Safe Trajectory Planning and Tracking Control for a Pursuer

Submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
by
Karnika Biswas

Supervisor:
Dr. Indrani Kar

Department of Electronics and Electrical Engineering
Indian Institute of Technology Guwahati
Guwahati-781039, Assam, India

November 2021

To

Women in Science & Engineering

Certificate

This is to certify that the dissertation entitled "**Safe Trajectory Planning and Tracking Control for a Pursuer**", submitted by **Karnika Biswas** (Roll No.:126102025), a research scholar in the Department of Electronics and Electrical Engineering, Indian Institute of Technology Guwahati, for the award of the degree of **Doctor of Philosophy**, is a record of an original research work carried out by her under my supervision and guidance. The dissertation has fulfilled all requirements as per the regulations of the Institute and in my opinion has achieved the standard satisfactory for submission. The results embodied in this dissertation have not been submitted to any other University or Institute for the award of any degree or diploma.

Indrani Kar

Date: 05.11.2021

Place: Guwahati

Dr. Indrani Kar
Dept. of Electronics and Electrical Engg.,
Indian Institute of Technology Guwahati,
Guwahati - 781039, Assam, India.

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Karnika Biswas

Roll No.: 126102025

Date: November 5, 2021

Place: Guwahati

Acknowledgements

The journey of scientific investigations in the past nine years had been both challenging and rewarding. Having reached this far, it gives me immense pleasure to look back and acknowledge the wonderful people who shaped my path.

I am truly indebted to my supervisor Dr. Indrani Kar, whose continuous inspiration and encouragement has guided me through all difficult situations. My deepest gratitude goes to the members of my Doctoral Committee, Prof. Chitralekha Mahanta, Prof. S.B. Nair and Dr. Srinivasan Krishnaswamy for their meticulous scrutiny of my work and critical feedback during the course of research. I have been privileged to have Dr. Debasattam Pal (Dept. of EE, IIT Bombay) not only as a teacher but a visionary, who opened my eyes to the wonders of 'Control Systems'. I am thankful to Dr. Anjan K. Chakrabarty, who allowed me to attended his lectures on mathematics. I wish to express my gratitude to Prof. Eric Feron for his invaluable suggestions in structuring the meaningful contributions of my research.

I would like to thank all the technical and non-technical staff of the Dept. of EEE for their cooperation. I gratefully acknowledge the assistance received from the HAB, Library, Academics, Student Affairs, Finance and Engineering Sections of IIT, Guwahati on many different occasions.

Colleagues became friends at the campus; friendship that I cherish and wish to carry forward in time. I would lovingly thank them all, without naming anyone in particular, in fear of inadvertently missing out anyone. My heartiest thanks goes to the drama group, 'Lubdhak' that gave me a wonderful co-curricular occupation at IIT, Guwahati.

Last but not the least, I wish to acknowledge the patience and enthusiasm that I have received from my family during this long journey. Without the love and support of my parents, my husband and my daughter, this dissertation would not have seen today's light.

Date: November, 2021 Karnika Biswas

Abstract

This dissertation aims to address some practical issues with guidance and control of pursuit vehicles; issues that demand attention regarding safety and performance optimization. After introducing the state-of-the-art navigation and control techniques, which served as motivations to this work, the challenges and scope of investigations pertaining to the precedent objectives have been identified. A 'pursuit vehicle' or 'pursuer' is an autonomous mobile system which has been deployed to track and intercept a target, wherein the target is characterized in exhibiting motion unknown to the pursuer. Applications including but not limited to surveillance, inspection, monitoring, man-machine interaction and assistance, industrial and service robotics, exploration and rehabilitation robotics, automated cars and the likes benefit from pursuit vehicles.

Traditionally, guidance and control of pursuit vehicles have been achieved using first order gradient descent along the surface of an artificial potential field. The dissertation illustrates optimal path planning with second order gradient descent directions which exhibit faster convergence and reduced oscillations than linear methods in both sparse and narrow, populated environments. Apart from target-tracking, it is required to address other path and control limitations pertaining to the pursuer's performance in accordance with the dynamic setting where the motion takes place. For example, questions like – how would a pursuer minimize expenditure of energy, regulate velocity depending on motion of surrounding obstacles or how would the pursuer maintain a predefined safety margin from an obstacle if the obstacle starts moving suddenly, need to be addressed. These challenges become pertinent if the pursuer is configured to move in restricted physical

spaces like narrow corridors and aisles having typical lane assignments.

To answer these questions, multi-objective optimal moving-target tracking controllers have been developed from the perspectives of free and fixed planning horizons. While a free planning horizon provides flexibility in crowded and uncertain workspaces, a fixed horizon is often preferred for quantification of performance metrics such as control effort required, pursuer's velocity, acceleration, safety margin achieved during dynamic update of trajectories and tracking error. Strategic design of an optimal backup trajectory for navigating narrow spaces like corridors, hallways etc. have been developed, with special emphasis to computation of optimal switching conditions between the tracking and collision avoidance modes of navigation. A technique to obtain smooth, jerk-free transition between tracking and collision-avoidance control variables has been proposed, which offers a novel safety assurance even when the obstacle exhibits unanticipated motion. The limitations of reactive control have been addressed by adopting the advantages of global optimal planning. A shrinking horizon based sub-optimal control strategy has been developed that integrates the merits of both short and long-term navigation guidance. Intent-awareness pertaining to the motion of the target and the obstacles have been incorporated into the optimal plan and logarithmic penalties have been imposed on path and control violations to achieve optimal constraint management and safe navigation under speed and lane restrictions.

To implement the designed optimal plans, a Lyapunov-stable local tracking controller has been developed based on actual state feedback. In cases where state feedback may become unreliable due to disturbances like wheel-slippage, an observer based state and slip estimation technique has been proposed, which can generate accurate state feedback using optical flow and avoid the need to replace the existing optimal tracking control by more complicated and computationally intensive adaptive or disturbance rejection methods. Both linear and non-linear observers have been studied in this regard. In conclusion, the main contributions of the dissertation have been summarized and future directions of research and development have been identified.

Contents

List of Figures

List of Tables

List of Abbreviations

APF	:	Artificial Potential Filed.
HPF	:	Harmonic Potential Filed.
GA	:	Genetic Algorithm.
GD	:	Gradient Descent.
GDM	:	Gradient Descent with Momentum.
NM	:	Newton's Method.
LMA	:	Levenberg Marquardt Algorithm.
CNG	:	Circular Navigation Guidance.
ODE	:	Ordinary Differential Equation.
BVP	:	Boundary Value Problem.
NAA	:	Neville Aitken Algorithm.
LO	:	Luenberger Observer.
EKF	:	Extended Kalman Filter.
IMU	:	Inertial Measurement Unit.
UDP	:	User Datagram Protocol.
PID	:	Proportional-Integral-Derivative.
MPC	:	Model Predictive Control.
RHC	:	Receeding Horizon Control.
RMS	:	Root Mean Square.
MSM	:	Minimum Safety Margin.
PV	:	Peak Velocity.

Chapter 1

Introduction

1.1 Prelude

Autonomy of vehicular (automobiles, robots, guided carts, industrial and retail carriers and the likes) motion is a wide area of research, that has been investigated intensively over the last couple of decades. The focal points of investigation point towards some important aspects of control including but not limited to safety, optimal performance, stability, robustness and implementability of designed trajectories with available resources. For practical purposes, it is often advised to obtain trade-offs in case of conflicting constraints, which is usually performed optimally on a case-to-case basis. This dissertation deals with a subset of autonomous navigation pertaining to tracking robots or pursuit vehicles. Navigation planners and control strategies have been developed for indoor tracking robots intended to carry merchandise and people under some special circumstances. The target may be another vehicle or a human proponent whose motion is unknown. Thereby, the workspace is assumed to include (but not limited to) environments such as corridors, lanes, alleys, hallways etc. made of different surface materials like concrete, hardwood, carpet and the likes and having varying degrees of clutter. The circumstances investigated in this dissertation deal with navigation control problems encountered in relation to these environments under the effects of dynamic changes causes by motion, possibly erroneous state measurements, disturbances

and path and control constraints. The dissertation has been divided into five chapters which are objectively related but each introducing at least one new concept or challenge and solution thereof. Motivation, existing related literature survey and scope of the proposed contributions of each chapter have been briefly discussed in the following paragraphs, as a prelude to the main discussions.

1.1.1 Outline of Chapter 2

Artificial potential field (APF) is a widely used optimal path finding algorithm which can handle collision avoidance and navigation towards a goal simultaneously. Local minima traps and oscillatory behaviour are major drawbacks of this method. A solution to the local minima problem related to unreachability of target due to an obstacle in close vicinity has been solved in [1]. Jia et al. [8] have proposed a new attractive function to solve the goal unreacheability problem for concave obstacles. Cen et al. [7] have proposed a coordinating field in which an attractive force comprising two orthogonal components having adaptive assignment of relative strengths have been defined.

Genetic algorithm (GA) combined with potential field technique has been proposed in [6] for path planning of mobile robots among moving obstacles. Huang [3] have also contributed towards path planning in a dynamic environment. Kim and Khosla [3] have proposed a harmonic potential function to address the issue of local minima, wherein harmonic potential field (HPF) drives the Laplacian of the potential function to zero in all positions other than the target. Superposition of harmonic functions proposed by [5] have been found to result in local-minima-free potential surface in presence of clutter.

While sparsely populated environments generate sluggish response with linear gradient descent, oscillations are generated in narrow valleys that have been shaped by multiple obstacles. Adequate literatures are not available on this problem except a treatise by Ren et al. [2]. In this context, investigations in Chapter 2 focus on reduction of oscillations in a cluttered workspace and finding fast-converging smooth trajectories through narrow corridors using second order gradient based

search directions.

1.1.2 Outline of Chapter 3

Primary disadvantage of gradient based methods is the possibility of a deadlock leading to loss of continuity in the trajectory. In addition, it is usually difficult to adjust the weighing parameters of artificial potential field for a dynamic enviroment. As a remedy, optimal controllers have been designed in Chapter 3, firstly because optimality inculcates future planning beyond the currently observed situation and secondly, due to its capability to handle trade-offs between performance and safety requirements.

Usually, the most important constraints appear in terms of safety, comfort and conservation of energy/time [10], [9]. A recent study [11] suggests an observer-based control scheme for a leader-follower system applicable in moving-target tracking scenario. Free finite-time interception of moving targets is fairly common in missile guidance problems [16], [10], [17], but rather uncommon in mobile robotics. In velocity-pursuit algorithm [12], line-of-sight tracking [13] and rendezvous guidance technique [18], attempts have been made to incorporate variants of the fundamental proportional navigation scheme in mobile robot motion, but they lack elaborate discussions on optimality of solutions. Moving-target tracking have also been implemented using other techniques like Lyapunov based control [14], on-line prediction of interception point [16] and finite-time optimal switching-surface design for convergence of tracking error [13]. In [18], the effects of both time-variant and time-invariant controllers for double pendulum system have been studied for fixed terminal-time, but free terminal-state. Bang-bang optimal control has been designed for a two-wheeled cart with bounded acceleration and known terminal-time in [12]. In [14] existence of unique optimal trajectory for a mobile robot in intercepting a moving target has been investigated.

Section 3 of this chapter deals with an optimal control problem with free, finite terminal-time and free terminal-state, which offers maximum flexibility subject to clutter and uncertainty. Applications wherein the time and state for catching

the target cannot be pre-conceived benefit from free terminal-time formulations. However, existence of solutions for this optimal control problem cannot be guaranted, which is why Section 5 of this chapter devises a fixed terminal-time optimal control problem. Unlike reactive controllers that are incapable to look beyond a short-term goal and learning methods, which fail to achieve optimal performances, the proposed optimal controllers can quantitatively predict run-time cost effectiveness and assign relative priorities to multiple objectives.

1.1.3 Outline of Chapter 4

Moving-target tracking is considered to be one of the primary goals in supervisory, assistive and service robotics. The growing trend of robots sharing workspace with humans has accelerated the need for developing dynamic and safe trajectories. Efficient guidance and control techniques have been reported for driverless automobiles [30], evasive target pursuit [28], multi-agent tracking [31], sensor-network based decentralized tracking [35] and software robotic applications [29]. However, not many literatures address the question of modeling safety paradigms in narrow spaces like corridors and aisles.

Performance and safety constraints can be handled in a decoupled fashion as suggested by Gurriet et al. [43], or with an integrated approach as in [41]. Although, a number of methods are available to reactively handle dynamic obstacles, very few discussions are available on design of safe backup plan subject to highly limited reachable state-space. This chapter introduces a 'priority' driven guidance mechanism to ensure safety in tracking applications, that optimally generates conditions for switching between tracking and collision avoidance controllers. Determination of optimal termination policy for anti-collision manoeuvre and sustenance of safety criteria until resumption of tracking are key features of this chapter.

Literatures available on strategic guidance is immensely diversified based on the medium of travel, constraints, performance metrics, control schemes and the dynamical systems they are applied to. Widely practiced methods include reactive

mechanism [12, 32, 34], relying on local sensing and action, gradient based path generation and control [3, 33], non-linear programming [14], involving an optimization problem and adaptive control [38], that looks forward to enhance efficiency by parameter adaptation. Over the last decade, these basic paradigms have been thoroughly upgraded to advanced algorithms pertaining to performance and safety features like disturbance rejection [39], uncertainty handling [42] and physical and kinodynamic constraint management [37, 40]. A perturbation-insensitive neural network topology [27] and a swarm-optimized fuzzy control method [24] have been reported in path-following literature. Later, a recurrent neural network with fuzzy decisions has been suggested in [25] for dealing with uncertainty in target motion. More recently, supervized learning [36] supported by visual cues have been found promising. It is universally accepted that, the switching point in decoupled controllers [43, 44] generate disturbances in the control channel which may lead to undesired mechanical vibrations. Chapter 4 proposes a smoothening filter for seamless transition of trajectories by dampening the switching noise.

1.1.4 Outline of Chapter 5

Comprehensive guidelines on safety have been formulated [45, 46, 47] with reference to self-driven automobiles. It may be noted that the baseline regulations remain similar for space exploring and commercial vehicles deployed in tracking, inspection, surveillance and service industries. In this context, Chapter 5 explores the scope of safe navigation planners for mobile pursuit vehicles (tracking robots) with speed and lane restrictions in compliance with the foregoing safety guidelines. Dynamic navigation is immensely benefitted from integrated planning method incorporating intent-awareness.

Intent inference may include the predicted goal of a moving vehicle or a human representing a target or obstacles. Existing literatures describe a probabilistic approach [49] to classifying intended motion of ships, robustness enhancement to Bayesian prediction [48] and mutual-intent decoding and planning between a number of vehicles [50]. A hybrid model composed of deterministic polynomials

and Gaussian distribution of obstacle positions has been proposed in [51]. Gait based probabilistic estimation of future activities of pedestrians have been studied in [52] pertaining to intent-aware cars.

Reactive navigation based on local sensing can be designed according to a number of methods like optical flow gradient [53], distributed sensing [54], and learning algorithms implemented by neural networks [55], fuzzy inferences [56] or a combination thereof [57]. 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 optimization. An integrated speed-steering controller [58] has been reported to leverage this knowledge in computing smooth cubic-spline trajectories, whereas, another integrated controller [59] has been reported to combine proportional speed control with differential braking. Computation of invariant reachable sets accommodating allowable lateral displacements [60], is yet another approach. In [61] the authors have suggested an efficient utilization 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 [62], where a reinforced learning policy has been adopted for velocity and orientation control.

The integrated controller proposed in this chapter applies logarithmic penalties to path and control limit violations and solves a non-quadratic optimization which yields more flexible speed-steering control and greater passenger comfort. This chapter uses a shrinking horizon control strategy for real-time implementation of the proposed plan in contrast to scalable, nonlinear Model Predictive Controller (MPC) demonstrated in [63]. Receding Horizon Control (RHC) strategy has also been explored in underwater surveillance [64] and in a modified form in dynamical systems with unknown, bounded and stochastic disturbances [65]. However, planning over a shrinking horizon has been found to be more resilient [66] in finding a solution than a receding horizon approach.

1.1.5 Outline of Chapter 6

Applying the concept of Model Predictive Control (MPC), iterative execution of reference optimal trajectories can be done with real-time state feedback. A local controller with stable feedback is capable to execute the optimal guidance trajectories in real-time. Advanced features like finite time convergence [67] and improved transient response [68] can further improve the controller design. However, disturbances like wheel slippage may make the state feedback unreliable and in such a scenario, local tracking controller would not be sufficient.

In this chapter, we propose an observer based state and slip estimation technique for generating accurate state feedback which can function independently in conjunction with any existing controller. Unlike commonly attributed mechanical faults such as non-ideal contact patch and deformation of tires [70, 77], our interest is to detect the slip-angle and the slip-ratio cased by sudden dynamicity of motion and predict its evolution with time.

Popular disturbance handling techniques for dynamic systems include disturbance rejection [88], robust controller design [89] and adaptive approaches [87]. However, introduction of new controllers to achieve improved performance is often non-economical. Robust controllers are difficult to design and extremely hard to manifest [90]. Reactive controllers are adaptive but require extensive parameter tuning. Another remedial measure includes replacing the kinematics by a detailed dynamics of the vehicle and substituting the existing controller by a more robust or adaptive scheme [81]. These measures introduce additional cost and complexity [75]. Experimentally obtained dynamics-free sideslip calculation of four-wheeled vehicles have been reported in [85]. Motivated by the precedent work, a modified slip-kinematic model has been developed for the pursuer. The idea of slip 'forecast' also appears in association with anti-lock braking systems reported in [71] and [73]. Discussions on wheel slippage in relation to vehicle stability appear in [69] and an examination of 'wheelie' has been reported in two-wheelers [76].

Adaptive learning [83] for coupled dynamics of helicopter and localization with

disturbance cancellation [82] illustrate state-of-the-art robust estimation methods. Deep learning techniques [84] for localization of high speed objects is a recent advancement in estimation literature. A unified nonlinear observation approach combining both Luenberger and parameter-estimation observers has been presented in [79]. Recently, a non-Bayesian filter using wideband beacons has been proposed for accurate state estimation and prediction of uncertainties in [78]. Observers have also proved to be efficient in analyzing responsiveness of multi-agent systems over classical L_1 gain approach [86]. Unlike the direct estimation of body slip angle for yaw rate control in [74], the aim of this chapter is to improve the reliability of state feedback without introdcing a new controller and using complementary sensing topology in alignment with [72] and [80].

1.2 Overview of Key Contributions

In this chapter, we have introduced the main contributions of the dissertation covered in the following Chapters 2-6, in context of the problems with existing methods, scope of development in the light of relevant literature and general motivation. The key points of the following chapters can be summarized as:

1. Chapter 2: Fast converging and smooth trajectories with reduced oscillations have been developed in flat gradients and narrow passages using gradient based optimal descent of the second order.

2. Chapter 3: Multi-objective optimization have been achieved with free and fixed terminal-time optimal guidance for vehicles in pursuit of moving targets having unknown motion.

3. Chapter 4: Optimal switching conditions have been derived for safe navigation using priority-driven, decoupled tracking and collision avoidance optimal control for a pursuer traversing narrow spaces.

4. Chapter 5: Logarithmic penalty-based intent-aware, integrated optimal controller has been designed for dynamic navigation of pursuit vehicles subject

to lane and speed restrictions using shrinking horizon approach.

5. Chapter 6: Validation of observer based state and wheel-slippage estimation has been performed for improving accuracy of state feedback utilized in the proposed Lyapunov-stable local tracking controller.

Bibliography

[1] S.S. Ge and Y.J. Cui, "New Potential Functions for Mobile Robot Path Planning," *IEEE Transactions on Robotics and Automation* **16**(5), 615–621 (2000).

[2] J. Ren, K.A. McIssac and R.V. Patel, "Modified Newton's Method Applied to Potential Field-Based Navigation for Mobile Robots," *IEEE Transactions on Robotics* **22**(2), 384–391 (2006).

[3] L. Huang, "Velocity planning for a mobile robot to track a moving target-a potential field approach," *Robotics and Autonomous Systems* **57**(2), 55–63 (2009).

[4] J. Kim, and P.K. Khosla, "Real-time obstacle avoidance using harmonic potential functions," *IEEE Transactions on Robotics and Automation* **8**(3), 338–349 (1992).

[5] A.A. Masoud, "Managing the Dynamics of a Harmonic Potential Field-Guided Robot in a Cluttered Environment," *IEEE Transactions on Industrial Electronics* **56**(2), 488–496 (2009).

[6] P. Vadakkepat, K.C. Tan and W.M. Liang, "Managing the Dynamics of a Harmonic Potential Field-Guided Robot in a Cluttered Environment," *Proceedings of the 2000 Congress on Evolutionary Computation* **1**, 256–263 (2000).

[7] Y. Cen, L. Wang and H. Zhang, "Real-time Obstacle Avoidance Strategy for Mobile Robot Based On Improved Coordinating Potential Field with Genetic Algorithm," *IEEE International Conference on Control Applications*, 415–419 (2007).

[8] Q. Jia and X. Wang, "Path planning for mobile robots based on a modified potential model," *International Conference on Mechatronics and Automation*, 4946–4951 (2009).

[9] G. Manor, Z. Ben-Asher and E. Rimon, "Time-optimal Trajectories for a Mobile Robot Under Explicit Acceleration Constraints," *IEEE Transactions on Aerospace and Electronic Systems*, **545**, 2220–2232 (2018).

[10] N. Ganganath, C. Cheng and C.K. Tse, "A Constraint Aware Heuristic Path Planner for Finding Energy Efficient Paths on Uneven Terrains," *IEEE Transactions on Industrial Informatics*, **113**, 601–611 (2015).

[11] M.F. Hassan and M. Hammuda, "Leader-follower Formation Control of Mobile Nonholonomic robots via a New Observer Based Controller," *International Journal of Systems Science*, **517**, 1243–1265 (2020).

[12] Y. Zheng and P. Moore, "The design of time-optimal control for two-wheel driven carts tracking a moving target," *Proceedings of 34th IEEE Conference on Decision and Control*, **4**, 3831–3836 (1995).

[13] Y. Wu, R. Yuan and X. Zheng, "Finite-time tracking controller design for a general class of nonholonomic systems," *Proceedings of 48th IEEE Conference on Decision and Control* **56**(2), 4408–4413 (2009).

[14] M. Lukka, "On the Optimal Searching Tracks for a Moving Target," *SIAM Journal of Applied Mathematics* **32**(1), 126–132 (1977).

[15] C.D. Yang and C.C. Yang, "Optimal pure proportional navigation for maneuvering targets," *IEEE Transactions on Aerospace and Electronic Systems* **33**(3), 949–957 (1997).

[16] E.A. Croft, R.G. Fenton and B. Benhabib, "Time-optimal interception of objects moving along topologically varying paths," *Proceedings of IEEE International Conference on Systems, Man and Cybernetics, Intelligent Systems for the 21st Century* **5**, 4089–4094 (1995).

[17] F. Belkhouche and B. Belkhouche, "A Method for Robot Navigation Toward a Moving Goal with Unknown Maneuvers," *Robotica* **23**(6), 709–720 (2005).

[18] F. Kunwar and B. Benhabib, "Rendezvous-Guidance Trajectory Planning for Robotic Dynamic Obstacle Avoidance and Interception," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* **36**(6), 1432–1441 (2006).

[19] F. Belkhouche, B. Belkhouche and P. Rastgoufard, "Line of sight robot navigation toward a moving goal," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* **36**(2), 255–267 (2006).

[20] L. Huang, "Control approach for tracking a moving target by a wheeled mobile robot with limited velocities," *IET Control Theory Applications* **3**(12), 1565–1577 (2009).

[21] M. Guelman and M. Idan and O.M. Golan, "Three-dimensional minimum energy guidance," *IEEE Transactions on Aerospace and Electronic Systems* **31**(2), 835–841 (1995).

[22] D. Ghose, "True proportional navigation with maneuvering targe," *IEEE Transactions on Aerospace and Electronic Systems* **30**(1), 229–237 (1994).

[23] Z. Ahmad, "An optimal robot tracking controller with application in the finite horizon case," *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics* **5**, 3277–3281 (2000).

[24] C.F. Juang,and Y.C. Chang, "Evolutionary-Group-Based Particle-Swarm-Optimized Fuzzy Controller with Application to Mobile-robot Navigation in Unknown Environments," *IEEE Transactions on Fuzzy Systems* **19**(2), 379–392(2011).

[25] 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 Transactions on Fuzzy Systems* **21**(4), 688–701 (2013).

[26] M. Xu, Z. Pan, H. Lu, Y. Ye, P. Lv and A.E. Rahlibi, "Moving-target Pursuit Algorithm Using Improved Tracking Strategy," *IEEE Transactions on Computational Intelligence and AI in Games* **2**(1), 27–39 (2010).

[27] S.X. Yang and M. Meng, "Neural Network Approaches to Dynamic Collision-free Trajectory Generation," *IEEE Transactions on Systems, Man, Cybernetics B, Cybernetics* **31**(3), 302–318 (2001).

[28] K. Zhou, and S.I. Roumeliotis, "Optimal Motion Strategies for Range-only Constrained Multisensor Target Tracking," *IEEE Transactions on Robotics* **24**(5), 1168–1185 (2008).

[29] M. Zhang, and H.H.T. Liu, "Game-theoretical Persistent Tracking of a Moving Target Using a Unicycle-type Mobile Vehicle," *IEEE Transactions on Industrial Electronics* **61**(11), 6222–6233 (2014).

[30] K.D. Campbell, "Tools for Trustworthy Autonomy: Robust Predictions, Intuitive Control, and Optimized Interaction," (Doctoral Dissertation). University of California, Berkeley (2017).

[31] H. Yang, and Y. Wang, "Formation Optimization and Control for Maneuvering Target Tracking by Mobile Sensing Agents," *IEEE Access* **7**, 32305–32314 (2019).

[32] B. Penin, P.R. Giordano and F. Chaumette, "Vision-Based Reactive Planning for Aggressive Target Tracking While Avoiding Collisions and Occlusions," *IEEE Robotics and Automation Letters* **3**(4), 3725–3732 (2018).

[33] S. Sun, H. Wang, J. Liu and Y. He, "Fast Lyapunov Vector Field Guidance for Standoff Target Tracking Based on Offline Search," *IEEE Access* **7**, 124797–124808 (2019).

[34] E.H. Thyri, E.A. Basso, M. Breivik, K.Y. Pettersen, R. Skjetne and A.M. Lekkas "Reactive collision avoidance for ASVs based on control barrier functions," *IEEE Conference on Control Technology and Applications* Montreal, Canada, 380–387.

[35] J. Li, Z. Xing, W. Zhang, Y. Lin and F. Shu, "Vehicle Tracking in Wireless Sensor Networks via Deep Reinforcement Learning," *IEEE Sensors Letters* **4**(3), 1–4 (2020).

[36] Q. Wu, X. Gong, K. Xu, D. Manocha, J. Dongand J. Wang, "Towards Target-driven Visual Navigation in Indoor Scenes via Generative Imitation Learning," *IEEE Robotics and Automation Letters* **6**(1), 175–182 (2020).

[37] Z. Sun, B. Zhang, L. Cheng and W.J. Zhang, "Application of the redundant servomotor approach to design of path generator with dy-

namic performance improvement," *Mechanism and Machine Theory* **46**(11), 1784–1795 (2011).

[38] C. Huang, S. Lucey and D. Ramanan, "Learning Policies for Adaptive Tracking with Deep Feature Cascade," *International Conference on Computer Vision* Venice, 105–114 (2017).

[39] R.S. Sharma, A. Mondal and L. Behera, "Tracking Control of Mobile Robots in Formation in the Presence of Disturbances," *IEEE Transactions on Industrial Informatics* **171**, 110–123 (2021).

[40] N.P.K. Chan, B. Jayawardhana and H.G. De Marina, "Angle-constrained Formation Control for Circular Mobile Robots," *IEEE Control System Letters* **51**,109–114 (2021).

[41] 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 Robotics and Automation Letters* , Preprint.

[42] I. Batkovik, U. Rosolia, M. Zanon and P. Falcon, "A Robust Scenario MPC Approach for Uncertain Multi-modal Obstacles," *IEEE Control Systems Letters* **513**, 947–952 (2021).

[43] 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).

[44] X. Wu, J. Angeles, T. Zou, J. Yang, H. Li and W Li, "Trajectory Planning with Lamé-Curve Blending for Motor-Saturation Avoidance Upon Mobile-Robot Turning," *IEEE Access* **8**, 58483–58496 (2020).

[45] J.L. Campbell,J.L. Brown, J.S. Graving, C.M. Richard, M.G. Lichty, L.P. Bacon, ... and T. Sanquist, "Human factors design guidance for level 2 and level 3 automated driving concepts," *(Report No. DOT HS 812 555), Washington, DC: National Highway Traffic Safety Administration.* (2018, August).

[46] 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), Washington, DC: National Highway Traffic Safety Administration.* (2018, August).

[47] 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), Washington, DC: National Highway Traffic Safety Administration.* (2016, August).

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

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

[50] 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).

[51] 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*, 3827–3832 (2019).

[52] 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).

[53] F. Capparella, L. Freda, M. Malagnino and G. Oriolo, "Visual servoing of a wheeled mobile robot for intercepting a moving object" **In**:*Proceedings of IROS'05, IEEE International Conference on Intelligent Robots and Systems*, 2737–2743 (2005).

[54] 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).

[55] H. Qu, SX. Yang, AR. Willms and Zr. Yi, "Real-time robot path planning based on a modified pulse-coupled neural network model," *IEEE Transactions on Neural Networks* **20**(11), 1724–1739 (2009).

[56] CL. Hwang, TH. Wang and CC. Wong, "A dynamic target tracking of car-like wheeled robot in a sensor-network environment via fuzzy decentralized sliding-mode Grey Prediction control" **In**:*Proceedings of ICRA'07, IEEE International Conference on Robotics and Automation*, 3463–3469 (2007).

[57] RJ. Wai and YW. Lin, "Adaptive moving-target tracking control of a vision-based mobile robot via a dynamic petri recurrent fuzzy neural network," *IEEE Transactions on Fuzzy Systems* **21**(4), 688–701 (2013).

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

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

[60] 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. Vehicles* **5**(1), 112–126 (2020).

[61] 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).

[62] 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).

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

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

[65] 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).

[66] 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).

[67] T.C. Lee, C.Y. Tsai and K.T. Song, "Fast Parking Control of Mobile Robots: A Motion Planning Approach With Experimental Validation," *IEEE Transactions on Control System Technology* **12**(5), 661–676 (2004).

[68] L. Cheng, Z. Hou, M. Tan and W.J. Zhang, "Tracking Control of a Closed-chain Five-bar Robot with Two Degrees of Freedom by Integration of an Approximation Based Approach and Mechanical Design.," *IEEE Transactions on Systems, Man, Cybernetics - B, Cybernetics* **42**(5), 1470–1479 (2012).

[69] D. Karnopp,"Vehicle Dynamics, Stability and Control," *Mechanical Engineering,CRC Press, (2013)*

[70] T.D. Gillespie,"Fundamentals of Vehicle Dynamics," *Premier Series,SAE International*, (1992).

[71] Y. Oniz and E. Kayacan and O. Kaynak, "A Dynamic Method to Forecast the Wheel Slip for Antilock Braking System and Its Experimental Evaluation," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **39**(2), 551–560 (2009).

[72] W. Zhu and T. Lamarche, "Velocity Estimation by Using Position and Acceleration Sensors," *IEEE Transactions on Industrial Electronics* **54**(5), 2706–2715 (2007).

[73] Y. Zhang, H. Zhao, L. Yuan and H. Chen, "Slip Ratio Estimation for Electric Vehicle with In-Wheel Motors Based on EKF without Detection of Vehicle Velocity," *Chinese Control and Decision Conference (CCDC)* 4427–4432,(2016).

[74] C. Geng, T. Uchida, Y. Hori, "Body Slip Angle Estimation and Control for Electric Vehicle with In-Wheel Motors," *33rd Annual Conference of the IEEE Industrial Electronics Society, IECON* 351–355,(2007).

[75] W. Wang and I. Li and M. Chen and S. Su and S. Hsu, "Dynamic Slip-Ratio Estimation and Control of Antilock Braking Systems Using an Observer-Based Direct Adaptive Fuzzy-Neural Controller," *IEEE Transactions on Industrial Electronics* **56**(5), 1746–1756 (2009).

[76] M. Corno and G. Panzani and S.M. Savaresi, "Traction-Control-Oriented State Estimation for Motorcycles," *IEEE Transactions on Control Systems Technology* **21**(6), 2400–2407 (2013).

[77] D.L. Desages, C. Grand, F.B. Amar and J. Guinot, "Doppler-Based Ground Speed Sensor Fusion and Slip Control for a Wheeled Rover," *IEEE/ASME Transactions on Mechatronics* **14**(4), 484–492 (2009).

[78] D. Fontanelli, F. Shamsfakhr, D. Macii and L. Palopoli, "An Uncertainty-Driven and Observability-Based State Estimator for Nonholonmic Robots," *IEEE Transactions on Instrumentation and Measurement* **70**, 1–12 (2021).

[79] B. Yi and W. Zhang, "On State Observers for Nonlinear Systems: A New Design and a Unifying Framework," *IEEE Transactions on Automatic Control* **64**(3), 1193–1200 (2019).

[80] L. Fusini, J. Hosen, H.H. Helgesen, T.A. Johansen and T.I. Fossen, "Experimental Validation of a Uniformly Semi-Globally Exponentially Stable Non-Linear Observers for GNSS and Camera-Aided Inertial Navigation for Fixed-Wing UAVs," *International Conference on Unmanned Aircraft Systems (ICUAS)*,851–860 (2015).

[81] Y. Liao and F. Berrelli, "An Adaptive Approach to Real-Time Estimation of Vehicle Sideslip, Road Bank Angles and Sensor Bias," *IEEE Transactions on Vehicular Technology* **68**(8), 7443–7454 (2019).

[82] A. Shimada, "Robust Localization and Position Control for Mobile Robot using EKF with Disturbance Estimation," *12th France-Japan and 10th Europe-Asia Congress on Mechatronics International Conference on Robotics and Bioinformatics (ROBIO)*,312–315 (2018).

[83] Y.C. Lai and T.Q. Le, "Adaptive Learning Based Observer with Dynamic Inversion for the Autonomous Flight of an Unmanned Helicopter," *IEEE Transactions on Aerospace and Electronic Systems* ,(2021)

[84] S. Chang, "A Deep Learning Approach for Localization Systems of High-Speed Objects," *IEEE Access*, 96521–96530 (2019).

[85] D. Selmanaj, M. Corno, G. Panzani and S.M. Savaresi, "Vehicle Sideslip Estimation : A Kinematic Based Approach," *Control Engineering Practice* **67**, 1–12 (2017).

[86] C. Shang, D. Liu, Q. Yang and H. Fang, "Responsiveness Analysis of Classic and Observer-Based Leader-Follower Systems," *IEEE Transactions on Control of Network Systems*,(2021).

[87] M. Cui, "Observer Based Adaptive Tracking Control of Wheeled Mobile-Robots with Unknown Slipping Parameters," *IEEE Access* **7**, 1696461–169655 (2019).

[88] D.E. Davison, P.T. Kabamba and S.M. Meerkov, "Limitations of Disturbance Rejection in Feedback Systems with Finite Band-

width," *IEEE Transactions on Automatic Control* **44**(6), 11321–1144 (1999).

[89] Z. Zwierzewicz, "Robust and Adaptive Path Following Control of an Underactuated Ship," *IEEE Access* **8**, 1201981–120207 (2020).

[90] F. Pretagostini, L. Ferranti, G. Berardo, V. Ivanov and B. Shyrokau, "Survey on Wheel Slip Control Design Strategies, Evaluation and Application to Antilock Braking Systems," *IEEE Access* **8**, 10951–10970 (2020).

Chapter 2

Smooth Trajectory Planning with Gradient Descent

2.1 Foreword

Artificial Potential Field (APF) method is one of the earliest optimal algorithms for tracking targets. APF is a gradient based approach, where the workspace is conceived as a 'field' or function of pseudo-forces (therefore, the term, 'artificial') generated by the target and the obstacles(s). The idea behind route-finding in this context is this – the target always attracts the tracking agent and the obstacle repulses. The resultant pseudo-force configures a safe direction of motion towards the target. Generation of a 'single' driving force in response to both tracking and collision avoidance is one of the major advantages of APF as a path planner. The mathematics is convincingly simple and it is easy to demonstrate theoretical stability if undesired local minima points can be carefully avoided by redefining the reachable set [3, 1, 8]. However, APF is not practically efficient as a controller if the tracking velocity is modeled as a function of the gradient. This is reason enough for gradient-descent methods being used for computing search directions and not for controlling of the pursuer. This chapter proposes a modified optimal route planner for tracking vehicles using a smooth, fast-converging variant of APF. The proposed method is applicable in situations characterized by

highly constrained reachable sets, such as corridors, aisles and the likes, wherein traditional APF generates sluggish and oscillatory response. Relevant prior-art other than a few articles such as [2] are hard to find.

2.2 Optimal Problem Formulation

The potential surface is a scalar field with peaks and troughs. Magnitude of the potential at any point on the surface indicates the depth (or height) of the undulations. The obstacles are assumed as points of local maxima and the target as a global minima on the potential surface. The objective is to formulate a non-trivial unique route starting from any point, $p(0)$, $(p(t) \in \mathbb{R}^2 \setminus \mathbb{S}$, on the potential surface where, $\mathbb{S}$ is the unreachable set comprising the positions of the local maxima. The point $p(0)$ marks the location of the pursuer at time, $t = 0$. The optimal trajectory, $p(t)$ is intended to terminate at some $t = T$, $(T \in \mathbb{R}^+)$ which marks interception (theoretically) of the pursuer with the target. The discrete-time optimal problem can be mathematically described by equation (2.1), where sampling instant, $k = 1$ corresponds to $t = 0$ and interception time is related to sampling interval $\delta(t)$ as $(N-1)\delta t = T$ $(k, N \in \mathbb{Z}^+, k \leq N)$.

$$
\begin{aligned}
P_{optimal}: \min_{q_k} \sum_{k=1}^{N} U_k, \\
\text{such that, } U_k =& U_k^{attract} + U_k^{repulse}, \\
\text{where, } U_k^{attract} :=& \frac{1}{2}\xi\rho^2(q_k, q_{target}), \\
\text{and, } U_k^{repulse} :=& \frac{1}{2}\eta\left(\frac{1}{\rho(q_k, q_{obstacle})} - \frac{1}{\rho_0}\right)^2 \mid \rho(q_k, q_{obstacle}) \leq \rho_0
\end{aligned} \tag{2.1}
$$

At any instant k, the resultant potential function is described by U_k, where, the attractive potential field exerted by the target is $U_k^{attract}$ and the repulsive potential field exerted by the obstacles is given by $U_k^{repulse}$. Here, ξ and η are positive scalar numbers representing relative strengths of the attractive and repulsive forces. The term ρ_0 is a pre-defined threshold factor called the 'radius of

influence', which means if an obstacle is identified within a ρ_0-neighbourhood of the pursuer at any k, then it needs to be avoided. The path function between the obstacle and the pursuer is given by $\rho(q_k, q_{obstacle})$, wherein it can be a quadratic function pertaining to Euclidean distance or a 1-norm, ∞-norm as defined by the designer. Similarly, the path function between the target and the pursuer at k^{th} instant can be defined by $\rho(q_k, q_{target})$. Eventually, the effective potential U_k at k^{th} instant is to be minimized. U_k is a positive semidefinite function which achieves a global minima at interception, where the minimum value of the potential evaluates to zero.

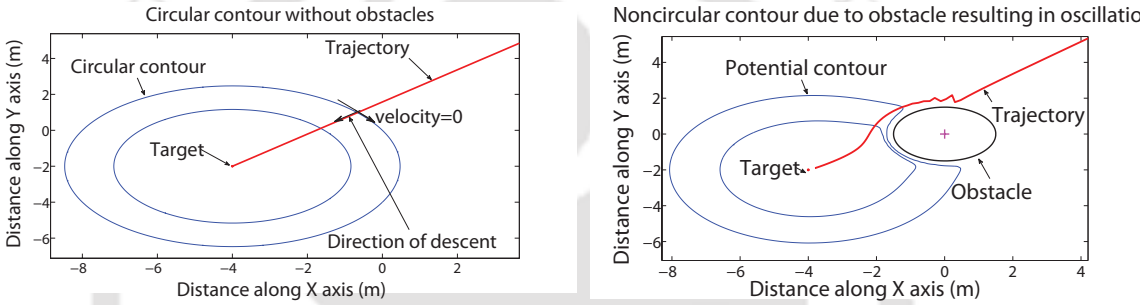

Figure 2.1: Smooth contour plot without obstacles vs. irregular potential contour due to obstacle

2.2.1 Linear Search Direction

Traditionally, optimal route finding with APF could be treated as a 'line search' problem, where pose q of the pursuer could be updated linearly according to (2.2), with the assumption of differential flatness.

$$q_{k+1} = q_k - \alpha_k d_k, \tag{2.2}$$

Search direction, $d_k = -\nabla U(k)$ is a vector that points along the 'steepest' descent from one level set of the potential surface to another and represented by a negative gradient of the resultant potential function, $U_k, \forall k \in \{1, 2, 3, ..., N\}$. The parameter, $\alpha \in \mathbb{R}^+$ in (2.2) is a positive constant playing the role of a fixed

step size. If there is only an isolated global minima (the target) and no obstacles, the potential surface appears flat with a point trough and the level curves are represented by concentric circles.[1] Figure 2.1(a) confirms that magnitude of the potential linearly reduces to zero as the search direction evolves along the negative orthogonal direction with respect to the level curves. With obstacles however, the level curves become irregular in shape and the search direction is no longer a straight line. Guided by the irregularity in potential terrain the pursuer's trajectory turns out oscillatory in nature, which can be observed in Figure 2.1(b).

2.3 Smoothness, Convergence and Solution

2.3.1 Problems with Line Search

Convergence along the linear search direction can be expedited by introducing a momentum term m, such that, the position update model is modified to equation (2.3):

$$q_{k+1} = q_k - \alpha_k d_k + m(q_k - q_{k-1}) \tag{2.3}$$

Here, $m \in \mathbb{R}^+$ is a weight to the gradient vector corresponding to the previous sampling instant. Note that, (2.3) is still linear with an added momentum to the previous time stamp. In a sparsely populated workspace the gradient becomes flat due to lack of features and gradient descent even with a momentum term tends to generate sluggish response. On the other hand, an exactly opposite scenario comprising a dense population of obstacles may give rise to several local minima traps. Avoiding undesired local minimas (not target) requires additional control and computational effort and a possible loss in continuity of state and control trajectories. Also, this might offer difficulties in selection of an optiml step size. Intuitively, a fixed step size of large magnitude that may help to speed up convergence in flat gradients may fail to converge at all along a steep gradient, which leads two possible solutions. We can use – 1. a variable step size, α,

[1] Level curves are the invariant sets where the potential evaluates to a constant.

that can be computed as function of time using the *Wolfe*[2] conditions and 2. a computationally intensive routine to calculate optimal value for α suited to both situations. A third possible solution, as already suggested in [2] is to leverage second order optimization with a fixed step size.

2.3.2 Curvature of Potential Surface

Curvature of the potential surface can be incorporated in the process of gradient computation. This results in a quadratic search direction shown in equation (2.4):

$$d_k = -H_k^{-1} \nabla U_k, \tag{2.4}$$

Here, $H_k = \nabla^2 U_k$ represents Hessian[3] of the artificial potential. Equation (2.4) represents the Newton's method (NM) [4] and the search direction is termed as Newton's direction. Considering a unit step-size, the pursuer's trajectory following Newton's direction intercepts the target with quadratic rate of convergence if and only if the Hessian matrix is well conditioned ($H_k \succ 0$). Second-order information contained in the Hessian is applicable only if the potential field is 'strongly convex'[4].

For solutions starting far from the target, non-singularity feature of the Hessian's inverse cannot be guaranteed. Also, to save computational expense true Hessian is usually never computed. This led to replacing the Hessian by an approximation of the Hessian by slight perturbation, but nevertheless preserving the second order characteristics of the Hessian. The resulting matrix is designed to be positive definite, even if the true Hessian turns out to be sign indefinite. As an alternative measure, Levenberg [5] proposed a descent direction given in equation (2.5),

[2]The measure of sufficient-decrease and the curvature condition together are termed as the Wolfe conditions, which are governed by the relations, $U_{k+1} \leq U_k + c_1 \alpha_k \nabla U_k^\top d_k$, $|\nabla U_{k+1}^\top d_k| \leq c_2 |\nabla U_k^\top d_k|$, where, $0 < c_1 < c_2 < 1$ and, $c_1, c_2 \in \mathbb{R}$.

[3]Hessian is a square matrix of second ordered partial derivatives of potential function

[4]A sufficient but not necessary condition for the Hessian to be invertible is: there exists a $c \in \mathbb{R}$, such that $U_k(q_k) - c\|q_k\|^2$ is convex for all $q_k \in \mathbb{R}^2$. Under this assumption, all eigenvalues of the Hessian are greater than $2c$.

where, $\mu \in \mathbb{R}^+$ is an adaptive parameter, often known as the 'weighing factor'.

$$d_k = (\mu\ I + H_k)^{-1} \nabla U_k, \tag{2.5}$$

The weighing factor (also called the damping factor) assigns adaptive weightage to the quadratic search direction in comparison to the line search direction. A numerically small value of μ helps to accentuate the Hessian at the k^{th} instant and speeds up descent, whereas, a numerically larger value of μ aligns the path along the linear search direction if quadratic approximation fails to obtain significant reduction in magnitude of the potential function in the current iteration. Levenberg's modification may fail to preserve the curvature along the eigenvectors of the true Hessian. Thereby, Levenberg's algorithm can be further modified by introducing eigenvalues of the Hessian, H_k instead of the identity elements. The eigenvectors corresponding to the eigenvalues pertain to asymptotes to the curvature of the potential surface. This algorithm preserves the curvature along one or more preferred eigenvector directions and is called the Marquardt [6] variant. The Marquardt modification is shown in equation (2.6).

$$d_k = (\mu\ diag(H_k) + H_k)^{-1} \nabla U_k, \tag{2.6}$$

The Identity matrix of appropriate dimensions in equation (2.5) has been replaced by a diagonal matrix composed of the eigenvalues of H_k. A similar role is played by μ in assigning relative weights to linear and quadratic search directions and the resultant matrix is designed to be positive definite. Levenberg-Marquardt algorithm (LMA) achieves the fastest convergence rate but this is achieved at the cost of additional computational burden. The enhanced computational complexity is attributed to calculation of the Hessian inverse.

2.4 Comparative Study

Efficiency of the second order algorithms discussed above can be parameterized by factors such as smoothness, task completion time, computational complexity, stability and speed of convergence. The scope of the present investigation pertains to

narrow spaces having restricted reachable set, which present a suitable landscape for evaluation of the efficiency metrics. We already know that quadratic route finding algorithms have a computational complexity of $O(\epsilon^2)$, ϵ being the size of the input space. Therefore, in the following experiments we focus on determining smoothness, speed and completion time assuming a fixed sampling rate. A low sampling interval (high sampling frequency) is the typical choice to avoid jitter. Safety and passenger comfort being major concerns for tracking vehicles, the desired optimal trajectory is intended to be oscillation-free. According to Ren et al. [2], a predefined 'ideal' number of iterations can be assumed as a suitable metric for evaluating task completion efficiency. This means if convergence to a desired accuracy is not achieved within a pre-defined time limit (or number of iterations, as sampling iterval is fixed), then the task is considered to be failed. Additionally, while traversing populated narrow valleys, gradient descent algorithms may fail because of local minima traps. To accommodate both conditions, we consider two criteria –1. pursuer reaches a predetermined neighbourhood of the target and 2. does so in a certain number of iterations, which is less than or equal to a predefined threshold. Under these assumptions, performance of first and second order search routines have been compared for both sparsely and densely populated environments.

2.4.1 Case Study 1: Sparse Environment

The first scenario deals with a sparse environment constituting a flat potential surface. We assume a single point of maxima denoted by the position of a static obstacle located at $(-2, 0)$m, and the radius of influence being 2m. Obstacle is assumed to have a convex hull, with radius of obstacle assumed to be 1m. The target is stationary and located at position $(-4, -2)$m. As the pursuer starts tracking the target from a position of $(3, 5.5)$m, the Euclidean distance between the pursuer and the target is observed and the number of iterations required to achieve desired convergence are noted. A final positional error of 0.1m or less is identified as interception (convergence). Figure 2.2(a) shows oscillations in the

pursuer's trajectory around the obstacle position. Figure 2.2(b) confirms that interception wih the target occurs after 98 steps or iterations for an admissible step size of $\alpha = 0.05$s. It has been experimentally observed that a higher magnitude of α tends to generate overshoots and undershoots leading to oscillations in the trajectories, affecting both positon and control variables. Rate of convergence in the

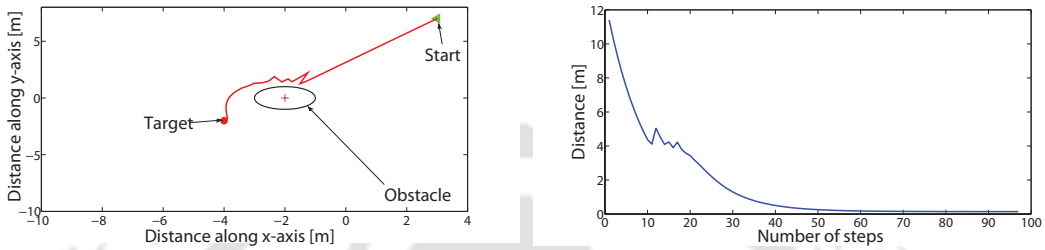

Figure 2.2: Performance of gradient descent with line search in a flat potential surface showing (a) oscillations in trajectory and b) convergence of tracking distance.

preceding experiment can be enhanced by using a momentum factor of $m = 0.5$, wherein, the number of iterations required for convergence reduces to 42 (Figure 2.3). Inspite of around 43% decrement in convergence time, gradient descent

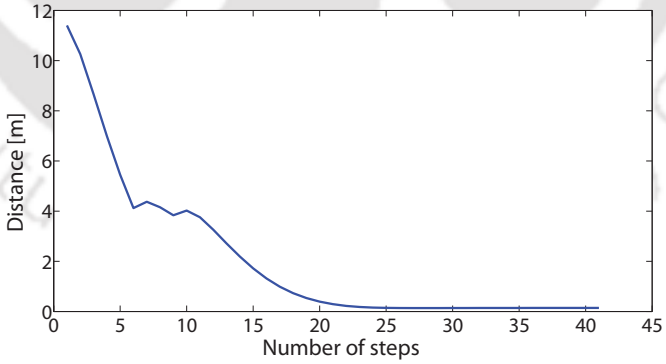

Figure 2.3: Linear gradient descent with momentum term yields faster interception.

with momentum is still a first order approximation of the potential surface. A local quadratic approximation of the potential field has been implemented us-

ing the Levenberg or the modified Newton's method, which achieves convergence within 38 iterations for the same scenario, thereby offering a further improvement of 5%. Figure 2.4(a) shows the trajectory of the pursuer for Levenberg method

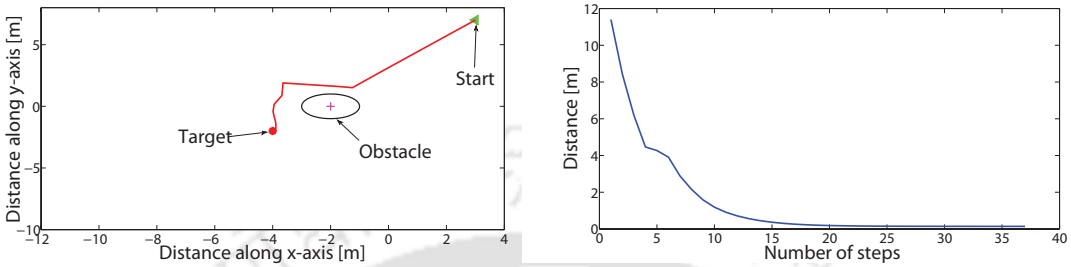

Figure 2.4: Trajectory using Levenberg Method showing (a) reduced oscillations and (b) rate of convergence of distance function between the target and the pursuer.

and correspondingly Figure 2.4(b) illustrates convergence timing diagram for the same. The trajectory is found to be further smoothened out after applying the Marquardt modification (Figure 2.5 (a)) for a comparative interception timing efficiency (34 iterations are required for convergence)(see Figure 2.5(b)). The weighing factor has been assumed as 0.3 and 0.7 for Levenberg and Levenberg-Marquardt algorithms respectively. It has been observed that the latter technique is able to withstand a larger value of μ that preserves the curvature information of the potential surface. On the other hand, $\mu > 0.3$ has been avoided in the Levenberg method as incrementing μ tends to prioritize the linear search direction.

2.4.2 Case Study 2: Populated Environment

The next case study pertains to a densely populated environment in contrast to the sparsely populated flat potential surface featured in the preceding discussion. The potential field is characterized by a target close to a narrow valley surrounded by a cluster of stationary obstacles of radius 1m each, positioned at $(-2, 0)$ m,

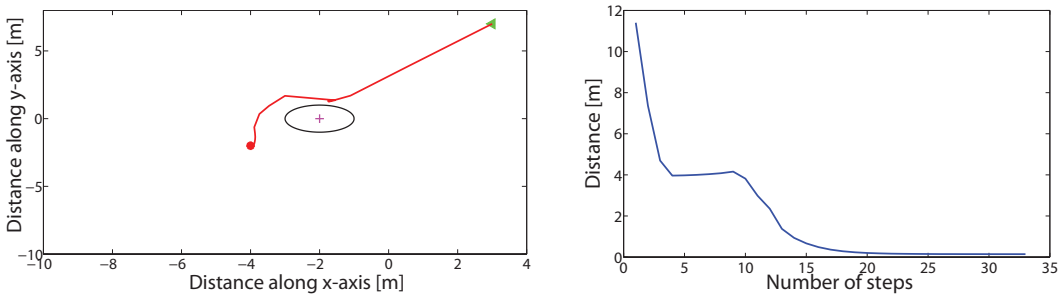

Figure 2.5: Performance evaluation of Levenberg-Marquardt Method in flat potential surface showing (a) smooth trajectory and (b) rate of convergence of distance function between the target and the pursuer.

$(-4, 2)$ m, $(-5, 0)$ m and $(-3, 2)$ m. The target is located at $(-5, -2)$ m and assumed to retain same position during tracking. The idea behind the static setting is to study the tracking performance of the first and the second order gradient based path planners when the pursuer is compelled to traverse through the narrow valley and simultaneously compute subsequent search directions. This means, we purposefully design the problem such that the pursuer does not have any option to bypass the obstacle cluster. The pursuer is assumed to start tracking from the position, $(3, 7)$ m.

Linear gradient descent achieves interception in 764 iterations as shown in Figure 2.6(a). Figure 2.6(b) illustrates the trajectory of the robot for the same navigation planner. Adding momentum term helps to achieve convergence in 65% less time, within 271 iterations. Levenberg method with μ=0.3 achieves convergence in 211 iterations, while Levenberg-Marquardt algorithm yields the same result in only 160 steps. In the following simulations an upper limit of 1000 iterations have been considered as admissible threshold for successful interception. Interception is assumed to be achieved if the distance function between the target and the pursuer is less or equal to 0.1m. Although analysis of results of the previous simulation confirms the faster convergence property of quadratic search routines over linear search, true efficiency of the second order optimization meth-

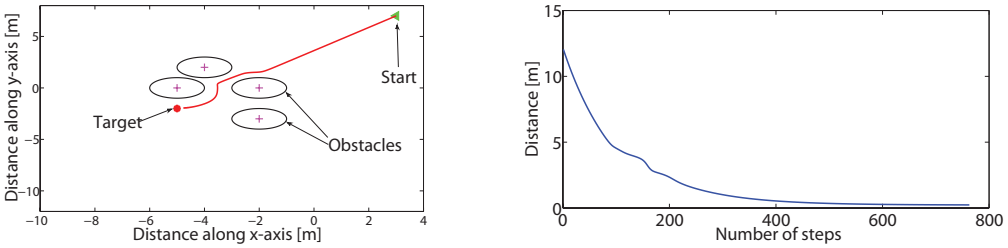

Figure 2.6: Performance evaluation of linear gradient descent in narrow valley (a) trajectory of pursuer (b) rate of convergence of distance function between the target and the pursuer.

ods may be observed in scenarios involving narrower passages. Passage through the obstacle cluster is further narrowed by 23% to a spacing equal to the radius of influence of an obstacle. This means, all four obstacles need to be compulsorily avoided. Under this stringent path constraint, Levenberg-Marquardt algorithm

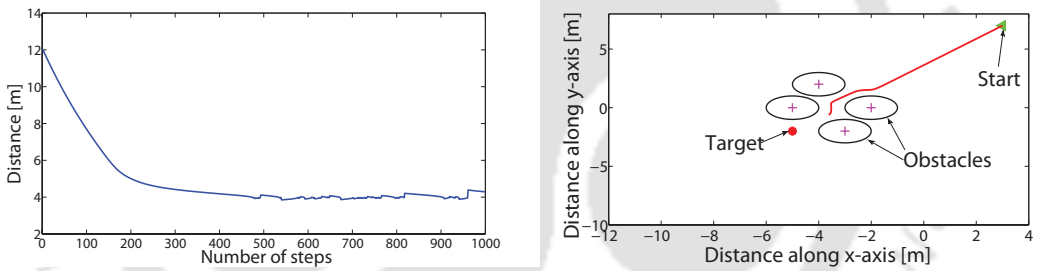

Figure 2.7: Finding optimal solutions with first and second order search directions in narrow valleys (a) trajectory computed with Newton's direction fails to converge within admissible threshold of iterations (b) trajectory fails to converge with line search.

(LMA) is found to successfully complete the task in 187 iterations, for a weighing factor chosen as 0.3. Modified Newton's mehtod (NM) or Levenberg method fails to reach the target within the predefined admissible limit of steps as shown in (Figure 2.7(a)). On the other hand, (Figure 2.7(b)) illustrates the pursuer's trajectory under linear gradient descent. The task remains incomplete as the tracking trajctory fails to satisfy the terminal distance function criterion for vali-

dating inteception. In both of these cases, it is concluded that no solution exists through a very narrow passage for all planners except the Levenberg-Marquardt technique. The performances of the first and the second order path search routines have been summarized in Table 2.1. Succesful execution of tracking by the

Method	Flat Gradient	Narrow Valley
GD	98	764
GDM	42	271
NM	38	211
LMA	34	160

Table 2.1: Comparative study of number of iterations required for convergence by first and secord order search directions in a narrow populated environment. GD: Gradient Descent, GDM: Gradient Descent with momentum, NM: Newton's Method or Levenberg's Method, LMA: Levenberg-Marquardt Algorithm

Levenberg-Marquardt algorithm through the obstacle cluster in both conditions have been illustrated in Figure 2.8(a) and Figure 2.8(b) respectively. However, as

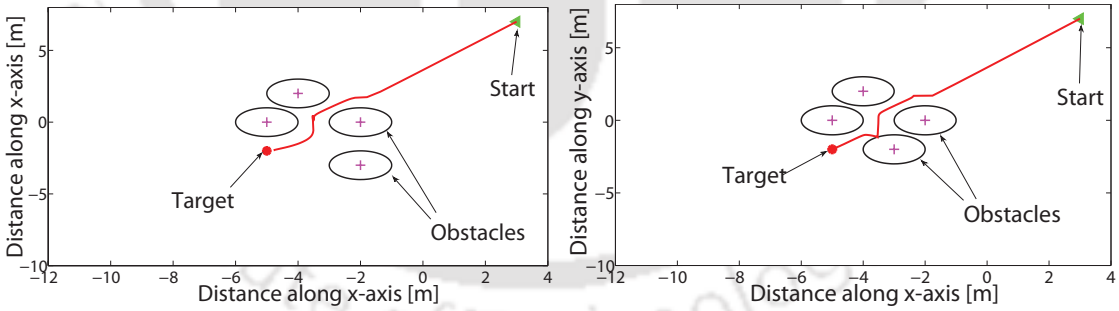

Figure 2.8: Performance evaluation of Levenberg-Marquardt in populated environments (a) trajectory in a narrow valley (b) trajectory when reachable set is further reduced.

pointed out in [4, 7], the second order methods are sub-optimal. Linear gradient descent is globally convergent but slow and inefficient in both sparsely populated and densely populated situations. Whereas, convergence of the quadratic meth-

ods can be guaranteed only if the tracking is assumed to start within a small neighbourhood of the target position (minima). For any arbitrary initial point, this refers to a continuous evaluation of search directions upon applying a change of variables. Intensive computations pertaining to calculation of true Hessian and its inverse in quadratic optimization routines may be bypassed by said approximation. According to Ren et al. [2] this approximation is feasible for systems with two dimensional state space. This further encouraged us to define the kinematics of the pursuer by a two-state, two-input model, to be explained in details in the next chapter.

Gradient based planners are global optimization algorithms that may suffer from possible loss in continuity of trajectory because of local minima traps generated by dynamic distribution of obstacles and target. Recently, a noise-free, reactive navigation technique has been proposed in [9] wherein, the pursuer is driven towards an extremum without computation of state derivatives. But the results indicate useful outcome only for well-behaved targets and the article remains oblivious to movement of obstacles in the environment. Another controller proposed in [10] uses Harmonic potential vector-field that matches a navigation function to the kinetic energy of the vehicle, but does not take into account issues caused by mobility of obstacles. Besides, pursuit vehicles such as guided carts, industrial inspection and surveilance robots may benefit from multiple other parameter optimizations for cost-effective performance and safety requirements, which cannot be achieved by gradient based guidance alone. Hence customizable optimal planners have been developed in Chapter 3. It will be shown in the following chapters that multi-objective optimal controllers are better equipped to planning with relative weightage and generate smoother trajectories for a variety of dynamic situations.

2.5 Summary of Chapter 2

In this chapter, efficiency of second order gradient based optimal path planners have been verified in both sparsely populated and densely populated environments

over traditional line search methods. Especially, for narrow valleys, constituted by obstacle clusters located near the target position, Levenberg-Marquardt search direction has been found to be the best in class in terms of convergence time, smoothness of trajectory and feasibility of solutions. The experiments discussed here can be extended to dynamic tracking scenarios by periodic state feedback, provided other limitations of artificial potential field, such as local minima problem, obstacle close to target near intended interception point and the likes are appropriately addressed.

Bibliography

[1] S.S. Ge and Y.J. Cui, "New Potential Functions for Mobile Robot Path Planning," *IEEE Transactions on Robotics and Automation* **16**(5), 615–621 (2000).

[2] J. Ren, K.A. McIssac and R.V. Patel, "Modified Newton's Method Applied to Potential Field-Based Navigation for Mobile Robots," *IEEE Transactions on Robotics and Automation* **22**(2), 384–391 (2006).

[3] J. Kim and P.K. Khosla, " Real-time obstacle avoidance using harmonic potential functions," *IEEE Transactions on Robotics* **8**(5), 338–349 (1992).

[4] J. Nocedal and S.J. Wright, "Numerical Optimization," *Springer, New York*, (1999).

[5] K. Levenberg, "A method for the solution of certain problems in least squares," *Quarterly of Applied Mathematics* **2**, 164–168 (1944).

[6] D. Marquardt, "An algorithm for least- squares estimation of nonlinear parameters," *SIAM Journal of Applied Mathematics* **11**, 431–441 (1963).

[7] Z. Richard, "Introduction to Machine Learning, Lecture 6: Optimization," *Lectures on course CSC2515, Fall 2007, University of Toronto*, (2007).

[8] Q. Jia and X. Wang, "Path planning for mobile robots based on a modified potential model," *International Conference o Mechatronics and Automation (ICMA)*, 4946–4951 (2009).

[9] A.S. Matveev, M.C. Hoy and A.V. Savkin, "Extremum Seeking Navigation Without Derivative Estimation of a Mobile Robot in a Dynamic Environmental Field," *EEE Transactions on Control Systems Technology* **24**(3), 1084–1091 (2016).

[10] S.G. Loizou and E.D. Rimon, "Mobile Robot Navigation Functions Tuned by Sensor Readings in Partially Known Environments," *IEEE Robotics and Automation Letters* **7**(2), 3803–3810 (2022).

Chapter 3

Optimal Guidance Trajectory: Free vs Fixed Horizon

3.1 Foreword

Tasks involving moving target tracking can be planned either with a free terminal time or with a fixed terminal time. Terminal time refers to a finite real number when the task is designed to conclude. In the present context, tracking terminates with an interception, which refers to a condition wherein the target and the pursuer trajectories occupy the same set of states and/or control variables at a single time instant. If the terminal time is not predetermined, trajectory planning corresponds to a free finite horizon problem. On the other hand, if the pursuer is planned to intercept the moving target at the end of a predetermined time interval after tracking initiates, it refers to a fixed finite horizon problem. While a free finite horizon seems to offer flexibility to the pursuer's motion, it can also be intuitively understood that a fixed finite horizon is often more convenient over the former. This happens due to traceability of kinematic behaviour of the pursuer, energy consumption and other efficiency indicators relevant to the mission. In this chapter we shall discuss both of these approaches from the perspective of optimal trajectory planning.

3.2 Kinematic Assumptions

The pursuer is assumed to be a two wheeled symmetric vehicle exhibiting point motion. Kinematic relations are goverened by the equation (3.1), where, $(x(t), y(t))$ are the states representing planar position variables and $(v(t), \theta(t))$ are the control variables, representing forward velocity and heading angle with respect to the x-axis. It may be noted that equation (3.1) is a sightly modified version of the unicycle model, with two states and two inputs instead of the traditional 3-variable state vector. The reasons behind the choice of this modified unicycle model has been elaborated in Appendix A, and may be referenced in all recitations related to the pursuer's kinematics.

$$\begin{aligned} \dot{x}(t) &= v(t)\cos(\theta(t)) \\ \dot{y}(t) &= v(t)\sin(\theta(t)) \end{aligned} \tag{3.1}$$

A moving target may be defined by the usual kinematic relations of a unicycle described by equation (3.2), where $v_{tar}(t)$ and $\omega_{tar}(t)$ are the input variables pertaining to forward velocity and turn rate respectively. The target states are constituted by the pose variables, $x_{tar}(t)$, $y_{tar}(t)$ and $\theta_{tar}(t)$. We assume, that the states of the target are periodically measurable by suitable sensors and the linear and angular velocities may be derived from the observed states. By default, prior knowledge of commanded target velocities are unavailable to the trajectory planner unless otherwise stated in specific cases. If the horizon is known, an estimate of pose and velocities of the target may be availed at certain 'future instants' of time for the sake of generating boundary values. Extrapolation of motion history may be leveraged in those situations, the details of which has been enlisted in Appendix A.

$$\begin{aligned} \dot{x}_{tar}(t) &= v_{tar}(t)\cos(\theta_{tar}(t)) \\ \dot{y}_{tar}(t) &= v_{tar}(t)\sin(\theta_{tar}(t)) \\ \dot{\theta}_{tar}(t) &= \omega_{tar}(t) \end{aligned} \tag{3.2}$$

3.3 Free Finite-Horizon Problem

When the interception time is not specified prior to the mission, it is assumed that there exists a real number, $T \in [0, \infty)$ such that, the pursuer shall be able to reach (at least, theoretically) the same states as that of the target at time, T. The concurrence of states of the target and the pursuer is termed as 'interception'. Basically, this a Bolza variant [7] of an infinite-time problem, which is supposed to run for infinite amount of time. In practice, an infinite runtime is modified into a routine that iterates over finite time. Free finite horizon based interception of moving targets had been fairly common with missile guidance problems [16], but rarely practiced in mobile robotics due to the involved dynamicity. Nevertheless, a few modern variants of the proportional navigation techniques [10, 17], reactive response [12] and stability driven control [14] operations have been investigated in the last two decades, out of which optimal solutions have been reportedly minority [14, 18, 14]. Even then, most problems were designed under assumptions of fixed terminal state or fixed terminal time, because of which boundary values could be easier to establish. Slightly differing from other approaches, ref. [19] presents a collision-free navigation model which computes area coverage of a surveillance robot and also determines the quality of coverage as an efficiency indicator.

3.3.1 Problem Formulation

In the current discussion, our objective is to design an optimal tracking and interception problem where the interception time and pursuer's state at that time are both unspecified but are connected by motion of the target; the mathematical particulars (states, velocities etc.) of the motion being measurable, but not predefined. A multi-objective functional comprising of a 'time' component, an 'energy' component and a 'tracking' component is formulated, which can be seen

in equation (3.3). The minimization problem can be summarized as:

$$P_{free}: \min_{(v(t),\theta(t))\in\mathcal{U}} J_{free} \quad \text{subject to kinematic constraints (3.1)}$$

$$\text{where, } J_{free}(t) = \int_{t=0}^{T} (w_t + w_v v^2(t) + w_d\Big((x(t) - x_{tar}(t))^2 + (y(t) - y_{tar}(t))^2\Big)dt$$

wherein, $T \in \mathbb{R}^+$ is unknown and interception is characterized by

$$\Psi(x(T), y(T)) = [x(T) - x_{tar}(T) \;\; y(T) - y_{tar}(T)]^\top = [0 \;\; 0]^\top \tag{3.3}$$

Here, $\mathcal{U} \subset \mathbb{R}^+ \times \mathbb{R}$ is the set of control inputs to the pursuer. The design parameters, w_t, w_d, w_d are known constants that assign relative weights to the time, energy and tracking components of the objective function. Initial states of the pursuer at $t = 0$ is given as $x(0) = x_0$, $y(0) = y_0$. At terminal time T, satisfaction of the terminal state function, $\Psi(x(T), y(T))$ indicates convergence of the pursuer's states with the target's trajectory. Practical convergence constitutes a distance function evaluating to a sub-threshold value. The weighted squared velocity term over the domain $[0, T]$ indicates achievement of minimum control effort apart from limiting the pursuer's velocity to finite values even in absence of a well-defined upper bound. Adjustments of the weighing parameters define priorities assigned to i) minimizing time-to-interception, ii) Euclidean distance between the pursuer and the target and iii) the control effort.

3.3.2 Necessary first-order conditions

The optimal conditions can be determined by constructing the Hamiltonian function. The Hamiltonian, is shown in equation (3.4). The kinematic constraints are incorporated in the Hamiltonian function using a non-trivial vector of weights, $\Lambda = [\lambda_0 \;\; \lambda_1 \;\; \lambda_2]^\top$, where $\lambda_i \in \mathbb{R}^+, (i \in \{0, 1, 2\})$ to formulate an augmented Lagrangian. Here, λ_1 and λ_2 are the Lagrange's multipliers (interchangeably known as the adjoint-variables or co-states). The Hamiltonian is a function of states,

co-states and control variables (see equation (3.4)) and is time invariant.

$$\begin{aligned} &H_{free}(x(t), y(t), \lambda_1(t), \lambda_2(t), v(t), \theta(t)) = \\ &- w_t - w_d \left((x(t) - x_{tar}(t))^2 + (y(t) - y_{tar}(t))^2\right) - w_v v^2(t) \\ &+ \lambda_1(t) v(t) \cos(\theta(t)) + \lambda_2(t) v(t) \sin(\theta(t)) \end{aligned} \tag{3.4}$$

As a side note, it may be clarified that $\lambda_0(t) = -1 \ , \forall t \in [0, T]$, is a standard choice that ensures non-triviality of Λ and validates applicability of the necessary optimality conditions [3].

The relation between the control variables and the co-states are computed from the 'Stationarity' condition, that establishes invariance of the Hamiltonian with respect to the control variables. Optimality requires the first variation of H_{free} to reach a local maxima with respect to the control variables, $v(t)$ and $\theta(t)$, which further deduce the relations shown in equation (3.5).

$$\begin{aligned} \frac{\partial H_{free}}{\partial v(t)} &= -2w_v v(t) + \lambda_1(t) \cos(\theta(t)) + \lambda_2(t) \sin(\theta(t)) = 0 \\ \frac{\partial H_{free}}{\partial \theta(t)} &= -\lambda_1(t) v(t) \sin(\theta(t)) + \lambda_2(t) v(t) \cos(\theta(t)) = 0 \end{aligned} \tag{3.5}$$

Assuming that the pursuer's velocity is non-zero for all $t \in [0, T]$ and rearranging the terms in equation (3.5), the control variables can be expressed explicitly in terms of the co-states as shown in equation (3.6). Positive square-root is considered to corroborate with the domain of definition.

$$\begin{aligned} v(t) &= \frac{\sqrt{\lambda_1^2(t) + \lambda_2^2(t)}}{2w_v} \\ \theta(t) &= \tan^{-1}\left(\frac{\lambda_2(t)}{\lambda_1(t)}\right) \end{aligned} \tag{3.6}$$

Now, the partial derivatives of the Hamiltonian can be computed with respect to the states and the co-states and references to control variables can be replaced by functions of the co-states according to equation (3.6). Thereby, the optimal

control problem is converted into a system of differential equations defined below.

$$\begin{aligned}
\dot{x}(t) &= \frac{\partial H}{\partial \lambda_1(t)} = \frac{\lambda_1(t)}{2w_v} \\
\dot{y}(t) &= \frac{\partial H}{\partial \lambda_2(t)} = \frac{\lambda_2(t)}{2w_v} \\
\dot{\lambda_1}(t) &= -\frac{\partial H}{\partial x(t)} = 2w_d(x(t) - x_{tar}(t)) \\
\dot{\lambda_2}(t) &= -\frac{\partial H}{\partial y(t)} = 2w_d(y(t) - y_{tar}(t))
\end{aligned} \tag{3.7}$$

Assuming that the current problem has at least one solution (refer to existence of solutions in Appendix B), the set of simultaneous differential equations given in (3.7) can be solved using the knowledge of boundary values of the states. In this case, analytical solutions for states and co-states may be computed in the form described in equation (3.8). The closed form solutions may be generated following standard procedures of computing the complementary functions and particular integral. However, one may require to ignore the higher order derivatives of the target trajectory appearing in (3.8) as a restrictive measure for addressing the free-horizon issue. Evaluation of the constants, $\{C_i \mid i \in \{1,2,3,4\}\}$ from boundary conditions have been described later in this chapter. Correctness of the analytic solutions may be verified with any numerical solver.

$$\begin{aligned}
x(t) =& C_1 e^{\sqrt{\frac{w_d}{w_v}}t} + C_2 e^{-\sqrt{\frac{w_d}{w_v}}t} + x_{tar}(t) - \left(\frac{w_v}{w_d}\right)\ddot{x}_{tar}(t) \\
y(t) =& C_3 e^{\sqrt{\frac{w_d}{w_v}}t} + C_4 e^{-\sqrt{\frac{w_d}{w_v}}t} + y_{tar}(t) - \left(\frac{w_v}{w_d}\right)\ddot{y}_{tar}(t) \\
\lambda_1(t) =& 2\sqrt{w_v w_d} C_1 e^{\sqrt{\frac{w_d}{w_v}}t} - 2\sqrt{w_v w_d} C_2 e^{-\sqrt{\frac{w_d}{w_v}}t} + 2w_v \dot{x}_{tar}(t) - \left(\frac{2w_v^2}{w_d}\right)\dddot{x}_{tar}(t) \\
\lambda_2(t) =& 2\sqrt{w_v w_d} C_3 e^{\sqrt{\frac{w_d}{w_v}}t} - 2\sqrt{w_v w_d} C_4 e^{-\sqrt{\frac{w_d}{w_v}}t} + 2w_v \dot{x}_{tar}(t) - \left(\frac{2w_v^2}{w_d}\right)\dddot{y}_{tar}(t)
\end{aligned} \tag{3.8}$$

3.3.3 Determination of Boundary Values

A careful observation reveals that only the initial pair of states are known, since the terminal time is itself an unknown parameter. However, the 'Transversality'

condition is capable to provide two additional terminal boundary values of co-states at T. The usage of mixed variable boundary conditions renders the name – *two-point split boundary value problem.*

A generalized expression for transversality is given by equation (3.9), where, Φ represents final cost function, Ψ represents final state function, K is an unknown constant vector, H_{free} is the Hamiltonian and $\tilde{\Lambda}(t)$ is the vector of Lagrange's multipliers. State vector is represented by $q(t)$ and all parameters of equation (3.9) must be evaluated at T.

$$\begin{aligned} &\left(\frac{\partial \Phi}{\partial q} + \left(\frac{\partial \Psi}{\partial q}\right)^{\top} K - \tilde{\Lambda}\right)^{\top} \Big|_{T} dq(T) + \\ &\left(\frac{\partial \Phi}{\partial t} + \left(\frac{\partial \Psi}{\partial t}\right)^{\top} K + H\right)^{\top} \Big|_{T} dT = 0 \end{aligned} \tag{3.9}$$

Since a final cost $\Phi(T)$ is not defined in this case, it is required to find a terminal time T (if it exists) such that $\Psi(T)$ equates to zero at T. The vector $\tilde{\Lambda}(T)$ depends only on the Hamiltonian evaluated at T. The state vector, $q(T)$ and terminal time, T are both 'free' in the sense that they are not constrained by any prior definition of time or trajectories. But $q(T)$ and T are not entirely independent of each other. Interception constrains the terminal states of the pursuer to be a subset of the target trajectory at terminal time, T. Suppose, $P = (x_{tar}(T), y_{tar}(T))$ is a moving point representing target position at T, then the velocities at T can be expressed by the equation (3.10).

$$dq(T) = \frac{dP(T)}{dT} dT = \begin{bmatrix} v_{tar}(T)\cos(\theta_{tar}(T)) \\ v_{tar}(T)\sin(\theta_{tar}(T)) \end{bmatrix} dT \tag{3.10}$$

The transversality condition of (3.9) can now be re-written as (3.11).

$$\begin{bmatrix} \lambda_1(T) & \lambda_2(T) \end{bmatrix} \begin{bmatrix} v_{tar}(T)\cos(\theta_{tar}(T))dT \\ v_{tar}(T)\sin(\theta_{tar}(T))dT \end{bmatrix} = H_{free}(T)dT = 0 \tag{3.11}$$

Terminal time being free, small variations about T are allowed; hence $dT \neq 0$. Also, the Hamiltonian must evaluate to zero for every t along the optimal trajectory as, Hamiltonian is not an explicit function of time [11]. This substantiates

equation (3.12). Also note that at T, the distance function between the target and the pursuer is by theory, zero.

$$\begin{aligned} H_{free}(T) &= \lambda_1(T)v(T)\cos(\theta(T)) + \lambda_2(T)v(T)\sin(\theta(T)) \\ &\quad - w_d((x(T) - x_{tar}(T))^2 + (y(T) - y_{tar}(T))^2) \\ &\quad - w_t - w_v v^2(T) = 0 \\ \Rightarrow &- w_t + \frac{(\lambda_1^2(T) + \lambda_2^2(T))}{4w_v} = 0 \end{aligned} \tag{3.12}$$

From the two relations, (3.11) and (3.12), the terminal co-states $\lambda_1(T)$ and $\lambda_2(T)$ can be computed in terms of $\theta_{tar}(T)$. We consider, the terminal velocity and heading of the target are predictable, irrespective of the value of T. Note that, this assumption does not make any explicit references to terminal positions, but the terminal rate of motion of the target. It is important to observe that the terminal target velocity is assumed non-zero. As mentioned before, this is a special case where conditions on terminal velocity and turn rate of the target need to be estimable in order to find a suitable time-to-interception. Clearly, this reservation restricts usage to very few situations and testifies against the viability of free horizon formulation in practice. By way of example, one such case may be designed for a leader-follower system. The follower may be enabled to reconstruct the velocity and heading of the leader from past observations and extrapolate the same to a range of future times. The terminal velocities of the leader may be roughly assigned an average value obtained from the said extrapolation. This procedure may be repeated through multiple iterations. An approximate programming is unlikely to be perfect but may prove practical so far as free-horizon is a mandate. Using the transversality conditions, the terminal co-states can now be expressed as in equation (3.13).

$$\begin{aligned} \lambda_2(T) &= \pm\sqrt{\frac{4w_t w_v}{\sec^2(\theta_{tar}(T))}} \\ \lambda_1(T) &= -\tan(\theta_{tar}(T))\lambda_2(T) \end{aligned} \tag{3.13}$$

3.3.4 Determination of Free Terminal Time

At this point it is now possible to compute an interception time. However, there can still be unfavorable outcomes, if a finite real solution for T does not exist or multiple solutions exist. The latter can be resolved by selecting the smallest positive real number from the set of admissible interception time. Considering the fact that interception drives the Euclidean distance between the target and the pursuer to zero at T, we can write the following equation (3.14).

$$\begin{aligned}&(x(T) - x_{tar}(T))^2 + (y(T) - y_{tar}(T))^2 = 0 \\ &\Rightarrow C_1^2 e^{2\sqrt{\frac{w_d}{w_v}}T} + C_2^2 e^{-2\sqrt{\frac{w_d}{w_v}}T} + C_3^2 e^{2\sqrt{\frac{w_d}{w_v}}T} + C_4^2 e^{-2\sqrt{\frac{w_d}{w_v}}T} + 2C_1C_2 + 2C_3C_4 = 0\end{aligned} \tag{3.14}$$

Since, $\Psi(x(T), y(T))$ is zero vector, it implies individually $C_1 e^{\sqrt{\frac{w_d}{w_v}}T} + C_2 e^{-\sqrt{\frac{w_d}{w_v}}T} = 0$ and $C_3 e^{\sqrt{\frac{w_d}{w_v}}T} + C_4 e^{-\sqrt{\frac{w_d}{w_v}}T} = 0$. Ignoring the higher order derivatives in equation (3.8), the relation (3.14) can be further simplified into equation (3.15). We shall soon see that the constants, $C_i (i \in \{1, 2, 3, 4\})$ are functions of T and boundary values. Therefore, assuming the knowledge of boundary values are correct, solution of (3.15) may yield a terminal time for interception.

$$C_1^2 e^{2\sqrt{\frac{w_d}{w_v}}T} + C_3^2 e^{2\sqrt{\frac{w_d}{w_v}}T} + C_1C_2 + C_3C_4 = 0 \tag{3.15}$$

3.3.5 Determination of The Constants

Five unknown factors originally appeared along with the constants, $C_i (i \in \{1, 2, 3, 4\})$ in the optimal solutions demonstrated by equation (3.8). Those factors were the terminal co-states $\lambda_1(T)$ and $\lambda_2(T)$, terminal time T and the terminal velocity and heading of the target, $v_{tar}(T)$ and $\theta_{tar}(T)$. The total number of unknowns exceeded the number of available equations and this required the last two factors to be approximated with sufficient accuracy. Eventually, it led to an approximation of the terminal co-states. Thereby, we arrive at the following relation (3.16)

where, $\lambda_1(T)$ and $\lambda_2(T)$ are imprecisely known.

$$\begin{aligned}\lambda_1(T) =& 2\sqrt{w_v w_d}C_1 e^{\sqrt{\frac{w_d}{w_v}}T} - 2\sqrt{w_v w_d}C_2 e^{-\sqrt{\frac{w_d}{w_v}}T} + 2w_v v_{tar}(T)\cos(\theta_{tar}(T)) \\ \lambda_2(T) =& 2\sqrt{w_v w_d}C_3 e^{\sqrt{\frac{w_d}{w_v}}T} - 2\sqrt{w_v w_d}C_4 e^{-\sqrt{\frac{w_d}{w_v}}T} + 2w_v v_{tar}(T)\sin(\theta_{tar}(T))\end{aligned} \tag{3.16}$$

Again, at $t = T$, $\Psi(T) = 0$ indicates $x(T) = x_{tar}(T)$ and $y(T) = y_{tar}(T)$, which further leads to equation (3.17):

$$\begin{aligned}C_1 e^{\sqrt{\frac{w_d}{w_v}}T} + C_2 e^{-\sqrt{\frac{w_d}{w_v}}T} = 0 \\ C_3 e^{\sqrt{\frac{w_d}{w_v}}T} + C_4 e^{-\sqrt{\frac{w_d}{w_v}}T} = 0\end{aligned} \tag{3.17}$$

Additonally, the measured initial states of the pursuer and the target at $t = 0$ can be replaced in equation (3.8) to yield equation (3.18).

$$\begin{aligned}x_0 = C_1 + C_2 + x_{tar}(0) \\ y_0 = C_3 + C_4 + y_{tar}(0)\end{aligned} \tag{3.18}$$

Rearranging and adding equations (3.16) and (3.17) and thereby utilizing equation (3.18) we can evaluate the constants, $C_i (i \in \{1, 2, 3, 4\})$ as functions of T. The equations (3.19) may be replaced in (3.15); first, to evaluate T, followed by computation of the constants:

$$\begin{aligned}C_1 &= \frac{\lambda_1(T) - 2w_v v_{tar}(T)\cos(\theta_{tar}(T))}{4\sqrt{w_v w_d}e^{\sqrt{\frac{w_d}{w_v}}T}} \\ C_2 &= x_0 - x_{tar}(0) - C_1 \\ C_3 &= \frac{\lambda_2(T) - 2w_v v_{tar}(T)\sin(\theta_{tar}(T))}{4\sqrt{w_v w_d}e^{\sqrt{\frac{w_d}{w_v}}T}} \\ C_4 &= y_0 - y_{tar}(0) - C_3\end{aligned} \tag{3.19}$$

3.4 Observations and Analysis

3.4.1 Role of Weighing Parameters

Assignment of numerical values to the weighing parameters has been investigated by simulation studies. We assumed a target moving along a predefined heading

of $-\pi/3$ radians with respect to the x-axis stating from a position of $(7, 7)$m and with a constant velocity of 0.3m/s. The pursuer initiates tracking from a position of $(4, 4)$m. This experiment was aimed at varying each of the parameters, w_t, w_d and w_v one at a time, while maintaining the other two at constant value of 1.

Table 3.1 gives a guideline for selection of w_t, the weight to minimizing time-to-intercept. As w_t is increased, interception is achieved earlier, but in lieu of a higher incurred cost. Unconstrained velocity and heading profiles of the pursuer corresponding to the set of values of w_t are also demonstrated in Figure 3.1. With reference to computation of boundary values, note that in Figure 3.1, for different values of w_t, the terminal target accelerations and angular velocities have been assumed same. It is observed that, a smaller numerical value of w_t results in lower terminal velocity of the pursuer but yields a larger final tracking error (see Table 3.1). While, a lower terminal velocity is an advantage for designing practical markers for interception, the tracking performance is equally important. A trade-off can thereby be achieved by choosing $w_t = 1.2$ corresponding to a computed terminal time of $T = 2.0285$s, which is characterized by a total cost of 20.7717 and a close to ideal final tracking error at T.

Now, referring to Table 3.2 we can see that, increasing w_d speeds up the process

w_t	T(s)	Total Cost	% Final tracking error
0.5	2.4006	19.2054	0.29
1.0	2.1096	20.3474	0.02
1.5	1.9276	21.3771	0.02
2.0	1.7963	22.3250	0.14
2.5	1.6941	23.2185	0.33

Table 3.1: Computation of free terminal time, total cost and final tracking error for variation of w_t.

Total cost=$\int_{t=0}^{T}(w_t + w_v v^2(t) + w_d\Big((x(t) - x_{tar}(t))^2 + (y(t) - y_{tar}(t))^2\Big)dt$

of minimizing the Euclidean distance between the target and the mobile robot. In a way this results in achieving faster interception. However, this advantage

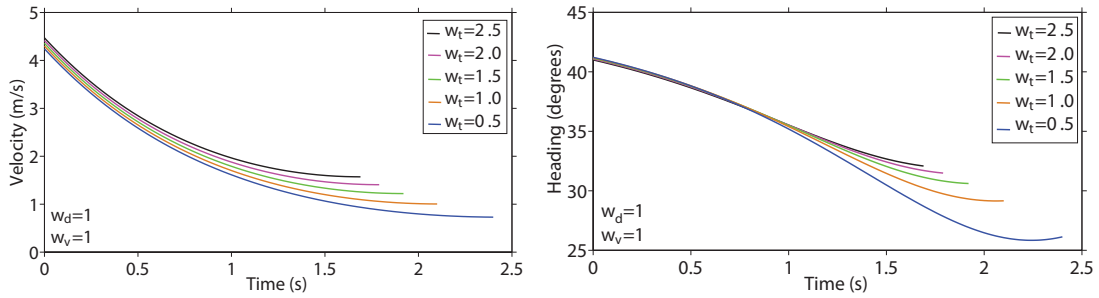

Figure 3.1: Velocity and heading profiles of pursuer and corresponding different values of computed free terminal-time, T.

can be obtained in exchange of an increased total cost. Tracking error remains unaffected by variations of w_d. Figure 3.2 shows, the final velocity of the pursuer

w_d	T(s)	Total Cost	% Final tracking error
0.5	2.5134	15.6467	0.05
1.0	2.1096	20.3474	0.02
1.5	1.8841	24.1532	0.02
2.0	1.7316	27.4352	0.01
2.5	1.6184	30.3537	0.01

Table 3.2: Computation of free terminal time, total cost and final tracking error for variation of w_d.

Total cost=$\int_{t=0}^{T}(w_t + w_v v^2(t) + w_d\Big((x(t) - x_{tar}(t))^2 + (y(t) - y_{tar}(t))^2\Big)dt$

at T is unaffected by changes in the tracking weight w_d, but the initial control effort rapidly increases with an increase in w_d. As a rule of thumb, it is preferred to assign w_d, a numerical value on the lower end of the admissible spectrum, an exemplary numerical value being chosen as 0.8 that yoields the time-to intercept at $T = 2.1479$s and a total cost of 19.0627. Computed value of time-to-intercept increases with an increase in w_v as does the total cost shown in Table 3.3. Initial values assumed by the pursuer's velocities are found to decrease in Figure 3.2(b). Table 3.3 confirms, a rise in the weight to control effort is met with a rapid increase in tracking error at T. Simulations have registered a minimum tracking

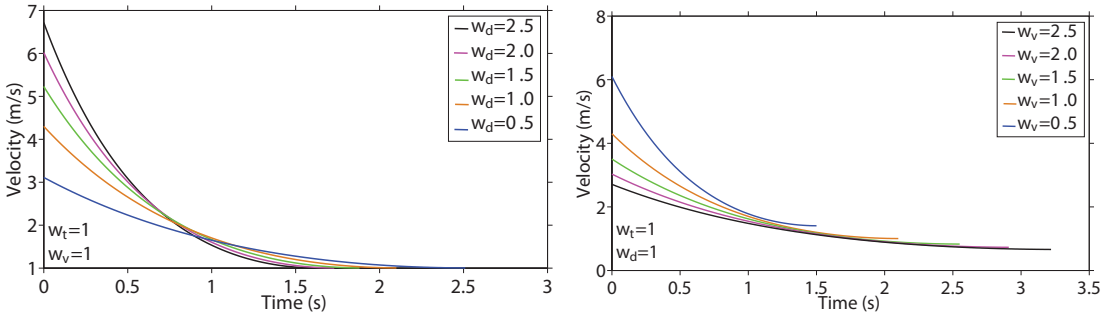

Figure 3.2: Free time-to-intercept vs velocity for different w_v.

error of 4.9 mm for the current example where $w_v = 0.9$ yields a time-to-intercept of 2.0412s and total cost of 18.1261 under the time and tracking weights, w_t and w_d selected as 1.2 and 0.8 respectively, from the preceeding discussions.

w_v	T(s)	Total Cost	% Final tracking error
0.5	1.5056	14.5312	0.07
1.0	2.1096	20.3474	0.02
1.5	2.5554	24.7858	0.23
2.0	2.9163	28.4941	0.56
2.5	3.2216	31.7554	0.99

Table 3.3: Computation of free terminal time, total cost and final tracking error for variation of w_v

Total cost=$\int_{t=0}^{T}(w_t + w_v v^2(t) + w_d\Big((x(t) - x_{tar}(t))^2 + (y(t) - y_{tar}(t))^2\Big)\Big)dt$

3.4.2 Performance Relative to Non-optimal Techniques

Performance of the proposed free horizon optimal tracking controller has been compared with some of the widely relied non-optimal state-of-the-art methods like Lypunov based control [14], Line of sight control [13] and velocity pursuit algorithm [12]. Each of the reference methods is representative of a different class of controller. Figure 3.3 illustrates that Lyapunov based controller [14], which un-

dertakes a stability oriented control strategy, deprioritizes tracking performance and requires a higher control effort (Figure 3.3(b)) despite a low terminal velocity. The line-of-sight method [13], is a reactive technique which iteratively minimizes the line of sight distance between the target and the pursuer. Quite predictably, this method takes the shortest route to interception and drives the pursuer with a pre-assigned constant velocity. Eventually, the average control effort shown in Table 3.4 is higher than our proposed method, although the tracking performance is better than Lyapunov based controller. The velocity pursuit algorithm [12] makes maximum utilization of resources by assigning the maximum allowable velocity to the pursuer. The magnitude and rate of change of velocities at the boundaries make this strategy practically unreliable. In contrast, our proposed optimal method demonstrates a monotonically exponentially decaying velocity profile until interception. Statistics demonstrating final tracking error (see Table 3.4) achieved by the different methods discussed here confirms the efficacy of our proposed optimal technique.

Parameters	LOS	VP	$LYAP$	OPT
Final Tracking Error (mm)	7.7	18.8	18.4	4.49
Control Effort (m^2/s)	5.5	1.6	1.153	0.992

Table 3.4: Final tracking error and control effort by different methods for comparable T.

LOS: Line-of-sight, VP: Velocity Pursuit, $LYAP$: Lyapunov, OPT: Proposed Optimal Method.

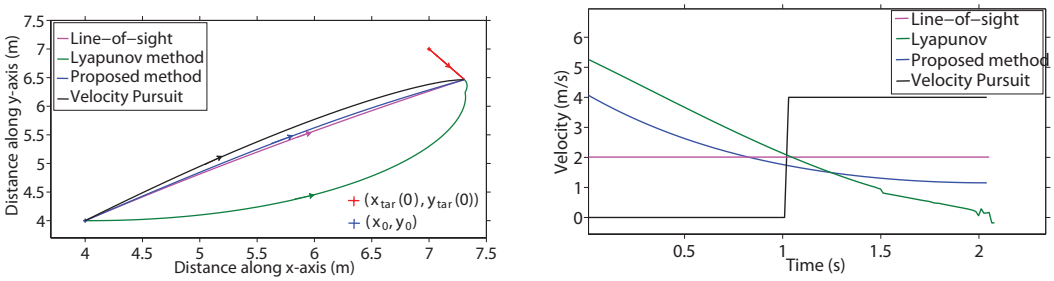

Figure 3.3: Comparison of trajectories and velocities of the pursuer by different methods.

3.5 Tracking with Fixed Terminal Time

Based on the study on weight selection in the preceding discussions, an optimal problem was designed with the parameters, $w_t = 1.2$, $w_d = 0.8$ and $w_v = 0.9$, for which, the free interception time was computed as 2.014s. A fixed horizon optimal control (discussions following) was designed for the same problem and it was observed that a marginally lower cost supported by a lower initial velocity and proportionately a slightly lower control effort was achieved by the formerly described free finite-horizon approach. Comparison of velocity and heading profiles of the pursuer under both tracking approaches have been demonstrated in Figure 3.4. However, the primary assumption that a finite terminal time invariably exists cannot be guaranteed. Moreover, the boundary value problem becomes difficult to be solved unless some assumptions pertaining to target motion is considered. These assumptions make the practical appeal of the optimal control approach rather limited in scope. This is why, a fixed finite horizon optimal tracking control has been studied and discussed in the following sections. Unlike reactive control [12, 7, 8] and learning methods [9, 2, 6], which basically implement intuitive actions based on local observations and operate on more or less consistent behaviour of the elements in the immediate environment, the proposed optimal strategy is capable to look beyond the current situation and plan over a longer horizon. On the other hand, traditional gradient based optimal planners [14, 4]

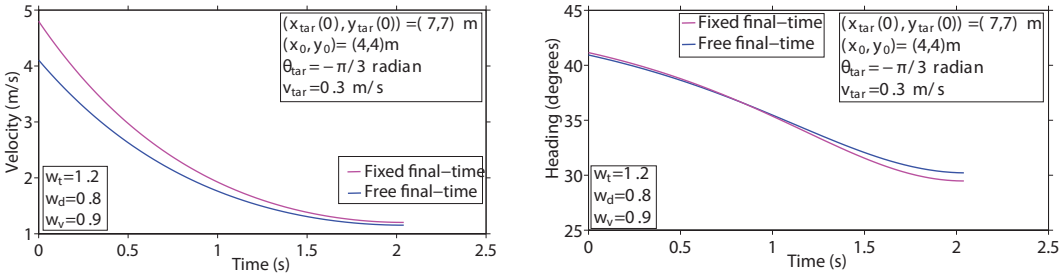

Figure 3.4: Optimal velocity and heading of the pursuer for free and fixed terminal time approaches.

suffer from the possibility of deadlock and loss of continuity in the trajectory. The proposed method not only inculcates a future planning through 'optimality' but is also well suited to handle trade-offs between multiple performance metrics. An important advantage of a fixed finite horizon trajectory planning is the guarantee on existence and uniqueness of solutions, provided the time-to-interception is chosen below a prescribed upper limit. In each iteration, the magnitude of interception time reduces, but the fundamental structure of the controller remains unaltered, which means, if a task is not completed within initially stipulated terminal time (which may happen due to multiple reasons), one can reassign a 'new' interception time in the following iteration.

Mathematical Modeling

The original tracking problem is intended to terminate with an interception at a pre-defined finite horizon, T_{int}. Desired optimal tracking controls can be obtained by minimizing the cost functional J_{fixed} given in (3.20), where $\mathcal{U} \in (\mathbb{R}^+ \times \mathbb{R})$ is the control space such that, forward velocity, $v(t) > 0$ and heading, $\theta(t) \in [0, 2\pi]$ $\forall t \in [0, T_{int}]$. The resulting state trajectories are also optimal. Tracking error is minimized by optimizing the squared norm of the Euclidean distance between the target,$(x_{tar}(t), y_{tar}(t))$ and the pursuer,$(x(t), y(t))$. The energy term

inside the integral denote minimization of velocity and hence the control effort.

$$\begin{aligned} P_{fixed} : \min_{(v(t),\theta(t))\in\mathcal{U}} J_{fixed} \quad &\text{subject to kinematic constraints (3.1)} \\ J_{fixed}(t) = \int_0^{T_{int}} \Big(w_v v^2(t) + w_d((x(t) - x_{tar}(t))^2 + (y(t) - y_{tar}(t))^2) \Big) \mathrm{d}t \end{aligned} \tag{3.20}$$

In equation (3.20), the integration limits 0 and T_{int} represent the tracking duration. The parameters w_v and w_d are positive, real constants and are relative weights to control effort and tracking error. They play a major role in deciding the shape of the tracking path and the total cost incurred. Terminal time being fixed at T_{int}, the objective function in (3.20) does not contain a time component unlike equation (3.3). Velocity being a control variable, energy term pertaining to squared velocity represents control effort.

$$\begin{aligned} \mathcal{H}_{fixed} = &-w_d \left((x(t) - x_{tar}(t))^2 + (y(t) - y_{tar}(t))^2 \right) - w_v v(t)^2 \\ &+ \lambda_1(t) v(t) \cos(\theta(t)) + \lambda_2(t) v(t) \sin(\theta(t)) \end{aligned} \tag{3.21}$$

The Hamiltonian, $\mathcal{H}_{fixed}$ is constructed from (3.1) and (3.20). It is an explicit function of states, $(x(t), y(t))$ and co-states, $(\lambda_1(t), \lambda_2(t))$. The co-states or Lagrange's multipliers are functions of time. They augment the state vector by introducing the kinematic constraints into the optimal problem. First order necessary conditions require $\mathcal{H}_{fixed}$ to be control invariant and to attain a maxima within $[0, T_{int}]$. Mathematically, this means, $\frac{\partial \mathcal{H}_{fixed}}{\partial v} = 0$ and $\frac{\partial \mathcal{H}_{fixed}}{\partial \theta} = 0$. Assuming the robot does not stop during the mission interval, $[0, T_{int}]$, the control variables can be rearranged and then expressed in terms of the costates (see equation (3.22)). Negative definiteness of the Hessian of $\mathcal{H}_{fixed}$ can be easily verified to confirm the status of maxima [3].

$$\left.\begin{aligned} v(t) &= \frac{1}{2w_v} \sqrt{\lambda_1^2(t) + \lambda_2^2(t)} \\ \theta(t) &= \tan^{-1}\left(\frac{\lambda_2(t)}{\lambda_1(t)} \right) \end{aligned}\right\} \tag{3.22}$$

Differentiating $\mathcal{H}_{fixed}$ with respect to the augmented state-co-state vector (the augmented co-state vector is $[x(t), y(t), \lambda_1(t), \lambda_2(t)]^\top$) yields equation (3.23). After rearranging (3.23) into a set of simultaneous linear second order differential

equations, the optimal control problem is thereby converted into a type-I boundary value problem [7].

$$\left.\begin{aligned} \dot{x}(t) &= \frac{\lambda_1(t)}{2w_v} \\ \dot{y}(t) &= \frac{\lambda_2(t)}{2w_v} \\ \dot{\lambda}_1(t) &= 2w_d(x(t) - x_{tar}(t)) \\ \dot{\lambda}_2(t) &= 2w_d(y(t) - y_{tar}(t)) \end{aligned}\right\} \tag{3.23}$$

Measured initial states of the pursuer provide one pair of boundary values. In every planning iteration, the terminal states must be same as that of the target to denote interception. Extrapolating the target kinematics and evaluating the same at T_{int} generates an approximate prediction of the terminal boundary values. An iterative polynomial interpolation method (Neville-Aitken) has been adopted for estimation of terminal states. With each new measurement, the prediction becomes better, even if initiated with a rather arbitrary guess.

A predictive modeling has been chosen over explicit modeling because the latter assumption often fails to accommodate the true dynamic behaviour. Alternative estimation techniques like, numerical methods, regression models and proabilistic approaches can also be used, especially if uncertainties and disturbances [8] are to be taken into account. Among recent studies, long term prediction of horizon for 'intent assessment' of driverless automobiles [16], is believed to be quite a promising technique. Evidently, estimation of target trajectory is actually an open end that can be explored and refined further.

Optimal tracking BVP

The boundary value problem, (3.23) is solvable in closed form and the solution can be expressed as in equation (3.24). Supplementary text in Appendix B elaborates the proof of existence of at least one solution, provided, the second order equivalent of (3.23) is Lipschitz continuous, which is quite an acceptable assump-

tion.

$$\left.\begin{aligned} x(t) &= C_1 e^{\sqrt{\frac{w_d}{w_v}}t} + C_2 e^{-\sqrt{\frac{w_d}{w_v}}t} + x_{tar}(t) - \left(\frac{w_v}{w_d}\right)\ddot{x}_{tar}(t) \\ y(t) &= C_3 e^{\sqrt{\frac{w_d}{w_v}}t} + C_4 e^{-\sqrt{\frac{w_d}{w_v}}t} + y_{tar}(t) - \left(\frac{w_v}{w_d}\right)\ddot{y}_{tar}(t) \\ \lambda_1(t) &= 2\sqrt{w_v w_d} C_1 e^{\sqrt{\frac{w_d}{w_v}}t} - 2\sqrt{w_v w_d} C_2 e^{-\sqrt{\frac{w_d}{w_v}}t} + \hat{w}_x \\ \lambda_2(t) &= 2\sqrt{w_v w_d} C_3 e^{\sqrt{\frac{w_d}{w_v}}t} - 2\sqrt{w_v w_d} C_4 e^{-\sqrt{\frac{w_d}{w_v}}t} + \hat{w}_y \end{aligned}\right\} \tag{3.24}$$

$\hat{w}_x$ and $\hat{w}_y$ in (3.24) refer to $2w_v(\dot{x}_{tar}(t) - ((\frac{w_v}{w_d})\dddot{x}_{tar}(t)))$ and $2w_v(\dot{y}_{tar}(t) - ((\frac{w_v}{w_d})\dddot{x}_{tar}(t)))$. $C_i, i \in \{1,2,3,4\}$ are unknown constants, which can be evaluated according to (3.25) by putting the boundary values in (3.24).

$$\begin{aligned} C_1 &= \frac{x_0(t_1) - (\ddot{x}_{tar}(T_{int})(\frac{w_v}{w_d})e^{\sqrt{\frac{w_d}{w_v}}T_{int}} + \left(\frac{w_v}{w_d}\right)\ddot{x}_{tar}(t_1))}{1 - e^{2\sqrt{\frac{w_d}{w_v}}T_{int}}} \\ C_2 &= -C_1 e^{2\sqrt{\frac{w_d}{w_v}}T_{int}} + \left(\frac{w_v}{w_d}\right)\ddot{x}_{tar}(T_{int})e^{\sqrt{\frac{w_d}{w_v}}T_{int}} \\ C_3 &= \frac{y_0(t_1) - \ddot{y}_{tar}(T_{int})(\frac{w_v}{w_d})e^{\sqrt{\frac{w_d}{w_v}}T_{int}} + \left(\frac{w_v}{w_d}\right)\ddot{y}_{tar}(t_1))}{1 - e^{2\sqrt{\frac{w_d}{w_v}}T_{int}}} \\ C_4 &= -C_3 e^{2\sqrt{\frac{w_d}{w_v}}T_{int}} + \left(\frac{w_v}{w_d}\right)\ddot{y}_{tar}(T_{int})e^{\sqrt{\frac{w_d}{w_v}}T_{int}} \end{aligned} \tag{3.25}$$

In the above equations, $x_0(t_1) = x(t_1) - x_{tar}(t_1)$ and $y_0(t_1) = y(t_1) - y_{tar}(t_1)$. The control variables are not explicitly driven to zero at T_{int} and what happens after interception is out of the scope of the present context. For all higher order derivatives appearing in (3.24) and (3.25), estimation and evaluation at T_{int} is done in each iteration, as explained before.

3.5.1 Selection of Design Parameters: Tracking

The process of determining the length of the fixed horizon, T_{int} is heuristic, but not completely arbitrary. It is possible to make a calculated guess about T_{int}, depending on how complicated the environment is. The current battery status of the vehicle is also a deciding factor. As a test case, consider a simulation example, where T_{int} is assigned to be 5s from the start. The target starts from $(3.2, 2)$m and moves with a speed of 0.2m/s along the x-axis. The pursuer initiates tracking from the position $(2, 2.5)$m and aims to intercept the linearly moving target

after 5s. Figure 3.5 demonstrates how the ratio of the relative weights, $w = \frac{w_d}{w_v}$

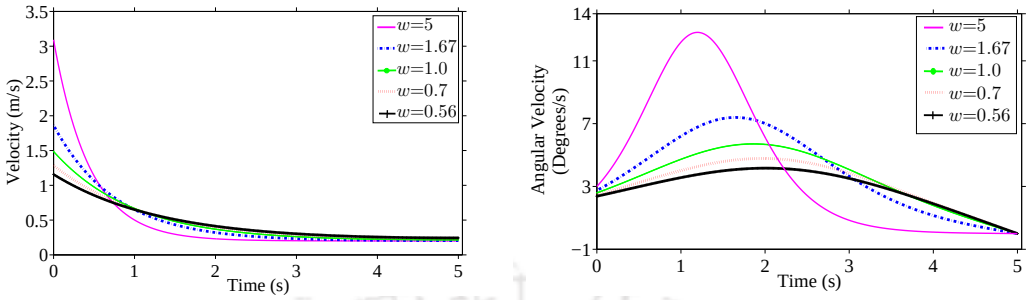

Figure 3.5: Pursuer's trajectory changes shape due to variation of linear and angular velocity profiles for different ratios of relative weights in the cost function.

is indicative of the changes in the linear and angular velocity. It has been observed that individual magnitudes of the weights, w_d (weight to tracking error) and w_v (weight to control effort) directly influence the cost incurred. The range of acceptable values are chosen on the basis of actuation and sensing limits. A high weightage on tracking ($w_d >> w_v$) improves tracking performance, but generates larger overshoots (see Figure (3.5)) in the trajectory, which is an indication of physical instability and higher expenditure of energy. Whereas, the converse results in a reduction of control effort, but also indicates slower convergence of tracking error (see Table 3.5). Although the inverse relationship of the weights w_d

w_d	w_v	0.2	0.6	1.0	1.4	1.8
1.0	$ATE(m)$	0.27	0.36	0.41	0.44	0.47
1.0	$CE(m^2/s)$	2.59	1.74	1.51	1.40	1.30
w_v	w_d	0.2	0.6	1.0	1.4	1.8
1.0	$ATE(m)$	0.59	0.46	0.41	0.38	0.35
1.0	$CE(m^2/s)$	1.10	1.30	1.51	1.68	1.80

Table 3.5: Inverse relation between ATE and CE : Role of weights.
ATE: Average Tracking Error, CE: Control Effort

and w_v has already become clear in terms of evaluating the average tracking error

(ATE) and control effort (CE) (see Table 3.5), a closer review (see Table 3.6) reveals that the ratio term, w is the primary factor that helps to achieve a desired performance level. On a practical note, tracking problems are usually supported

w_d	w_v	w	$ATE(m)$	$CE(m^2/s)$
1.66	1	*1.66*	0.36	1.74
1	0.6	*1.66*	0.36	1.74
0.5	0.3	*1.66*	0.36	1.74

Table 3.6: Performance actually depends on w
ATE: Average Tracking Error, CE: Control Effort

by backup plans upon encountering obstacles enroute. In the following chapter, we shall see how the proposed fixed horizon tracking problem can be incorporated into safe trajectory planning in dynamic situations, when one or more mobile obstacles with unknown motion constrain the admissible state-space of the pursuer. Comparative analysis and performance evaluation of optimal tracking using fixed terminal time approach that has been described here, will be illustrated in combination with a proposed safe planning procedure and in the light of a specific problem statement.

3.6 Summary of Chapter 3

In this chapter, we have introduced moving-target tracking from free and fixed horizon perspectives. Methods for computation of optimal trajectories, assumptions and scopes concerning those computations have been discussed. Computational and practical demerits of a free horizon problem have been elaborated and solution thereof have been proposed by a fixed terminal time approach. It is strongly suggested to follow Appendix A and Appendix B for related discussions on pursuer modeling and target state estimation used in the said problem formulations. In the following chapter proposed fixed horizon model will be assumed as default tracking mode.

Bibliography

[1] P.B. Bailey, L.F. Shampine and P.E. Waltman, "Nonlinear Two Point Boundary Value Problems," *NewYork and London : Academic Press*, (1968).

[2] C.F. Juang and Y.C. Chang, " Evolutionary-Group-Based Particle-Swarm-Optimized Fuzzy Controller with Application to Mobile-robot Navigation in Unknown Environments," *IEEE Transactions on Fuzzy Systems* **19**(2), 379–392 (2011).

[3] 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 Transactions on Fuzzy Systems* **21**(4), 688–701 (2013).

[4] S. Sun, H. Wang, J. Liu and Y. He, " Fast Lyapunov Vector Field Guidance for Standoff Target Tracking Based on Offline Search," *IEEE Access* **7**, 124797–124808 (2019).

[5] N. Malone, " High-Dimensional Motion Planning and Learning Under Uncertain Conditions," *Doctoral Dissertation* **11**(2), University of New Mexico (2015).

[6] S.X. Yang and M. Meng, " Neural Network Approaches to Dynamic Collision-free Trajectory Generation," *IEEE Transactions on Systems, Man, Cybernetics: B, Cybernetics* **31**(3), 302–318 (2001).

[7] B. Penin, P.R. Giordano and F. Chaumette, " Vision-Based Reactive Planning for Aggressive Target Tracking While Avoiding Collisions and Occlusions," *IEEE Robotics and Automation Letters* **4**(3), 3725–3732 (2018).

[8] E.H. Thyri, E.A. Basso, M. Breivik, K.Y. Pettersen, R. Skjetne and A.M. Lekkas, "Reactive collision avoidance for ASVs based on control barrier functions," *IEEE Conference on Control Technology and Applications, Montreal, Canada* 380–387, (2020).

[9] K.D. Campbell, "Tools for Trustworthy Autonomy: Robust Predictions, Intuitive Control, and Optimized Interaction," *Doctoral Dissertation, University of California, Berkeley*, (2017).

[10] C.D. Yang and C.C. Yang, "Optimal pure proportional navigation for maneuvering targets," *IEEE Transactions on Aerospace and Electronic Systems* **33**(3), 949–957 (1997).

[11] F.L. Lewis, D.L. Vrabie and V.L. Syrmos, "Optimal trajectories for nonholonomic mobile robots," *Optimal Control, John Wiley & Sons, Inc.*, (2012).

[12] F. Belkhouche and B. Belkhouche, "A Method for Robot Navigation Toward a Moving Goal with Unknown Maneuver," *Robotica* **23**(6), 709–720 (2005).

[13] F. Belkhouche, B. Belkhouche and P. Rastgoufard, "Line of sight robot navigation toward a moving goal," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* **36**(2), 6255–6267 (2006).

[14] L. Huang, "Control approach for tracking a moving target by a wheeled mobile robot with limited velocities," *IET Control Theory Applications***12**(3), 1565–1577 (2009).

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

[16] M. Guelman, M. Idan and O.M. Golan, "Three-dimensional minimum energy guidance," *IEEE Transactions on Aerospace and Electronic Systems* **31**(2), 835–841 (1995).

[17] D. Ghose, "True proportional navigation with maneuvering target," *IEEE Transactions on Aerospace and Electronic Systems* **30**(1), 229–237 (1994).

[18] Z. Ahmad, "An optimal robot tracking controller with application in the finite horizon case' *IEEE International Conference on Systems, Man, and Cybernetics* (5), 3277-3281 (2000).

[19] H. Huang and A.V. Savkin, "An Algorithm of Reactive Collision Free 3-D Deployment of Networked Unmanned Aerial Vehicles for Surveillance and Monitoring," *IEEE Transactions on Industrial Informatics* **16**(1), 132–140 (2020).

Chapter 4

Optimal Backup Path with Decoupled Controllers

4.1 Foreword

Practical environments involving target tracking usually contain obstacles. Obstruction may be imposed by static objects of fixed-geometry like pieces of furniture, walls, partitions in whole or portions thereof; or by mobile objects including living elements, autonomous and semi-autonomous vehicles. Nature of motion, probability of collision and occupancy density of obstacles depend on type of tracking problem. In such a scenario, the idea is to suspend the existing tracking control and initiate a safe backup plan. Tracking can be resumed after anti-collision manoeuvre with a careful consideration of switching conditions between the tracking and the collision avoidance modes. In addition, optimal performance markers for back-up trajectory not only need to be well-defined, but also validated by testing under a variety of dynamic constraints. This chapter describes a safe trajectory planning and optimal guidance controller with focus to special environments where physical navigation space is limited, such as corridors, alleys and aisles. Fixed-horizon optimal tracking discussed earlier in Chapter 3 shall remain the primary control objective in this case.

4.2 Concept of Decoupled Controllers

Without discrediting an existing primary tracking controller, collision avoidance strategy may be designed as a 'Plan-B' mechanism for a finite time interval. Upon detection and identification of obstacle(s) within a predefined vicinity of the pursuer, the obstacle intending to interfere with the current course of navigation is avoided by adopting a de-tour. Understandably, a collision needs to be predicted along the current direction of motion of the pursuer. Hence, the concept of 'avoidance cone' [19] came into being. A safe trajectory (interchangeably referred to as Plan-B, anti-collision path and avoidance trajectory, herewith) may be generated by a secondary set of control actions. The objective of the secondary controller is to avoid expected collision for a 'certain' duration, which has to be designed precisely to accommodate disturbances, unmodelled dynamics, estimation errors and unanticipated changes in the intent of the moving elements. The secondary control action is inevitably followed up by subsidiary control adjustments that *couples* the primary (tracking) and the secondary (anti-collision) control variables into a smooth seamless transition. This step is essential to avoid generation of oscillations and mechanical vibrations.

Reactive and self-adaptable planners like reinforced learning [10], recurrent neural network with fuzzy decisions [9] are fast evolving and realisable approaches to safe target tracking applications. Supervised learning has already been explored in multiple applications like dynamic navigation of outdoor robot subject to wheel sinkage (internal disturbance) [13], multiple-robot path planning subject to external disturbances [12] and navigation in a cluttered environment (reachability constraint) [11]. Not only that, deliberation on constraints related to target-tracking are also abundant. Usually, the most important constraints appear in terms of safety, comfort and conservation of energy/time [14, 15]. Performance and safety constraints have been handled in a decoupled fashion in [5], as well as by a contrasting integrated approach in [4]. Inspite of this vast prior-art, very few discussions are available on how to design a 'safe' backup plan when the reachable

state-space is very limited. This serves as a motivation for the work reported in this chapter.

4.2.1 Obstacle: Assumptions and Representation

A real-life obstacle may have arbitrary shape; however it is advantageous to represent an obstacle as a convex object with well defined geometry for modeling and representation purposes. In the present discussion we define an obstacle as a circular object completely circumscribing the actual obstacle. This assumption may seem to connive wastage of reachable space to some extent, but it largely simplifies mathematical definitions of forbidden states and reduces chances of encountering local minima traps, which may otherwise arise due to concave corners. The representative circle being a superset, convexification of obstacle boundary results in enlargement of forbidden space such that, all conditions imposed by path constraints are still valid. This means, collision avoidance with the representative obstacle implies collision avoidance for the original obstacle(s). The advantages offered by 'buffer space' of the representative obstacle are pronounced in case the obstacle exhibits unanticipated motion. Care has been taken to minimize wastage of reachable space while constructing the representative obstacle. The procedures that help obtain a tight approximation have been discussed in Appendix A, where we have illustratively shown the transformation of exemplary isolated obstacles, a cluster of closely spaced obstacles, walls and corners into one or more representative obstacles. Assuming, the environment has adequate features for detection and identification and advanced image processing techniques for reasonably accurate estimation of position and extent(size) of the detected object are available, radius r_{obs} and position $(x_{obs}(t), y_{obs})(t)$ of a representative obstacle may be computed at any time, t.

4.3 Problem Formulation

A simple space-saving technique for cost effective deployment of optimal path planners in 'narrow' spaces, is to design an anti-collision trajectory in the light of 'wall following' manoeuvre. In short, this means the vehicle (pursuer, in tracking applications) is obligated to maintain a fixed distance from an obstacle for the entire duration of Plan-B. Although, maintaining a fixed distance is a tight constraint that may seem too conservative and particularly sensitive to external disturbances; nevertheless, it ensures constant safety score and controlled expenditure of energy. The proposed anti-collision trajectory is basically a circular arc drawn at a *safe* distance d, from the center of the representative obstacle, as can be seen in equation (4.1). Ref. [19, 20] discuss a circular navigation guidance (CNG) scheme that has been inspired by missile guidance applications. In CNG too, the pursuer follows a cicular arc and intercepts a target at a pre-defined impact angle. The schemes discuss optimization of a minimum miss-distance error that performs far better than the conventional proportional navigation guidance; however, the problem of dynamic routing due to manoeuvring targets and limited workspace have not been addressed. Equation (4.1) represents the said *path constraint*. The term, 'safe' is relative and hence, d is a design variable. Safety mandates d to be related to radii r_{obs} (obstacle) and $r_{pursuer}$ (pursuer) according to the relation $\|d^2\| > \|r_{obs}^2 + r_{pursuer}^2\|$.

$$(x(t) - x_{obs}(t))^2 + (y(t) - y_{obs}(t))^2 - d^2 = 0 \tag{4.1}$$

The optimal control problem for anti-collision trajectory (interchangeably known as Plan-B, backup plan or safe trajectory, herein) minimizes control effort (squared velocity is an energy term representing control effort) subject to the path constraint defined by equation (4.1), a final state constraint, $F(t_3 + \Delta)$ and kinematic constraints of the pursuer, defined by state equations, $\dot{x}(t) = v(t)\cos(\theta(t))$ and

$\dot{y}(t) = v(t)\sin(\theta(t))$. The minimization problem is summarized below as P_{avoid}:

$$
\begin{aligned}
&P_{avoid}: \min_{(v(t),\theta(t))\in\mathcal{U}} J_{avoid} \quad \text{subject to kinematic constraints} \\
&\begin{cases} \dot{x}(t) = v(t)\cos(\theta(t) \\ \dot{y}(t) = v(t)\sin(\theta(t)) \end{cases} \\
&J_{avoid}(t) = F(t_3+\Delta) + \int_{t_3}^{t_3+\Delta} \left(w_v v^2(t)\right) \mathrm{d}t \\
&F(t_3+\Delta) = \frac{1}{2}\Big((x(t_3+\Delta) - x_{tar}(t_3+\Delta))^2 + (y(t_3+\Delta) - y_{tar}(t_3+\Delta))^2\Big)
\end{aligned} \tag{4.2}
$$

The lower limit of integration, t_3, marks the time instant when Plan-B is activated and Δ denotes duration of Plan-B. Selection of t_3 implicitly depends on the choice of the safety margin, d. On the other hand, the design parameter, Δ is determined by two factors. One factor is d, which effectively refers to size (extent) of the representative obstacle. The second factor is a future estimate of 'closeness' of the pursuer from the target while executing the avoidance trajectory. In fact, the final state constraint, $F(t_3+\Delta)$ in the cost functional (4.2) represents the optimal conditioning that prepares the avoidance trajectory (even before the pursuer embarks upon Plan-B), such that, the pursuer is enabled to reach closest possible location from the target subject to said path constraint at a time $(t_3+\Delta)$. This is the time when the pursuer resumes tracking or Plan-A mode of navigation.

In an exemplary situation, let us assume a first switching event from tracking (Plan-A) to collision avoidance (Plan-B) mode at t_3 and a second switching event from Plan-B to Plan-A at $(t_3+\Delta)$. This optimal backup plan is novel in the sense that it offers tremendous benefits to both collision avoidance and tracking. In the former case, it helps maintaining safety margin till resumption of Plan-A. This step plays a pivotal role in eliminating or at least minimizing possibilities of collision caused by mis-timing of the second switching event. In the latter case, it aids in minimizing tracking error due to the optimal placement of the pursuer at the time of second switching. The relation between Δ and the switching condi-

tions can be summarized in equation (4.4), where, dt is sampling time. The first switching condition corresponds to equation (4.3) and the second switching condition corresponds to equation (4.4), provided the entire task of tracking, collision avoidance and switching are completed within $[0, T_{int}]$, where T_{int} is a finite time interval obtained either by pre-design (for a fixed finite time tracking problem) or by computing a free terminal-time tracking problem.

Switching Conditions:

Tracking to Avoidance:

$$t_3 := \min t \mid \{x, y\} \text{ satisfies optimal tracking tracjectories and} (4.1), \forall t \in [0, T_{int}] \tag{4.3}$$

Avoidance to Tracking:

$$t_3 + \Delta := \min t \mid \{J_{switch}(t - dt) > J_{switch}(t) < J_{switch}(t + dt)\} \tag{4.4}$$

$$J_{switch}(t) := ((x(t) - x_{tar}(t))^2 + (y(t) - y_{tar}(t))^2 \mid \{x, y\} \text{ satisfies (4.1)}$$

$$\forall t \in [t_3 + dt, T_{int}]$$

4.4 Optimal Conditions and Trajectories

The Hamiltonian function for the collision avoidance mode, $\mathcal{H}_{avoid}$ is constructed from (4.2) according to necessary optimality conditions as described in (4.5), where $\mathcal{H}_{avoid}$ is a function of states $(x(t), y(t))$, co-states $(\lambda_1(t), \lambda_2(t))$ and control inputs to the pursuer, $(v(t), \theta(t))$.

$$\begin{aligned} \mathcal{H}_{avoid} = &- w_v v(t)^2 + \lambda_1(t) v(t) \cos(\theta(t)) + \lambda_2(t) v(t) \sin(\theta(t)) \\ &+ \lambda_3(t) \Big((x(t) - x_{obs})^2 + (y(t) - y_{obs})^2 - d^2 \Big) \end{aligned} \tag{4.5}$$

The 'Stationarity' conditions, indicative of the control invariance of $\mathcal{H}_{avoid}$ are derived as in equation (4.6) by partially differentiating the Hamiltonian function (4.5) with respect to the control variables, forward velocity $v(t)$ and heading $\theta(t)$. We assume that the pursuer's velocity is positive definite[1] for all $t \in [0, T_{int}]$

[1]Velocity is always along forward direction; there is no backward motion.

and slightly rearrange the variables to obtain equation (4.6). Position of the representative obstacle, (x_{obs}, y_{obs}) is obatined at every t as a numerical value obtained through measurement or estimation.

$$\begin{aligned} v(t) &= \frac{1}{2w_v}\sqrt{\lambda_1^2(t) + \lambda_2^2(t)} \\ \theta(t) &= \tan^{-1}\left(\frac{\lambda_2(t)}{\lambda_1(t)}\right) \end{aligned} \tag{4.6}$$

The path constraint (4.1) has already established the shape of the avoidance trajectory as a circular arc. Equality of the constraint to a suitably designed safety margin introduces an additional state equation into the boundary value problem (4.7) explained below, because of which, a third Lagrange's multiplier λ_3 has been used in addition to λ_1 and λ_2. The first and the second Lagrange's multipliers are functions of time and correspond to the two kinematic state equations governing non-holonomic motion of the pursuer. Therefore, the dimension of the optimal problem increases by two (one equation from each additional state and co-state) as compared to the tracking mode. We will see later in the discussion, that λ_3 must be a constant.

$$\left.\begin{aligned} \dot{x}(t) =& \frac{\lambda_1(t)}{2w_v} \\ \dot{y}(t) =& \frac{\lambda_2(t)}{2w_v} \\ \dot{z}(t) =& \left((x(t) - x_{obs})^2 + (y(t) - y_{obs})^2 - d^2\right) = 0 \\ \dot{\lambda}_1(t) =& -2\lambda_3(x(t) - x_{obs}) \\ \dot{\lambda}_2(t) =& -2\lambda_3(y(t) - y_{obs}) \\ \dot{\lambda}_3(t) =& 0 \end{aligned}\right\} \tag{4.7}$$

From a control systems designer's point of view, equation (4.7) represents an interesting problem, since the pursuer states are unavailable and in fact, cannot be predicted at both boundaries. It may be recalled that, these computations are performed on or before t_3, wherein, the target's, obstacles's and pursuer's positions at a future time, $(t_3 + \Delta)$ are all unknown. Also, unlike Plan-A, Plan-B is not supposed to terminate at a specific set of states or on a predefined associate

trajectory, characterized in sharing terminal time and states with the pursuer. This makes the relation between the pursuer and the target states at $(t_3 + \Delta)$ inconclusive. Hence, equations in (4.7) have to be solved as 'split-condition two-point boundary value problem' [7], that requires evaluation of 'Transversality' conditions.

An iterative polynomial interpolation method (Neville-Aitken [4]) has been adopted for estimation of terminal states. Predictive modeling has been preferred over explicit modeling because the latter often fails to accommodate the true dynamic behaviour of the mobile elements. Alternative estimation techniques like, numerical methods, regression models and proabilistic approaches can also be used, especially if uncertainties and disturbances [8] are to be taken into account. Among recent studies, long term prediction of horizon for 'intent assessment' of driverless automobiles [16], is believed to be quite a promising technique. In short, estimation of target trajectory for computation of boundary values is actually open to further exploration.

Verification of anti-collision trajectory

Problems of this kind (4.7) usually warrant numerical solutions [1, 2]. Finding an explicit solution to (4.7) is unnecessary in this case, because equation (4.1) represents the optimal avoidance trajectory. However, we can verify how the trajectories evolve, in order to validate our claim that the safety margin is maintained through entire duration of the avoidance mode. Differentiating $\dot{z}(t) = 0$ once again, gives (4.8).

$$\frac{\lambda_2(t)}{\lambda_1(t)} = -\frac{(x(t) - x_{obs})}{(y(t) - y_{obs})} \tag{4.8}$$

Point to be noted, λ_3 is a non-zero constant and hence, replacing expressions for $\dot{\lambda}_i(t), i \in \{1, 2\}$ from (4.7) into (4.8) result in the relation (4.9).

$$\lambda_1\dot{\lambda}_1 + \lambda_2\dot{\lambda}_2 = 0 \tag{4.9}$$

It is now evident that $\lambda_1^2(t) + \lambda_2^2(t)$ is a constant. A further differentiation of (4.9) followed by appropriate replacements of the time derivatives of state and costates

from (4.7) eventually leads to the relation (4.10). The left hand side of (4.10) can be replaced by $4w_v^2v^2(t)$ from equation (4.6) and this step clearly substantiates that the velocity is constant during the avoidance mode.

$$\lambda_1^2(t) + \lambda_2^2(t) = 4\lambda_3 w_v d^2 \tag{4.10}$$

Now, to verify the shape of the collision avoidance trajectory, expression for orientation in (4.6) is differentiated and the derivatives of the co-states are replaced from equation (4.7). The resulting expression for angular velocity shown in (4.11) can be further differentiated to yield $\ddot{\theta}(t)$=0.

$$\begin{aligned}\dot{\theta}(t) &= \frac{1}{\sec^2\theta}\left(\frac{\dot{\lambda}_1\lambda_2 - \dot{\lambda}_2\lambda_1}{\lambda_1^2}\right)\\ &= \frac{2\lambda_3\left((y(t)-y_{obs})\lambda_2 - (x(t)-x_{obs})\lambda_1\right)}{\lambda_1^2(t)+\lambda_2^2(t)}\\ &= \frac{(y(t)-y_{obs})\lambda_2 - (x(t)-x_{obs})\lambda_1}{2w_v d^2}\end{aligned} \tag{4.11}$$

This means, rate of change of orientation also remains constant during avoidance. These two results together signify that the pursuer indeed moves along a circular arc and maintains the path constraint throughout the avoidance mode.

Solution to the Boundary Value Problem

Although solution to (4.7) is not required for implementing the proposed strategy, but actually finding one becomes a really intriguing problem, with only the state boundary values at t_3 being available. However, the costates, $\lambda_i, i \in \{1,2\}$ can be computed from the 'Transversality' conditions given in (4.12), when tracking resumes at $(t_3+\Delta)$.

$$\left.\begin{aligned}\lambda_1(t_3+\Delta) &= x_{tar}(t_3+\Delta) - x(t_3+\Delta)\\ \lambda_2(t_3+\Delta) &= y_{tar}(t_3+\Delta) - y(t_3+\Delta)\end{aligned}\right\} \tag{4.12}$$

Evidently, (4.1) and (4.10) are also valid at $(t_3+\Delta)$ and hence we can write equations (4.13) and (4.14).

$$(x(t_3+\Delta)-x_{obs})^2 + (y(t_3+\Delta)-y_{obs})^2 - d^2 = 0 \tag{4.13}$$

$$\lambda_1^2(t_3+\Delta)+\lambda_2^2(t_3+\Delta)=4\lambda_3 w_v d^2 \tag{4.14}$$

To solve for the admissible controls, (4.12) and (4.13) are combined and (4.15) is obtained.

$$\begin{aligned}&\big(x_{tar}(t_3+\Delta)-\lambda_1(t_3+\Delta)-x_{obs}\big)^2+\big(y_{tar}(t_3+\Delta)\\&-\lambda_2(t_3+\Delta)-y_{obs}\big)^2=d^2\end{aligned} \tag{4.15}$$

Replacing $\dot{\lambda}_1(t_3+\Delta)=-2\lambda_3(x(t_3+\Delta)-x_{obs})$ and $\dot{\lambda}_2(t_3+\Delta)=-2\lambda_3(y(t_3+\Delta)-y_{obs})$ from (4.7) into (4.9) gives (4.16). States $x_{tar}(t_3+\Delta)$ and $y_{tar}(t_3+\Delta)$ are estimated by polynomial extrapolation as mentioned before (See Appendix A).

$$-\lambda_1(t_3+\Delta)(2\lambda_3(x(t_3+\Delta)-x_{obs}))-\lambda_2(t_3+\Delta)(2\lambda_3(y(t_3+\Delta)-y_{obs}))=0 \tag{4.16}$$

Now, simultaneously solving (4.14), (4.15) and (4.16), the terminal boundary values $\lambda_1(t_3+\Delta)$ and $\lambda_2(t_3+\Delta)$ and also the constant λ_3 can be calculated. The initial co-states $\lambda_1(t_3)$ and $\lambda_2(t_3)$ can also be derived from (4.8) and (4.10). Constant avoidance velocity can be computed from $\lambda_1(t_3+\Delta)$ and $\lambda_2(t_3+\Delta)$ following the relation (4.6). The heading is found to change linearly with time according to (4.17).

$$\theta(t)=\dot{\theta}(t)(t-t_3)+\tan^{-1}\left(\frac{\lambda_2(t_3)}{\lambda_1(t_3)}\right) \tag{4.17}$$

With an appropriate change of variables, it can be easily shown that the linear second order equivalent of (4.7) is bounded and Lipschitz continuous [3]. Hence a unique solution is guaranteed to exist. Necessary derivations have been discussed in Appendix B.

4.4.1 Smoothening Filter

The need for a smoothening filter stems from the fact that switching controllers are prone to generate sudden jerk due to a mismatch in control variables of either modes. A control command which is not smooth is not only an inefficient way

to drive the vehicle, but also introduces disturbances in motion in the form of mechanical vibrations. To avoid this situation, a smoothening filter has been designed to operate at switching. The filter takes a convex combination of control trajectories from the tracking and avoidance modes within a certain interval about the switching point. This simple step efficiently dampens the disturbances and creates a smooth transition between the two modes. The filter uses a 'blending function', $\alpha(t)$ given in equation (4.18).

$$\alpha(t) = 2 - \exp\left(\frac{\ln(t + 2\tau - t_s)}{\tau}\right) \tag{4.18}$$

Here, $t = (t_s - \tau)$ marks the start of the smoothening process and τ is the duration of the operation (t_s is the switching instant). The blending function assumes a value between 1 and 0. At $(t_s - \tau)$ the current trajectory receives full weightage and at t_s the second trajectory has the complete hold. This concept can be applied to switching from tracking to avoidance mode and vice versa.

Choice of the blending function is not unique. Other functions such as an inverse hyperbolic tangent can also generate a suitable blending profile. Magnitude of τ has been determined by trials. It depends on the difference in the levels of the control variable and the slew-rate. In general, we can define a δ-ball as small as possible around the switching point. By doing so, the benefits of the optimal trajectories in either mode are preserved and safety conditions are not violated. The trajectory generated by the smoothening filter is sub-optimal, a feature unavailable with the geometric curve proposed in [6]. Yet, it is a very practical way to make smooth transitions and does not require intensive computations. The state trajectories inside the δ-ball can be reconstructed by converting the kinematic relations into time-discrete form and ignoring the higher order derivatives.

4.5 Observations and Analysis

Performance evaluation, validation of the proposed controller and smoothening technique have been tested by simulation studies. First, a set of experiments have

been designed to determine a guideline for selecting the parametric values. Then the smoothening filter has been validated, followed by comparative performance analysis of the proposed control strategy in the light of current literature.

4.5.1 Selecting the Design Parameters

The most significant design parameter in the anti-collision mode of navigation is the duration of avoidance, Δ. The parameter, Δ is a function of the obstacle size, r_{obs} and hence, d. Safety margin, d is selected according to the relation $d^2 > r_{obs}^2 + r_{pursuer}^2$. Choice of an optimal avoidance duration, Δ is made before t_3, which marks the first switching instant from the tracking to the anti-collision mode. A novel approach has been introduced here, which makes a calculated projection of cost functional (4.2) over time. The time $t \in [t_3 + dt, T_{int}]$ which yields the minimum cost, is selected as $(t_3 + \Delta)$. This will be clarified further by another test case.

In an exemplary simulation we consider a target to start moving along the x-axis from the position $(3.2, 2)$m and at speed of 0.2m/s. An obstacle is assumed to be located at $(2.9, 3)$m. The pursuer starts tracking the target from the position $(2, 5)$m and is expected to intercept the moving target in 8s. The avoidance duration has been selected as 1.6s, safety margin is 0.45m and the weighing parameters of the objective function are chosen as, w_d=0.5 and w_v=1. Investigation shows that the cost reaches a minimum of 2.05 for Δ = 1.7s and increases on either directions as we reduce or increase Δ from 1.7, for the given value of d (see Table 4.1).

Δ	1	1.5	1.6	1.7	1.8	1.9	2
Cost	2.32	2.1	2.07	2.05	2.06	2.09	2.14

Table 4.1: Relation between avoidance cost and Δ

Figure 4.1 demonstrates how an arbitrary assignment of Δ can influence the

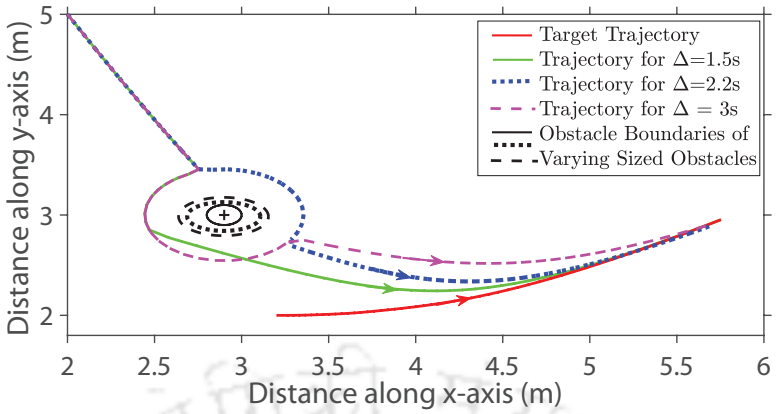

Figure 4.1: The optimal trajectory varies due to different avoidance durations. Inappropriate avoidance duration can violate safety.

resulting optimal avoidance trajectories. It may be noted that in the concerned example, same value of d has been chosen for obstacles of different sizes. Among the trajectories depicted in Figure 4.1, the green trajectory corresponding to $\Delta =$ 1.5s may be appropriate for the smallest obstacle, but it may not be suitable for the largest obstacle. In case the obstacle starts to move without prior intimation, the pursuer following the green trajectory may encounter collision. This is an example, where inappropriate computation of switching point can lead to violation of safety and this is why an optimal termination condition for the anti-collision mode is so necessary. In this context one may refer to the famous MIT vs. Cornell collision case of DARPA challenge in 2007, where unprecedented movement of one vehicle caused the other vehicle to collide while performing a change of lane motion. Figure 4.1 also points out that the anti-collision curve may resemble either a clockwise or an anticlockwise arc around the obstacle. By definition of the final constraint function $F(t_3 + \Delta)$ in equation (4.2), there can be only one solution for each case, because, the target is non-stationary and the pursuer is optimally placed to resume tracking at the end of the anti-collision manoeuvre.

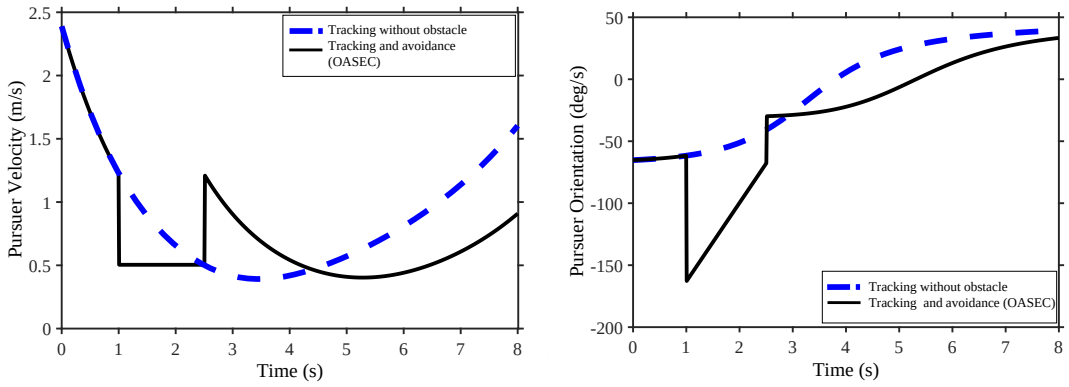

Figure 4.2: Computed optimal linear velocity and heading profile of the pursuer for tracking and obstacle avoidance. The proposed method is indicated as 'Obstacle Avoidance with State Equality Constraint' (OASEC). The controls for no-obstacle case is shown in blue.

4.5.2 Smooth Trajectory Design at Switching

The following results describe the effects of the smoothening filter. Figure 4.2(a) and Figure 4.2(b) confirm that the linear and angular optimal velocities are constant in the collision avoidance mode. It further affirms the presence of jerk in the control variables at the switching instants. In practice, no dynamical system is capable of handling this kind of discontinuity in the control signal. While a sudden jump in the control channel is an undesired phenomenon from the controller's perspective, this is also considered to be a negative indicator of 'comfort' if the vehicle is a carrier for passenger or fragile goods. Jerk is a natural source of mechanical vibrations and oscillations which can destabilize the functioning of the system and interfere with positional accuracy in worse cases. In the simulated experiments reported above, the smoothening timeframe has been chosen to be equal to a sampling interval. As explained before, it ranges between the values 1 and 0. Figure 4.3 shows how the control signals, forward velocity, v and heading, θ of the pursuer make smooth transitions between tracking and avoidance modes. The resultant state trajectories are also smooth, wherein the state trajctories are

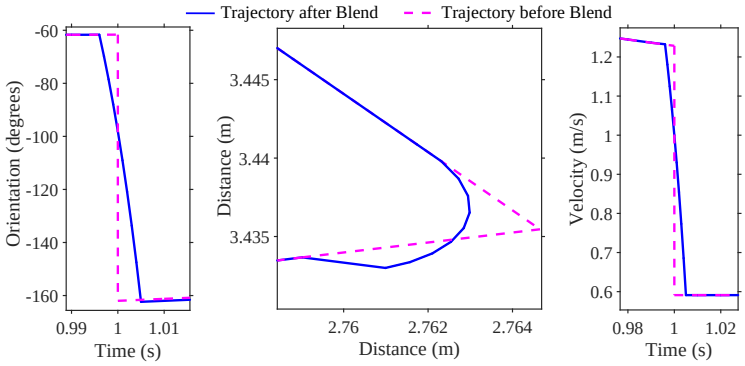

Figure 4.3: Smoothening of control trajectories at switching between tracking and avoidance modes generate practically realisable path.

reconstructed based on the state equations. It may be noted that, the smoothened path is not simply a geometric curve as proposed by [6] but follows the kinematic relations. Hence, the reconstructed states within the smoothening interval do not make an exact fit inside the asymptotic references provided by the original trajectories. The percentage of deviation from desired optimal path introduced by the sub-optimal smoothening filter is well within accepted dynamic limits.

4.5.3 Comparison of Performance

Having described the design subsystems, a comparative study of the proposed guidance scheme against two recent literatures is now presented.

The first literature discloses is a safety-critical backup controller (SCBC) designed on the basis of a control barrier function according to [5]. Similar to our proposed strategy, it develops a safe backup trajectory, decoupled from the original task. However, the mechanism leverages the concept of control invariant 'reachable backup sets' under a stable backup control law (a proportional-integral-derivative controller has been chosen in the experiment) and the backup trajectories are computed following the gradient information. The second guidance scheme considered for comparative study is a nonlinear integrated control approach (NIC) which solves a constrained optimization problem combining both tracking and

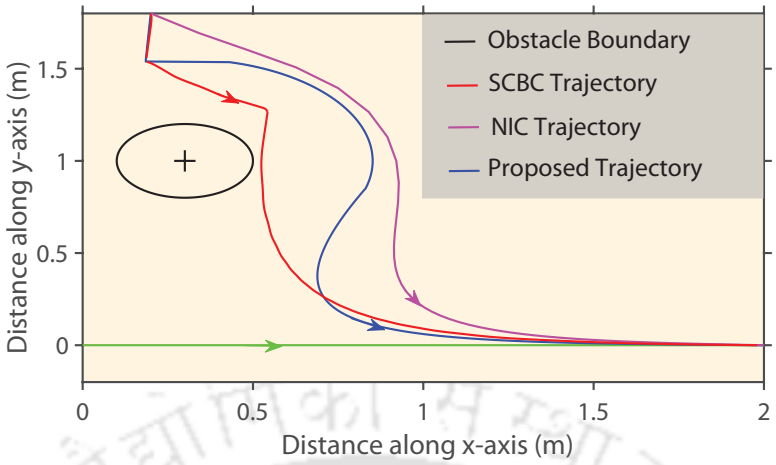

Figure 4.4: Comparison of trajectories from different guidance methods. SCBC [5] is a control barrier function based method and NIC [4] is an integrated optimization and control method. The yellow region is the arena representing a corridor.

collision avoidance, inspired by [4]. This integrated controller performs an 'active' collision avoidance, which means, the pursuer tends to proactively move away from the detected obstacle by utilization of a quadratic distance function. To understand the following simulation, consider a corridor-like environment where the target is at the position $(0, 0)$m at $t = 0$ and is moving at speed of 0.1m/s. An obstacle located at $(0.3, 1)$m is detected at $t = 0$. Suppose, the obstacle is stationary when detected. At $t = 0$, the pursuer is at the position $(2, 5)$m. The objective is to plan a tracking trajectory from $t = 0$ till interception at $t = T_{int} = 8$s, such that the obstacle is avoided keeping a safety margin of 0.5m. The appropriate avoidance duration has been computed to be around 1s. The weights to the compound cost components have been selected as w_d=0.5 and w_v=0.4.

Table 4.2 summarizes the merits of our proposed method in terms of performance and safety parameters. Although the results only depict a specific case study, a qualitative analysis can be carried out to explain other possibilities. SCBC, is a more flexible mechanism, which may be intuitively more relevant for

Methods	ATE (m)	CE (m^2/s)	MSM(m)
$SCBC$	0.912	1.946	0.224
NIC	0.900	8.298	0.609
$Proposed$	0.669	1.555	0.533

Table 4.2: Comparison of performance and safety metrics.
ATE: Average Tracking Error, CE: Control Effort,MSM: Minimum Safety Margin

cases where multiple obstacles forbid the maintenance of a conservative safety margin. However, the advantage of proposed method in this narrow workspace is that, it provides an appropriate optimal switching point for resumption of tracking. Figure 4.4 shows that the SCBC trajectory marginally avoids the obstacle because of a tight approximation of the reachable set. Now, it may be carefully considered that, if the obstacle in Figure 4.4 starts to exhibit sudden movement, there might be a possibility for collision for the trajectory devised by SCBC. On the other hand, experiments show that NIC generates a very relaxed trajectory because of the trade-off between the tracking and collision avoidance subtasks. A major portion of the safe reachable set is unnecessarily left out due to this trade-off and it is observed that the trajectory moves dangerously close towards the wall. Reassignment of the weighing parameters is capable of making little or no improvement to the existing situation because of the contrasting nature of active tracking and active collision avoidance strategies. The control effort in this case is also substantially higher than proposed technique.

4.6 Summary of Chapter 4

In this chapter, we have discussed a decoupled navigation planner for moving-target tracking, with priority driven generation of collision-safe optimal backup plan upon suspension of the primary tracking trajectory. Tracking trajectory is resumed after a designed optimal duration, at the beginning and the end of which, controllers switch navigation modes based on a set of constraints. The optimal

switching conditions have been established and a smooth transition mechanism has been proposed between the tracking and anti-collision controllers. The proposed strategy has been shown to be well suited to limited space scenarios such as corridors and aisles. Characteristics of the complete navigation profile inclusive of tracking and collision avoidance have been validated by simulations and experimental studies in relation to evaluation of comparative performances.

Bibliography

[1] U.M. Ascher and L.R. Petzold, "Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations," *Society for Industrial and Applied Mathematics,1st edn, Philadelphia, PA, USA*, (1998).

[2] J. Kierzenka and L.F. Shampine, " A BVP Solver Based on Residual Control and the Maltab PSE," *ACM Transactions on Mathematical Software* **27**(3), 299–316 (2001).

[3] P. Waltman, "Existence and Uniqueness of Solutions of Boundary Value Problems for Two Dimensional Systems of Nonlinear Differential Equations," *Transactions on American Mathematical Society* **153**, 223–234 (1971).

[4] 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 Robotics and Automation Letters*, Preprint.

[5] 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).

[6] X. Wu, J. Angeles, T. Zou, J. Yang, H. Li and W. Li, "Trajectory Planning with Lamé-Curve Blending for Motor-Saturation Avoid-

ance Upon Mobile-Robot Turning," *IEEE Access* **8**, 58483–58496 (2020).

[7] P.B. Bailey, L.F. Shampine and P.E. Waltman, "Nonlinear Two Point Boundary Value Problems," *NewYork and London : Academic Press*, (1968).

[8] N. Malone, "High-Dimensional Motion Planning and Learning Under Uncertain Conditions," *Doctoral Dissertation, University of New Mexico* **11**(2), (2015).

[9] 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 Transactions on Fuzzy Systems* **21**(4), 688–701 (2013).

[10] Q. Wu, X. Gong, K. Xu, D. Manocha, J. Dong and J. Wang, "Towards Target-driven Visual Navigation in Indoor Scenes via Generative Imitation Learning," *IEEE Robotics and Automation Letters* **6**(1), 175–182 (2020).

[11] L. Liu, D. Dugas, G. Cesari, R. Siegwart and R. Dubé, "Robot navigation in Crowded Environments Using Deep Reinforcement Learning," *IEEE/RSJ International Conference on Intelligent Robots and Systems* Las Vegas, 5671–5677 (2020). **6**(1), 175–182(2020)

[12] C. He, Y. Wan, Y. Gy and F.L. Lewis, "Integral Reinforcement Learning-based Minimum time-energy Path Planning Subject to Collision Avoidance and Unknown Environmental Disturbances," *IEEE Control System Letters* **5**(3), 983–988(2021).

[13] H. Taghavifar, B. Xu, L. Taghavifar and Y. Qin, "Optimal Path-planning of Nonholonomic Terrain Robots for Dynamic Obstacle

Avoidance Using Single-time Velocity Estimator and Reinforcement Learning Approach," *IEEE Access* **7**, 159347–159356 (2020).

[14] G. Manor, Z. Ben-Asher and E. Rimon, "Time-optimal Trajectories for a Mobile Robot Under Explicit Acceleration Constraints," *IEEE Transactions on Aerospace and Electronic Systems* **545**, 2220–2232 (2018).

[15] N. Ganganath, C. Cheng and C.K. Tse, "A Constraint Aware Heuristic Path Planner for Finding Energy Efficient Paths on Uneven Terrains," *IEEE Transactions on Industrial Informatics* **113**, 601–611(2015).

[16] K.D. Campbell, "Tools for Trustworthy Autonomy: Robust Predictions, Intuitive Control, and Optimized Interaction," *Doctoral Dissertation, University of California, Berkeley*, (2017).

[17] 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).

[18] Y. Kuwata, M.T. Wolf, D. Zarzhitsky and T.L. Huntsberger, "Safe Maritime Autonomous Navigation With COLREGS, Using Velocity Obstacles," *IEEE Journal of Oceanic Engineering* **39**(1), 110–119 (2014).

[19] I.R. Manchester and A.V. Savkin, "Circular Navigation Missile Guidance with Incomplete Information and Uncertain Autopilot Model," *IEEE Journal of Guidance* **27**(6), 1078–1083 (2004).

[20] I.R. Manchester and A.V. Savkin, "Circular Navigation Guidance Law for Precision Missile/Target Engagements," *IEEE Conference an Decision and Control* **2**, 1287-1292 (2002).

Chapter 5

Dynamic Navigation with Optimal Integrated Controller

5.1 Foreword

Decoupled controllers are generally configured to offer temporally variable priority to long-term and shot-term goals. In the previous chapters we have designed moving-target tracking as a global optimization objective and collison avoidance as a local optimization strategy. Several existing literatures advocate in favour of decoupled controllers in terms of wider and more flexible range of admissible control inputs and reachable state-space. However, a major drawback of switching controllers is noise. On a different note, overall quantitative analysis of run-time cost effectiveness and safety score is dificult to obtain with decoupled controllers. In particular, applications such as autonomous pursuit vehicles, which are subjected to velocity and lane restrictions aligned with safety regulations, heavily rely on these parameters. These requirements serve as motivation to design an integrated controller for dynamic pursuit with intent-awareness related to co-commuters. This is an example of target tracking in a constrained dynamic environment which benefits from long-term optimization and at the same time, requires jerk-free motion for passenger comfort. Investigations have confirmed that intent-awareness and integrated optimal planning are mutually beneficial

characteristics of the proposed strategy.

5.2 Integrated Controller Design

The aim of integrated control is to generate an optimal trajectory over a planning horizon that maintains a balance between dynamic tracking and collision avoidance throughout the planning horizon. A weighted cost function has been designed to assign relative priorities to collision avoidance and tracking and achieve a trade-off in the process. The optimal control problem presents interesting challenges due to path and velocity constraints in addition to interception requirements at the end of a predefined time frame. Multiple obstacles can be simultaneously accommodated in the cost function, the number of obstacles being variable in each planning horizon. A 'region of interest' is a pre-defined 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. Obstacles beyond the region of interest are not considered in the current plan.

Although a gradient driven repulsive behaviour [5] has been disclosed in prior-art that describes avoidance of multiple obstacles at the same time, 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. In a related example [22], a stochastic path planning problem for delivery drones has been described, which maximizes probability of on-time delivery. Although path feasibility is determined based on battery health, lack of intent estimation of 'competetive' agents may limit applicability of the method to actual traffic. In another literature [23], local intent awareness is practised to reactively overcome deadlock. But the concerned technique may fail to regulate the vehicle's velocities in restricted motion-space, in case the crowd

pattern changes abruptly. This is why a long-horizon perspective is necessary for a back-up plan. In the present problem, keeping in mind, that the pursuer is required to intercept the target at a pre-determined time, the restrictions imposed by controlling the pursuer's velocity can be conflicting with the tracking operation.

The proposed controller-design aims to solve this intent driven complex self-regulated navigation problem. However, unlike a quadratic optimization suggested in [6], a penalty function based planning and constraint management has been applied in an optimal control framework. The problem statement can be summarized 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:

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 equation (5.1).

$$\Psi(x(T_{int}), y(T_{int})) = [x(T_{int}) - x_{tar}(T_{int}) \;\; y(T_{int}) - y_{tar}(T_{int})]^\top = [0 \;\; 0]^\top \quad (5.1)$$

The function, $\Psi(x(T_{int}), y(T_{int}))$ is called the final state function and equation (5.1) is referred to as the *final state constraint*.

Collision avoidance with the i^{th} obstacle can be achieved provided, the pursuer is able to maintain a pre-defined minimum distance of separation or safety margin, d_i from the obstacle according to equation (5.2).

$$(x(t) - x^i_{obs}(t))^2 + (y(t) - y^i_{obs}(t))^2 > d_i^2 \quad (5.2)$$

Here, d_i is the minimum safety margin 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 (5.2) will be referred to as the *path constraint*.

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 equation (5.3).

$$v(t) < v_{max} \tag{5.3}$$

The objective function for the navigation problem can be defined as a weighted sum of these constraints as shown in equation (5.4).

$$J_{int} = \int_0^{T_{int}} \Big(- w_v \ln(v_{max} - v(t)) - w_b \sum_{i=1}^{m} \ln \big((x(t) - x^i_{obs}(t))^2 + (y(t) - y^i_{obs}(t))^2 - d_i^2\big) + w_d\big((x(t) - x_{tar}(t))^2 + (y(t) - y_{tar}(t))^2\big)\Big) \mathrm{d}t \tag{5.4}$$

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 optimal control problem is fundamentally a minimization problem for the objective function, J_{int} as demonstrated by equation (5.5).

$$P_{integrated} : \min_{(v(t),\theta(t))\in\mathcal{U}} J_{int} \tag{5.5}$$

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 (5.1), (5.2) and (5.3)

where $\mathcal{U} \in (\mathbb{R}^+ \times \mathbb{R})$ is the set of 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. To solve the optimization problem $P_{integrated}$, optimality conditions for the control variables, $v(t)$ and $\theta(t)$ are derived by applying necessary optimality conditions (Section 5.2.1). The solution to $P_{integrated}$ 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.

5.2.1 Optimality Conditions

A real valued Hamiltonian function, $\mathcal{H}_{int}$ is constructed as in (5.6). At any $t \in [0, T_{int}]$, if the constraints (5.1),(5.2) and (5.3) are satisfied, then the Hamiltonian $\mathcal{H}_{int}$ reaches its maxima and incurs a minimal cost.

$$\begin{aligned}\mathcal{H}_{int} =& w_v \ln(v_{max} - v(t)) - w_d((x(t) - x_{tar}(t))^2 + (y(t) - y_{tar}(t))^2) \\ &+ w_b \sum_{i=1}^{m} \ln((x(t) - x^i_{obs}(t))^2 + (y(t) - y^i_{obs}(t))^2 - d_i^2) \\ &+ \lambda_1(t) v(t) \cos(\theta(t)) + \lambda_2(t) v(t) \sin(\theta(t)) \end{aligned} \tag{5.6}$$

Two new variables, $\lambda_1(t)$ and $\lambda_2(t)$ have been introduced in the expression for $\mathcal{H}_{int}$ 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 [3] this problem utilizes 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 minimization upon satisfaction of (5.3). This further implies that the controller facilitates minimum control effort, computed as $\int_t^{T_{int}} v^2(t)dt,\ (0 \le t \le T_{int})$.

'Stationarity' of the Hamiltonian with respect to variations in the control vector leads to equation (5.7),which expresses the control variables in terms of the

costates.

$$\frac{\partial \mathcal{H}_{int}}{\partial v} = 0 \Rightarrow v(t) = v_{max} - \frac{w_v}{\sqrt{\lambda_1^2(t) + \lambda_2^2(t)}}$$
$$\frac{\partial \mathcal{H}_{int}}{\partial \theta} = 0 \Rightarrow \theta(t) = \tan^{-1}\left(\frac{\lambda_2(t)}{\lambda_1(t)}\right) \tag{5.7}$$

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 the steering is insufficient to maintain the minimum safety margin and a penalty has to be paid. In both of these cases the constraints are not violated but shows a possible violation in near future. To negate this possibility the cost is increased and the minimization problem, $P_{integrated}$ is triggered back into action. The role of the barrier function will be further elucidated in Section 5.2.3.

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 $\mathcal{H}_{int}$ 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 (5.8). This step converts the optimal control problem (5.5) into a system of non-linear ordinary differential equations (5.8), which can be solved with the knowledge of the boundary values.

$$\dot{x}(t) = \frac{v_{max}\lambda_1(t)}{\sqrt{\lambda_1^2(t) + \lambda_2^2(t)}} - \frac{w_v\lambda_1(t)}{\lambda_1^2(t) + \lambda_2^2(t)}$$
$$\dot{y}(t) = \frac{v_{max}\lambda_2(t)}{\sqrt{\lambda_1^2(t) + \lambda_2^2(t)}} - \frac{w_v\lambda_2(t)}{\lambda_1^2(t) + \lambda_2^2(t)}$$
$$\dot{\lambda}_1(t) = 2w_d(x(t) - x_{tar}(t)) - \sum_{i=1}^{m} \frac{2w_b(x(t) - x^i_{obs}(t))}{(x(t) - x^i_{obs}(t))^2 + (y(t) - y^i_{obs}(t))^2 - d_i^2}$$
$$\dot{\lambda}_2(t) = 2w_d(y(t) - y_{tar}(t)) - \sum_{i=1}^{m} \frac{2w_b(y(t) - y^i_{obs}(t))}{(x(t) - x^i_{obs}(t))^2 + (y(t) - y^i_{obs}(t))^2 - d_i^2} \tag{5.8}$$

5.2.2 Boundary Value Problem and its Solution(s)

This type of simultaneous non-linear equations as in (5.8) usually do not have closed form solution. In this case, equation (5.8) has been solved numerically by using the *bvp4c* ODE solver of Matlab. A comprehensive survey [7] on different numerical methods warns that the selection of a numerical solver is crucial to generate a solution(s). Popular recipes 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 (5.8) 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 [8] offers better numerical stability and is used by the ODE solver adopted in this chapter. The computation proceeds in the following sequence:

i. 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 till t.

ii. 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 (5.1), where the final states of the target are estimated according to the previous step.

iii. Using the initial and approximate final states of the pursuer, (5.8) 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 [2], 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 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 [9] we have applied a polynomial estimation method to predict boundary values. References to evolution of trajectory from past knowledge also appears in [4]. Polynomial estimation using the Neville-Aitken [15] 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 given in Appendix A.

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, equation (5.8) can have as many solutions as there are distinct sets of admissible initial values for the augmented state vector. Recall, 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 (5.8) 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, 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, (5.8) will have a unique

solution in the said closed interval. A detailed discussion on the existence of solution(s) has been provided in Appendix B.

5.2.3 Logarithmic Penalty Function

The process of converting hard constraints (equality versions of equations (5.2) and (5.3)) into soft-constraints by replacing them with equivalent functions is not new. An excellent survey on conversion approaches is provided by [21]. 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 chapter, we have shown how the logarithmic barrier can be customized 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 minimization problem like $P_{integrated}$, 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 [10]. This can be explained with an example.

Let us consider, a real valued natural logarithm of an argument, μ, where, μ is the equality version of the path constraint, (5.2). The equality indicates the minimum safety margin 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 minimization of the compound cost, (5.4). 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 minimization of J_{int}. 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 minimization problem to generate a sequence of control variables that contradicts the violation of minimum safety margin. The relation between the value function of μ and the actions generated by the logarithmic barrier is summarized in Table 5.1

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

Table 5.1: Different values of μ and their meaning

The role played by logarithmic barrier on steering control of the vehicle includes maintaining a balance between excessive and insufficent lateral adjustment [1]. Automatic lane changes, imposing access restrictions on certain paths (for example, one way, no turn etc.) are some of the practical driving situations that can benefit from this formulation. Verscheure et al. [11] have also documented the rapid path tracking capabilities of logarithmic barrier formulations in time-optimal applications. In addition, we demonstrate a novel use of logathmic 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, $v(t) = v_{max}$. On the other hand, logrithmic barrier provides a multi-point violation indicator, $0 < v_{max} - v(t) < 1$ and has better chances of avoiding constraint violation.
The upper bound, v_{max} need not 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

[2]. 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 [3], 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 a safe naigation route in dynamic and cluttered environments.

The features depicting the merits of the logarithmic barrier function have been summarized 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.

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 minimizing the pursuer's velocity and thereby also minimizes control effort.

5.3 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 analyzed in the context of a variety of autonomous driving scenarios, that have been recognized and regulated by competent authorities. Comparison with strategically different benchmark techniques further explain the advantages of the optimal formulation presented in this paper.

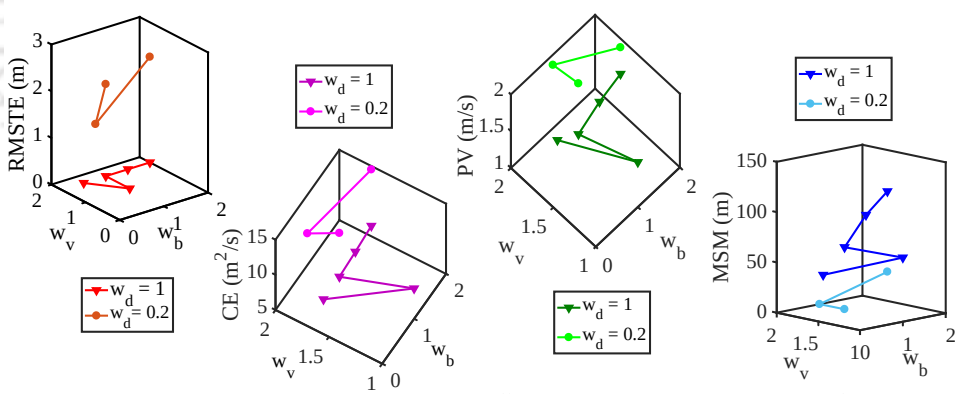

Figure 5.1: Effects of weighing parameters on the performance and safety metrics – a guideline to weight selection.

5.3.1 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/control-effort respectively. Individual impacts of these parameters on a particular sub-task 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.

1. A higher w_d reduces the root mean square (RMS) tracking error, but at the cost of a high control effort (CE). It has a supressive effect on the penalty functions.

2. A higher w_v reduces both control effort and the peak velocity (PV) attained, but relaxes the minimum safety margin (MSM) from the obstacle(s).

3. Increasing w_b helps maintaining a larger safety margin, but increases control effort.

4. 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 control effort.

Figure 5.1 gives a guideline about how the performance and safety metrics like RMS tracking error, control effort, peak velocity and minimum safety margin 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 minimum safety margin can be imposed by selecting weights such that $w < w_b$. In each planning iteration, the weights can be reassigned depending upon the estimated motion of the target and the obstacles.

5.3.2 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, piece of furniture etc.) 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 mid point of the workspace, $(0, 0)$m to the position $(-1.56, 0.79)$m. A static obstacle of radius 0.2m 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 5s . The tracking weight, w_d is maintained at a low priority of 0.2 and a higher priority 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 Figure 5.2(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 1.1m/s. Whereas, the black curve increases the steering weight, w_b to 0.8. However, the lowest RMS tracking error (0.27m) is achieved by steering control (black curve) (see Figure 5.2(b)), while the lowest control effort ($1.92m^2/s$) is achieved (see Figure 5.2(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 [2] conclude that 60% of collisions are accounted for transverse and head on approach angles.

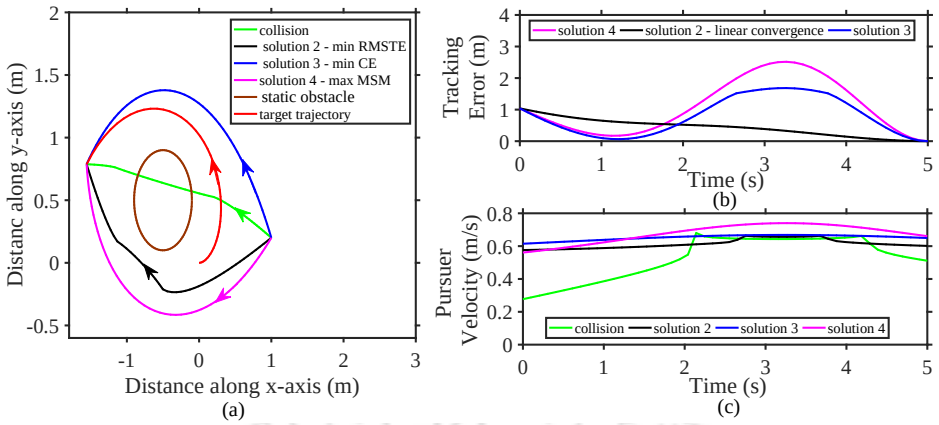

Figure 5.2: Steering and velocity control – how to select design parameters in avoiding a static obstacle.

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

(a) In the first case, an obstacle starts moving from $(7, 0)$m wih a velocity of 2m/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 2m/s, which means the obstacle is always faster than the pursuer. Allowable minimum safety margin is 0.5m. 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 5.3(a) shows that, low steering weight $w_b = 0.2$ and high velocity barrier, $v_{max} = 1.8$m/s, causes the pursuer to collide with the obstacle (red curve; assume, 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 1.3m/s. Simulation shows that steering action yields a better safety score (MSM=0.96m, blue curve) than velocity control (MSM=0.73m, magenta curve), but at the cost of a higher control effort 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 Figure 5.3(b).

(b) In the second case, the simulated obstacle starts moving from $(6, 6)$m wih a

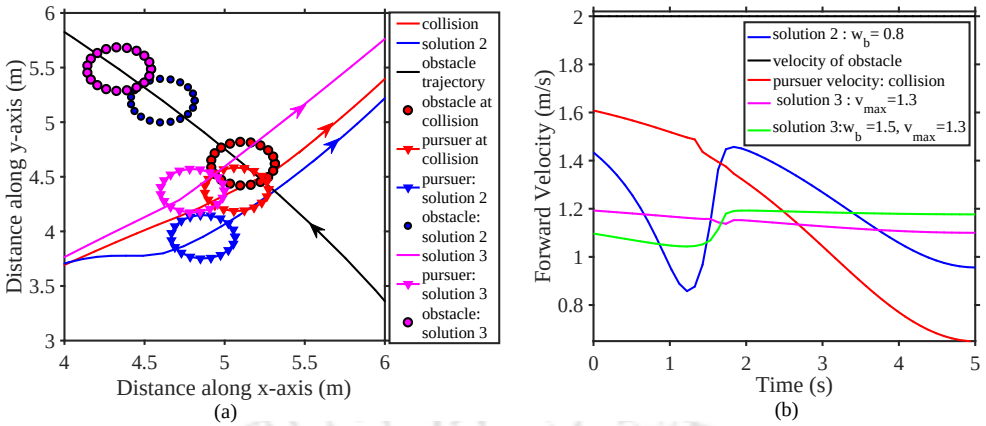

Figure 5.3: Steering and velocity control – how to avoid a fast moving obstacle crossing transversely.

velocity of 0.2m/s along 225°, making an almost head-on approach towards the pursuer. Allowable MSM is 0.5m. The weights, w_d and w_v have been maintained at 0.2 and 1 respectively. A velocity barrier of 2m/s results in a head-on collision (red curve, see Figure 5.4(a)), irrespective of the steering weight. Simulations have been conducted to study whether a reduction in velocity barrier is sufficient to avoid the collision.

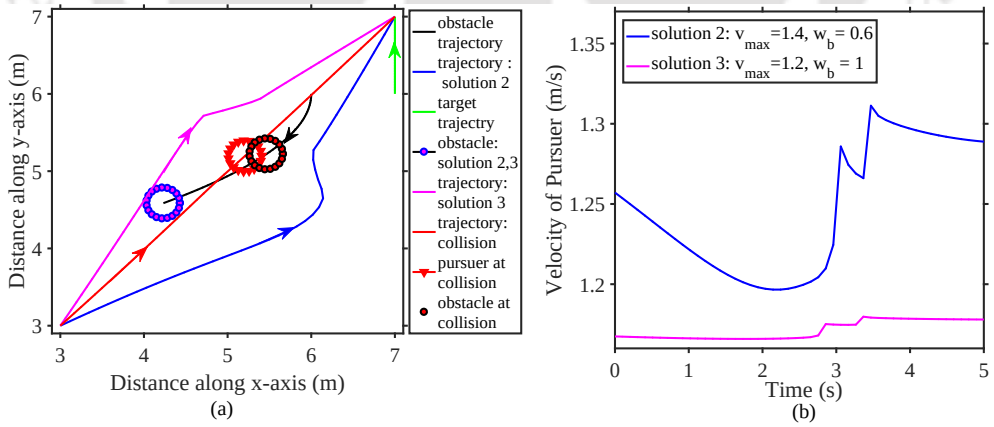

Figure 5.4: Steering and velocity control – how to avoid a head-on collision.

Figure 5.4(b) shows that, magnitude of velocity barrier and steering weight bear an inverse relationship in obtaining a desired solution. If velocity barrier is

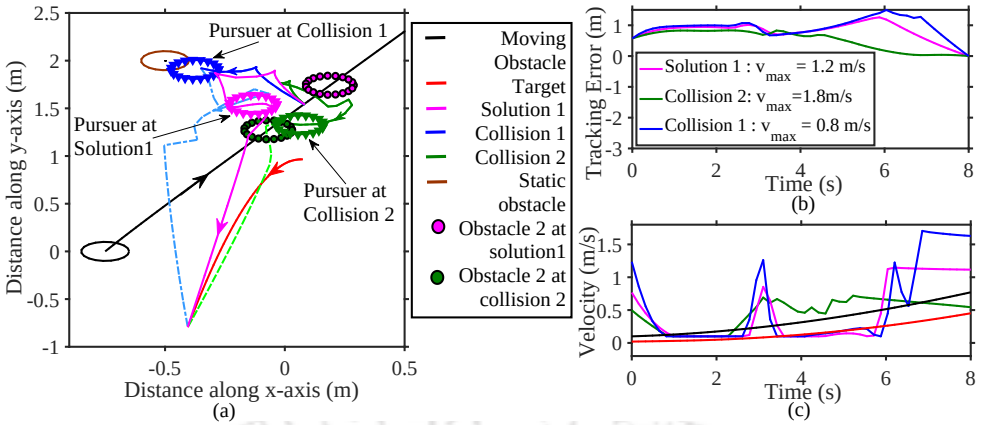

Figure 5.5: Velocity control –A slow down approach.

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 1.418m and a control effort of $8m^2/s$ over the mission interval. But, if the velocity barrier is reduced further to 1.2m/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 (1.46m) and a slightly lower control effort ($7.01m^2/s$). In situations like automatic lane changing and passing over a slower preceding vehicle, the foregoing framework can be applied, provided the exact parameter values are selected depending on a case-to-case basis.

Statistical reports on automated driving [1] show that the leading causes of crash are unanticipated turning of vehicles at non-signalized 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 upon an estimation of the obstacle's motion (intent inference) in order to achieve safe navigation.

Varying the Velocity Barrier–*Example 3:*

(a) In Figure 5.5, an accelerating target is assumed to move from $(0.07, 0.96)$m.

The pursuer, which was originally travelling with a velocity barrier of 1.5m/s starts tracking the target from $(0.08, 1.54)$m and attempts to intercept after 8s. 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 and retain the current lane (or one close to the current lane). In accordance, the steering weights to the static and the moving obstacles are maintained at 0.7 and 1.2 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. Minimum safety margin is 0.2m.

The current velocity barrier results in a collision. Note, in Figure 5.5(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 0.8m/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 Figure 5.5(b), Figure 5.5(c) and Table 5.2 (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 control effort. 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 change of speed limit allowance are some of the practical situations that represent the current simulation.

(b) Consider a case, where the pursuer encounters two moving obstacles, while

VB(m/s)	**RMSTE**(m)	**CE**(m^2/s)	**MSM-s**(m)	**MSM-m**(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 5.2: Performance metrics for different velocity barriers in *Example 3a*
VB: Velocity Barrier RMSTE: Root Mean Square Tracking Error MSM-s: Minimum Safety Margin-static MSM-m: Minimum Safety Margin-moving

tracking an accelerating target that moves from $(-0.25, 0.08)$m with an initial velocity of 0.36m/s along a complex trajectory. Time-to-interception is 10s. 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 velcity of 0.47m/s from a position $(-0.5, 0.2)$. The pursuer encounters a second obstacle, that moves linearly along the x-axis from $(0.55, 2.39)$m. Minimum safety margin is set to 0.2m. 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 5.3, which mean penalty to tracking and velocity constraint are given full weightage. (Notations for Table 5.3 – MSM-i: Minimum Safety Margin w.r.t. obstacle i, w_b^i: Steering weight to obstacle i.)

Referring to Table 5.3, 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 prioritized 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 1.8m/s to avoid the faster and nearer obstacle and setting a

Config.	RMSTE(m)	CE(m^2/s)	MSM-1(m)	MSM-2(m)	PV(m/s)
1	0.42	5.5	0.77	1.09	1.01
2	0.23	6	0.51	0.43	1.35
3	0.18	8.16	0.63	0.68	1
4	0.31	8.23	**0.10**	0.94	1.43
5	0.36	6.13	0.34	0.91	1.04
6	0.54	4.05	0.82	0.93	0.74
7	0.48	4.05	**0.12**	**0.10**	0.92

Table 5.3: Guidelines for selecting velocity barrier and weights for dynamic navigation –*Example 3b*

RMSTE: Root Mean Square Tracking Error CE: Control Effort MSM-1: Minimum Safety Margin-Obstacle1 MSM-2: Minimum Safety Margin-Obstacle2 PV: Peak Velocity

Configurations:

1: $v_{max} = 1.2$m/s, $w_b^1 = 2$,$w_b^2 = 0.2$ 2: $v_{max} = 1.2$m/s, $w_b^1 = 0.2$,$w_b^2 = 2$

3: $v_{max} = 1.8$m/s, $w_b^1 = 0.2$,$w_b^2 = 2$ 4: $v_{max} = 1.8$m/s, $w_b^1 = 1$,$w_b^2 = 1$

5: $v_{max} = 1.2$m/s, $w_b^1 = 1$,$w_b^2 = 1$ 6: $v_{max} = 0.8$m/s, $w_b^1 = 1$,$w_b^2 = 1$

7: $v_{max} = 1.2$m/s, $w_b^1 = 0.2$,$w_b^2 = 0.2$

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 control effort. 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 control effort, 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 summarizes the foregoing discussions on velocity and steering control.

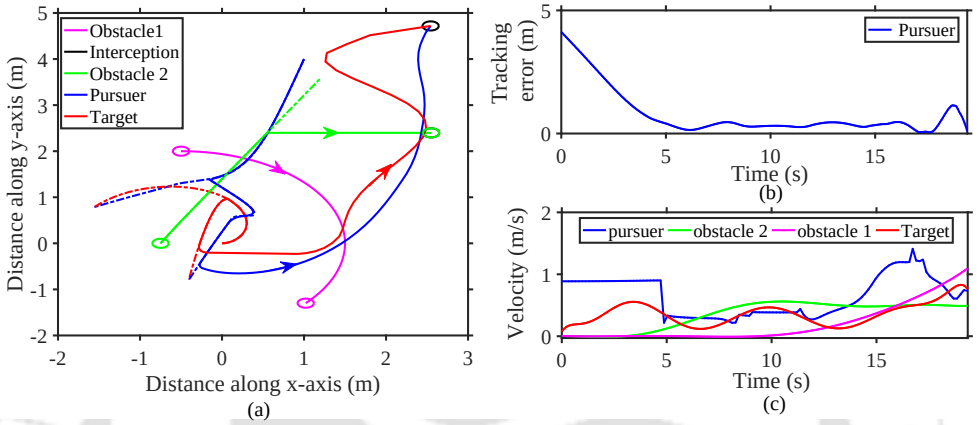

Figure 5.6: A shrinking horizon approach to trajectory planning and control in a complex dynamic scenario.

Figure 5.6(a) represents a multiphased 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 0.34m/s. A single static obstacle at $(-0.5, 2)$m is avoided by regulating the steering weight to 0.3 and maintaining a moderate velocity barrier of 1m/s.

After 2.96s, the change in target trajectory is estimated by constructing and extrapolating a polynomial. A 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 0.5m. 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 0.36m/s and accelerates further), while the second obstacle slows down, changes direction and maintains a constant speed of 0.7m/s alng the x-axis. The magnitude of velocity barrier is increased to 1.8m/s and an unequal steering weightage is assigned to avoid the two moving obstacles (configuration 3 of Table 5.3). Finally, interception occurs at 19.36s. The tracking error (see Figure 5.6(b)) and the velocity profiles of the pursuer, target and the obstacles can be seen in Figure 5.6(c) for a better understanding of the above description.

5.3.3 Comparison With Other Methods

The effectiveness of the proposed method against other techiques 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 [12], 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 [13], which shares a non-aggressive avoidance like the proposed strategy. The third technique is rendezvous guidance (RG) [14], 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 0.2m/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. The velocity barrier of the pursuer has been set to $v_{max} = 2$m/s. An obstacle starts moving at an initial velocity of 0.2m/s and an initial acceleration of $0.1m/s^2$ from the position, $(5, 5)$m. Minimum safety margin is set to 0.5m.

Figure 5.7(a) demonstrates the trajectories planned by the different methods compared here. While all of the methods are found to satisfy the minimum safety margin constraint, the oscillations in the Artificial Potential Field (APF) driven path (blue) is noteworthy. Undesired noise related to artificial potential field is

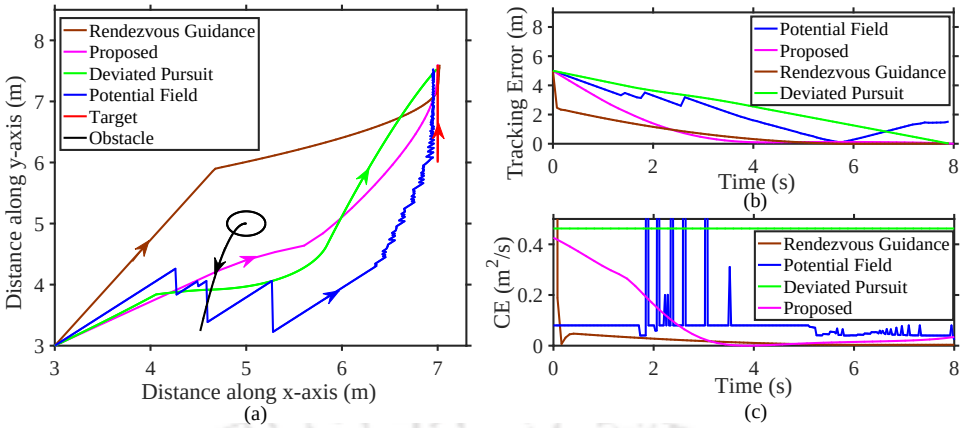

Figure 5.7: A comparative study of the proposed method against strategically different benchmark techniques.

also observed in Figure 5.7(b) and Figure 5.7(c), which represent the tracking error and control effort over the mission horizon. Whereas, Rendezvous Guidance (RG) generates a smooth trajectory (brown)(see Figure 5.7(a)), achieving a faster rate of convergence of tracking error (see Figure 5.7(b)), but at the cost of a very high control effort, as can be seen in Figure 5.7(c). In contrast, a Deviated Pursuit (DV) method results in a trajectory (green) with slowest error convergence curve and a constant control effort, which is considerably higher that the average control effort for the other methods. Our proposed method computes a trajectory (magenta) with optimal steering (see Figure 5.7(a)), which has a fast tracking error convergence rate, numerically comparable to that obtained by Rendezvous Guidance (see Figure 5.7(b)) and requires a control effort that has a lower peak value than most of the other methods and a fast decaying profile (Figure 5.7(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 [6], but uses a quadratic optimization formulation in contrast to our proposed logarithmic barrier. This reference method shall be termed as Integrated Control based Quadratic Optimization ($ICQO$) in further analysis. Simulation and experimental validation of both the methods have been shown to demonstrate the advantages of the logarithmic penalty function applied to both

path and control constraints.

Experimental Example: 1 A target is commanded to move along the x-axis from $(0, 0)$m with a constant velocity of 0.1m/s in a laboratory set-up. The interception time, T_{int} is chosen as 20s, so that the experiment can be contained within the field of view (2.5m $\times$ 2.5m) of an overhead camera. An obstacle was commanded to travel from $(0.3, 1)$m with a velocity of 0.05m/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 1.8m/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 0.1m. The pursuer had a radius of about 0.22m. Minimum safety margin has been set to 0.4m. 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 0.8s. Sampling time is 400ms and the maximum idle time (time to reset the communication link due to inactivity) is 1.2s (see client-server algorithm in Appendix C). The range of weighing constants used for the experiment are, $0.8 \leq w_v \leq 1.2$, $0.01 \leq w_b \leq 0.04$ and $0.1 \leq w_d \leq 0.2$. The pursuer on reaching a δ_{int} ball of 0.4m around the target in a neighbourhood of T_{int} has been assumed to conclude interception and tracking is 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 0.4m (same as minimum safety margin specification) from the target. As observed in Figure 5.8(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 (Figure 5.8(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 control effort (see

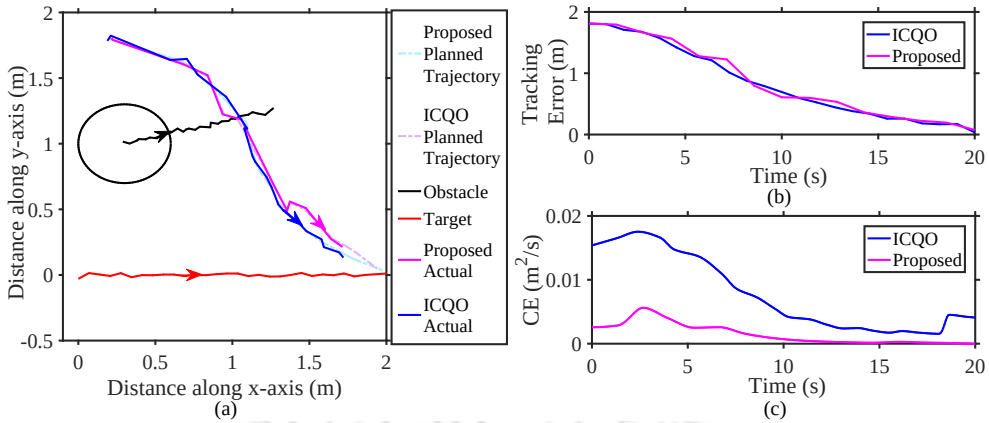

Figure 5.8: Comparison of trajectory, tracking error and control effort of proposed method with ICQO [6].

Figure 5.8(c)) in achieving a similar tracking error score. It will be further clear from the acceleration (Figure 5.9(c)) and angular velocity (Figure 5.9(d)) profiles of the pursuer illustrated in Figure 5.9, 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 find-

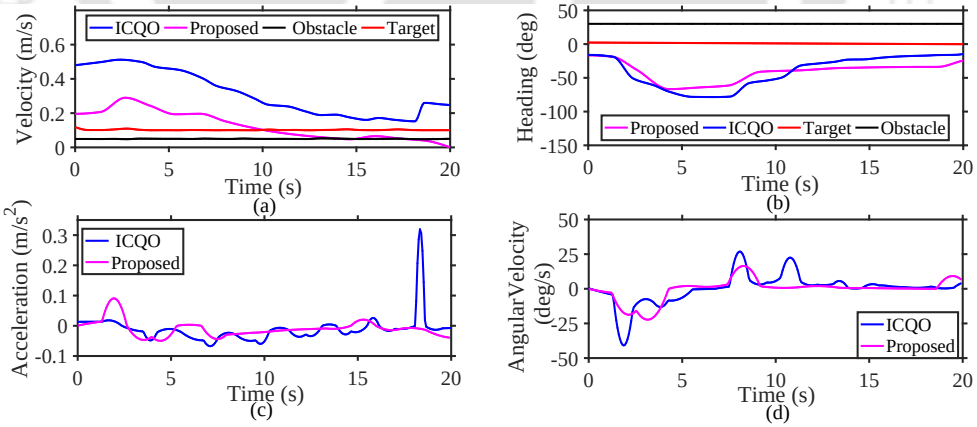

Figure 5.9: Comparison of velocity, heading, acceleration and angular velocity of proposed method with ICQO [6].

ing 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 [17] a hybrid global planning strategy has been devised in conjunction with Jump-A^* algorithm whereas [18] discusses an approach that involves estimation of distribution of obstacles for generating quick responses. Proposed 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 [16]. However, in contrast to the proposed optimal trajectory with unknown target motion, the target considered in [16] has predefined states and in 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 [16] has been used. But the proposed method performs obstacle avoidance by minimizing optimal cost utilizing non-quadratic penalty which does not require addressing constraints like non-intersecting obstacle boundaries, angle based spirals, and parametric path curvature illustrated in [16]. In general, the proposed method performs functionalities of both reactive (local) and model predictive optimal 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 [19], 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.

5.4 Summary of Chapter 5

In this Chapter, we have proposed an intent-aware dynamic navigation strategy using integrated controller. The navigation planner is configured to perform track-

ing and collision avoidance over the entire planning horizon, which is repeated in each planning window. The planning window shrinks with time until interception with the moving target is achieved. Target and obtacle motions are unknown and estimated at a time corresponding to the end of the current planning window in order to derive intent of motion. The integrated control policy applies a logarithmic penalty to control and path inequality constraints corresponding to velocity and lane restrictions and as a result, peforms a passive collision avoidance. Navigation is free from switching noise and helps to quantify run-time cost effectiveness of various parameters like control effort, steerability, velocity control, safety score and the likes in addition to obtaining smoother trajectories.

Bibliography

[1] 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), Washington, DC: National Highway Traffic Safety Administration.* (2018, August).

[2] 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), Washington, DC: National Highway Traffic Safety Administration.* (2016, August).

[3] 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).

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

[5] 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).

[6] B. Li, Y. Ouyang, Y. Zhang, T. Acarman, Q. Kong and Z. Shao, "Optimal cooperative maneuver planning for multiple nonholo-

nomic robots in a tiny environment via adaptive-scaling constrained optimization," *IEEE Robot. and Autom. Lett.*, Preprint (2021).

[7] JT. Betts, "Survey of numerical methods for trajectory optimization," *Journal of Guidance, Control and Dynamics* **21**(2), 193–207 (1998).

[8] J. Kierzenka and LF. Shampine, "A BVP solver based on residual control and the Maltab PSE," *ACM Transactions on Mathematical Software* **27**(3), 299–316 (2001).

[9] 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 Robotics And Automation Letters* **2**(3), 1696–1703 (2017).

[10] RM. Freund, "Penalty and barrier methods for constrained optimization," *15.084J Nonlinear Programming*, Massachusetts Institute of Technology: MIT OpenCourseWare (2004).

[11] D. Verscheure, M. Diehl, J. De Schutter and J. Swevers, "Recursive log-barrier method for on-line time-optimal robot path tracking" **In**:*Proceedings of ACC'09: American Control Conference*, 4134–4140 (2009).

[12] L. Huang, "Velocity planning for a mobile robot to track a moving target - A potential field approach," *Journal of Robotics and Autonomous Systems, Elsevier* **57**(2), 55–63 (2009).

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

[14] F. Kunwar and B. Benhabib, "Rendezvous-guidance trajectory planning for robotic dynamic obstacle avoidance and interception,"

IEEE Transactions on Systems, Man, Cybernetics, Part B: Cybernetics **36**(6), 1432–1441 (2006).

[15] WH. Press, SA. Teukolsky, WT. Vetterling and BP. Flannery, "Numerical Recipes in C : The Art of Scientific Computing," Cambridge University Press (2002).

[16] 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," *Robotics and Autonomous Systems* **60**(6), 769–788 (2012).

[17] L. Liu et al., "Global Dynamic Path Planning Fusion Algorithm Combining Jump-A* Algorithm and Dynamic Window Approach," *IEEE Access* **9**, 19632–19638 (2021).

[18] 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 Transactions on Industrial Electronics* **68**(10), 9998–10006 (2020).

[19] Y. Kuwata, M.T. Wolf, D. Zarzhitsky and T.L. Huntsberger, "Safe Maritime Autonomous Navigation With COLREGS, Using Velocity Obstacles," *IEEE Journal of Oceanic Engineering* **39**(1), 110–119 (2014).

[20] ER. Pinch, "Optimal Control and the Calculus of Variations," Oxford University Press, Oxford, New York (1993).

[21] ML. Bell, and R.W.H. Sargent, "Optimal control of inequality constrained DAE Systems," *Elsevier Computers And Chemical Engineering* **24**(11), 2385–2404 (2000).

[22] H. Huang, A.V. Savkin and C. Huang, "Reliable Path Planning for Drone Delivery Using a Stochastic Time-Dependent Public Trans-

portation Network," *IEEE Transactions on Intelligent Transportation Systems* **22**(8), 4941—4950 (2021).

[23] M. Kamezaki, Y. Tsuburaya, T. Kanada, M. Hirayama and S. Sugano, "Reactive, Proactive, and Inducible Proximal Crowd Robot Navigation Method Based on Inducible Social Force Model," *IEEE Robotics and Automation Letters* **7**(2), 3922–3929 (2022).

Chapter 6

Stable Tracking Controller with Accurate State Feedback

6.1 Foreword

Optimal trajectories designed in Chapter 3, 4 and 5 are desired trajectories that act as reference for navigation and guidance of a pursuer. The actual trajectory of the pursuer may deviate from the reference trajectory due to uncertainties, disturbances, noise and measurement errors. If the deviation is bounded, the pursuer may be guided back to the reference optimal path by obtaining periodic state feedback. An asymptotically stable local tracking controller has been proposed in this chapter for executing the desired optimal trajectories in real-time. State measurements of the pursuer are usually performed based on odometry. However, certain situations like wheel slippage may make the wheel encoders unreliable for state measurement. Eventually, the state feedback will be erroneous and optimal trajectories generated in the subsequent iterations may fail to achieve interception with the target. In such a scenario, an observer based state and slip estimation techique has been proposed in this chapter, which is capable to generate accurate state feedback in presence of wheel slippage disturbance and helps to retain the existing tracking controller. Investigation of wheel slippage due to kinematics and proposal of a modified slip-kinematic model are novel aspects of this chapter.

6.2 Closed-loop Tracking Controller

In the previous chapters, we have discussed the methods for generating the desired optimal trajectories. Decoupled optimal trajectories have been designd in Chapter 3 and Chapter 4, whereas, integrated optimal trajectories have been formulated in Chapter 5 as part of navigation planning. In order to execute the planned reference trajectoies in real-time a stable local control action needs to be devised that converges the actual trajectory to the reference trajectory in a given time (both finite and asymptotic convergence rates may be considered). In order to account for the dynamicity of motion, the designed reference trajectories require being updated at regular intervals. The control and planning horizon can both be reassigned at the end of each iteration. We have used a fixed-size control horizon but a variable planning horizon that shrinks in size as time progresses. In each planning horizon, a tracking controller has been assigned to measure the current actual (or estimated states in case of unreliable sensors) states and execute motion pertaining to the reference trajectory between two consecutive updates. We define local control variables for the tracking controller which drives the robot along the reference optimal path. The plan involving iterative computation of reference trajectories followed by the local tracking control also provides immunity against bounded deviations from the intended path.

To describe the mathematical formulation of the tracking controller, let us assign a suffix-'D' (desired) to the reference states and a suffix-'A' (actual) to the measured states. The error states have been defined as in equation (6.1), where, $x_{\hat{e}} = (x_A(t) - x_D(t))$ and $y_{\hat{e}} = (y_A(t) - y_D(t))$.

$$\begin{aligned} x_e(t) &= x_{\hat{e}} \sin(\theta_A(t)) - y_{\hat{e}} \cos(\theta_A(t)) \\ y_e(t) &= x_{\hat{e}} \cos(\theta_A(t)) + y_{\hat{e}} \sin(\theta_A(t)) \\ \theta_e(t) &= \theta_D(t) - \theta_A(t) \end{aligned} \tag{6.1}$$

Also, we define the error velocity as $v_e(t) = v_D(t) - v_A(t)$. The dynamical equations of the error states can now be written as in (6.2). The local control variables of the tracking controller are u_0 and u_1, defined as $u_0(t) = \dot{\theta}_D(t) - \dot{\theta}_A(t) =$

$\omega_D(t) - \omega_A(t)$ and $u_1(t) = v_A(t) - v_D(t)\cos(\theta_e(t))$.

$$\begin{aligned}
\dot{x}_e(t) &= y_e(t)(-u_0(t) + \omega_D(t)) + v_D(t)\sin(\theta_e(t)) \\
\dot{y}_e(t) &= u_1(t) - x_e(t)(-u_0(t) + \omega_D(t)) \\
\dot{\theta}_e(t) &= u_0(t)
\end{aligned} \tag{6.2}$$

The aim is to find a feedback control law for $u_0(t)$ and $u_1(t)$ such that the actual trajectories of the robot asymptotically converge to the desired reference. We assign the following control law given in equation (6.3).

$$\begin{aligned}
u_0(t) &= -\hat{m}\theta_e(t) - x_e(t)v_D(t)\frac{\sin(\theta_e(t))}{\theta_e(t)} \\
u_1(t) &= -\hat{m}y_e(t)
\end{aligned} \tag{6.3}$$

The design variable, $\hat{m}$ is a positive constant, which controls the rate of convergence. Let us now define a Lyapunov function candidate of the form $V = \frac{1}{2}\big(x_e^2(t) + y_e^2(t) + \theta_e^2(t)\big)$. Taking the derivative of V with respect to time along the trajectories of the closed-loop system (6.1) and applying the feedback control law defined in (6.3), we get equation (6.4).

$$\dot{V} = -\hat{m}(y_e^2(t) + \theta_e^2(t)) \leq 0 \tag{6.4}$$

Note that, $\lim_{\theta_e(t)\to 0}\frac{sin(\theta_e(t))}{\theta_e(t)} = 1$. From equation (6.4) it can be inferred that $y_e(t)$, $\theta_e(t)$ and also $x_e(t)$ are bounded functions. We can further verify that the second derivative (see equation (6.5)) of the Lyapunov function is also bounded.

$$\ddot{V} = -2\hat{m}\big(-\hat{m}(y_e^2(t)) - x_e(t)y_e(t)(-u_0(t) + \omega_D(t)) - \theta_e(t)u_0(t)\big) \tag{6.5}$$

Therefore, $\dot{V}$ is uniformly continuous. Hence, using Barbalat's lemma [3], we conclude that $\dot{V} \to 0$ as $t \to \infty$, which further implies $y_e(t) \to 0$, $\theta_e(t) \to 0$ and also $x_e(t) \to 0$ as $t \to \infty$. It may be noted that the very definition of $u_1(t)$ implies $v_e(t) \to 0$ as $\theta_e(t) \to 0$. This proves that the tracking controller is asymptotically stable under the chosen control law.

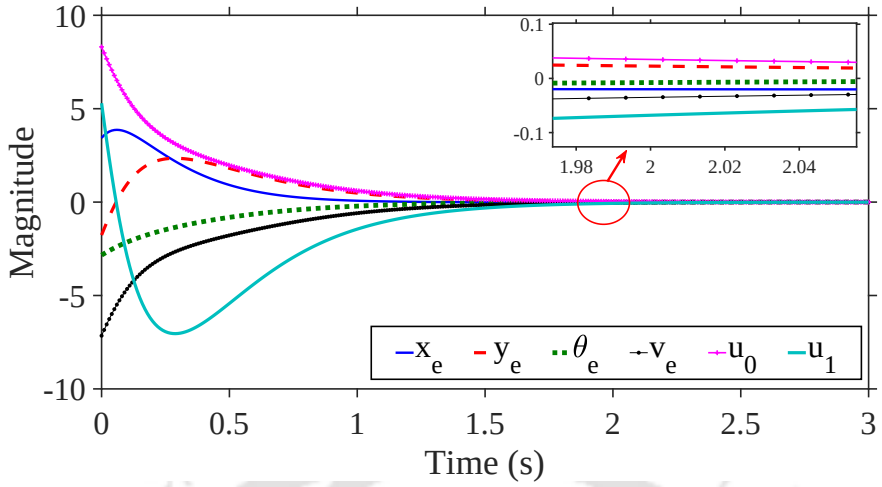

Figure 6.1: Tracking controller guarantees asymptotic convergence of states and controls. $\hat{m}$ determines the rate of convergence. Blue, red and green are state error curves; magenta and teal are local control signals.

6.2.1 Stability of Feedback Tracking Control

Convergence of the tracking controller has been shown in Figure 6.1. This example shows the evolution of the state and control errors over an interval of 3s, for a section of the trajectory. The convergence-rate control parameter, $\hat{m}$ has been chosen as 3. A higher value of $\hat{m}$ speeds up the convergence, but this comes at the cost of a higher initial magnitude of the local control variables. Therefore, the choice of $\hat{m}$ depends on the saturation limits of the vehicle. Ideally, the tracking control is supposed to exhibit asymptotic convergence. Simulation (Figure 6.1) shows, that the tracking errors fall below 4% of the initial magnitude within 66% of the chosen interval of the planning horizon. Tracking controllers of other forms will also be applicable for the designed optimal trajectories. The objective here is only to validate the idea that a local tracking controller is capable of executing the desired optimal trajectories with real-time state feedback. Advanced features like finite time convergence [2] and improved transient response [3] may also be incorporated in the controller design.

6.3 Wheel-Slip as Disturbance

Slippage occurs when the patch of tire in contact with the ground is subjected to dynamic forces, that act in directions other than forward. As a result, the actual trajectory deviates from the intended path. As lateral forces develop at the contact patch, *pure rolling* motion is compromised [4] and the wheel encoders fail to record this deviation, thereby giving an incorrect localization response. Any control action taken upon this 'incorrect' pose feedback can lead to serious errors.

Instead of treating longitudinal and lateral slip in each wheel independently [5], we consider an effective slip-angle and slip-ratio about the geometric center of the vehicle and maintain this definition throughout the entire discussion.

$$\begin{aligned} \alpha &= \theta_{commanded} - \theta_{observed} \\ \sigma &= \frac{|v_{encoder} - v_{observed}|}{\max\{v_{encoder}, v_{observed}\}} \end{aligned} \tag{6.6}$$

Slippage can be quantified by two parameters. The first one α, in equation (6.6) is the lateral deviation of heading from the commanded direction. The second parameter, σ (see (6.6)), is defined as the ratio of forward velocity error between the data recorded by wheel encoders and the observation. The angular shift, α is called the slip-angle and σ is known as the slip-ratio. The mobile vehicle is assumed to have no backward motion. Therefore, σ is a positive, dimensionless variable ranging between 0 and 1. A perfect non-holonomic motion corresponds to $\sigma = 0$, indicating a complete agreement between the odometry and observed pose. Conversely, $\sigma = 1$ indicates a complete disagreement implying either a *skid* or a *false-rolling* condition, depending on whichever is greater between the driving torque and traction.

6.4 Observer Design

The objective is to validate the concept of on-line slip estimation through observation and thereby to advocate in favour of a modified vehicle kinematics that can accommodate slippage. For a fair understanding of the proposed approach, we have investigated the effects of both linear [7] and nonlinear observers [9, 11].

6.4.1 Linear Observer

The system to be observed can be explained with the help of equation (6.7), where (A, B) is the system and input matrix pair, q is the state vector, u is the input vector and n is Gaussian process noise.

$$\dot{q} = Aq + Bu + n \tag{6.7}$$

where,

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, q = \begin{bmatrix} p \\ v \\ \theta \end{bmatrix}, u = \begin{bmatrix} a - c_a \\ \omega - c_\omega \end{bmatrix} \tag{6.8}$$

Here the states are, forward travelled distance: p, forward velocity: v and heading: θ. The input pair, (a, ω) are the linear acceleration along the forward direction and angular velocity about the center of mass of the vehicle. The parameters, c_a and c_ω are the accelerometer and gyroscope biases having Gaussian distribution. Assuming there is no absolute-position sensor and dedicated velocity sensor like GPS/GNSS and tachometer, we have used optical flow from an overhead camera for observation. In absence of expensive commercial sensors, techniques and utility of deploying low-cost vision sensor in velocity-estimation of a vehicle have been explained in [12]. The output z, is related to the state vector q (see (6.9)) through the observation matrix C (identity matrix of rank 3) and is supposed to represent the actual trajectories. Measurement noise m is assumed Gaussian.

$$z = Cq + m \tag{6.9}$$

The system (6.7) defined above is observable and pertaining to (6.9) a generalized state equation for a Luenberger observer [6] can be defined as in (6.10).

$$\dot{\hat{q}} = A\hat{q} + Bu + L(z - \hat{z}) \tag{6.10}$$

The pair, $(\hat{q}, \hat{z})$ denote the estimated state and output vectors and L represents the observer gain matrix of full rank. The observer error, e is a difference vector of the actual and the estimated states and is represented by $q - \hat{q}$. It may be recalled that, the elements of the observer gain matrix, L can be determined independent of the tracking controller (any controller which drives the vehicle along a reference trajectory) by virtue of the 'Separation principle'. For the same reason, the observer error can be designed to converge asymptotically by selective pole placement of the matrix $(A - LC)$, irrespective of the controller design, $(A - BK)$ (K being the controller gain). Using Ackerman's method or otherwise, the elements of L can be placed such that the eigenvalues of the closed loop observer are located in the negative half of the *s-plane*. However, it may be noted that L may not be unique in this case. In the light of the correlated observation gain designed for a non-linear observer in [11], the matrix $(A-LC)$ is block-diagonal at best. This is subject to an assumption that there exists little or no correlation between heading and forward travelled distance, and heading and forward velocity. Thereby, the observer error dynamics can be rearranged into equation (6.11), which guarantees global asymptotic stability to the designed observer. Convergence proof of Luenberger observer of similar construction can be found in [14] in addition to pole-placement methods discussed in [13].

$$\dot{e} = (A - LC)e \tag{6.11}$$

The process and measurement noises have no effect on the designed observer. In equation (6.11), $(A-LC)$ is a 3×3 matrix. Since odometry cannot be trusted in presence of slippage, the states must be estimated from additional observations, like a camera in this case. Optical flow, most certainly, does not give a very accurate measurement due to its sensitivity to light, processing delay and low sampling

rate in response to high-speed motion segments of short duration. If available, it is recommended to use a sensor network for facilitating accurate observation over a distributed space comprising of multiple mobile units. A combination of optical and proximity sensors would be a preferable choice for fast and accurate localization [18]. Unified sensing using onboard sensors and beacons have recently been found to yield good results in state estimation in presence of uncertainties [10]. While high speed motion capture cameras are best suited to this case, with proper static and dynamic calibration prior to the experiments, satisfactory performance can be achieved with a low-cost, moderately efficient vision sensor [16] paired with advanced image recognition techniques.

6.4.2 Nonlinear Observer

In order to strengthen our investigations, we have also studied an *Extended Kalman Filter* (EKF) assuming the that the system to be observed is governed by the non-linear kinematics given in equation (6.12), where k is the sampling instant and dt is the sampling time.

$$\begin{aligned} q_k &= f_k(q_{k-1}, u_{k-1}, n_{k-1}) \\ \begin{bmatrix} x_k \\ y_k \\ v_k \\ \theta_k \end{bmatrix} &= \begin{bmatrix} x_{k-1} + v_{k-1}\cos(\theta_{k-1})dt \\ y_{k-1} + v_{k-1}\sin(\theta_{k-1})dt \\ v_{k-1} + (a_k - c_a)dt \\ \theta_{k-1} + (\omega_k - c_\omega)dt \end{bmatrix} + \begin{bmatrix} n_{k-1} \end{bmatrix} \end{aligned} \tag{6.12}$$

The states are, Cartesian positions: x and y, forward linear velocity: v and heading: θ and n is the vector of zero-mean Gaussian white state-noises. The observation is given by $z_k = Hq_k + m_k$, where the observation matrix, H is basically an identity matrix of rank 4 and m is a zero-mean Gaussian measurement-noise vector. The apriori estimation covariance matrix is initialized by identity matrix and we assume that $det(R) < det(Q)$, where, Q and R are the process and measurement covariance matrices. The filter equations for prediction and correction are given in (6.13). Predictions are driven by odometry while corrections are

driven by optical flow.

$$\begin{aligned}
\text{Prediction:} \quad & q_{k|k-1} = f(q_{k-1|k-1}, u_{k-1}) \\
& P_{k|k-1} = J_{k|k-1} P_{k-1|k-1} J_{k|k-1}^{\top} + \bar{Q} \\
\text{Kalman Gain:} \quad & K_k = P_{k|k-1} H_k^{\top} (H_k P_{k|k-1} H_k^{\top} + \bar{R})^{-1} \\
\text{Correction:} \quad & q_{k|k} = q_{k|k-1} + K_k (\bar{z}_k - z_k)^{\top} \\
& P_{k|k} = [I - K_k H_k] P_{k|k-1}
\end{aligned} \tag{6.13}$$

The elements $\bar{Q}$ and $\bar{R}$ represent the noise terms $n_k Q_{k-1} n_k^{\top}$ and $m_k R_k m_k^{\top}$ respectively. The Jacobian, $J = \frac{\partial f}{\partial q}$ locally linearizes the state model. The estimated (prior) and corrected (posterior) state covariances are expressed as $P_{k|k-1}$ and $P_{k|k}$ respectively. The Kalman gain at the k^{th} instant is given as K_k and $\bar{z}_k$ is the observed state from the camera.

The acceleration and angular velocity inputs are obtained from a 6 DOF inertial measurement unit (IMU) attached to the vehicle. However, the deviation of inertial measurement builds up incremental errors over time and need to be updated regularly. In absence of a yaw-rate measure (magnetometer), heading provided by the optical flow has been assumed to be the reference for the update. For velocity estimation, the observer uses both integration and differentiation, which represent the core operations of the IMU and the camera and are performed simultaneously. These complementary operations cancel out the lead and lag errors introduced by them [7].

Ideally, the accelerometer and the gyro biases also need to be estimated online. In contrast to the complicated bias models presented in [7], we have assumed a 'random walk', inspired by the bias calculation for a state partitioning particle filter in [8]. The time-varying bias terms, $\hat{c}_a$ and $\hat{c}_\omega$ are effectively zero-mean Gaussian white noises. Stability and local convergence properties of EKF in the role of an observer for deterministic nonlinear dynamics has been well established in existing literatures [15, 17].

6.5 Slip Propagation Model and Modified Kinematics

Following the pose and velocity estimation (by using a linear or a non-linear observer), the evolution of the expected slip-angle and the slip-ratio can be modeled according to the differential equations given in (6.14), which have been formulated by differentiating equation (6.6) and using estimated states instead of the observed states, wherever applicable. The $\pm$ signs in the ODE govering the slip-ratio are decided by the 'skid' and 'false-rolling' conditions (see Appendix B). Slip-dynamics illustrated in equation (6.14) can be represented by Euler discretization with a fixed sampling interval, wherein at each sampling instant, k, current estimated states can be obtained from the observer. Using the estimated state at k in the discretized version of equation (6.14), 'anticipated' discrete-time propagation models of slip-angle and slip-ratio can be computed for instants $(k+1)$ and beyond. At $(k+1)$, these anticipated values can be revised based on a new set of estimates (if available) provided by the observer. Expected slippage projected by the propagation models and the actually observed values computed from the camera and the wheel encoders have been found to be in close agreement (see Section 6.6).

$$\begin{aligned} \dot{\alpha}(t) &= \omega_{commanded}(t) - \dot{\hat{\theta}}(t) \\ \dot{\sigma}(t) &= (1 \pm \sigma(t)) \left(\frac{\dot{\hat{v}}(t)}{\hat{v}(t)} - \frac{\dot{v}_{encoder}(t)}{v_{encoder}(t)} \right) \end{aligned} \tag{6.14}$$

Using the current expected slip-angle as an input, a modified vehicle kinematics can be constructed according to equation (6.15). The longitudinal and the lateral components of velocity (due to slip disturbance) about the originally desired direction of motion are defined as $v_L = \hat{v}\cos\alpha$ and $v_\delta = \hat{v}\sin\alpha$ respectively. The estimated angular velocity, $\hat{\omega}$ can also be computed from the observer equation

(6.10) or (6.13).

$$\begin{aligned}\dot{x} =& v_L \cos(\theta_{commanded}) - v_\delta \sin(\theta_{commanded}) \\ \dot{y} =& v_L \sin(\theta_{commanded}) + v_\delta \cos(\theta_{commanded}) \\ \dot{\theta} =& \hat{\omega}\end{aligned} \tag{6.15}$$

A discrete-time linearization of (6.15) can now correctly predict the reachable state in the next instant. It may again be recalled that the proposed method aims to facilitate an accurate estimation of state in presence of slip-disturbance and has no direct bearing with the controller used or the quality of measurement.

6.6 Experiments and Simulations

In this section we shall discuss how the proposed method works, with the help of different navigation scenarios. We have selected four case studies, which are prone to undergo wheel slippage. Observation of the vehicle's states using optical flow has been considered to be a true representation of ground truth. For validation purpose we have used a single overhead camera monitoring a 3m $\times$ 3m arena. We have used a low-cost commercial grade camera with limitations on response time and frame rate. This has been done in an attempt to identify the lower limit of achievable performance in a tracking robot, as explained in Section 6.6.1 and a collision avoidance situation which will be explained further in Section 6.6.3. Efficiency of the 'estimation' process discussed in this chapter is not controlled by the quality of measurement. This means that a better vision sensor will yield measurements closer to ground truth, but does not imply a reduced estimation error.

The experiments relevant to wheel slippage estimation and state feedback correction have been implemented on Patrolbot using in-built differential wheel encoders and inertial measurement unit attached to the vehicle close to the base and at the centre of gravity of the vehicle (see Appendix C for details). A schematic representation of the interconnections between the key functional blocks of the

architecture has been illustrated in Figure 6.2, which highlights the role of the observer in the trajectory planning and control unit.

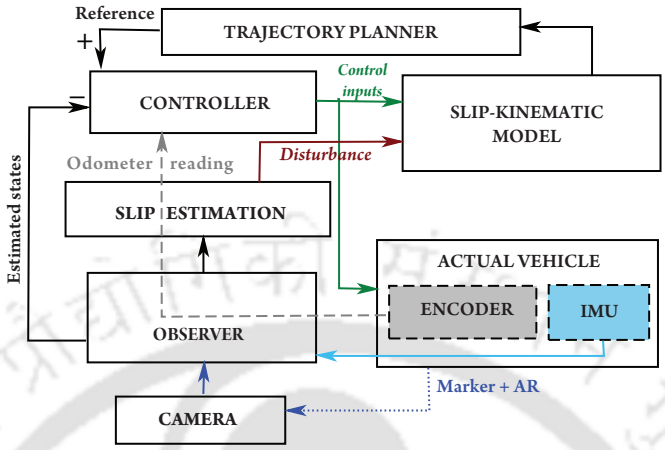

Figure 6.2: Data flow between the system, observer and controller.

The experimental trajectories have been designed to challenge the kinematic limits of the pursuer. The observation time has been kept short, typically within 3s, pertaining to the limitations imposed by the vision sensor's field of view. Accordingly, sudden acceleration, sudden deceleration, cornering and lane-change are the four types of trajectories that have been commanded to the robot. In the first two cases, the effect of suddden changes in linear acceleration and sudden braking have been studied. Ideally, slippage in these examples indicate contrasting phenomena – wheels rotating in-place without forward movement and motion due to inertia without actual rolling of the wheels, respectively. The third and the fourth cases deal with turning manoeuvres. A high speed turn and a series of quick sharp turns respectively represent situations like cornering and lane-change.

6.6.1 Case Study : Changes in Linear Velocity

Separate experiments have been designed for a tracking robot where – (i) the robot brakes due to an unanticipated and fast movement of an obstacle and (ii) the robot speeds up to avoid a slow-moving object. In the former situation (braking), the instantaneous drop in the driving torque stalls the motors, but the

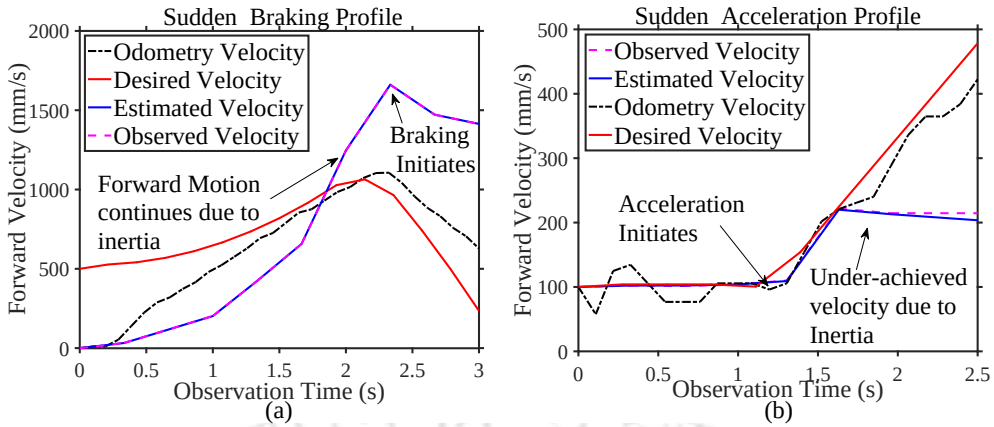

Figure 6.3: Contrasting effects of sudden braking and sudden acceleration gives rise to skidding and false-rolling. Estimated forward velocities depict the correct trajectories.

vehicle continues to move due to inertia. Whereas, the second event (acceleration) is followed by a 'phase', in which, the driving torque fails to exceed the tractive force, and the wheels rotate in-place without making an actual forward motion. In both cases, dead-reckoning by the wheel encoders infer incorrect results.

Figure 6.3 illustrates the mismatch between the velocities computed from odometry and optical flow in the said experiments, the reasons being explained before. In the first experiment (braking), the commanded velocity demonstrates a sharp decline between 2.2s and 3s. While the odometry seems to follow suit, the actual (observed) velocity is found to keep increasing for a few more seconds due to inertia and starts decreasing at a comparatively slower rate around 2.3s. In the second experiment (acceleration) the desired velocity involves a sharp rise between 1.5s and 2s, but the observed velocity fails to catch up with the desired trajectory. The under-achieved velocity profile is indicative of a false-rolling condition, contrary to the odometry readings. Figure 6.4(a) shows a comparison of path profiles due to sudden braking. The estimated time-evolution of the slip parameters, α and σ have been computed from the estimated states of the observer. The actual slip-ratio and slip-angle have been computed from the observed states, commanded states and corresponding odometer readings according to equation

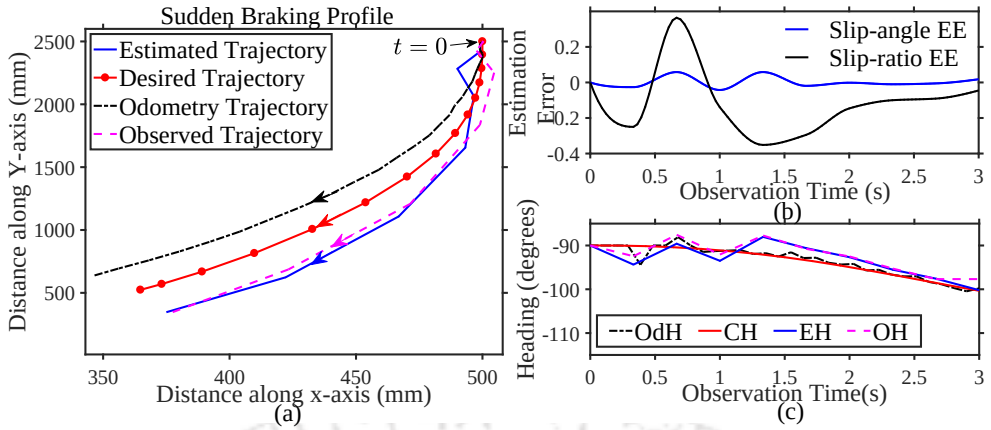

Figure 6.4: Path and heading profiles of the vehicle on event of sudden braking. Estimation errors for slip-angle (blue) and slip-ratio (black) has been shown.
EE: Estimation Error OdH: Odometry Heading CH: Commanded Heading EH: Estimated Heading OH: Observed Heading.

(6.6). Figure 6.4(b) shows the slip-angle estimation error over commanded heading and the slip-ratio estimation error against observation time. The errors are bounded and their low magnitude verifies the efficiency of the estimation, given that slippage does not evolve in a regular pattern. The desired, observed and estimated heading profiles in Figure 6.4(c) help to correlate the foregoing comparisons in Figures 6.4(a) and 6.4(b).

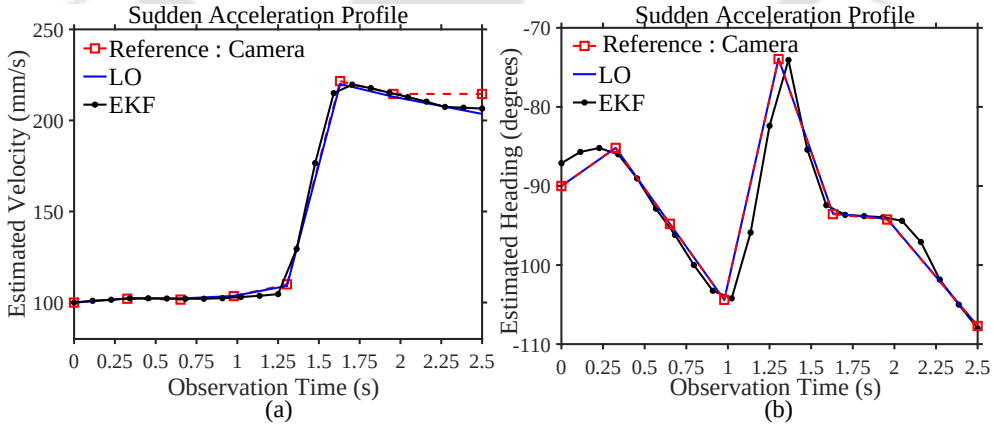

Figure 6.5: Comparison of velocity and heading estimation by a linear observer and extended Kalman filter for sudden acceleration of the vehicle.
LO: Linear Observer EKF:Extended Kalman Filter.

In Figure 6.5, we have compared the performances of a linear observer (described in Section 6.4.1) and an extended Kalman filter (non-linear) (described in Section 6.4.2) in estimating the forward velocity and heading for a sudden acceleration scenario. The EKF and the linear filter have almost similar symmetric mean average estimation errors, as can also be inferred from Table 6.1. The only disadvantage of the linear observer is that it requires nine gain parameters to be designed and tuned to achieve the desired result. However, this tuning needs to be done offline and only once before the experiment is performed. It is clear from this study that a suitably designed observer can be used to estimate slippage, irrespective of whether it is a linear filter or a nonlinear one. It is important to note that estimation is necessary in case the observations are unavailable due to a temporary period of occlusion or communication delay. During that time, the vehicle can be driven by the modified kinematics with projected slip-angle estimates from the last iteration.

Table 6.1: Mean Average Estimation Error : SA

Estimated Parameter	Linear Observer	EKF
Position (mm)	0.00079	0.00017
Velocity (mm/s)	0.0036	0.0105
Heading (degrees)	0.0015	0.0084

SA: Sudden Acceleration, EKF : Extended Kalman Filter

6.6.2 Case Study : Turning manoeuvres

The next set of experiments is aimed at studying slippage during turning manoeuvres. The first case studies a high speed motion along an arc, wherein the vehicle is subjected to slippage causing understeer or oversteer or both, depending on its rate of acceleration. This phenomenon is known as 'cornering'. The second case investigates motion along sharp turns, which is constrained by the swing radius of

the vehicle. Figure 6.6(a) illustrates the estimated path traced out by the vehicle

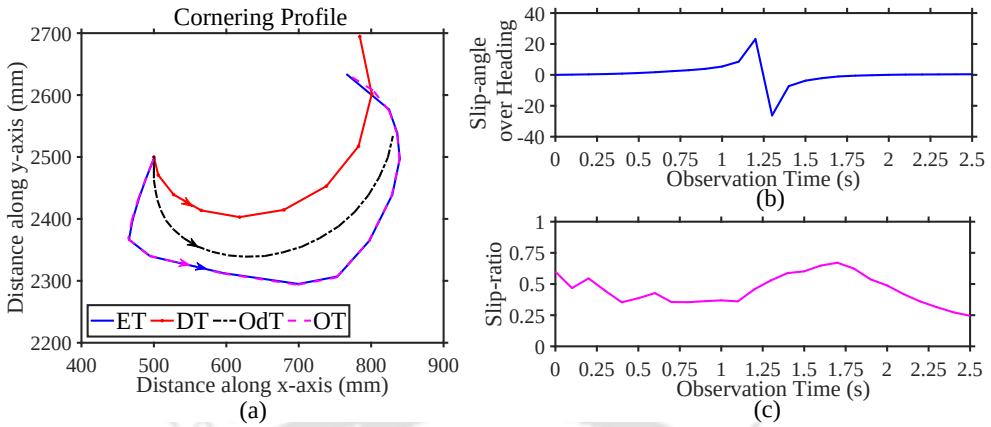

Figure 6.6: Comparison of path trajectories due to cornering. Also shows the estimated slip parameters.

ET: Estimated Trajectory DT:Desired Trajectory OdT:Odometry Trajectory OT:Observed Trajectory.

compared to the desired path. While the path computed from odometry preserves the spiral shape due to linearly increasing velocity (see Figure 6.7(b)), the deviation from the desired path is only due to a miscalculated initial heading (see Figure 6.7(c) for the heading from odometry) deliberated by inertia. However, odometry fails to recognize the actual slippage when the vehicle gains enough speed and oversteers between 1s and 1.8s. During understeer, the wheel spins in-place and the observed velocity lags the commanded velocity. Whereas, once the vehicle gains momentum, the vehicle steers inwards across the arc. At this point, the wheel skids in an effort to maintain the commanded motion and the estimated velocity leads the desired velocity. The slip-angle attains the maximum in the said interval (see Figure 6.6(b)) and changes sign due to the switch from understeer to oversteer. Figure 6.7(c) clearly shows that the transition of estimated heading is stimulated by the said slippage and the result concurs with the present analysis. The cornering effects gradually subside after 1.8s and can be observed from the decline of the slip-ratio curve in Figure 6.6(c). In the second experiment of this set, slippage is encountered as the robot tries to make a sharp

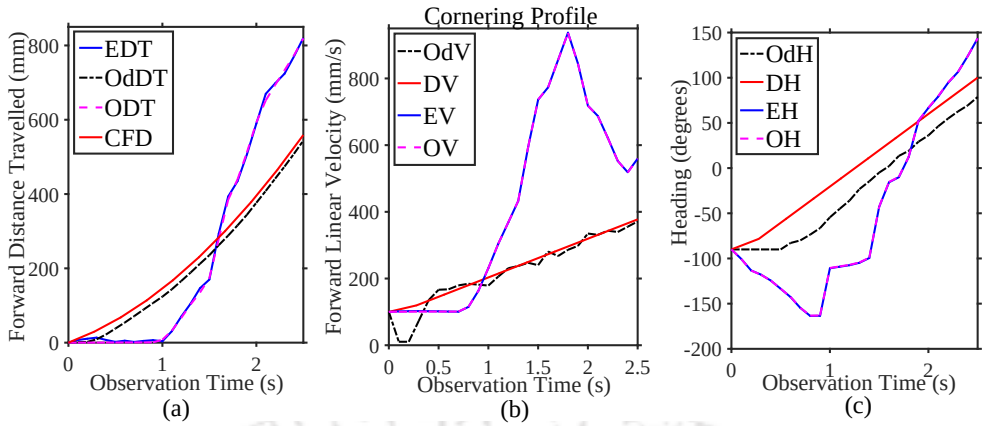

Figure 6.7: Comparison of estimated distance travelled, forward velocity and heading with commanded, observed and odometry readings in relation to slippage-induced over and understeer.

OdV/H: Odometry Velocity/Heading DV/H: Desired Velocity/Heading EV/H: Estimated Velocity/Heading OV/H: Observed Velocity/Heading EDT:Estimated Distance Travelled OdDT:Odometry Distance Travelled ODT:Observed Distance Travelled CFD:Commanded Forward Distance.

turn at a constant acceleration. A 'lane change' can be interpreted as a sequence of sharp turns in quick succession, thus challenging the robot's ability to stabilize itself along the desired heading. With the velocity increasing linearly, the initial lag in observed velocity and heading are caused by slippage due to false-rolling, as we can see in Figures 6.8(b) and 6.8(c). After 1.7s, both the estimated velocity and estimated heading make an attempt to stabilize about the reference values. Figures 6.8 also provides a performance comparison of the linear observer and extended Kalman filter in estimating pose and velocity. The preceding analysis can be further confirmed by the time-evolution of the slip parameters demonstrated by Figure 6.9. The close agreement between the observed slip parameters (slip angle in Figure 6.9(a) and slip-ratio in Figure 6.9(b)) and the estimated slip paramaters validates the efficiency of the estimation process.

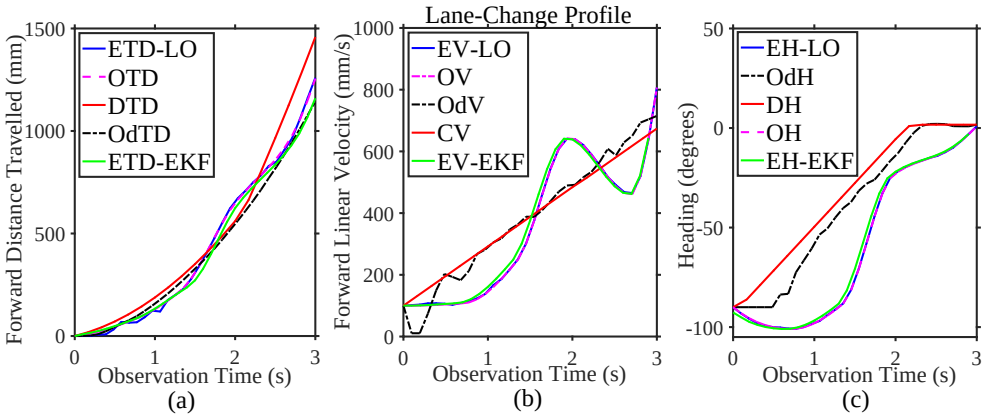

Figure 6.8: Trajectory, forward velocity and heading with lane change manoeuvre. LO stands for linear observer and EKF for extended Kalman filter.

LO: Linear Observer EKF:Extended Kalman Filter EV/H/TD: Estimated Velocity/Heading/TravelledDistance OV/H/TD: Observed Velocity/Heading/TravelledDistance OdV/H/TD: Odometry Velocity/Heading/TravelledDistance DH/TD Desired Heading/ TravelledDistance CV:Commanded Velocity.

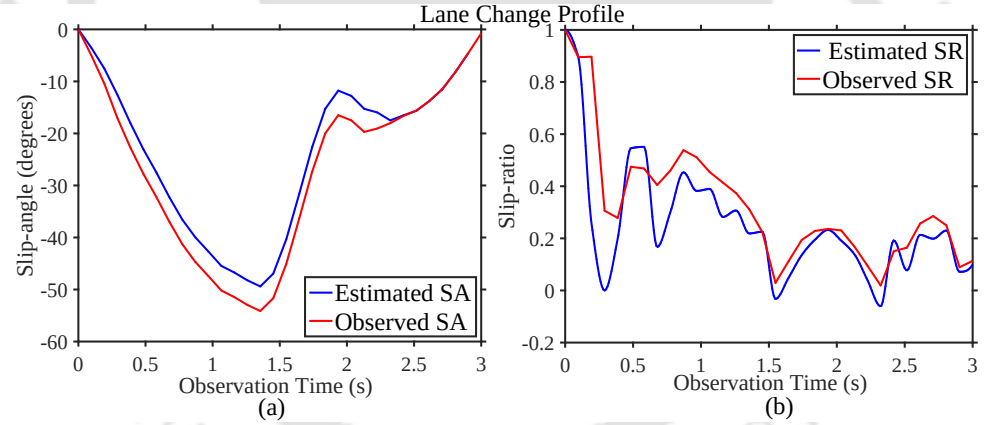

Figure 6.9: Demonstration of actual and predicted slip parameters for motion with lane change.

SA: Slip-Angle SR:Slip-Ratio.

6.6.3 Effectiveness of Slip-kinematic Model

An experiment concerning collision avoidance has been performed for validating the effectiveness of the slip-kinematic model. In Figure 6.10, the desired trajectory represents a path constraint, designed to limit the vehicle's motion to a

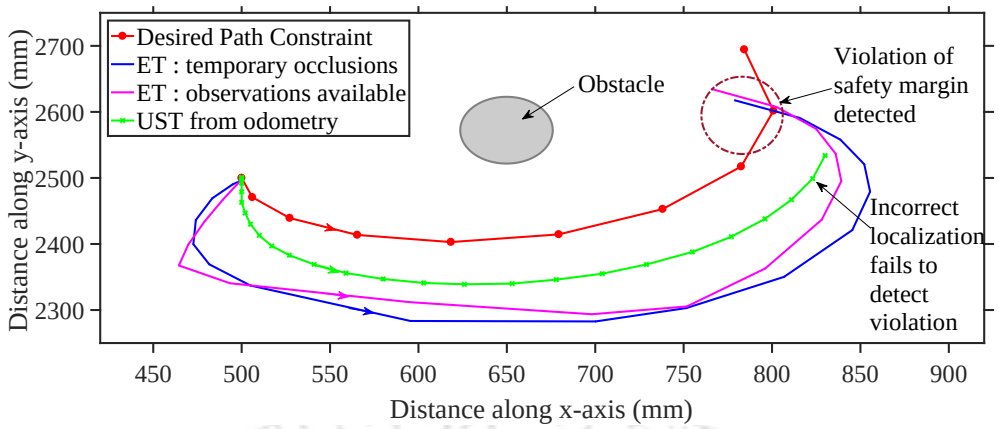

Figure 6.10: An example showing the effectiveness of the slip-kinematic model in detecting safety violation condition.

ET: Estimated Trajectory UST:Undetected Slippage in Trajectory.

safe distance from the obstacle. The vehicle is commanded to follow the desired trajectory at a constant linear acceleration (wall-following manoeuvre), thereby making the situation a classic example of cornering. Odometry, as expected, fails to detect slippage and gives a false idea of safety. Whereas, the proposed estimation approach shows the possibility of an imminent collision, as the vehicle slips, oversteers and crosses the safety boundary. Any suitable controller such as a PID controller, optimal tracking controllers designed in Chapter 3, Chapter 4 and Chapter 5 and the likes can be benefitted by a correct state feeedback in replanning the trajectory and avoiding the collision.

In a subsequent trial of this experiment, we have selectively blocked some samples of data obtained from the camera to imitate a case of temporary occlusion or data loss. At those instants, the modified kinematic model (6.15) applies a reverse engineering using the time-sequence of estimated slip-angle and slip-ratio given in (6.14) along with equation (6.6) to predict the future states. The estimated trajectory computed therein is shown in blue in Figure 6.10. It may be noted that this trajectory also indicates the violation and hence confirms that the proposed method can be utilized by a controller for taking necessary actions for improved navigation.

6.6.4 Selection of Observer Gain

The main challenge in designing the Luenberger observer is the number of parameters that need to be tuned in order to achieve the desired error convergence. While we can safely rule out the dependence of forward travelled distance and forward velocity on the vehicle's heading (and vice versa), there exists a correlation between the estimated p and v [11]. In fact, it has been experimentally confirmed that it is best to design the gain matrix in a block diagonal form. The dependence on velocity term is crucial because, it is the only state which is not directly measured but computed either by differentiating position information or integrating acceleration or a combination thereof. Experimental results indicate that k_2 and k_4 must be smaller or atmost equal to k_1 and k_5 respectively. It has been found that poles placed further away from the origin enhances the observer's efficiency, which typically means that better results can be expected if the gain is high. This is especially true for the heading, wherein camera data is the sole reference. For turning manoeuvres, however, the gain can be chosen as a diagonal matrix. The primary diagonal elements, which determine the stability of the observer, have been selected close in numerical values, typically in the order of $\sim 20 - 25$ in the above experiments. For case studies reported in Section 6.6.1, k_2 and k_4 have been numerically assigned half of the values compared to k_1 and k_5 respectively. Typical eigenvalue assignment for the four cases discussed above in selecting the gain parameters of the Luenberger observer have been shown in Table 6.2. These eigenvalues have been experimentally validated to yield acceptable convergence

Table 6.2: Eigenvalue assignment in Linear Observer

Eigenvalues	SB	SA	LC	C
λ_1	-29	-29	-20	-20
λ_2	-52	-31	-10	-22
λ_3	-10	-11	-27	-20

SA: Sudden Acceleration SB : Sudden Braking LC: Lane-change C : Cornering

rate and were found to produce realisable gain with the hardware used.

The advantage of an extended Kalman filter is that, a rigorous gain tuning procedure is inessential, but requires well-conditioned covariance matrices to effect convergence. Having said so, our investigations do not recognise any significant difference and/or inconsistency in the estimation errors incurred by both the filters. Both linear and non-linear observers have been tested on the four dynamic cases discussed above, and the number of iterations taken by the respective observers for the observer errors to reach 95% of steady-state value (5% tolerance band) has been shown in Table 6.3. EKF exhibits a faster convergence in accordance with its exponential convergence properties, but increases computational complexity by 66%. The only practical disadvantage of the Linear observer is gain-tuning, which is a single-time procedure and can be carried out offline before execution of the task. In terms of implementation, it can be concluded that either of the estimators will work well in the present application.

Table 6.3: Comparison of observer error convergence

Motion	LO (N)	EKF (N)	η
SA	7-8	2-5	4
SB	9-10	2-5	5
LC	28-30	3-5	10
C	21-25	3-4	10

SA: Sudden Acceleration SB : Sudden Braking LC: Lane-change C : Cornering
LO:Linear Observer EKF: Extended Kalman Filter N:no of iterations,
dimensionless η: Sampling Rate (samples per second)

6.7 Summary of Chapter 6

In this chapter, we have discussed local tracking controller that aids in maintaining the actual trajectories of the vehicle in close agreement with the desired optimal trajectories designed in the previous chapters. A Lyapunov based asymptotically stable tracking controller has been proposed in this context, assuming wheel encoders as primary state measurement devices. However, in presence of wheel slippage, where encoders may exhibit erroneous readings, estimation of states using linear/non-linear observers have been proposed. The proposed observers have been introduced irrespective and exclusive of the tracking controller and fundamentally aim to enhance accuracy of the state feedback in presence of slip disturbance, without having the need to replace the tracking controller by computationally intensive disturbance rejection and adaptive mechanisms.

Bibliography

[1] ER. Pinch, "Optimal Control and the Calculus of Variations," Oxford University Press, Oxford, New York (1993).

[2] T.C. Lee, C.Y. Tsai and K.T. Song, "Fast Parking Control of Mobile Robots: A Motion Planning Approach With Experimental Validation," *IEEE Transactions on Control System Technology* **12**(5), 661–676 (2004).

[3] L. Cheng, Z. Hou, M. Tan and W.J. Zhang, "Tracking Control of a Closed-chain Five-bar Robot with Two Degrees of Freedom by Integration of an Approximation Based Approach and Mechanical Design," *IEEE Transactions on System, Man, Cybernetics - B, Cybernetics* **42**(5), 1470–1479 (2012).

[4] D. Karnopp, "Vehicle Dynamics, Stability and Control," *Mechanical Engineering, CRC Press* , (2013).

[5] T.D. Gillespie, "Fundamentals of Vehicle Dynamics," *Premier Series, SAE International* , (1992).

[6] E. Frazzoli and M. Daleh, "Observers, Model Based Controllers," *6.241J Lectures on Dynamic Systems and Control, Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu. License: Creative Commons BY-NC-SA, Spring*, (2011).

[7] W. Zhu and T. Lamarche, "Velocity Estimation by Using Position and Acceleration Sensors," *IEEE Transactions on Industrial Electronics* **54**(5), 2706–2715 (2007).

[8] K. Berntorp, "Particle Filter for Combined Wheel-Slip and Vehicle-Motion Estimation," *Proceedings of American Control Conference (ACC)* **54**(5), 5414–5419 (2015).

[9] A. Katriniok and D. Abel, "Adaptive EKF-Based Vehicle State Estimation with Online Assessment of Local Observability," *IEEE Transactions on Control Systems Technology* **24**(4), 1368–1381 (2016).

[10] D. Fontanelli and F. Shamsfakhr, D. Macii and L. Palopoli, "An Uncertainty-Driven and Observability-Based State Estimator for Nonholonmic Robots," *IEEE Transactions on Instrumentation and Measurement* **70**, 1–12 (2021).

[11] L.Fusini, J.Hosen, H.H.Helgesen, T.A.Johansen and T.I.Fossen, "Experimental Validation of a Uniformly Semi-Globally Exponentially Stable Non-Linear Observers for GNSS and Camera-Aided Inertial Navigation for Fixed-Wing UAVs," *International Conference on Unmanned Aircraft Systems (ICUAS)* 851–860, (2015).

[12] K. Sun, Y. Yu, W. Zhou, T. Wang and Z. Li, "A Low-cost and Robust Optical Flow CMOS Camera for Velocity Estimation," *International Conference on Robotics and Bioinformatics (ROBIO)* 1181–1186, (2013).

[13] K. Ogata, "Discrete-Time Control Systems," *PHI Private Limited Spring*, (2009).

[14] M. Cui, "Observer Based Adaptive Tracking Control of Wheeled Mobile-Robots with Unknown Slipping Parameters," *IEEE Access* **7**, 169646–169655 (2019).

[15] M. Boutayeb, H. Rafaralahy and M. Darouach, "Convergence Analysis of The Extended Kalman Filter used as an Observer for Nonlinear Deterministic Discrete-time Systems," *IEEE Transactions on Automatic Control* **42**(4), 581–586 (1997).

[16] R.M. Amarante and A.L.C. Fujarra, "Low-cost Experimental Apparatus for Motion Tracking Based on Image Processing and Camera Calibration Techniques," *Springer Nature Applied Sciences* **2**, 1511–1522 (2020).

[17] S. Bonnabel and A. Barrau., "The Invariant Extended Kalman Filter as a Stable Observer," *IEEE Transactions on Automatic Control, Institute of Electrical and Electronics Engineers* **62**(4), 1797–1812 (2017).

[18] M.A. Al-Jarrah, A. Al-Dweik, N.T. Ali and E. Alsusa, "Joint Estimation of Location and Orientation in Wireless Sensor Networks Using Directional Antennas," *IEEE Sensors Journal* **20**(23), 14347–14359 (2020).

Chapter 7

Conclusion

This dissertation deals with navigation planning and control of autonomous vehicles deployed in tracking moving-targets. The key objectives behind planning of trajectories include safety, tracking efficiency and control effort. The parameters of interest have been optimally designed to yield best possible outcomes in dynamic situations involving narrow spaces that allow restricted movements and obstacles of unknown motion. In the foregoing chapters, a variety of dynamic environments have been considered, wherein the pursuer has been subjected to path and velocity constraints. Restricted space, such as corridors, alleys, hallways, lanes etc. offer challenges in dynamic navigation and need special attention to ensure safety, without compromising on performance. The problems have been addressed from optimal control perspective, which offer long term planning beyond the current situation and at the same time are flexible enough to be reconfigured to accommodate unexpected changes. The proposed control strategies and guidance plans have not only been verified through simulations, but also validated on hardware. Experiments have been performed on a real-life robotic platform in standard office environment to establish the merits of the designed navigation methods. Key contributions of this dissertation are listed as follows:

1. In Chapter 2, artificial potential field based planner has been proposed which

computes optimal quadratic descent directions to yield fast convergence and smooth trajectories in narrow valleys flanked by cluster of obstacles. Usual sluggish response of linear gradient descent in flat featureless surfaces and oscillations in narrow passages have been shown to improve using second order methods.

2. In Chapter 3, multi-objective optimal controllers have been designed with free and fixed planning horizons. While the former offers enhanced flexibility of navigation under dynamic and uncertain situations, the latter is easier to realise and guarantees existence of optimal solution(s).

3. Chapter 4 discusses a decoupled optimal controller design for navigating corridors and sustaining safety requirements for entire duration of collision avoidance. Collision avoidance controller has been designed to switch optimally to or from tracking controller designed in Chapter 3. The optimal switching criteria are capable to accommodate unanticipated changes in motion of the obstacle. Switching noise have been reduced by designing a seamless transition path between the two control modes.

4. In Chapter 5, an integrated control policy has been developed for automated pursuit under speed and lane restrictions. The proposed controller operates with estimation of intent of motion of the target and the obstacles and generates a safe optimal trajectory using logarithmic penalty applied to control and path constraints.

5. Chapter 6 introduces an observer based method for generating accurate state feedback in presence of wheel slippage, wherein it is not required to replace the existing optimal controller by adaptive or disturbance-rejection controllers. The proposed observer (linear or non-linear) has been shown to perform satisfactorily in conjunction with the optimal controllers designed in previous chapters and a local Lyapunov-stable tracking controller is deliberated to iteratively execute the optimal plan.

Practical experiments have been performed using shrinking horizon approach, a variant of the sub-optimal model predictive control policy. The experimental set-up has been assembled and tested in the Control and Instrumentation laboratory of the department of EEE in IIT Guwahati.

Having said so, scope of improvements are multi-directional. In all cases, we have assumed obstacles to be convex in shape, violation of which may affect convergence of gradient based planners. For optimal formulations developed in Chaper 3 and 4, approximation of reachable sets will need to be revised to maximize utilization of available space if the obstacles are allowed to be of arbitrary shape. Investigations related to both decoupled and integrated controllers designed in Chapters 4 and 5 can be extended to include uncertainty and disturbance handling. Robustness and resilience studies may be introduced in the proposed designs in dealing with crowded spaces and fast-moving obstacles. Transient response of the controllers may be improved and finite-time stability of tracking control can be examined. Stability analysis of the pursuer under proposed strategies subject to cooperative/ anti-cooperative intent of the obstacles and the target can be an important aspect of further research. Exploring automatic weight allocation system for the designed controllers is another possible route to future work. Robust state estimation combined with disturbance rejection can be an interesting addition to observer based state feeedback corrective approach. With multiple fully automated robotic systems being increasingly used in commercial and retail sectors, safe optimal navigation needs extensive investigations in the coming days.

Chapter 8

Appendix-A

8.1 Obstacle Mapping

An object of arbitrary shape can be mapped into convex shaped representative obstacle. Assuming cross-sectional view of an obstacle is detected using overhead vision sensor, the procedure consists of three steps, elaborated below:

1. Polygon Approximation: Detected object is approximated by an irregular non-selfintersecting polygon of predetermined number of vertices, the vertices being points selected at predetermined intervals from the identified boundary of the object, such that, error between observed area of the arbitrary shaped object and computed area of the approximated polygon is minimized.

2. Computation of centroid: Centroid of the approximated polygon is computed based on approximate lengths of the sides of the polygon.

3. Construction of Circumcircle: With the centroid at the centre and the longest distance between the centroid and any vertex of the polygon being assumed as the radius, a circle is constructed, such that the circle completely contains the arbitrary shaped object inside it.

The resultant circumcircle acts as a representative obstacle with convex properties. More specifically, the obstacle can be characterized by a circle whose centre and radius are available (by computation) at any $t \in [0, T]$, where T is a finite task completion time. This means, an i^{th} isolated mobile obstacle can be traced by the 3-tuple, $(x^i_{obs}(t), y^i_{obs}(t), r^i_{obs})$. For more than one closely spaced obstacles (having

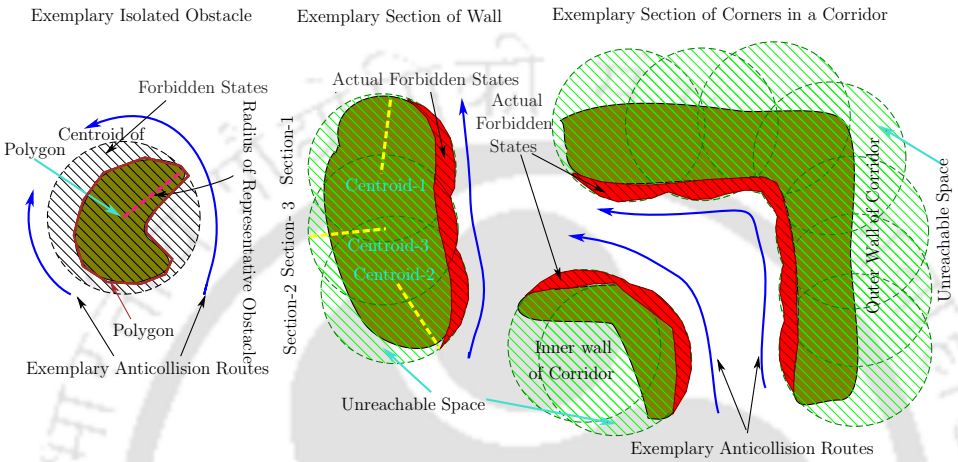

Figure 8.1: Construction of representative obstacles in different cases.

no passage space in between) the representative obstacle needs to accommodate all elements of the set, because overlapping constraint boundaries are difficult to address simultaneously. The set of obstacles can be grouped together into a larger obstacle. Each obstacle is approximated by a suitable polygon and centroid of the set of obstacles is computed by considering the centroid of each individual obstacle as a vertex of a planar polygon. The point corresponding to the centroid of the resultant planar polygon forms the centre of the representaive j^{th} obstacle, $(x^j_{obs}(t), y^j_{obs}(t))$. Calculation of centroid and area for a n-sided polygon is given by the following expressions, where A denotes area of the polygon [1]:

$$
\begin{aligned}
C_x &= \frac{1}{6A}\sum_{i=0}^{n}(x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \\
C_y &= \frac{1}{6A}\sum_{i=0}^{n}(y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \\
\text{where, } A &= \frac{1}{2}\sum_{i=0}^{n}(x_i y_{i+1} - x_{i+1} y_i)
\end{aligned}
\tag{8.1}
$$

Suppose, the largest line segment joining the point $(x^j_{obs}(t), y^j_{obs}(t))$ and any of the centroids of the individial obstacles of the set is denoted as $r_{max}(t)$. Also suppose, $r_{max}(t)$ corresponds to the i^{th} individual obstacle of the set at t. Then assume, the line segment joining the centroid of the i^{th} obstacle and its farthest vertex is termed as r_i. Herein, a representative circular obstacle, $r^j_{obs}(t)$ may be constructed with $r_{max}(t) + r_i$ as the radius and $(x^j_{obs}(t), y^j_{obs}(t))$ as the centre at any time t. Figure 8.2 explains the mechanism pictorially.

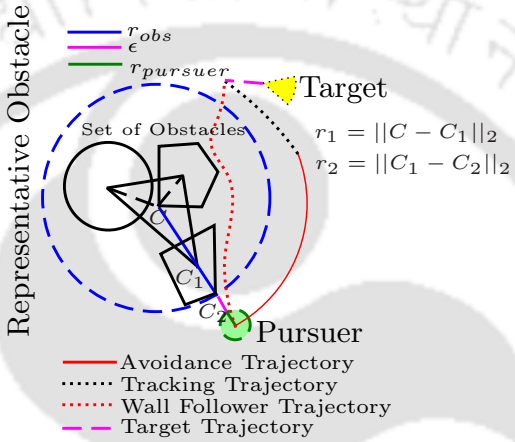

Figure 8.2: Representative obstacle created from a set of closely-spaced obstacles of arbitrary shapes, the obstacles having no passage through them.
$r_1 = r_{max}$, $r_2 = r_i$, $C = (x^j_{obs}, y^j_{obs})$

8.2 Pursuer Model

The modified pursuer model has been described by only two states –the 'x' and the 'y' positions representating a point motion model. Assumed geometric symmetry of the pursuer's physical structure has been adopted for simplification and the concerned shape has been considered circular in the motion plane. The circle represents the extent of the pursuer and is defined by the radius, $r_{pursuer}$ in addition to the position variables. The inputs are the linear velocity and heading angle. This simple but still non-linear representation of state-space is advantageous for a number of reasons.

1. *Enhances comprehensibility of controller mathematics*: For all numerical calculations, this model can be easily converted into the linearized discrete-time version of a standard velocity-driven single-integrator model. In this regard, it has been reported that for complex path planning and coordination control problems, a simple single integrator model is practically suitable and sufficient [2].

2. *Reduces computational footprint*: This minimal state-space representation aids faster run-time execution of iterations of the concerned optimization routine. This is especially true for position-controlled systems which are designed to receive foward velocity and orientatinon commands. For obtaining improved accuracy an inertial measurement unit may be integrated with the pursuer's hardware for velocity and heading correction.

3. *Minimizes discretization errors*: Continuous-time systems having two degrees of freedom or less are known to be the most compliant models to their discrete-time versions [3]. Therefore, the modified 2-state model generates minimal discretization error during hardware implementations.

4. *Finite velocity boundary*: This model inherently imposes a finite boundary to the forward linear velocity that validates applicability of the maximum principle. Persistent excitation condition (velocity does not become zero during task execution) relates the forward velocity to Langrange's multipliers, which are not physical variables but are finite by definition.

8.3 State Estimation : Fixed Finite Horizon

We suppose, poses of target/obstacle(s) are available by measurement or computation at each sampling instant, δt from the beginning of the task ($t = 0$) till current time, t. At t, the time till interception is $(T_{int} - t)$. From a practical aspect, time-to-interception is configured to comprise of a maximum of $n = 1 + \frac{T_{int}}{\delta t}$

datapoints in a planning iteration. For the duration $[0, t]$ state feedback is assumed to be available for target and obstacle(s). Each new measurement obtained periodically, contributes revised information about the motion/intent of the corresponding vehicle. This revised information precisely characterizes the evolution of respective states (or control inputs, if kino-dynamic state propagation models are available). The idea is to extrapolate the currently known pattern of states and control variables (whichever applicable) of the target and obstcale(s) in the form of a polynomial function of time. The aim is to predict states and control variables pertaining to a future time belonging to the interval, $(t, T_{int}]$. Polynomial estimation has the felxibility to predict states at one or more points pertaining to $(t, T_{int}]$, and hence can be used iteratively without incurring additional computational complexity. Fixed sampling rate limits the estimated polynomial to a maximum degree of $(n-1)$.

The procedure of extrapolation by Neville Aitken algorithm [4] generates a polynomial of 0^{th} order when only one data is available at $t = 0$. Then it proceeds by adding a degree to the polynomial when a new pose is read from the sensors. As the planning horizon shrinks, the maximum degree of polynomial constructed remains the same, thereby reducing the sampling time. Possibilities of constraint violation are also reduced as the fixed terminal time is approached.

In the following description, t represents the independent variable and x and y are the dependent variables. By way of example, two polynomials are constructed in each planning iteration, one for x and the other for y (states x and y may belong to target/obstacle(s)). It may be recalled again, the parametric form of target and obstacle motion are used only for intent assessment at a future time, which can be either the time-to-intercept or a time-to-collide. For target, only the final states at T_{int} are utilized from the process of extrapolation. Initial guess for constructing the polynomials is usually inaccurate. But as new measurements are obtained, the estimated pattern of states accurately represents the vehicle's intent. As a result, the generated polynomial represents a reliable approximation of the actual trajectory (which is not-arbitary, but unknown). Dataflow of the

Neville-Atkin algorithm in an exemplary 3^{rd} iteration proceeds in the sequence, $P_{23} \rightarrow P_{123} \rightarrow P_{0123}$ (see Table 8.1):

At the i^{th} instant in any planning iteration, the resultant polyomial is given by $P_{012...i}$, which is then evaluated at a future instant (for example, T_{int} for target) to predict states of the target and the obstacles at that future instant.

Iteration	Independent variable	Measured variable	Polynomial D(1)	Polynomial D(2)	Polynomial D(3)
0	$t = 0$	$x_0 = f(0) = P_0$			
			P_{01}		
1	$t = \delta t$	$x_1 = f(\delta t) = P_1$		P_{012}	
			P_{12}		P_{0123}
2	$t = 2\delta t$	$x_2 = f(2\delta t) = P_2$		P_{123}	
			P_{23}		
3	$t = 3\delta t$	$x_3 = f(3\delta t) = P_3$			

Table 8.1: Neville Aitken's Algorithm

$$P_{ij} = \frac{t-t(j)}{t(i)-t(j)} f(t(i)) + \frac{t-t(i)}{t(j)-t(i)} f(t(j)), \qquad P_{ijk} = \frac{(t-t(k))P_{ij} - (t-t(i))P_{jk}}{t(i)-t(k)}$$

Bibliography

[1] J.G. Tough, " The computation of the area, centroid, and principal axes of a polygon," *Computers & Geosciences* **14**(5), 715-717 (1988).

[2] S. Zhao and Z. Sun, "Defend the Practicality of Single-Integrator Models in Multi-robot Coordination Control," *13th IEEE International Conference on Control & Automation*, Ohrid, 666–671 (2017).

[3] T.S. Chung and J.W. Chia, "A Computationally Efficient Numerical Algorithm for The Minimum-time Control Problem of Continuous Systems," *Automatica* **28**(4), 841–847 (1992).

[4] 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).

Appendix-B

Existence theorems for solutions of boundary value problems involving second order differential equations of the form $\ddot{y}(t) + f(y, \dot{y}, t) = 0$ requires satisfaction of following conditions, the domain for $f(y, \dot{y}, t)$ being $[a, b] \times [m_1, m_2] \times [t_1, t_2]$:

1. The function, $f(y, \dot{y}, t)$ is continuous over the domain of $f(y, \dot{y}, t)$, at least within the interior of the domain.

2. The function, $f(y, \dot{y}, t)$ is Lipschitzian.

3. The function, $f(y, \dot{y}, t)$ is bounded in the closure of the finite interval $[t_1, t_2]$ and $y(t)$ and $\dot{y}(t)$ exist within the said interval.

The first and the third conditions are ususally met with easily. The second condition, however, needs to be established, not only for the sake of understanding how many solutions the optimal problem can have, but also to find a 'suitable' interval which can guarantee existence of exactly one solution. Thus, it is the Lipschitzian constants and the interval, $[t_1, t_2]$ that define the unique solution (if any) for the given problem. Lipschizian property can be verified by a number of ways depending on the nature of the function $f(y, \dot{y}, t)$, some requiring evaluation of one or more pairs of constants, while some necessitating the verification of boundedness of partial derivatives, $D_y, D_{\dot{y}}, D_t$ of the function $f(y, \dot{y}, t)$. The case by case studies have been explained below:

8.4 Existence and Uniqueness: Tracking

Existence and uniqueness of the tracking trajectory depends on the final time to interception, T_{int}(or T). For a free-horizon problem, at least one finite final time to interception, T needs to be found for analyzing uniqueness of solution. For a fixed-horizon problem, the finite time,T_{int} selected as the time of interception shall decide the uniqueness property. The boundary value problems of both free and fixed horizon tracking modes can be remodelled into a second order system discussed above. Let us illustrate the case of a fixed horizon problem, starting with equation (3.23) in Chapter 3.

Let us define a new set of variables, $\tilde{x}(t)$, $\tilde{y}(t)$, $\tilde{\lambda_1}(t)$, $\tilde{\lambda_2}(t)$, such that

$$\tilde{x}(t)=x(t)-x_{tar}(t), \quad \tilde{\lambda}_1(t)=\lambda_1(t)-2w_v v_{tar}\cos(\theta_{tar})$$
$$\tilde{y}(t)=y(t)-y_{tar}(t), \quad \tilde{\lambda}_2(t)=\lambda_2(t)-2w_v v_{tar}\sin(\theta_{tar})$$

With a change of variables, the ordinary differential equations in (3.23) can be linearly transformed into equation (8.2). The functions $f_1(t)$ and $f_2(t)$ depend explicitly on the target states and its derivatives at t, and therefore are not functions of $x(t)$, $y(t)$, $\dot{x}(t)$ and $\dot{y}(t)$.

$$\begin{aligned}\dot{\tilde{x}}(t)&=\frac{\tilde{\lambda}_1(t)}{2w_v}, \quad \dot{\tilde{\lambda}}_1(t)=2w_d(\tilde{x}(t))-f_1(t)\\ \dot{\tilde{y}}(t)&=\frac{\tilde{\lambda}_2(t)}{2w_v}, \quad \dot{\tilde{\lambda}}_2(t)=2w_d(\tilde{y}(t))-f_2(t)\end{aligned} \tag{8.2}$$

The transformed equations in (8.2) can be reformulated as second order differential equations (8.3) in terms of the new states $\tilde{x}$ and $\tilde{y}$, where, $f(\tilde{x},t) = -(\frac{w_d}{w_v})\tilde{x}(t)+f_1(t)$ and $f(\tilde{y},t)=-(\frac{w_d}{w_v})\tilde{y}(t)+f_2(t)$.

$$\ddot{\tilde{x}}(t)+f(\tilde{x},t)=0, \quad \ddot{\tilde{y}}(t)+f(\tilde{y},t)=0 \tag{8.3}$$

In equations (8.3), the boundary values of $\tilde{x}$ and $\tilde{y}$ at known at both boundaries, $t=0$ and $t=T_{int}$ (target states are estimated and made to satisfy final state constraint). For the free horizon boundary value problem given by equation (3.7), the terminal time, T is first computed and then the solutions $x(t)$, $y(t)$ are evaluated, wherein the terminal states can be computed with the knowledge of T.

Suppose, $(\tilde{x}_1, \tilde{y}_1)$ and $(\tilde{x}_2, \tilde{y}_2)$ are two tuples satisfying equation (8.3) and w_o is a constant defined as $w_o = -(\frac{w_d}{w_v})$. Triangle inequality can be applied to arrive at equation (8.4).

$$\begin{aligned} |f(\tilde{x}_1, t) - f(\tilde{x}_2, t)| &= |(w_o)(\tilde{x}_1 - \tilde{x}_2)| \leq |w_o| \cdot |\tilde{x}_1 - \tilde{x}_2| \\ |f(\tilde{y}_1, t) - f(\tilde{y}_2, t)| &= |(w_o)(\tilde{y}_1 - \tilde{y}_2)| \leq |w_o| \cdot |\tilde{y}_1 - \tilde{y}_2| \end{aligned} \tag{8.4}$$

Since the target states are assumed to be continuous and bounded, interception of target and the pursuer implies $f(\tilde{x}, t)$ and $f(\tilde{y}, t)$ are also continuous and bounded functions within $[0, T_{int}]$. We can choose any pair of non-negative Lipschitizian constants, K_x and K_y such that, $|w_o| \leq K_x$ and $|w_o| \leq K_y$ [3]. Existence of the constants K_x and K_y implies Lipschitizian property of $f(\tilde{x}, t)$ and $f(\tilde{y}, t)$. If it is possible to select a positive real interception time, such that, $T_{int} < \frac{\pi}{max\{K_x, K_y\}}$, the current problem will assure existence of exactly one solution [2] within $[0, T_{int}]$ and for the given boundary values. T_{int} being a design parameter, it is one of the advantages of fixed-horizon tracking approach over free-horizon tracking, where T is not chosen but computed.

8.5 Existence-Uniqueness: Plan-B

For the avoidance trajectory, it can be shown by Lipschitz continuity, that one and only one solution exists. Given, the split boundary value problem in (4.7) of Chapter 4, both existence and uniqueness of the solution is valid in the domain $[t_3, t_3 + \Delta]$, only if both the boundary values are zero. Linear transformations have been applied to the states $x(t)$ and $y(t)$ of (4.7) and new variables have been defined as $\tilde{x}(t)$, $\tilde{y}(t)$. Let, $\tilde{x}(t) = x(t) - l_x(t)$ and $\tilde{y}(t) = y(t) - l_y(t)$, such that:

$$\begin{aligned} \tilde{x}(t_3) = 0, \quad \tilde{y}(t_3) = 0 \\ \dot{\tilde{x}}(t_3 + \Delta) = 0, \quad \dot{\tilde{y}}(t_3 + \Delta) = 0 \end{aligned}$$

The linear functions of time, $l_x(t)$ and $l_y(t)$ have been defined as in (8.5) such that the boundary relations described in (8.6) holds.

$$\begin{aligned} l_x(t)&=x(t_3)-(t_3-t)\frac{\lambda_1(t_3+\Delta)}{2w_v} \\ l_y(t)&=y(t_3)-(t_3-t)\frac{\lambda_2(t_3+\Delta)}{2w_v} \end{aligned} \tag{8.5}$$

$$\begin{aligned} l_x(t_3)&=x(t_3), \quad \dot{l}_x(t_3+\Delta)=\dot{x}(t_3+\Delta) \\ l_y(t_3)&=y(t_3), \quad \dot{l}_y(t_3+\Delta)=\dot{y}(t_3+\Delta) \end{aligned} \tag{8.6}$$

The first derivatives, $\dot{l}_x(t)$ and $\dot{l}_y(t)$ are obtained as numerical values, which means the second derivatives can be ignored and this leads to the assumption, $\ddot{\tilde{x}}(t)=\ddot{x}(t)$ and $\ddot{\tilde{y}}(t)=\ddot{y}(t)$. As a result, we obtain a pair of second order ordinary differential equations (8.7) in terms of the variables, $\tilde{x}(t)$ and $\tilde{y}(t)$, where, $p=\frac{\lambda_3}{w_v}$ (λ_3 is a constant).

$$\begin{aligned} \ddot{\tilde{x}}(t)+p(\tilde{x}(t)+l_x(t)-x_{obs}(t))&=0 \\ \ddot{\tilde{y}}(t)+p(\tilde{y}(t)+l_y(t)-y_{obs}(t))&=0 \end{aligned} \tag{8.7}$$

Defining $f(t,\tilde{x})=p(\tilde{x}(t)+l_x(t)-x_{obs}(t))$ and $f(t,\tilde{y})=p(\tilde{y}(t)+l_y(t)-y_{obs}(t))$, we arbitrarily choose $(\tilde{x}_1,\tilde{y}_1)$ and $(\tilde{x}_2,\tilde{y}_2)$ to be two tuples satisfying equation (8.7) at any $t\in[t_3,t_3+\Delta]$. Applying triangle inequality gives (8.8).

$$\begin{aligned} |f(t,\tilde{x}_1)-f(t,\tilde{x}_2)|&=|p(\tilde{x}_1(t)+l_x(t)-x_{obs}(t))-p(\tilde{x}_2(t)+l_x(t)-x_{obs}(t))| \\ &\leq|p|\cdot|\tilde{x}_1(t)-\tilde{x}_2(t)\| \\ |f(t,\tilde{y}_1)-f(t,\tilde{y}_2)|&=|p(\tilde{y}_1(t)+l_y(t)-y_{obs}(t))-p(\tilde{y}_2(t)+l_y(t)-y_{obs}(t))| \\ &\leq|p|\cdot|\tilde{y}_1(t)-\tilde{y}_2(t)| \end{aligned} \tag{8.8}$$

Hence, $f(t,\tilde{x})$ and $f(t,\tilde{y})$ are Lipschitzian as, it is always possible to choose two non-negative constants, K_x and K_y, such that, $|p|\leq K_x$ and $|p|\leq K_y$. Assuming $f(t,\tilde{x})$ and $f(t,\tilde{y})$ are continuous in $t\in[t_3,t_3+\Delta]$, both existence and uniqueness of the transformed problem is guaranteed [2]. Since, $l_x(t)$ and $l_y(t)$ are also Lipschitz continuous, the original problem will also have one and only one solution.

8.6 Existence: Integrated Approach

The differential equations in (5.8) of Chapter 5 represent a class of boundary value problems, having the form $\dot{s} = \mathfrak{f}(t, s)$, where s is the 4-element state vector $[x(t), y(t), \lambda_1(t), \lambda_2(t)]$. The boundary constraint functions are of the form $g(s_i(0)) = a$ and $g(s_i(T_{int})) = b, \forall i \in \{1, 2, 3, 4\}$, where, a and b are real numbers and the independent variable is time t, $(t \in [0, T_{int}])$ and T_{int} is the fixed time-to-interception. The existence-uniqueness of such problems have been investigated in reference [1] using Perov's comparison theorem.

In this case, the boundary values of the co-states $\lambda_1(t)$ and $\lambda_2(t)$ are unknown. In fact, the co-states being non-physical variables, are free to assume any numerical value satisfying the constraint functions, $g(s_i(0))$ and $g(s_i(T_{int}))$. It is justified to reason that, for each unique pair of states $(x(t), y(t))$ there is *at least* a pair of $(\lambda_1(t), \lambda_2(t))$, $\forall t \in [0, T_{int}]$, which means there is a possibility that there can be more than one pair of co-states validating the same pair of states in the given interval. However, it can be proved that there cannot exist more than one linearly independent vector of (λ_1, λ_2) in the said interval, corresponding to each unique state vector $(x(t), y(t))$ defined in the same interval. In simple terms this means, a pursuer starting out with a specific initial velocity and intercepting the target at another specific velocity cannot travel in more than one trajectories. If multiple solutions do exist, then the initial and terminal co-states of the second solution will definitely be different form the first solution set.

High degree of non-linearity and correlation between the variables result in absence of closed form solutions for (5.8). Thereby, an approximate numerical solution can be computed that satisfies (5.1). Then, tracing back to the start of this approximate sequence, one can find a set of $(\lambda_1(0), \lambda_2(0))$ corresponding to the solution generated by the numerical solver. A hands-on validation of the above conjecture can be obtained in this process. We can try to determine, if starting with the computed approximate initial values of the co-states and the known initial values of the states, it is possible to arrive at two different solutions

(with same initial and terminal values only) in the above mentioned interval. The following fundamental assumptions are responsible for the above inference:

1. By definition, the costates, $\lambda_i(t)$ are continuous and bounded functions of time.

2. There is a unique set of $(x(t), y(t))$ that satisfy the bounary conditions in the closed interval $[0, T_{int}]$ (if there are a number of solutions, $(x(t), y(t))$, the following discussion applies to each, separately).

In reference, [2], a detailed explanation has been provided using Picard's theorem, which points out the validity of second condition. The proof involves converting the boundary value problem into an initial value problem and then applying the conditions of continuous evolution of states into an extended state space, starting from the original domain of the initial value problem. Starting with the said set of initial values of states and co-states, if the right hand side of (5.8) can be shown to be continuous, bounded and Lipschitz, in the closed interval, $[0, T_{int}]$, then it can be proved by applying *continuous dependence theorem* [2] on the initial conditions that a solution, $(x(t), y(t), \lambda_1(t), \lambda_2(t))$, starting inside the closed interval, $[0, T_{int}]$ will evolve *uniquely* over the said closed interval. Obviously, the magnitude of the interval, $[0, T_{int}]$ plays a crucial role in satisfaction of the unique evolution of states. Based on these assumptions, our task is narrowed down to verification of the Lipschitz condition by checking the boundedness property of the partial derivatives of the expressions in the right hand side of (5.8).
Let us define the augmented states of the problem $s(t) := [x(t)\ \ y(t)\ \ \lambda_1(t)\ \ \lambda_2(t)]^\top$ and rewrite (5.8) as given in (8.9).

$$\dot{s}(t) = \mathfrak{f}(t, s(t)) = [\dot{x}(t)\ \ \dot{y}(t)\ \ \dot{\lambda}_1(t)\ \ \dot{\lambda}_2(t)]^\top \tag{8.9}$$

Here, $\mathfrak{f}(t, s(t))$ is continuous in $s(t)$ and bounded, because, the augmented state trajectories are continuous and bounded functions of time. Now, the partial derivatives of $\mathfrak{f}(t, s(t))$ with respect to $s(t)$ can be computed as follows:
$\Delta\mathfrak{f}_i = \left[\frac{\partial \mathfrak{f}_i}{\partial x(t)}\ \ \frac{\partial \mathfrak{f}_i}{\partial y(t)}\ \ \frac{\partial \mathfrak{f}_i}{\partial \lambda_1(t)}\ \ \frac{\partial \mathfrak{f}_i}{\partial \lambda_2(t)}\right]^\top$, $i \in \{1, 2, 3, 4\}$.

For brevity, let us observe one of the partial derivatives, $\Delta\mathfrak{f}_1$.

$$\begin{aligned}\Delta\mathfrak{f}_1 =&[0 \;\; 0 \;\; e_3 \;\; e_4]^\top \\ \text{where, } e_3 =&-\lambda_2^2\left(\frac{v_{max}}{(\lambda_1^2+\lambda_2^2)^{1.5}}+\frac{w_v}{(\lambda_1^2+\lambda_2^2)^2}\right) \\ \text{where, } e_4 =&\left(\frac{v_{max}(\lambda_1\lambda_2-(\lambda_1^2+\lambda_2^2))}{(\lambda_1^2+\lambda_2^2)^{1.5}}-\frac{2w_v\lambda_1\lambda_2}{(\lambda_1^2+\lambda_2^2)^2}\right)\end{aligned} \tag{8.10}$$

Each non-zero element of $\Delta\mathfrak{f}_1$ can be expressed as a rational fraction, whose denominator polynomial has a greater degree than that of the numerator, which implies, the partial derivative with respect to $s(t)$ has a finite limiting value, and that the function is bounded. The process is same for other partial derivatives. The Lipschitz constant K can be computed as $[\hat{K}_1 \;\; \hat{K}_2 \;\; \hat{K}_3 \;\; \hat{K}_4]^\top$, where, $\hat{K}_i := sup|\Delta\mathfrak{f}_i|$.

We can summarize that, a pursuer starting at some point within the domain, $\mathfrak{S}$ of the corresponding initial value problem with a set of initial heading and velocity cannot follow two different routes (route implying state and control vectors) to interception, where interception takes place in a larger set $\tilde{\mathfrak{S}}$ containing $\mathfrak{S}$. For more than one candidate solution of the optimal control problem, the initial states and co-states must be different for each solution, which is why the numerical solver generates one of the possible solutions depending on the intial conditions specified for the solver.

8.7 Derivation of Slip-ratio Propagation Model

$$\begin{aligned}\sigma &= \frac{|v_{encoder}(t)-\hat{v}(t)|}{\max\{v_{encoder}(t),\hat{v}(t)\}} \\ \dot{\sigma}(t) &= \frac{d}{dt}\left(\frac{|v_{encoder}(t)-\hat{v}(t)|}{v_{encoder}(t)}\right) \text{ if}(a) \\ &= \frac{d}{dt}\left(\frac{|v_{encoder}(t)-\hat{v}(t)|}{\hat{v}(t)}\right) \text{ if}(b)\end{aligned}$$

$$\left.\begin{aligned}\dot{\sigma}(t) &= \frac{-\hat{v}(t)}{v_{encoder}(t)}\frac{\dot{v}_{encoder}(t)}{v_{encoder}(t)} + \frac{\dot{\hat{v}}(t)}{v_{encoder}(t)} \text{ if}(a)\\ &= \frac{v_{encoder}(t)}{\hat{v}(t)}\frac{\dot{\hat{v}}(t)}{\hat{v}(t)} - \frac{\dot{v}_{encoder}(t)}{\hat{v}(t)} \text{ if}(b)\end{aligned}\right\}$$

$$\left.\begin{aligned}\dot{\sigma}(t) &= (\sigma(t)-1)\left(\frac{\dot{v}_{encoder}(t)}{v_{encoder}(t)} - \frac{\dot{\hat{v}}(t)}{\hat{v}(t)}\right) \text{ if}(a)\\ &= (\sigma(t)+1)\left(\frac{\dot{\hat{v}}(t)}{\hat{v}(t)} - \frac{\dot{v}_{encoder}(t)}{v_{encoder}(t)}\right) \text{ if}(b)\end{aligned}\right\}$$

where, $(a) : v_{encoder}(t) > \hat{v}(t)$ and $(b) : v_{encoder}(t) < \hat{v}(t)$.

Bibliography

[1] P. Waltman, "Existence and uniqueness of solutions of boundary value problems for two dimensional systems of nonlinear differential equations," *Transactions of the American Mathematical Society* **153**, 223–234 (1971).

[2] PB. Bailey, LF. Shampine and PE. Waltman, "Nonlinear Two Point Boundary Value Problems," *Computational Mathematics : Theory and Mathematics (General), Lecture Notes in Mathematics in Science and Engineering, Academic Press, Philadelphia, USA*, (1968).

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

Appendix-C

8.8 Hardware Description

The hardware unit broadly comprises of the mobile units (target, obstacles and pursuer), sensors, a server computer, and a client computer communicatively coupled to the server.

The target and the obstacles have been realized with small sized autonomous robots, each having a radius of about 0.1m. The robots are capable to execute motion according to commanded control signals with the help of motors attached to wheel encoders.

The pursuer has been implemented on a programmable research-oriented mobile robot by 'Patrolbot' MobileRobotics Inc.[1]. Patrolbot is a two-wheeled differential drive system supported by two additional caster wheels at the front and the back of the chassis. The symmetrically placed drive-wheels are controlled by an in-system proportional-integral-derivative (PID) controller. The PID commands are generated by an internal processor having an embedded firmware. The firmware interfaces with higher level optimal controllers designed in Chapters 2-5 through a programmable layer called the *server*. The server runs on a piggy-backed computer as shown in Figure 8.3. Patrolbot has a swing radius of 29cm, saturation velocity of 1.8m/s and a saturation turn rate of 300°/s.

Quadrature encoders attached to the driven wheels provide in-built dead-reckoning capabilities. In addition, a 6-DOF inertial measurement unit (IMU) (MPU6050) has been attached close to the base along the centre of gravity. The accelerometer operates in the $\pm 2g$ mode and the gyroscope at $\pm 250°/s$. The IMU is commu-

nicatively coupled to an Arduino programming board operating at 115200 baud rate. The quadrature encoders measure pose and the inertial measurement unit measure acceleration and angular velocity of the vehicle.

For localization of the target, obstacles and the pursuer, different pre-registered visual markers have been attached on the tops, each having a unique and asymmetric pattern. A camera has been attached in the ceiling of the designated workspace so that it receives an aerial view of the movement of the robots in the arena. The camera is a fixed focus $720p/30$fps (maximum resolution) webcam with 60° field of view. A remote computer connected to the camera operates as the client. Figure 8.3 illustrates the components of the experimental setup.

Figure 8.3: Different elements of the hardware setup used.

8.9 Control Architecture

The architecture, as illustrated in Figure 8.4(a), describes a control loop which operates sequentially as the client and the server. The online implementation uses a shrinking horizon policy, which is configured to consider the entire mission

horizon in the first iteration and is better suited to provide a backup plan in case a solution is unavailable at any $t \in [0, T_{int}]$. The article in [6] confirms that shrinking horizon policy consumes lower energy, takes lower computation time and provides better transient performance than receding horizon model predictive control.

The client computer receives localization information (state feedback) pertaining to the target, obstacles and the pursuer from the overhead camera and generates closed loop optimal tracking trajectories. The client communicates with the server via ad-hoc wireless network using User Datagram Protocol (UDP) based [3],[4] serial communication. A customized handshaking protocol has been implemented for identification of loss of data packets. The client 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 prioritizes fast data transfer over data accuracy. The transmission has been designed with a fault detection routine (in 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.

The set of optimal control sequence 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 (camera, for example). Since the

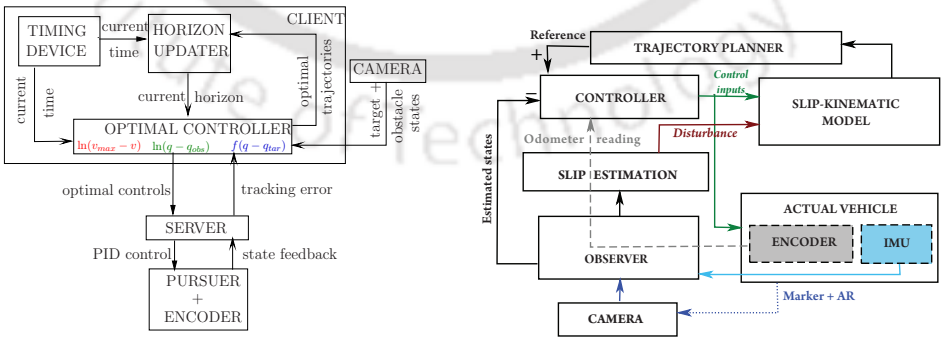

Figure 8.4: (a) An overview of the control architecture (b) Data flow between the system, observer and controller.

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 this replanning action is reflected by the shifting of trajectories in subsequent iterations.

The server generates PID motion commands, which are transmitted to the actuators of the pursuer through the firmware interface. The interface also 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 stabilizes the open-loop dataflow of the trajectory planner. 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 [5] this architecture can reconfigure the reference trajectory of the closed loop controller. This feature, in association with the resilience propoerty of shrinking horizon policy, can accommodate unanticipated changes in the intent of the target and the obstacles. Role of observers in estimating accurate state feedback in presence of wheel slippage disturbance has been depictd in Figure 8.4(b).

8.10 Software Tools

Detection and identification of the pursuer, target and the obstacles are performed using an *Augmented Reality* toolkit [2] and 'Unity 3D', which is a graphical user interface based pose mapper application. Exemplary experimental snapshots observed in Unity 3D has been shown in Figure 8.5. This application is enabled to recognize the visual markers on the vehicles and correspondingly generate a stream of position and orientation data. Localization data from the software is

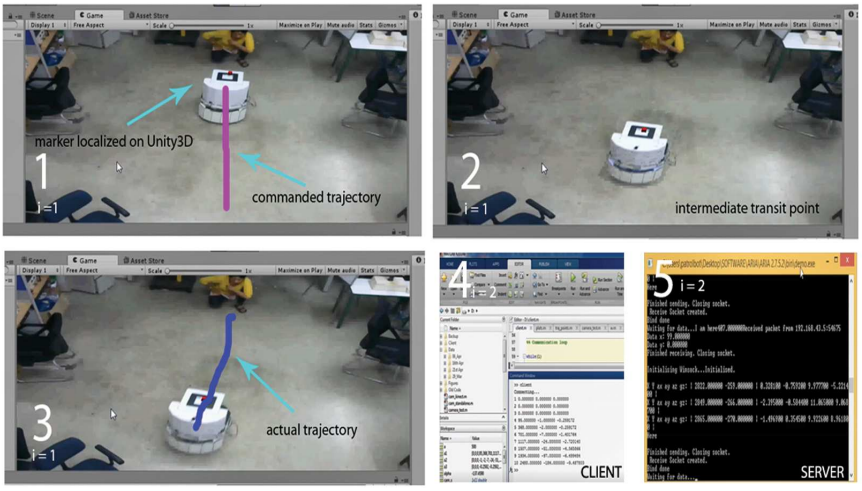

Figure 8.5: Typical experimental execution observed in Unity 3D with client and sever communication scripts.

available in the camera plane, which is subjected to a homogenous transformation for obtaining position and orientation information in the motion plane. The data is received by the client, transformed using a suitable transformation matrix and maps observed data onto the workspace by 2-dimensional projection. 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.

Prior to the experiments, we perform two calibrations to ensure correctness of data obtained from the camera and the IMU. First, a commanded trajectory is physically marked on the workspace and the camera data is verified by displacing the visual marker through a predetermined sequence of points on the commanded path. Linear and angular displacement offsets of the transformation matrix are tuned till desired accuracy is obtained. Secondly, the IMU is calibrated by static placement and independent axial motion experiments. The IMU offsets have been calibrated with an Arduino-Uno programmer using the open-source 'I2Cdevlib' libraries [7]. Calibrations ensure reliability of the experimental observations and help to minimize modeling and measurement errors, which means, deviations can

be safely attributed to slippage and not to the designed observer.

8.11 Algorithms

Updating the trajectory planning horizon, computing the optimal controls and localizing the target and the obstacles are the major activities of the client routine. The server, on the other hand, receives the desired or reference optimal trajectories from the client, performs slip estimation using IMU, encoders and optical flow data and generates local PID motion commands for the pursuer. Algorithm 1 represents a general routine for determination of optimal trajectories and controls by a navigation and guidance planner. The operations of the client and the server computers have been shown in Algorithm-2 and Algorithm-3 respectively.

Algorithm 1 Optimization

$t \leftarrow current\_time$

Start: *while (1)*

Receive $x(t), y(t), x_{tar}(t), y_{tar}(t), x^i_{obs}(t), y^i_{obs}(t), t, i \in [1, m]$

Initial boundary values $\leftarrow x(t), y(t)$

$T_{int} \leftarrow T_{int} - t, t \leftarrow 0$

if Target/obstacle update available **then**

  goto close

end if

Estimate: (Intent Awareness)$x_{tar}(t_1), y_{tar}(t_1), x^i_{obs}(t_1), y^i_{obs}(t_1), t_1 \in (t, T_{int}]$

Terminal Boundary values $\leftarrow x_{tar}(T_{int}), y_{tar}(T_{int}), \Psi(T_{int}) = [0 \;\; 0]^\top$

Assign: w_v, w_d, w_b

Minimize J

*Compute:*optimal $x^*(t_1), y^*(t_1), v^*(t_1), \theta^*(t_1), t_1 \in (t, T_{int}]$

close

Algorithm 2 Client

Activate: *UDP_Receiver, UDP_Sender*

Initialize $T_{int}, T_{update}, tol, current\_time$

$i \leftarrow 0, flag \leftarrow 0$

Start:*while (1)*

if $current\_time \geq T_{update}$ **then**

$(x_{tar}(0), y_{tar}(0)) \leftarrow (x_{cam}(i), y_{cam}(i))$

$(x(0), y(0)) \leftarrow (x_{enc}(i), y_{enc}(i))$

$T_{int} \leftarrow T_{int} - current\_time$

Compute:optimal$(x, y), (v, \theta)$

$dist := \|(x - x_{tar}, y - y_{tar})\|_2$

end if

if $dist \leq tol$ **then**

$flag \leftarrow 1$

end if

$UDP\_DataSent \leftarrow (x, y, v, \theta, T_{int}, flag)$

if flag $= 0$ **then**

$current\_time \leftarrow 0$

$i \leftarrow i + 1$, **goto Start**

end if

close

enc : encoder.

cam : camera.

T_{update} : Planning horizon update rate.

tol : Tolerance value of closeness with target.

Algorithm 3 Server

Activate: *Robot*

Initialize: T_{update}, T_{idle}, t

$i \leftarrow 1, flag \leftarrow 0$

Start: *while (1)*

$CurrentPose \leftarrow (x_{enc}(i), y_{enc}(i))$

Open: *UDP_SendSocket*

$SendBuffer \leftarrow CurrentPose$

$t \leftarrow 0$

Close: *UDP_SendSocket*

Open: *UDP_ReceiveSocket*

if $t \geq T_{update}$ **then**

goto Reset

end if

if $t \geq T_{idle}$ and $RBuff = NULL$ **then**

goto Reset

end if

if $t < T_{idle}$ and $RBuff \neq NULL$ **then**

if $flag = 1$ **then**

break

else

$Robot \leftarrow (v, \theta, T_{int})$

Reset:

Close: *UDP_ReceiveSocket*

Open: *UDP_SendSocket*

$i \leftarrow i + 1$, **goto Start**

end if

end if

close

$RBuff$: Receive Buffer

T_{idle} : Idle Time

Bibliography

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

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

[3] DK. Cho, SH. Chun, JG. Ahn, YS. Kwon, JH. 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*, 1–4 (2009).

[4] Y. Wang, C. Wu, X. Zhang, B. Li and Y. Zhang, "An embedded wireless transmission system based on the extended user datagram protocol (EUDP)," *Proceedings of ICFCC'10: IEEE International Conference on Future Computer and Communication*, 690–693 (2010).

[5] Y. Zheng and P. Moore, "The design of time-optimal control for two-wheel driven carts tracking a moving target," **In**: *Proceedings of CDC'95: IEEE Conference on Decision and Control*, 3831–3836 (1995).

[6] 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).

[7] InvenSense, "https://www.github.com/jrowberg/i2cdevlib/tree/master/Arduino/MPU6050," (accessed October 13, 2020).

List of publications

Journal Publications:

1. Biswas, K., Kar, I., Feron, E., "A Priority Driven Optimal Guidance Approach to Moving Target Tracking for Safe Navigation in a Narrow Space", *International Journal of Systems Science*, Taylor & Francis, 2021.

2. Biswas, K., Kar, I., "Validating Observer Based On-line Slip Estimation for Improved Navigation by a Mobile Robot", *International Journal of Intelligent Robotics and Applications*, Springer, 2022.

3. Biswas, K., Kar, I., Feron, E., "Intent-Aware Optimal Collision Avoidance and Trajectory Planning for a Pursuit Vehicle", *Robotica*, Cambridge University Press, 2022.

Conference Publications:

1. Biswas, K. and Kar, I., "On reduction of oscillations in target tracking by artificial potential field method', *Proceedings of 9th International Conference on Industrial and Information Systems*, Gwalior, India, 1-6, December 2014.

2. Biswas, K. and Kar, I. "An Optimal Solution to Fixed Time Horizon Moving Target Tracking with Obstacle Avoidance', *Proceedings of European Control Conference*, Alborg, Denmark, 1610-1615, July 2016.

3. Biswas, K., Kundu, A.S. and Kar, I., "Real-time Energy-Optimal Moving Target Tracking by Holonomic Vehicle', *Proceedings of 13th International IEEE India Conference, INDICON*, Bangalore, India, 1-6, December 2016.

4. Biswas, K. and Kar, I., "Maximum Principle in Finding Free final-Time Optimal Trajectory of Mobile Robot for Moving Target Tracking'. *Proceedings of 3rd Indian Control Conference*, Guwahati, India, 117-122, , January 2017.