Nat. Volatiles & Essent. Oils, 2021; 8(5): 1714 - 1725

NVEO
Natural Volatiles &
Essential Oils

Physics-Based Traffic Motion Techniques In Motion Planning Of Intelligent Autonomous Robots

[1]Dr.Raji Pandurangan, [2]Dr. Anandakumar Haldorai, [3]Dr. Mohd Ashraf, [4]Dr. Md. Zair Hussain, [5]Dr. Dinesh Kumar Singh, [6]Dr.Ram Subbiah

1 Associate Professor, Saveetha Engineering College, Thandalam Rajipandurangan0528@gmail.com
2 Professor (Associate), Department of Computer science and engineering, Sri Eshwar College of Engineering, Coimbatore, Tamil Nadu, India, 641202. anandakumar.psgtech@gmail.com
3 Associate Professor, Computer Science & Engineering, School of Technology, Maulana Azad National Urdu University, Hyderabad (TS). ashraf.saifee@gmail.com
4 Associate Professor, Information Technology, School of Technology, Maulana Azad National Urdu University, Hyderabad (TS). mdzairhussain@gmail.com
5 Assistant Professor, Department of IT, DSMNRU, Lucknow, UP, India. dineshsingh025@gmail.com
6 Associate Professor, Mechanical engineering Gokaraju Rangaraju institute of Engineering and Technology, Hyderabad. ram4msrm@gmail.com

ABSTRACT

We provide a motion strategic plan for an automated robot navigation pathway that takes into account both the ambiguity posed by the automated robot as well as the uncertainty posed by other vehicles involved. The ambiguity along the anticipated trajectory is estimated using Gaussian dispersion, as well as the future movement of vehicles involved is forecasted using such a regional planner. The ambiguity from relocation and handling is calculated for the mobile robot using Linear-Quadratic Gaussian (LQG) architecture. Due to direct ambiguity data feedback to the designer, our architecture helps a planner to prevent risky scenarios more effectively than existing safety evaluation techniques. In comparison to planning just using an LQG architecture, we show that our planner may yield safer paths.

Keywords: Automated robot, motion planning, physics-based traffic, Linear-Quadratic Gaussian (LQG).

1. INTRODUCTION

Tracked robots are first developed to aid navigate over several terrain situations, including ice, loose sand, muck, steep hills, rubble-strewn terrain, or any mixture of such (hence known as difficult ground) however, conventional wheeled robots would not be able to do so. Such robots are frequently the best choice for tasks requiring a substantial degree of traction, like moving big military weapons or agricultural activities. The advantages of tracking mobility over wheeled robots in such situations are due to improved grip and lower surface strain. Tracked mobility is frequently regarded as especially relevant for search and saving operations [1] where terrain circumstances are generally dangerous and the ecosystem is particularly unstable due to the aforementioned characteristics. This is proved in actuality as well, with tracks serving as the principal mode of mobility for the vast number of relief and recovery robots operated in the area over the last several decades [2–5].

One of the most important criteria for a relief and recovery platform is increasing mobility, among other things. After environmental or man-made tragedies, communication links between the impacted zone and the rest of the globe are frequently clogged, with restricted bandwidth as well as increasing delays. Permanently attached recovery systems have had different levels of efficacy in the previous, providing for a reliable communication connection [6–9]. The use of a cable, however, restricted the robots' mobility and increased the chance of the cord getting trapped in the debris. With a communications connection, remote control of a recovery robotic system in an unorganized dynamic situation is a difficult undertaking due to the crucial nature of rescuing operations. This necessitates increased levels of automation in rescuing robots, like the capacity to reliably handle hazardous terrain situations on their own. The US Army's latest necessity for unmanned casualty extraction systems [10, 11], as well as the inclusion of CasEvac/MedEvac situations in robotic benchmarks, contests like El-Rob, underscore the increasing necessity for these automated robots.

2. RELATED WORK

Restricted vision from unseen locations in the surroundings must be factored into safer motion planning. Several studies have looked into various areas of risk analysis and prudent planning in low-visibility situations. The stochastic risk analysis of obstructed areas has been studied by few researchers. One of those [12] presented a stochastic risk analysis depending upon the amount of traffic along the road, while another [13] suggested an identical method depending upon on weights and speeds of two vehicles. Such methods, meanwhile, have drawbacks from that they need traffic density and robot mass that cannot be recorded by sensors. Another way uses time-to-entry (TTE) as well as Bayesian networks [14] to show the danger level as a probabilistic dispersion. Furthermore, ref. [15] provided a graphical approach suitable for expressing the danger of the road section over the period, and used a dynamical Bayesian approach to estimate occupancy. Even though such research defines the hazard as a possibility so it can be included in the strategic planning, presenting the hazard as a possibility does not guarantee demonstrable security.

Several works adopt a stochastic technique that resolves a worst-case issue to reduce the ambiguity of probabilities. [16] Demonstrated a planner that deals with ambiguity projections at junctions and took emergency braking into account before arriving at the crossing. Furthermore, the study did not take into account sensor distance or cars approaching from behind the vision zone. Refer [17] analyzed the occupation set of automobiles in the surroundings and assessed the presence of an urgent movement, such as a lane shift, to arrange for such a fail-safe action. The vehicle's perceptual field was not considered in this study. Another research [18] proposed a strategy to avoid potentially dangerous circumstances by having the car gently approach the junction while keeping an eye on possibly obstructed vehicles involved using virtual grid maps. A strategy for evaluating the security of a particular trajectory in terms of occlusions was presented in ref. [19]. The motion prediction in reference [20] was centered on an ambiguous environment modeling with deformation. They demonstrated a way of avoiding collisions during a scene's worst-case development. References [21, 22] addressed the possible risk of deformation and restricted sensor capabilities by employing state durations to over-approximate all conceivable states of non - observable barriers. These methods, meanwhile, cannot be used in valet parked situations. Because these methods believe that impediments appear based on the geometry of the blueprint, they are unable to analyze the gaps among parked vehicles with caution.

3. PROPOSED METHODOLOGY

Robots are represented as dynamic units controlled by physics laws in physics-based mobility systems. The development of the condition of the robot is anticipated using dynamical and kinematic designs that relate certain controller inputs (e.g. turning, accelerate), automobile attributes (e.g. weight), and exterior factors (e.g. frictional factor of the track surface) (e.g. location, heading, velocity). The preliminary path defined by the high-level controller is delivered to such a physical robot as a set of checkpoints depending on the outcomes of a physical engine. Depending upon the latest location, the low-level planner on the physical robots directs it to the next path (specific destination). The robot gathers more sensor information as it goes to the next path in the proximity of already discovered units. The high-level controller receives the altered sensor data and, if needed, reanalyzes and produces the altered course. This approach is repeated till the physical robots reach the destination or find that there is any other way. Figures 1 depict a schematic diagram as well as a flow diagram that describes how the suggested technique works.

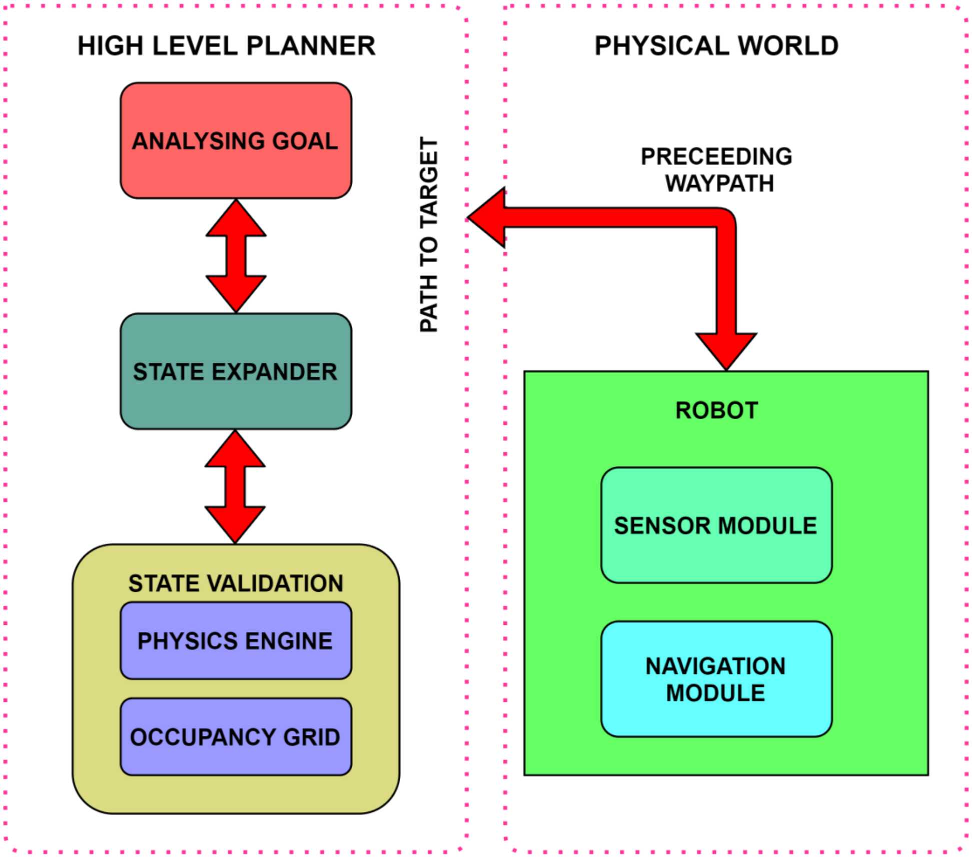

Figure 1: Physical-based motion planning.

3.1. UNCERTAINTY DETECTION OF TRAFFIC PARTICIPANT

We exclusively examine mobile robots as traffic participants in this article. The condition of an automated robot is x= $[x\ y\ \theta]$, as well as the controller input, is u= $[v\ k]$, where x & y are now the positions, θ seems to be the rotation, k seems to be the curve, and v seems to be the velocity.

Assuming a quantization time interval ofΔt, the robotic movements are defined by:

$$x_t = A_t x_{t-1} + B_t u_{t-1} \qquad (1)$$

$$A_t = \begin{bmatrix} 1 & 0 & -v_{t-1}\sin\theta_{t-1}\Delta t \\ 0 & 1 & v_{t-1}\cos\theta_{t-1}\Delta t \\ 0 & 0 & 1 \end{bmatrix} \qquad (2)$$

$$B_t = \begin{bmatrix} \cos\theta_{t-1}\Delta t & 0 \\ \sin\theta_{t-1}\Delta t & 0 \\ 0 & v_{t-1}\Delta t \end{bmatrix} \qquad (3)$$

The existing condition of traffic participants, as well as forthcoming controller input, impacts their future paths. Assume there seem to be N_c automated robots in the scenario, with u^i denoting their controller parameters,$i \in [1, N_c]$. The future path of every robot i can be estimated by providing controller input unit($t \in [0, T]$, when T is the forecast period) to the autonomous vehicles, assuming its original condition x_0^i (1). The robot's sensing system can detect the present conditions of other automated vehicles. Nonetheless, the traffic participants' prospective controller inputs are unclear. In this study, we present a local controller that forecasts traffic participants' upcoming movements. Motion representations are created for every traffic participant depending upon a typical robot's navigation attitude and then assessed utilizing a value function. Ultimately, the best estimate of such a traffic participant's upcoming motion is selected as optimum motion fundamental.

3.1.1. Gaussian Propagation:

The condition is often partly viewable, and the robot dynamic approach can be represented or modeled as a time-varying linear concept. The discontinuous linear model is denoted by the following:

$$x_t = A_t x_{t-1} + B_t u_{t-1} + w_t, \qquad w_t \sim N(0, W_t) \qquad (4)$$

$$z_t = C_t x_{t-1} + v_t, \qquad v_t \sim N(0, V_t) \qquad (5)$$

Where x_t represents the condition vector, u_t represents the controller input vector, z_t represents the measuring vector, w_t represents processing distortion, and v_t represents measuring distortion. The following are the initial Kalman filter calculations:

The following is a procedure update:

$$\hat{x_t^-} = A_t\hat{x}_{t-1} + B_t u_{t-1} \tag{6}$$

$$\sum_t^- = A_t \sum_{t-1} A_t^T + W_t \tag{7}$$

The latest measurement is as follows:

$$L_t = \sum_t^- C_t^T (C_t \sum_t^- C_t^T + V_t)^{-1} \tag{8}$$

$$\hat{x}_t = \hat{x_t^-} + L_t(z_t - C_t\hat{x_t^-}) \tag{9}$$

$$\sum_t = (I - L_t C_t)\sum_t^- \tag{10}$$

here L_t is seems to be Kalman gain, $\hat{x_t^-}$ is the preliminary state estimation, $\hat{x}_t$ seems to be the posterior state estimation, and $\sum_t$ seems to be the posterior estimate fault variance.

Both controller input u_t as well as the measure z_t remain undetermined at the moment of scheduling in our example. Depending upon the value function, u_t is expected to be the best fundamental movement, and z_t is supposed to be a Gaussian dispersion. The state vector along an upcoming path is calculated as $\hat{x}_t$, assuming the starting condition x_0 as well as the controller input ut. x_t's final estimation is as follows:

$$P(x_t) = N(\breve{x}_t, \sum_t) \tag{11}$$

3.2. UNCERTAINTY PROPAGATION FOR THE AUTONOMOUS ROBOT

The robot's dynamic framework is the same as its traffic participants, which is specified in (4) and (5). Unlike many other traffic participants, we have absolute control over the automated robot. The LQR controller can also be developed to follow the predicted path for such a time-varying linear model. A state vector is usually written as$\breve{x}_t$, the controller input vector is marked by $\breve{u}_t$, and the measuring vector is represented by $\breve{z}_t$ along the intended path. By specifying the following, the monitoring issue can be turned into a regulated issue:

$$\tilde{x}_t = x_t - \breve{x}_t \tag{12}$$

$$\tilde{u}_t = u_t - \breve{u}_t \tag{13}$$

$$\tilde{z}_t = z_t - \breve{z}_t \tag{14}$$

Here $\tilde{x}_t$, $\tilde{u}_t$, and $\tilde{z}_t$ denote the variance among the predicted and actual trajectory throughout implementation. The following is an equation for the monitoring device:

$$\tilde{x}_t = A_t\tilde{x}_{t-1} + B_t\tilde{u}_{t-1} + \tilde{w_t} \qquad \tilde{w_t} \sim N(0, \tilde{w_t}) \tag{15}$$

$$\tilde{w_t} = C_t\tilde{x}_{t-1} + \tilde{v}_t, \qquad \tilde{v}_t{\sim}N(0, \boldsymbol{\tilde{v}}_t) \tag{16}$$

A value function is described as follows, assuming a fixed condition and source weight matrix Q and R:

$$J = \sum_{t=0}^{T}\left(\tilde{x}_t^T Q \tilde{x}_t + \tilde{u}_t^T R\, \tilde{u}_t\right) \tag{17}$$

During path creation, the time range T is determined. The value function is minimized by obtaining the optimum control strategy.

$$\tilde{u_t} = -K_{t+1}\tilde{x}_t \qquad (18)$$

Depending upon the dynamical Riccati solution K_t is determined repeatedly backward.

$$K_t = (R + B_t^T P_t B_t)^{-1} B_t^T P_t A_t \qquad (19)$$

$$P_{t-1} = Q + A_t^T P_t A_t - A_t^T P_t B_t K_t \qquad (20)$$

$$P_N = Q \qquad (21)$$

For such a path monitoring device, the Kalman filter processing formulas are just like (6) – (10). The correlation after system updating is $\widetilde{\sum_t}^-$, the posterior correlation is t after measuring change is $\widetilde{\sum_t}$, as well as the Kalman gain, equals $\widetilde{L_t}$. As per [5], the state vector's terminal dispersion is:

$$P(x_t) = N(\check{x}, \widetilde{\sum_t} + \Lambda_t) \qquad (22)$$

Where

$$\Lambda_t = (A_t - B_t K_t)\Lambda_{t-1}(A_t - B_t K_t)^T + \widetilde{L_t} C_t \widetilde{\sum_t}^- \qquad (23)$$

3.2.1. Trajectory Generation

The path generating method in this work is taken from the authors' prior article. With a reference route, which is usually the midline of a sector, the road coordinate is (l, d), while l and d are also the referencing path's transverse and horizontal offsets, correspondingly. A location on the standard path is denoted by the condition (l, d) with$d = 0$. The vertical and transverse directions both are surveyed for endings. We samples N_l layers parallel to the standard route and N_d sites perpendicular to a standard path in every layer. A sampling point's condition (l, d) is described as Λ

$$x(l, d) = x(l, 0) + d\cos\left(\theta(l, 0) + \frac{\pi}{2}\right) \qquad (24)$$

$$y(l, d) = \ y(l, 0) + d\sin\left(\theta(l, 0) + \frac{\pi}{2}\right) \qquad (25)$$

$$\theta(l, d) = \ \theta(l, 0) \qquad (26)$$

$$K(l, d) = (K(l, 0)^{-1} - d)^{-1} \qquad (27)$$

Paths are created by employing cubic curve polynomial to link terminals in adjacent layers. The path's curve is a cubic polynomial with arc length.

$$\kappa(s) = \gamma_0 + \gamma_1 s + \gamma_2 s^2 + \gamma_3 s^3 \qquad (28)$$

[4] has the specifics on how to solve the variables for $\kappa(s)$. The path's length is measured in square feetss_f. The speed characteristic can be constructed for every path section by dividing the velocity v_f and acceleration a_f only at the terminus. We take $a_f = 0$ to decrease the state space. N_v stands for the amount of discretized velocities.

Cubic polynomials of duration are used to determine the velocity characteristic to every path:

$$v(t) = \rho_0 + \rho_1 t + \rho_2 t^2 + \rho_3 t^3 \qquad (29)$$

Speed like a factor of duration is simpler for a control system to understand that velocity as a factor of arc distance, which is distinct from [4].

The time duration T can be calculated for every path using the velocity v_f and acceleration a_f just at the end destination:

$$T = \begin{cases} -\frac{c}{b} & if \quad a = 0 \\ \frac{-b \pm \sqrt{b^2-4ac}}{2a}, & if \quad b^2 - 4ac \geq 0 \\ \emptyset & if \quad b^2 - 4ac < 0 \end{cases} \qquad (30)$$

$$a = (a_0 - a_f)/12 \qquad (31)$$

$$b = (v_0 + v_f)/2 \qquad (32)$$

$$c = -S_f \qquad (33)$$

After that, the variables for (29) are calculated as follows:

$$\rho_0 = v_0 \qquad (34)$$

$$\rho_1 = a_0 \qquad (35)$$

$$\rho_2 = \frac{1}{T^2}\left(3(v_f - v_0) - (a_f + 2a_0)T\right) \qquad (36)$$

$$\rho_3 = \frac{1}{T^3}\left(-2(v_f - v_0) - (a_f + a_0)T\right) \qquad (37)$$

From pathway creation technique, we get to state like a variable of arc length $[x(s)\ y(s)\ \theta(s)\ \kappa(s)]$, so from the velocity creation technique, we get velocity like a function of period $v(t)$. Integral of $v(t)$ can be used to calculate the arc length duration $s(t)$. Replacing $s(t)$ to condition as a function of arc length $[x(s)\ y(s)\ \theta(s)\ \kappa(s)]$ yields condition as a function of duration$[x(t)\ y(t)\ \theta(t)\ \kappa(t)]$.

3.2.2. *Prediction for Traffic Participants*

A regional planner forecasts the movement of traffic participants. We set $N_l = 1$ for traffic a participant, which implies we just prepare for one layer. The explanation for this is that the lengthier we forecast, the more possible the forecast will be inaccurate. Furthermore, it aids in the reduction of computing time. If we estimate the travel of the traffic participant, we must also evaluate the mobility of the robot ahead of time; this creates the chicken-and-egg conundrum. We made the premise in this study because when we estimate the mobility of a traffic participant, and the autonomous robot will continue to travel at their existing speeds without ambiguity about their placements.

As a forecast of such a traffic participant's upcoming movements, we employ the ideal path from the regional controller. Once the future path has been anticipated, a Kalman filter can be used to calculate the condition dispersion along that path.

3.2.3. Planning for Automated Robot

We explore $N_l(N_l \geq 3)$ tiers, N_d vertices on every layer, and N_v velocities on every vertex for the automated robot. A straightforward extensive searching has a computational efficiency of $O((N_d N_v)^{N_l})$. We examine the state space using dynamic computing (Method 1) rather than an exhausting approach, which decreases the processing order to $O(N_l(N_d N_v)^2)$.

The condition is specified as $s = (l, d, v)$ in the selection phase. The aggregated costs of paths linking the beginning state to such a state are represented by the value of a condition $c(s)$. Using the value functions, the value of a path $c(\tau)$ is calculated. The minimum cost at every state is determined through dynamic coding. To use an LQG architecture, the dispersion of states along a path may be estimated for every path during the searching stage. It must also take into account the apprehension of traffic participants. A method that calculates the risk of a crash between an automated robot and a traffic participant while taking into account the unpredictability of both traffic participants and also the mobile robot.

Every robot's location is predictable from the perspective of such a traffic participant. A polygon overlap method is used to identify an accident among two autonomous robots. Both the automated robot's condition and the statuses of other robots are probabilistic, which is formalized using regular deviations. The ellipse created by the correlation matrix can offer an upper limit of the likelihood that now the robot would crash with predictable impediments for a pointed robot having normal dispersion. In other terms, the crash chance is at maximum if the elliptical does not intersect with any predictable barrier. In this situation, though, the robot's form is rectangular. A Minkowski total of a rectangular and elliptical, which is a curved rectangle, is computed. As a prudent estimate of the curved rectangle, we chose the smallest enclosing rectangle. If a new rectangular has no overlapping with barriers, it still retains the characteristic that the crash risk is at the maximum δ. Furthermore, to identify conflicts, we use a polygon overlap technique on the smallest enclosing rectangles.

4. EXPERIMENTAL RESULTS

The suggested planning framework was built in MATLAB, which further communicated with V-REP via remote API interfaces, comprising its D* Lite high-level planners as well as the low-level hybridized automated controllers. Figure 2 shows a picture of a robot's movement as observed by MATLAB planners and the matching location of the robot on V-REP throughout the harsh terrain simulations. The low-level controllers operating in MATLAB received the robot's location and direction (both six aspects) straight from the simulator over the simulation. Particle Physics is utilized to describe the dynamical parts of a scheduling issue in headless form. The suggested planner's findings were contrasted to the results of kinematic planners in both simulated instances. The main distinction between suggested and kinematic planners seems to be when the kinematic edition ignores the dynamic components of an issue by not checking the accuracy of the suggested path by employing a physics engine. It enables the kinematic planners to generate the ideal shortest route in all circumstances, regardless of whether or not the suggested path is feasible. Bounding barriers surround both modeled terrains, preventing the robot from leaving the simulated territory. Except for the boundary walls, neither terrain map had any other permanent impediments. The low-level controllers were adjusted for simulation so that the emulated robot will not perceive terrain characteristics as barriers.

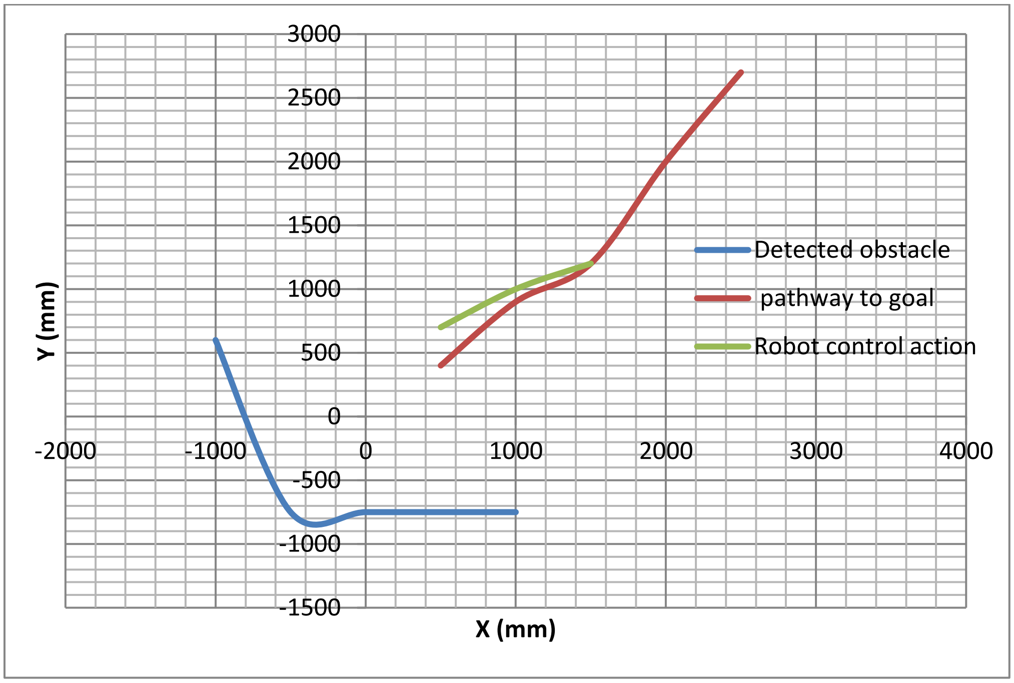

Figure 2: Planner Robotic motion.

Our approach was also contrasted to the accessible set-based approach [16]. The autonomous robot (blue) is attempting to overtake a stationary vehicle (red) on such a two-lane highway while incoming traffic approaches, as seen in Figure 3. In Figure 3a, probabilistic planning (i.e., the planners suggested in this work but not addressing ambiguity) first plans the optimum itinerary for the mobile robot, and then the accessible set is calculated along its path. The path is flagged as dangerous since the automated robot's, as well as traffic participant's accessible sets. Since this accessible set-based technique gives no precise data to a planner, we must reorganize a path using the probabilistic planner and analyze it repeatedly to discover a viable route. Furthermore, once a secure, although not necessarily ideal, route is determined, it will quit exploring.

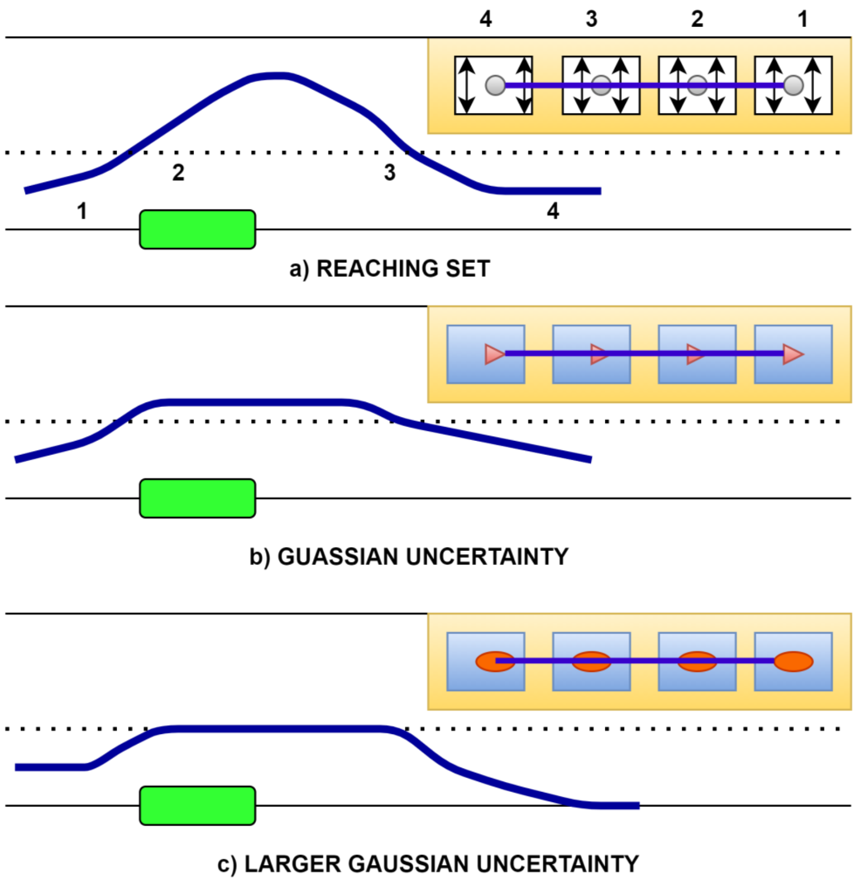

Figure 3: The proposed framework model under various conditions.

The planning architecture described in this work, on either hand, provides a straight secure track, as illustrated in Figure 3b, in which grey ellipses reflect the degree of condition ambiguity. The accessible set-based approach, on the other hand, can offer a promise (i.e., 100% assurance) that a crash will not happen, whereas our Gaussian-based technique just provides a probability (e.g., 75%) limit. In Figure 3c, we suppose that the mobile robot obtains no information about the x dimension of a traffic participant's location. So because the traffic participant's x dimension is more unclear, the automated robot returns to a correct lane quicker than that in Figure 3b.

5. CONCLUSION

We've proposed a paradigm for automated robot motion planning with ambiguity. Positioning and operation of an automated robot, as well as detection and forecasting of traffic participants, all contribute to the ambiguity examined in this work. The controller has demonstrated the capacity to cope with unpredictability posed by both the automated car and other traffic participants. While contrast to our earlier probabilistic controller and designers that ignore the unpredictability created by other traffic participants, this can generate better paths. One proposed enhancement is to make the motion of traffic participants highly unpredictable. Furthermore, depending upon intentions, which are shown in the track record and current circumstances, a superior forecast may be made. Another enhancement that might be made is to predict a more accurate probability for GPS inaccuracy than the Gaussian probability, which would aid in obtaining a more accurate evaluation of the positioning difficulty.

REFERENCE

1. 1. Odedra, S., Prior, S., Karamanoglu, M.: Investigating the mobility of unmanned ground vehicles. In: Proceedings of the international conference on manufacturing and engineering systems. Huwei, Yunlin (2009).
2. Murphy, R.R.: A decade of rescue robots. In: 2012 IEEE/RSJ international conference on intelligent robots and systems, pp. 5448–5449 (2012).
3. Wang, C., Chao, K., Sivaperumal, S., Suresh, P., “Anti-PVT-Variation Low-Power Time to Digital Converter Design Using 90 nm CMOS Process” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 28, Issue 9, 2020, pp. 2069 – 2073.
4. Nagatani, K. et al.: Emergency response to the nuclear accident at the Fukushima Daiichi Nuclear Power Plants using mobile rescue robots. J. F. Robot. 30(1), 44–63 (2013).
5. Murphy R.R.: Disaster robotics, vol. 1. The MIT Press, Cambridge (2014).
6. Davids, A.: Urban search and rescue robots: from tragedy to technology. IEEE Intell. Syst. 17(2), 81–83 (2002).
7. Snyder, R.G.: Robots assist in search and rescue efforts at WTC. IEEE Robot. Autom. Mag. 8(4), 26–28 (2001).
8. Suresh, P., Mariyal, C., Sivasubramonia Pillai, T.V., Rajesh, K.B. and Jaroszewicz, Z., “Study on polarization effect of azimuthally polarized LG beam in high NA Lens system,” Elsevier – Optik, vol. 124. Issue 21, 2013. pp. 5099-5102
9. Murphy, R.R., Kravitz, J., Stover, S.L., Shoureshi, R.: Mobile robots in mine rescue and recovery. IEEE Robot. Autom. Mag. 16(2), 91–103 (2009).
10. Karthikeyan, V., Suresh, P., Saravanakumar, U., Anil Venkata Rama Sekhar, AKM. and Srikanth, S., “Wide Band Millimeter Wave Fabric Antenna for 60GHz Applications,” International Journal of Innovative Technology and Exploring Engineering, Vol. 8, Issue 9, 2019, pp. 1211-1214.
11. Beebe, M.K., Gilbert, G.R.: Robotics and unmanned systems – ‘Game changers’ for combat medical missions. In: Proc. NATO RTO-HFM 182 Symp. Adv. Technol. New Proced. Med. F. Oper. (2010).
12. Suresh, P., T Venkata Prasad, B Uday kiran Yadav, SV Narayana Reddy, S Yaswanth Reddy, R.Rajkumar, U.Saravanakumar, 2020 “Dynamic Partial Reconfiguration Performance Analysis In Field Programmable Gate Array For Streaming Applications,” International Journal of Scientific & Technology Research, Vol. 9, Issue 02, 2020, pp. 3098-3102.
13. Damerow, F.; Puphal, T.; Li, Y.; Eggert, J. Risk-based driver assistance for approaching intersections of limited visibility. In Proceedings of the 2017 IEEE International Conference on Vehicular Electronics and Safety (ICVES), Vienna, Austria, 27– 28 June 2017; pp. 178–184.
14. Noh, S. Decision-Making Framework for Autonomous Driving at Road Intersections: Safeguarding Against Collision, Overly Conservative Behavior, and Violation Vehicles. IEEE Trans. Ind. Electron. 2019, 66, 3275–3286.

15. McGill, S.G.; Rosman, G.; Ort, T.; Pierson, A.; Gilitschenski, I.; Araki, B.; Fletcher, L.; Karaman, S.; Rus, D.; Leonard, J.J. Probabilistic Risk Metrics for Navigating Occluded Intersections. IEEE Robot. Autom. Lett. 2019, 4.
16. De Campos, G.R.; Runarsson, A.H.; Granum, F.; Falcone, P.; Alenljung, K. Collision avoidance at intersections: A probabilistic threat-assessment and decision-making system for safety interventions. In Proceedings of the 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), Qingdao, China, 8–11 October 2014; pp. 649–654.
17. Magdici, S.; Althoff, M. Fail-safe motion planning of autonomous vehicles. In Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 1–4 November 2016; pp. 452–458.
18. Hoermann, S.; Stumper, D.; Dietmayer, K. Probabilistic long-Term prediction for autonomous vehicles. In Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, USA, 11–14 June 2017; pp. 237–243.
19. Naumann, M.; Konigshof, H.; Lauer, M.; Stiller, C. Safe but not overcautious motion planning under occlusions and limited sensor range. In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019; pp. 140–145.
20. Tas, O.S.; Hauser, F.; Stiller, C. Decision- Time Postponing Motion Planning for Combinatorial Uncertain Maneuvering. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; pp. 2419–2425.
21. Orzechowski, P.F.; Meyer, A.; Lauer, M. Tackling Occlusions Limited Sensor Range with Set-based Safety Verification. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; pp. 1729–1736.
22. Orzechowski, P.F.; Li, K.; Lauer, M. Towards responsibility-sensitive safety of automated vehicles with reachable set analysis. In Proceedings of the 2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE), Graz, Austria, 4–8 November 2019.