Robotica (2022), **40**, pp. 2505–2526
doi:10.1017/S0263574721001764

CAMBRIDGE
UNIVERSITY PRESS

RESEARCH ARTICLE

Intent-Aware Optimal Collision Avoidance and Trajectory Planning for a Pursuit Vehicle

Karnika Biswas[1], Indrani Kar[1,*] and Eric Feron[2]

[1]Department of EEE, IIT Guwahati, Assam, India and [2]CEMSE, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia
*Corresponding author. E-mail: indranik@iitg.ac.in

Received: 24 April 2020; **Revised:** 29 October 2021; **Accepted:** 3 November 2021; **First published online:** 9 February 2022

Keywords: optimal control, moving target, dynamic collision avoidance, inequality constraints, velocity barrier, shrinking horizon, penalty function

Summary

This paper presents an integrated optimal control framework for velocity and steering control of an autonomous pursuit vehicle, where the control objectives satisfy the requirements of collision avoidance and moving target tracking. A distinctive feature of the proposed velocity and steering control is the application of logarithmic penalty functions to both. The control barrier imposed by logarithmic function provides a unique tool in computing a balanced trajectory with optimal tracking error, control effort and safety margin. Trajectories compliant with the safety regulations for autonomous driving have been planned based on estimated intention of the target and the obstacles. Effects of the controller weights have been extensively simulated to assess the performance of the proposed strategy in a variety of dynamic situations. The controller has been validated on a real-life robot by using a shrinking horizon control policy for iterative optimisation.

1. Introduction

Research and development regarding autonomy of automobiles started a few decades back, but it was only recently that it gained momentum. In 2018, after a self-driving car crash caused a casualty, comprehensive guidelines on safety have been formulated by competent authorities like Society for Automotive Engineers and National Highway Traffic Safety Administration of USA[1–3]. The European Auto Drive Project is also a major proponent for this cause that involves industrial stakeholders like Ford, Jaguar, Tata, Hyundai and Volvo. Even, the Ames research centre of NASA has collaborated with Nissan over similar activities. It is evident that the baseline regulations remain similar for space exploring and commercial vehicles deployed in tracking, inspection, surveillance and service industries. This paper explores the scope of safe navigation planners for mobile pursuit vehicles (tracking robots), such that the plan is compliant with the foregoing safety guidelines. An upward trend of robots sharing workspace with humans is a major motivation for incorporating intent awareness in planning policy.

The goal of this paper is not intent detection, but to use intent inference as an aid to intelligent trajectory planning. As pointed out in ref. [4], intent inference may include the predicted goal of a moving vehicle or a human sharing the workspace, within a prescribed time interval. The prediction dynamically takes into account the change in expected behaviour if the goal or nature of motion changes with time. Existing literature describes a probabilistic approach [5] to classifying intended motion of ships into compliant and non-compliant categories for designing anti-collision paths. Improvements to Bayesian prediction is claimed in ref. [4], which introduces a robust control formulation and ref. [6], which considers mutual-intent decoding and planning between a number of vehicles. A hybrid model composed of deterministic polynomials and Gaussian distribution of obstacle positions has been proposed in ref. [7].

Gait-based probabilistic estimation of future activities of pedestrians have been studied [8] in the context of intent awareness of self-driving cars.

The most popular approach to dynamic pursuit is reactive navigation. It works on the principle of computing a local control action that relies on instantaneous sensing and actuation. Reactive navigation can be based on a number of methods like optical flow gradient [9], distributed sensing [10], and learning algorithms implemented by neural networks [11], fuzzy inferences [12] or a combination thereof [13]. However, reactive planning is unable to look beyond a local planning horizon.

In contrast, knowledge of intention can improve the performance, safety and stability features of pursuit vehicles with multi-objective optimisation. An integrated speed-steering controller [14] has been reported to leverage this knowledge in computing smooth cubic spline trajectories contoured by lane curvature. Another integrated controller [15] combines proportional speed control with differential braking to address emergency collision avoidance. In contrast to steering, yaw moment is generated due to differential braking in the two wheels and the controller suggests selecting a yaw moment that satisfies the stability criterion. Computation of invariant reachable sets that can accommodate a family of allowable lateral displacements of the vehicle under variable velocities [16] is yet another approach to trajectory generation and control in this context. In the reference, [17] the authors have suggested an efficient utilisation of intent inference by incorporating the extent (size) and orientation of the vehicle in addition to state estimation. An adaptive integrated controller has been studied in ref. [18], where a reinforced learning policy has been adopted to assign weightage to the velocity and orientation control mechanisms.

This paper proposes an optimal trajectory planning and control design for a pursuit vehicle using an estimated knowledge of intent of other mobile entities (obstacles and target). The aim is to find answers to decisive factors like – should the pursuer speed up and overtake a slow obstacle moving along the same direction, should the pursuer slow down and allow a fast obstacle to pass by, should change in steering control be combined with velocity adaptation for avoiding a head-on collision, and so on. The proposed plan has been designed to comply with the safe navigation mandates mentioned before, while the basic task is to track and intercept a moving target at a finite time. In contrast to separate lateral and longitudinal control of a vehicle, this paper proposes an unified optimal framework that allows variations in priorities and bounds to steering and velocity control. This integrated approach is better suited to present a comprehensive overview of performance over local planning and geometric control methods.

Adaptive [19] and model predictive control techniques [20] are widely used in real-time implementation of optimal algorithms. A scalable, non-linear model predictive controller has been demonstrated in ref. [21] which has been found to improve the ergodicity of a hybrid system in real-time operation. Receding horizon control strategy has also been explored in underwater surveillance [22] and in a modified form in dynamical systems with unknown, bounded and stochastic disturbances [23,24].

The paper uses a shrinking horizon control strategy for real-time implementation of the proposed plan. Planning over a shrinking horizon is more resilient [24] in finding a solution than a receding horizon approach. We thereby summarise the *key contributions* of this paper as follows:

1. **Intent-aware navigation planner:** Intent awareness of the proposed optimal strategy is not only a local/reactive response to identify possible collisions but also helps in global optimal trajectory planning based on motion estimation of the mobile agents at the end of the current planning horizon. With special emphasis to target tracking, proposed logarithmic penalty function yields an efficient passive anti-collision behaviour over entire planning horizon subject to speed and lane restrictions usually imposed on auto-driven pursuit vehicles.
2. **Integrated control without switching noise:** The proposed method functions as both trajectory planner (long goal) and controller (short goal) that integrates optimal trajectory generation for dynamic target pursuit and local safety pertaining to collision avoidance. This helps in achieving configurable and assessable performance and safety scores by a *single* control action, involves no switching noise and provides clearer qualitative and quantitative prediction of runtime cost effectiveness than decoupled controllers. Non-quadratic cost optimisation yields enhanced passenger comfort.

https://doi.org/10.1017/S0263574721001764 Published online by Cambridge University Press

3. **Shrinking horizon control:** The proposed autonomous path planning and control approach have been experimentally validated on a real system, which harvests the benefits of a shrinking horizon policy in terms of resilience and the ability to render a broad optimal perspective.

The paper has been organised as follows. After introducing the mobile components in Section 2, the main contribution of the paper has been discussed in Section 3. This section describes the integrated controller design, conversion of the optimal control into a suitable boundary value problem and solution thereof. Section 3 also describes the advantages of a novel usage of logarithmic penalty function in constraining velocity apart from path. The experimental set-up and the shrinking-horizon-based control architecture has been described in Section 4. Section 5 provides a detailed analysis of simulated and experimental data obtained from a real-life robot in action. This section includes an exploration of the design space and the role played by logarithmic barrier in velocity and steering control, in the light of intent-aware control policy. Simulated and experimental comparison with strategically different benchmark techniques and an integrated control formulation of different type have been presented to elucidate the merits of the proposed design, before concluding and presenting the future work directions.

2. Preliminaries

This section briefly explains the elements of the optimal control problem, which shall be formulated in Section 3. Any model of the elements discussed here are only for a clear understanding of the controller design, and explicit dependence of the optimal solution(s) or the efficacy of the optimal controller on any model is disclaimed.

Pursuer : We assume that the pursuer is a differentially driven, two-wheeled mobile vehicle of symmetric geometry (with radius $r_{pursuer}$), governed by the non-linear kinematics described in (1). A set of position vectors $[x(t), y(t)]^\top$ represent the states, while linear velocity $v(t)$ and heading $\theta(t)$ are the control variables. Equation (1) is a modified unicycle model, with reduced dimension that (i) improves comprehensibility of the mathematics governing the controller design and (ii) reduces computational footprint as the optimisation steps are iteratively executed during runtime:

$$\left.\begin{aligned} \dot{x}(t) &= v(t)\cos(\theta(t)) \\ \dot{y}(t) &= v(t)\sin(\theta(t)) \end{aligned}\right\} \tag{1}$$

Target : We assume, there is a single, non-evasive mobile unit having a circular hull with radius r_{tar} assigned as the 'target'. The target moves independently according to some predefined dynamics, which is not known to the pursuer. However, it is assumed that the pose of the target is measurable at certain intervals of time and approximate target positions at certain future instants (intent) can be constructed by extrapolating the navigation history [25]. Tracking the target does not involve following the same path. Instead, the objective is to command the pursuer such that the Euclidean distance between the pursuer and the target is reduced over a planning horizon and finally the pursuer intercepts the target at the end of the planning horizon, which is at a predetermined time T_{int}.

Obstacle : An obstacle can be a static or mobile object other than the target and the pursuer and represented by a circle of a radius of r_{obs}, centered at $(x_{obs}(t), y_{obs}(t))$ at any time t within the planning horizon. The dynamics behind the obstacle's motion are also unknown and no look-up table has been used either. We only assume that the obstacle's position can be measured by suitable sensors (e.g., camera) at regular intervals. An actual obstacle may have an arbitrary shape but for simplicity of representation, we conceive the representative obstacle to be a convex planar object completely containing the actual obstacle. Further details about the construction of representative obstacle have been furnished in Appendix A.

https://doi.org/10.1017/S0263574721001764 Published online by Cambridge University Press

3. Integrated Controller Design

In this paper, the optimal navigation problem has been formulated using an integrated control approach. This approach generates an optimal trajectory that maintains a balance between dynamic tracking and collision avoidance. A weighted cost function is designed to prioritise collision avoidance over tracking when an obstacle(s) is present within a region of interest. A 'region of interest' is a predefined set of states such that if an obstacle is located within that set, it is considered to be a collision threat and needs to be avoided.

Existing literatures confirm that multiple obstacles within a region of interest at a particular time can be avoided by a gradient-driven repulsive behaviour [26]. Our proposed design performs this function optimally and without having to increase the dimension of the optimal control problem because of multiplicity of obstacles. However, finding a viable solution according to the safety guidelines for autonomous driving can be difficult if the clutter is significantly high. In such a case, the pursuer can follow the intent of the obstacle and apply a brake or slow down and retry until a solution is found again. Keeping in mind that the pursuer is required to intercept the target at a predetermined time, the restrictions imposed by controlling the pursuer's velocity can be conflicting with the tracking goal.

The proposed controller design aims to solve this intent-driven complex self-regulated navigation problem. However, unlike a quadratic optimisation suggested in ref. [27], a penalty-function-based planning and constraint management has been applied in an optimal control framework. The problem statement can be summarised as follows:

Problem statement : Suppose, at time t, $(x_{tar}(t), y_{tar}(t))$, $(x^i_{obs}(t), y^i_{obs}(t))$ and $(x(t), y(t))$ are the instantaneous positions of the target, an i^{th} obstacle and the pursuer, respectively. Let there are m obstacles within the region of interest at time, $t \geq 0$. Assume, the navigation starts from $t = 0$ and T_{int} is the desired interception time, such that, $T_{int} \in \mathbb{R}^+$ and $0 < T_{int} < \infty$. Consider, $v(t)$ is the instantaneous velocity of the pursuer. The objectives necessary for planning a safe navigation profile can be recited as:

1. **Interception** of the pursuer with the target at final time T_{int} can be established if the target states and the pursuer states become the same at T_{int} (theoretically). This condition is demonstrated by Eq. (2):

$$\Psi(x(T_{int}), y(T_{int})) = [x(T_{int}) - x_{tar}(T_{int})\ \ y(T_{int}) - y_{tar}(T_{int})]^\top = [0\ \ 0]^\top \tag{2}$$

The function $\Psi(x(T_{int}), y(T_{int}))$ is called the final state function and Eq. (2) is referred to as the *final state constraint*.

2. **Collision avoidance** with the i^{th} obstacle can be achieved, provided that the pursuer is able to maintain a predefined minimum distance of separation or safety margin, d_i from the obstacle according to Eq. (3):

$$(x(t) - x^i_{obs}(t))^2 + (y(t) - y^i_{obs}(t))^2 > d_i^2 \tag{3}$$

Here, d_i is the minimum safety margin (MSM) between the centres of the i^{th} obstacle and the pursuer, such that $\|d_i^2\| > \|r^2_{obs^i} + r^2_{pursuer}\|$, where r_{obs^i} is the radius of the i^{th} obstacle and $r_{pursuer}$ is the radius of the pursuer. Equation (3) will be referred to as the *path constraint*.

3. **Velocity barrier** is an upper limit imposed on the pursuer's velocity to restrict the pursuer's motion. It can be the specified saturation limit recommended for the vehicle or a reconfigurable upper bound (numerically less than or equal to the saturation limit), designed to manipulate the navigation profile in each planning iteration. Suppose, v_{max} is the chosen velocity barrier defined over the interval, $[t, T_{int}]$, such that the *velocity saturation constraint* is satisfied according to Eq. (4):

$$v(t) < v_{max} \tag{4}$$

https://doi.org/10.1017/S0263574721001764 Published online by Cambridge University Press

The objective function for the navigation problem can be defined as a weighted sum of these constraints as shown in Eq. (5).

$$J = \int_0^{T_{int}} \Big(- w_v \ln (v_{max} - v(t)) - w_b \sum_{i=1}^{m} \ln \big((x(t) - x^i_{obs}(t))^2 + (y(t) - y^i_{obs}(t))^2 - d_i^2 \big) + \\ w_d \big((x(t) - x_{tar}(t))^2 + (y(t) - y_{tar}(t))^2 \big) \Big) \mathrm{d}t \qquad (5)$$

The elements w_d, w_b and w_v are positive real constants, known as *weighing* constants. These are design variables which assign relative priorities to the individual components contributing to the compound cost. Among these, w_d determines priority of tracking, the steering weight w_b determines how the pursuer will steer away from the obstacles and the velocity weight w_v determines how strictly the velocity barrier is implemented. The role of these weighing constants will be explained further in Section 5.

The optimal control problem P_1 is fundamentally a minimisation problem for the objective function, J as demonstrated in Eq. (6):

$$P_1 : \min_{(v(t),\theta(t)) \in \mathcal{U}} J \qquad (6)$$

subject to kinematic constraints $\begin{cases} \dot{x}(t) = v(t) \cos (\theta(t)) \\ \dot{y}(t) = v(t) \sin (\theta(t)) \end{cases}$

and subject to constraints recited in (2), (3) and (4)

where $\mathcal{U} \in (\mathbb{R}^+ \times \mathbb{R})$ is the set of admissible control variables ($v(t) \in \mathbb{R}^+$ and $\theta \in \mathbb{R}$), and x and y are the state trajectories. Admissible forward velocity is assumed to be non-zero and positive during the tracking task, assuming no backward motion. Admissible heading can take values between $-\pi$ to π radians. In order to solve the optimisation problem P_1, optimality conditions for the control variables $v(t)$ and $\theta(t)$ are derived by applying Pontryagin's maximum principle (Section 3.1). The solution to P_1 is the set of optimal control variables $v^*(t)$ and $\theta^*(t)$, and the state trajectories $(x^*(t), y^*(t))$ being functions thereof are also optimal.

3.1 Optimality conditions

A real-valued Hamiltonian function $\mathcal{H}$ is constructed as in (7). At any $t \in [0, T_{int}]$, if the constraints (2), (3) and (4) are satisfied, then the Hamiltonian $\mathcal{H}$ reaches its maxima and incurs a minimal cost:

$$\mathcal{H} = w_v \ln (v_{max} - v(t)) - w_d((x(t) - x_{tar}(t))^2 + (y(t) - y_{tar}(t))^2) + w_b \sum_{i=1}^{m} \ln ((x(t)- \\ x^i_{obs}(t))^2 + (y(t) - y^i_{obs}(t))^2 - d_i^2) + \lambda_1(t)v(t) \cos (\theta(t)) + \lambda_2(t)v(t) \sin (\theta(t)) \qquad (7)$$

Two new variables, $\lambda_1(t)$ and $\lambda_2(t)$, have been introduced in the expression for $\mathcal{H}$ corresponding to each state equation. These variables are the costates or the Lagrange's multipliers. Unlike slack variable representation of inequality constraints, no additional variable is necessary to describe the Hamiltonian. Instead of the traditional quadratic form of control energy, [28] this problem utilises a logarithmic barrier to the velocity. For a given v_{max}, the negative sign of the term $-\ln (v_{max} - v(t))$ results in cost minimisation upon satisfaction of (4). This further implies that the controller facilitates minimum control effort (CE), computed as $\int_t^{T_{int}} v^2(t)dt$, $(0 \leq t \leq T_{int})$.

https://doi.org/10.1017/S0263574721001764 Published online by Cambridge University Press

'Stationarity' of the Hamiltonian with respect to variations in the control vector leads to Eq. (8), which expresses the control variables in terms of the costates:

$$\begin{aligned}\frac{\partial \mathcal{H}}{\partial v} &= 0 \Rightarrow v(t) = v_{max} - \frac{w_v}{\sqrt{\lambda_1^2(t) + \lambda_2^2(t)}} \\ \frac{\partial \mathcal{H}}{\partial \theta} &= 0 \Rightarrow \theta(t) = \tan^{-1}\left(\frac{\lambda_2(t)}{\lambda_1(t)}\right) \end{aligned} \tag{8}$$

Recall that w_v is positive and for a chosen w_v, if $w_v << \sqrt{\lambda_1^2(t) + \lambda_2^2(t)}$ (positive square root, since velocity must be non-negative) holds at any t, it indicates the velocity of the pursuer is marginally close to the imposed barrier and draws a penalty on the cost. Similarly, if $(x(t) - x^i_{obs}(t))^2 + (y(t) - y^i_{obs}(t))^2$ is less than $1 + d_i^2$ but greater than 0 at any t, then it indicates that the steering is insufficient to maintain the MSM and a penalty has to be paid. In both of these cases the constraints are not violated but show a possible violation in near future. To negate this possibility the cost is increased and the minimisation problem P_1 is triggered back into action. The role of the barrier function will be further elucidated in Section 3.3.

Assuming that $v(t)$ is continuous and upper-bounded, it is reasonable to further assume that the costates $\lambda_i, i \in \{1, 2\}$ are also continuous and bounded functions of time. Thereby, we replace $v(t)$ and $\theta(t)$ in $\mathcal{H}$ by equivalent functions of $\lambda_i, i \in \{1, 2\}$. Let us define the augmented state vector as $[x(t), y(t), \lambda_1(t), \lambda_2(t)]^\top$. Now, differentiating the Hamiltonian with respect to the augmented state vector yields a set of modified dynamical relations as shown in (9). This step converts the optimal control problem (6) into a system of non-linear ordinary differential equations (ODE) (9), which can be solved with the knowledge of the boundary values:

$$\begin{aligned}\dot{x}(t) &= \frac{v_{max}\lambda_1(t)}{\sqrt{\lambda_1^2(t) + \lambda_2^2(t)}} - \frac{w_v\lambda_1(t)}{\lambda_1^2(t) + \lambda_2^2(t)} \\ \dot{y}(t) &= \frac{v_{max}\lambda_2(t)}{\sqrt{\lambda_1^2(t) + \lambda_2^2(t)}} - \frac{w_v\lambda_2(t)}{\lambda_1^2(t) + \lambda_2^2(t)} \\ \dot{\lambda}_1(t) &= 2w_d(x(t) - x_{tar}(t)) - \sum_{i=1}^{m} \frac{2w_b(x(t) - x^i_{obs}(t))}{(x(t) - x^i_{obs}(t))^2 + (y(t) - y^i_{obs}(t))^2 - d_i^2} \\ \dot{\lambda}_2(t) &= 2w_d(y(t) - y_{tar}(t)) - \sum_{i=1}^{m} \frac{2w_b(y(t) - y^i_{obs}(t))}{(x(t) - x^i_{obs}(t))^2 + (y(t) - y^i_{obs}(t))^2 - d_i^2} \end{aligned} \tag{9}$$

3.2 Boundary value problem and its solution(s)

This type of simultaneous non-linear equations as in (9) usually do not have a closed-form solution. In this case, Eq. (9) has been solved numerically by using the *bvp4c* ODE solver of Matlab. A comprehensive survey [29] on different numerical methods warns that the selection of a numerical solver is crucial to generate a solution(s). Popular recipes [30] like direct shooting, multiple shooting and reduced superposition are inappropriate due to the complex relation of the augmented state variables. The high degree of non-linearity of (9) severely affects convergence if the initial guess is poor. The method of relaxation, on the other hand, is slow to converge and requires heavy computational resources. In comparison, collocation [31] offers better numerical stability and is used by the ODE solver adopted in this paper. The computation proceeds in the following sequence:

1. It may be recalled that the target dynamics is not known in advance and target states have only been measured until current time, t. Hence, at any t, we implement an estimator that predicts 'approximate' target states that would be achieved at T_{int}, following the history of target's motion available till time, t.

https://doi.org/10.1017/S0263574721001764 Published online by Cambridge University Press

2. The initial states, $x(0)$ and $y(0)$, of the pursuer are known and the final states, $x(T_{int})$ and $y(T_{int})$, that would be achieved at interception are computed from the final state function (2), where the final states of the target are estimated according to the previous step.
3. Using the initial and approximate final states of the pursuer, (9) is solved. The costates, $\lambda_i(t)$, $i \in \{1, 2\}$ are unconstrained and are free to take values according to their relations with the states $x(t)$ and $y(t)$.

In the safety guidelines of ref. [3], kinematic modeling has been suggested to predict situations like time required by a pedestrian to reach/clear a collision zone, time-to-collision, distance-to-collision and the likes. In the absence of a known model and given the dynamicity of the problem, our aim is to design the system with the least possible computational overhead. It is not unknown that non-linear state estimators like unscented Kalman filter involve intensive recursive computations. So, instead of a Kalman filter [32] we have applied a polynomial estimation method to predict boundary values. References to evolution of trajectory from past knowledge also appears in ref. [14]. Polynomial estimation using the Neville–Aitken [33] routine is simple to implement. Inaccuracies of the initial iterate can be overcome as new measurements are available over time. The polynomial representing the target's state trajectory is updated in every iteration. Eventually, as t moves towards the horizon T_{int}, the approximate states of the target closely follows the actual dynamics (unknown to the pursuer). Detailed steps of the estimation process has been discussed in Appendix B.

It is possible to encounter certain situations like, too many obstacles, inappropriate velocity barrier, etc. for which the set of non-linear ODEs may not have any solution at t. In this aspect, a shrinking horizon model predictive control plays an advantageous role. Such unanticipated situations can be overcome by following the optimal path generated in the previous planning iteration. We assume that the sampling frequency is high enough such that the trajectories do not suffer drastic changes until the next planning iteration.

However, in general, Eq. (9) can have as many solutions as there are distinct sets of admissible initial values for the augmented state vector. Recall that the initial boundary conditions for the states $x(0)$ and $y(0)$ are known, but the costates are free variables. So, if two or more mutually exclusive sequences of $(x(t), y(t))$ satisfy (9) in the interval $(0, T_{int})$, the corresponding sequence of costates $(\lambda_1(t), \lambda_2(t))$ are mutually exclusive within the interval, $[0, T_{int}]$. This means that the initial states for the augmented state vector are different for each solution. Depending on the initial guess, a solver will generate one of the different solutions under same experimental conditions. In a special case, if only one sequence of $(x(t), y(t))$ satisfies the boundary values in the closed interval $[0, T_{int}]$, then it can be guaranteed that a unique set of $(\lambda_1(t), \lambda_2(t))$ will exist corresponding to the sequence $(x(t), y(t))$. In that particular case, (9) will have a unique solution in the said closed interval. A detailed discussion on the existence of solution(s) has been provided in Appendix C, which can be verified against [34,35].

3.3 Logarithmic penalty function

The process of converting hard constraints (equality versions of Eqs. (3) and (4)) into soft constraints by replacing them with equivalent functions is not new. The idea is to formulate a smooth gradient of multiple stages leading to a possible violation of the hard constraint. Our choice for the appropriate mapping function is a natural logarithm of the constrained variable. In this paper, we have shown how the logarithmic barrier can be customised to meet the requirements of safe navigation depending on the intent inference of the obstacle(s) and the target.

Logarithmic barrier is a penalty-inducing function. As the system trajectory tends to move closer to the constraint boundary, the penalty increases exponentially, eventually increasing the cost. For a minimisation problem like P_1, the increase in cost due to penalty creates a conflict of interest. In response, the optimal control problem generates a sequence of control vectors that tries to bring down the penalty until the system reaches a state of minimum cost again [36]. This can be explained with an example.

https://doi.org/10.1017/S0263574721001764 Published online by Cambridge University Press

Table I. *Different values of μ and their meanings*

Values	$\mu \leq 0$	$0 < \mu < 1$	$\mu = 1$	$\mu > 1$
Inferences	Invalid	Penalty	No Effect	Cost Reduction

Let us consider, a real-valued natural logarithm of an argument, μ, where μ is the equality version of the path constraint, (3). The equality indicates the MSM or the absolute, strict constraint that must not be violated. When μ is 'safely' positive, that is the squared Euclidean distance between the i^{th} obstacle and the pursuer is greater than $1 + d_i^2$, then there is no penalty. In fact, the steering action aids in the process of minimisation of the compound cost (5). When μ is 'acceptably' positive, that is the squared Euclidean distance between the obstacle and the pursuer is equal to $1 + d_i^2$, then also there is no penalty, but the resulting trajectory does not contribute anything to the minimisation of J. However, if both of the foregoing stages are crossed and for some reason, μ becomes 'barely' positive, that is $0 < \mu < 1$, this indicates a situation of imminent collision. Over the range of the value function (0, 1), the cost incurred increases exponentially and forces the minimisation problem to generate a sequence of control variables that contradicts the violation of MSM. The relation between the value function of μ and the actions generated by the logarithmic barrier is summarised in Table I. The role played by logarithmic barrier on steering control of the vehicle includes maintaining a balance between excessive and insufficient lateral adjustment [2]. Automatic lane changes, imposing access restrictions on certain paths (e.g., one way, no turn etc.) are some of the practical driving situations that can benefit from this formulation. Verscheure et al. [37] have also documented the rapid path-tracking capabilities of logarithmic barrier formulations in time optimal applications. In addition, we demonstrate a novel use of logarithmic barrier in the form of velocity constraint. Penalty is imposed on the cost if the pursuer's velocity $v(t)$ tends to reach the upper bound v_{max}. While this could also be achieved by a quadratic expression, the latter may fail to detect a violation, given that in real systems the cost is computed at discrete time instants. It is straightforward to see that a quadratic form yields the same cost irrespective of the sign of the arguments and there is a single-point violation indicator, for example, $v(t) = v_{max}$. On the other hand, logarithmic barrier provides a multiple-point violation indicator $0 < v_{max} - v(t) < 1$ and has better chances of avoiding constraint violations.

The upper bound v_{max} need not to be the saturation velocity of the vehicle. Different upper bounds can be used to constrain the velocity in different planning iterations, depending on the intent inference. For an instance, the pursuer may be allowed to slow down while dealing with fast-moving obstacles or while handling an imminent head-on collision or during a transition into a lower-speed lane [3]. Likewise, the pursuer can be allowed to speed up while overtaking slow obstacles or during a high-speed/freeway merge. This type of behaviour has been termed as 'courteous' multi-agent interaction [6] and has often been found to be an important decisive factor at crossroads in the context of autonomous driving. Velocity control can be treated as an integral part of optimal trajectory planning, by allowing v_{max} to vary, when a new planning iteration begins. The proposed strategy is resilient to deadlocks and increases the chances of finding safe navigation routes in dynamic and cluttered environments.

The features depicting the merits of the logarithmic barrier function have been summarised below:

i. It is defined over a real field which means when the value function of arguments tend towards zero or less, it indicates a malfunction.

ii. It enables cost reduction if the argument is highly positive.

iii. It imposes a penalty if the value function of the argument ranges between (0,1), with penalty increasing exponentially if the value tends to zero.

iv. It helps maintaining dimension of the Hamiltonian function to a minimum.

v. A negative logarithmic barrier is a non-increasing, monotonic and convex function.

vi. It can be used for creating a balance between passive and aggressive steering for collision avoidance.

https://doi.org/10.1017/S0263574721001764 Published online by Cambridge University Press

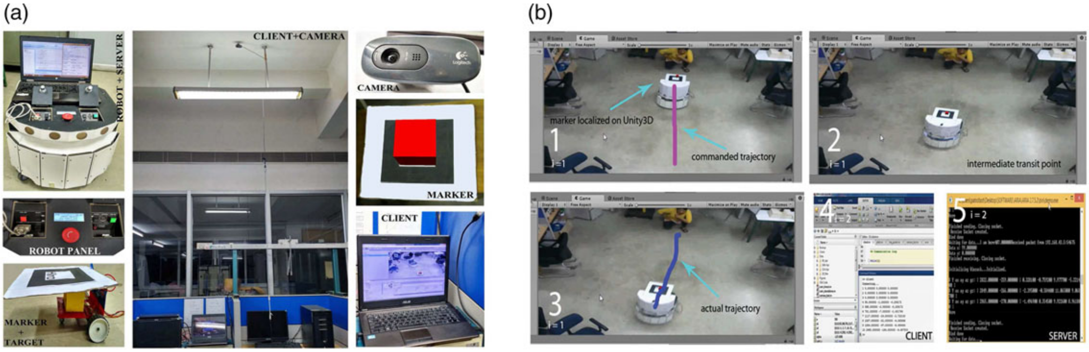


Figure 1. *(a) Different elements of the hardware set-up used. (b) Typical experimental execution observed in Unity 3D with client and server communication scripts.*

vii. It can be used for intent inference-based velocity control of the pursuer while dealing with obstacles of different speeds, numbers, dimensions and approach angles.

viii. It helps in minimising the pursuer’s velocity and thereby also minimises control effort.

4. Experimental Set-up

The experimental set-up used for validating the proposed optimal controller has been described under two subsections – the hardware unit and the control architecture.

4.1 Hardware unit

The hardware unit broadly comprises of the mobile units (target, obstacles and pursuer), sensors, a server computer, a client computer and interfacing circuitry of a communication link between the server and the client devices.

The target and the obstacles are small-sized autonomous robots, each having a radius of about 0.1 m, whereas, the pursuer has been configured on a research robotic platform, ‘Patrolbot’ by Adept MobileRobots, LLC. [38]. It is a two-wheeled differential drive system supported by two additional caster wheels at the front and the back of the chassis. Patrolbot has a saturation velocity of 1.8 m/s and a saturation turn rate of 300°/s. The pursuer is equipped with quadrature encoders to measure pose and an inertial measurement unit for measuring acceleration and angular velocity. For localisation of the target and the obstacles, visual markers of different patterns are attached to them, each having a unique, asymmetric pattern.

The actuators of the pursuer are controlled at a lower level by a proportional integral derivative (PID) controller. The PID commands are generated by an internal processor having an embedded firmware. The firmware interfaces with the higher-level optimal controller through a programmable layer called the *server*. The server runs on a piggybacked computer as shown in Fig. 1(a).

An off-board computer that is termed as the *client* performs trajectory planning. It is connected to an overhead camera configured to record pose of the target and the obstacles. Detection and identification of the target and the obstacles are performed using an augmented reality toolkit and ‘Unity 3D’ [39] which is a graphical user interface-based pose mapper application. Example of an experimental validation study as observed in Unity 3D is shown in Fig. 1(b). Localisation data from the software are available in the camera plane, which is subjected to a homogenous transformation for obtaining position and orientation information in the motion plane. The transformation matrix has been designed according to height, tilt and pan of the camera mount. The transformed pose can be expressed with respect to any user-defined local frame placed in the motion plane.

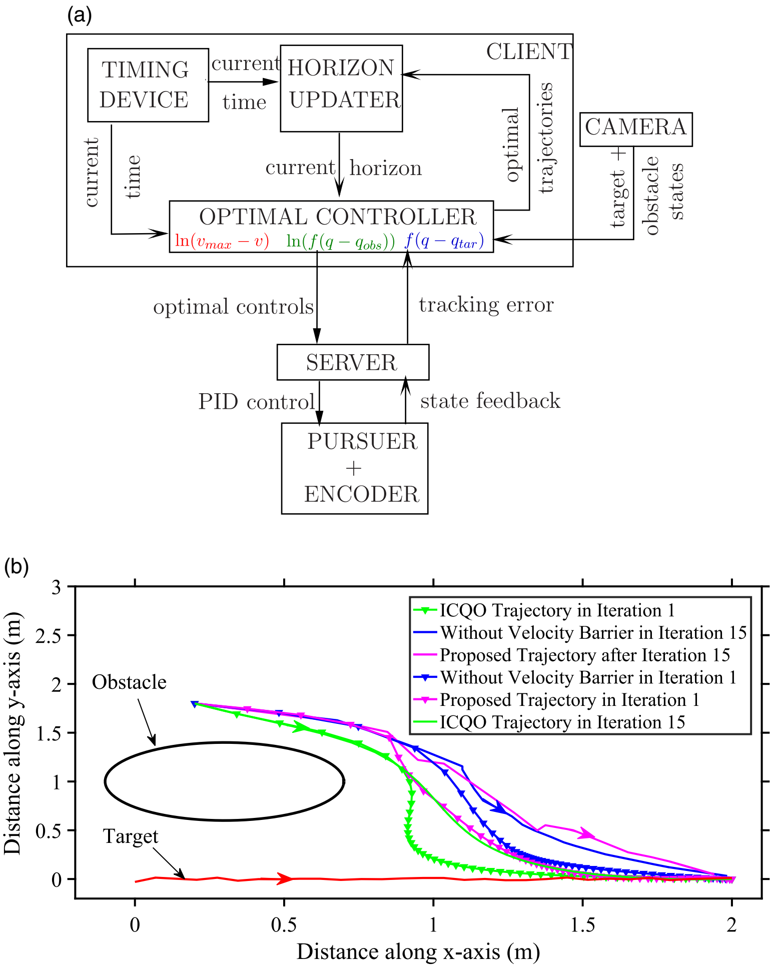

(a)
CLIENT
TIMING DEVICE
current time
HORIZON UPDATER
current horizon
optimal trajectories
CAMERA
target + obstacle states
current time
OPTIMAL CONTROLLER
$\ln(v_{max} - v)$ $\ln(f(q - q_{obs}))$ $f(q - q_{tar})$
optimal controls
tracking error
SERVER
PID control
state feedback
PURSUER + ENCODER
(b)
ICQO Trajectory in Iteration 1
Without Velocity Barrier in Iteration 15
Proposed Trajectory after Iteration 15
Without Velocity Barrier in Iteration 1
Proposed Trajectory in Iteration 1
ICQO Trajectory in Iteration 15
Obstacle
Target
Distance along y-axis (m)
Distance along x-axis (m)
0
0.5
1
1.5
2
2.5
3
0
0.5
1
1.5
2

Figure 2. *(a) An overview of the control architecture explained along with (b) a demonstration of optimal trajectories generated using a shrinking-horizon control. ICQO (integrated control with quadratic optimisation) is another integrated control technique taken from ref. [27]. Tracking and avoidance trajectory without velocity barrier is shown in blue.*

The server is configured to communicate with the client through an ad hoc wireless network, which has been designed with User Datagram Protocol (UDP) [40]. The client then computes the optimal trajectory and serially transmits datapackets containing optimal control commands to the server for execution. We have chosen UDP over a more popular Transmission Control Protocol (TCP), because tracking is a real-time application which prioritises fast data transfer over data accuracy. The transmission has been designed with a fault detection routine (Visual C++/Matlab) that sequentially monitors the virtual outbound and inbound transmission sockets and resets the communication link when period of inactivity exceeds a certain threshold.

4.2 Control architecture

The architecture, as illustrated in Fig. 2(a), describes a control loop which operates sequentially as the client and the server. The online implementation uses a shrinking horizon policy (Fig. 2(b)) which is configured to consider the entire mission horizon in the first iteration, and is better suited to provide

https://doi.org/10.1017/S0263574721001764 Published online by Cambridge University Press

a backup plan in case a solution is unavailable at any $t \in [0, T_{int}]$. The article in ref. [24] confirms that shrinking-horizon policy consumes lower energy, takes lower computation time and provides better transient performance than receding-horizon model predictive control.

Client – Updating the trajectory planning horizon, computing the optimal controls and localising the target and the obstacles are the major activities of the client routine. It may be recalled that the target and the obstacles may have specific motion models, but those models are assumed to be unknown to the pursuer. This necessitates the assessment of intent which can be done by a number of different methods already discussed. In the present control architecture, motion estimation of the target and the obstacles is carried out by extrapolating a polynomial history of their states, using the Neville–Aitken algorithm (see Appendix B). Thereby, the boundary value problem is solved in real time to generate the optimal trajectories. Under a variable length planning horizon approach, only a part of the computed optimal trajectory is transmitted to the server. The set of optimal control sequence that is actually used in a planning iteration, is defined by an interval called the 'update horizon'. Length of update horizon is determined by the data rate of the slowest sensor (e.g., camera). Since the mission time is fixed, the length of the planning horizon reduces as time progresses. Upon exhaustion of the update horizon, the planning horizon is updated with the knowledge of current states measured by the sensors (encoder/IMU feedback from server and camera from client) and an estimation of intent. A new plan is established and the action of replanning is reflected by the shifting of trajectories in subsequent iterations as shown in Fig. 2(b).

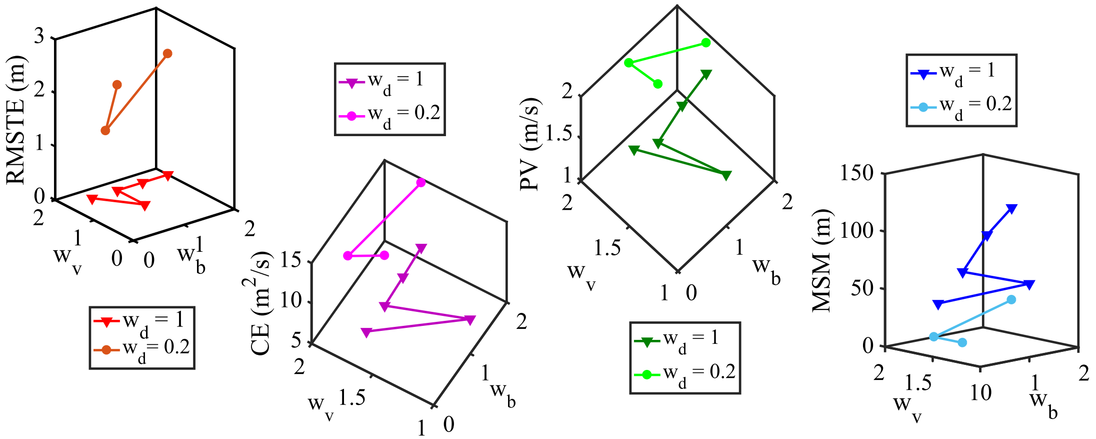
Server – The client communicates with the server via a wireless serial communication pathway which transmits control variables to the server and fetches tracking error from the server. The server, on the other hand, sends PID motion commands to the pursuer through the firmware interface and retrieves the current state feedback from the encoders and inertial measurement unit attached to the pursuer. It may be noted that the optimal trajectory planned by the client is the reference and the PID controller with state feedback is the closed-loop reference-tracking controller. The state feedback is used to compute the tracking error and stabilises the open-loop dataflow of the trajectory planner.

Communication – In the event of a malfunction of the client–server communication, the current plan is terminated and the horizon updater re-initiates the trajectory planner. Unlike error-dynamics-based reference-tracking control, this architecture can reconfigure the reference trajectory of the closed-loop controller. This feature, in association with the resilience property of shrinking-horizon policy, can accommodate unanticipated changes in the intent of the target and the obstacles. The pseudocodes for the client and server routines have been provided in Algorithm-*Client* and Algorithm-*Server* (Appendix D), and Algorithm-*Optimization* describes the routine for formulating desired or reference optimal trajectory.

5. Results and Analysis

In this section, we have explored the relation between the weighing constants and their combined effects on the performance and safety metrics for dynamic pursuit. The role played by logarithmic penalty on steering and velocity control of the pursuer have been discussed in the light of intent knowledge. An approximation of the intent is assumed to be obtained from a polynomial reconstruction of the past (measured) trajectories of the target and the obstacles, and then extrapolating in time. Experiments demonstrating validation of the proposed strategy on a real robot in comparison with a recent literature on similar approach have been discussed. The results have been analysed in the context of a variety of autonomous driving scenarios that have been recognised and regulated by competent authorities. Comparison with strategically different benchmark techniques further explain the advantages of the optimal formulation presented in this paper.

https://doi.org/10.1017/S0263574721001764 Published online by Cambridge University Press



RMSTE: Root Mean Square Tracking Error, CE : Control Effort, PV: Peak Velocity, MSM: Minimum Safety Margin

Figure 3. *Effects of weighing parameters on the performance and safety metrics – a guideline to weight selection.*

5.1 Selection of weighing constants

The weighing constants w_d, w_b and w_v are design parameters, configured to assign relative priorities to tracking, collision avoidance/steering and regulation of velocity/CE respectively. Individual impacts of these parameters on a particular subtask is difficult to determine. Hence, they need to be studied as an ensemble. The following observations indicate a qualitative overview of the roles of these parameters:

- A higher w_d reduces the root mean square (RMS) tracking error, but at the cost of a high CE. It has a suppressive effect on the penalty functions.
- A higher w_v reduces both CE and the peak velocity (PV) attained but relaxes the MSM from the obstacle(s).
- Increasing w_b helps maintaining a larger safety margin, but increases CE.
- Compared to individual magnitudes of w_d and w_v, the ratio $w = \frac{w_d}{w_v}$ is a more appropriate indicator for determining CE.

Figure 3 gives a guideline about how the performance and safety metrics like RMS tracking error, CE, PV and MSM vary with changes in the weights. For a particular application, a suitable set of values for the parameters may be chosen according to this guideline, depending on the situation. For example, a larger penalty on MSM can be imposed by selecting weights such that $w < w_b$. In each planning iteration, the weights can be reassigned depending on the estimated motion of the target and the obstacles.

5.2 Logarithmic barrier: Steering and velocity control

Steering around a static obstacle (stationary but movable objects like a human or another vehicle, or a purely non-movable entity like a traffic island) may not be feasible in presence of clutter, where access to many of the possible routes are restricted; or in case, the pursuer is already moving at a high speed. In the following simulation, we have studied the effects of both steering and velocity control in designing a collision avoidance trajectory for a single static obstacle.

Static obstacle – *Example 1:* An accelerating target is simulated to move along an arc from the midpoint of the workspace (0,0) m to the position $(-1.56, 0.79)$ m. A static obstacle of radius 0.2 m is located at $(-\,0.5, 0.5)$ m. The pursuer starts from $(1, 0.2)$ m and makes an attempt to track and intercept the target after 5 s. The tracking weight w_d is maintained at a low priority of 0.2, and a higher priority

https://doi.org/10.1017/S0263574721001764 Published online by Cambridge University Press

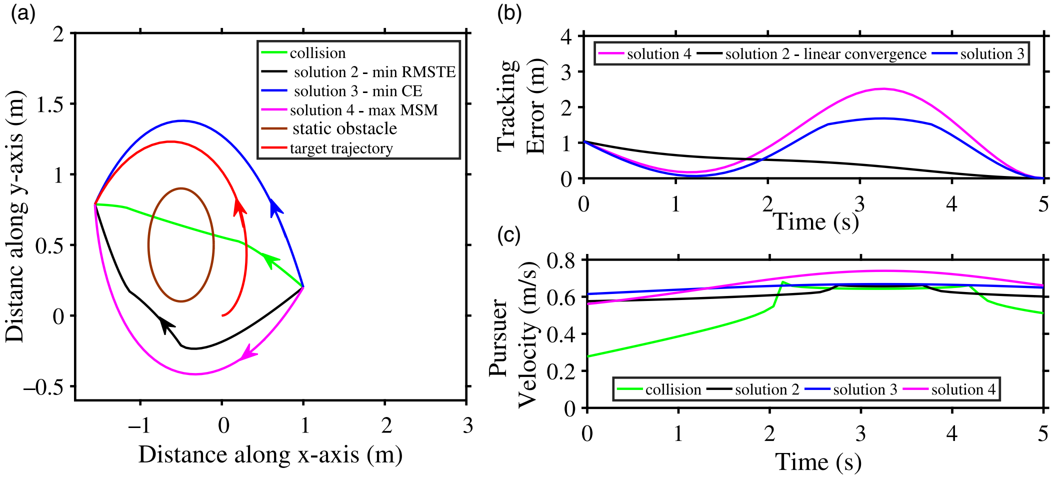


Figure 4. *Steering and velocity control – how to select design parameters in avoiding a static obstacle.*

is assigned to collision avoidance. With w_v selected as 1, if low values are chosen for both velocity barrier $v_{max} = 0.8$ m/s and steering weight $w_b = 0.3$, it results in a collision (green curve). Out of the three successful trajectories shown in Fig. 4(a), the blue curve achieves the objective by increasing the weight to velocity barrier w_v to 1.6. The magenta curve succeeds by increasing the velocity upper bound v_{max} to 1.1 m/s. Whereas, the black curve increases the steering weight w_b to 0.8. However, the lowest RMS tracking error (0.27 m) is achieved by steering control (black curve) (see Fig. 4(b)), while the lowest CE (1.92 m^2/s) is achieved (see Fig. 4(c)) by enforcing a stricter velocity barrier (blue curve).

In the next simulation example, we have explored the relation between steering and velocity control using logarithmic barrier for avoiding moving obstacles having different approach angles. In general, approach angles between 180° and 360° (obstacles crossing at right angles or approaching from behind) and head-on approach (90°) are considered difficult to avoid. Reports [3] conclude that 60% of collisions are accounted for transverse and head-on approach angles.

Obstacle approach angle –*Example 2:* A target is assumed to move linearly along y-axis from (7, 6) m with a constant velocity of 0.2 m/s. The pursuer starts tracking the target from (3, 3) m and attempts to intercept after 5 s.

(a) In the first case, an obstacle starts moving from (7, 0) m with a velocity of 2 m/s along 135°, making an almost right angle while crossing path with the pursuer. We assign $v_{max} = 2$ m/s assuming that the saturation velocity for the pursuer is 2 m/s, which means the obstacle is always faster than the pursuer. Allowable MSM is 0.5 m. The weights, w_d and w_v have been maintained at 0.2 and 1, respectively, and the effects of w_b and v_{max} are studied.

Figure 5(a) shows that low steering weight $w_b = 0.2$ and high velocity barrier $v_{max} = 1.8$ m/s cause the pursuer to collide with the obstacle (red curve; assume that the obstacle has no avoidance policy). Constraint violation can be avoided either by increasing the steering weight to 0.8 or slowing the vehicle down by decreasing v_{max} to 1.3 m/s. Simulation shows that steering action yields a better safety score (MSM=0.96 m, blue curve) than velocity control (MSM = 0.73 m, magenta curve), but at the cost of a higher CE and possibly passenger discomfort caused by the oscillations in velocity profile. In fact, a hybrid control has been observed to solve both issues and at the expense of a lower control energy, as illustrated in Fig. 5(b).

https://doi.org/10.1017/S0263574721001764 Published online by Cambridge University Press

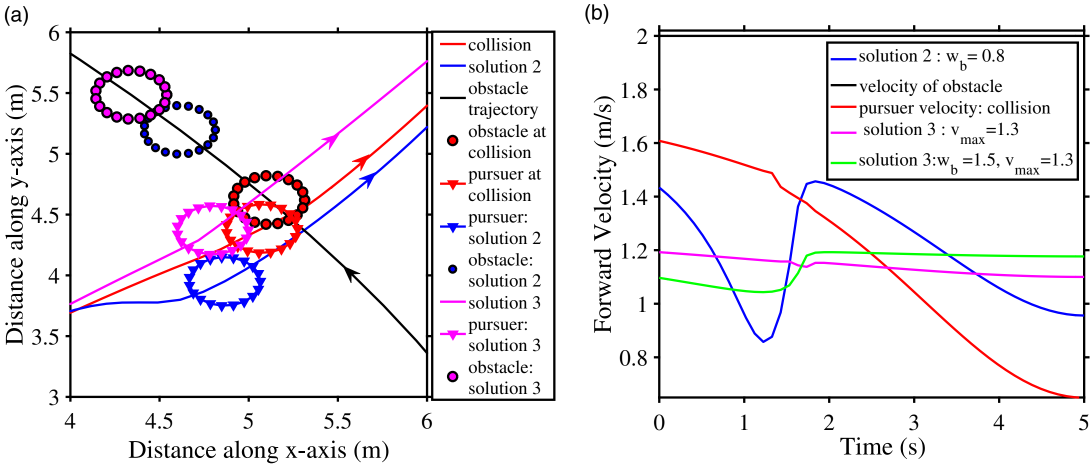


Figure 5. *Steering and velocity control – how to avoid a fast-moving obstacle crossing transversely.*

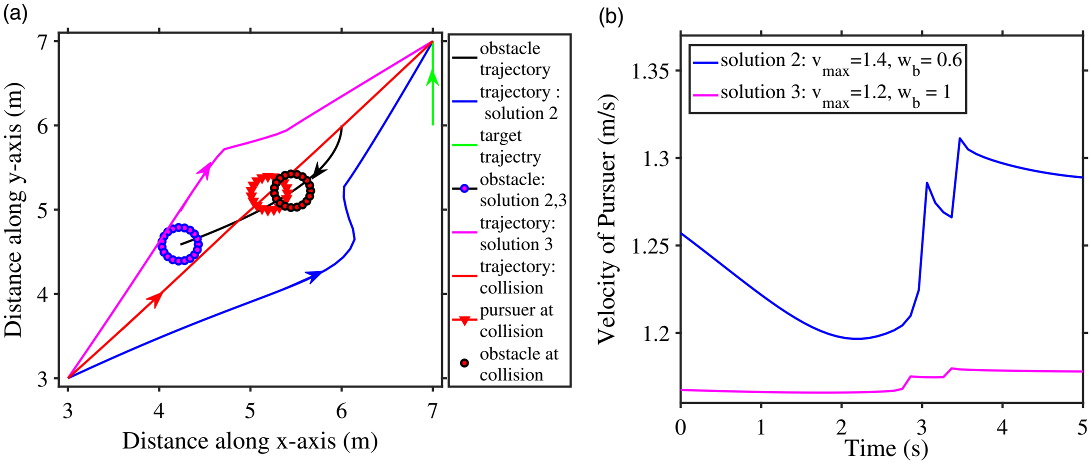


Figure 6. *Steering and velocity control – how to avoid a head-on collision.*

(b) In the second case, the simulated obstacle starts moving from (6,6) m with a velocity of 0.2 m/s along 225°, making an almost head-on approach towards the pursuer. Allowable MSM is 0.5 m. The weights w_d and w_v have been maintained at 0.2 and 1 respectively. A velocity barrier of 2 m/s results in a head-on collision (red curve, see Fig. 6(a)), irrespective of the steering weight. Simulations have been conducted to study whether a reduction in velocity barrier is sufficient to avoid the collision.

Figure 6(b) shows that the magnitude of velocity barrier and steering weight bear an inverse relationship in obtaining a desired solution. If velocity barrier is reduced to $v_{max} = 1.4$ m/s, the steering weight can be chosen as 0.6 to yield desired interception with the target. The resulting trajectory (blue curve) has an RMS tracking error of 1.418 m and a CE of 8 m^2/s over the mission interval. But, if the velocity barrier is reduced further to 1.2 m/s, then the steering action has to be increased to avoid a collision. With $w_b = 1$ and $v_{max} = 1.2$ m/s, the magenta trajectory illustrates an alternative solution with slightly higher RMS tracking error (1.46 m) and a slightly lower CE (7.01 m^2/s). In situations like automatic lane changing and passing over a slower preceding vehicle, the foregoing framework can be applied, provided that the exact parameter values are selected depending on a case-to-case basis.

https://doi.org/10.1017/S0263574721001764 Published online by Cambridge University Press

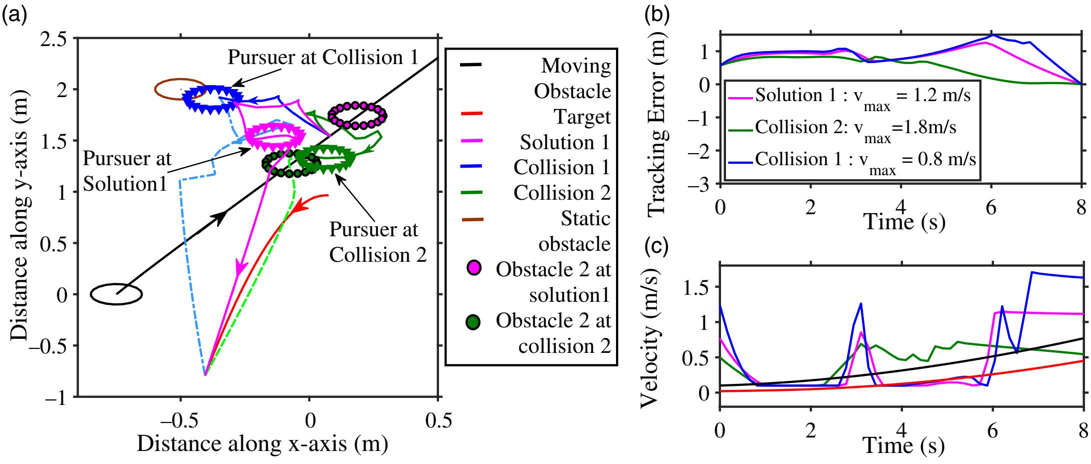


Figure 7. *Velocity control –A slow down approach.*

Statistical reports on automated driving [2] show that the leading causes of crash are unanticipated turning of vehicles at non-signalised junctions and stopping/decelerating actions by the lead vehicle. In this context, the pursuer can avoid collision by maintaining its velocity if it is at a specific distance away from the lead vehicle or it is capable of altering its own speed. The next discussion demonstrates a simulation study showing how the velocity barrier v_{max} can be varied depending on an estimation of the obstacle's motion (intent inference) in order to achieve safe navigation.

Varying the velocity barrier–*Example 3:*

(a) In Fig. 7, an accelerating target is assumed to move from (0.07, 0.96) m. The pursuer which was originally travelling with a velocity barrier of 1.5 m/s starts tracking the target from (0.08, 1.54) m and attempts to intercept after 8 s. Within the region of interest, there is a static obstacle at (– 0.5, 2) m and another accelerating obstacle that starts moving towards the pursuer at an angle of 60° with respect to the x-axis. The moving obstacle is faster than the target, which usually demands tracking to assume a lower priority than collision avoidance. However, the simulation assigns full tracking weight ($w_d = 1$) in order to discourage the pursuer from taking a large steering action in response to the current situation. In accordance, the steering weights to the static and the moving obstacles are maintained at 0.7 and 1.2, respectively, and the weight to velocity barrier is also chosen as 1. The objective is to study the effects of different velocity barriers in achieving a safe trajectory, assuming that the moving obstacle has no avoidance policy of its own. MSM is 0.2 m.

The current velocity barrier results in a collision. Note, in Fig. 7(a), if v_{max} is increased by 20%, the pursuer trajectory (blue) oscillates because of the low steering input and the increased velocity and finally collides with the static obstacle at $t = 2.78$ s. Now, if v_{max} is decreased by 20%, the pursuer (magenta curve) slows down and successfully avoids both the obstacles before intercepting the target. The velocity barrier v_{max} is further decreased to 0.8 m/s. The pursuer collides (green curve) with the moving obstacle at $t = 4.4$ s, since both velocity and steering inputs are low. However, it may be observed from Fig. 7(b), (c) and Table II (collisions marked in bold) that, if the collision could be avoided, tracking trajectory with a lower velocity barrier achieves the lowest RMS tracking error and the lowest CE. In practice, if the obstacle's intent does not exhibit a cooperative attempt to avoidance, it is best to reduce the velocity barrier and enhance the steering effect. Handling unexpected hazards, safe lane change manoeuvres and changing velocity in response to speed limit changes of roads are some of the practical situations that represent the current simulation.

https://doi.org/10.1017/S0263574721001764 Published online by Cambridge University Press

Table II. *Performance metrics for different velocity barriers in Example 3a*

Velocity Barrier(m/s)	**RMSTE**(m)	**CE**$(\mathbf{m}^2/s)$	**MSM-static**(m)	**MSM-moving**(m)
1.8	0.98	5.32	**0.17**	0.27
1.2	0.82	3.34	0.25	0.25
0.8	0.60	2.08	0.54	**0.01**

Table III. *Guidelines for selecting velocity barrier and weights for dynamic navigation –Example 3b*

Configurations	**RMSTE**(m)	**CE**$(\mathrm{m}^2/\mathrm{s})$	**MSM-1**(m)	**MSM-2**(m)	**PV**(m/s)
1: $v_{max} = 1.2$ m/s, $w_b^1 = 2, w_b^2 = 0.2$	0.42	5.5	0.77	1.09	1.01
2: $v_{max} = 1.2$ m/s, $w_b^1 = 0.2, w_b^2 = 2$	0.23	6	0.51	0.43	1.35
3: $v_{max} = 1.8$ m/s, $w_b^1 = 0.2, w_b^2 = 2$	0.18	8.16	0.63	0.68	1
4: $v_{max} = 1.8$ m/s, $w_b^1 = 1, w_b^2 = 1$	0.31	8.23	**0.10**	0.94	1.43
5: $v_{max} = 1.2$ m/s, $w_b^1 = 1, w_b^2 = 1$	0.36	6.13	0.34	0.91	1.04
6: $v_{max} = 0.8$ m/s, $w_b^1 = 1, w_b^2 = 1$	0.54	4.05	0.82	0.93	0.74
7: $v_{max} = 1.2$ m/s, $w_b^1 = 0.2, w_b^2 = 0.2$	0.48	4.05	**0.12**	**0.10**	0.92

(b) Consider a case, where the pursuer encounters two moving obstacles, while tracking an accelerating target that moves from $(-0.25, 0.08)$ m with an initial velocity of 0.36 m/s along a complex trajectory. Time-to-interception is 10 s. The pursuer starts from a position $(-0.3, -0.5)$ m and encounters a fast-moving obstacle, which travels along an arc with an initial velocity of 0.47 m/s from a position $(-0.5, 0.2)$. The pursuer encounters the second obstacle, that moves linearly along the x-axis from $(0.55, 2.39)$ m. MSM is set to 0.2 m. The objective is to study how to select the steering weights and the magnitude of the velocity barrier in a dynamic scenario representing situations like a passing over manoeuvre, navigating a signal-less crossing and moving through a pedestrian-rich area. The weights w_v and w_d are maintained at unity in all simulated configurations in Table III, which mean penalty to tracking and velocity constraint are given full weightage. (Notations for Table III – MSM-i: Minimum Safety Margin w.r.t. obstacle i, w_b^i: Steering weight to obstacle i.)

Referring to Table III, we can observe that a variable steering policy for the two moving obstacles yields quite different results. In configuration 1, the nearer and faster obstacle is prioritised from the steering perspective, while in configuration 2, the condition is reversed. The second configuration yields a lower tracking error, but the safety margins are also reduced. In such a situation, the pursuer can be triggered to speed up and avoid collision as seen in configuration 3. With v_{max} increased to 1.8 m/s to avoid the faster and nearer obstacle and setting a higher weightage to steering of the farther and slower obstacle results in a much reduced RMS tracking error. The safety margins increase too, but at the cost of a higher CE. Note that, an equal and high steering weightage for both the obstacles (configuration 4) does not yield a safe trajectory if the velocity barrier is high. This is because the effects of steering at high velocity and reduction of tracking error are contrasting by nature. It can be safely concluded from the observed performance and safety metrics for configurations 5 and 6 that lowering the velocity barrier improves the safety margin for collision avoidance and also the CE but degrades the RMS tracking error score. The collisions reported for configuration 7 (the values in bold are collisions) are obvious because the steering weights are both low and equal.

(c) The following example demonstrates a simulated tracking scenario implementing the shrinking-horizon model predictive control policy. The simulation involves different dynamic situations encountered in different phases of motion and appropriately summarises the foregoing discussions on velocity and steering control.

https://doi.org/10.1017/S0263574721001764 Published online by Cambridge University Press

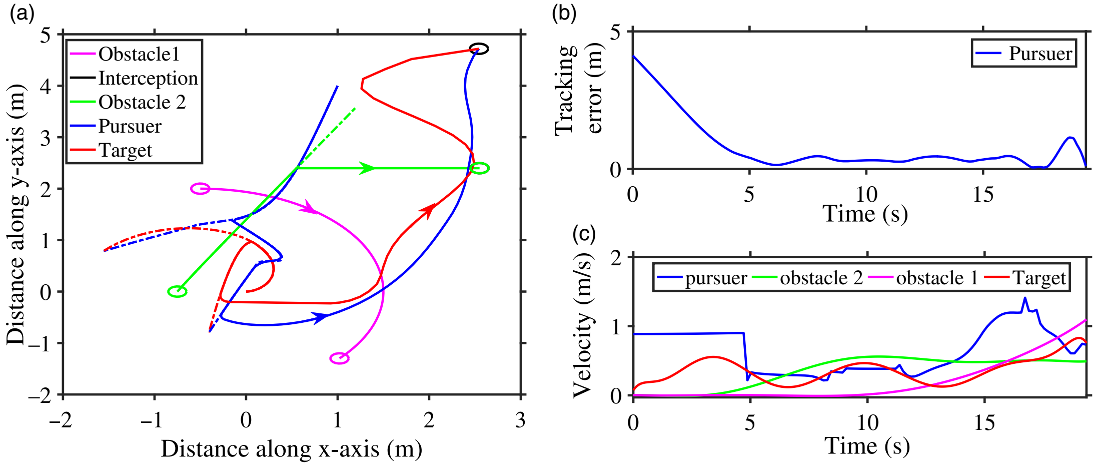


Figure 8. *A shrinking-horizon approach to trajectory planning and control in a complex dynamic scenario.*

Figure 8(a) represents a multiphase trajectory exhibited by a moving target that changes its course without prior intimation to the pursuer. The target starts at (1,4) m and moves with an initial velocity of 0.34 m/s. A single static obstacle at $(-0.5, 2)$ m is avoided by regulating the steering weight to 0.3 and by maintaining a moderate velocity barrier of 1 m/s.

After 2.96 s, the change in target trajectory is estimated by constructing and extrapolating a polynomial. The second obstacle starts moving from $(-0.75, 0)$ and accelerates towards the pursuer. Constrained by both the obstacles and tracking, the velocity barrier is reduced to 0.5, while keeping the steering weights unchanged. The pursuer is capable to achieve a safety margin greater than 0.5 m.

The second phase of motion is interrupted, and a replanning of optimal tracking trajectory is induced by another change in the target trajectory. The planning horizon is reduced in length (duration) with each replanning activity. In this phase, the static obstacle starts moving along an arc at a high speed (gains a velocity of 0.36 m/s and accelerates further), while the second obstacle slows down, changes direction and maintains a constant speed of 0.7 m/s along the x-axis. The magnitude of velocity barrier is increased to 1.8 m/s, and an unequal steering weightage is assigned to avoid the two moving obstacles (configuration 3 of Table III). Finally, interception occurs at 19.36 s. The tracking error (see Fig. 8(b)) and the velocity profiles of the pursuer, target and the obstacles can be seen in Fig. 8(c) for a better understanding of the above description.

5.3 Comparison with other methods

The effectiveness of the proposed method against other techniques has been studied under two aspects. In the first line of study, we have chosen some benchmark methods that consider combined control of velocity and steering of a pursuit vehicle. Three different generic techniques have been selected for this purpose. The first one is the gradient-based artificial potential field (APF) method [41] that shares the integrated control approach but uses an aggressive form of collision avoidance. The second approach is a proportional navigation scheme, called the deviated pursuit (DP) method [42], which shares a non-aggressive avoidance like the proposed strategy. The third technique is rendezvous guidance (RG) [43], which is basically a reactive technique in contrast to optimal planning.

Strategy-based comparison –*Example 6:* A simulation has been designed where the target starts from (7, 6) m and moves along y-axis, with a velocity of 0.2 m/s. Time-to-interception is selected to be $T_{int} = 8$ s after the start. The pursuer starts tracking the moving target from the position, (3, 3) m.

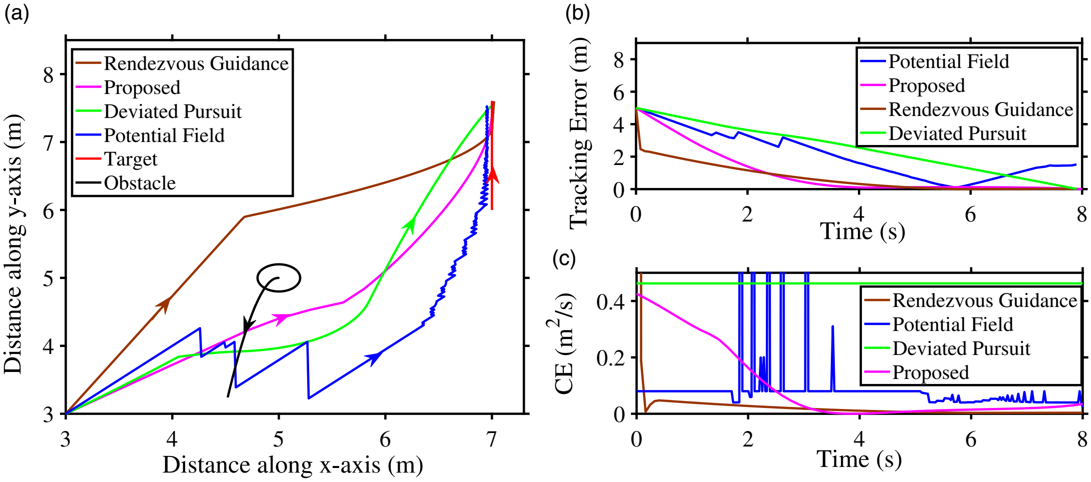


Figure 9. *A comparative study of the proposed method against strategically different benchmark techniques.*

The velocity barrier of the pursuer has been set to $v_{max} = 2$ m/s. An obstacle starts moving at an initial velocity of 0.2 m/s and an initial acceleration of $0.1 m/s^2$ from the position, (5, 5) m. MSM is set to 0.5 m.

Figure 9(a) demonstrates the trajectories planned by the different methods compared here. While all of the methods are found to satisfy the MSM constraint, the oscillations in the APF-driven path (blue) is noteworthy. Undesired noise related to APF is also observed in Fig. 9(b) and (c), which represent the tracking error and CE over the mission horizon. Whereas, RG generates a smooth trajectory (brown) (see Fig. 9(a)), achieving a faster rate of convergence of tracking error (see Fig. 9(b)), but at the cost of a very high CE, as can be seen in Fig. 9(c). In contrast, DP method results in a trajectory (green) with slowest error convergence curve and a constant CE, which is considerably higher that the average CE for the other methods. Our proposed method computes a trajectory (magenta) with optimal steering (see Fig. 9(a)), which has a fast tracking-error convergence rate, numerically comparable to that obtained by RG (see Fig. 9(b)) and requires a CE that has a lower peak value than most of the other methods and a fast decaying profile (Fig. 9(c)).

The second line of study compares the proposed method against a recent literature, which describes a strategically similar integrated control method for cooperative robots [27] but uses a quadratic optimisation formulation in contrast to our proposed logarithmic barrier. This reference method shall be termed as *ICQO* (integrated control with quadratic optimisation) in further analysis. Simulation and experimental validation of both the methods have been shown to demonstrate the advantages of logarithmic penalty function which has been applied to both path and control constraints.

Experimental example: 1 A target has been commanded to move along the x-axis from (0, 0) m with a constant velocity of 0.1 m/s in a laboratory set-up. The interception time, T_{int} was chosen as 20 s so that the experiment can be contained within the field of view (2.5 m × 2.5 m) of an overhead camera. An obstacle was commanded to travel from (0.3, 1) m with a velocity of 0.05 m/s at an angle of 30°. The pursuer starts tracking the target from a position of (0.2, 1.8) m, under a prescribed velocity barrier of 1.8 m/s.

The entire workspace was considered to be the region of interest. The target and moving obstacle were configured as small autonomous robots, each having radius of about 0.1 m. The pursuer has a radius of about 0.22 m. MSM has been set to 0.4 m. The navigation planner's update frequency was decided by repeated trials upon observing the average time required for image processing and transmission and was set to 0.8 s. Sampling time was selected as 400 ms and the maximum idle time (time to reset the

https://doi.org/10.1017/S0263574721001764 Published online by Cambridge University Press

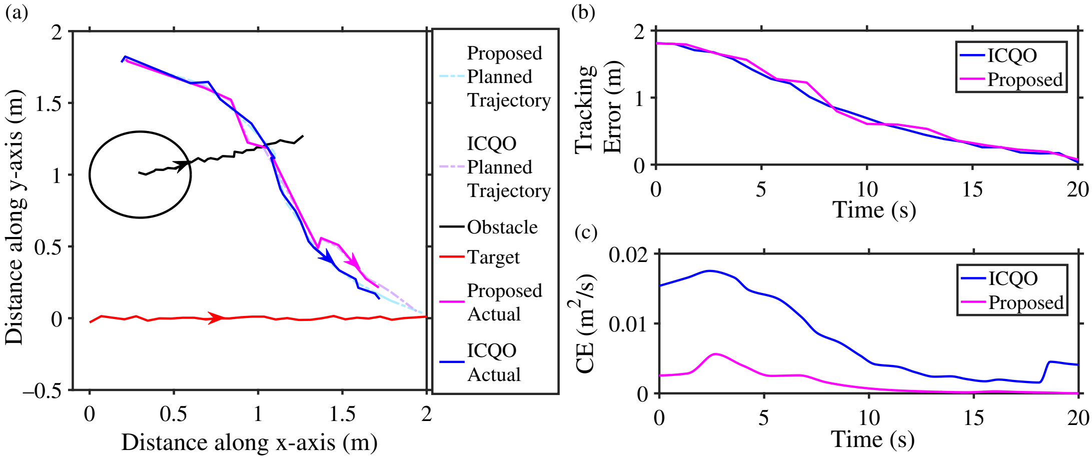


Figure 10. *Comparison of trajectory, tracking error and control effort of proposed method with ICQO [27].*

communication link due to inactivity) was 1.2 s (see client–server algorithm in Appendix D). The range of weighing constants used for the experiment were $0.8 \le w_v \le 1.2$, $0.01 \le w_b \le 0.04$ and $0.1 \le w_d \le 0.2$. The event of the pursuer reaching a δ_{int} ball of 0.4 m around the target, in a neighbourhood of T_{int}, has been assumed to conclude interception and thereby tracking was terminated.

Note that the planned trajectories intercept the target at $T_{int} = 20$ s, but in practice, the actual trajectories are terminated when the pursuer reaches a ball of radius 0.4 m (same as MSM specification) from the target. As observed in Fig. 10(a), both the proposed (magenta) and reference (blue) method (ICQO) yield similar trajectories under the shrinking-horizon control policy and also achieve similar convergence curves of tracking errors (Fig. 10(b)). This gives a general indication of collision avoidance capabilities of the two methods. But, the steering capabilities of the two methods become inconclusive in absence of multiple obstacles of different velocities and approach angles because of the restricted workspace. However, we observe that the logarithmic velocity barrier of our proposed method performs better in terms of CE (see Fig. 10(c)) in achieving a similar tracking error score. It will be further clear from the acceleration (Fig. 11(c)) and angular velocity (Fig. 11(d)) profiles of the pursuer illustrated in Fig. 11 that our proposed method is more efficient in providing passenger comfort due to reduced fluctuations and magnitude of the control variables.

The concept of dynamic window-based trajectory planner which refers to finding the most suitable velocity for a pursuer among a set of admissible velocities by comparing performances with respect to certain evaluation function has been exploited in a number of prior arts. In ref. [44], a hybrid global planning strategy has been devised in conjunction with Jump-A^* algorithm, whereas [45] discusses an approach that involves estimation of distribution of obstacles for generating quick responses. Our method is different from these literatures from the aspect of intent awareness, which evaluates predictive states at a future time corresponding to the end of the current planning horizon. An extensive study on generation of avoidance trajectories for different degrees of clutter and various types of obstacles including static, moving, isolated, connected, rigid and deformable objects have been explained in the paper [46]. However, in contrast to our proposed optimal trajectory with unknown target motion, the target considered in ref. [46] has predefined states and in the absence of obstacles, the vehicle is configured to move to the target along line of sight. For obstacle avoidance, a range-based safety margin protocol similar to ref. [46] has been used. But our proposed method performs obstacle avoidance by minimising optimal cost utilising non-quadratic penalty which does not require addressing constraints like non-intersecting obstacle boundaries, angle-based spirals, and parametric path curvature illustrated in ref. [46]. In general, our proposed method performs functionalities of both reactive (local) and model predictive optimal

https://doi.org/10.1017/S0263574721001764 Published online by Cambridge University Press

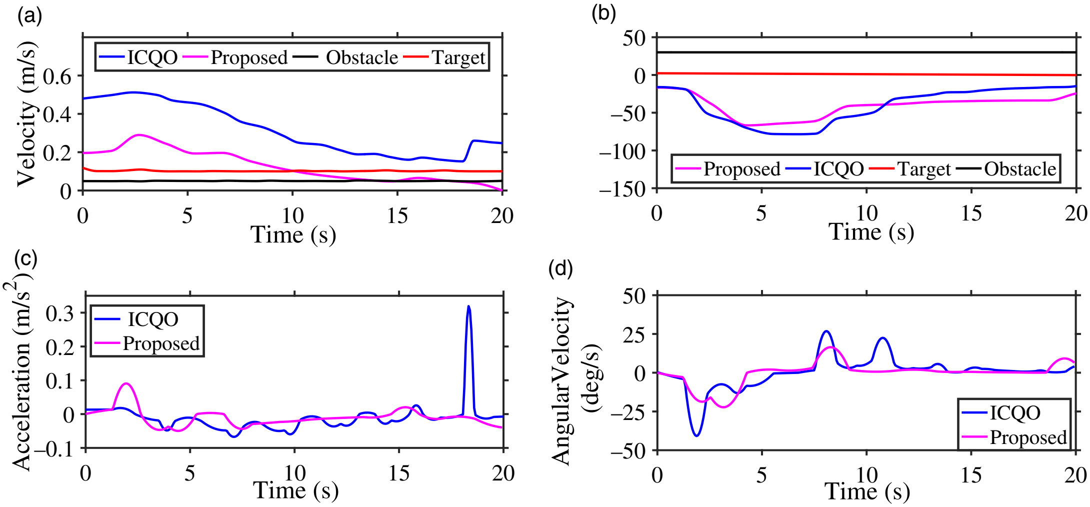


Figure 11. *Comparison of velocity, heading, acceleration and angular velocity of proposed method with ICQO [27].*

controllers (global) in an integrated fashion compared to switching-based sliding mode reactive controller which may possibly attract control mismatch at switching points and additional efforts to balance it. A velocity cone formulation for deciding avoidance direction around an obstacle also appears in ref. [47], which is completely avoided in our proposed method. All obstacles within current sensing range are optimally avoided by a logarithmic barrier function constraining both velocity and safety margin.

6. Conclusion

This paper presents an integrated optimal control approach to find safe navigation routes for dynamic pursuit vehicles, where collision avoidance strategies are influenced by an estimated knowledge of intention of the target and the obstacles. The main contribution of the proposed controller is to demonstrate the flexibility and efficacy of velocity and steering control achieved by a logarithmic penalty-function-based nonlinear optimization routine. Practical implementation of the proposed control strategy has been done using a shrinking horizon policy, which offers better chances of finding solutions than receding horizon control by applying a backup plan, which is complete till interception. Extensive simulations and hardware experiments have been performed and analysed to illustrate the merits of logarithmic velocity barrier in a variety of autonomous driving situations, including dynamic motion, clutter, unanticipated changes in target motion, accelerating/decelerating obstacles, obstacles approaching at different angles, etc. The proposed control strategy has been found to offer lower tracking error and lower CE in comparison to existing literature. This being said, the possible directions for extending the current work are many. Stability analysis of a pursuit vehicle under the proposed scheme, studies on impacts of cooperative/anti-cooperative behaviour of obstacles on the optimal planner, enhancing robustness to disturbances and exploring automatic weight allocation system for the optimal controller are some of the possible routes of future work.

Disclosure

Karnika Biswas, Indrani Kar and Eric Feron declare no competing interests or financial ties to disclose. This research received no specific grant from any funding agency, commercial or not-for-profit sectors.

Supplementary Material

To view supplementary material for this article, please visit https://doi.org/10.1017/S0263574721001764.

References

[1] J. L. Campbell, J. L. Brown, J. S. Graving, C. M. Richard, M. G. Lichty, L. P. Bacon, J. F. Morgan, H. Li, W. N. Williams and T. Sanquist, Human Factors Design Guidance for Level 2 and Level 3 Automated Driving Concepts, *(Report No. DOT HS* 812 *555)* (National Highway Traffic Safety Administration, Washington, DC, August 2018).

[2] C. Becker, L. Yount, S. Rosen-Levy and J. Brewer, Functional Safety Assessment of an Automated Lane Centering System, *(Report No. DOT HS 812 573)*, (National Highway Traffic Safety Administration, Washington, DC, August 2018).

[3] E. Swanson, M. Yanagisawa, W. G. Najm, F. Foderaro and P. Azeredo, Crash Avoidance Needs and Countermeasure Profiles for Safety Applications based on Light-vehicle-to-pedestrian Communications, *(Report No. DOT HS 812 312)* (National Highway Traffic Safety Administration, Washington, DC, August 2016).

[4] A. Bajcsy, S. Bansal, E. Ratner, C. J. Tomlin and A. D. Dragan, "A robust control framework for human motion prediction," *IEEE Robot. Autom. Lett.* **6**(1), 24–31 (2021).

[5] Y. Cho, J. Han and J. Kim, "Intent inference of ship maneuvering for automatic ship collision avoidance," *IFAC-PapersOnline* **51**(29), 384–388 (2018).

[6] Y. Wang, Y. Ren, S. Elliott and W. Zhang, "Enabling courteous vehicle interactions through game-based and dynamics-aware intent inference," *IEEE Trans. Intell. Vehicles* **5**(2), 217–228 (2020).

[7] A. Patterson, A. Lakshmanan and N. Hovakimyan, "Intent-aware Probabilistic Trajectory Estimation for Collision Prediction with Uncertainty Qualification," *Proceedings of IEEE International Conference on Decision and Control* (2019) pp. 3827–3832.

[8] R. Q. Minguez, I. P. Alonso, D. FernÁndez-Llorca and M. Á. Sotelo, "Pedestrian path, pose and intention prediction through Gaussian process dynamical models and pedestrian activity recognition," *IEEE Trans. Intell. Transport. Syst.* **20**(5), 1803–1814 (2019).

[9] F. Capparella, L. Freda, M. Malagnino and G. Oriolo, "Visual Servoing of a Wheeled Mobile Robot for Intercepting a Moving Object," *Proceedings of IROS'05, IEEE International Conference on Intelligent Robots and Systems* (2005) pp. 2737–2743.

[10] J. Li, Z. Xing, W. Zhang, Y. Lin, and F. Shu, "Vehicle tracking in wireless sensor networks via deep reinforcement learning" *IEEE Sens. Lett.* **4**(3), 1–4 (2020).

[11] H. Qu, S. X. Yang, A. R. Willms and Zr. Yi, "Real-time robot path planning based on a modified pulse-coupled neural network model," IEEE Trans. Neural Netw. 20(11), 1724–1739 (2009).

[12] C. L. Hwang, T. H. Wang and C. C. Wong, "A Dynamic Target Tracking of Car-like Wheeled Robot in a Sensor-network Environment via Fuzzy Decentralized Sliding-mode Grey Prediction Control" *Proceedings of ICRA'07, IEEE International Conference on Robotics and Automation* (2007) pp. 3463–3469.

[13] R. J. Wai and Y. W. Lin, "Adaptive moving-target tracking control of a vision-based mobile robot via a dynamic petri recurrent fuzzy neural network," *IEEE Trans.* Fuzzy Syst. **21**(4), 688–701 (2013).

[14] H. Ren, S. Chen, L. Yang and Y. Zhao, "Optimal path planning and speed control integration strategy for UGVs in static and dynamic environments," *IEEE Trans. Vehicular Technol.* **69**(10), 10619–10629 (2020).

[15] R. Hajiloo, M. Abroshan and A. Khajepour, A. Kasaiezadeh and S. Chen, "Integrated steering and differential braking for emergency collision avoidance in autonomous vehicles," IEEE Trans. Intell. Transport. Syst. Early Access.

[16] K. Berntorp, R. Bai, K.F. Erliksson, C. Danielson, A. Weiss and S. D. Cairano, "Positive invariant sets for safe integrated vehicle motion planning and control," *IEEE Trans. Intell. Veh.* **5**(1), 112–126 (2020).

[17] B. Tuncer and E. Özkan, "Random matrix based extended target tracking with orientation : A new model and inference," *IEEE Trans. Signal Process*. **69**, 1910–1923 (2021).

[18] Y. Shan, B. Zheng, L. Chen, L. Chen and D. Chen, "A reinforcement learning-based adaptive path tracking approach for autonomous driving," *IEEE Trans. Vehicular Technol*. **69**(10), 10581–10595 (2020).

[19] X. Liu, X. Xin, Z. Li and Z. Chen, "Near optimal control based on the tensor-product technique," *IEEE Trans. Circ. Syst. II Exp. Briefs* **64**(5), 560–564 (2017).

[20] H. Zhang, B. De Saporta, F. Dufour, D. Laneuville and A. NÈgre, "Stochastic control of observer trajectories in passive tracking with acoustic signal propagation optimisation," *IET Radar, Sonar Navigat.* **12**(1), 112–120 (2018).

[21] A. Mavrommati, E,. Tzorakoleftherakis, I. Abraham and T. D. Murphey, "Real-time area coverage and target localization using receding-horizon Ergodic exploration," *IEEE Trans. Robot.* **34**(1), 62–80 (2019).

[22] G. Ferri, A. MunafÒ and K. D. LePage, "An autonomous underwater vehicle data-driven control strategy for target tracking," *IEEE J. Oceanic. Eng*. **43**(2), 323–343 (2018).

[23] X. Zhang, W. Jiang, S. Yu, X. Xu and Z. Li, "A dual-level model predictive control scheme for multi-timescale dynamical systems – Extended version," arXiv: 1906.07334v6 (eess) (2021).

[24] S. S. Farahani, R. Majumdar, V. S. Prabhu and S. Soudjani, "Shrinking Horizon model predictive control with signal temporal logic constraints under stochastic disturbances," *IEEE Trans. Autom. Control* **64**(8), 3324–3331 (2019).

[25] H. Zhang, X. Sun and M. Chen, "Azimuth Tracking for Underwater Moving Target Based on MUSIC and Function Approximation," *Proceedings of IEEE International Conference on Mechatronics and Automation* (2018) pp. 449–454.

https://doi.org/10.1017/S0263574721001764 Published online by Cambridge University Press

[26] T. Gurriet, M. Mote, A. Singletary, P. Nilsson, E. Feron and A. D. Ames, "A scalable safety-critical control framework for nonlinear systems," *IEEE Access* **8**, 187249–187275 (2020).

[27] B. Li, Y. Ouyang, Y. Zhang, T. Acarman, Q. Kong and Z. Shao, "Optimal cooperative Maneuver planning for multiple nonholonomic robots in a tiny environment via adaptive-scaling constrained optimization," *IEEE Robot. Autom. Lett.*, Preprint (2021).

[28] E. R. Pinch, *Optimal Control and the Calculus of Variations* (Oxford University Press, Oxford, New York, 1993).

[29] J. T. Betts, "Survey of numerical methods for trajectory optimization," *J. Guid. Cont. Dynam.* **21**(2), 193–207 (1998).

[30] N. Jorge and S. J. Wright, *Numerical Optimization*, Operations Research and *Financial Engineering* (Springer-Verlag, New York, 2006).

[31] J. Kierzenka and L. F. Shampine, "A BVP solver based on residual control and the Maltab PSE," *ACM Trans. Math. Softw.* **27**(3), 299–316 (2001).

[32] T. Nageli, J. Alonso-Mora, A. Domahidi, D. Rus and O. Hilliges, "Real-time motion planning for aerial videography with dynamic obstacle avoidance and viewpoint optimization," *IEEE Robot. Automat. Lett.* **2**(3), 1696–1703 (2017).

[33] W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, *Numerical Recipes in C : The Art of Scientific Computing* (Cambridge University Press, 2002).

[34] P. Waltman, "Existence and uniqueness of solutions of boundary value problems for two dimensional systems of nonlinear differential equations," *Trans. Am. Math. Soc.* **153**, 223–234 (1971).

[35] P. B. Bailey, L. F. Shampine and P. E. Waltman, "Nonlinear two point boundary value problems," In: *Computational Mathematics : Theory and Mathematics (General)*, Lecture Notes in Mathematics in *Science and Engineering* (Academic Press, Philadelphia, USA, 1968).

[36] R. M. Freund, Penalty and Barrier Methods for Constrained Optimization, *15.084J Nonlinear Programming*, Massachusetts Institute of Technology: MIT OpenCourseWare (2004).

[37] D. Verscheure, M. Diehl, J. De Schutter and J. Swevers, "Recursive Log-barrier Method for On-line Time-optimal Robot Path Tracking," *Proceedings of ACC'09: American Control Conference* (2009) pp. 4134–4140.

[38] Mobilerobotics, https://www.generationrobots.com/media/Patrolbot-PTLB-RevA.pdf

[39] Unity3d, https://unity3d.com//how-to/create-AR-games-in-Unity-efficiently.

[40] D. K. Cho, S. H. Chun, J. G. Ahn, Y. S. Kwon, J. H. Eom and Y. Kim, "A UDP-based protocol for mobile robot control over wireless internet," *Proceedings of ROBOCOMM'09: IEEE International Conference Robot Communication and Coordination* (2009) pp. 1–4.

[41] L. Huang, "Velocity planning for a mobile robot to track a moving target – A potential field approach," *J. Robot. Autonom. Syst.* **57**(2), 55–63 (2009).

[42] F. Belkhouche and B. Belkhouche, "A method for robot navigation toward a moving goal with unknown maneuvers," *Robotica* **23**(6), 709–720 (2005).

[43] F. Kunwar and B. Benhabib, "Rendezvous-guidance trajectory planning for robotic dynamic obstacle avoidance and interception," *IEEE Trans. Syst. Man Cybernet. B Cybernet.* **36**(6), 1432–1441 (2006).

[44] L. Liu, J. Yao, D. He, J. Chen, J. Huang, H. Xu, B. Wang and J. Guo, "Global dynamic path planning fusion algorithm combining jump-A$*$ algorithm and dynamic window approach," *IEEE Access* **9**, 19632–19638 (2021).

[45] D. H. Lee, S. S. Lee, C. K. Ahn, P. Shi and C. C. Lim, "Finite distribution estimation-based dynamic window approach to reliable obstacle avoidance of mobile robot," *IEEE Trans. Indus. Electron.* **68**(10), 9998–10006 (2020).

[46] A. S. Matveev, C. Wang and A.V. Savkin, "Real-time navigation of mobile robots in problems of border patrolling and avoiding collisions with moving and deforming obstacles," *Robot. Autonom. Syst.* **60**(6), 769–788 (2012).

[47] Y. Kuwata, M. T. Wolf, D. Zarzhitsky and T. L. Huntsberger, "Safe maritime autonomous navigation with COLREGS, using velocity obstacles," *IEEE J. Ocean. Eng.* **39**(1), 110–119 (2014).

Cite this article: K. Biswas, I. Kar and E. Feron (2022). "Intent-Aware Optimal Collision Avoidance and Trajectory Planning for a Pursuit Vehicle", Robotica **40**, 2505–2526. https://doi.org/10.1017/S0263574721001764