., 2019). The two most notable versions of Autoware i.e. Autoware.AI and Autoware.Auto make use off Robotic Operating System (ROS) and ROS2 respectively. In this research, we use Autoware.AI as our autonomy stack. While Autoware gives us solid baselines to start working with, it still lacks different AV functionalities. Yielding for other vehicles is one of the missing functionalities. The main contribution of this research is as follows:

1. We propose a general open-source solution to yielding problem that is implemented and integrated into Autoware.AI. Our proposed solutions makes use of HD-map for iteratively filtering vehicles of interest and afterwards outputs a binary yielding decision by performing collision estimation of filtered vehicles with AV.

2. We evaluate the performance of multiple yielding scenarios in both simulation and in the real world.

The novelty of our work is the proposition of solution to a problem that is addressed in the literature with very limited scope i.e. limited number of scenarios, less generalizability and simulation tests only. To the best of our knowledge, real-world experiments in the literature on yielding for autonomous vehicles are very scarce. Because of that, we are unable to quantitatively compare our results with those published in the literature. We will shed some light on works which relate well to our work in the next section.

The rest of the paper is organized as follows. Section 2 highlights the related work on navigation and decision-making in different give-way intersections. Section 3 formalizes the yielding problem keeping in view all state space variables. Section 4 first presents a detailed step-by-step implementation procedure of our approach and then summarizes the implementation into a comprehensive yielding algorithm pseudocode. Section 5 discusses simulation and real-life experiments and compares the results of different evaluations. Finally, section 6 concludes the study with a summary of the work and valuable takeaways.

2 Related work

Yielding problem has been addressed by different names and strategies such as trajectory planning or manoeuvre planning problem for intersections and roundabouts etc. A good starting point to understand the available literature is to categorize it on the basis of insertion strategies. A comprehensive categorization of intersection or give-way areas driving strategy is given by (Wei et al., 2021). For an AV, the individual driving strategy is broadly categorized into classical and learning-based strategies. The classical strategies are often called rule-based or state-machine-based solutions. The state changes from one state to another according to some predefined rules. Although rule-based is a widely used solution, it is difficult to adapt these solutions to generalize over all dynamic traffic scenarios (Lin et al., 2019). Learning-based driving strategies on the other hand include Bayesian networks and machine learning models for decision-making. They handle complex scenarios better but suffer in terms of reliability, safety guarantees and excessive training times. In terms of classical strategy, (Nilsson et al., 2016) discusses a trajectory planning algorithm for automated yielding manoeuvres. It proposes four steps of trajectory planning algorithm for yielding regions i.e. determining longitudinal and lateral safety corridors and longitudinal and lateral safety trajectory. For a certain traffic scenario where ego vehicle does not have right of way, the corresponding space is divided into three regions: Pre, Peri, and Post. The authors define Pre region as a region in which ego vehicle should account for vehicles and other objects while approaching the Peri region. The Peri region is region where ego does not have the right of way and lastly, the post region which includes the traffic participants which the ego should account for before leaving Peri region. A common blend of classical strategy and statistical models with high-definition (HD) maps is often observed in different works. (Masi et al., 2020) proposes a methodology to allow an AV to safely cross a multi-lane roundabout. The proposed strategy uses HD maps and intervals representing road occupancy by vehicles, with roads being widened to reflect uncertainties in localization. A roundabout insertion strategy was introduced which describes that AV should maintain a safe distance, follow traffic rules and avoid unnecessary stopping at roundabout entry point. Another map-based approach was proposed in (Noh, 2018) for navigating controlled and uncontrolled intersections. Here, map aware predicted trajectories were used for motion prediction and Bayesian statistical model was employed for risk assessment and filtering, given noisy estimates of other agents. A framework of decision-making at intersection is presented in (De Beaucorps et al., 2017) where human participant's data was recorded for lateral and longitudinal decision making. Later the recorded speed profiles were clustered into passing, yielding and stopping behaviour for an intersection and roundabout. In (Bey et al., 2021) authors describe safe roundabout as a planning problem and formulate the problem using a Partially Observable Markov Decision Process (POMDP). Partial observability refers to parts of the state being hidden from the agent which includes whether roundabout vehicle exits the roundabout or keeps navigating the roundabout. Here, a particle filter is used to track the vehicle in the roundabout. It creates probability distribution of multiple possible states of the target vehicle afterwards the actual planning happens using the belief tree from which the potential future state trajectories of the target vehicle are sampled. Similar to previous work, authors in (Song et al., 2016)(Hubmann et al., 2017)(Hubmann et al., 2018) also make use of POMDP for decision-making in uncontrolled intersections. Some works have employed pure machine learning based solutions to solve the decision-making for give-way areas. Examples include (Tollner et al., 2019) where a simplified binary classification method using artificial neural network (ANN) was proposed. Here the authors trained a network with different configurations of hidden layers to classify if an ego vehicle should enter the roundabout or stop. The tests were conducted in (Krajzewicz, 2010) SUMO simulator with two simulated vehicles i.e. an ego vehicle and a target vehicle. Another approach proposed in (Li et al., 2020) uses deep reinforcement based decision-making framework for navigating intersections. Here, are deep-Q network was employed to learn a policy that maximizes the reward. The reward function is tuned in a way that AV avoids collision with any other vehicle and is not too conservative i.e. travels as fast as possible through the intersection.

As discussed previously, our main objective here is to implement an open-source generic solution of simple give-way area navigation w.r.t. road signs i.e. we intend to solve automated yielding for multiple traffic situations keeping in view the traffic priority. The literature we reviewed previously mostly solves specific give-way intersections without generalization. Most of the solutions discussed do not even take into account different types of road priority signs. For this purpose, we leverage the models discussed in the literature and implement a yielding manoeuvre in Autoware.AI. To be specific, we extend a third party motion planner called OpenPlanner (Darweesh et al., 2017) (which is integrated into Autoware.AI) and implement yielding functionality in it. OpenPlanner is example of classical state-machine approach where AV's manoeuvres are built on top of a behaviour state machine. Our approach extends OpenPlanner's behaviour state machine by adding yielding states. It makes use of HD maps with map-aware motion prediction of other vehicles. Our approach identifies uni-directional and multi-directional traffic yielding i.e. planning and paying attention to only those vehicles which have priority over AV. To the best of our knowledge, none of the models discussed in the literature is readily available nor integrated into any existing open-source autonomy stack.

3 Problem statement

The goal of this work is to implement a yielding driving manoeuvre state for ego vehicle for traversing simple give-way areas. Here, the yielding areas can have variable number of other vehicles. The system can be formulated as follows:

In each frame of the scene, there are variable number of agents $\mathcal{A} = \{A_0, ..., A_I\}$, where $I \in \mathbb{N}_0$ that represents total number of agents in the current scene. For each agent $A_i = (x_i, y_i, z_i, \theta_i, v_i, w_i, h_i) \in \mathbb{R}^7$ where x_i, y_i, z_i represents agent i's position in 3D space and θ_i, v_i, w_i, h_i represent yaw, velocity, width and height respectively. Each agent has hypothetical future behaviours. These behaviours are represented as a set of trajectories $\mathcal{T}_i = \{T_{i,0}, ..., T_{i,J}\}$ where $J \in \mathbb{N}_0$ which represents multi-modal trajectory and J varies for each agent depending on the road scenario. For an agent A_i possible trajectories can be defined as

$T_{i,j} = \{(x^t_{i,j}, y^t_{i,j}, \theta^t_{i,j}) | t = 0, ..., t_{pred}\}$ and $(x^t_{i,j}, y^t_{i,j}, \theta^t_{i,j}) \in \mathbb{R}^3$ being the trajectory point with rotation angle of hypothesis j of agent i at timestep t. A set of high-definition map elements $\mathcal{M}$, represented by lanes, yielding stop lines etc. Here, lanes are represented as $\mathcal{L} = \{L_0, ..., L_M\}$ for $M \in \mathbb{N}$ where each lane is composed of way-points and can be represented as $L_m = \{(x_{m,0}, y_{m,0}, \theta_{m,0})...(x_{m,n}, y_{m,n}, \theta_{m,n}))\}$ for $m, n \in \mathbb{N}$ and $(x_{m,n}, y_{m,n}, \theta_{m,n}) \in \mathbb{R}^3$ being the top-down 2D position and rotation of waypoint n in lane m. The yielding stop lines are represented as $\mathcal{S} = \{S_0, ..., S_K\}$ where $K \in \mathbb{N}$ and $K \not\equiv M \not\equiv I$ and $\mathcal{L}, \mathcal{S} \subseteq \mathcal{M}$. Each stop line is essentially a waypoint at the end of a lane or start of another lane. Here, $S_k = (x_k, y_k, \theta_k)$ is kth stop line's waypoint. Every agent is associated to the nearest lane and agent's trajectories are confined within the topological lanes of the map. For simplicity, we are going to assume that A_0 is always the ego vehicle, this means there will be only one future path that which ego will follow thus for $T_{0,j}$ the number of modes j will always be 1. The yielding decision then boils down to the following function that takes in all relevant information and outputs a binary decision:

$$yield(\mathcal{A}, \mathcal{T}, \mathcal{L}, \mathcal{S}) = \begin{cases} 1, & \text{if any other vehicle(s) has priority} \\ 0, & \text{otherwise} \end{cases}$$

4 Approach

We leverage an existing open-source autonomy stack (Autoware.AI) and the integrated planner (OpenPlanner) that provides us with the future navigation path of ego vehicle. OpenPlanner also predicts future motion estimates and trajectories of other vehicles using particle filter and fuses HD map information into the prediction. The planner keeps track of ego's behaviour state machine and switches between defined states based on certain conditions. Figure 2 shows some of the abstract behaviour states which are currently implemented within OpenPlanner. As an example, consider traffic light stop state. An AV transitions from *Follow* or *Forward* state to *TrafficLightStop* when AV encounters a red traffic light. In this state, the car starts decelerating until it reaches zero velocity. Once the zero velocity is achieved (and the traffic light is still red) the behaviour state transitions from *TrafficLightStop* to *TrafficLightWait* state where the AV waits at a full stop until the traffic light signal turns green. In our approach, we extend the existing state machine of the planner and introduce two new concrete behaviours. Following a similar stop-and-wait state machine architecture, we propose two new states as illustrated within dotted area in Figure 2.

AV switches from *Follow* or *Forward* state to *YieldStop* state and starts decelerating to zero velocity when

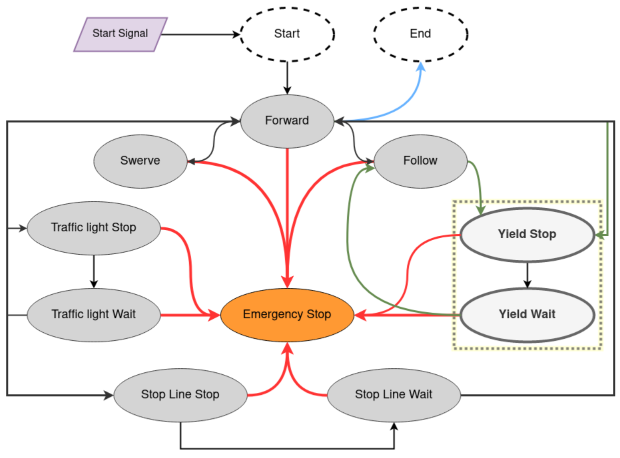


Figure 2: Behaviour state machine of AV adapted from (Darweesh et al., 2017) where proposed new states are shown in dotted highlighted square.

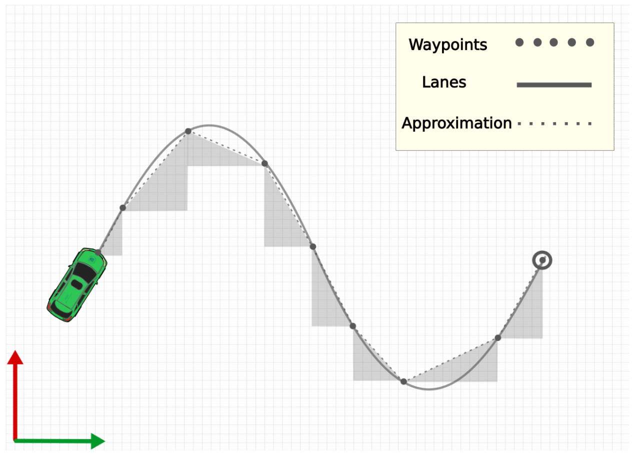


Figure 3: Approximation of ego's distance to a stopping point on its path.

there is a potential vehicle to yield. Transition from *YieldStop* to *YieldWait* happens if the conditions mentioned for *YieldStop* persist and AV has approached zero velocity. It is to be noted that *StopLineStop* and *StopLineWait* are built-in states of OpenPlanner where the default functionality is to always stop at a stop line and wait for a specified number of seconds. With the newly proposed behaviour states, we divide the implementation into following general steps.

4.1 Identifying yielding area

The very first step to trigger yielding manoeuvre is to identify where the ego vehicle should evaluate its future rollout for yielding. As human drivers, we get this cue from the yield road sign. For an AV this information can come from either road sign detector or using HD maps. Luckily, OpenPlanner gives us a rich map API that translates different types of HD maps into vector map format that integrates well within the planning framework. As we intend to validate our implementation results on real vehicle, we make use of HD maps and where each agent in the environment is associated to the nearest lane. The lanes are composed of waypoints that are placed after regular intervals as formulated in section 3. In our approach, the AV initiates the evaluation of its future rollouts for yielding when it encounters a yielding stop line within its future planned route. This yielding stop line discovery is tied to a planning horizon parameter (shown in later sections). The longitudinal distance from one point to another along a specified lane is essentially an approximated summation of hypotenuse of right triangles between two waypoints as illustrated in Figure 3.

4.2 Filtering vehicles of interest

At this point, we assume that the AV has entered a specific yielding evaluation area. The next step is to start the rollout evaluation. The future rollout of an AV should be evaluated against vehicles of interest (VOIs). (Noh, 2018) identifies this step as situational awareness or specifically threat identification in an uncontrolled 4-way intersection with stop signs. Another example of threat identification can be observed from (TierIV and Apex.AI, 2021) where for a roundabout and unprotected left turns, AV comes to a full stop and highlights certain map areas to look for vehicles to yield. Our work classifies this step as filtering VOIs for which an AV should yield. Instead of maintaining a separate topological database to look VOIs for each give-way area, we utilize the geometrical aspects of give-way areas as well as behavioural aspects of other vehicles. We divide the VOI filtering into two sub-levels.

4.2.1 Ego's route followers

In the very first filter step, we intend to identify and remove vehicles which follow the local navigation route of ego vehicle. More intuitively, all those vehicles which are leading and trailing the ego vehicle in the same lane should be removed as no potential yielding is required for them. It is to be noted that leading vehicles in ego's lane are considered non-priority obstacles because they are handled by *Follow* state in OpenPlanner. For an agent A_i, we take its top-down center point and yaw rotation $\{x_i, y_i, \theta_i\}$ where $i > 0$, and project it to the local navigation route of ego vehicle. An agent which is supposedly following ego's route will have a projection where perpendicular distance is below the lane association threshold and yaw angle approximately facing in a similar direction as the nearest lane waypoint.

Recall from section 3 that A_0 is always ego vehicle, thus $T_{0,0}$ is ego's local route. Let p_0 be the nearest waypoint from vehicle under consideration (VUC) in ego's route and p_1 the waypoint followed by p_0. Here, p_i represents (x_i, y_i, θ_i) and $\{p_0, p_1\} \subseteq T_{0,0}$. Let λ be the waypoint $(\hat{x}, \hat{y}, \hat{\theta})$ consisting position and orientation of VUC. The projection of λ on route $T_{0,0}$ is as follows:

$$p'_i = p_i.\mathbf{T}$$

$$where \qquad \mathbf{T} = \begin{bmatrix} \cos(-\theta_0) & -\sin(-\theta_0) & -\hat{x} \\ \sin(-\theta_0) & \cos(-\theta_0) & -\hat{y} \\ 0 & 0 & 1 \end{bmatrix}$$

p'_i shows the transformed lane waypoints w.r.t. VUC's frame of reference. Assuming these points form a straight line, we can calculate the $y - intercept$ or lateral distance of VUC to $T_{0,0}$.

$$m = \frac{y'_1 - y'_0}{x'_1 - x'_0}$$

$$b = y'_0 - m(x'_0) = y'_1 - m(x'_1) \tag{1}$$

$$\Delta\theta = \theta'_0 - \hat{\theta} \approx \theta'_1 - \hat{\theta} \tag{2}$$

Here, b is the perpendicular distance and $\Delta\theta$ is angle difference of VUC from ego's route. In Equation (2), the approximation holds since θ'_0 and θ'_1 are angles of consecutive waypoints in same lane. From equation (1) and (2), we can generate a perpendicular waypoint of VUC on $T_{0,0}$:

$$p_{perp} = \begin{bmatrix} 0 \\ b \\ 1 \end{bmatrix}.\mathbf{T}^{-1}$$

Here, the line from point λ to p_{perp} is perpendicular to line passing from point p_0 and p_1. An agent A_i is AV's route follower if $|b|$ is less than lane assignment threshold and $|\Delta\theta|$ is less than maximum yaw deviation. These thresholding parameter values are discussed in later sections.

4.2.2 Behavioural intersects

The next step in vehicle filtering is to remove vehicles which have trajectory intersects before the yielding stop line. The idea here is to reduce unnecessary braking for vehicles with low priority which are crossing over or merging into AV's lane before the yielding area while AV is evaluating its rollout for actual vehicles to yield. This can be considered as removing vehicles which have behavioural intersects in peri region proposed in (Nilsson et al., 2016) or decision zone as proposed in (Masi et al., 2020). For this step, we make use of multi-modal trajectories of surrounding vehicles and estimate the intersections with AV's navigation route. For an agent A_i, we have $T_{i,j}$ multi-modal predicted trajectories according to the road geometry where j is the number of trajectory modalities or hypothesis as explained in section 3. We make use of particle filter given in OpenPlanner to get the discrete conditional probability of multiple modes.

$$\epsilon = \underset{j}{\operatorname{argmax}}(P(T_{ij}|A_i)) \tag{3}$$

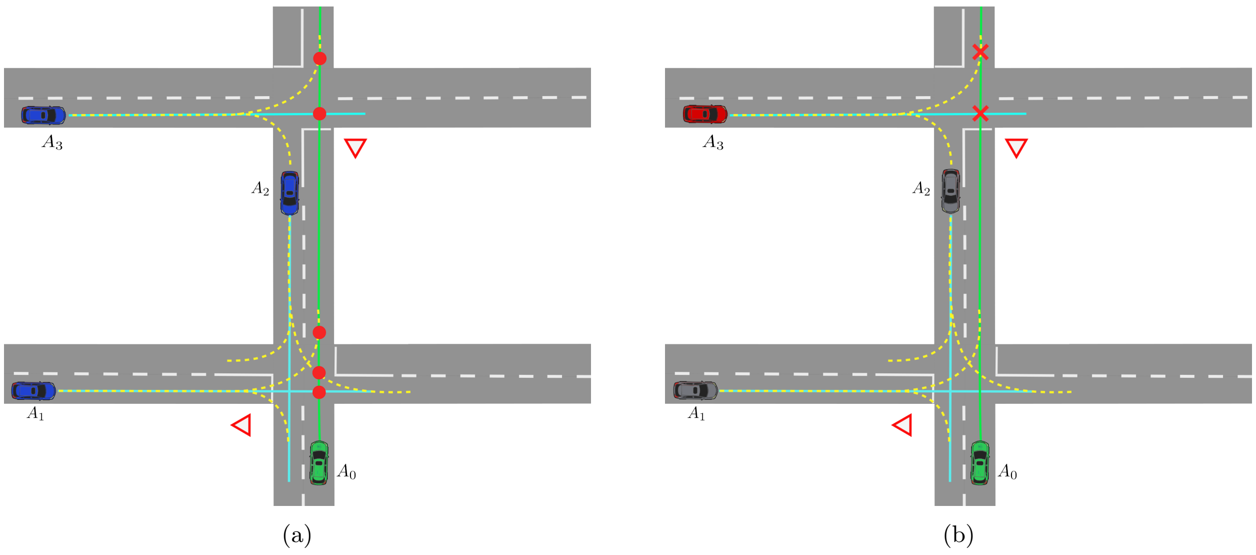


Figure 4: Example of VOI filtering using behavioural intersects. Here green vehicle is ego vehicle, blue vehicles are vehicles before behavioural intersect filtering, gray vehicles are vehicles which are ignored for yielding and finally red vehicle is the vehicle which will be given right of way.

From equation (3), $P((T_{ij})|A_i)$ is the probability of jth trajectory of agent i given agent's information A_i. ϵ is the hypothesis with highest likelihood over a set of trajectories. Since most-likely trajectory is not always the actual trajectory taken by VUC, we calculate intersects of all trajectory modes of the surrounding vehicles with AV's future trajectory T_0. Figure 4 illustrates a scenario where behaviour filtering is shown in action where highest probable trajectories are shown in cyan colour while yellow dotted lines are less likely trajectories (notice the behavioural intersects are calculated with all trajectories). Here, Figure 4a shows that AV i.e. A_0 has behavioural intersects with two agents A_1 and A_3. The AV should yield for A_3 and A_1 should yield for AV according to the road signs. Figure 4b shows the result of behaviour-based filtering where AV pays attention to A_3 for yielding and considers A_1 and A_2 for emergency braking only.

4.3 Decision making

Once we have filtered the VOIs, the final step is to perform decision making i.e whether the AV should yield or not. The decision-making step is often followed by a risk assessment step as all potential risks should be calculated before making a decision. In our approach, we identified the risks by filtering out VOIs in previous step. The final decision now rests on evaluation of possible future collisions with VOIs. Different collision mitigation systems use different collision evaluation criteria. For example, (Noh, 2018) uses time to enter (TTE) i.e. time it takes for a vehicle to enter a certain intersection area. Similarly, (De Beaucorps et al., 2017) uses post-encroachment time (PET) i.e. time difference between first exit and last entrance into a collision zone. Works such as (Ayres et al., 2001) have employed time headway (TH) which is another collision avoidance criterion. In our approach we use the classical time to collision (TTC) metric used by most of the collision avoidance systems (Ward et al., 2014). Take Figure 5 as an example of checking collisions on ego's rollout. There is a trajectory intersect between ego A_0 and agent A_1 at time t_7 and $t_8^{'}$. Here t_n and $t_n^{'}$ where $n \in \mathbb{N}_0$ is the accumulated time cost at waypoint n if both vehicles move along their respective lanes with their current velocities V_i. For any given lane waypoint, the time cost is given by:

$$\begin{aligned} t_n &= \frac{D}{V_i} \\ TTC &= |t_n - t_n^{'}| \end{aligned} \tag{4}$$

Here, D is approximated distance of a vehicle from the specific waypoint and V_i is the velocity of ith vehicle. Figure 5 shows that if both agents follow same their current speed profile within their lane and we take dimensions into account, then actual collision between A_0 and A_1 happens few waypoints before lane intersection point i.e. at around t_5 and $t_6^{'}$. The ego vehicle has a critical safety box around it such that width of safety box is $w + 2 * lat_{safety}$ and length of safety box is $l + 2 * long_{safety}$ where l and w are length and width of the ego vehicle. The critical lateral and longitudinal distance from vehicle's center point is given by C_x and C_y. To make the final collision check, the safe lateral and longitudinal bounds should also be evaluated in addition to TTC. Let α and β be the lateral and longitudinal distance between agent and ego at any given future time step. The collision check including vehicle bounds can then be formulated as:

$$collision\_check(TTC, \alpha, \beta) = \begin{cases} 1, & \text{if } TTC < t_{thresh} \text{ and } |\alpha| < C_x \text{ and } |\beta| < C_y \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

Here t_{thresh} is the safety time threshold which is tuned after testing different yielding scenarios. These scenarios will be discussed in experiments section. Once the AV experiences a collision with VOI(s) in its future rollout, a control signal is sent to the behaviour state machine that triggers the yielding manoeuvre which is merely a deceleration command to the ego vehicle with stopping point being the yielding stop line

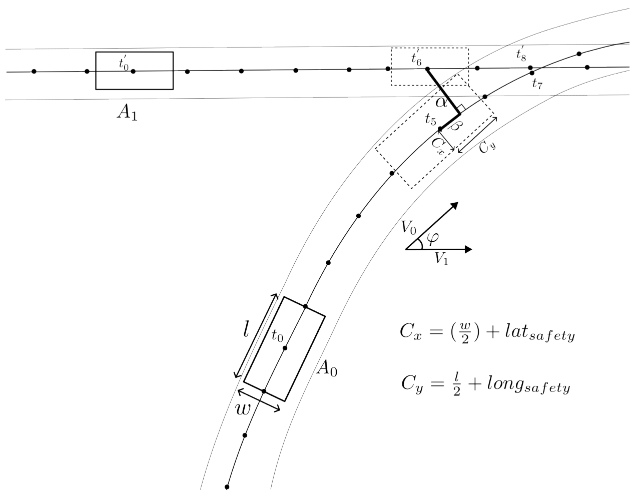


Figure 5: An illustration of collision checking where A_0 is ego vehicle and A_1 is other vehicle. Here, a potential collision is shown in future time-steps with dotted boxes

in AV's lane. The complete yielding approach is summarized in Algorithm 1. The code base of our work is available at `github.com/MahirGulzar/autoware_ut`.

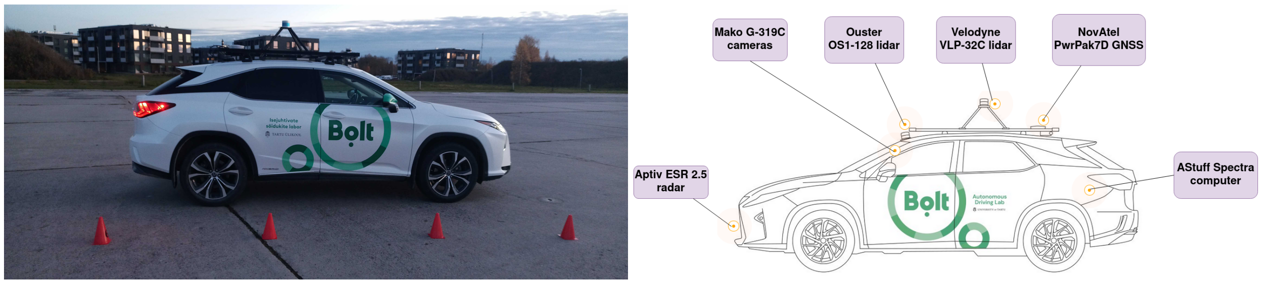
Algorithm 1 Yielding algorithm pseudocode

```
function YIELD($A, T, L, S$)    ▷ Where $A, T, L, S$ are vectors of agents, trajectories, lanes and stoplines
    $s \leftarrow GetClosestStopLine(L, S)$
    $d \leftarrow GetDistanceToStopLine(T_0, s)$    ▷ Distance to yielding stopline.
    if $d < d_{thresh}$ then
        $A \leftarrow FilterRouteFollowers(T_0, \{A_1, ..., A_n\})$    ▷ Filtering route followers, Refer to equation 1,2
        $N = length(A)$
        for $k \leftarrow 1$ to $N$ do
            $P_k \leftarrow intersect(T_0, T_k)$    ▷ Behavioural intersects of VUC with ego's route
            for $p$ in $P_k$ do
                if $distance(p_i, s) > 0$ then    ▷ If intersect is ahead of the stop line then proceed
                    $t \leftarrow TTC(T_0, p, A_0.v)$    ▷ AV's time-to-collision from intersection point
                    $t' \leftarrow TTC(T_k, p, A_k.v)$    ▷ Agent's time-to-collision from intersection point
                    $\alpha \leftarrow lateral\_offset(T_0, p)$
                    $\beta \leftarrow longitudinal\_offset(T_0, p)$
                    if $|t - t'| < t_{thresh}$ and $|\alpha| < C_x$ and $|\beta| < C_y$ then    ▷ Refer to equation 5
                        return True
    return False
```

5 Experiments

This section describes testing experiments of yielding algorithm on our AV.

5.1 Vehicle specification



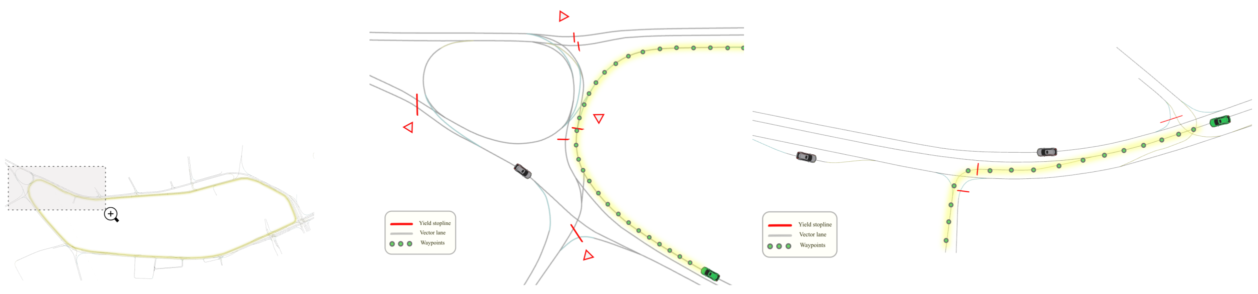
(a) (b)

Figure 6: ADL's AV sensor setup.

Table 1: Sensor specifications

Sensor	**Specification**
Ouster OS1-128 lidar	Vertical resolution: 128 channels Horizontal resolution: 512, 1024, or 2048 Horizontal FOV: 360° Horizontal angular resolution: min 0.17° Vertical angular resolution: 0.35° Range: 120 meters
Velodyne VLP-32C lidar	Vertical resolution: 32 channels Horizontal resolution: 900-3600 Horizontal FOV: 360° Vertical FOV: 40° Horizontal angular resolution: 0.1°-0.4° Vertical angular resolution: min 0.33° Range: 200 meters
2x Allied Vision Mako G-319C cameras	Resolution: 2064 × 1544 Max frame rate: 37.6 fps Left camera FOV: 25.3° Right camera FOV: 32.8°
NovAtel PwrPak7D-E2 GNSS device	Signal tracking: GPS, GLONASS, BeiDou, Galileo Accuracy: 1 cm + 1 ppm (in practice 5-10 cm) INS rate: max 200 Hz RTK service: ESTPOS°
Aptiv's multimode ESR 2.5 radar	Long-range Range: 174 m FOV: ±10° Update rate: 50 ms Range rate: -100-25 m/s
AStuff Spectra computer	CPU: Intel Xeon Memory: 32GB Storage: 1TB SSD GPU: NVIDIA RTX2080Ti Networking: 6x Gigabit Ethernet OS: Ubuntu 18.04

This study is carried out using AV of Autonomous Driving Lab (ADL) at University of Tartu, Estonia as shown in Figure 6a. The ADL research focuses on evaluating current state of autonomous driving using open-source (modular) solutions as well as data-driven (end-to-end) methods. In this study, the base autonomy of the AV uses ROS based Autoware.AI software as autonomy stack. The software package has built-in ready-to-use modules for localization, perception and planning etc. The AV is Lexus RX450h equipped with multiple sensors which are prerequisites for basic autonomy. The placement of these sensors are shown in Figure 6b whereas the sensor specifications are listed in Table 1. The object detection module in our pipeline makes use of two lidars and one radar sensor. The Velodyne VLP-32 is mounted on top of the car to cover 360-degree surrounding view, ouster OS1-128 is mounted on the top-front grill above the windshield to acquire a dense view of the frontal area and ESR 2.5 radar sensor is placed inside the front bumper. The raw point clouds generated by two lidars are processed using ground filtering to acquire non-ground objects which are later merged with radar detections. The merged detections are then tracked using tracking module. The tracked detections in the end are fed to OpenPlanner nodes within Autoware.AI for planning.



(a) (b) (c)

Figure 7: Map of Tartu demo route. Here, (a) is overall map view of the route, (b) shows an entry point of roundabout yielding area and (c) shows an unprotected turn yielding area.

5.2 Testing

As we intend to validate our implementation results on real vehicle, we make use of a demo track around University of Tartu's Delta Centre. This is illustrated in Figure 7a. Within this demo track, we have quite a few yielding areas. These areas are shown in Figure 7b and Figure 7c. Each yielding area has one or more entry points marked with yielding stop lines on the map. In our approach, the AV initiates evaluation for yielding when it encounters a yielding stop line within its future planned route. Since testing all possible yielding scenarios and yielding areas is impractical, for performing the experiments we reduce the search space to two yielding areas i.e. roundabout and an unprotected left turn. The demo track we intend to test the yielding algorithm has both roundabout and unprotected left turn in it. The yielding algorithm uses thresholding parameters which affect the decision-making discussed in Section 4.3. These parameters are tuned empirically by running several scenario experiments. Table 2 summarizes these parameters.

The experiments performed to evaluate the proposed yielding algorithm are carefully devised w.r.t. roundabout and intersection as shown in the experiment maps in Figure 7. Since safety is of the utmost importance for any autonomous vehicle, we evaluate the yielding algorithm in a hierarchical manner i.e. extensively experimenting with customized scenarios in simulation before moving towards real traffic.

5.2.1 Simulated obstacles

The very first step of our experiments starts with simulated obstacles. These scenarios are devised from easy to extreme difficulty in simulation. The idea here is to initiate testing in a step-by-step manner so that success and failures can be identified and yielding algorithm can be tuned accordingly. Table 3 gives a detailed summary of customized testing scenarios for experiments performed with simulated obstacles in different yielding areas.

Table 2: Empirically tuned yielding decision making parameters

Parameter	**Value**	**Description**
lat_{safety}	0.5m	Additional horizontal safety box distance.
$long_{safety}$	3.5m	Additional vertical safety box distance.
C_x	$(\frac{ego\_width}{2}) + lat_{safety}$	Critical lateral distance from other vehicle.
C_y	$(\frac{ego\_wheel\_base\_length}{2}) + (\frac{ego\_length}{2}) + long_{safety}$	Critical longitudinal distance from other vehicle.
t_{thresh}	11s	Time-to-collision threshold, below this ego should yield for priority vehicle. This resonates well with max speed of the ego vehicle i.e. $40km/h$ or $11.11m/s$ and $1m/s^2$ smooth deceleration value.
lat_{skip}	20m	Trajectory point's lateral skip distance, beyond this, a trajectory point is not evaluated as collision point. (Overlapping lanes should be very close anyway but this value roughly accounts for irregular overlaps of lengthy vehicles such as Trucks and busses.)
max_lane_dist	2m	Maximum perpendicular distance of VUC from ego's route after which it should be evaluated for yielding. Setting this value too low would mean that vehicle's drive strictly on lane center which is not the case in real world. On the other hand, setting this value too high could ignore potential vehicles to yield merging from other lanes to ego's lane which is dangerous. For context, refer to equation (1).
$max\_yaw\_deviation$	15^o	Maximum yaw angle difference between obstacle's heading and ego's route. If heading angle of a VUC's projection on ego's route is beyond this value then we should assume that VUC is not in our path and treat it as potential vehicle to yield. Setting this value too low would mean that vehicle's heading strictly align with lane headings which is not the case in real world. On the other hand, setting this value too high could ignore potential vehicles to yield merging from other lanes to ego's lane which is dangerous. For context, refer to equation (2).
$horizon\_dist$	$200m$	Yield stop line discovery distance. Can be less than $200m$ but shouldn't be less than $61.21m$ for smooth braking with max speed of $40km/h$ and $1m/s^2$ deceleration value.

Table 3: Scenarios of roundabout and unprotected left turn navigation in demo route

Scenario	**Description**
1	Approaching yielding area, No leading/trailing vehicle. At Least 1 moving vehicle to yield.
2	Approaching yielding area, No leading/trailing vehicle. At Least 2 or more moving vehicles to yield.
3	Approaching yielding area, No trailing vehicle. At least 1 leading vehicle ahead of us before the yielding stop line and at Least 1 moving vehicle to yield.
4	Approaching yielding area, No trailing vehicle. At least 1 leading vehicle ahead of us before the yielding stop line and at Least 2 or more moving vehicles to yield.
5	Approaching yielding area, At least 1 leading vehicle ahead of us before the yielding stop line, At least 1 trailing vehicle behind us and at least 2 or more moving vehicles to yield.
6	Approaching yielding area, At least 1 leading vehicle ahead of us before the yielding stop line, At least 1 trailing vehicle behind us and at least 2 or more static vehicles to yield in the forming a traffic congestion.

Table 4: Evaluations of simulated scenario obstacles for roundabout entry in demo route

Scenario	**No. of Tests**	**Successes**	**Unnecessary Braking**
1	12	12	0
2	12	12	0
3	12	12	1
4	12	12	0
5	12	12	0
6	12	12	0
Total	72	72	1

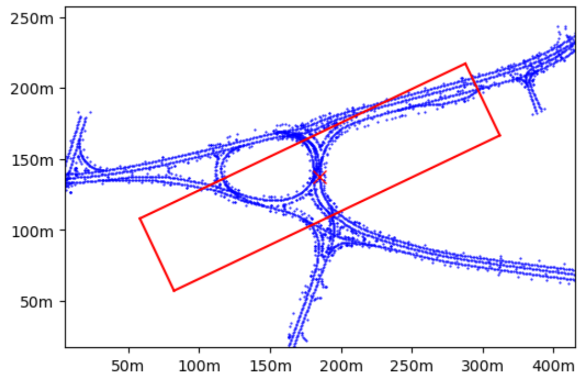
Table 5: Evaluations of simulated scenario obstacles for unprotected left turn in demo route

Scenario	**No. of Tests**	**Successes**	**Unnecessary Braking**
1	12	12	0
2	12	12	0
3	12	12	0
4	12	12	0
5	12	12	0
6	12	12	0
Total	72	72	0

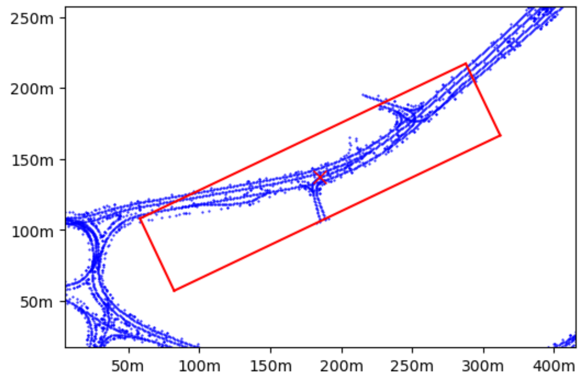
Table 4 and Table 5 lists the evaluations performed for simulated obstacles. Each scenario comprises 12 experiments where different speed profiles and distances were used for simulated obstacles. The success of an experiment here is the collision-free navigation of the yielding area. Additionally, the unnecessary braking shows the count of intervals in which ego applied phantom braking.

5.2.2 Simulated obstacles on test-site

To further evaluate the performance of the yielding algorithm, we move to the test-site and evaluate simulated virtual obstacles with actual ego vehicle for yielding. The test-site is 70×270 meters empty parking-lot area. We take yielding areas from the map of the demo route and overlay them on the test-site to construct similar testing environment. This is illustrated in Figure 8. The experiments performed on the test-site use exactly the same simulated obstacles setup described in Table 3. The only difference is that the real ego vehicle is performing the navigation here while a safety driver engages the autonomy. In addition to collision free navigation of the yielding area, the success here also includes zero disengagements from the safety driver during the yielding manoeuvre. Table 6 and Table 7 summarizes results of these experiments.



(a) Roundabout



(b) Unprotected-left turn

Figure 8: Map overlay over test-site, here rectangle box shows the test-site parking lot where yielding stopline under testing is located at the center point of the box.

Table 6: Evaluations of simulated scenario obstacles for roundabout entry on test-site

Scenario	No. of Tests	Successes	Unnecessary Braking
1	2	2	0
2	2	2	0
3	2	2	1
4	2	2	0
5	2	2	0
6	2	2	0
Total	12	12	1

Table 7: Evaluations of simulated scenario obstacles for unprotected left turn yield on test-site

Scenario	No. of Tests	Successes	Unnecessary Braking
1	2	2	0
2	2	2	0
3	2	2	0
4	2	2	0
5	2	2	0
6	2	2	0
Total	12	12	0

For each scenario discussed in Table 3 we performed 2 experiments on the test site. Similar to the results we got from the simulation in previous section, the yielding algorithm successfully navigated the yielding areas without interventions and collisions in both yielding areas.

5.2.3 Real obstacles on test-site

For the next step in evaluation, we perform yielding experiments with a real obstacle vehicle on test site. In addition to ego vehicle, there is an actual vehicle driven by a human driver inside yielding area with priority. We perform multiple experiments by asking the human driver to navigate the yielding area with different speed profiles at his own convenience. The ego vehicle starts its navigation in non-priority area is given a goal point that can only be reached by navigating through the yielding area. The results of the experiments performed are listed in Table 8.

Table 8: Evaluations of real obstacles for roundabout and unprotected left turn yield on test-site

Yielding Area	No. of Tests	Successes	Unnecessary Braking
Roundabout	12	11	0
Unprotected Turn	12	4	0

The real obstacle navigating the yielding area was successfully given right of way for roundabout but partially failed in unprotected turn as shown in the results from Table 8. These failures are analyzed and discussed in later sections.

5.2.4 Real traffic

To test the yielding functionality in real traffic, we first record data from the two yielding areas we discussed in our previous experiments. We replay the recorded data and run the autonomy pipeline over it. The idea here is to observe what decisions the algorithm will take without actuating the decisions on ego car. This evaluation gives a useful insight into what should be expected by the algorithm in real traffic. The results of evaluations made on the recorded data are shown in Table 9. Similar to previous results, we saw significant failures in case of unprotected turn for real traffic.

Table 9: Evaluations of real traffic for roundabout and unprotected left turn yield in demo route

Yielding Area	No. of Tests	Successes	Unnecessary Braking
Roundabout	12	10	4
Unprotected Turn	12	3	0

Table 10: Comparison of some of the related studies which implement yielding behaviour.

Works	**Generic Scenarios**	**No. of tests**	**Criterion**	**Real data**	**Intersections**	**Scope**	**Open-source**
(Bey et al., 2021)	8 (Distance variations)	140	Collision	No	Roundabouts	Roundabouts	No
(Masi et al., 2020)	N/A	∼18 (1 hour of simulation tests, 200-400 seconds per test)	Interdistance, Insertion rate, crossing time	Yes	Roundabouts	Roundabouts	No
(Rodrigues et al., 2018)	9 (Speed & distance variations)	27	Near miss, Collision	No	Roundabouts	Roundabouts	No
(Liu et al., 2015)	N/A	N/A	Failure rate, Travelling time	Yes	Roundabouts T junction	Roundabouts, T junction	No
Ours	**6 (Speed, distance & traffic variations**	**192**	**Collision, Driver Take overs Unnecessary braking**	**Yes**	**Roundabouts, Unprotected Left Turn**	**Roundabouts Unprotected Left Turn, Unprotected U-Turn, Highway-merge**	**Yes**

Table 10 shows a comparison of the some of the related works. It can be seen that our work had most number of scenario variations. Since the experiments were conducted on both simulation and real traffic data, the number of tests are also significantly higher making our experimentation more reliable. Additionally, it can be seen that there isn't one standard metric of evaluation. A combination of metrics were employed by these works for evaluation.

5.3 Risk assessment

Testing AV in field comes with risks. Following are some of the identified potential risks (along with their mitigation) associated to our real world testing setup.

5.3.1 Risk of incorrect decision (False negative)

Both test site and real traffic experiments are prune to incorrect algorithmic decision. A false negative by the algorithm can risk the AV to experience collision with vehicle of higher priority. On the test site with controlled environment, both obstacle vehicle and ego vehicle were in constant communication. A low speed profile of both vehicles was always agreed prior to conducting the test. The autonomy engagement on ego vehicle was prompted by the computer operator on ego vehicle, in case of unsuccessful attempt or potential hazard the safety driver always takes over full control by pressing brake pedal or moving the steering. While conducting field trials on test site, cones were setup to guide obstacle vehicle to follow map lanes. In real traffic, the autonomy was never engaged. The computer operator observes and records the algorithm decision in shadow mode (no actuation).

5.3.2 Risk of unnecessary braking (False positive)

A false positive by the algorithm could lead to both smooth and sudden unnecessary braking of the ego vehicle. This situation is potentially hazard free on test site since the tests were conducted with low speed profiles. Additionally, since we only had one obstacle vehicle on test site which was not ego's route follower, sudden braking of ego vehicle had no practical impact on obstacle's behaviour. On the other hand, in real traffic, as all behaviours were observed in shadow mode so unnecessary braking was never actuated by the ego vehicle.

5.3.3 Risk of stack failure

Just like any other computer program, AV's autonomy stack can also fail at any given point. This failure can occur in module level e.g. localization failure, perception failure, controller failure etc. Stack failure could also occur in the form of node crashes, sensor crash, bit flips etc. Both computer operator in AV and safety driver keenly observe vehicle's behaviour. Safety driver disengages autonomy whenever an unplanned event occurs.

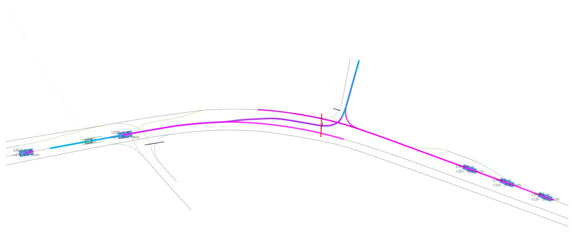
5.4 Discussion

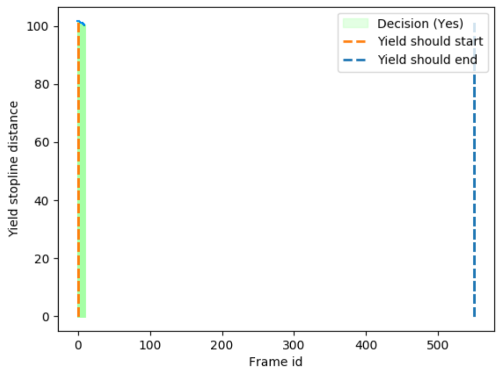
This section discusses the experiments performed in previous sections qualitatively and possibility of extending the yielding algorithm to accommodate more complex intersections and yielding areas in future.

5.4.1 Qualitative analysis

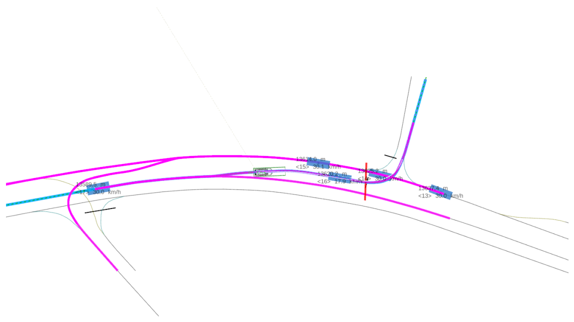
For analyzing an experiment we take frame snapshots of the scene at starting, middle and end of the experiment. Each snapshot is followed by a graph that plots algorithm's yielding decision until the ego crosses the yielding stop line.

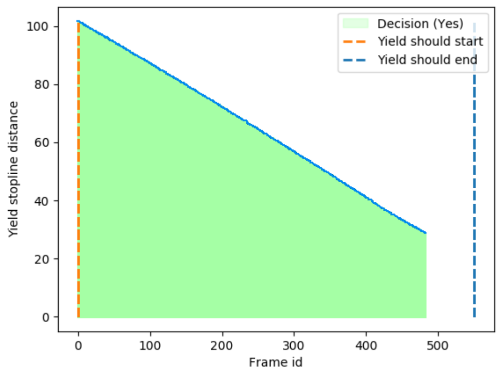
Simulated obstacles: An example of unprotected turn scenario-5 of simulated objects is shown in Figure 9. In Figure 9a the ego vehicle starts approaching the unprotected left turn intersection with a leading and a trailing vehicle. There are 3 vehicles with priority approaching the yielding area from the other side. Figure 9b shows the current distance of the ego vehicle to the yield stop line. Additionally, the graph also shows that from a human's perspective when should ego vehicle yield. These bounds are shown as interval (dotted vertical lines) between frames 0 to 550. The highlighted green area under the curve shows the true decision by the yielding algorithm i.e. the algorithm says that ego should come to a full stop before the yielding line. As the scene unfolds, we see final results in Figure 9e and Figure 9f. The ego vehicle successfully navigated the yielding area and gave the right of way within the desired yielding interval.


(a)

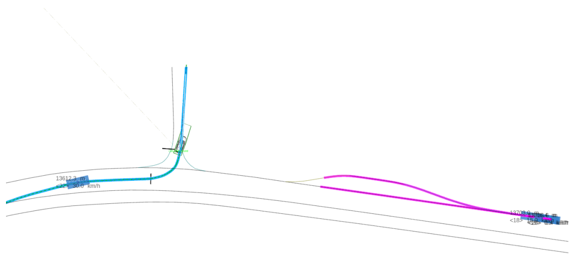


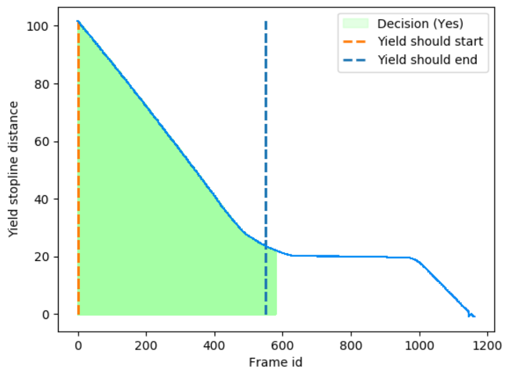
(b)


(c)



(d)

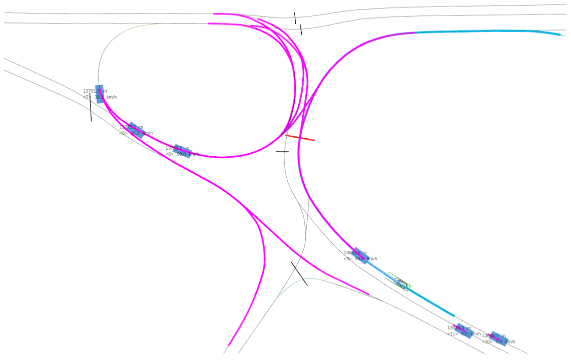

(e)



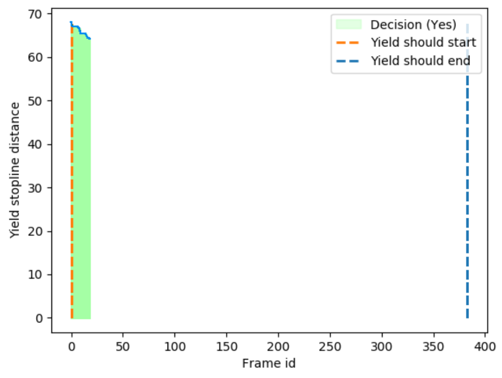
(f)

Figure 9: Unprotected-left turn simulation scenario-5. On left is the top-down view of the scenario whereas on right is the associated graph that shows yielding decision w.r.t. stop line distance and frames.

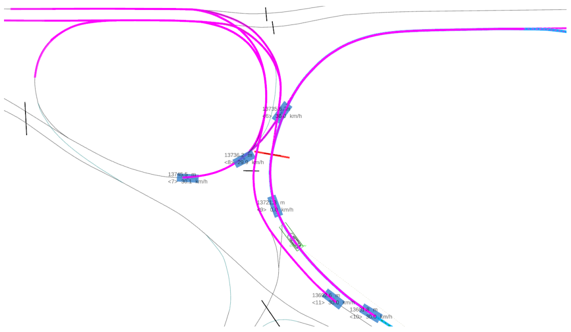
Similar to Figure 9, Figure 10 shows the analysis of decision-making in a roundabout entry. Here, we have a leading vehicle and trailing vehicle behind the ego vehicle and there are 3 vehicles within the roundabout which have priority. The three top-down snapshots are taken at the start, middle and end of the scenario. The ego vehicle starts at a distance of approximately 70 meters away from the yielding stop line. The final graph in Figure 10f shows that the ego vehicle successfully navigated the yielding area without any collisions and within the yielding bounds. The final graph between frames 180 and 400 shows a point where the leading vehicle is giving way to the priority traffic and yield decision is positive. Even though there is a vehicle ahead of us, the algorithm still evaluates ego's rollout in parallel and knows that the ego should yield and stop before the stop line if the leading vehicle decides to proceed aggressively.



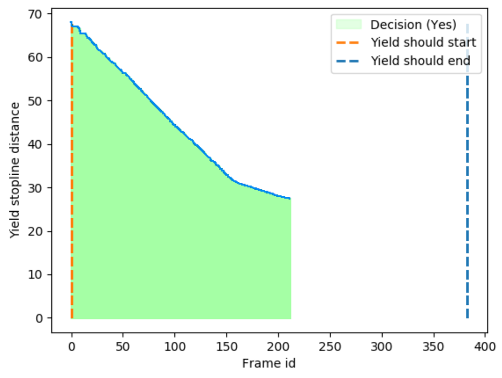
(a)



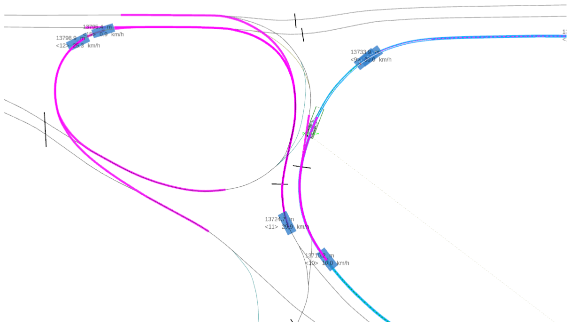
(b)



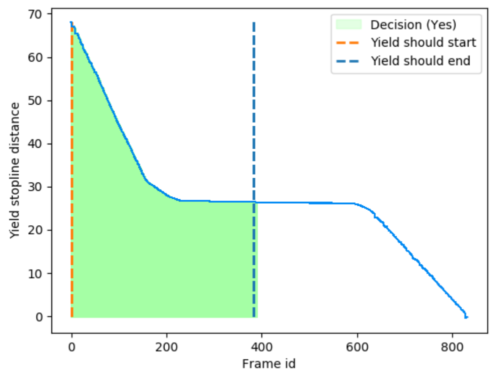
(c)



(d)



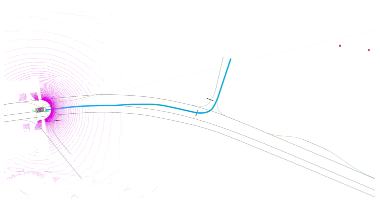
(e)



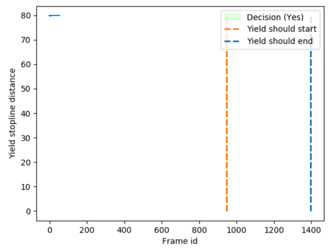
(f)

Figure 10: Roundabout entry simulation scenario-5. On left is the top-down view of the scenario whereas on right is the associated graph that shows yielding decision w.r.t. stopline distance and frames.

Real obstacles: We performed some of the experiments with real obstacles on test site. An example of unprotected turn is shown in Figure 11. Here, although the ego vehicle successfully navigated the yielding area without safety driver intervention; the overall yielding manoeuvre was a failure in terms of safety. The yielding decision graphs show that within the desired interval of yielding, the algorithm failed to produce a positive yield decision most of the time and braking was applied at the very end. This happened because the perception pipeline failed to detect the obstacle vehicle until the other vehicle came very close. The limitation also came from the sensors too. For example, a radar detects moving obstacles easily even if they are located 100 meters away but in this case, the other vehicle is not located longitudinally in a straight line of sight of the frontal mounted radar but is rather offsetted due to a curve near the yielding area on the map. Additionally, for object detection using LiDAR sensor, we rely on ground removal and clustering based approach which also failed in this case to detect the approaching vehicle within 60 meters range. Similar to unprotected turn, another example of test-site experiment is shown in Figure 12, here the yielding area is a roundabout and ego successfully yielded within the desired interval without safety driver's intervention. It is to be noted that this experiment was initiated near the yielding stop line due to limitation of the parking lot area space and the overlayed map size this can also be observed in Figure 8a.

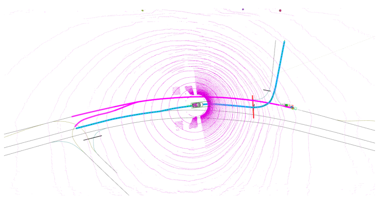


(a)

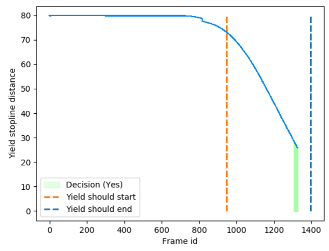


(b)

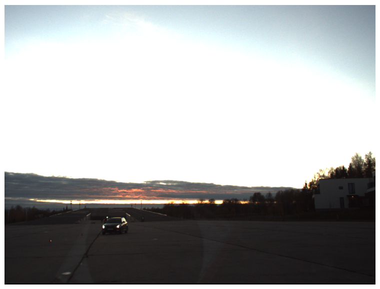
(c)



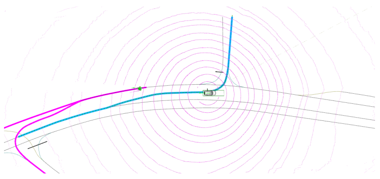
(d)



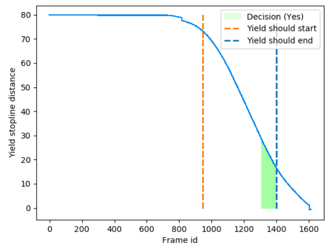
(e)



(f)



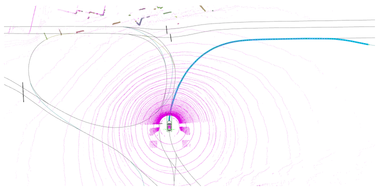
(g)



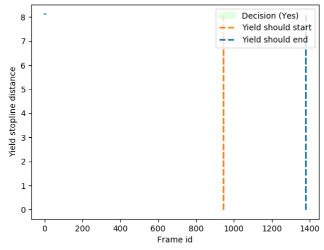
(h)

(i)

Figure 11: An example of unprotected turn with real obstacle on test-site. On left is the top-down view of the ROS rviz visualizer with sensor data, the middle column shows the associated graphs of yielding decision w.r.t. to stopline distance and frames whereas the column on right shows camera image taken from frontal camera mounted on the ego vehicle.

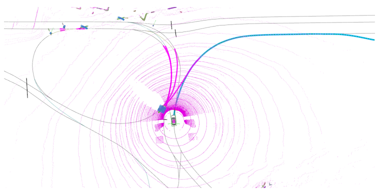


(a)

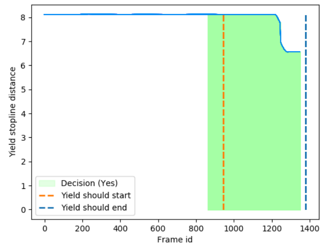


(b)

(c)

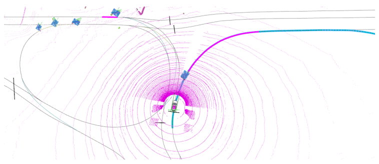


(d)

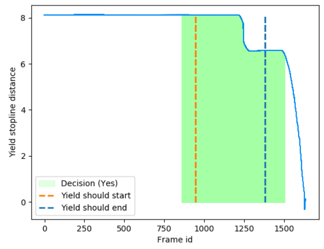


(e)

(f)



(g)

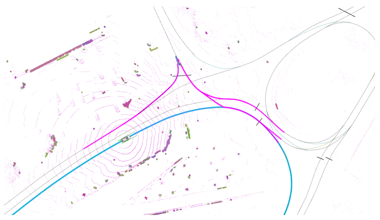


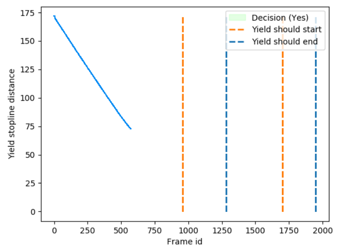
(h)

(i)

Figure 12: An example of a roundabout entry with real obstacle on test site. On left is the top-down view of the ROS rviz visualizer with sensor data, the middle column shows the associated graphs of yielding decision w.r.t. to stop line distance and frames whereas the column on right shows camera image taken from frontal camera mounted on the ego vehicle.

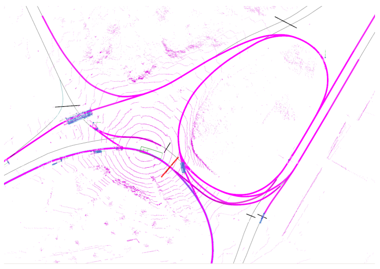
Real traffic: In our final analysis we take two examples of real traffic. Figure 13 shows an example of roundabout entry from one side of the entry points. There are two intervals in the graph where yielding algorithm should give positive decision. Here, the algorithm successfully gave positive decision in these intervals. In first interval, for a brief number of frames, the algorithm gave false negative decision. This again as described in previous example is the result of object mis-detection. A normal human driver would have seen other vehicle but our current perception pipeline failed to detect it for a brief moment.

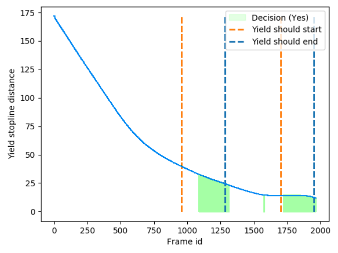

(a)



(b)

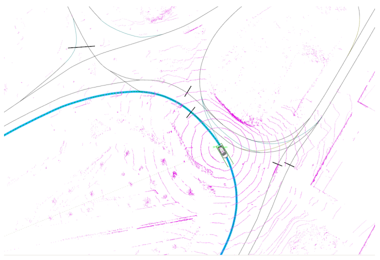
(c)

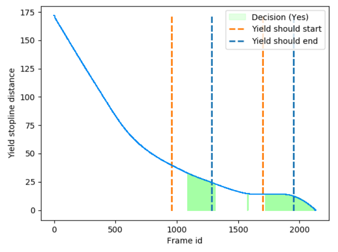

(d)



(e)

(f)


(g)



(h)

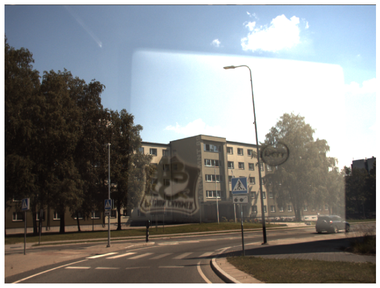
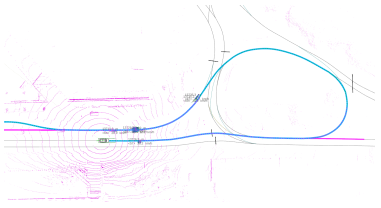
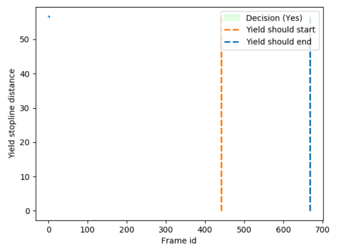

(i)

Figure 13: An example of roundabout entry from one of the entry points in demo route. On left is the top-down view of the ROS rviz visualizer with sensor data, the middle column shows the associated graphs of yielding decision w.r.t. to stop line distance and frames whereas the column on right shows camera image taken from frontal camera mounted on the ego vehicle.

Figure 14 shows an example of roundabout entry from another side of the roundabout. In this example, there is one interval of yielding and the algorithm successfully gave positive decision within this yielding interval. The graphs also shows minor unnecessary braking before the yielding intervals which were caused by false positives in perception pipeline.

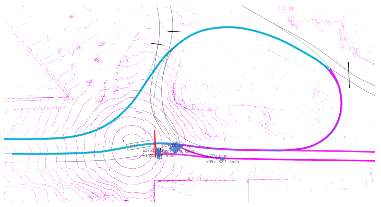


(a)

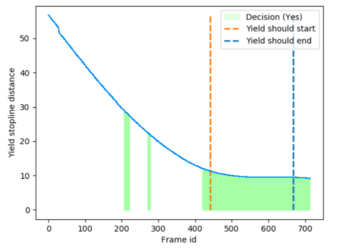


(b)

(c)

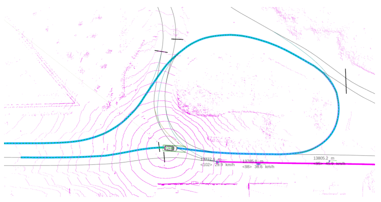


(d)

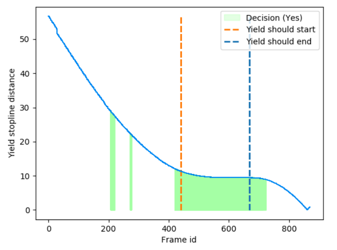


(e)

(f)



(g)



(h)

(i)

Figure 14: An example of roundabout entry from another entry point in demo route. On left is the top-down view of the ROS rviz visualizer with sensor data, the middle column shows the associated graphs of yielding decision w.r.t. to stop line distance and frames whereas the column on right shows camera image taken from frontal camera mounted on the ego vehicle.

5.4.2 Other potential yielding intersections

As discussed earlier, it is not feasible to test all types of yielding intersections on both simulation and real world. However, we can list number of potential yielding areas where our proposed method should theoretically work. Authors advise high caution before directly employing the yielding methodology in real world traffic. A rigorous testing pipeline similar to the hierarchical testing we did in this work should be followed before using the methodology on any other test data.

High-way merge : An example of high-way merge is illustrated in Figure 1a. Our proposed approach should work in this yielding area as this yielding area can be directly compared with Figure 5 of the decision making section where we explicitly explained that a positive yielding decision is made when the ego vehicle's lane merges into another high priority lane in future. For details kindly refer to section 4.3.

Unprotected u-turn : An example of unprotected u-turn is illustrated in Figure 15. When our proposed method is applied here, following should be the output of the decision maker. ***Filtering:*** Agent A_1 will be

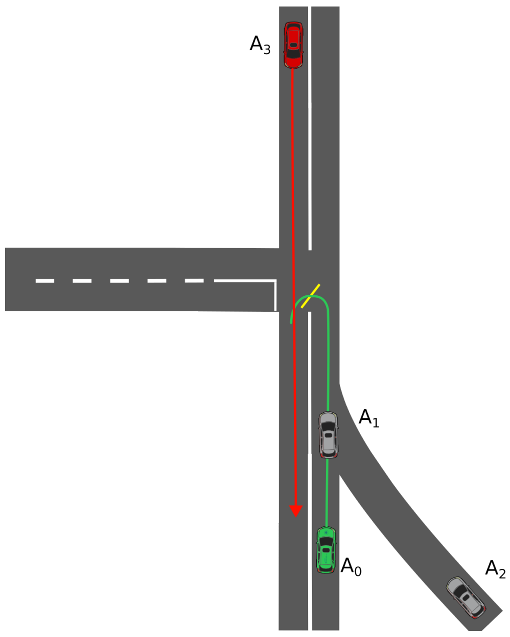


Figure 15: An illustration of decision making in an unprotected u-turn. Here, green vehicle A_0 is ego vehicle and red vehicle A_3 is potential vehicle to yield to. The yellow marking in ego's path is the yielding stop line for the turning lane.

counted as ego i.e. A_0's route follower whether it goes for a u-turn or proceeds straight. Thus A_1 will be filtered out from potential vehicles to yield list. A_2 and A_3 are not ego's route follower until they both merge into ego's planned path in future so both will be passed as potential vehicles to yield to the behavioural intersects step. ***Intersects:*** From agents A_2 and A_3, agent A_2's predicted trajectory intersects ego's path before yielding marked area. Hence, A_2 will be removed from the potential yielding vehicles list. On the other hand A_3 remains in the potential yielding list since its future trajectory intersects lies after the yielding stop line. ***Decision:*** Following the decision making parameter values, since A_3 is falls under t_thresh from Table 1 and has potential collision with ego in future, the decision maker will send positive control command to the state machine triggering the YieldStop state (followed by a YieldWait if required) bringing the ego vehicle to a stop until the priority vehicle merges into ego's future path.

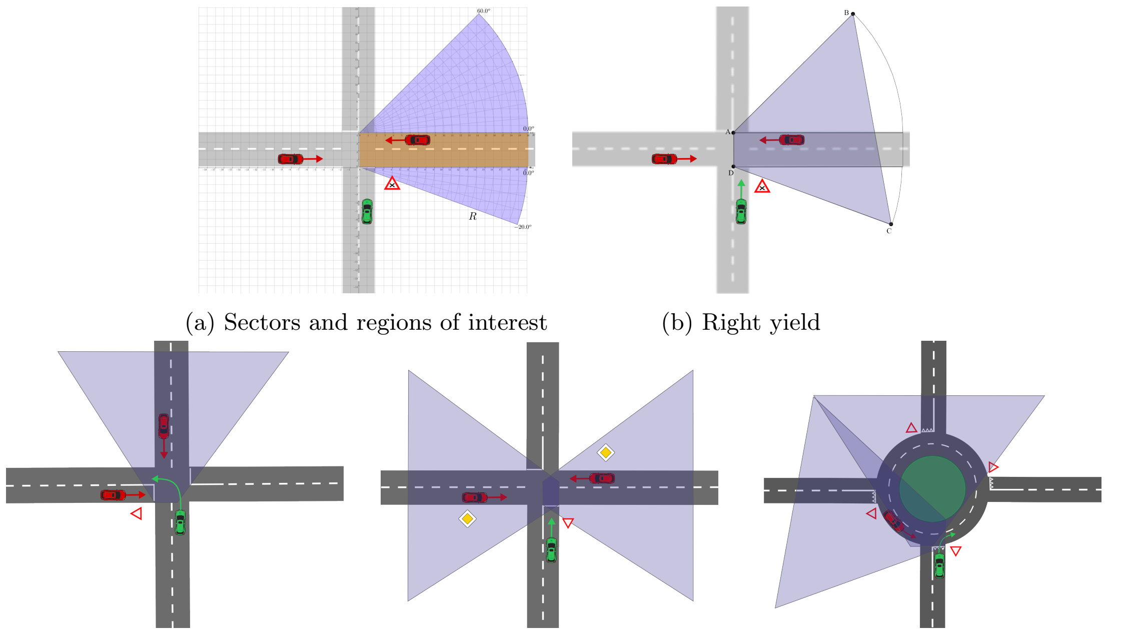
5.4.3 Future work

In above experiments, the performance of the yielding algorithm is evaluated only on an unprotected turn and a roundabout. A yielding area can have more complex intersections where filtering steps explained in section 4.2 are insufficient, they require attention maps or regions of the yielding area to access which vehicles have priority based on direction of movement of the traffic. For this purpose, we propose a future enhancement of the algorithm that identifies and filters vehicles in ROIs.

There are special cases in daily traffic in which an AV should only yield for traffic coming from a specific direction of the give-way area. For instance, while approaching 4-way intersection with yielding sign, the AV should only yield to traffic coming from left or right. For this purpose, we generalize these special cases into following categories.

- Forward yield
- Left yield
- Right yield

- Backward yield
- Compound yield



(a) Sectors and regions of interest (b) Right yield

(c) Forward yield (d) Compound yield (left and right) (e) Compound yield (left and forward)

Figure 16: Illustration of highlighting give-way areas using ROIs according to priority traffic. Here AV is denoted by green color and red vehicles are vehicles with priority.

We establish two polar grids and a Cartesian grid around the yielding stop line such that one polar grid originates from yield line and the other originates at some x-y offset depending on the category of yield. The intention here is to highlight areas of interest. Figure 16a illustrates this grid generation. Here, AV should be yielding for right side traffic. The Cartesian coordinates boundary is shown in orange highlighted area. The polar grids originate from upper-left and bottom-left corner of orange rectangle. These polar sectors are highlighted in purple. The sector size here depends on the radius and angle intervals. For example, the upper sector has angle interval from 0 to 60 degrees and lower sector has angle interval from 0 to 20 degrees where radius is 20 meters. It is to be noted that these values are shown as an example and can be tuned and generalized to cover specific regions of the yielding areas. Instead of filtering through 3 different regions for one-directional yield, we can simplify the shape of regions of interest. Let, ABCD be the corner points of the highlighted regions as shown in Figure 16b. Applying convex hull gives an irregular quadrilateral. This irregular quadrilateral can be further simplified into a trapezoid if the angular intervals for upper and lower polar sectors are equal as illustrated in Figure 16c. Region of interest (ROI) filtering can be formally defined as follows.

$$\mathcal{A}^* = \{x : x \in \mathcal{A} \wedge \sum_{n=1}^{k} in\_polygon(R_n, x) > 0\} \tag{6}$$

Here, k represents total number of ROI's i.e. R_n is nth ROI under processing. The function $in\_polygon$ checks if an agent x is inside ROI polygon. $\mathcal{A}^*$ is set of updated agents after applying ROI filter. The ROI filtering is meant to be an optional step of filtering that is only invoked in special cases when there is a condition on direction of yielding. For example, Figure 16b shows a scenario in which an AV should give way to right-side traffic only (notice give-way to the right sign) making it right yield scenario. Figure 16c shows a scenario in which an AV is about to perform unprotected left turn and should pay attention to oncoming traffic only making it forward yield scenario. Lastly, Figure 16d shows a scenario in which an AV should give way to both right and left side traffic which has priority over us (notice the priority road signs) but not necessarily to the oncoming traffic. Figure 16e shows a roundabout with ROIs. Although in our above experiments we didn't require any ROIs for roundabout but given the road structure and geometry constraints, one can use ROIs in roundabouts too if necessary. Scenarios using multiple ROIs can be categorized as compound yield. It is to be noted that once implemented, ROI filtering is meant to be an optional step and can be replaced with more robust approaches in future iterations.

6 Conclusions

In this study, we proposed a generic open-source solution to yielding problem for an autonomous vehicle (AV). The yielding problem was formulated as decision making function. We proposed a three-step solution which is implemented in Autoware.AI (more specifically in OpenPlanner). To evaluate the decision-making of the proposed solution, we devised a four-level hierarchy of tests in two yielding areas on a track surrounding the University of Tartu Delta building. Rigorous tests were performed both in simulation (scenarios), test-site (70×270 meters parking-lot area) and real-world traffic for the two yielding areas i.e. roundabout and an unprotected turn. The evaluations showed that the decision-making for yielding works quite well in ideal perception. The ideal perception of the environment is non-trivial using off-the-shelf solutions. To handle the perception failures either a better object detection model is required or the speed profiles of the non-priority lanes should be reduced. This means that the ego vehicle should approach any yielding area with cautious speed. This will give the decision-making algorithm more time to get a better view of other traffic participants with priority. Reducing ego vehicle's lane speed profile near yielding intersection before actuating the yielding decision will also put less stress on the safety driver thus reducing premature takeovers.

Acknowledgements

This publication has been financed by the European Social Fund via "ICT programme" measure, and by collaboration project LLTAT21278 with Bolt Technologies. The authors would like to thank Tambet Matiisen, Edgar Sepp, Muhammad Zain Bashir and Kertu Toompea for assisting in code reviews, experiments and test drives.

References

Ayres, T., Li, L., Schleuning, D., and Young, D. (2001). Preferred time-headway of highway drivers. In *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No. 01TH8585)*, pages 826–829. IEEE.

Bey, H., Sackmann, M., Lange, A., and Thielecke, J. (2021). Pomdp planning at roundabouts. In *2021 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)*, pages 264–271. IEEE.

Buehler, M., Iagnemma, K., and Singh, S. (2009). *The DARPA urban challenge: autonomous vehicles in city traffic*, volume 56. springer.

Carballo, A., Wong, D., Ninomiya, Y., Kato, S., and Takeda, K. (2019). Training engineers in autonomous driving technologies using autoware. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 3347–3354. IEEE.

Convention on Road Signs and Signals (2006). Conv_road_signs_2006v_en. `https://unece.org/DAM/trans/conventn/Conv_road_signs_2006v_EN.pdf`. (Accessed on 10/29/2022).

Darweesh, H., Takeuchi, E., Takeda, K., Ninomiya, Y., Sujiwo, A., Morales, L. Y., Akai, N., Tomizawa, T., and Kato, S. (2017). Open source integrated planner for autonomous navigation in highly dynamic environments. *Journal of Robotics and Mechatronics*, 29(4):668–684.

De Beaucorps, P., Streubel, T., Verroust-Blondet, A., Nashashibi, F., Bradai, B., and Resende, P. (2017). Decision-making for automated vehicles at intersections adapting human-like behavior. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 212–217. IEEE.

Hubmann, C., Becker, M., Althoff, D., Lenz, D., and Stiller, C. (2017). Decision making for autonomous driving considering interaction and uncertain prediction of surrounding vehicles. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 1671–1678. IEEE.

Hubmann, C., Schulz, J., Becker, M., Althoff, D., and Stiller, C. (2018). Automated driving in uncertain environments: Planning with interaction and uncertain maneuver prediction. *IEEE transactions on intelligent vehicles*, 3(1):5–17.

Kato, S. (2017). “autoware.

Krajzewicz, D. (2010). Traffic simulation with sumo–simulation of urban mobility. In *Fundamentals of traffic simulation*, pages 269–293. Springer.

Li, G., Li, S., Li, S., Qin, Y., Cao, D., Qu, X., and Cheng, B. (2020). Deep reinforcement learning enabled decision-making for autonomous driving at intersections. *Automotive Innovation*, 3(4):374–385.

Lin, X., Zhang, J., Shang, J., Wang, Y., Yu, H., and Zhang, X. (2019). Decision making through occluded intersections for autonomous driving. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 2449–2455. IEEE.

Liu, W., Kim, S.-W., Pendleton, S., and Ang, M. H. (2015). Situation-aware decision making for autonomous driving on urban road using online pomdp. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 1126–1133. IEEE.

Masi, S., Xu, P., and Bonnifait, P. (2020). Roundabout crossing with interval occupancy and virtual instances of road users. *IEEE Transactions on Intelligent Transportation Systems.*

Nilsson, J., Brännström, M., Fredriksson, J., and Coelingh, E. (2016). Longitudinal and lateral control for automated yielding maneuvers. *IEEE Transactions on Intelligent Transportation Systems*, 17(5):1404–1414.

Noh, S. (2018). Decision-making framework for autonomous driving at road intersections: Safeguarding against collision, overly conservative behavior, and violation vehicles. *IEEE Transactions on Industrial Electronics*, 66(4):3275–3286.

Rodrigues, M., McGordon, A., Gest, G., and Marco, J. (2018). Autonomous navigation in interaction-based environments—a case of non-signalized roundabouts. *IEEE Transactions on Intelligent Vehicles*, 3(4):425–438.

SAE Levels of Autonomy (2021). J3016_202104: Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles - sae international. `https://www.sae.org/standards/content/j3016_202104/`. (Accessed on 10/29/2022).

Singh, S. and Saini, B. S. (2021). Autonomous cars: Recent developments, challenges, and possible solutions. In *IOP Conference Series: Materials Science and Engineering*, volume 1022, page 012028. IOP Publishing.

Song, W., Xiong, G., and Chen, H. (2016). Intention-aware autonomous driving decision-making in an uncontrolled intersection. *Mathematical Problems in Engineering.*

TierIV and Apex.AI (2021). Tieriv and apex.ai. `https://www.youtube.com/watch?v=IuSsS-5VRNk&t=2m11s`. (Accessed on 10/29/2022).

Tollner, D., Cao, H., and Zöldy, M. (2019). Artificial intellgence based decision making of autonomous vehicles before entering roundabout. In *2019 IEEE 19th International Symposium on Computational Intelligence and Informatics and 7th IEEE International Conference on Recent Achievements in Mechatronics, Automation, Computer Sciences and Robotics (CINTI-MACRo)*, pages 000181–000186. IEEE.

Ward, J., Agamennoni, G., Worrall, S., and Nebot, E. (2014). Vehicle collision probability calculation for general traffic scenarios under uncertainty. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pages 986–992. IEEE.

Wei, L., Li, Z., Gong, J., Gong, C., and Li, J. (2021). Autonomous driving strategies at intersections: Scenarios, state-of-the-art, and future outlooks. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 44–51. IEEE.