SVM Enhanced Frenet Frame Planner For Safe Navigation Amidst Moving Agents

Unni Krishnan R Nair∗, Nivedita Rufus∗, Vashist Madiraju2, K Madhava Krishna1

Abstract—This paper proposes an SVM Enhanced Trajectory Planner for dynamic scenes, typically those encountered in on road settings. Frenet frame based trajectory generation is popular in the context of autonomous driving both in research and industry. We incorporate a safety based maximal margin criteria using a SVM layer that generates control points that are maximally separated from all dynamic obstacles in the scene. A kinematically consistent trajectory generator then computes a path through these waypoints. We showcase through simulations as well as real world experiments on a self driving car that the SVM enhanced planner provides for a larger offset with dynamic obstacles than the regular Frenet frame based trajectory generation. Thereby, the authors argue that such a formulation is inherently suited for navigation amongst pedestrians. We assume the availability of an intent or trajectory prediction module that predicts the future trajectories of all dynamic actors in the scene.

Fig. 1: Teaser, (figure best viewed at 400%)

I. INTRODUCTION

The research in the field of autonomous navigation is making significant progress everyday. Any autonomous system must interpret the sensor data to compute the trajectories for safe navigation in an environment. But the problem becomes more convoluted when the autonomous system shares its environment with humans. In such scenarios the autonomous system has to make assumptions on the motion of the humans based on the intent displayed. Many previous works assume that the autonomous system is aware of the participant with which it is interacting and its application is limited to service robots.

Works like [1][2] assumes that the humans in the environment are dynamic obstacles and tries to avoid them. Whereas the works like [3] propose a collaborative method where the humans and the autonomous system try to avoid each other. However these methods predict the human motions using a single motion model and it is not very effective as human motions are generally more unpredictable.

With the recent advent of LSTMs and GRUs for pedestrian and vehicular intent prediction, it becomes convenient and beneficial to formulate a motion planning framework amidst crowds for predictive path planning and trajectory generation. A trivial measure of safety while path planning would be the margin of separation between the obstacles and the ego-vehicle and none of the methods guarantee a maximum margin separation during all time steps. In this work, we present a intent integrated safety-first approach combining the learning based approach to predict the human intent with search based methods for reactive path planning and a margin maximization method using SVM to modify the waypoints. Inspired from residual learning, we predict the human intent using an existing deep learning methodology to model the motion information of each human. The position of each individual is then mapped to high dimensional feature space to get the spatial affinity, which is used to get the displacement prediction. These pixel coordinates are then transferred to world frame. We then use a soft margin SVM to find the support vectors that has the maximum margin of safety considering the future predicted position of each participant in the environment. These support vectors form the waypoints which form the control points for generating a Bézier curve which is both curvature continuous and has curvature bound. Over this we fit in a minimal jerk trajectory.

A. Contribution and Main Results

• We contribute through a novel SVM based control/waypoint generator that computes maximally separated waypoints with respect to all dynamic actors for a specified time horizon into the future.
• We show that the SVM enhanced Frenet frame planner achieves larger offset with respect to all actors thereby enhancing safety especially in the context of navigation in the presence of humans in an on-road setting.
• We successfully integrate this planner on a self driving car controlled by a commodity laptop indicating the computational feasibility and elegance of the proposal.

Fig. 2 presents the overview of the proposed approach.
II. RELATED WORK

Motion planning in the presence of dynamic obstacles is a well studied problem with various flavours. Model Predictive Control (MPC) formulations has been a popular theme [4] that has been extended to uncertainty based formulations [5]. Time scaling formulations that modulate velocities on highly non-linear curves without changing the original planned trajectories represent another paradigm [6],[7]. In a reactive multi-robot setting Reciprocal Velocity Obstacles and its variants have been popular [8],[9] especially amidst moving amongst pedestrians [10]. However to the best of our knowledge these approaches have not formulated their method as one that strives to maintain maximum margin separation between the ego vehicle and other moving obstacles in the scene thereby providing for safety. Also these approaches have not shown a real implementation on a car, while [10] does show it on a ground robot moving amidst pedestrians.

The current work differentiates by providing for a SVM layer that computes waypoints that are maximally separated from the dynamic actors in the scene. This waypoint generation layer can be integrated with any of the above mentioned frameworks such as MPC or with other popular trajectory generation methods[11],[12]. In this paper we show examples of how adding this waypoint generation layer provides for safer trajectories when used in tandem with the Frenet frame trajectory planner [12] that is quickly becoming a standard for on road navigation.

The overview of the method can be understood with the help of Fig. 2. The perception and prediction module are assumed to be available to us. We consider deterministic prediction of future trajectories through this module. The future locations predicted by this module is used by the SVM prediction which eventually is used by the SVM to get control points for n timesteps.

III. MAXIMUM DISTANCE AS A MEASURE OF SAFETY

For a collision to occur, the ego-vehicle and the pedestrian have to be at close proximity both laterally and longitudinally. A safe longitudinal and lateral separation between two vehicles moving with their respective velocities and accelerations can be computed as per [13]. The same notion can be extended to the situation wherein one party is the ego-vehicle and the other a pedestrian. Meeting these required conditions at all points of time ensures that the ego-vehicle is on a collision free course. Ensuring safe longitudinal separation is fairly easy to achieve by means of perception through which the separation between agents is gauged and necessary control actions can be executed (as simple as decelerating when separation is less than safe distance).

But in the case of lateral separation the solution is not as trivial, owing to the non-holonomic nature of the ego-vehicle. [13] also states that the safe distance to be maintained both laterally and longitudinally is directly proportional to the reaction time, hence maximizing this distance margin gives both parties more time to react, thereby maintaining safety.

IV. SUPPORT VECTOR MACHINES (SVM)

We use a soft margin support vector machine with rbf kernel (SVM) to find a safe path amidst obstacles as per [14]. Safety in our path is guaranteed in terms of maximum margin boundary provided by SVM. The formulation to find a path is given as follows.

Let $X = \{ (x_1, l_1), (x_2, l_2), \ldots, (x_n, l_n) \}$ be a set of training examples. Where $x_i = [x_i, y_i]^T$ are $x$ and $y$ coordinates sampled from obstacle locations in occupancy map and $l_i = \{+1, -1\}$ is the label assigned to each example. Labels are assigned based on obstacle’s location with respect to center line (left/right). A simple clustering algorithm like K-means proves adequate. The hyper plane separating two sets of labels in $X$ is given by $w^T x + b = 0$. The parameters $w = [w_1, w_2]^T$ can be obtained by solving the following optimization problem:

$$\begin{align*}
\text{minimize} & \quad \frac{1}{2} ||w||^2 \\
\text{subject to} & \quad l_i(w^T x_i + b) \geq 1, \quad i = 1, \ldots, n.
\end{align*}$$

Such a boundary is limited by two factors: (a) It doesn’t allow any mis-classification of training examples (b) It doesn’t allow non-linear separating boundary. To overcome limitation (a), we adopt [15] formulation to allow few mis-classifications. This leads to the problem:

$$\begin{align*}
\text{minimize} & \quad \frac{1}{2} ||w||^2 + C \sum_{i=1}^{n} \zeta_i \\
\text{subject to} & \quad l_i(w^T x_i + b) \geq 1 - \zeta_i, \quad i = 1, \ldots, n.
\end{align*}$$

Here, $\zeta_i$ is a slack variable and $C$ controls the relative importance given to maximum margin and mis-classifications. If $C$ is large, separating hyper plane prioritizes mis-classifications over margin and if $C$ is small, hyper plane allows mis-classifications. The dual of this problem can be written as,
maximize $\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j l_i l_j x_i^T x_j$  
subject to $0 \leq \alpha_i \leq C \quad i = 1, \ldots, n.$  
$\sum_{i=1}^{n}\alpha_i l_i = 0$  

This dual problem is convex and can be solved efficiently using interior point solvers. Now, limitation (b) can be tackled by incorporating kernel trick to project $x_i$ to higher dimensional space. This leads to kernel $\phi$ so that dual formulation in (Eq. 1) only deals with dot product between $x_i$ and $x_j$. So, we can define a RBF kernel $K(x_i, x_j)$ that return dot product in higher dimensional space.

$$K(x_i, x_j) = \exp\left(-\frac{||x_i - x_j||^2}{2\sigma^2}\right)$$

The equation of the boundary is given by

$$\sum_{i=1}^{n}\alpha_i l_i K(x_i, x) + b = 0$$

Where $b$ is given by,

$$b = l_k(1 - \zeta_k) - \sum_{i=1}^{n}\alpha_i l_i K(x_i, x_k) ; \quad k = \arg\max_i \alpha_i$$

This formulation is very much on the lines of the work of [14]. The result of the formulation is shown in (Fig.3)

$$\text{Fig. 3: RBF kernel SVM for path planning, (figure best viewed at 400%)}$$

V. OUR SVM FORMULATION FOR MAXIMUM-MARGIN PATH GENERATION

Our goal is to generate a maximum margin path as a seed to evaluate frenet frame trajectories in section IV. We generate such path in two steps.

A. SVM based way-points

At each time step $t_i$ we find a maximum margin boundary given by,

$$w_i^T x + b_i = 0$$  
(3)

Here $w_i$ and $b_i$ be parameters of boundary at $t_i$ and $x = [x_{ij}, y_{ij}]^T$, where $j = 1, 2, \ldots, n$ if there are $n$ dynamic actors (for example in Fig. 4 $n = 4$), $i = 1, 2, \ldots, m$ representing the $i^{th}$ time step. Fig. 4 shows the initial condition of the simulation. Let the initial positions of the actors be as indicated in the diagram as $(P_1, P_2, P_3, P_4)$ i.e the position of $P_1$ in the first time step, $t = 1$ is $[x_{11}, y_{11}]^T$ and the that of the car be at the origin. The formulation of the SVM according to the section IV we get a boundary with the maximum margin separation from the two classes. In order to have maximum progress along the path, we assume a constant forward speed of $v_{max}$ along the path. Using this velocity we sample points on the boundary as

$$\begin{bmatrix} S_x^i \\ S_y^i \end{bmatrix} = \begin{bmatrix} \hat{x}_0 + v_{max}(t_i - t_0) \\ \frac{-b_i - w_i^T y_{max}^i}{w_i^T S_x^i} \end{bmatrix}$$

(4)

Here, $S_x^i$ is the progress along x-axis with a speed of $v_{max}$ and $S_y^i$ is the y coordinate of boundary at $S_x^i$.

$$S = \{(S_x^1, S_y^1), (S_x^2, S_y^2), \ldots, (S_x^m, S_y^m)\}$$

(5)

Fig. 5 shows hyper line(yellow dashed line) complying to eq. 3 and the corresponding control points for that time step is given by $[S_x^i, S_y^i]^T$ according to Eq. 4, while Fig. 6 shows the same for subsequent time steps. This way the corresponding control points are mapped to the maximal margin separation for the entire time horizon.

$$\text{Fig. 4: The initial states for the actors in the scene, (figure best viewed at 400%)}$$
B. SVM trajectory vs SVM waypoints

Introducing non-holonomic constraints to the SVM path planning formulation increases the complexity and computational time to generate a SVM boundary. Instead we use these generated points as control points to form a cubic Bézier curve interpolation (for continuous curvature minimum required degree is three). This interpolation gives a smooth curve which is kinematically feasible.

C. Path through the waypoints

Our framework analytically interpolates the ordered waypoints with continuous curvature using cubic Bézier curves. This is inspired by the work[16],[7] and is of closed form expression which makes it computationally efficient and robust. From the set of waypoints derived above, we fit a Bernstein polynomial given by

\[
x(t) = \sum_{i=0}^{p} \alpha_i B(i, p, t) \tag{6}
\]

\[
y(t) = \sum_{i=0}^{p} \beta_i B(i, p, t) \tag{7}
\]

Where, \(B(i, p, t) = \binom{p}{i}(1-t)^{p-i}t^i\) \tag{8}

\[\mu = \frac{t - t_m}{t_m - t_0} \tag{9}\]

Here, \(\alpha_i\) and \(\beta_i\) are weights for Bernstein basis. These weights \(\alpha_i\) can be obtained by solving a set of linear equations from waypoints as given below:

\[
\begin{bmatrix}
B(0, p, t_0) & B(1, p, t_0) & \cdots & B(p, p, t_0) \\
B(0, p, t_1) & B(1, p, t_1) & \cdots & B(p, p, t_1) \\
\vdots & \vdots & \ddots & \vdots \\
B(0, p, t_m) & B(1, p, t_m) & \cdots & B(p, p, t_m)
\end{bmatrix}
\begin{bmatrix}
\alpha_0 \\
\alpha_1 \\
\vdots \\
\alpha_p
\end{bmatrix}
= \begin{bmatrix}
S_x^1 \\
S_x^2 \\
\vdots \\
S_x^m
\end{bmatrix}
\tag{10}
\]

Similar procedure can be used to obtain weights \(\beta_i\). This curve is shown in Fig. 7.

VI. TRAJECTORY GENERATION

The path we have obtained from the above interpolation is used as the center line. The longitudinal and lateral path planning are done separately and later superimposed to generate the safe and cost efficient trajectory. Quintic polynomials are used to generate trajectories with minimal jerk along the path. The generated trajectories are evaluated according to the proposed methods in [12] which ensures an optimal control method for reactive autonomous navigation.
ego-vehicle and $\ddot{x}$ is the acceleration profile of the ego-vehicle. The state space of the ego-vehicle is modeled as:

$$
\dot{X} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} X + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} U
$$

(11)

which is of the form of a double integrator and $U$ is the input.

The start constraint $X_s = (\dot{x}_s, \ddot{x}_s)$ and the goal constraint $X_g = (\dot{x}_g, \ddot{x}_g)$ are used to obtain the solution set of quintic polynomials by solving the following Eq. (12)

$$
X(t) = \begin{bmatrix} 1 & t & t^2 & t^3 & t^4 & t^5 \\ 0 & 1 & 2t & 3t^2 & 4t^3 & 5t^4 \\ 0 & 0 & 2 & 6t & 12t^2 & 20t^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix}
$$

(12)

where $a_n$, $n = \{0, 1, 2, 3, 4, 5\}$ are the coefficients and can be found from the initial and final constraints. Using this we first generate the lateral trajectories($d(t)$). The reference trajectory is $d_{ref} = 0$ which represents the center line. The limits of lateral displacement are set to the road boundaries. This property seamlessly integrates the road constraints while solving for trajectories. Next, we generate the longitudinal trajectories till the planning horizon along the reference trajectory($s_{ref}$) which is obtained from the previous section. By sampling the final position on the current lane with lateral deviations and also by varying the final time to reach the final position we generate a temporal and spatial grid. Then the longitudinal trajectories are overlaid on their respective lateral trajectories. The trajectories which lead to a collision are omitted. By back transforming we get the global coordinates $[x_1, x_2]$ from this grid of trajectories. The orientation $\theta(t)$ and velocity $v(t)$ are determined by the following equations as derived in [12]

$$
\theta = \theta_c + \tan \left( \frac{\dot{d}}{1 - \kappa_c \dot{d}} \right)
$$

(13)

Then $\Delta \theta = \dot{\theta} \cdot \theta_c$. Then the velocity($v$) profile is evaluated as

$$
v = \frac{(1 - \kappa_c \dot{d})}{\cos \Delta \theta} \dot{\dot{d}}
$$

(14)

VII. EXPERIMENTAL RESULTS

To evaluate the performance of the presented methodology we have tested it in several configurations. All the simulations were performed on Intel i7 processor @ 3.2 GHz clock speed. The execution time in the simulation for the presented approach was in the order of 10 ms for a reasonably populated area i.e. 20 human beings per 400 sq m (refer Fig. 9 and Fig. 3). The methodology is also validated on a real car (Mahindra e2O). The car is equipped with a velodyne VLP 16, zed stereo camera, GPS rtk and an IMU.

This vehicle is controlled by a drive-by-wire system which was developed in-house. A single pedestrian was used for the realtime evaluation of our method. It is also observed that the average minimum lateral distance maintained was always greater than of those from the standard reactive planner based methods. This comparison can also be validated from the realtime implementation as shown in Fig. 11 and Fig. 12 The following video shows our framework and the simulations in action. [link to video] Fig. 10 shows the difference in the trajectories taken by the car with and without our framework. It also highlights that the maneuver of our frame work starts much earlier owing to maintain the lateral distance. Fig. 13 shows the trajectory of the ego-vehicle with and without our framework. Our framework always strives to maintain maximum distance margin thus ensuring safety of the pedestrian. Fig. 14, 16, 15 show the reactive planner using the Bézier curve obtained from the new waypoints as the centre line for various obstacle configurations. Note :-

All figures in this page best viewed at 400%.
Fig. 11: With the proposed framework

Fig. 12: Without the proposed framework

Fig. 13: Centre line for ego-vehicle with and without our framework in space-time, (figure best viewed at 400%)

Fig. [14a](Legend in next page as [NOTE]) shows a sample scenario in which two pedestrians cross from either side of the ego-vehicle. Our Framework hands off the maximum separation waypoints to the path planner which then tries to generate minimum jerk trajectory along the waypoints. Fig. [14b] shows the trajectories of the vehicle with and without using our framework in 3D (space and time). Similarly Fig. [16] depicts the situation wherein two pedestrians are crossing the road from the same side and in front of the ego-vehicle. Fig. [15] depicts the situation wherein one pedestrian tries to cross from the front and the other from the back of the ego-vehicle. We see in all the three simulated cases that our framework offers a better margin of separation ensuring a safer path.
Fig. 14: Two people crossing from the opposite sides of the road in front of the car, (figure best viewed at 500%)

NOTE :- Legend for Fig. 14a, Fig. 15a, and Fig. 16a is as follows: red solid line represents center line without our framework, white dashed lines represents the trajectory followed by our car, green solid line represents center line with our framework, yellow dashed lines represents predicted trajectory of the pedestrians.

VIII. CONCLUSIONS

The above experiments show the proposed method’s ability to maintain maximum lateral distance margin to travel on the safest path possible. The modified waypoints account for the predictive paths of the pedestrians while the polynomial path planning ensures a reactive component in case of unexpected scenarios. Thus, our global path is always the safest one with alternative local paths being generated if any unexpected development occurs. The proposed method adds very little extra time cost to its inclusion and is robust with a very dense area of human beings. The same idea can be extended to other moving obstacles provided good and robust prediction of their states. It is also possible to use weighted samples while using K-means to cluster the space with different weights for different obstacle types. Other clustering methods can also be explored for the initial separation into classes.

REFERENCES

[1] S. Kim, S. J. Guy, W. Liu, D. Wilkie, R. W. Lau, M. C. Lin, and D. Manocha, “Brvo: Predicting pedestrian trajectories using velocity-space reasoning,” The International Journal of Robotics Research, vol. 34, no. 2, pp. 201–217, 2015.
[2] S. Choi, E. Kim, and S. Oh, “Real-time navigation in crowded dynamic environments using gaussian process motion control,” in 2014 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2014, pp. 3221–3226.

[3] P. Trautman, J. Ma, R. M. Murray, and A. Krause, “Robot navigation in dense human crowds: Statistical models and experimental studies of human–robot cooperation,” The International Journal of Robotics Research, vol. 34, no. 3, pp. 335–356, 2015.

[4] B. Zhou, W. Schwarting, D. Rus, and J. Alonso-Mora, “Joint multi-policy behavior estimation and receding-horizon trajectory planning for automated urban driving,” 05 2018, pp. 2388–2394.

[5] H. Zhu and J. Alonso-Mora, “Chance-constrained collision avoidance for mavs in dynamic environments,” IEEE Robotics and Automation Letters, vol. PP, pp. 1–1, 01 2019.

[6] A. K. Singh and K. M. Krishna, “Reactive collision avoidance for multiple robots by non linear time scaling,” in 52nd IEEE Conference on Decision and Control, Dec 2013, pp. 952–958.

[7] A. S. B. Kumar, A. Modh, M. Babu, B. Gopalakrishnan, and K. M. Krishna, ”A novel lane merging framework with probabilistic risk based lane selection using time scaled collision cone,” in 2018 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2018, pp. 1406–1411.

[8] J. van den Berg, Ming Lin, and D. Manocha, “Reciprocal velocity obstacles for real-time multi-agent navigation,” in 2008 IEEE International Conference on Robotics and Automation, May 2008, pp. 1928–1935.

[9] J. van den Berg, S. Guy, M. Lin, and D. Manocha, Reciprocal n-Body Collision Avoidance, 04 2011, vol. 70, pp. 3–19.

[10] Y. Luo, P. Cai, A. Bera, D. Hsu, W. S. Lee, and D. Manocha, “Porca: Modeling and planning for autonomous driving among many pedestrians,” IEEE Robotics and Automation Letters, vol. 3, no. 4, pp. 3418–3425, Oct 2018.

[11] L. Biagiotti and C. Melchiorri, Trajectory Planning for Automatic Machines and Robots, 01 2008.

[12] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, “Optimal trajectory generation for dynamic street scenarios in a frenet frame,” in 2010 IEEE International Conference on Robotics and Automation. IEEE, 2010, pp. 987–993.