Learning of Unknown Environments in Goal-Directed Guidance and Navigation Tasks: Autonomous Systems and Humans

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Abhishek Verma

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Doctor of Philosophy

Professor Bérénice Mettler, Advisor

December, 2017

Acknowledgements

There are many individuals that have earned my sincere gratitude for their contribution to my successful completion of doctorate research and a memorable time and learning experience in graduate school.

First and foremost, to my advisor, Prof. Bérénice Mettler, who introduced me to the interesting research area of human spatial navigation capabilities, I would like to express my sincere gratitudes. She guided me with her bright insights and novel experience in the area. She helped me throughout my research period in formulating research questions and approach to study those questions. She also gave her valuable time to help me edit various conference and journal articles, and helped me improve my writing and communication skills significantly. She has been an inspiration to me as a scientist and a dream-follower. I truly enjoyed working with her and I wish our collaboration lasts forever.

To my dissertation committee members, Prof. Peter Seiler, Prof. Yohannes Ketema, and Prof. Tryphon Georgiou, I would like to express sincere gratitudes for your availability, giving time to read my dissertation, and giving valuable advice to improve the dissertation.

To my colleagues and friends, Navid Dadkhah, Jon Andersh, Zhaodan Kong, Bin Li, Andrew Feit, and Kuo-Shih Tseng, at the Interactive Guidance and Control Lab, I would like to thank all of you for sharing with me your insights and making my graduate research experience memorable. I learned a lot of skills through our various group discussions and lab presentations.

I would also like to thank the administrative staff of the Department of Aerospace Engineering and Mechanics, the Graduate School, the International Student and Scholar Services and the Center for Writing. They have been supportive and helpful.

At last but most importantly, to my mother Sushma Verma and my father Virendra Singh, who devoted their lives and sacrificed their comforts to educate me, I can not express enough gratitude. Coming from a low income family, they made sure that I got proper education. They always supported my goals and dreams. Without their devotion and sacrifice, I could not be here pursuing my dreams. To them, I owe everything I have achieved and will achieve.

Dedication

To my parents

Abstract

Guidance and navigation in unknown environments requires learning of the task environment simultaneous to path planning. Autonomous guidance in unknown environments requires a real-time integration of environment sensing, mapping, planning, trajectory generation, and tracking. For brute force optimal control, the spatial environment should be mapped accurately. The real-world environments are in general cluttered, complex, unknown, and uncertain. An accurate model of such environments requires to store an enormous amount of information and then that information has to be processed in optimal control formulation, which is not computationally cheap and efficient for online operations of autonomous guidance systems. On the contrary, humans and animals are in general able to navigate efficiently in unknown, complex, and cluttered environments. Like autonomous guidance systems, humans and animals also do not have unlimited information processing and sensing capacities due to their biological and physical constraints. Therefore, it is relevant to understand cognitive mechanisms that help humans learn and navigate efficiently in unknown environments. Such understanding can help to design planning algorithms that are computationally efficient as well as better understand how to improve human-machine interfaces in particular between operators and autonomous agents. This dissertation is organized in three parts: 1) computational investigation of environment learning in guidance and navigation (chapters 3 and 4), 2) investigation of human environment learning in guidance tasks (chapters 5 and 6), and 3) autonomous guidance framework based on a graph representation of environment using subgoals that are invariants in agent-environment interactions (chapter 7).

In the first part, the dissertation presents a computational framework for learning autonomous guidance behavior in unknown or partially known environments. The learning framework uses a receding horizon trajectory optimization associated with a spatial value function (SVF). The SVF describes optimal (e.g. minimum time) guidance behavior represented as cost and velocity at any point in geographical space to reach a specified goal state. For guidance in unknown environments, a local SVF based on

current vehicle state is updated online using environment data from onboard exteroceptive sensors. The proposed learning framework has the advantage in that it learns information directly relevant to the optimal guidance and control behavior enabling optimal trajectory planning in unknown or partially known environments. The learning framework is evaluated by measuring performance over successive runs in a 3-D indoor flight simulation. The test vehicle in the simulations is a Blade-Cx2 coaxial miniature helicopter. The environment is a priori unknown to the learning system. The dissertation investigates changes in performance, dynamic behavior, SVF, and control behavior in body frame, as a result of learning over successive runs.

In the second part, the dissertation focuses on modeling and evaluating how a human operator learns an unknown task environment in goal-directed navigation tasks. Previous studies have showed that human pilots organize their guidance and perceptual behavior using the interaction patterns (IPs), i.e., invariants in their sensory-motor processes in interactions with the task space. However, previous studies were performed in known environments. In this dissertation, the concept of IPs is used to build a modeling and analysis framework to investigate human environment learning and decision-making in navigation of unknown environments. This approach emphasizes the agent dynamics (e.g., a vehicle controlled by a human operator), which is not typical in simultaneous navigation and environment learning studies. The framework is applied to analyze human data from simulated first-person guidance experiments in an obstacle field. Subjects were asked to perform multiple trials and find minimum-time routes between prespecified start and goal locations without priori knowledge of the environment. They used a joystick to control flight behavior and navigate in the environment.

In the third part, the subgoal graph framework used to model and evaluate humans is extended to an autonomous guidance algorithm for navigation in unknown environments. The autonomous guidance framework based on subgoal graph is an improvement to the SVF based guidance and learning framework presented in the first part. The latter uses a grid representation of the environment, which is computationally costly in comparison to the graph based guidance model.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Humans are capable of learning complex unknown environments in a variety of guidance tasks and use the knowledge to determine near-optimal (e.g., minimum-time) performance and remain versatile and adaptive to unexpected changes in the environment. This capability is not unique to spatial environment navigation but is also essential to other spatial tasks involving interactions with the environment such as pertaining to surgery. The general goal of this research is to understand how humans achieve efficient environment learning and path-planning capabilities despite their limited sensing, information processing, and memory capabilities. Such understanding can help to improve planning algorithms to be computationally efficient and adaptive to changes in the task environment as well as better understand how to improve human-machine interfaces in particular between operators and teleoperated or autonomous agents.

An agile guidance task in an unknown environment primarily involves three steps. The first step is environment sensing and assimilating the sensed environment information into global knowledge. The second step is path planning, i.e., trajectory optimization, using the known/learned knowledge and planning an immediate trajectory. The last step is tracking the planned trajectory. The three steps are repeated online. An autonomous guidance operation requires a mechanism for sensing and learning the environment and representing the learned environmental information in computationally efficient ways, in order to process online trajectory planning. Given the limited sensing and information processing capabilities of autonomous guidance systems, it is challenging to develop efficient learning and representation methods for real-world tasks

in spatial environments.

Humans have constraints on their memory, sensing, and information processing capabilities. Human limitations are limited field of view and visual attention, limited information processing (e.g., working memory), and perceptual guidance at sensing, planning, and control levels, respectively. Despite these limitations, they can navigate complicated unknown environments, exhibit efficient behavior in agile guidance tasks, and given enough trials, learn near-optimal solutions (e.g., minimum-time route between two places). Strengthening the knowledge base about environment learning in humans could be a key to overcoming computational complexities in autonomous guidance systems that arise from having to process enormous amount of information available in real-world task environments. This research investigates principles that underlie robust human guidance, learning process/mechanism, and memory structure in dynamic spatial behavior/navigation of unknown environments.

Humans and other animals have evolved a system of processes to navigate and interact with their environments. Gibson [1] introduced the idea that spatial behaviors are mediated by affordances, available through the interaction with task and environmental elements. For example, Lee [2, 3, 4] showed how principles like time-to-closure of a gap, e.g., distance, angle, force, etc., and optical flow can be used to regulate motion, rather than requiring complex models and computations. Tau-control and optical flow show how humans and animals use limited cues from the environment, which help them overcome their perceptual and information processing constraints. Inspired by the concept of affordances and limited cues, this research investigates cues and affordances used by humans to navigate unknown environments.

The central concept this research is based on is "invariants". Simon [5] quoted "The fundamental goal of science is to find invariants". An invariant can break down a complex problem into smaller subproblems such that a similar solution can be used for a set of subproblems. For guidance tasks in spatial environments, previous studies [6, 7, 8] with human pilots operating remote-control miniature rotorcraft showed that pilots organize spatial behavior by using invariants in their sensory-motor behavior (guidance, control, and perceptual processes) in interaction with the spatial environment and task elements. The invariants in sensory-motor behavior are called interaction patterns (IPs) as these emerge from interactions between the agent and the task environment. IPs are

transferable to similar task domains via symmetry transformations such as rigid-body transformation (translation, rotation, and reflection), which mitigates a guidance task complexity. Mettler et. al [8] proposed that IPs function as units of organization for planning spatial behavior in guidance tasks. Previously, IPs have been studied and used for modelling human guidance behavior in known environments. This dissertation investigates what functions IPs play in human learning during goal-directed guidance tasks in unknown environments. The dissertation builds on IPs to propose a framework that allows to formally investigate human environment learning in agile guidance tasks.

1.1 Preceding Work

This section briefly reviews the concepts of spatial value function (SVF), interaction patterns (IPs), and hierarchical model of human pilots' guidance and perceptual behavior. The study of autonomous and human environment learning presented in this dissertation is based on the concepts of SVF, IPs, and hierarchical model of human guidance behavior.

1.1.1 Spatial Value Function (SVF)

For a trajectory optimization problem in which a vehicle has to reach a specified goal state $\mathbf{x}_g$ from a start state, spatial value function (SVF) [9] describes optimal cost-to-go (CTG) and velocity vector field (VVF) over a geographical space. A detailed mathematical formulation of SVF is given in Chapter 3.

1.1.2 Spatial Structures (Patterns) in SVF

Kong and Mettler [10] described structural features (subgoals, repelling and attracting manifolds) in the SVF. They investigated these elements using a toy example based on the optimal solution for a Dubins vehicle that has to reach a goal in an obstacle field, as shown in Fig. 1.1 from [11]. Subgoals partition a task space such that optimal solution in each partition converges to a subgoal. A common boundary of two space partitions is defined as either repelling or attracting manifold. Velocities converge and diverge along attracting and repelling manifolds, respectively. These features make it possible to abstract the solution. The entire solution, i.e., SVF, can be described as a directed

graph of subgoals. The subgoal graph representation of task space accounts for both the vehicle dynamics and environment. Trajectory to the goal from any location in the task space can be represented by a subgoal sequence.

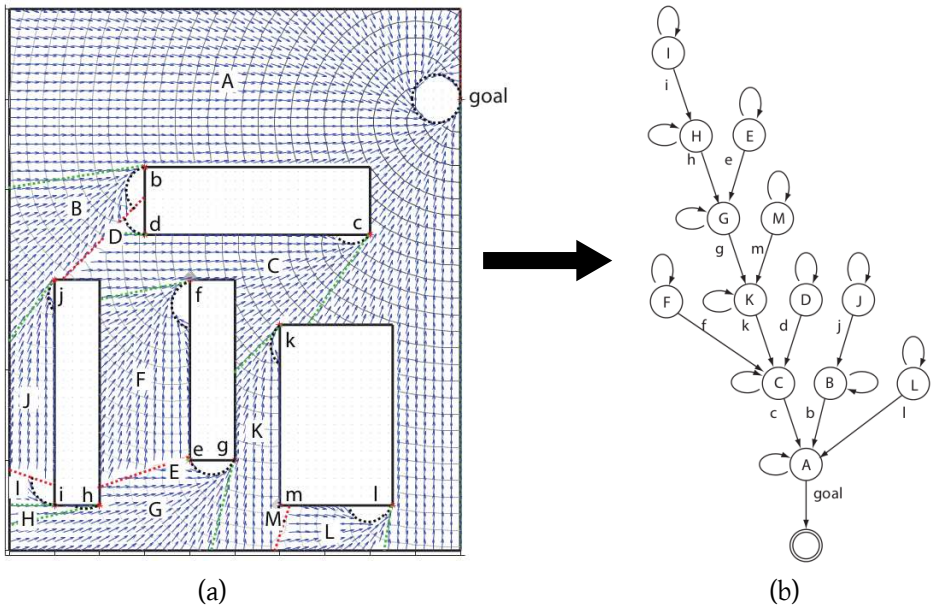


Figure 1.1: Example of spatial structures in spatial value function (SVF): (a) partition of the task space for a Dubins optimal solution using spatial structures (subgoals, repelling and attracting manifolds) and (b) graph representation of the task space based on the spatial structures. The figures are from [11].

1.1.3 Human SVF

Spatial value function (SVF) describes spatial guidance behavior associated with an optimal guidance policy (e.g., cost and velocity maps over geographical space). Mettler and Kong [12] showed that the guidance behavior of a trained operator can be described as SVF. They described a method to extract SVF maps from experimental trajectories in a goal interception task. The extracted SVF maps were compared with an optimal policy based on a mass-point model. The results in [12] showed that guidance behavior of a trailed pilot was sufficiently stationary in time, and continuous and consistent over the space. Therefore, the concept of SVF is a valid tool for the analysis of human

guidance behavior. Kong and Mettler [7] subsequently extended the analysis to investigate the organization of guidance behavior over large task environments with obstacles. They suggested that humans exploit invariants in the dynamic interactions with the environment to mitigate complexity, which is discussed next.

1.1.4 Interaction Patterns: Human Pilot

The patterns described in Dubins solution space [10] are a result of interaction between vehicle dynamics and environment. To account for human operators in human-piloted guidance tasks, Kong and Mettler [7] used the concept of closed-loop agent-environment dynamics [13]. The authors applied the concept of "invariances" on closed-loop agent-environment system and described interaction patterns (IPs), i.e., subgoals and guidance primitives, that accounts for interactions of human operator's control, guidance, and perceptual mechanisms with the environment. A mathematical formulation for IPs is given in Chapter 5.

1.1.5 Functional Model of Human Guidance

Mettler et. al [8] combined the elements such as IPs and perceptual guidance to propose a hierarchical multi-loop model explaining organization of human guidance behavior. The hierarchical model delineates planning, perception, and control as shown in Fig. 1.2. At the highest-level, i.e. planning, a human pilot decomposes the global task into subtasks as a sequence of subgoals. To navigate between subgoals, the pilot deploys a series of guidance primitives. A guidance primitive is an invariant control profile coupled with particular perceptual processes (e.g., gaze movements). Thus a human pilot uses IPs as units to organize their guidance behavior in a task space. The goal of this research is to investigate the functions of planning, perceptual guidance, and control levels in the hierarchical model of human guidance behavior for learning new environments.

Mettler et. al [8] also presented a hierarchical model of perceptual behavior that models visual attention as a function of three levels (planning, perceptual guidance, and tracking and pursuit) in the hierarchical guidance model. Andersh et. al [14] tested the hypothesis based on the functional model. The authors investigated visuo-motor

control in a remote-control goal-interception task. The analysis showed that pilots' gaze follow the vehicle. In between, pilots use saccades to rapidly switch gaze to the goal location and fixate gaze at the goal for a small duration. The smooth pursuit, i.e., gaze following the vehicle, and saccades provide estimates of vehicle velocity and motion gap to the goal location, respectively.

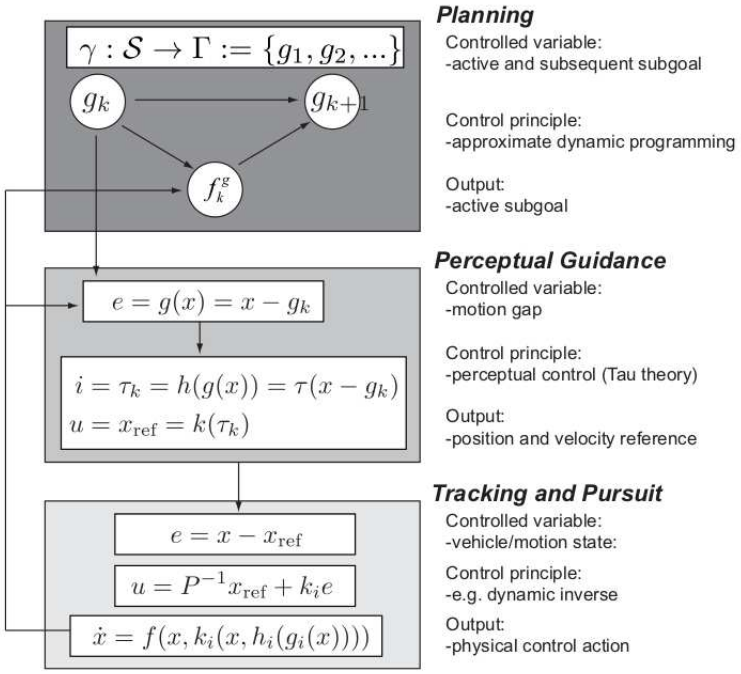


Figure 1.2: Hierarchical multi-loop model of human guidance behavior proposed in [8].

1.2 Research Questions

Previous studies [7, 8] focused on functions of IPs in known environments. They used humans as a remote pilot, which enables a third-person view for the guidance task. The task environment was known and in field of view of a pilot. However, in most real-world tasks, the environment is not fully known and only partially visible (in field of view). Therefore, the planning of subgoal sequence requires memory, learning, and decision-making processes. This research investigates the role of IPs in learning and navigation of unknown environments.

The general goal is to model cognitive functions that facilitate environment learning in spatial guidance tasks in humans. Downs and Stea [15] gave a formal definition of cognitive mapping: "Cognitive mapping is a process composed of a series of psychological transformations by which an individual acquires, codes, stores, recalls, and decodes information about the relative locations and attributes of phenomena in his everyday spatial environment." Following the definition in [15], this dissertation formulates specific questions for environment learning in spatial guidance tasks in humans. The questions are: 1) what information is extracted from interactions with the environment?, 2) what is the memory structure for coding and storing the information?, and 3) how the information is represented to support planning and decision-making?

1.3 Hypothesis

The hypothesis for guidance tasks in unknown environments is that interaction patterns serve as units of organization for learning the task (e.g., learning an optimized such as time-optimal behavior in the environment). A human pilot uses IPs to abstract the task environment as a graph network of subgoals. A skilled pilot learns guidance primitives that represent trajectory maneuvers optimized for a given cost function (e.g., time). For the guidance primitive of the skilled pilot, perceptual and control policies are linked in a way that allows the pilot to focus his/her attention at task-relevant features of the environment. The optimal guidance primitives are required for the learning of the optimal subgoals and their network. The subgoal graph enables the pilot to layout the global plan as a sequence of subgoals and implement control as a series of guidance primitives.

1.4 Approach Overview

This section briefly describes the approach and experiment systems, used in this dissertation, for studying the learning of new environments in autonomous and human guidance tasks.

1.4.1 Autonomous Guidance

The mathematical formulation and details of the guidance framework and experiment system are given in Chapter 3. This section gives only a brief overview.

The autonomous guidance framework is based on a receding horizon trajectory optimization with SVF [9]. The framework consists of an online planning cycle that has three primary steps as follows. First, a local (in a neighborhood of vehicle's current position) SVF is updated online using the environment data sensed from exteroceptive sensors. Second, the updated SVF map is used to compute an intermediate goal point in the local neighborhood. Third, an online planner uses a numerical trajectory optimization method to travel to the intermediate goal.

For simulations, the dissertation uses a detailed 3-D indoor environment that is unknown to the guidance system a priori. Successive runs are simulated between specified start and goal states. The approach in this research is based on trajectory optimization, therefore the quality of the solution depends on how close the solution is to the optimal trajectory. In this research, time-to-go is used as objective function. Therefore, an optimal trajectory corresponds to the minimum time trajectory. Other cost functions such as path length, energy, or a function of them, can also be used, which will be mentioned in an overview of SVF computation using quantized state-space in Chapter 3. The simulations presented in this dissertation uses a discrete-time linear state-space model of a Blade-Cx2 coaxial miniature helicopter.

1.4.2 Human Guidance

The details of the analysis framework and experiment system to study human learning of new environments in agile guidance tasks are given in Chapter 5. This section gives only a brief overview.

The hypothesis is that the interaction patterns provide the mechanisms needed to abstract a task environment. The task space is modeled as a graph network of subgoals (IPs). The graph model is applied to investigate how subjects learn the task structure and optimal behavior. The solution from a Dijkstra's shortest path formulation is used as a baseline to evaluate decision-making (subgoal selection) process in human subjects. The framework applies a hierarchical clustering method on trajectory data to identify

guidance primitives.

To study task environment learning in humans, the dissertation uses a simulated environment proposed by Feit and Mettler [16] for first-person guidance experiments. The system uses a monitor to display a simulated environment (a maze made of vertical walls) unknown to a human subject. A subject can navigate in the environment using a joystick that simulates a vehicle with unicycle dynamics. The system records the control inputs, vehicle trajectory, and human gaze location in the 3-D environment displayed on the screen.

In this dissertation, the hypothesis for human environment learning is tested by evaluating human guidance data using a benchmark subgoal graph and extracting guidance primitives from the data. If the hypothesis that interaction patterns aid task learning is correct, a skilled pilot will show emergence of focused, distinct, and cost-optimal interaction patterns than a novice pilot.

1.5 Dissertation Outline and Contributions

Figure 1.3 shows the organization of this dissertation. The contributions of this dissertation are presented next.

1.5.1 Computational Investigation of Environment Learning (Chapters 3 and 4)

Guidance of an autonomous UAV (unmanned aerial vehicle) can be formulated as a trajectory optimization problem using optimal control [17]. Such formulations, however, are NP-hard [18]. For real-time guidance applications in unknown and uncertain environments, the optimization problem has to be solved repeatedly and online. Therefore, an entire (infinite-horizon) trajectory optimization is not a practical solution for online operations. Receding horizon (RH) control [19, 20, 21, 22, 9, 23] is a practical solution to UAV guidance problems. In RH planning, a finite-horizon trajectory optimization based on current vehicle and immediate environment states is solved repeatedly. A cost-to-go (CTG) function is used to approximate the cumulative cost of discarded tail of the global trajectory. The CTG accounts for the global environment and task parameters. The minimum of total cost (CTG + cost of the finite-horizon trajectory) gives a target

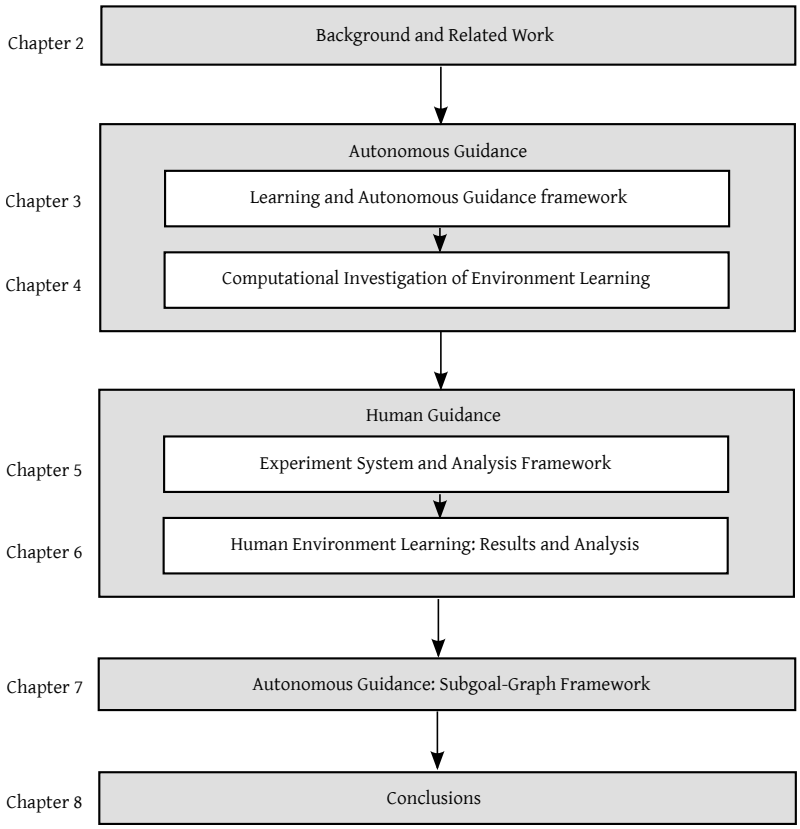


Figure 1.3: Dissertation outline.

point called the active waypoint (AWP) [23]. The RH planner follows the finite-horizon trajectory to the AWP. The closer the cost-to-go is to its true value, the more optimal (closer to the infinite-horizon optimal control solution) the receding horizon solution and the shorter the planning horizon is [22].

In the past, a number of approaches have been used to compute the CTG map for RH planning. Bellingham et. al [19] used a CTG function based on a visibility graph to account for obstacles. The CTG function in [19] is defined at the graph's vertices, using a shortest path algorithm. Mettler and Bachelder [20] used an offline computed CTG map based on a cell decomposition of the 3-D environment, which was related to vehicle turning radius and flight path angle. The CTG map in [20] incorporates vehicle state information. The offline CTG map is used for online RH planning. Mettler et. al [9]

introduced the concept of spatial value function (SVF) that describes optimal cost-to-go (CTG) and velocity vector at a point in geographical space. The guidance policy over geographical space can be called as spatial guidance behavior. The SVF accounts for effects of vehicle dynamics and environment characteristics such as obstacles layout and length-scale. Note that the SVF term refers to the CTG and both terms are used interchangeably in the dissertation. An approximate optimal SVF map is computed using discretized state-space (geographical space and vehicle dynamics) and dynamic programming [9].

When the environment is unknown or partially known, the CTG map has to be updated repeatedly as the task progresses and the environment is learned through on-board sensors. Mettler et. al [9] presented a sensory-predictive guidance system based on an integration of RH trajectory optimization and SVF. The integration of sensing and planning processes in [9] is done using a local SVF map that is defined over the domain of online trajectory optimization in RH planning. The local SVF map adapts to online sensory data. The sensory-predictive guidance system in [9] enables adaptive behavior, based both on local and global information. The learning autonomous guidance framework presented in Chapter 3 is based upon the sensory-predictive guidance system in [9].

The previous work [9, 23], however, did not study the guidance performance coupled with the learning process. This dissertation uses the proposed framework to formally investigate this learning process, and simultaneously analyze the associated information processing (propagation and assimilation) over repeated local SVF updates and successive runs. With this approach it is possible to determine how guidance performance and various aspects of dynamic behavior evolve as the SVF is learned over successive runs. The analysis leads to a better understanding of the type of abstractions needed to achieve sparse description of guidance policy over complex geographical environments.

Another issue this research addresses is that the AWP selection process, in the previous approaches [20, 21, 22, 9, 23] , requires solving the online trajectory optimization for a set of points (AWP candidates) in the local CTG map. Numerical optimization such as nonlinear programming would be intractable. In this dissertation an approximate cost-to-come (CTC) map is introduced. This map is computed using the same set of quantized state-space as for the CTG function, however, it only spans the reachable

space (which is contained in the sensory space), i.e., the space vehicle can reach from its current state within a finite time horizon. The CTC map provides a lower bound on the actual cost-to-come and allows efficient computation of the AWP. This information is then used for the trajectory generation. Note that with this framework the CTC could also be learned or improved based on performance data. The framework also provides the capability to validate the performance of the finite-horizon trajectory with respect to the CTC map.

1.5.2 Human Environment Learning (Chapters 5 and 6)

This research investigates how humans learn a new task environment in goal-directed guidance tasks. The dissertation presents an analysis framework that models the task space as a graph network of subgoals. The graph framework provides a formal assessment of task environment learning by tracking the emergence of subgoals, connectivity between subgoals, and convergence of CTG at subgoals. The framework uses a hierarchical clustering method to identify guidance primitives (IPs) in human trajectory data. The method is applied to track the emergence of guidance primitives as a subject learns the task environment over successive trials. The dissertation presents a comparison study between skilled and novice pilots. The study highlights the characteristics of a skilled subject, which assist skilled subjects in efficient environment learning and performance.

1.5.3 Autonomous Guidance: Subgoal-Graph Framework (Chapter 7)

Chapter 7 uses the subgoal-graph framework used to analyze human environment learning in guidance tasks to present an autonomous guidance framework for navigation in unknown environments. The framework learns a topographical representation of the task space. The method is an improvement to the autonomous guidance and learning framework presented in Chapter 3, which learns the environment as SVF, i.e., a grid based representation of the task space. The graph representation is efficient in terms of storage memory requirements and computations.

1.6 List of Publications by Chapters

The following is the list of publications corresponding to each chapter:

- **Chapters 3 and 4**
  - Verma, A. and Mettler, B., "Learning Optimal Guidance Behavior in Unknown Environments within Receding Horizon Planning," American Helicopter Society 70th Conference, Montreal, Canada, 2014, Vol. 3, pp. 2158–2168.
  - Verma, A. and Mettler, B., "Computational Investigation of Environment Learning in Guidance and Navigation," Journal of Guidance, Control, and Dynamics, 2017, Vol. 40, Special Issue on Computational Guidance and Control, pp. 371-389.
  - Verma, A. and Mettler, B., "Scaling Effects in Guidance Performance in Confined Environments," Journal of Guidance, Control, and Dynamics, Vol. 39, No. 7, 2016, pp. 1527-1538.
- **Chapters 5 and 6**
  - Verma, A., Feit, A., and Mettler, B., "Investigation of Human First-Person Guidance Strategy from Gaze Tracking Data," Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on, Hong Kong, Oct. 912, 2015, pp. 1066–1072.
  - Verma, A. and Mettler, B., "Analysis of Human Guidance and Perceptual Behavior in Navigation of Unknown Environments," American Helicopter Society 72th Conference, Florida, USA, 2016.
  - Verma, A. and Mettler, B., "Investigating Human Learning and Decision-Making in Navigation of Unknown Environments," 1st IFAC Conference on Cyber-Physical & Human-Systems, Florianopolis, Brazil, 2016.
  - Verma, A. and Mettler, B., "Human Learning of Unknown Environments in Agile Guidance Tasks," arXiv:1710.07757, 2017.

Chapter 2

Research Background

This chapter provides the background on autonomous guidance and models of human guidance behavior to highlight the significance of the formulation and approach used in this dissertation. Section 2.1 gives an overview of autonomous path planning techniques in unknown environments. Section 2.2 discusses existing models of human drivers/pilots, visual guidance, spatial navigation, decision-making, and cognitive limitations (e.g., working memory).

2.1 Background: Autonomous Guidance

A number of motion planning algorithms for autonomous UAV guidance have been proposed in the past (see the survey paper by Goerzen et. al [24]). This section briefly reviews the graph search techniques, practical planning techniques for dynamic systems, learning and path planning, optimization approaches (such as model predictive or receding horizon control), SLAM (simultaneous localization and mapping), and topological map learning methods. This section is taken from article [25].

2.1.1 Graph Search Techniques

Graph search techniques are common for robotic path planning [26]. Dijkstra's [27] and A* [28] algorithms are used for path planning in known terrains. These graph search methods use a heuristic (e.g. distance) to anticipate the remaining cost to reach the goal, and minimize the sum of the anticipated cost and the cost of the path travelled so

far. For path planning in unknown terrains, the path has to be recomputed (updated) iteratively as the task progresses and unknown obstacles are revealed through onboard sensors. Stentz [29] proposed the D* algorithm for optimal and efficient re-planning in partially known environments. D* is a dynamic version of A*, which updates local path based on the local changes in the environment rather than re-computing the entire path from start to goal.

Graph search methods have been applied for path-planning of UAVs. For example, Bortoff [30] proposed a two step path-planning algorithm for UAVs to trade-off stealth versus path length through a set of enemy radar sites. The first step is to find an optimal path on Voronoi graph built around the radar locations. The second step accounts for the UAV dynamics using the graph solution as initial conditions. Bellingham et. al [31] presented a method based on visibility graph for an optimal task allocation of a fleet of UAVs to visit specified waypoints, minimizing a cost function such as time. Graph search techniques in general provide an optimal sequence of nodes (spatial waypoints) in a graph but do not account for vehicle dynamics and limits on maneuvering capabilities.

2.1.2 Practical Planning Techniques for Dynamic Systems

Path planning for autonomous vehicles requires techniques that can generate dynamically feasible plans exploiting vehicle dynamic capabilities. Fox et al. [32] proposed a dynamic window approach derived from the motion dynamics of a robot. The dynamic window refers to a reduced velocity space consisting of velocities that are reachable within a short time interval from the current velocity state, considering its acceleration constraints. LaValle and Kuffner [33] described a method for kinodynamic planning in the configuration space using rapidly exploring random trees (RRT). RRT based methods try to find the shortest feasible path in the vehicle's visible set by randomly sampling control actions. Techniques in [32, 33] directly account for vehicle dynamics in path planning unlike graph search methods. However, these techniques are ad hoc in their formulation and implementation [9] and don't provide a formulation for learning the optimal guidance behavior.

2.1.3 Learning Techniques

Learning techniques have been applied for autonomous guidance in unknown complex environments [34, 35, 36]. Michels et al. [34] used supervised learning to estimate depth of a scene using monocular visual cues on single images of outdoor environments, and reinforcement learning to generate steering commands based on the depth estimates. Abbeel et. al [35] trained a controller on human pilot demonstrations of helicopter maneuvers. The controller was demonstrated on a real RC helicopter to perform autonomous aerobatic maneuvers. Reinforcement learning applied in [35] is focused on control and maneuvering, which is disconnected from the environment. Richards and Boyle [36] combined receding horizon control with reinforcement learning, which learns a cost function and improve performance over multiple surveillance tasks in known environments. The method in [36] randomly switches between exploration and exploitation, and the probability of exploration decays as more trials are performed. The authors in [36] tested the algorithm in a known 2-D obstacle field. The cost-to-go function used in [36] is based on the Euclidean distance from the goal, which does not account for the vehicle dynamics.

This dissertation uses the SVF [9] that accounts for vehicle dynamics as well as environmental constraints. An unknown complex 3-D environment is used to demonstrate learning processes. The dissertation investigates the evolution of SVF with learning, and what spatial features emerge in SVF as a result of learning.

2.1.4 Optimization Based Techniques

Optimization based techniques, with state and control constraints, for trajectory generation are costly and impractical to use online, if the trajectory over an entire problem space has to be optimized at once. Model predictive control [37, 38] uses a model of a system to predict the future evolution of the system. A performance index is optimized with respect to a sequence of future moves, given the operating constraints. The first of such optimal moves is applied to the system and the process is repeated at each time step. Such a method makes it practical to account for constraints on states and controls in online operations [37]. Model predictive control has been applied to trajectory planning of UAVs (e.g. [39, 40]).

Receding horizon (RH) trajectory optimization is similar to model predictive control. In RH optimization, a finite-horizon trajectory optimization problem is solved repeatedly based on the current vehicle state and immediate environment. Successful results have been achieved in simulation and experiments on a number of aerial vehicle platforms [41, 42, 43, 44]. However, as pointed out in [9], one of the key challenges in the formulation of RH planning is the selection of cost-to-go (CTG) function used to approximate the cumulative cost of the discarded tail of trajectory. The closer the CTG is to the actual CTG, the closer the RH based approximation of the original optimal control problem is to the infinite-horizon optimal control problem [22].

Mettler et. al [9] presented the concept of spatial value function (SVF) relating the optimal guidance solution and geographical space. The SVF describes the optimal cost-to-go from a given point in geographical space to the goal, accounting for obstacles and vehicle dynamics. It represents the complete information needed for guidance throughout the geographical space. An approximate optimal SVF is computed using quantized state-space and dynamic programming [9], which is described in Chapter 3. The authors in [9] presented a sensory-predictive guidance system that integrates SVF with RH planning, using the concept of local SVF. Onboard sensory data is fused with the local SVF to integrate sensing and planning processes. Dadkhah and Mettler [23] used a risk map update function to iteratively update the environmental occupancy probability map based on sensory data. The updated occupancy map is used to update the local SVF map using a dynamic version of Dijkstra's algorithm [45]. This dissertation evaluates and validates the sensory-predictive guidance system using an indoor flight simulation setup. The study shows that a global convergence is achieved through local updates across a number of successive runs.

2.1.5 SLAM (Simultaneous Localization and Mapping)

SLAM [46] is a technique to simultaneously map an environment using sensors onboard a vehicle and keep track of its position in the environment. An accurate estimate of the vehicle position is required to build an accurate map of the environment and vice versa. Solving for both the vehicle position and the environment map is not trivial in presence of noise in vehicle's navigation and errors in sensors. SLAM techniques use iterative feedback between environment mapping and vehicle position estimation to increase the

consistency between both vehicle position and environment map estimates. SLAM has been applied for environment learning and autonomous navigation of robots and UAVs in unknown environments [47, 48].

SLAM in general focuses on learning the geographical aspects (topology, landmarks etc.) of the environment and vehicle state in the environment. This dissertation uses an approach that learns optimal guidance policy over geographical space. In uncertain and unknown environments, a SLAM algorithm has to be used to produce the local environment data that is assimilated into the local SVF by the proposed learning framework. However, integration of the learning framework with SLAM is out of scope for current study. To focus on behavior learning processes, it is assumed that the vehicle state is known.

2.1.6 Topological Map Learning

Meyer and Filliat [49] presented a survey on topological map learning methods. Topological maps abstract spatial environments as graphs describing relationships between different landmark locations. A common-sense knowledge of space is a topological description of paths and places [50]. Topographical maps have been used for path planning, e.g., [51, 52, 53, 54]. Mataric [51, 52] used a dynamic approach to detect landmarks for topological mapping. In the dynamic approach, onboard sensors monitor for consistencies in the sensory data as a robot moves next to objects in an environment. Thrun [53] used an approach in which topological maps are generated on top of grid-based maps. The approach in [53] combines properties from both maps, which are accuracy and efficiency for grid-based and topological maps, respectively. Ranganathan and Dellaert [54] presented a Bayesian inference method to estimate the posterior probability distribution on the space of all topologies, using online measurements. The authors in [54] demonstrated the algorithm with different sensors in different environments. Topographical maps using connectivity information between landmarks are an efficient representation of space for path planning. A question, however, is what environmental elements are critical to topological maps that account for vehicle dynamics and its interaction with the environment.

The learning framework presented herein uses grid-based maps since they allow to describe information about the spatial guidance behavior as SVF (CTG) map. The

understanding gained from the investigation of learning and information assimilation processes helps determine what environmental elements are relevant from a guidance perspective. This knowledge, in turn, can be used to define the appropriate data structure. These results will be discussed in Chapter 4 following the simulation results.

2.2 Background: Human Guidance, Navigation, and Decision-Making

This section gives an overview of existing models of human driver/pilot control, visual guidance, and cognition. The section also discusses background on human spatial navigation, memory, and representation. Next, the section provides a background on human decision-making, learning, and cognitive limitations such as working memory, which are relevant factors in learning of new environments and route planning.

2.2.1 Driver Modelling

Study and modelling of human drivers is relevant to improving automated vehicles and human-machine interfaces. A human driver has three primary components: cognition (planning), visual perception (visual guidance), and control. This section discusses models that focus on human control or cognitive processes. Visual guidance in humans is discussed in subsequent Section 2.2.2.

Optimal Control

In the past, human drivers/pilots have been generally assumed to operate as an optimal controller and modelled using conventional control techniques, e.g.,[55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66]. Tuskin [55] analyzed human tracking of moving targets and proposed a "nearest linear law", i.e., a human control input responds to instantaneous values and the rate of change of error. Mcruer [56, 58, 59] studied human control in certain closed-loop dynamic systems. McRuer and Krendel [56] proposed two models of human control, which are precognitive and pursuit. In precognitive model, a human can accurately predict the output based on an input, and therefore the human operates as an open-loop controller. In pursuit model, the human requires feedback such as visual

to manipulate its control, i.e., a closed-loop controller. McRuer and Jess [58] proposed a crossover model to describe that human pilot response has a limited bandwidth. Kleinman et. al [61] modelled humans' psychophysical limitations using time delays. Hess [62, 63, 64, 66] worked towards developing mathematical models of an adaptive human pilot. Hess [63] presented a human pilot model for pursuit tracking tasks. Hess and Modjtahedzadeh [64] used feedback control design principles to model a driver steering behavior for lane-keeping driving task on a curving road. Hess [66] presented a system identification of human pilot behavior in time-varying dynamic systems. It used a real-time, pilot-in-the loop simulation environment for the identification based on an output error model estimation algorithm. Zeyada and Hess [65] presented a framework to investigate how pilots perceive and utilize visual, proprioceptive, and vestibular cues in a ground-based flight simulator. This helped to develop a pair of metrics that can be used to assess a simulator fidelity.

Human behavior modeling as a control element gives a mathematical solution to certain tasks. Mathematical models of manual control, however, do not capture the total task of driving an automobile [67]. Therefore, mathematical models in general are not applicable for explaining human behavior in real-world spatial tasks such as navigating in real-world terrains (cities, forests, etc.), driving/piloting in a real-world environment, maneuvering a tool in a surgical task, etc. Also, it is not possible to comprehend human behavior and related cognitive mechanisms based on theoretic-control models.

Machine Learning

Machine learning techniques such as hidden markov models (e.g., [68, 69, 70]) and neural-networks (e.g., [71]) have been applied to model human driver control and perceptual behavior. Pentland and Liu [68] developed a computational model of human driver behavior using Hidden Markov Models. Oliver and Pentland [69] used a SmartCar with a real-time data acquisition system that recorded drivers' controls such as the brake, gear, and steering wheel angle. The system used video signals to capture a driver's head and viewpoint. The authors in [69] examined the experimental data, using the computational framework presented in [68], to learn Hidden Markov Models for driver maneuvers, such as turning and changing lanes. The models in [68, 69] predicted drivers' maneuvering behavior around 1 second before maneuvers took place, with a

high accuracy. Liu and Salvucci [70] used Hidden Markov model to predict a driver's intended actions based on a sequence of internal mental states. Each mental state consisted of a characteristic pattern of behavior and environmental state. The authors used the driver's visual scanning behavior as another source of information about the driver's state. Suzuki et. al [71] applied a neural-network modeling to analyze human-pilot control inputs during the landing phase in the visual approach on a flight simulator. The neural network was trained to simulate the inputs of a human pilot, based on the time history of visual cues and control inputs.

Machine learning methods are applicable to model real-world tasks and account for a large number of dimensions (a large portion of total task) unlike mathematical models. However, it is not simple to understand cognitive mechanisms related to driving tasks, using statistical models.

Cognitive Models

Driving is a multi-tasking activity including low-level control (e.g., steering, accelerating/braking) and high-level planning (e.g., maintaining situational awareness), and requires attention management among various elements [72]. For example, a driver has to regularly switch his/her attention between inside-car displays and scanning the road to maintain situational awareness. Therefore, cognitive psychology has been applied to model human driver behavior, e.g., [73, 74, 75, 76, 77]. For example, Bellet and others [73, 77] presented a cognitive architecture of driver's mental representation in car driving tasks. The architecture is primarily based on concepts of long-term and working memories, situational awareness, and action-perception cycle. Song et. al [74] presented a hybrid model that includes levels such as strategic, tactical, and operational. In the hybrid model [74], a driver uses the information from a perception module to decide the type (level) of behavior to activate.

Cognitive models are generally comprehensive and account for human factors (e.g., memory limitations), unlike theoretic-control or machine learning models. However, they do not explain versatile and agile spatial skills of a skilled human driver/pilot. This research is based on the concept that a skilled pilot exploits the invariants (patterns) in guidance, control, and perceptual processes in interactions with the task environment [8]. These invariants (interaction patterns) are used as units for organizing

planning, perception, and control behavior in the task space. These patterns exploit symmetries in the task space, which makes spatial planning robust and efficient.

Human-Inspired Controllers

Humans and animals are in general good at navigating around complex obstacles in real-world environments. In the past, researchers have studied obstacle avoidance behavior of humans. For example, Fajen and Warren [78, 79] presented the method of "point attractors" to model human obstacle avoidance behavior and route selection in goal-directed path planning. The point attractors model uses a superposition of goal attractions and obstacle repulsions. The method controls second derivative of steering angle unlike steering angle in potential methods in [80], which generates smooth trajectories. The authors concluded that in humans route selection can be modelled as local steering dynamics (obstacle avoidance behavior), and does not require an advance planning of global route based on an explicit representation of the world, i.e., a common approach in robotics. However, Patla et. al [81] showed that in cluttered environments humans route selection cannot be modelled as a reactive planning but involve a global planning such as visual scanning of clusters of obstacles and avoid them. The point attractors method has been used for modeling autonomous navigation in obstacles using human driving data (e.g., [82]). Expert human drivers such as race car drivers can make high-speed off-road turns. Such control skill is relevant for agile military vehicles operating in rough terrains. Huang et. al [83] presented an experimental platform and pattern recognition method for characterizing human control during tight turns with sliding in high-speed off-road driving. Burns et. al [84] used human control data in a simulated driving task in an obstacle field to train a receding horizon controller.

A drawback of human-inspired controllers such as "point attractors" is that the methods in general focus on steering and obstacle avoidance behavior, which is undoubtedly an important aspect in designing autonomous guidance systems, but do not address the problem of learning new environments and global task planning in large environments with limited environment visibility. The methods do not answer how humans despite limited memory and information processing capacities learn and plan agile guidance and navigation tasks in complex unknown spatial environments.

2.2.2 Visual Perception in Driving/Locomotion

Vision is a primary source of perception of the world in humans and most animals. This subsection gives a brief overview of studies about roles of human vision in guidance/driving tasks and self-locomotion.

Ecological Perception

Gibson [1] introduced the theory of "ecological perception". He argued that perception of visual world is not mere sensation of external (visual) stimuli but directly extracts information from the visual world, relevant to the activities. As Rosenbaum [85] quoted:

> The visual system, in this theory, need not decipher the structure of the external world by piecing together bits of visual evidence, as some have argued. Rather, according to this theory of *ecological perception*, the optic array contains adequate information to make the structure of the external environment immediately and unambiguously apparent.

The ecological perception provides affordances that are opportunities of action in an environment [1]. The interaction between an animal (human) and an environment is mediated by affordances. Perception is action-specific, i.e., the action abilities of a perceiver is reflected in perception [86].

Visual Kinesthesis

Vision has an important role in estimating one's own body movement, i.e., kinesthesis. Lee [87, 88] studied visual kinesthesis using a swinging room example in which human subjects stood in a room that was actually an inverted box made of four walls and ceiling. When the room (box) was swayed, the subject swayed too as he/she perceived himself/herself falling forwards or backwards. One explanation for the subjects' behavior in swinging room is the optic flow [89] that carries the information of subject's bearing relative to the environment.

Lee [2, 3] used the optic flow theory to present the concept of time-to-closure of a gap (e.g., distance, force, etc.). The author showed how the simplest type of information, i.e. time-to-collision, would be sufficient for a driver to control braking, rather than

information about distance, speed, or acceleration/deceleration. The time-to-closure gap reduces the problem complexity of interception tasks such as braking or landing.

Visually-controlled locomotion (or visual guidance) is discontinuous since the visual system of a species should be free to perform functions other than the locomotion [90, 91]. For example, experimental studies on moving hands to grasp objects [92] and catching a ball [93] have showed that visual-motor control is intermittent. Jeannerod and Prablanc [92] concluded that future trajectory is planned in advance and executed as open-loop without any visual input. The visual feedback is consulted again when more precision is required.

Role of Vision in Driving Tasks

Car driving has been a popular platform for studying the role of vision in guidance tasks. For example, Andersen and Sauer [94] proposed a DVA (driving by visual angle) model for car following. The model uses visual angle formed by the leading vehicle and the rate of change of the angle. Land and Lee [95] experimentally showed that when steering on a curvy road, drivers look at tangent point on the inside curve, around 1-2 seconds ahead. The hypothesis was that the visual direction to the tangent is an indicator for the curvature of the road and provides input for steering control. The tangent point hypothesis was supported by Kandil et. al [96], Mars [97], etc. A different theory for gaze behavior during steering on a curvy road and maintaining an instructed road position is that drivers look at a future point on planned trajectory and not the tangent [98]. However, if a driver does not have to maintain a particular position on road, the driver will cut the corner and look at the tangent point [95, 98].

Salvucci and Gray [99] presented a two-point visual control model of steering, which uses the visual direction of two points: 1) a 'near' point to monitor its lateral position and 2) a 'far' point to maintain lateral stability and infer steering angle to keep up with the upcoming road profile. The two-point visual control model of steering [99] was supported by experimental study presented by Neumann and Deml [100]. Vansteenkiste et. al [101] studied visual behavior in bicycle steering. The authors in [101] presented a gaze constraints model for goal-directed locomotion. The model in [101] is based on needs of direct control and anticipation. Need of direct control requires more visual attention near the vehicle and is used by novice drivers. Need of anticipation is required

for hazard perception and is used at high-speeds. In case of high needs, visual attention is mostly on relevant features due to increased attentional workload. When both needs are low, visual behavior is unconstrained according to the model.

Visual Attention in Multi-Tasking

Drivers in general multi-task while driving, and they switch attention between in-car displays and outside scenes. Johnson et. al [102] investigated visual attention while driving, which requires a human to attend to multiple simultaneous tasks. They presented a soft barrier approach for modeling eye movements in human drivers. The results showed that task priority and uncertainty, that grows when a task is not attended to, are primary controlling factors in allocating gaze in driving. Kujala and Salvucci [103] investigated how drivers divide attention between in-car displays and outside car scenes. The authors proposed that drivers adjust their attention time on in-car displays based on their driving performance on the road. For example, a driver decreases the visual attention on in-car displays and looks more on the road if the driving performance does not unfold as predicted. Mackenzie and Harris [104] presented a comparison of eye movement behavior and hazard reaction times in a simulated driving task and in a video-based (non-driving) hazard perception task. The results showed that drivers were slower at detecting hazardous situations when driving than when not driving and only seeing the video for hazard perception. The hypothesis in [104] was that drivers have more cognitive (e.g., attentional) load due to dual-tasking (driving and checking for hazardous conditions on the road). The higher cognitive load increases the reaction time. Crundall and Underwood [105] performed experiments with experienced and novice drivers, and showed that experienced drivers adapted visual strategies according to the complexity of road while novice drivers did not. For example, experience drivers scanned wider area as the complexity of road grew. Visual scanning of novice drivers remained same for all types of roads. Underwood et. al [106] investigated why novice drivers exhibit less scanning than experienced drivers. The authors experimented with novice and experienced drivers, in which subjects watched recorded video of driving and their eye movements were recorded. The analysis showed that novice subjects exhibited less scanning than experienced ones, even in non-driving tasks. This leads to the hypothesis that novice drivers' limited scanning is not because of the limited mental

capacity while driving simultaneously, but because of their poor mental model of the driving task.

2.2.3 Spatial Memory and Representation

Humans and animals walk from one site to another in everyday life and it is rarely aimless [85]. A dominant hypothesis has been the theory of spatial memory, which has been investigated by studying rats' movement in mazes (e.g. [107, 108, 109, 110, 111]). However, before the theory of spatial memory, walking was thought to be a reflex chain, i.e., a sequence of stimulus-response mechanisms, but the results from experiments with rats in mazes argued against the reflex chain theory [109].

Cognitive Map

Tolman [109] proposed that a rat builds a mental (cognitive) map of the environment (maze) describing routes, paths, and environmental relationships. He proposed that rats use the cognitive map to determine (select) which responses will be released when bombarded by various stimuli in navigating a maze, rather than responding based on stimulus-response relationships.

Route vs. Survey Maps

Cognitive (mental) maps take two primary forms [109, 85]: strip-like (route) and comprehensive (survey) maps. A route map encodes a specific path as a series of locations and turns. Such a map is not flexible to changes in the original environment or the start position. The survey map encodes relative positions of landmarks in an environment, and is more reliable than a route map for travelling between any two points in the environment.

Studies on the theory of spatial memory (e.g., [109, 110, 111, 112]) show that survey and route maps are selected based on the specific application. For example, if one takes a particular route regularly, the travelling process becomes automatized and is better explained by route maps [111, 112]. On the other hand, survey maps better explain the behavior of rats in maze when a rat can find a food from a new start position [109, 110].

Spatial Representation in Humans' Brain

Researchers (e.g. [113, 114, 115, 116]) have investigated how humans store spatial relationships within locations in an environment. For example, Stevens and Coupe [113] experimented with human subjects. Based on their observations, they presented a model that stores spatial relationships hierarchically and is governed by "storage-computation trade off". Spatial relationships that are not stored have to be determined by combining the stored spatial relations. Thorndyke [114] showed experiments in which human subjects were asked to estimate distances between two points on a map while viewing the map. Based on observations, the author in [114] proposed a model that expresses the estimated distance of a route as a linear combination of the true distance and the number of intervening points on the route. A highly cluttered map corresponds to a large number of intervening points. The hypothesis in [114] was that a subject visually scans along a route and judges the distance based on the scan time. At an intervening point, the subject pauses the scan to check if the point is the destination. Therefore, each intervening point takes a non-zero scan time and increases the overall scan time on the route, which increases the resulting distance estimate. Hirtle [115] investigated humans' spatial representations of natural environments that do not have an obvious or well-defined hierarchical structure. The analysis in [115] supported the view that a mental model of a real-world environment is composed of both spatial and nonspatial (non-Euclidean) information. The nonspatial information is stored in a hierarchical data structure based on subjective quantities such as intuitively pleasing and stability over time. McNamara [116] tested three classes of theories of the mental representation of spatial relations, which are nonhierarchical, strongly hierarchical (maximizing storage efficiency by storing minimum spatial relations required to represent a layout accurately), and partially hierarchical (storing many spatial relations that can be induced by other stored spatial relations) theories. Human experiments in [116] supported partially hierarchical representation of spatial relationships.

Thomson [90] presented a study of "blind" walking in humans. Subjects were first showed the target for 5 seconds and then they had to walk blindfolded towards the target. The results showed that performance degraded gradually with increasing target distance (beyond 9 meters). They investigated two hypotheses, perceptual and memory limitations, for the performance degradation. More experiments and analysis in [90]

suggested that subjects use the spatial memory (like a mental map or image of the environment) to move towards the target rather than a blind motor program to guide themselves. However, the mental map fades with time (specially beyond 8 seconds).

Several experimental studies as discussed above have supported the concept of cognitive (hierarchical survey) maps. However, there have been other theories of spatial knowledge such as landmark-based, path-integration, etc. For example, Foo et. al [117] presented navigation experiments in a virtual environment. The analysis in [117] showed that humans rely on visible landmarks while navigating an environment. The authors in [117] suggest that humans spatial knowledge not necessarily fall into any single class (cognitive map, route map, path-integration, landmark-based, or etc.). This dissertation uses a graph representation of task space, which is a form of cognitive map, based on sensory-motor patterns to investigate human environment learning in guidance and navigation. The framework based on sensory-motor patterns accounts for dynamic interactions between the agent and the environment, whereas the above studies involve either static or quasi-steady interactions and are discrete decision problems. The framework also investigates how visibility of nodes in the graph representation affects environment learning, which is similar to how much humans rely on visible landmarks.

Cognitive Robotics

Cognitive robotics [118] is inspired from human/animal spatial cognition. Jefferies and Yeap [119] provided a survey on robotic and cognitive approaches for spatial mapping, to motivate cross-fertilisation between the two areas. As stated in the survey, roboticists work on "sensor problems" and the cognitive researchers focus on "knowledge problems". The latter is defined as what people remember most when they visit new places and how they organize spatial information to form knowledge of their environment [119]. To achieve high-level cognitive capabilities for robots, human or animal spatial cognition has been studied and used to model environments. For example, Chakravorty and Junkins [120] presented an "intelligent path planning" method in an uncertain environment. The method uses sensors that allow the sensing of the environment non-locally, which is inspired from human vision. Vasudevan et. al [121] proposed a hierarchical probabilistic representation of space, based on high-level environment features such as household objects and doors. The goal was to make robots represent an environment in a way that

is comprehensible to humans. Manning et. al [122] presented a cognitive map-based computational model for wayfinding, which consists of three primary modules: vision (acquire and process visual information), cognitive map (store spatial information from the vision), and route generation (generate a route using the cognitive map). The model uses two parameters: vision and memory. The vision parameter accounts for accuracy of visual information (e.g., scene in peripheral vision is less accurate). The memory parameter accounts for that spatial memory fades with time. The wayfinding model in [122] was able to capture a range of behavior from directed route search to random walking.

2.2.4 Environment Representation

Humans' spatial navigation capabilities outperform autonomous robots in versatility, robustness, and effectiveness (e.g., success-rate). Spatial navigation in humans have been studied in the past. For example, Chase [123] investigated how taxi drivers navigate in large-scale urban environment that can not be perceived from a single vantage point. The author found that drivers use a hierarchical representation of the environment, which validates the theory of cognitive maps. Gillner and Mallot [124] studied the effect of local visual information on human environment learning, using movement data from experiments in a virtual maze. The results indicated that humans learn a maze as a view graph, i.e., sequence of local views and movements. Information at a node includes a recognized position, movement decisions, and expected next views for different decisions. Spiers and Maguire [125] presented a study of taxi drivers, which involves retrospective verbal reporting by drivers and gaze tracking. The method in [125] allows to do a temporal analysis of thoughts and understand cognitive (thinking) processes relevant to wayfinding.

Nested Environments

Most real-world environments are nested (e.g., a university campus, buildings in the campus, and then laboratories in the building). A theory is that nested environments are represented by a combination of different representations organized in a nested hierarchy [113, 116, 126] providing efficient structure for cognitive processing. Wang and Brockmole [126] presented an experimental study that concluded that humans don't

necessarily incorporate newly learned section of an environment into existing spatial knowledge, but switch between spatial representations when they cross specific spatial regions. At switching location, humans update their orientation information based on new spatial representation.

Three-Dimensional Spatial Representation

Real-world environments are three-dimensional which makes its spatial representation complicated. Jeffery et. al [127] suggested that 3-D world are not represented by a fully volumetric map, but are represented by a combination of several planar representations that correspond to the plane of locomotion. Representation in the orthogonal plane to the plane of locomotion is based on some non-metric way, different from the representation in the plane of locomotion. The authors suggested that even animals that move freely in 3-D world (e.g, birds) use such quasi-planar representations.

Body vs. World Frame

Learning an environment involves integrating first-person experiences into the global environmental knowledge [128]. Therefore, the alignment between the head/ego-centric frame (up,down,left,right) and the world-centric frame (north,south,east,west) affects learning of an environment. The mental (cognitive) maps are generally based on the world-centric frame and used for planning of global navigation. The ego-centric frame is used for local (immediate) movements. May et. al [129] suggested that cognitive (survey) maps are processed to determine a route for navigational purposes. This process has to deal with the mismatch between the map orientation and current head orientation (or direction of the perceived environment). The author showed that navigational performance usually decreases with the orientation mismatch.

Route Selection (Wayfinding)

In everyday navigation tasks, humans have to select a route among many possibilities. Researchers have investigated what factors influence route selection in humans. For example, Golledge [130] experimentally investigated what selection criteria, other than traditional ones such as minimum time, humans use to select a route in a map. Some

non-traditional criteria are initial heading (direction of perception), number of stops on a route, fewer turns, shortest leg first, aesthetically pleasing routes, etc. Hochmair and Frank [131], for instance, showed that humans use a least-angle strategy at intersections, i.e., select most straight lines, for wayfinding-decisions in unknown street networks. Hartley et. al [132] showed that cognitive processes are different for travelling a new (or less-travelled) route than a well-known (frequently-travelled) route. For well-known routes, sequences of body movements (motor commands) get automated, which requires less perceptual and attentional processing [133, 134].

Cues in Wayfinding

Darken and Sibert [135] presented a study involving humans in virtual space to investigate what cues can aid humans to improve wayfinding performances. Some cues suggested in [135] are directions indicators, path restrictions, absolute reference points, etc. Ruddle at. al [136] showed that if familiar objects are used as landmarks, the wayfinding performance is better. Waller et. al [137] showed that for learning their location, humans may rely more on distance information than bearing information of landmarks, and suggested to account for this finding in modeling human place learning. Kato and Takeuchi [138] showed that a good sense of direction aids a human in wayfinding by assisting in either using a global frame of reference (e.g., cardinal directions) or memorizing landmarks and their relative locations. Kelly et. al [139] studied the effects of environmental geometry on wayfinding performance. The study in [139] showed that visible angular corners help humans estimate spatial orientation. Vilar et. al [140] experimentally showed that horizontal signage prove more helpful than vertical signage in improving wayfinding performance of humans.

Asymmetry in Route Choices

Bailsenson et. al [141, 142] investigated why subjects choose different routes if start and target locations are switched. The study in [141, 142] found that subjects prefer routes that have longer and straighter initial segments, which are called hill-climbing or initial segment strategy (ISS). Bruny et. al [143, 144] showed that some humans have preference for routes that are Southbound. A possible explanation for the southern preference is misperceptions of increased elevation in North direction [143]. Vreeswijk

et. al [145] presented a study that shows that when travel times of two routes are within a range, human drivers are biased with one route and are not willing to alter their choice even if the traffic conditions and other factors change.

2.2.5 Decision-Making

Route selection in environment learning and spatial navigation involves decision-making, i.e., selecting a route among many possibilities or choosing between exploring new options and exploiting known ones. Simon [146] described decision-making as "a search process guided by aspiration levels. An aspiration level is a value of a goal variable which must be reached or surpassed by a satisfactory decision alternative". This section presents a brief overview of various factors in human decision-making.

Bounded Rationality and Satisficing

In traditional optimal control, decision-making refers to optimization of an objective function, which is called rational (optimal) behavior. In classical economics, humans were usually modelled as "economic (rational) man" [146]. Simon [146] argued against the economic man assumption and introduced the concept of bounded rationality that accounts for the fact that human decision-making is constrained by limits on time, available information, and cognitive processing capacities. He further introduced the concept of satisficing that replaces the goal of maximizing an objective function [146]. A possible way of satisficing is to try available alternatives in a sequential order and stop when an alternative that meets all criteria of an acceptable solution is found [146].

Simon [147, 146] also suggested that humans use the structures in task environment for their decision-making. In the present research, the concept of invariants (patterns) in agent-environment interactions in guidance tasks accounts for structure in the agent's behavior resulting from its interactions with environment. The interaction patterns [7, 8] provide a way to abstract the search space which in turn can be represented as a graph. This framework is used here to investigate to model human behavior and decision-making in environment learning.

Information Processing Model and Working Memory

Humans have limits on their memory and information processing, which is a primary reason that humans are in general not optimizers but satisficers. Cowan [148] presented an information processing model for humans, which consists of long-term memory storage, working memory, and focus of attention. Due to limited cognitive processing capabilities, humans can recall or remember only a limited amount of information at a time, which is called working memory. It is defined as a subset of long-term memory. Both bottom-up (involuntary factors: salient features in the perceived environment) and top-down (voluntary factors: personal beliefs) factors contribute to what information is held in working memory [149]. The information held in the working memory forms a basis for decision-making.

To overcome working memory limitations, a hypothesis is that humans use chunking mechanism [150]. In chunking, bits of information that have some type of similarity are combined into larger units called chunks. For example, expert players in chess create perceptual chunks of similar sub-configuration of pieces [151]. Another approach to overcoming the limitations in working memory involves pruning decision trees by using heuristics (e.g., Branch and Bound method [152]). Huys et. al [153] presented a study of human decision-making in a sequential decision-making task. The results in [153] showed that humans stop any further evaluation of a sequence if it exceeds a cost value higher than a threshold.

Situational Awareness in Dynamic Systems

Guidance tasks in unknown environments require dynamic decision-making that is making decisions repeatedly as new environment information unfolds. Decision-making in dynamic systems require a sufficient level of situational awareness [154]. For human decision-making in dynamic environments, such as a human pilot flying an aircraft, Endsley [154] defined situational awareness as a three-stage process: perception, comprehension, and projection. The first step involves perception of status, attributes, and dynamics of relevant elements in the environment. The second step, comprehension, goes beyond simply being aware of the relevant elements in the environment and requires understanding the significance of elements in light of desired goals. The last step

"projection" entails predicting future actions of elements in the environment. All three stages of SA are important for decision-making in dynamic systems.

Economic vs. Perceptual Choices

Human decision-making is in general investigated either on a computational or a neural basis [155]. These approaches are called economic decision making (EDM) and perceptual decision making (PDM), respectively. In EDM, it is investigated how choices are made based on a value of alternatives. In PDM, the investigation focuses on perceptual properties (e.g., saliency) of alternatives. Towal et. al [156] presented a study that shows that a combined model of EDM and PDM is more accurate for humans than either model alone.

A number of studies have shown that gaze fixations create a bias in decision-making [157, 158, 159, 156, 160]. For example, Shimojo et. al [157] modelled gaze bias as a "cascade effect". According to the cascade effect model, in starting the gaze is evenly distributed between alternatives and it gradually shifts to the option that is eventually selected. Krajbich et. al [158] showed that the probability of first-seen option being selected increases with the duration of first fixation. The early gaze bias [158] was observed by Sakellaridi et. al [160] in his study of visual exploration of city maps. In [160], humans subjects were asked to look at a city map and asked to choose a target (from given choices) to go to from a centre point on the map. The eye fixation analysis in [160] showed that humans shown an early selection bias even from the beginning of a trial.

Exploration vs. Exploitation

In learning or search tasks, exploration vs. exploitation is a well-known phenomena [161]. Exploration refers to searching new/unknown regions of problem space in order to find new solutions. Exploitation refers to repeating known (or already explored) solutions. In this dissertation, human subjects learn a task environment and find time-optimal routes between specified start and goal states. Such task involves trying different routes, which requires exploration-exploitation trade-off. In the experiments, a run is marked as exploration when a subject takes a new route and exploitation when a subject repeats a previously explored route. In general, there is no optimal policy for

trading-off between exploration and exploitation [162]. A problem that is widely studied in exploration vs. exploitation is multi-armed bandit problem (e.g., [163, 164, 165]) In bandit problem, a person has to chose from a set of options with unknown rewards, and has to maximize reward over a sequence of trials [165].

A number of studies (e.g., [166, 167]) have presented heuristics for exploration vs. exploitation trade-off. For example, Auer [166] presented confidence bounds based on statistics such as mean and variance, which can be used to guide exploration and exploitation. In this research, entropy is used to model human learning of CTG values at subgoals that are nodes in a decision-making tree (graph representation of task space).

2.3 Engineering vs. Spatial Cognition

The above studies on spatial cognition in general focused on pedestrians or simple movements. As stated by the author in [168], "simple forms of navigation, or way finding, have been the main focus of spatial cognition but without accounting for the effects of dynamics." In agile guidance tasks, such as a pilot operating a high-speed vehicle in a complex environment or surgeons under time pressure, the interactions between vehicle dynamics and task environment play a role in determining what elements of the environment are more relevant than others.

The overall behavior of a human pilot in a spatial environment is laid out by the interactions between pilot control and cognitive characteristics and the environmental characteristics such as scale and layout. Warren [13] used the term "behavioral dynamics" that represents the closed-loop agent-environment dynamics. The concept is originally inspired from the Gibson's idea of ecological perception [1]. Gibson's ecological approach suggests that a human or animal learns (represents) an environment based on the task in hand and desired goals. Therefore, the study of spatial cognition (representation and learning) should be integrated with the study of pilot dynamic and perceptual behavior.

Mettler [168] highlighted that traditional optimal control formulation of trajectory planning problems does not take advantage of the problem structures that play a fundamental role in humans' and animals' skills. The author proposed the idea that skilled

human pilots possess a system to conceptualize spatial behavior that preserves the interrelation between movement dynamics and geometry and topology of the environment. In subsequent studies, Kong and Mettler [7] studied the guidance behavior in complex environments focusing on the agent-environment interactions. The study showed that skilled operators organize their behavior according to interaction patterns. These sensory-motor patterns represent units of behavior which satisfy the various system constraints and exploit the equivalences in the problem space. Furthermore, the interaction patterns make it possible to abstract a task environment as a graph of subgoals. Such graph framework can be elaborated to build a cognitive map to model and investigate human learning and decision-making in complex task environments. This dissertation uses the subgoal graph to investigate human environment learning and spatial navigation in guidance tasks where human subjects navigate using a complex dynamic vehicle.

Chapter 3

Autonomous Guidance and Learning Framework

This chapter briefly reviews the mathematical formulation for optimal guidance problem using receding horizon framework and the concept of spatial value function [9]. Next, the chapter shows that the computation of SVF is an implementation of reinforcement learning. Finally, the chapter presents the learning autonomous guidance framework based on the sensory-predictive guidance system proposed in [9]. The chapter highlights the algorithmic contributions of this dissertation. The chapter is taken from article [25].

3.1 Optimal Guidance Problem (OGP)

An optimal guidance problem determines a control trajectory $\mathbf{u}(\cdot)$ for a vehicle travelling from a state $\mathbf{x}$ to a goal state $\mathbf{x}_{goal}$, which minimizes a cost function as following:

$$J(\mathbf{x}, \mathbf{u}(\cdot)) = \int_0^{t_f} g(\mathbf{x}(t), \mathbf{u}(t))\, dt, \tag{3.1}$$

where $\mathbf{u}(t) \in \mathbb{R}^m$ and $\mathbf{x}(t) \in \mathbb{R}^n$ are control and state vectors, respectively. g is the instantaneous cost function. t_f is the time to travel from state $\mathbf{x}$ to $\mathbf{x}_{goal}$. $f(\mathbf{x}, \mathbf{u})$ describes the vehicle dynamics. The optimal cost from state $\mathbf{x}$ to $\mathbf{x}_{goal}$ is called the

optimal value function V^* at state $\mathbf{x}$:

$$\begin{aligned} V^*(\mathbf{x}) &= \min_{\mathbf{u}(\cdot)} J(\mathbf{x}, \mathbf{u}(\cdot)). \\ \text{s. t. } \dot{\mathbf{x}} &= f(\mathbf{x}, \mathbf{u}), \end{aligned} \tag{3.2}$$

In receding horizon (RH) planning (Fig. 3.1), a finite-horizon trajectory from current time t to time $t + t_H$ is computed as following:

$$\underset{\mathbf{u}_H^*(\cdot)}{\text{argmin}}\{\int_{\tau=t}^{\tau=t+t_H} g(\mathbf{x}(\tau), \mathbf{u}(\tau))d\tau + V^*(\mathbf{x}(t+t_H))\}. \tag{3.3}$$

$\mathbf{u}_H^*(\cdot)$ is the optimal control trajectory to drive the vehicle from the current state $\mathbf{x}(t)$ to the state $\mathbf{x}(t+t_H)$. The cost-to-go from state $\mathbf{x}(t+t_H)$ to the goal state is represented by the optimal value function at $\mathbf{x}(t+t_H)$.

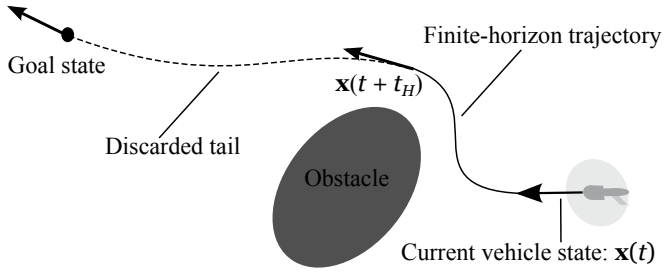


Figure 3.1: Receding horizon (RH) trajectory optimization.

3.1.1 Spatial Value Function (SVF)

The concept of spatial value function [9] was introduced to describe optimal spatial behavior for guidance. The state vector $\mathbf{x}$ is partitioned into two parts: 1) $\mathbf{x}_p \in \mathbb{R}^3$: vehicle position vector and 2) $\mathbf{x}_v \in \mathbb{R}^{n-3}$: vehicle dynamic state such as velocity, acceleration, and higher derivatives. The spatial value function V_S^* is defined over spatial position vector $\mathbf{x}_p$ as following:

$$V_S^* : \mathbf{x}_p \mapsto \min_{\mathbf{x}_v}\{V^*(\mathbf{x}_p, \mathbf{x}_v)\}, \tag{3.4}$$

The value function is defined over the entire state-space $\mathbf{x}$, while the spatial value function defines the optimal cost-to-go value and corresponding dynamic state at any

spatial location $\mathbf{x}_p$. $\mathbf{v}^*$ is the optimal velocity vector in the inertial frame. A function Υ^* [9] is defined over the spatial position vector ($\mathbf{x}_p$) space as following:

$$\Upsilon^* : \mathbf{x}_p \mapsto \{V_S^*(\mathbf{x}_p), \mathbf{v}^*(\mathbf{x}_p)\}. \tag{3.5}$$

$\Upsilon^*(\mathbf{x}_p)$ is the optimal cost (V_S^*) and velocity vector ($\mathbf{v}^*$) at location $\mathbf{x}_p$. These variables describe the scalar cost-to-go (CTG) map and velocity vector field (VVF) for optimal spatial behavior. SVF refers to the function Υ^*, i.e., CTG and VVF. The cost of the discarded tail, $V^*(\mathbf{x}(t+t_H))$ in Eq. 3.3, is approximated by the SVF at $\mathbf{x}_p(t+t_H)$.

For most practical problems, the SVF cannot be solved analytically. An approximate SVF Υ is computed using computational techniques. Frazzoli et. al [169] presented the motion primitive automaton (MPA) for vehicle guidance. The MPA defines a finite library of motion primitives (MPs) that allow transition between two states (not necessarily all pairs of quantized states) in a finite set of quantized states and controls. Mettler and Kong [22] used grid-based MPs that are constrained to start from and end at finite points in a quantized position space. The grid-based MPs discretize the environment and use the MPA to generate a cost-to-go function that accounts for the dynamics. With the grid-based MPs, the trajectory optimization problem (or computation of SVF) is converted into a sequential decision problem that can be solved using dynamic programming [170, 22]. The details of the grid-based MPs, used in this dissertation, are given later in this chapter.

3.1.2 Reinforcement Learning of SVF

Sutton and Barto [171] defined reinforcement learning as following: "Reinforcement learning is learning what to do–how to map situations to actions–so as to maximize a numerical reward signal. The learner is not told which actions to take, as in most forms of machine learning, but instead must discover which actions yield the most reward by trying them" (section 1.1 in [171]). Elements of reinforcement learning are: state, policy (action), reward function, and value function. Reinforcement learning uses value functions over the state-space to search for optimal policies that maximize cumulative reward from any state. The optimal value function and policy contain the information needed to determine the best action for any possible state. Iterative methods such as

value iteration and policy iteration are used to compute the optimal value function and policy [171].

For guidance problems, the SVF describes the value function for reinforcement learning. The entire state-space (spatial position vector $\mathbf{x}_p$ and vehicle dynamic state vector $\mathbf{x}_v$) is discretized. For an optimal solution, each spatial position has to be assigned an optimal dynamic state $\mathbf{x}_v^*$ (e.g. velocity $\mathbf{v}^*$) and optimal cost (SVF). The vehicle uses a motion primitive mp from a finite library $MP_{library}$ to transit from a state $\mathbf{x}$ to state $\mathbf{x}'$. In presence of noises and uncertainties in the system, $P_{\mathbf{x}\mathbf{x}'}^{mp}$ is the transition-probability between states $\mathbf{x}$ and $\mathbf{x}'$ if motion primitive mp is applied at state $\mathbf{x}$. ct is the incremental cost (e.g. time) of a motion primitive, which is the reward function for reinforcement learning. The optimal policy describes the optimal motion primitive at any position $\mathbf{x}_p$. For the optimal SVF, the Bellman equation is true for any state $\mathbf{x}$:

$$SVF(\mathbf{x}) = \min_{mp \in MP_{library}} \sum_{\mathbf{x}'} P_{\mathbf{x}\mathbf{x}'}^{mp} [ct_{\mathbf{x}\mathbf{x}'}^{mp} + SVF(\mathbf{x}')]. \tag{3.6}$$

Eq. (3.6) is identical to Eq. (3.15) in [171]. To account for obstacles in the optimal guidance solution, the transition between two spatial locations using a motion primitive is prohibited if the primitive intersects an obstacle. Since this dissertation studies a deterministic case, a motion primitive mp from a state $\mathbf{x}$ reaches a specific state $\mathbf{x}'$. Therefore, the probability term $P_{\mathbf{x}\mathbf{x}'}^{mp}$ is dropped for the analysis presented in this dissertation.

In reinforcement learning, the value function is initialized randomly unless some bootstrapping method is used and then value or policy iteration [171] is used to solve Eq. (3.6). In this dissertation, the a priori SVF (CTG and VVF) accounts for known obstacles, and is computed using a dynamic programming approach, such as Dijkstra's algorithm [27, 22] applied backwards from the goal state. As the system navigates through the environment, it adapts Eq. (3.6) based on current sensory data about the immediate environment. Ideally, the SVF should be updated throughtout the problem space. However, only a local SVF based on the current vehicle state is updated due to limited online computational resources. The dissertation uses successive runs to achieve a consistent solution to Eq. (3.6) throughout the geographical space, which is illustrated in Section 3.3 using an information propagation and assimilation model.

3.2 Autonomous Guidance and Learning Framework

This section describes the sensory-predictive guidance system presented in [9] as a learning autonomous guidance framework. The section highlights the algorithmic contributions of this dissertation. Since the dissertation focuses on the SVF learning and guidance behavior, it is assumed that the vehicle pose is known. The onboard sensor is a laser scanner that gives a depth map in a specified range. For simulations presented in this dissertation, the sensor is assumed to give an accurate depth map to help isolate effects of sensing errors from the guidance behavior learning processes.

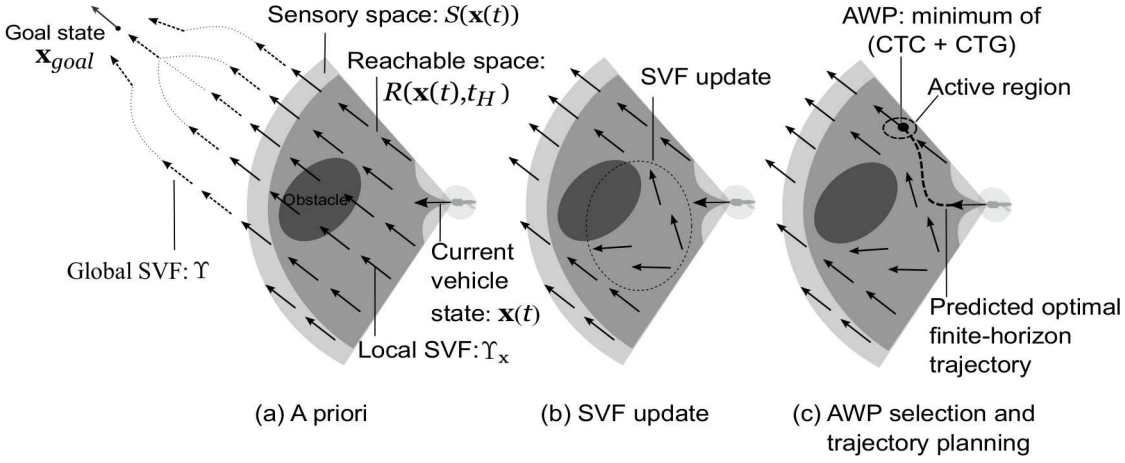
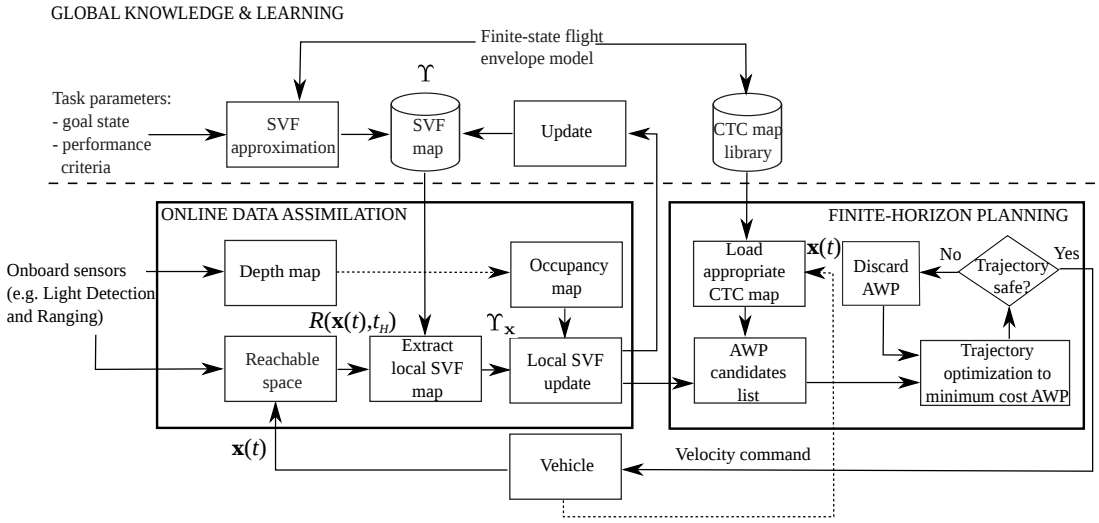


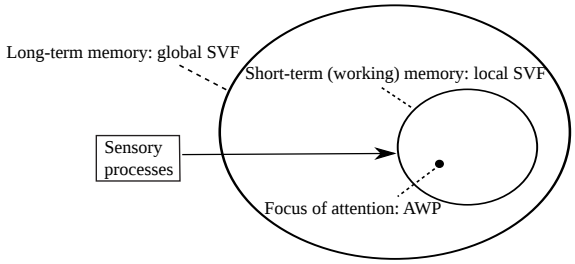
Figure 3.2: Planning cycle for the sensory-predictive guidance system [9].

Before presenting the learning framework, Fig. 3.2(a) illustrates the sensory and reachable spaces for the receding horizon trajectory optimization with SVF (notations based on [9]). The sensory space $S(\mathbf{x}(t))$ represents the space around the vehicle current state $\mathbf{x}(t)$, which can be covered by onboard sensors. The reachable space $R(\mathbf{x}(t), t_H)$ [9] is defined as a subset of the sensory space, which the vehicle can reach within a fixed time horizon t_H from its current state $\mathbf{x}(t)$, given the vehicle dynamic constraints (e.g. maximum speed and acceleration). $R(\mathbf{x}(t), t_H)$ is constrained to be inside the sensory space for safety, as the system cannot see the environment outside the sensory space. The local SVF $\Upsilon_{\mathbf{x}}$ is defined over the reachable space $R(\mathbf{x}(t), t_H)$ and is extracted from the global SVF Υ as shown in Fig. 3.2(a). One can determine t_H based on the sensing

range and online computational resources, because the computation required for finite-horizon trajectory optimization depends on its length ($\propto t_H$).



(a) Learning autonomous guidance framework



(b) Local SVF activation as working memory concept in Cowan's information-processing model [148].

Figure 3.3: Learning and information-processing structure for the sensory-predictive guidance system in [9].

The central component of the learning framework shown in Fig. 3.3(a) is an online data assimilation system that integrates current global SVF, depth sensory data, and vehicle state. Online sensory data is used to update the local SVF map $\Upsilon_{\mathbf{x}}$ so that the information is available for subsequent operations. A set of CTC (cost-to-come) maps, computed offline prior to the task and stored in long-term memory, are used to approximate the cost of the finite-horizon trajectory, i.e., the first term in Eq. (3.3). The concept of offline computed CTC maps is a contribution of this dissertation and is described later in this section. The CTC and the local SVF maps are overlapped in the

reachable space to determine the optimal active waypoint (AWP) that serves as local goal state for the online finite-horizon trajectory optimization.

The learning framework in Fig. 3.3(a) can be related to the information-processing system proposed by Cowan [148], as shown in Fig. 3.3(b). In Cowan's information-processing model, the working memory represents a subset of the long-term memory. The subset is activated by sensory inputs. In the guidance system shown in Fig. 3.3(a), sensory inputs provide the reachable space for the vehicle, which determines the local SVF. The local SVF operates as a form of working (short-term) memory since the guidance system uses the local SVF information to select the current goal state, i.e, AWP, for the online trajectory optimization, which, in turn, determines the control action. Finally, AWP is similar to the focus of attention in the information-processing model. The planner selects an optimal AWP and plans a trajectory to the AWP.

3.2.1 Real-Time Sensory and Guidance Processes

The learning framework consists of a real-time sensory guidance planning cycle [9] that includes the following four operations in the given order: 1) environment sensing, 2) SVF update, 3) AWP candidates selection, and 4) finite-horizon trajectory optimization. Each operation is described further in this section.

Environment Sensing

This step has no new development. Only for completeness, a brief description is given about this step. The environment is discretized into rectangular cells and each cell has an occupancy probability (0-1). A priori occupancy map accounts for the known obstacles. The autonomous guidance system uses the same procedure as presented by Dadkhah and Mettler [23] to update the occupancy probability map in the sensory space. A laser scanner mounted on the vehicle measures the depth map in sensory range. The depth map is converted to an occupancy probability map using a risk map update equation proposed by Marlow and Langelaan [172]. This study assumes an ideal sensor for a baseline study to concentrate only on learning and planning mechanisms. For an ideal sensor, the occupancy probability of a cell is either 0 or 1 as the sensor provides an accurate depth map within its sensory range.

SVF Update

The algorithm used to update the SVF is adapted from [23]. A new development in this dissertation is the use of the concept of maneuvering environment scale ratio proposed in [173] as a basis to generate saturated motion primitives.

The spatial value function (CTG and VVF) is updated based on changes in the occupancy probability map as shown in Fig. 3.2(b). The velocity vectors behind the obstacle diverge from that region creating a repelling manifold [11], which causes high cost-to-go values in that region. Dadkhah and Mettler [23] presented an algorithm for the SVF update using the dynamic version of Dijkstra's algorithm [45] combined with motion primitives. Rather than re-computing the SVF map over the entire task environment, the algorithm is used to adjust the SVF locally. The details of grid-based motion primitives for time-optimal solutions, used in this dissertation, are given next.

For a mass-point vehicle, a_{max}, v_{max}, and v_{zmax} are maximum horizontal acceleration, maximum horizontal speed, and maximum vertical speed, respectively. This study does not consider vertical acceleration. Also, vertical and horizontal motions are assumed to be independent of each other. Horizontal and vertical spatial resolutions are dxy and dz, respectively. For finite-state representation, horizontal and vertical speeds are discretized into finite sets v_l and v_{zl}, respectively. n_v and $2n_{vz}+1$ are the number of discrete speed levels in sets v_l and v_{zl}, respectively. r_{min} is the minimum turning radius at maximum speed v_{max}. The number of horizontal speed levels n_v and the set of discrete horizontal speed levels v_l are given by:

$$\begin{aligned} n_v &= \frac{r_{min}}{dxy} = \frac{1}{dxy}\frac{{v_{max}}^2}{a_{max}}, \\ v_l(i) &= \sqrt{a_{max}\ i\ dxy} \qquad \forall\ i = 1, ..., n_v. \end{aligned} \tag{3.7}$$

Minimum turning radius $r_{min} = v_{max}^2/a_{max}$ is a critical length relating to the dynamic fit [173] between vehicle maneuvering capabilities (e.g. maximum speed and acceleration) and length-scales of maneuvers required for time-optimal performance. For example, if the vehicle is flying in an environment section that constrains the vehicle to take a turn of radius $r < r_{min}$, the vehicle will use a speed $v < v_{max}$ due to the limited acceleration a_{max}. In environment sections with large length-scales ($\geq r_{min}$), the vehicle will use maximum speed for time-optimal performance. This dissertation

uses a spatial resolution dxy so that r_{min} is an integer multiple of dxy. For a desired number of horizontal speed levels, spatial resolution dxy is determined according to Eq. (3.7).

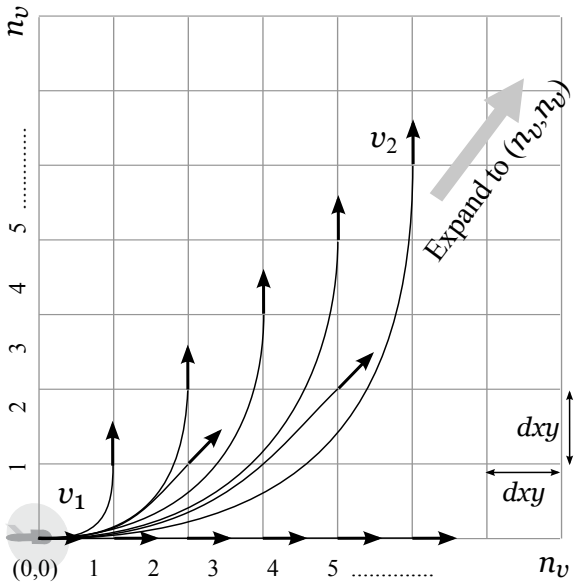


Figure 3.4: Horizontal motion primitives.

Figure 3.4 shows an example of grid-based MPs for horizontal motion. MPs enable transition between states in the discretized state-space. In Fig. 3.4, the vehicle is at (0,0) and its speed level is $v_1 = v_l(i), i \in [1, ..., n_v]$ parallel to the grid. It can go straight, turn right, or left. Figure 3.4 shows only left turning given the symmetry in planar motion. A grid-based motion primitive is constrained to start at (0,0) and end at (I, J) where $I \in$ [1,...,n_v] and J is $0, I/2$, or I. $J = I/2$ is possible only if I is an even number. $J = 0, I/2$, and I correspond to straight motion, $\pi/4$ turn, and $\pi/2$ turn, respectively. The heading resolution is $\pi/4$, i.e., the heading can either be parallel or diagonal to the grid. Such heading resolution makes sure that MPs are symmetric to rotation. For instance, if the direction of v_1 is diagonal to the grid, the tree of MPs starting from (0,0) is achieved by rotating the tree shown in Fig. 3.4 by $\pi/4$ and increasing the length scale by $\sqrt{2}$. v_2 is the speed level at the end of a motion primitive. For a MP, v_1 is any speed level from 1 to n_v. Due to the acceleration constraint, v_2 can only be a speed level from 1 to m where $m \leq n_v$ as acceleration required for maneuvering from v_1 to

$v_2 = v_l(m{+}1)$ is greater than a_{max}. Time-optimal solutions, for a mass-point model with speed and acceleration limits and no jerk limit, use either maximum speed or maximum acceleration at all times, which is a saturated behavior [173]. A MP is saturated if it uses either maximum acceleration to maneuver (turning, accelerating/decelerating) or maximum speed in straight line. For time-optimal solutions, this dissertation considers only the transition to $v_2 = v_l(m)$. For each $v_1 \in v_l$, a tree of saturated MPs as shown in Fig. 3.4 is computed.

The vertical speed set is v_{zl} = [-v_{zmax},...,-2v_{zmax}/n_{vz},-v_{zmax}/n_{vz}, 0, v_{zmax}/n_{vz}, 2v_{zmax}/n_{vz},...,v_{zmax}]. As no vertical acceleration is considered, the vehicle can change altitude from z to $z \pm dz$ from any vertical speed level to the other one. Depending on the vehicle a designer is trying to model, climb/descent rates can be restricted based on the horizontal speed level.

The incremental cost of each motion primitive is defined based on the cost function that has to be optimized For example, the cost can be time, path-length, control energy, a function of these, etc. In this dissertation, the cost is time. For three-dimensional motion, a pair of horizontal and vertical MPs is applied. As the two motions are independent, the cost of a pair of horizontal and vertical MPs is either the cost of horizontal or vertical MP depending on which one is higher. The described MPs form the library $MP_{library}$ used in Eq. (3.6).

Note that a designer can add zero speed in the set v_l. In this dissertation, zero horizontal speed is not used since the smallest transition (straight path or turn) is between two neighbor grid points and such transition is possible at minimum speed $v_l(1) = \sqrt{a_{max}\ dxy} > 0$ within the acceleration limit a_{max}.

AWP Candidates Selection

This step includes an algorithmic development in this dissertation. After the SVF update, the guidance system (Figs. 3.2 and 3.3) selects an active waypoint (AWP) in the reachable space, to plan a finite-horizon trajectory from the vehicle current state to the AWP (see the first term in Eq. 3.3). Dadkhah and Mettler [23] used a weighted sum of the following costs to select the best AWP:

1. Composite cost: cost-to-come (CTC) to the AWP from the vehicle current state + cost-to-go (CTG) to the goal from the AWP.

2. Risk measure function: occupancy probability at the AWP.

3. Penalty: to avoid frequent changes in the flight direction and altitude, cost terms to penalize the difference between previous and current AWPs' altitude and heading.

The best candidate for AWP has the lowest value for a weighted sum of the three cost functions. As this dissertation focuses on a baseline case with a perfect sensor and zero noise, it considers only the objective function (flight-time) for selecting the best AWP as following:

$$\begin{aligned} \min_{\mathbf{x}_p} \quad & \mathrm{CTC}(\mathbf{x}(t), \mathbf{x}_{\mathbf{x}_p}) + \mathrm{CTG}(\mathbf{x}_p), \\ \text{subject to} \quad & \mathbf{x}_p \in R(\mathbf{x}(t), t_H), \end{aligned} \tag{3.8}$$

where $\mathbf{x}_{\mathbf{x}_p}$ is the optimal state (velocity), represented by the SVF, at spatial position $\mathbf{x}_p$. $\mathrm{CTG}(\mathbf{x}_p)$ represents the cost-to-go from $\mathbf{x}_{\mathbf{x}_p}$ to the goal state. $\mathrm{CTC}(\mathbf{x}(t), \mathbf{x}_{\mathbf{x}_p})$ represents the cost-to-come to $\mathbf{x}_{\mathbf{x}_p}$ from the current state $\mathbf{x}(t)$.

The $\mathrm{CTG}(\mathbf{x}_p)$ for any position in the reachable space is extracted from the SVF map. Computation of $\mathrm{CTC}(\mathbf{x}(t), \mathbf{x}_{\mathbf{x}_p})$ is a two point boundary value problem. The vehicle starts from the current state and reaches the AWP with the terminal velocity assigned by the SVF (VVF) at the AWP position. The vehicle is modeled as a point mass and its dynamics are described by a state-space discrete-time system with second-order constraints on speed and acceleration (described later in this section). Such trajectory optimization problem has in general no analytical solution and is solved using a numerical approach. In previous approaches presented in [20, 21, 22, 9, 23], CTC map is computed by executing an online numerical optimization method for each $\mathbf{x}_p$ in the reachable space. Such approach, however, is costly if number of points in the reachable space are large enough.

Dadkhah and Mettler [23] used a heuristic based on the Hamiltonian solution for a uniformly accelerated point mass model, to provide the lower bound on CTC. However, the method in [23] has two drawbacks: 1) The velocity constraint can be violated due to the uniform acceleration; 2) The method does not account for the terminal velocity at the AWP. This dissertation presents a computationally cheap approach to account for the CTC map. The approach is to compute CTC maps offline for all possible vehicle states, store them in long-term memory, and recall them online based on the current

vehicle state (Fig. 3.3(a)). The total number of vehicle states is finite, i.e., combinations of all speed levels and heading angles in the quantized state-space. A CTC map is an inverted SVF map in the vehicle body frame as shown in Fig. 3.5(b), which describes the optimal cost and velocity fields that start from the vehicle current state and travel to any point in the reachable space. The CTC map for a start state is computed using the same motion primitives and dynamic programming as used in the SVF computation. Thus, a CTC map accounts for the velocity and acceleration constraints. A library of CTC maps for all possible vehicle states is computed offline. A CTC map is computed for free space (no obstacles) in a fixed volume, equal to the sensory volume, around the vehicle pose that is the origin or CTC=0. Note that the AWP is in the reachable space that is a subset of the sensory space. For online operation, a CTC map based on the current vehicle velocity is recalled and transformed, using a translation and rotation, into the global coordinates.

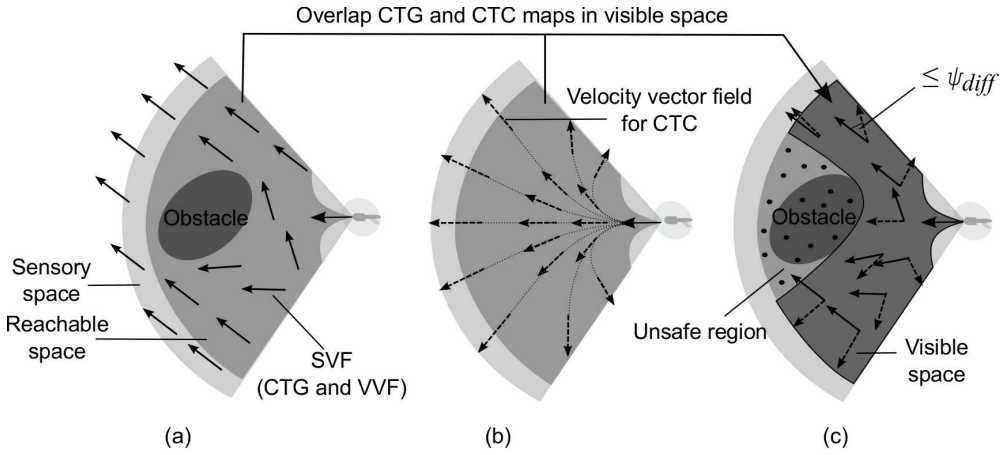


Figure 3.5: Overlapping of CTG and CTC maps in the visible space.

The offline computed CTC maps do not account for obstacles in the reachable space, as shown in Fig. 3.5(b). For safety, the reachable space is divided into unsafe region $Ur(\mathbf{x}(t))$ and visible space $Vis(\mathbf{x}(t))$ as shown in Fig. 3.5(c). The following constraint is applied with Eq. (3.8):

$$\mathbf{x}_p \in Vis(\mathbf{x}(t)). \tag{3.9}$$

Figure 3.5(c) shows the overlapping of CTC and CTG maps in the visible space. At any point in the visible space, the velocity for CTC map is not necessarily the same

as the velocity for the SVF map. The velocity vector for SVF at any point gives the optimal direction that the vehicle velocity direction should align with in order to reach the goal in minimum time and avoid obstacles. Therefore, the vehicle should reach the AWP with the velocity direction assigned by the SVF (CTG) at the AWP position. To ensure the velocity alignment, any $\mathbf{x}_p$ that does not satisfy the following condition is not an AWP candidate:

$$\&|\psi_{ctg} - \psi_{ctc}| \leq \psi_{diff}, \tag{3.10}$$

where ψ_{ctg} and ψ_{ctc} represent velocity directions for CTG and CTC maps, respectively. ψ_{diff} is the allowed difference between the two velocity directions, as shown in Fig. 3.5(c). ψ_{diff} in this study is 45^0 as that is the heading resolution for MPs. Finally, a list of AWP candidates that satisfy constraints (3.9) and (3.10) is computed. Cost function in Eq. (3.8) is used to sort the AWP candidates list in increasing order of cost. The sorted list is AWP_{list}.

If the sensor is not ideal and there are disturbances in the system, a single point as AWP is not appropriate. In a probabilistic model, the guidance system should select an active region where the probability of cost being minimum is above a specified threshold. The baseline case, presented in this dissertation, uses a single point as immediate goal (AWP).

Finite-Horizon Trajectory Optimization

This step has no new development. Just like step 1, a brief description is given about this step for the purpose of completeness. The last step of the planning cycle is to compute an optimal trajectory to the AWP. The finite-horizon trajectory optimization is a two-point boundary value problem. The aircraft used in the simulations in this dissertation is based on Blade-Cx2 coaxial indoor helicopter (see [174] for the helicopter model system identification). The vehicle dynamics are approximated by a discrete-time linear state-space system as shown in Eq. (3.11). State vector $\mathbf{x}$ consists of horizontal acceleration, 3-D velocity, and 3-D position. Input vector $\mathbf{u}$ consists of the velocity command in each direction. Δt is the time-step (sampling time). N is the number of

time-steps between the current state and AWP.

$$\begin{aligned} &\mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k,\ k\ \in [1, ..., N], \\ &\mathbf{x} = [a_x\ \ v_x\ \ x\ \ a_y\ \ v_y\ \ y\ \ z\ \ v_z]^{\mathrm{T}},\ \mathbf{u} = [u_x\ \ u_y\ \ u_z]^{\mathrm{T}}, \end{aligned} \tag{3.11}$$

u_{max} and u_{zmax} represent the maximum velocity commands in horizontal and vertical directions, respectively. The horizontal and vertical motions are assumed to be independent of each other. The dynamic constraints are as following:

$$\begin{aligned} {a_x}^2 + {a_y}^2 \leq {a_{max}}^2\ \ ;\ \ {v_x}^2 + {v_y}^2 \leq {v_{max}}^2\ \ ;\ \ {u_x}^2 + {u_y}^2 \leq {u_{max}}^2 \\ |v_z| \leq v_{zmax}\ \ ;\ \ |u_z| \leq u_{zmax} \end{aligned} \tag{3.12}$$

The number of time-steps N has to be minimized to achieve the minimum time trajectory. The initial state is the vehicle current state. The final condition includes AWP position and horizontal speed. This dissertation uses a mixed integer linear programing (MILP) formulation [19, 23] to solve the trajectory optimization problem . The MILP formulation uses a binary decision variable b_k that is 1 only at one time-step $k = H$ between 0 and N, as shown in Eq. (3.13). M is a large number. At $k = H$ ($b_H = 1$), AWP is reached.

$$\begin{aligned} \min \sum_{k=1}^{k=N} M(N + 1 - k)(1 - b_k), \\ \sum_{k=1}^{k=N} b_k = 1. \end{aligned} \tag{3.13}$$

ϵ_p and ϵ_v are the allowed errors for the position and velocity vectors at the AWP (see [23] for details). The MILP problem is solved using CPLEX [175]. N is given as:

$$N = \left\lceil \frac{\text{CTC at AWP}}{\Delta t} \right\rceil + N_0 \tag{3.14}$$

N_0 is a specified margin due to model mismatch between CTC computation (based on MPs) and online trajectory optimization (state-space system that includes control effects). N_0 for simulation experiments presented in current study is 5.

Note that if the time does not have to minimized, convex programming can be used to find whether a feasible trajectory exists for a horizon N ($b_N = 1$) or not. Minimum number of time-steps ($N = H$) can be found by bisection method (a sequence of convex

programming). Convex programming is faster than MILP. This study, however, sticks to the MILP formulation since the test simulation is run offline and the average computation time of MILP is 0.17 s (shown in Chapter 4) due to the small planning horizon N (mean is 9.5 and standard deviation is 3.6, Figure 4.3). Therefore, convex programming does not seem essential at this point.

Safety of Finite-Horizon Trajectory

Finite-horizon trajectory planning accounts for the environment in the current sensory space. Outside of the sensory space, the guidance system can only account for obstacles that are either a priori known, or learned during the task. A critical scenario is when the best AWP is close to the boundary of sensory space and right outside the boundary an unknown obstacle exists. In such a case, exercising the entire predicted trajectory to the AWP has a collision risk. Therefore, the guidance system exercises only a fraction of the predicted finite-horizon trajectory. H_{track} is the number of time-steps the guidance system exercises:

$$H_{track} = \min(H, H_{max}). \tag{3.15}$$

H_{max} is the maximum horizon that the guidance system is allowed to exercise between two consecutive AWPs (or planning cycles). A designer can determine H_{max} based on how large the sensing range is and how fast planning cycles can be completed. The following relationship should hold for safety:

$$H_{max} \times \Delta t \times v_{max} = \alpha \times (\text{sensing range}), \quad 0 < \alpha < 1. \tag{3.16}$$

Eq. (3.16) indicates that maximum distance travelled along finite-horizon trajectory should be less than the sensing range.

An AWP is selected based on the CTG and CTC maps that are based on MPs. The finite-horizon trajectory optimization uses a discrete-time linear state-space model that include control effects. Due to model mismatch between AWP selection and online trajectory optimization, the trajectory may not be safe. Ideally, the online trajectory optimization should include environmental constraints such as obstacles. However, accounting for those in the online optimization would be computationally expensive as obstacles are cluttered and not necessarily convex. The guidance system presented in this dissertation iterates through the list AWP_{list} that is arranged in increasing order

of total cost (CTG + CTC), and checks if the trajectory to an AWP is safe. The first AWP with a safe trajectory is selected and the corresponding velocity commands are sent to the vehicle (Fig. 3.3). n_{iter} is the number of iterations to find the safe AWP. This dissertation uses the following safety criteria for a finite-horizon trajectory: "At any time-step of the finite-horizon trajectory, there should be no obstacle within distance $dmin_{stop} = v_{max}^2/(2a_{max})$ along the velocity direction (tangent to the trajectory)". $dmin_{stop}$ is the minimum distance required to come to a full stop at the maximum speed.

3.3 Information Propagation and Assimilation Model

Planning in unknown environments requires learning the map online and re-planning the path. For re-planning, the guidance system has to propagate and assimilate the environment information, received through onboard sensors within the sensory space, into the global spatial value function so that the information can be used for current and future planning. The global SVF update, however, is not practical for online operations because the computational complexity of SVF computation grows exponentially with the size (volume) of geographical space. Thus, a limited processing speed in online operations puts an upper limit on the size of spatial volume for the SVF update in each planning cycle. This section presents an information propagation range that models the limit on information processing (propagation and assimilation of environment information into the SVF) in real-time guidance operations. The section first uses a quasi 1-D example to illustrate the role of the information propagation range in learning the SVF over a succession of runs. Next, the model is illustrated with the learning autonomous guidance framework.

3.3.1 Agent Definition and Sample Problem

Agent A in Fig. 3.7 represents the guidance system. r_1 and r_2 are the agent's sensory and information propagation ranges, respectively. The agent can detect only immediate environment that lies within range r_1. The environment information acquired within range r_1 can only be propagated and assimilated into the SVF within range r_2 from the agent position. The information propagation range represents an upper limit on the volume of the local SVF update. For a discretized environment, the computation time

of local SVF update can be related to the number of cells in the local SVF volume, by an empirical analysis. A designer can determine the information propagation range based on the computational platform (processor speed, memory size, etc.) used for online operations, the spatial resolution chosen for the environment, and the average time available for local SVF update between two consecutive planning cycles.

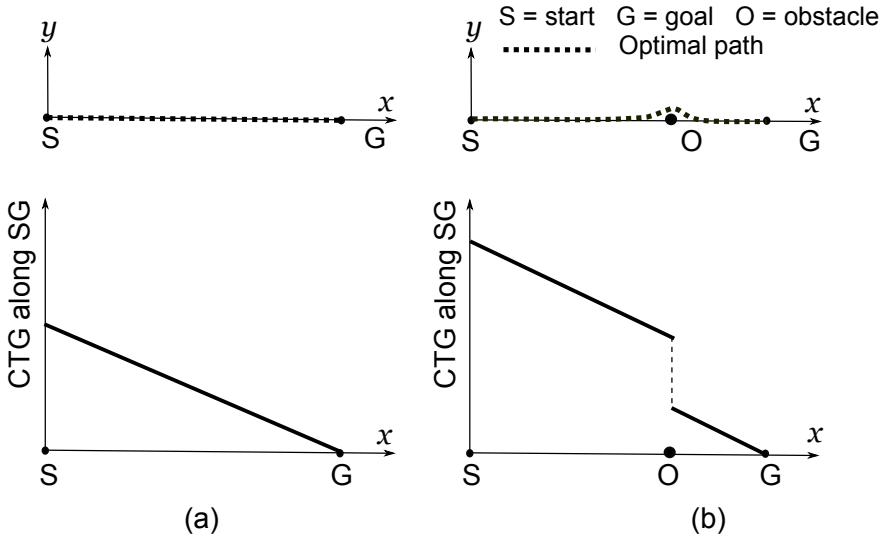


Figure 3.6: Quasi 1-D problem: (a) no obstacle (b) obstacle is present. The agent can move only in xy plane.

Figure 3.6 illustrates a simple quasi 1-D problem. The agent has to reach the goal G from the start location S in minimum time. The optimal solution is to travel along the line SG. If an obstacle O is introduced on SG, the optimal solution is to go around O in the xy plane as shown in Fig. 3.6(b). For the presented problem, the true CTG map along SG in absence of obstacle O is a continuous and monotonically decreasing function from S to G (Fig. 3.6(a)). In the presence of obstacle O, the CTG map between O and G remains unaffected but it shifts up between S and O, causing a discontinuity at O (Fig. 3.6(b)). If the presence of obstacle O is unknown prior to the task, the a priori CTG map would be the same as shown in Fig. 3.6(a).

For the quasi 1-D problem, the SVF update (information propagation and assimilation) rate is assumed to be infinite, i.e., a new environment information that falls in the agent's sensory range is detected and assimilated instantaneously into the SVF in

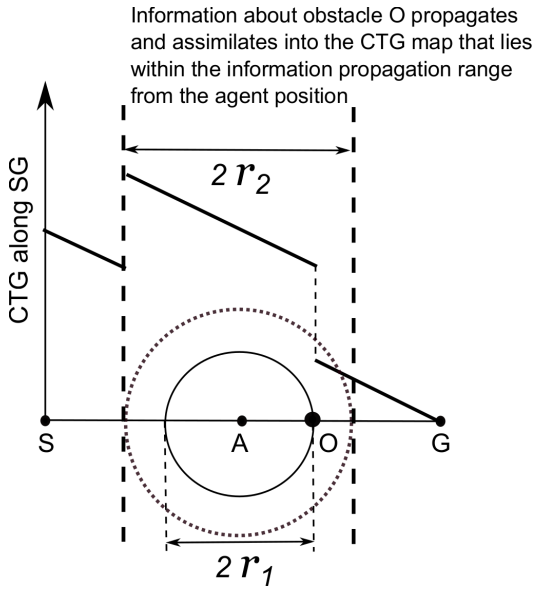


Figure 3.7: Learning through run 1.

the agent's propagation range.

3.3.2 Learning Process

Figure 3.7 illustrates the learning process through the first run starting from S. The agent A moves towards G following the negative gradient of the a priori CTG map. The circles representing the sensory and propagation ranges also move with the agent. When the sensory perimeter intersects (detects) the obstacle O the very first time, the CTG map is updated (environment information is assimilated into the CTG map) within the information propagation range. The CTG map beyond the information propagation range is unaffected, therefore only a line-segment shifts up as shown in Fig. 3.7. However, the agent still keeps moving towards G as guided by the negative gradient of CTG map. Following the first run, the agent learns the presence of the obstacle, but this information has not been propagated throughout the whole environment, and therefore, has not been assimilated properly into the global SVF (CTG) map.

Figure 3.8 depicts the learning process taking place through successive runs, and shows that the CTG map is eventually learned throughout the whole environment. For the quasi 1-D problem presented in Fig. 3.6, a single run followed by offline processing

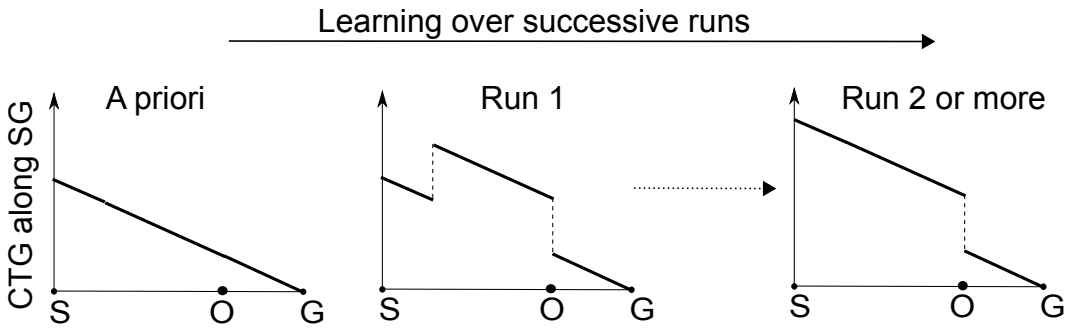


Figure 3.8: Learning through successive runs.

would be sufficient. For complicated 2-D or 3-D environments, it may take multiple runs just to capture sufficient knowledge about the environment.

The sensory range r_1 and the information propagation range r_2 are independent of each other. However, if r_2 is less than r_1, the environment that lies between the two ranges is perceived by the onboard sensors but is not assimilated into the local SVF. In such a case, the environment information between the two ranges is useless because the current planning cannot account for the information. To avoid wasting the sensory resources, the following condition should hold:

$$r_1 \leq r_2 \tag{3.17}$$

Information propagation range serves as an upper limit for the sensory range. A poor (slow) computational platform for online information processing limits the use of good sensors.

3.3.3 Application in Learning Autonomous Guidance Framework

The information propagation and assimilation model is illustrated in Fig. 3.9 for the proposed learning autonomous guidance framework. Figure 3.9 shows an illustration of the information propagation in space via local SVF updates and successive runs. N_{cycles} is the number of planning cycles in the global trajectory. In the i^{th} planning cycle, the SVF is updated in a spatial volume Vol_i around the vehicle position. Note that Vol_i corresponds to the information processing range r_2. The length of a planning cycle is proportional to H_{track}, and is less than r_2 (Eqs. (3.15), (3.16), and (3.17)). A high-speed

vehicle will travel the finite-horizon trajectory (H_{track}) faster, and therefore will require faster computations of planning cycles, i.e., a higher SVF update rate.

The SVF update in Vol_i indicates that Eq. (3.6) is consistent in Vol_i. The SVF information embedded (learned) in Vol_i is propagated back and assimilated into Vol_{i-1} during local SVF update in Vol_{i-1} in the consecutive run, if the following condition holds:

$$\text{Vol}_{i-1} \cap \text{Vol}_i \neq \emptyset \tag{3.18}$$

Condition (3.18) ensures that the environmental information embedded in the SVF over a spatial volume is not trapped locally, and propagates through space over successive planning cycles and runs. If successive runs are simulated along a trajectory (or going through the same region in the environment), the SVF eventually converges (Eq. 3.6 becomes consistent) in the region around the trajectory. In each run, the information in Vol_i is propagated back to Vol_{i-1}. Therefore, the number of runs required for the SVF convergence is nearly proportional to the path length between start and goal locations, and inversely proportional to the guidance system's sensing and information propagation ranges. For instance, if computational resources for information processing are infinite (r_2 is infinite), the SVF can be updated globally in each planning cycle and global convergence can be achieved at the end of the first run.

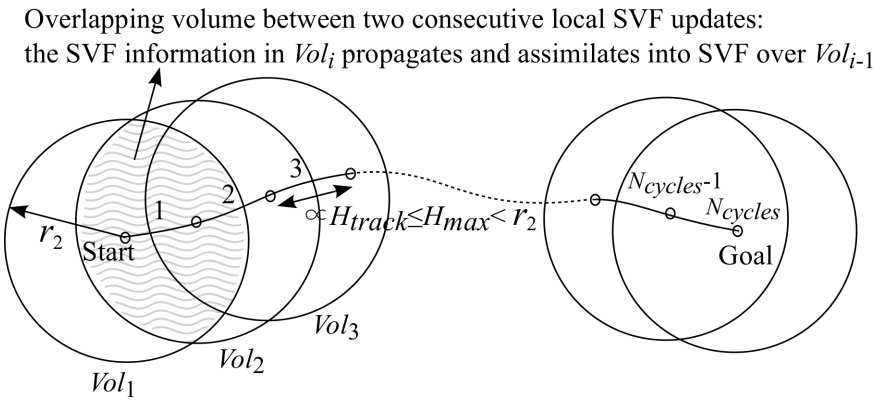


Figure 3.9: Information propagation in the environment over successive runs, via local SVF updates.

Chapter 4

Computational Investigation of Environment Learning

The guidance scheme presented in Chapter 3 is evaluated using an indoor flight simulation system. A detailed model of Akerman Hall at the University of Minnesota is used. Relevant sections of the indoor environment are labeled as shown in Fig. 4.1 for future reference. The chapter is taken from article [25].

4.1 Method and Baseline Test Case

In this dissertation, the evaluation focuses on a baseline test case designed to isolate the effects pertaining to learning and planning from sensing. The baseline setup assumes an ideal sensor (perfect depth map over specified sensor range), perfect tracking of the computed trajectory (perfect navigation and estimation, no control uncertainties or disturbances) and a static environment. The start and goal locations are shown in Fig. 4.1.

The a priori environment knowledge contains only the goal state, i.e., the a priori SVF does not account for any obstacles. The purpose of the simulation demonstration is to investigate how the performance of the system evolves with learning the SVF over successive runs from the specified start to goal states. The guidance system is simulated until SVF convergence is achieved from the same pre-specified start state. Each run is fed the SVF (CTG and VVF) map learned in the preceding run (the first

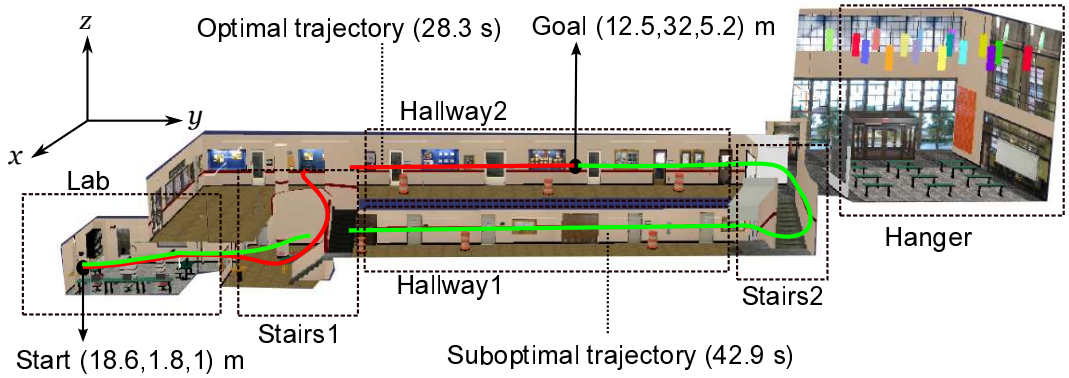


Figure 4.1: Simulated indoor environment (Akerman Hall, University of Minnesota).

run starts from the a priori SVF map). The objective function used for the SVF is the minimization of flight-time. An optimal SVF map, computed for the true environment, is used as a benchmark [176] to evaluate the learned performance through successive runs. Benchmark CTG at the start location is 28.3 s.

Figure 4.1 shows the specified start and goal locations situated in Lab and Hallway2, respectively. There are two choices after exiting Lab: take Stairs1 or follow Hallway1. The optimal trajectory (shown by red in Fig. 4.1) takes Stairs1 after exiting Lab. The green trajectory in Fig. 4.1 is a local optimal (suboptimal) solution when the Stairs1 is blocked. The benchmark flight-times for the optimal and suboptimal trajectories are 28.3 and 42.9 s, respectively.

The parameters for the guidance system are given in Table 4.1. The information propagation volume is assumed to be the same as the sensory volume. Matrices A and B for the discrete-time linear state-space system shown in Eq. (3.11) and speed levels used for the MPs are given in the Appendix. The baseline case is run offline to focus on learning processes therefore time-lag, due to non-zero computation time of planning cycles, is not considered in current study. This section investigates how flight-time, flight dynamic performance, information processing, SVF, and vehicle control/dynamic behavior in body frame evolve and converge over successive runs.

Table 4.1: Parameters for the guidance simulation

Parameter	Value
v_{max}	1.5 m/s
a_{max}	1.5 m/s^2
u_{max}	1.5 m/s
v_{zmax}	0.5 m/s
u_{zmax}	0.5 m/s
dxy	0.25 m
dz	0.2 m
n_v	6
n_{vz}	2
ψ_{diff}	45^0
N_0	5
H_{max}	5
Δt	0.2 s
ϵ_p	0.05 m
ϵ_v	0.05 m/s
Sensory volume ($\sim r_1$)	$6 \times 6 \times 2$ m^3 (center at the vehicle)
Information propagation volume ($\sim r_2$)	$6 \times 6 \times 2$ m^3 (center at the vehicle)

4.2 Performance Objective

Figure 4.2 shows the trajectories for successive runs. In the first two runs, the guidance system takes Hallway1 but in run 3 it takes Stairs1 after exiting Lab. In runs 4 and 5, the system again takes Hallway1. Trajectories in the starting runs are erratic, and not close to the optimal. For runs 6 to 20, the system takes Stairs1. Trajectories for runs 15 to 20 are similar.

Figure 4.3 shows the performance criteria (flight-time) for runs 1 to 20. The benchmark flight-time is 28.3 s, which is the benchmark CTG at the start location. For the first six runs, the flight-time is large (45-65 s) as the system often takes Hallway1 after exiting Lab. The route to the goal from Hallway1 is longer than the route from Stairs1 (see Fig. 4.1). The system takes Stairs1 in run 3 but it takes a temptative route as shown in Fig. 4.2. The flight-time eventually converges to 34.8 s. The learned CTG at the start location, after a run finishes, gives a prior estimate of flight time for the next

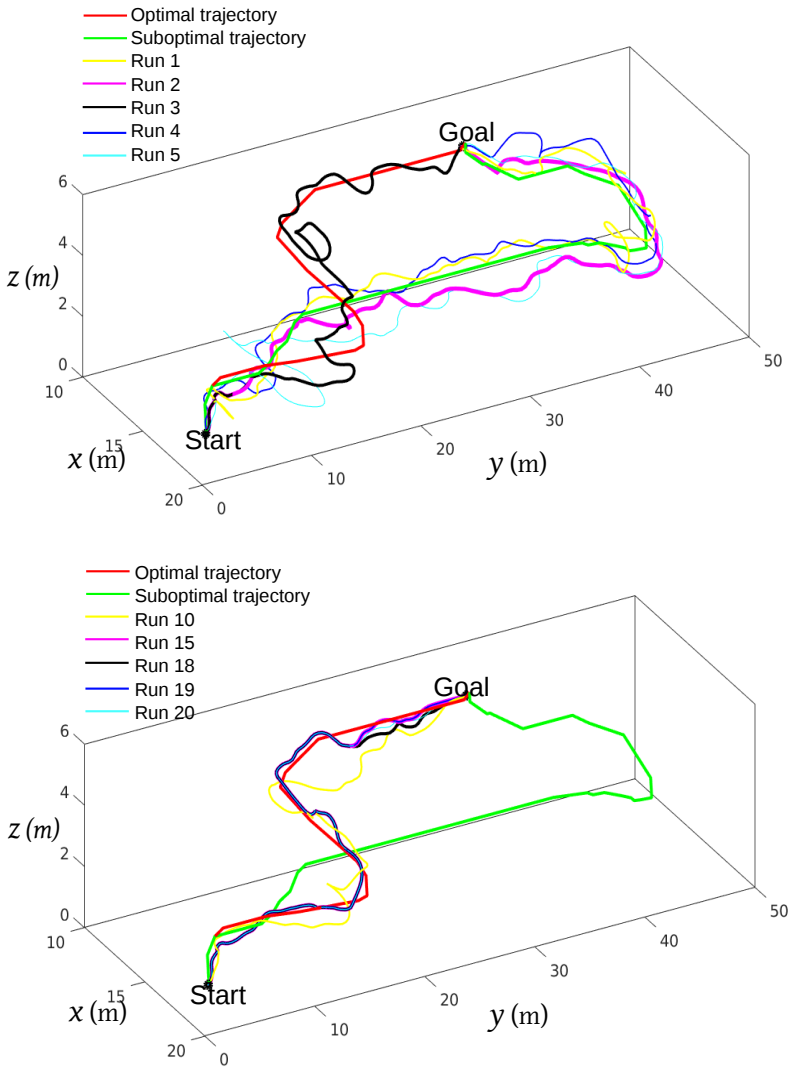


Figure 4.2: Trajectories for runs 1, 2, 3, 4, 5, 10, 15, 18, 19, and 20.

run. Figure 4.3 shows the learned (in the preceding run) CTG at the start location for all runs. The gap between the learned CTG and the benchmark CTG at the start location converges to zero over successive runs. The gap between simulated and benchmark flight-times should ideally converge to zero. Figure 4.3 shows that there remains a gap of 6.5 s (23 % of the benchmark flight-time) between simulated and benchmark flight-times even after flight trajectories and performance converge. This is due to model mismatch. The simulation uses a discrete-time linear state-space model that includes factors such as time-delay in response of applied control while the SVF (CTG) is computed using motion primitives that idealize the system behavior [176].

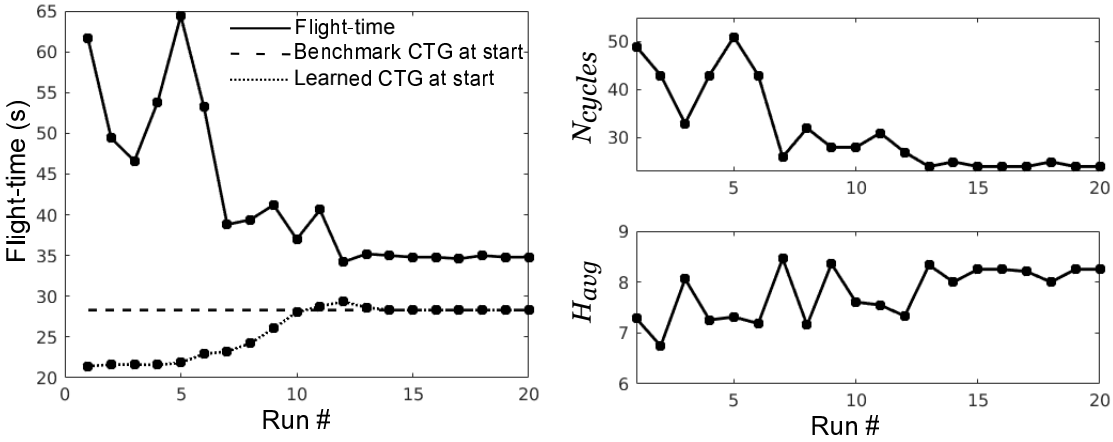


Figure 4.3: Flight-time, number of planning cycles, and average planning horizon length for successive runs.

Figure 4.3 shows the number of planning cycles (N_{cycles}) and the average planning horizon length (H_{avg}) for successive runs. N_{cycles} varies in almost the same fashion as the flight-time does over successive runs. H_{avg} changes only slightly from around 7 to 8 over successive runs. Flight-time for a run is approximately $H_{avg} \times N_{cycles}$. H_{avg} remains almost constant over successive runs. Therefore, it can be stated that flight-time is almost proportional to the number of planning cycles (or AWPs).

4.3 Flight Dynamic Performance

Figure 4.4 shows the distribution of horizontal acceleration (a_t and a_n are tangential and normal accelerations, respectively) and speed for runs 1, 5, 10, and 20. Speed distribution is almost the same over successive runs. The system uses higher speeds ($> 0.9\times\ v_{max}$) 49.8, 56.7, 47.8, and 50.1 % of total flight-time for runs 1, 5, 10, and 15, respectively. The acceleration distribution in Fig. 4.4 shows that the system is using maximum acceleration to turn more frequently in run 1 than it does in latter runs. In run 20, the dynamic behavior shows a new mode of zero acceleration. The zero-acceleration mode corresponds to flying in a straight path. The emergence of the zero-acceleration mode in latter runs is explained as follows. In run 1, the system does not know the environment and it takes sharp ($a_n \approx a_{max}$) turns to avoid unknown obstacles when they are revealed for the first time in the vehicle's path. In latter runs, the system has learned the environment and updated the SVF to account for the obstacles. The learned SVF enables the system to account for global environment knowledge in the finite-horizon trajectory planning. Therefore, in latter runs, the system flies more in straight lines and takes sharp turns less frequently. For a Dubins vehicle, the policy is consistent with turn-straight-turn behavior for time-optimal performance [177].

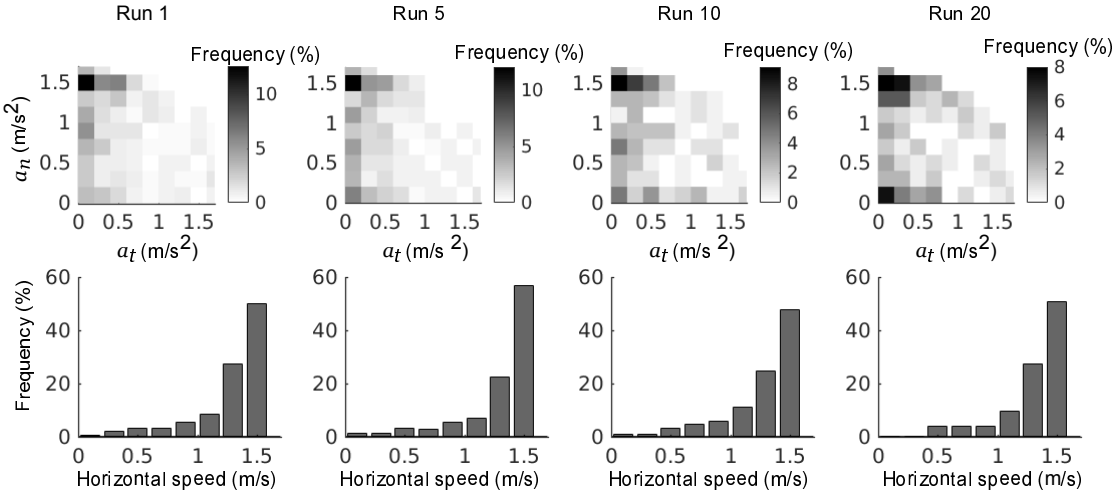


Figure 4.4: Horizontal speed and acceleration (tangential a_t and normal a_n) for runs 1, 5, 10, and 20.

4.4 Information Processing

In a planning cycle, the SVF update assimilates the sensory data into the local SVF map. The amount of change in the SVF (CTG) map represents the amount of processed information in a planning cycle. Due to the quantization of 3-D space into cells, the CTG change can be measured as the number of cells that change their CTG. To present the statistics of CTG change (update), the absolute change in CTG is divided in the following ranges: i) >1 s, ii) 0.5-1 s, iii) 0.2-0.5 s, iv) 0.1-0.2 s, and v) 0.01-0.1 s. Figure 4.5 shows the number of cells that change their CTG in each planning cycle for runs 1, 5, 10, and 20. The large changes (>1 s) in CTG map are dominant in starting runs, indicating a large amount of information being processed in those runs. After about 10 successive runs, the CTG map does not show any significant change. The learning process eventually converges and no new information is processed.

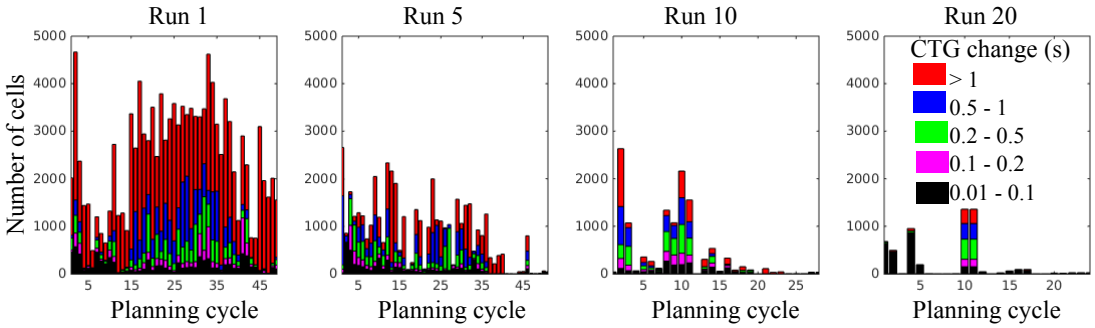


Figure 4.5: Statistics for absolute changes in CTG map for runs 1, 5, 10, and 20.

4.5 Learning SVF

The SVF represents the information needed to determine the optimal guidance behavior over the geographical space. This subsection investigates how the SVF evolves in space with learning over successive runs. The following discussion focuses on the environment section joining the Lab, Stairs1, and Hallway1 sections as shown in Fig. 4.6. The environment section is a critical decision point; therefore, investigating the SVF in that section can answer why the guidance system chooses Stairs1 or Hallway1 in a run.

Figure 4.7 shows the CTG map at two horizontal planes $z = 1.0$ and 1.8 m, in the

environment section shown in Fig. 4.6. The figure also shows the optimal choice between Hallway1 and Stairs1 after Lab exit for the next run, based on the CTG learned in a run. The a priori CTG map is a constant slope plane. The low cost region along Hallway1 justifies the system's decision to take Hallway1 after exiting the Lab in run 1. As the CTG evolves with learning over runs 1 and 2, the low cost region shifts from Hallway1 to Stairs1. Therefore, the system chooses Stairs1 in run 3. The low cost region shifts back to Hallway1 after learning in run 3, which makes the system choose Hallway1 in runs 4 and 5. After learning in run 5, the low cost region shifts towards Stairs1 and remains so for all runs beyond 5. Therefore, the guidance system chooses Stairs1 in all runs beyond 5. The CTG map after run 20 is close to the benchmark CTG (see the last two columns in Fig. 4.7).

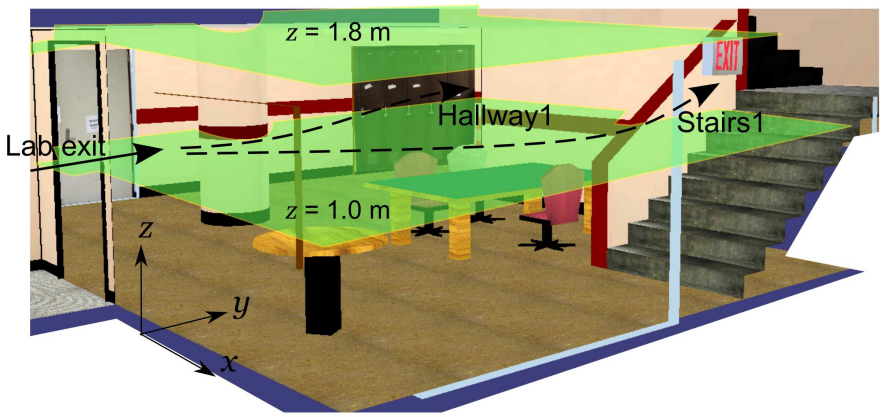


Figure 4.6: Environment section joining Lab, Stairs1, and Hallway1.

Note that the noisy CTG map (Fig. 4.7) is an artifact due to the quantized state-space that turns the optimization problem into a combinatorial problem of discrete spatial positions and fixed trajectory (motion) segments. Also, the horizontal and vertical motions are independent of each other and therefore may not saturate together. If one motion is saturated, the other motion has multiple solutions for the same optimal cost [176].

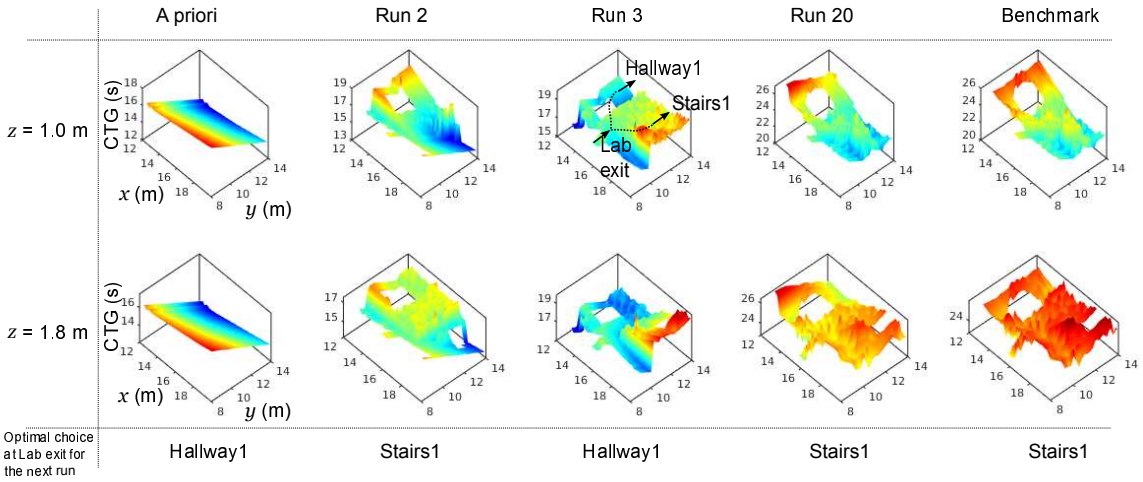


Figure 4.7: CTG at two horizontal planes (z = 1.0 and 1.8 m) in the environment section shown in Fig. 4.6.

4.5.1 SVF Convergence

A gap E between the benchmark (true optimal) and the learned CTG values for a cell in discretized 3-D space is defined as following:

$$E = |\frac{\text{CTG}_{\text{learned}} - \text{CTG}_{\text{benchmark}}}{\text{CTG}_{\text{benchmark}}}| \times \ 100\ \%. \tag{4.1}$$

Figure 4.8 shows the statistical distribution of the gap E, between the benchmark and the learned CTG maps after run 20, for the obstacle-free environment. A high gap (assume $> 5\%$) in a region indicates that the learned CTG has not converged to its benchmark value in that region. Fig. 4.8 shows that the learned CTG map has not converged to its benchmark value in 17.5 % of the free environment.

4.5.2 Flight Without Environment Sensing (FWES)

A flight without environment sensing (FWES) is a test case that simulates the guidance system from the specified start to goal locations without relying on the environment sensing. FWES is simulated after the SVF is learned and converged. In FWES, the guidance system relies on the learned SVF map. For perfect learning, the performance of FWES is expected to be close to the optimal (benchmark) solution.

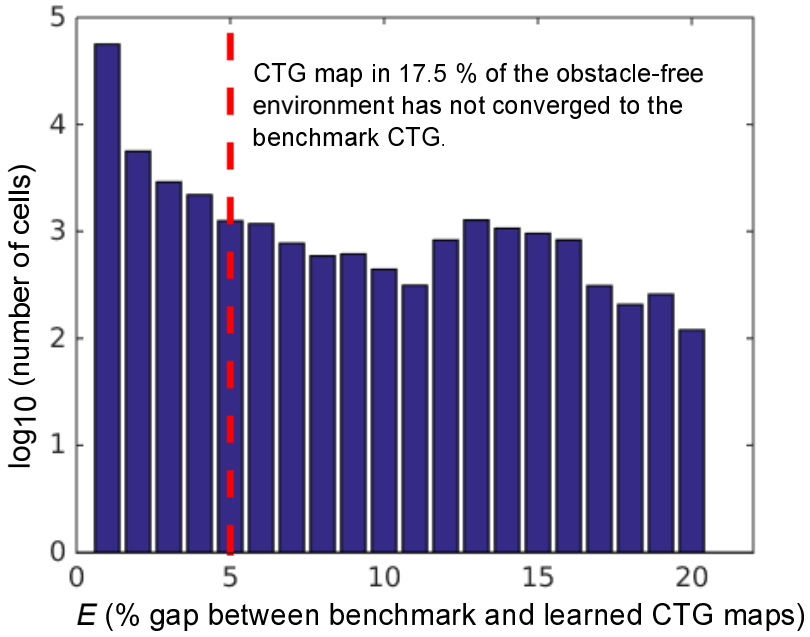


Figure 4.8: Statistical distribution of the gap E for the obstacle-free environment.

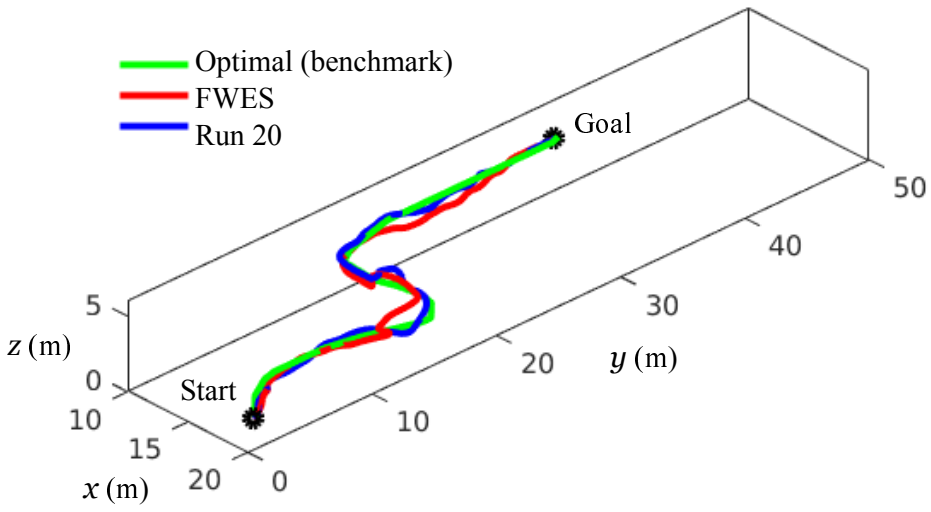


Figure 4.9: Trajectories for the FWES, benchmark solution, and run 20.

To verify the learned SVF, a FWES is simulated using the SVF learned after run 20. Figure 4.9 shows the trajectories for FWES, benchmark solution, and run 20, which are similar. For the FWES, the system takes Stairs1 after exiting Lab. The flight-time for FWES is 33.6 s, which is close to the flight-time of 34.8 s achieved after learning over successive runs (see Fig. 4.3).

4.6 Evolution in Control Behavior

This section investigates if patterns emerge in control behavior as a result of learning. Vehicle control/dynamic behavior is represented by mapping its overall dynamic performance in the body frame. The analysis in vehicle body frame helps investigate the evolution in control behavior. The global trajectory is a series of finite-horizon trajectory segments computed during each planning cycle. Figure 4.10 illustrates how to transform trajectory segments for all planning cycles into the vehicle body frame.

The longitudinal-lateral axes are attached to the vehicle. For example, a trajectory is made of three segments, or planning cycles, that are P1, P2, and P3 as shown in Fig. 4.10. All segments are transformed into the vehicle body frame through an appropriate translation and rotation. Further, the mirror symmetry about the longitudinal axis is taken care of by taking mirror images of P2 and P3. The transformation technique in Fig. 4.10 maps the global dynamic behavior in the vehicle body frame.

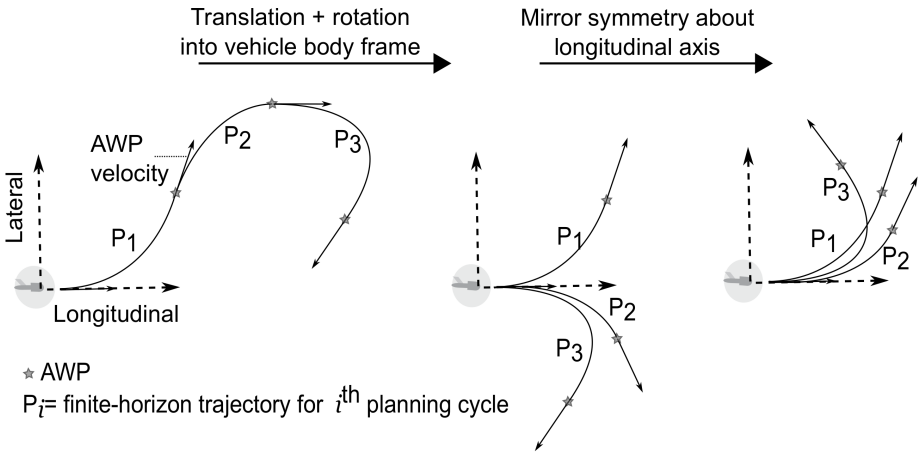


Figure 4.10: Transformation of trajectory segments for all planning cycles into the vehicle body frame.

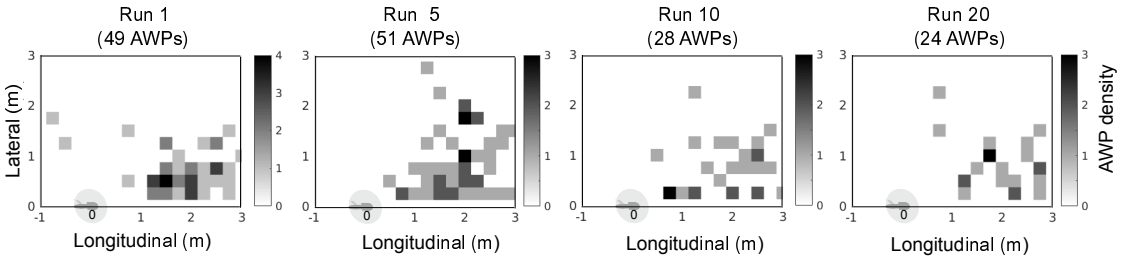


Figure 4.11: AWP distribution in the vehicle body frame for runs 1, 5, 10, and 20.

The vehicle state, in current study, includes position and velocity. The position and velocity direction are irrelevant to vehicle dynamic behavior in its body frame due to translation, rotation, and mirror symmetry, as shown in Fig. 4.10. Only vehicle speed may have a relationship with its dynamic behavior in the body frame, and the relationship is not necessarily linear. Current analysis uses the following three regimes of speed for mapping the behavior in the body frame: 1) $\geq$ 1.2 m/s, 2) $\geq$ 1.0 m/s & $<$ 1.2 m/s, and 3) $<$ 1.0 m/s.

Figure 4.11 shows the distribution of AWPs in the vehicle body frame for runs 1, 5, 10, and 20. The spatial maps for velocity and cost (time-to-go) in the vehicle body frame are extracted using the spatial averaging technique presented in [12]. The spatial resolution for the mapping is 0.2 m and averaging window is 0.4 m. Figure 4.12 shows the velocity and cost (time-to-go) maps in the vehicle body frame for runs 1, 5, 10, and 20. With learning over successive runs, the dynamic behavior gets segregated in the body frame based on the vehicle speed. At high speeds ($\geq$ 1.2 m/s), the vehicle often flies straight. At lower speeds, the vehicle flies in curved paths and the curvature increases as the vehicle speed decreases.

4.7 Iterations for Safe AWP

In a planning cycle, the guidance system iterates through the list AWP_{list} until it finds a safe AWP. n_{iter} represents the number of iterations. Table 4.2 shows the statistics of n_{iter} across all planning cycles, for runs 1, 10, and 20. The ideal performance is that it takes only one iteration in all planning cycles. The mean n_{iter} is around 2, which implies that it takes two iterations on average to find a safe AWP. The median is 1 for all runs.

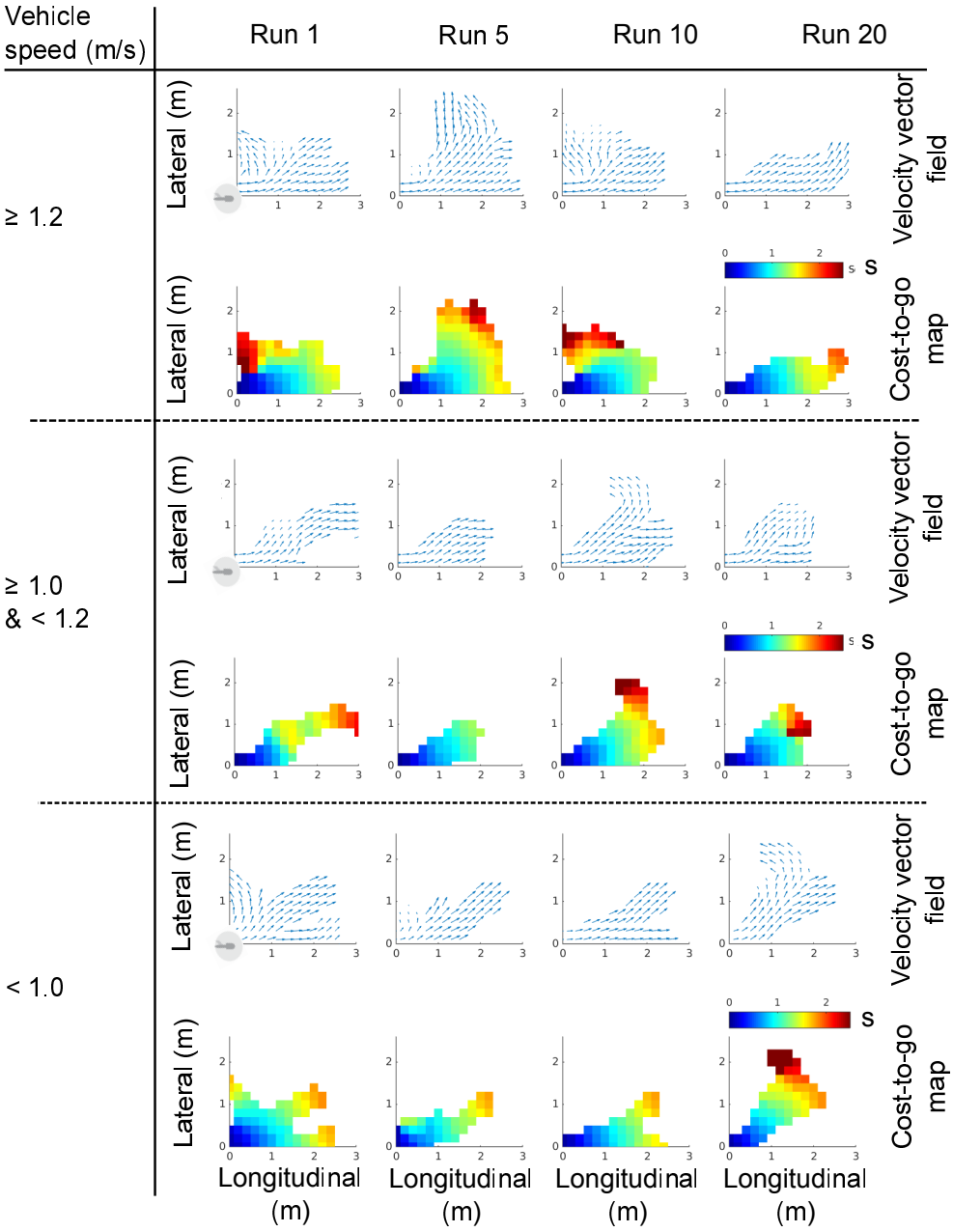


Figure 4.12: Velocity and cost (time-to-go) maps in the vehicle body frame as a function of # of runs and speed.

In run 20, 79 % of all planning cycles require only one iteration to find safe AWPs. The low frequencies of $n_{iter} \geq 5$ (Table 4.2) implies that there are few specific planning cycles that require large number of iterations to find a safe AWP. These specific planning cycles occur in particularly complex parts of the environment, e.g. see the loopy part of the trajectory in run 1 inside Lab and on Stairs2 in Fig. 4.2. Frequency of $n_{iter} \geq 5$ reduces for latter runs indicating that the guidance system learns and plans a safer trajectory (compare the trajectories for runs 1 and 20 in Fig. 4.2).

Table 4.2: Statistics for the number of iterations (n_{iter}) to find a safe AWP.

Number of iterations	Run #			
	1	10	20	Ideal
Mean	2.3	1.7	1.7	1
Variance	13.1	1.9	5.1	0
Median	1	1	1	1
Median frequency (%)	69	71	79	100
Frequency (%) of $n_{iter} \geq 5$	10	7	4	0

This study assumes a static environment but does not make any assumption about the environment layout, length-scale, obstacle shape and configuration. As discussed in [173], the dynamic fit between a vehicle dynamic scale (e.g., v_{max} and a_{max}) and the environment length-scale (e.g., street width and curvature) dictates the vehicle performance in the environment. Therefore, a designer should check the compatibility of a vehicle with its task environment.

Schouwenaars et. al [178] proposed basis states for safety guarantees in RH planning. A basis state is which a vehicle can remain in for indefinite period of time without hitting any obstacle. For example, a vehicle can loiter in a circle. To guarantee safety, the guidance algorithm presented in this dissertation can be extended to include a condition that the finite-horizon trajectory should always end in a basis state.

4.8 Emergence of Spatial Features in SVF

This subsection investigates what spatial features emerge in the SVF with learning over successive runs. Figure 4.13 draws a comparison between the a priori CTG and the learned CTG after run 20 at the horizontal plane $z = 4.7$ m in a relevant environment

section. The a priori CTG map is an inclined plane with a constant slope. C is an area on the plane $z = 4.7$ m, as shown by the dashed area with dotted lines. If a vehicle starts from any location $c \in C$ and follows the negative gradient of the a priori CTG map, the vehicle would hit the wall shown by the green boundary in Fig. 4.13. The a priori SVF (CTG) map does not account for the environment. The CTG map after run 20 has accounted for the environment and spatial features emerge in the learned SVF, as shown in Fig. 4.13.

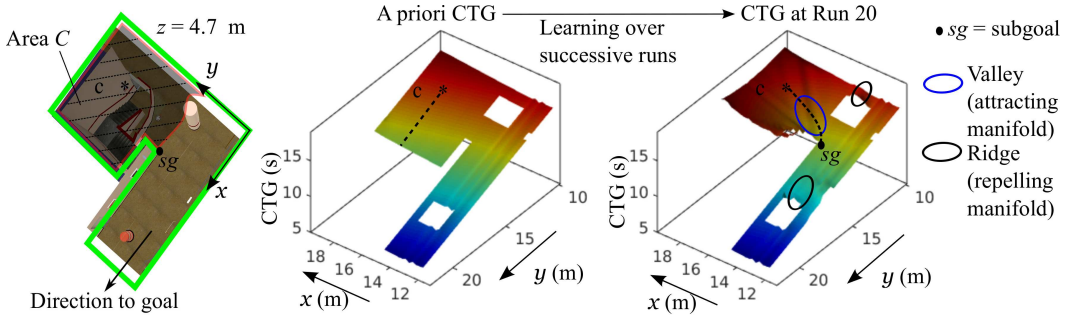


Figure 4.13: A priori vs learned CTG maps at a horizontal plane $z = 4.7$ m in the junction of Stairs1 and Hallway2.

The spatial features in SVF are called subgoals, attracting manifold, and repelling manifold [10, 7]. At a subgoal, optimal trajectories from a set of other locations converge and continue as one beyond the subgoal. An attracting manifold is a valley-like structure in the cost map, and it represents a relatively low cost region in space. A repelling manifold is a ridge-like structure in the cost map, and represents a relatively high cost region in space, e.g. behind an obstacle. If a vehicle starts from a location $c \in C$ and follows the negative gradient of the learned CTG map, it would go along the curvature as shown by the black dashed curve in the rightmost plot of Fig. 4.13. The vehicle goes along the valley in the cost map and avoids collision with the wall (green boundary). Trajectories from all locations $c \in C$ converge at subgoal sg. The converged (learned) SVF map shows spatial features as expected.

4.9 Discussion

This section discusses the significance of results shown in the previous section. It also shows the statistics of computation time for each part of the guidance algorithm.

4.9.1 Dynamic Performance

The results (Fig. 4.4) show that the modes in dynamic behavior change with learning over successive runs. In the last run, the vehicle uses more straight motion and less turning, in comparison to run 1. Speed behavior over successive runs, as can be seen in Fig 4.4, stays almost the same. A reason may be that the vehicle maneuvering capability overpowers the maneuvering requirements specified by the task environment length-scale. For the test vehicle in the simulations, the minimum turning radius at the maximum speed is 1.5 m and the minimum stopping distance is 0.75 m. The simulation environment is cluttered and has different length-scales in different regions but the environment rarely constrains the vehicle to fly through a region that has a width less than 1 m (Lab exit is 1 m wide). Such a relationship between vehicle dynamics and environment scale is called "underfit" [173]. For instance if the maximum speed is 10 m/s and the maximum acceleration stays the same as 1.5 m/s^2, the speed profile is expected to show a significant variation with learning over successive runs. Speed behavior is expected to switch from frequent abrupt variations in speed to a constant profile with less frequent accelerating/decelerating as the system learns the task over successive runs.

4.9.2 SVF Convergence

Results show that the learned SVF has not converged in 17.5 % of the obstacle-free environment. However the performance converges by run 20 and no significant change in the CTG map is recorded. Note that all runs are simulated from the same start position. The environment, where the SVF has not fully converged, is not significant to the task for the specified start and goal locations, which can be explained as follows. The guidance system starts with the a priori SVF map in run 1. The a priori SVF map is the optimal SVF for the unknown (no obstacles) environment. The guidance system is designed to always follow a trajectory that is optimal based on the current

SVF map. As changes in the environment are recorded by the onboard sensors, the system updates the SVF map and starts following the updated optimal trajectory. If a section of the task environment does not affect the optimal trajectory, that section would not be explored (for example, the Hanger in the simulations presented in this study). This enables the system to avoid the regions of space that are insignificant to the task. It is intuitive that the sensory range also plays a role in the overall learning performance. For example, with a sufficiently large sensory range, the system would be able to map the Hanger while turning from Stairs2 to Hallway2 during runs 1, 2, 4, and 5. The proposed framework allows the study of these effects.

4.9.3 Metric to Topological Representation: Spatial Features in SVF

Results (Fig. 4.13) show that spatial features emerge in the learned and converged SVF as a result of learning. The presented framework uses a metric representation for the SVF. Kong and Mettler [10, 11] showed, using a Dubins optimal solution for reaching a specified goal in an obstacle field, that the optimal guidance behavior (SVF) can be abstracted as a graph using subgoals. The optimal trajectory from any spatial position can be represented by a sequence of subgoals. In a further study with remote human pilots, Kong and Mettler [7] showed that human pilots organized their spatial guidance behavior using subgoals. Subgoals allow human pilots to develop a general strategy represented by a finite set of guidance trajectories, which are applicable in similar environments or configurations. Between two subgoals, an appropriate trajectory from the finite set is selected and applied.

If subgoals can be directly learned in real-time without requiring the metric representation of SVF, the guidance system would be more efficient in terms of computation, storage, and planning. Graph representation using subgoals only stores a directed graph of subgoals, not the optimal policy (cost and velocity maps) at every position in a quantized space. In a graph representation of SVF, the system has to compute the current optimal subgoal based on the current state and extract the SVF or optimal policy to generate controls to reach the subgoal. Such a direct learning of subgoals can enable an efficient use of onboard sensory resources. For example, onboard sensors can focus near obstacle corners that are subgoal candidates [179, 180, 181], instead of investing the sensory resources uniformly over the sensory space.

4.9.4 Computation Time

The guidance system presented in this dissertation repeatedly computes a planning cycle that includes environment sensing, SVF update, AWP candidates selection, and finite-horizon trajectory optimization. The computer used for the simulations has 32-bit Ubuntu 12.04 LTS, Intel Quad Core 2 CPU, and 3.8 GB memory. Each operation costs a non-zero computation time. For the baseline case presented in this study, environment sensing involves reading from the true environment occupancy probability map, so is a quick process. AWP candidates selection is also a quick process, as a CTC map is recalled from the long-term memory. It is the SVF update and the trajectory optimization that take significant computation times. Figure 4.14 shows the computation time statistics for SVF update and trajectory optimization (MILP in CPLEX). The runtime statistics in Fig. 4.14 are computed for the data from all runs. The spatial volume of the SVF update is $6 \times 6 \times 2$ m^3 that has 5760 cells, given spatial resolution $dxy =$ 0.25 m and $dz = 0.2$ m (Table 4.1). The number of motion primitives in the $MP_{library}$ are 217. The average computation time for the SVF update is 0.50 s with a standard deviation of 0.20 s. Sampling time for the online trajectory optimization is $\Delta t = 0.2$ s (Table 4.1). Trajectory optimization takes 0.17 s on average with a standard deviation of 0.16 s. As it was mentioned earlier in Section 3.2, bisection convex programming can be used for the trajectory optimization, which would be faster than MILP formulation.

The SVF update requires maximum runtime. One solution to reduce the runtime is to use a lower resolution for geographical space or vehicle dynamics (less number of MPs). However, it is a trade-off between the accuracy of SVF and the runtime of SVF update. The less the latter is, the closer the framework is to a real-time application.

4.9.5 Situational Awareness

In the presented learning and guidance framework, the central component is the online data assimilation (learning of SVF) that is functionally similar to the central stage "comprehension" in Endsley's SA model [154]. The online data assimilation takes environmental information from onboard sensors and incorporates the information into a guidance policy, i.e., SVF, in light of specified task objectives. Once the environment information is interpreted as guidance policy, the framework predicts a local goal

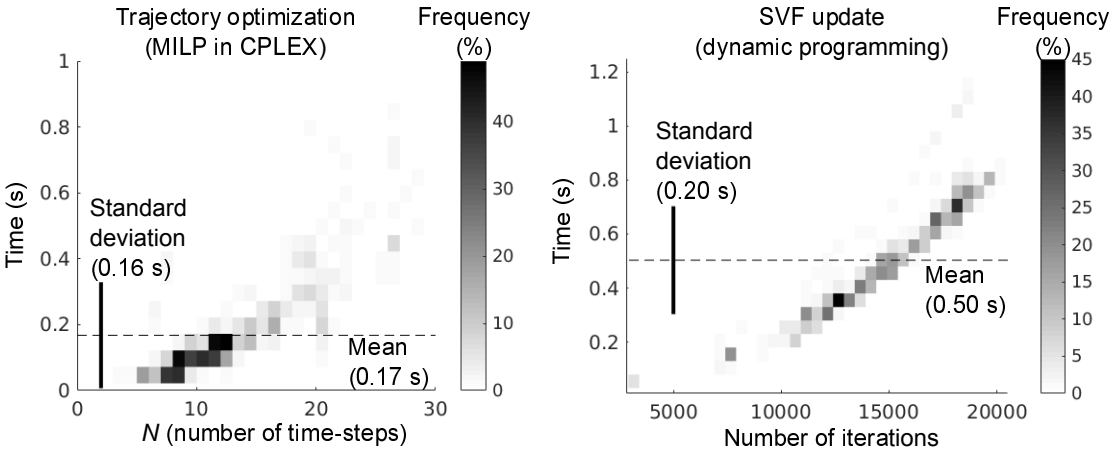


Figure 4.14: Computation time statistics for SVF update and trajectory optimization (MILP in CPLEX).

state, i.e., AWP, to determine the best immediate control action. The selection of AWP is similar to "projection" stage in the SA model [154]. Thus, the presented learning and guidance framework delineates the three stages (perception, comprehension, and projection) of SA.

4.9.6 Future Directions

The current framework has to be extended to account for practical considerations such as time-delay, path-tracking errors, uncertainties in the environment sensing due to a practical sensor model, and disturbances in the control that are always present in the integration of sensing, planning, and control.

Chapter 7 presents a modification of the learning framework that learns a topological representation of the task environment using subgoal-graph, instead of learning the table representation of SVF. The subgoal-graph framework is used to evaluate human task learning and decision making during navigation in unknown environments in Chapters 5 and 6.

Chapter 5

Human Environment Learning: Experiments and Analysis Framework

This chapter first presents the experiment system used for human guidance experiments. Next, the chapter presents a mathematical formulation of guidance task, interaction patterns, memory structure for representing and learning a guidance task, and agent-environment system. Finally, the chapter introduces an analysis framework using the proposed memory structure for environment learning and representation. This chapter is taken from article [182].

5.1 Experiments and Data

This section gives an overview of the experiment system and human data used for the investigation of human environment learning.

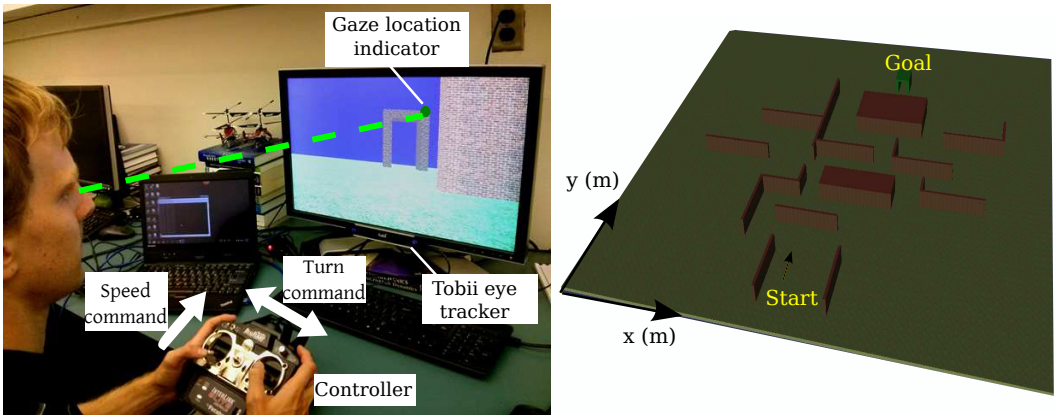
5.1.1 Experiment System

The guidance experiments were conducted on the system introduced in [16] (see in Fig. 5.1(a)). The system consists of a monitor to display a simulated task environment, a joystick to control flight behavior and navigate in the environment, and a gaze tracking

device to record 3-D gaze location. The system provides a first-person view with a limited field of view (60^0) to human subjects. The longitudinal and lateral control inputs (u_{lon} and u_{lat}, respectively) correspond to forward speed (v) and turn-rate (ω). There is a delay between speed command u_{lon} and vehicle speed v. Turn-rate is inversely proportional to the speed. Vehicle dynamic model is given in Section 5.2.

5.1.2 Experiments

Figure 5.1(b) shows the task environment used for the guidance experiments. The environment is quasi 3-D and made of vertical walls. The experiments in this dissertation involve only horizontal (planar) motion. Eight subjects participated in the experiments. The task objective was to find fastest (minimum-time) routes between pre-specified start and goal locations as shown in Fig. 5.1(b). Before the experiment, the subjects had no knowledge of the environment layout and the goal was described to them as an archway (visually distinguishable from obstacles/walls) situated north of their start orientation. Subjects performed multiple runs from the same start location. At the end of each run, flight-time was displayed on the monitor as a feedback about their performance. Each subject was instructed to try at least 20 runs or as many runs as he/she required to explore the environment in order to find the fastest route. At the end of the experiment, each subject was asked which route was the best (fastest).



(a) (b)

Figure 5.1: (a) First-person guidance experiment system proposed in [16] and (b) Task environment used for human guidance experiments presented in this dissertation.

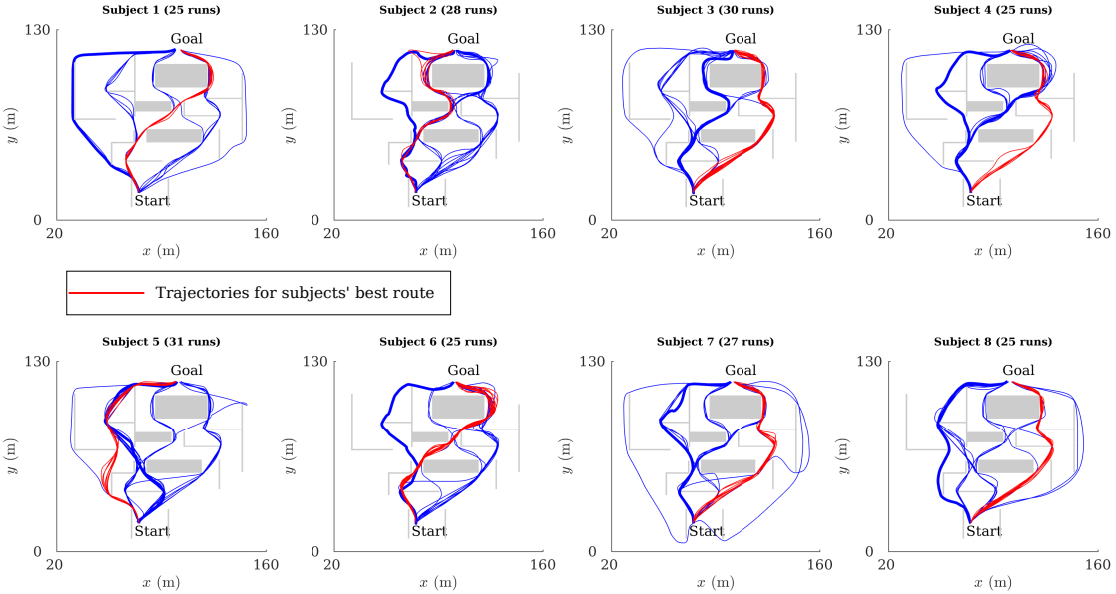


Figure 5.2: Trajectories for all runs for subjects 1 to 8.

Figure 5.2 shows trajectories for all runs for subjects 1 through 8. For each subject, trajectories on his/her best route are shown in red. Figure 5.3 shows the flight-times for runs on the best route for each subject. Subject 1 achieved the best overall flight-time of 31.0 s.

5.2 Mathematical Formulation

This section first presents a mathematical formulation of guidance task and interaction patterns. Next, the formulation is used to model memory structure for representing and learning a guidance task environment. Finally, it presents the agent-environment system and its components.

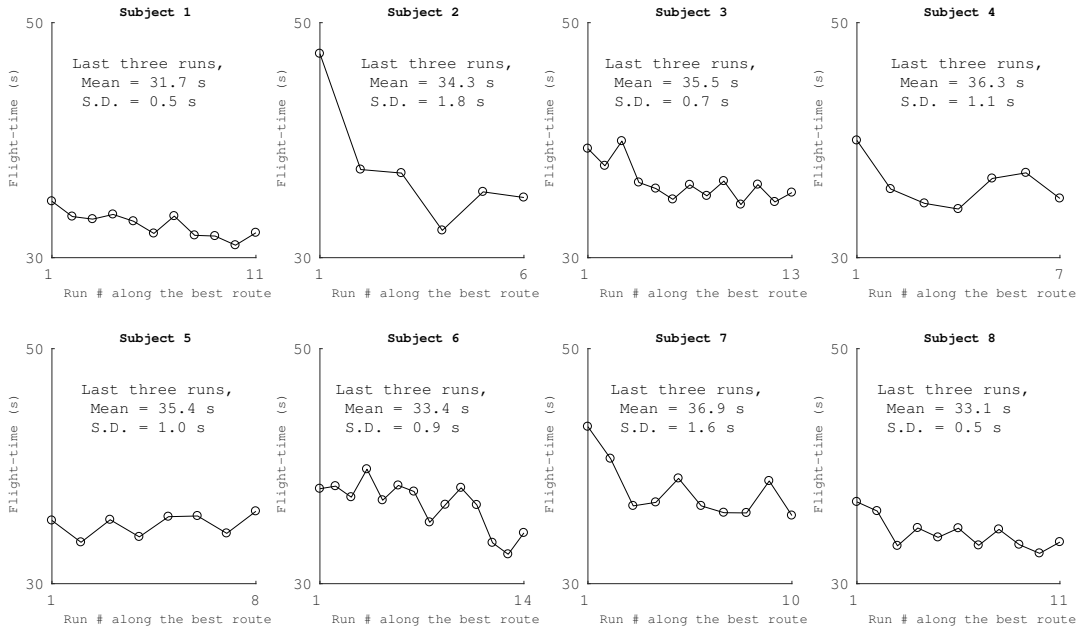


Figure 5.3: Flight-times for runs on best routes for subjects 1 to 8. S.D. is the standard deviation.

5.2.1 Guidance Task

In a guidance task, an agent travels from a state $\mathbf{x} \in \chi \subseteq \mathbb{R}^n$ to a given goal state $\mathbf{x}_g$, using control $\mathbf{u} \in \mathcal{U} \subseteq \mathbb{R}^m$. Vehicle dynamics are described by:

$$\begin{aligned} \dot{\mathbf{x}} &= f(\mathbf{x}, \mathbf{u}), \\ \mathbf{x}_p &\in \mathcal{W} \subset \chi, \end{aligned} \tag{5.1}$$

where $\mathbf{x}_p$ is spatial position vector and $\mathcal{W}$ is allowed workspace (e.g., position and orientation). The time to reach the goal is represented by t_f. A control trajectory $\overleftarrow{u}$ drives the agent from a start state $\mathbf{x}$ to the goal state $\mathbf{x}_g$. The corresponding state trajectory is represented by $\overleftarrow{s}$. The set of all feasible trajectories from all start states satisfying constraints χ and $\mathcal{W}$ is represented by $\overleftarrow{S}$, which represents guidance behavior.

An optimal trajectory ($\overleftarrow{u}^*$ and $\overleftarrow{s}^*$) minimizes a cost function J (e.g., time-to-go) as follows:

$$\underset{\overleftarrow{u}}{\text{Min}} \int_0^{t_f} J(\mathbf{x}(\mathbf{t}), \mathbf{u}(\mathbf{t}))\mathbf{dt}. \tag{5.2}$$

The set of $\overleftarrow{s}^*$ from all start states is represented by $\overleftarrow{S}^* \subset \overleftarrow{S}$, which represents optimal

guidance behavior. Optimal spatial guidance behavior $\overleftarrow{S}^*_p$ is defined over spatial position vector $\mathbf{x}_p$ space. $\overleftarrow{S}^*_p$ is the set of optimal trajectories from all $\mathbf{x}_p \in \mathcal{W}$. Spatial value function (SVF) describes optimal guidance policy (e.g., cost-to-go (CTG) and velocity maps) over geographical space for $\overleftarrow{S}^*_p$.

5.2.2 Interaction Patterns

Kong and Mettler [7] described two equivalence relations that are fundamental to the organization of spatial behavior: subgoals (g's) equivalence and the symmetry group guidance primitives (π's), in $\overleftarrow{S}^*_p$. These two equivalences provide the elements to formally describe patterns in interactions between agent dynamics and environment.

A subgoal $g \in \chi$ is a state that two trajectories $\overleftarrow{s}^*_i$ and $\overleftarrow{s}^*_j$, in $\overleftarrow{S}^*_p$, meet at and then follow a same trajectory to the goal. Trajectories related by a same subgoal g are said to be equivalent, i.e., $\overleftarrow{s}^*_i \sim_S \overleftarrow{s}^*_j$. Subgoals divide the task space $\mathcal{W}$ into partitions $\mathcal{W}_i$'s such that trajectories from all $\mathbf{x}_p \in \mathcal{W}_i$ converge to the same subgoal g_i. Therefore, trajectory $\overleftarrow{s}^*$ from a point can be represented as a sequence of subgoal states.

A trajectory segment is a continuous portion from a trajectory $\overleftarrow{s}^*_i$. If two trajectory segments π_i and π_j are equivalent after a rigid-body transformation (translation and rotation), the segments are related to same guidance primitive, i.e., $\pi_i \sim_G \pi_j$. The guidance primitive library Π is as follows:

$$\Pi = \{\pi_1, \pi_2, ...\} \tag{5.3}$$

A trajectory $\overleftarrow{s}^*$ can be represented as a string of guidance primitives.

5.2.3 Subgoal Graph

The optimal guidance solution over spatial position vector, which is $\overleftarrow{S}^*_p$, can be abstracted as a directed graph of subgoals represented by G as follows:

$$\begin{aligned} G &= [g_0\ g_1\ g_2\ ..\ g_k\ ..\ g_N], \\ (g_k)_c &= g_i\ \&\ CTG_i < CTG_k, \end{aligned} \tag{5.4}$$

where N is the number of subgoals. Goal is represented by $g_0 = \mathbf{x}_g$. CTG_k is cost-to-go to the goal state (g_0) from subgoal g_k. CTG_0 is zero. Each subgoal (other than goal)

g_k has one child subgoal g_i, i.e., there is a directed edge in the graph from node g_k to g_i. Graph edges are represented by a connection matrix Q as follows:

$$\begin{aligned} Q = [Q_{ki}]_{(N+1)\times(N+1)}, k \in [0\ ..\ N], i \in [0\ ..\ N]; \\ Q_{kk} = 0\ \forall\ k \in [0\ ..\ N]; \\ Q_{0i} = 0\ \forall\ i \in [0\ ..\ N]; \\ \forall\ k \in [1\ ..\ N],\ \exists!\ i\ (Q_{ki} = 1,\ Q_{kj} = 0\ \forall\ j \neq i). \end{aligned} \tag{5.5}$$

The matrix element Q_{ki} is 1 only if g_i is the child subgoal of g_k, otherwise Q_{ki} is 0.

State vector $\mathbf{x}$ is position $\mathbf{x}_p$ and dynamic (e.g., velocity and higher derivatives) state $\mathbf{x}_v$. In presented experiments, position vector is $[x\ y]$ and dynamic state is velocity $[v\ \psi]$ where v and ψ are velocity magnitude and direction, respectively. A subgoal g_k is $\mathbf{x}_{g_k} = [x_{g_k}\ y_{g_k}\ v_{g_k}\ \psi_{g_k}]$. The position $\mathbf{x}_{p_{g_k}} = [x_{g_k}\ y_{g_k}]$ is associated with obstacle boundaries (or corners in polygonal obstacle fields)[Give ref.]. The subgoal velocity $\mathbf{x}_{v_{g_k}} = [v_{g_k}\ \psi_{g_k}]$ depends on its child subgoal state $\mathbf{x}_{(g_k)_c} = [x_{(g_k)_c}\ y_{(g_k)_c}\ v_{(g_k)_c}\ \psi_{(g_k)_c}]$ as follows:

$$\underset{v_{g_k},\ \psi_{g_k}}{\text{Min}} \int_{\mathbf{x}_{p_{g_k}}}^{\mathbf{x}_{(g_k)_c}} J(\mathbf{x}(\mathbf{t}), \mathbf{u}(\mathbf{t}))\mathbf{dt} \tag{5.6}$$

For a low-order dynamics (e.g., no acceleration constraint), velocity direction $\psi_{(g_k)_c}$ will overlap with edges in the visibility graph of subgoal positions, which is as follows:

$$\psi_{(g_k)_c} = \tan^{-1}\left[\frac{y_{(g_k)_c} - y_{g_k}}{x_{(g_k)_c} - x_{g_k}}\right]. \tag{5.7}$$

Formulation for subgoal velocity (Eq. 5.6) is a two-point boundary value optimization, which is usually solved using numerical techniques. With a finite and efficient (e.g., non-repeating and optimized cost) library Π of guidance primitives as units for motion planning, the optimization problem in Eq. 5.6 can be converted into finding a sequence of guidance primitives to transition between subgoals. The computational cost in such approach depends on $|\Pi|$.

5.2.4 Learning

Subgoal Graph

In unknown environments, the agent has to learn the subgoal graph G. The task environment in presented experiments is made of polygonal obstacles, and therefore

subgoal positions are assumed to be associated with obstacle corners. The connection matrix for the agent is a probability distribution as follows:

$$\sum_{i=0}^{i=N} Q_{ki} = 1, \tag{5.8}$$

where Q_{ki} is the probability that g_i is the child subgoal of g_k. An approximation of a priori Q_{ki} is as follows:

$$Q_{ki} = \begin{cases} 1/M, & \text{if } V(k,i)=1 \\ 0, & \text{if } V(k,i)=0, \end{cases} \tag{5.9}$$

where M is the number of subgoals that are connected with g_k in visibility graph V. With environment learning, the child subgoal is learned, i.e., Q_{ki} shifts to 1 for a particular i and zero for all others.

Guidance Primitive Library

Π_F is the set of trajectory-segments π, i.e., motion primitives, that satisfy the vehicle dynamics f and state constraints χ. Two same trajectories are represented by a same π. $\Pi_W \subseteq \Pi_F$ is the set of trajectory-segments that emerge from interactions with environment constraints $\mathcal{W}$. $\Pi_W^* \subseteq \Pi_W$ includes trajectory-segments that are optimal for a cost function (e.g., time).

Before the environment is learned, agent's library Π can be assumed to be:

$$\Pi \subset \Pi_F. \tag{5.10}$$

When agent interacts with the task environment, Π becomes:

$$\Pi \subset \Pi_W. \tag{5.11}$$

As agent learns optimal control (e.g., a skilled pilot), Π becomes:

$$\Pi \subset \Pi_W^*. \tag{5.12}$$

When the task environment is learned, the library consists of trajectory segments that are specific for the task environment.

Learning can be assessed by changes in Π. Environment learning can be measured by two quantities: 1) reduction in $|\Pi|$ (cardinality of guidance primitive library),

2) constancy of each primitive π. Because of noises and uncertainties in real world, trajectory-segments in same π do not overlap completely but are within a threshold error (e.g., a limit on area between two trajectory segments).

5.2.5 Agent-Environment System

Figure 5.5 shows an example of first-person view of the task environment. The agent-environment system has three elements: 1) vehicle dynamics (forward speed v and turnrate ω), 2) human gaze vector $\vec{r}_g$ (distance r_g and angle θ_g in agent's body frame), and 3) environment cues .

Vehicle Dynamics

The forward speed v and turnrate ω are controlled by longitudinal (u_{lon}) and lateral (u_{lat}) inputs, respectively. Turnrate is limited based on vehicle speed. Vehicle dynamics model is as follows:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} v\cos\psi \\ v\sin\psi \\ \min(u_{lat}/v, \omega_{max}) \end{bmatrix} \qquad (5.13)$$
$$\dot{v} = k_{acc}u_{lon} - k_{drag}v,$$

where ω_{max} is the maximum allowed turn-rate. k_{acc} and k_{drag} are acceleration and drag coefficients, respectively. v_{max} is the maximum speed. In experiments, the values are set as the following:

$$v_{max} = 5.2\ m/s;\ \omega_{max} = 37.6\ deg/s; \qquad (5.14)$$
$$k_{acc} = 0.12\ m/s^2;\ k_{drag} = 0.88\ 1/s.$$

Data sampling time Δt is 0.02 s.

Environment Cues

A cue is a signal used to gain information about some property of the surrounding world. Cues can be visual, auditory, or different sensory types. Visual cues are dominant for humans. In this research, the simulated task environment is made of polygonal

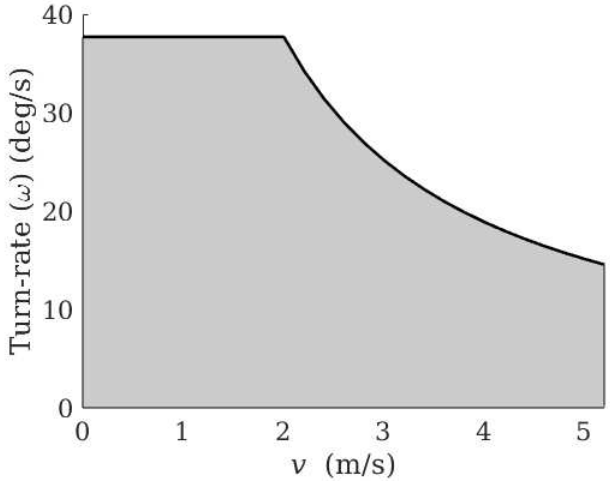
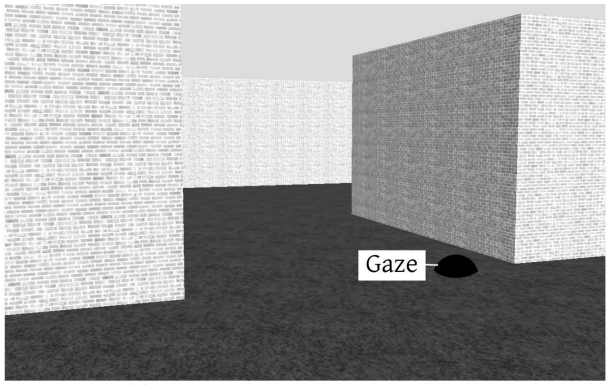


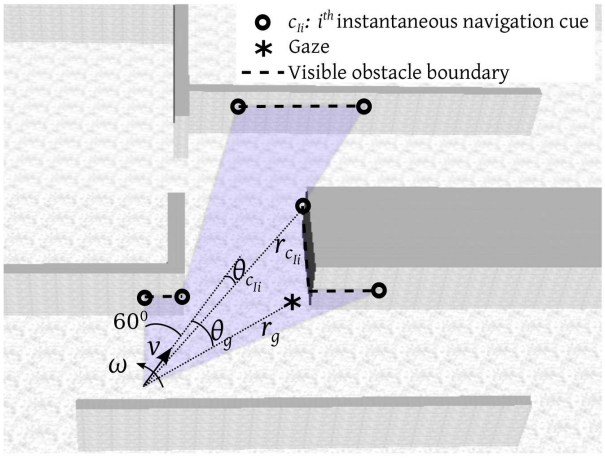
Figure 5.4: Speed turnrate envelope of the vehicle used in human experiments.

obstacles that have two primary features, edges and corners. To keep the environmental cues simple enough for analysis, the simulated environment is presented otherwise homogeneously, i.e., uniform colors for walls and ground, and no other landmarks. Even in an environment composed of polygonal walls, many types of cues are possible, such as a gap between two walls, a point on the edge, lateral or longitudinal distance from the walls. A human subject may use any of these cues to assess his/her state relative to the environment, maintain a safe distance from obstacles, or perceptual guidance (e.g., Tau guidance). For global planning, however, a subject activates a subgoal and approaches the subgoal. Obstacle corners serve as candidates for subgoals. Therefore, the corners or endpoints of the known/learned obstacle boundary can be described as global navigation cues (GNCs) that aid global path planning and navigation.

An instantaneous navigation cue (INC) is an end point on the visible obstacle boundary as shown in Figure 5.5(b). An INC is represented by $c_I = [r_{c_I}\ \theta_{c_I}]$ where r_{c_I} and θ_{c_I} are cue distance and bearing angle in agent's body frame.



(a) First-person view to a human subject



(b) Top view

Figure 5.5: Agent-environment system measurements.

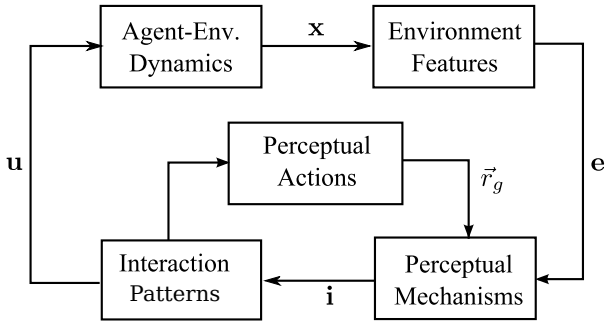


Figure 5.6: Agent-environment dynamics.

Agent-Environment Dynamics

Warren [13] described closed-loop agent-environment dynamics (Fig. 5.6) using the following formulation:

$$\dot{\mathbf{x}} = f(\mathbf{x}, k(\mathbf{x}, h(g(\mathbf{x})))), \tag{5.15}$$

. The agent is considered to be embedded in the environment. In the closed-loop model Eq. 5.15, $g(.)$ describes how the agent state affects the environment state $\mathbf{e}$. For example, environment state can be defined by relative position and orientation of obstacles and navigation cues c_I's (subgoal heuristics), which depend on the agent's current state. Next, perceptual processes $\mathbf{i} = \mathbf{h}(\mathbf{e})$ use environment cues to extract information $\mathbf{i}$. For example, relative bearing of obstacles can be used to estimate motion gap for perceptual guidance. Navigation cues are used for subgoal selection (decision-making) using a priori known and learned knowledge about task structure (subgoal graph). Next, the agent applies control $\mathbf{u} = \mathbf{k}(\mathbf{i})$ based on a guidance primitive π_k from its guidance primitive library Π, and moves gaze in a coupling with π_k.

5.3 Analysis Framework

This section first uses a Dubins vehicle to illustrate the subgoal graph for the task environment used in human guidance experiments. Second, the section applies the subgoal graph model presented in Section 5.2 for human data processing. Third, it presents an optimal (benchmark) decision-making model to evaluate human decision-making. Fourth, the section presents an exploration metric. Finally, a clustering method to extract guidance primitives is presented.

5.3.1 Benchmark Subgoal Graph

This dissertation uses the time-optimal solution for a Dubins vehicle (speed and turning radius of $v_{max} = 5.2\ m/s$ and $1\ m$, respectively) as a benchmark solution for the task environment shown in Fig. 5.1(b). Figure 5.7 shows the optimal cost(time)-to-go and velocity vector field for the benchmark solution. The structures such as subgoals and repelling manifold, as described in [10], can be seen in the velocity map in Fig. 5.7. For the optimal Dubins solution, subgoal locations coincide with obstacle corners.

Figure 5.7 also shows the subgoal graph representation, based on the benchmark solution in Fig. 5.7, for the task environment. A subgoal graph is a directed graph as shown in Fig. 5.7. Terms 'subgoal' and 'node' are used interchangeably in this dissertation. The solution from each point in free space goes to a subgoal and then it follows a sequence of subgoals (nodes). For example, the subgoal sequence from the start location is start$\rightarrow 33 \rightarrow 28 \rightarrow 26 \rightarrow 18 \rightarrow 11 \rightarrow 9 \rightarrow 5 \rightarrow 2 \rightarrow 1$(goal) .

An optimal subgoal graph satisfies the dynamic programming formulation (Algorithm 1 in [181]) as follows:

$$CTG_k = \min_{i \in [0 \ .. \ N] \setminus k} (DC_{ki} + CTG_i) \, \forall k \in [1 \ .. \ N], \tag{5.16}$$

where DC_{ki} is the incremental cost-to-go from subgoal g_k to subgoal g_i. DC is $(N + 1) \times (N + 1)$ matrix. A transition from g_k to g_i is allowed only if the optimal trajectory from g_k to g_i in the absence of obstacles is collision-free in the presence of obstacles.

5.3.2 Human Data Processing

N Nodes, CTG, DC, and Q are a priori unknown to subjects. Subjects arguably learn these quantities over successive runs. This section describes how to extract learned cost-to-go and node connectivity information from human data.

A characteristic of a time-optimal trajectory is that it passes close to obstacle corners. This attribute can also be seen in human trajectories (see Fig. 5.2). This characteristic of time-optimal solutions enable the presentation of a human trajectory as a sequence of subgoals $[k_1 \ k_2 \ .. \ k_i \ k_{i+1} \ .. \ 0]$, where k_i is the index of subgoal g_{k_i} in the benchmark subgoal graph. Human cost-to-go at a subgoal g_{k_i} is represented by CTG'_{k_i} and is extracted from a trajectory as follows:

$$CTG'_{k_i} = t_0 - t_{k_i}, \tag{5.17}$$

where t_0 and t_{k_i} are times at goal and at trajectory point closest to the subgoal g_{k_i}'s position, respectively. CTG'_{k_i} from a run is tracked in a list $CTG'_{k_i list}$. For a human subject, Q' is initiated as a zero matrix. In each run, Q' is updated as follows:

$$Q'_{k_i k_{i+1}} = Q'_{k_i k_{i+1}} + 1. \tag{5.18}$$

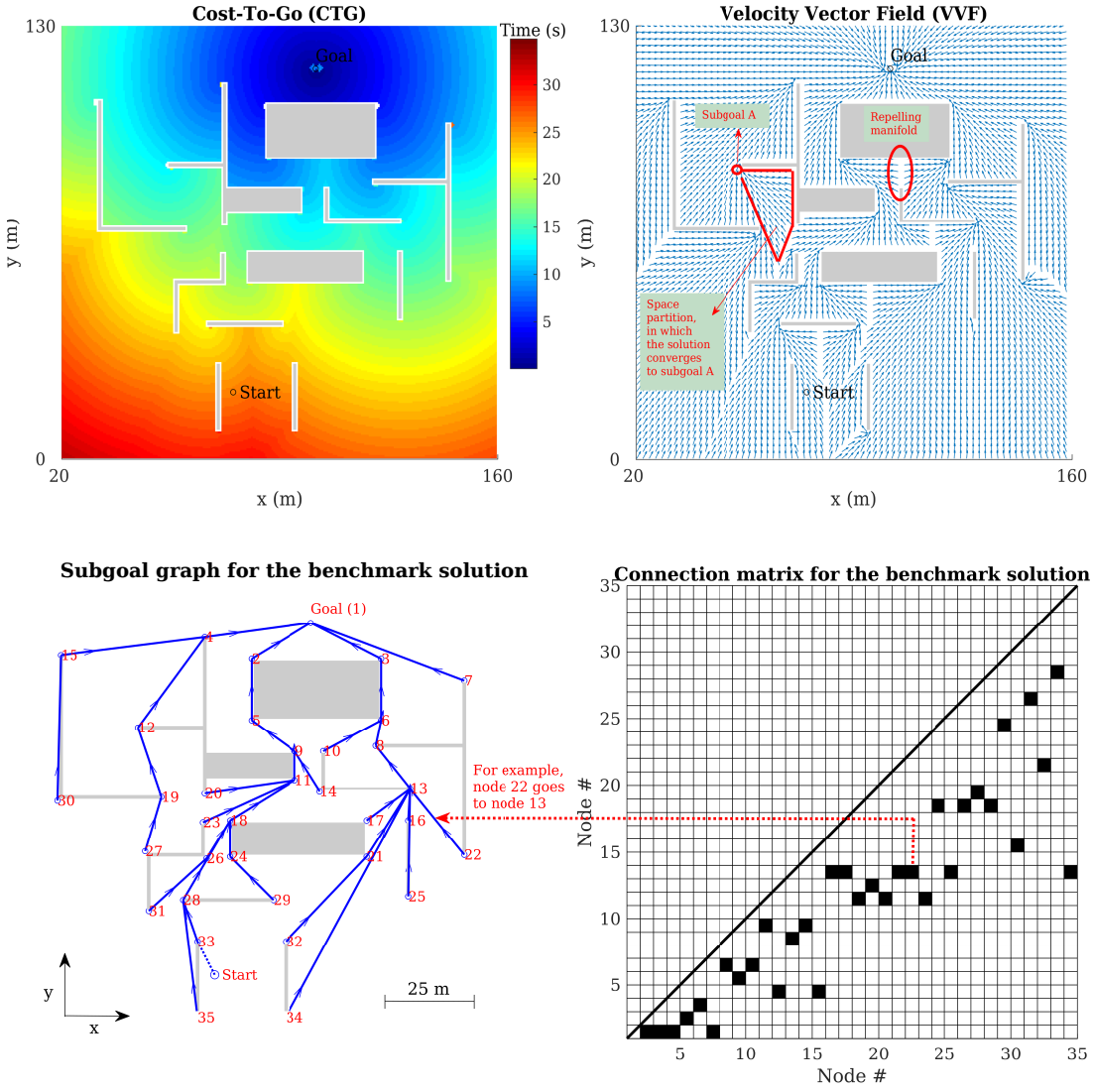


Figure 5.7: Benchmark solution: Dubins optimal solution, subgoal graph, and connection matrix.

Incremental cost between consecutive subgoals in human trajectory is extracted as follows:

$$DC'_{k_i k_{i+1}} = t_{k_{i+1}} - t_{k_i}. \tag{5.19}$$

$DC'_{k_i k_{i+1}}$ from each run is stored in a list $DC'_{k_i k_{i+1}\, list}$.

In the presented framework, human environment knowledge is represented by cost-to-go at nodes ($CTG'_{k_{list}}$), travelling cost from one node to another ($DC'_{ki_{list}}$), and number of times a segment from one node to another has been travelled (Q'_{ki}). The following are definitions regarding human knowledge about the environment, which will be used to present a decision-making rule later in this section:

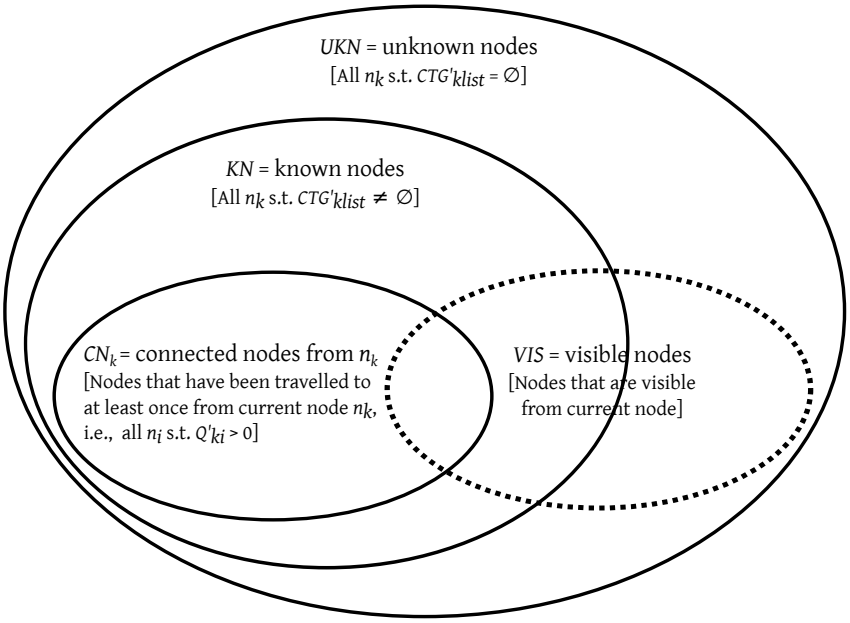


Figure 5.8: Known, unknown, connected, and visible nodes.

Definition 1 *Unknown Nodes (UKN) is the set of all nodes that have never been visited, and is presented as follows:*

$$UKN = \{k \in [1 \,..\, N] : CTG'_{k_{list}} = \emptyset\} \tag{5.20}$$

Definition 2 *Known Nodes (KN) is the set of all nodes that have been visited at least once, and is presented as follows:*

$$KN = \{k \in [1 \,..\, N] : CTG'_{k_{list}} \neq \emptyset\} \tag{5.21}$$

Definition 3 *Connected Nodes at a node n_k, represented by CN_k, is the set of all nodes that have been travelled to from the node n_k, and is presented as follows:*

$$CN_k = \{i \in [0 \,..\, N] : Q'_{ki} > 0\} \tag{5.22}$$

Visible Nodes

In the presented experiments, subjects have a limited field of view (60^0) which is expected to affect their exploratory behavior and choices of routes. A subject has to decide which node to go to after the current node. To study the effect of visibility on decision-making, the set of nodes that are visible from the current node n_{curr} is tracked in VIS. t^* is the time at which trajectory is closest to n_{curr}. This dissertation uses a time window t_w around $t = t^*$ to evaluate all nodes visible at any instant from $t = t^* - t_w/2$ to $t = t^* + t_w/2$. They are then stored in VIS. If t_w is too big, there are too many overlaps and variables are confounded. A very small t_w is unrealistic from human attention span standpoint. Therefore, it is necessary to identify t_w that explains human behavior and decision-making at nodes. At this point, t_w is set to 1 s.

5.3.3 Decision-Making Model

This section presents Dijkstra's algorithm for shortest path search in human-learned subgoal graph. The algorithm gives a decision-rule to evaluate human decision-making in navigation tasks.

Decision Cases at a Node

At a node, there are two primary types of behavior possible (see table 5.1): exploration or exploitation, which correspond to trying a new solution or repeating a known solution, respectively. In exploration mode, a subject at a current node n_k goes to a next node n_i that was never visited from n_k ($Q'_{ki} = 0$ or $n_i \notin CN_k$) in preceding runs. In exploitation mode, the subject goes to a next node n_i that was previously visited from the current node n_k ($Q'_{ki} > 0$ or $n_i \in CN_k$) in one or more preceding runs.

Table 5.1 shows the three types of decision-making scenarios (called cases A, B, and C) at a current node n_k. In case A, there is no connected node ($|CN_k| = 0$) from node n_k, i.e., there is no node n_i that $Q'_{ki} > 0$. In cases B and C, there are only one connected

Table 5.1: Choice at a node n_k.

Decision case	Choices
A) $\|CN_k\| = 0$	Exploration: go to any node
B) $\|CN_k\| = 1$	1) Exploitation: go to the node $n_i \in CN_k$ 2) Exploration: go to a new node $n_i \notin CN_k$
C) $\|CN_k\| > 1$	1) Exploitation: go to a node $n_i \in CN_k$ (what is the decision-rule?) 2) Exploration: go to a new node $n_i \notin CN_k$

node ($|CN_k| = 1$) and two or more connected nodes ($|CN_k| > 1$), respectively, from node n_k. Frequency of case A reduces and increases for cases B and C as a subject learns the environment over successive runs.

Decision-Making Model

Figure 5.9 presents a decision-making model based on the Dijkstra's shortest-path search method proposed in [27]. The model is used to select the best node to go in case C (table 5.1). The decision-making model has two parameters: discount factor (γ) and maximum depth (D_{max}) for graph pruning. In a run, the model uses the $CTG'_{k_{list}}$, $DC'_{ki_{list}}$, and Q' information extracted from data in preceding runs. At any node, the model uses Dijkstra's algorithm to search for the shortest path to the goal node. The graph is expanded from a node using Q' information. The cost of an edge is given by a function $f(DC'_{ki_{list}})$. This function, for instance, can be mean, minimum, maximum, or median. In this dissertation, f is the minimum function, i.e., a greedy approach. Humans' limited working memory is accounted for by setting a maximum search depth D_{max}. If the goal is not found after expanding the graph to depth D_{max}, the cost-to-go from a node n_k at depth D_{max} is approximated by $f(CTG'_{k_{list}})$. The model also uses a discount factor γ ($0 < \gamma \leq 1$). The cost at depth d is weighted by γ^{depth}. Therefore, the lower the discount factor, the less importance the model gives to the cost at a depth. Discount factor models if a subject is biased towards immediate (local) cost than global cost.

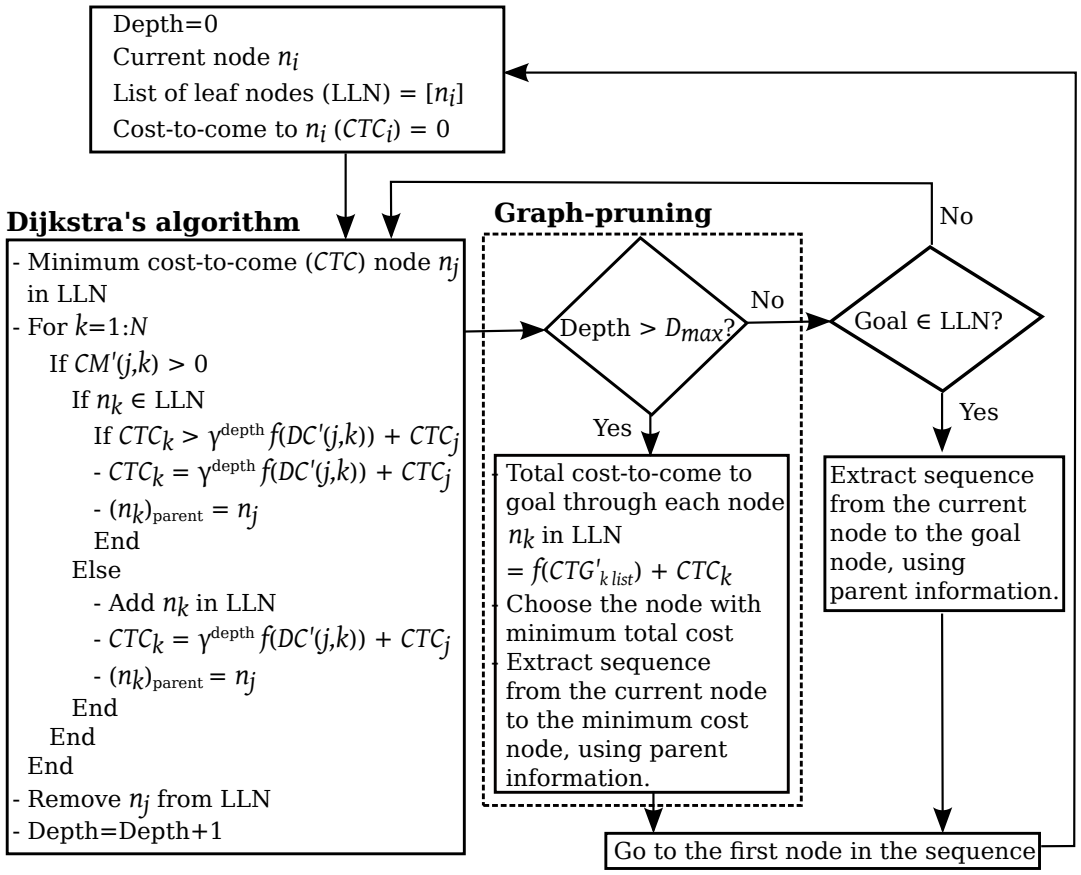


Figure 5.9: Decision-making model: Dijkstra's algorithm with discount factor γ and graph pruning at maximum depth D_{max}.

5.3.4 Exploration Metric

Learning or search tasks in general involves trade-off between exploration (learning new knowledge) and exploitation (using current knowledge to make optimal decisions) [161]. In this dissertation, the connection matrix extracted from human data is used to quantify exploration behavior. Q'_{ki} gives the number of times the segment associated with the edge $n_k \rightarrow n_i$ is taken by a subject. This information is used to determine M_h which represents the number of segments that are taken h times. An exploration metric EM is calculated as follows:

$$EM = \sum_{h=1}^{h=\infty} \left(\frac{M_h}{h} \right) \tag{5.23}$$

A large EM corresponds to when a subject explores many different segments only a few times (e.g., once or twice) and a small EM results from a subject taking a subset of edges many times. EM is a measure of exploration behavior of a subject.

5.3.5 Extracting Guidance Primitives (GPs)

In human data, it is observed that at large distances from obstacle corners subjects mostly travel in straight lines at high speeds. Agent-environment interactions take place when subjects pass close to obstacle corners. The hypothesis for task environment learning is that a pilot learns invariant perceptual and guidance strategies, i.e., guidance primitives [7], in interactions with the task space. The analysis of guidance behavior in this dissertation focuses on trajectory segments in vicinity of corners. For this purpose, trajectories are aggregated and described in a common reference frame. Fig. 5.10 shows the corner-frame used to investigate the guidance primitives. The corner-frame axes are the bisectors of angles formed by walls (boundaries) that meet at the corner.

First, candidate guidance primitive (GP) segments are extracted as follows. Trajectory segments are transformed into corner frame by translations, rotations, and reflections. Time-origin ($t_c = 0$) for a trajectory in corner frame is set at the closest point to the corner (see Fig. 5.10). A trajectory segment s_i in corner frame is a sequence of points as follows:

$$\begin{aligned} s_i = \{.., (x^i_{cl}\ y^i_{cl}), ..\},\ l \in [1\ ..\ L], \\ t_c(1) = -T,\ t_c(L) = T, \end{aligned} \tag{5.24}$$

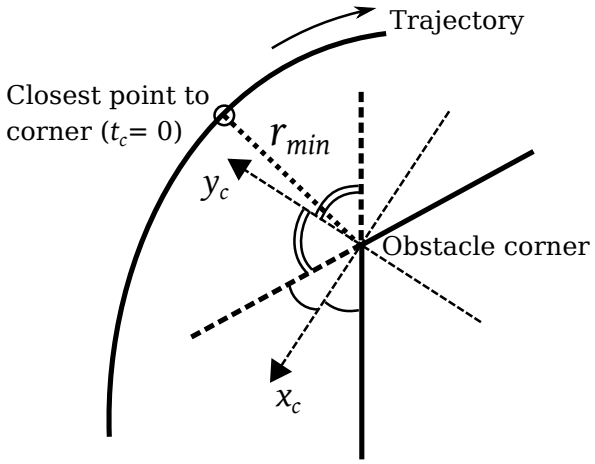


Figure 5.10: An example trajectory in corner-frame.

where $2T$ is the time-duration of trajectory segment considered for subsequent analysis of candidate GPs. L is the number of discrete points in time-duration $2T$. Distance d_s^{ij} between two trajectories s_i and s_j is defined as follows:

$$d_s^{ij} = \sum_{l=1}^{l=L} w\sqrt{(x_{cl}^i - x_{cl}^j)^2 + (y_{cl}^i - y_{cl}^j)^2}, \tag{5.25}$$

$$w = 1 - \frac{|t_c(l) - T|}{2T}.$$

The distance in Eq. 5.25 is based on points that have the same time-instant, which distinguishes trajectories that are similar in geographical space but have different motion behavior (e.g., speed and turnrate). Points on trajectory segments are weighed based on how far they are from closest point to corner.

Distance d_π^{IJ} between two clusters π_I and π_J is the average distance between all pairs of trajectories $s_i \in \pi_I$ and $s_j \in \pi_J$ as follows:

$$d_\pi^{IJ} = \frac{1}{|\pi_I||\pi_J|} \sum_{i=1}^{i=|\pi_I|} \sum_{j=1}^{j=|\pi_J|} d_s^{ij}, \tag{5.26}$$

where $|\pi_I|$ is the number of trajectories in I^{th} cluster, i.e., π_I. Trajectories are clustered using the bottom up hierarchical clustering. Each trajectory starts as a single cluster. As moving up the hierarchy, two closest (minimum d_π^{IJ}) clusters are merged. The process is repeated until a specified number of clusters is achieved.

Chapter 6

Human Environment Learning: Results and Analysis

This chapter presents an analysis of human data using the framework proposed in the previous section. First, it presents general observations that focus on planning, exploration, convergence in CTG at subgoals, and evolution in control and gaze behavior with environment learning. Finally, the chapter presents a quantitative analysis of guidance primitives associated with interaction patterns that emerge with environment learning. This chapter is taken from article [182].

6.1 Planning (Decision-Making)

Figure 6.1 shows the decision model accuracy (for $D_{max} = \infty$ and $\gamma = 1$) and mean and standard deviation of flight-time for each subject's last three runs on their best route. Model accuracy and flight-time correspond to operator rationality and performance, respectively. It is reasonable to assume that a better model accuracy should result in a lower flight-time. The best line fit between model accuracy and flight-time is shown by the dotted line in Fig. 6.1. Subject # 1 is the best, i.e., maximum accuracy (87.5 %) and best flight-time (mean and standard deviation are 31.7 s and 0.5 s, respectively). Subject # 8 is an outlier and achieves the second best flight-time (mean and standard deviation are 33.1 s and 0.5 s, respectively) despite the worst model accuracy (56.3 %). Subject # 7 shows the worst flight-time (mean and standard deviation are 36.9 s and

1.6 s, respectively) and second worst model accuracy (57.1 %).

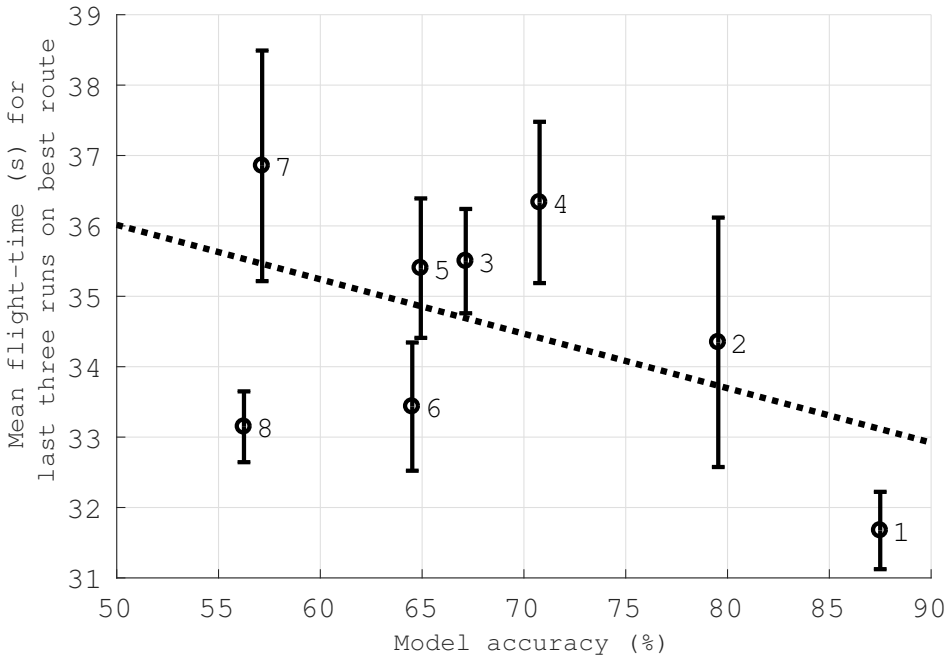


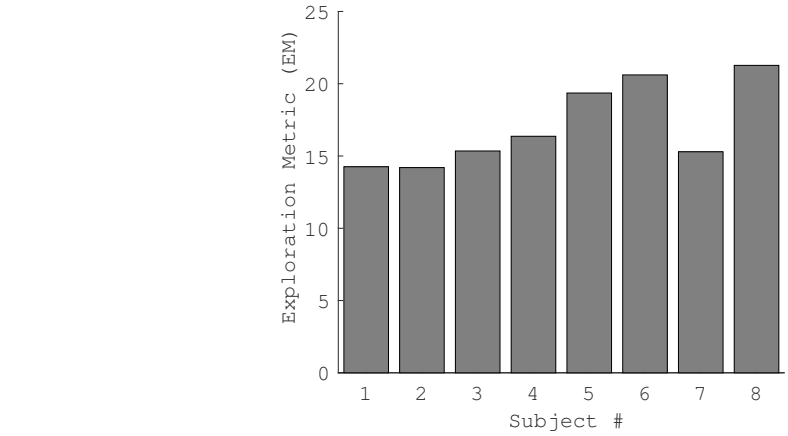
Figure 6.1: Flight-time on a subject's best route vs the model accuracy.

6.1.1 Exploration vs. Exploitation

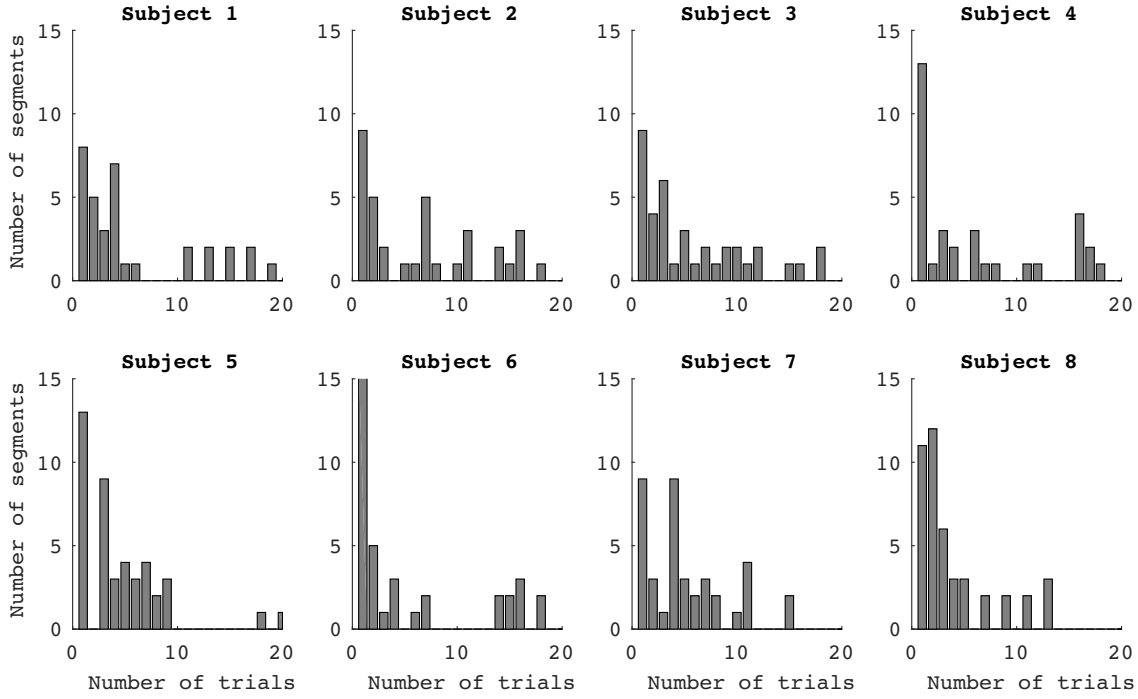
Figure 6.2(a) shows the exploration metric (EM) for all subjects. Subject # 8 has the largest $EM = 21.3$. Figure 6.2(b) shows the distribution of segments based on their trial frequency. Subject # 8 tries several segments few times unlike other subjects. This high exploration tendency of subject # 8 may be a reason why the subject has the lowest model accuracy (56.3 %) despite the second best flight-time (mean is 33.1 s) on its best route.

6.1.2 Visibility

The simulation system models the environment that is within the field of view (60^0) of an operating subject. A node is visible if it is in the field of view and not obscured or hidden by obstacles. Figure 6.3 shows the number of occurrences that the next node



(a)



(b)

Figure 6.2: (a) Exploration metric (EM) and (b) Distribution of segments based on trial frequency.

n_{next} chosen by a subject is $\in VIS$, $\notin VIS$, or $VIS = \{\}$. It can be seen that subjects often (mean frequency is 93 % for all subjects) choose visible nodes when there is any.

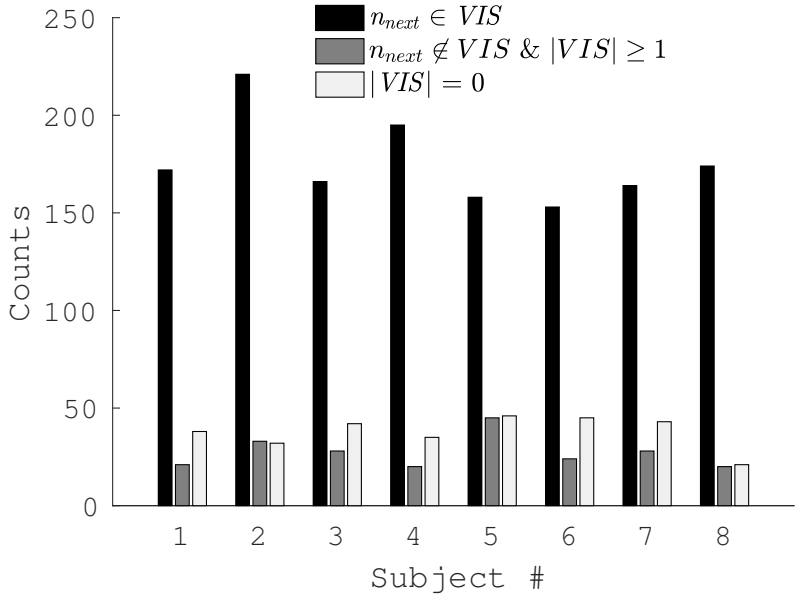


Figure 6.3: Number of occurrences for $n_{next} \in VIS$, $n_{next} \notin VIS$, and no visible nodes for all subjects.

6.2 Environment Learning

This section compares subjects # 1 and # 7 who give best and worst flight-times, respectively, for environment learning analysis. Figure 6.4 shows speed time-histories for first and last runs on best routes of subjects # 1 and # 7. In starting runs, subjects slow down as they approach any obstacle corner (or subgoal g_k) because parent subgoal $(g_k)_p$ and therefore subgoal velocity $[v^{g_k}\ \psi^{g_k}]$ are unknowns in starting runs. As the environment is learned, subgoal network and velocities are learned. In later runs, subjects reduce speed, when approaching a subgoal g_k, based on turning required to align with the next (parent) subgoal $(g_k)_p$.

Figure 6.6(a) shows frequencies of high-speeds ($\geq$ 90 % of v_{max}) for starting (1-15) and final (16-last) runs for subjects # 1 and # 7. The frequencies are computed using trajectory data near corners (within time-window $T = 2\tau$ from a corner, where $\tau = 1.13$

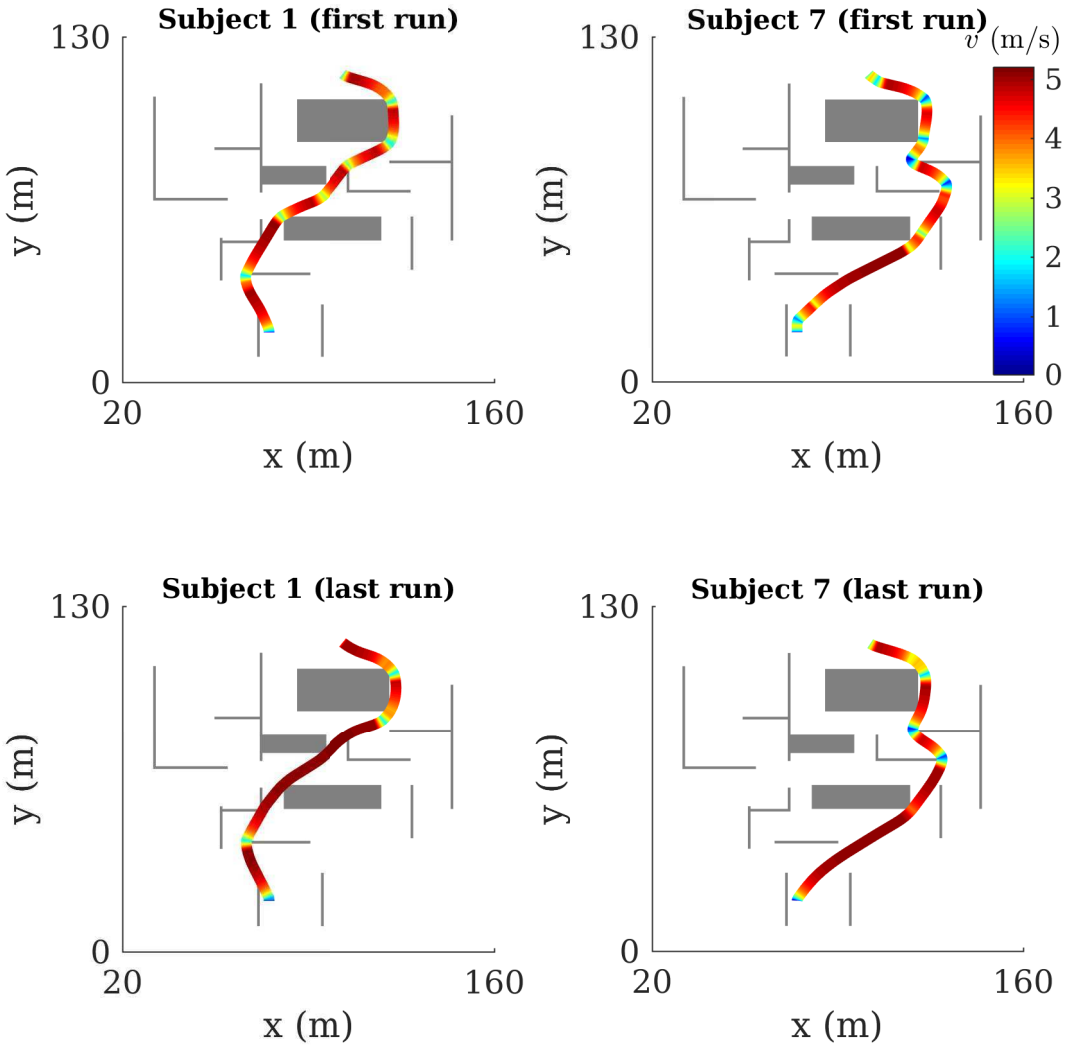


Figure 6.4: Speed trajectories for first and last runs on best routes of subjects # 1 and # 7.

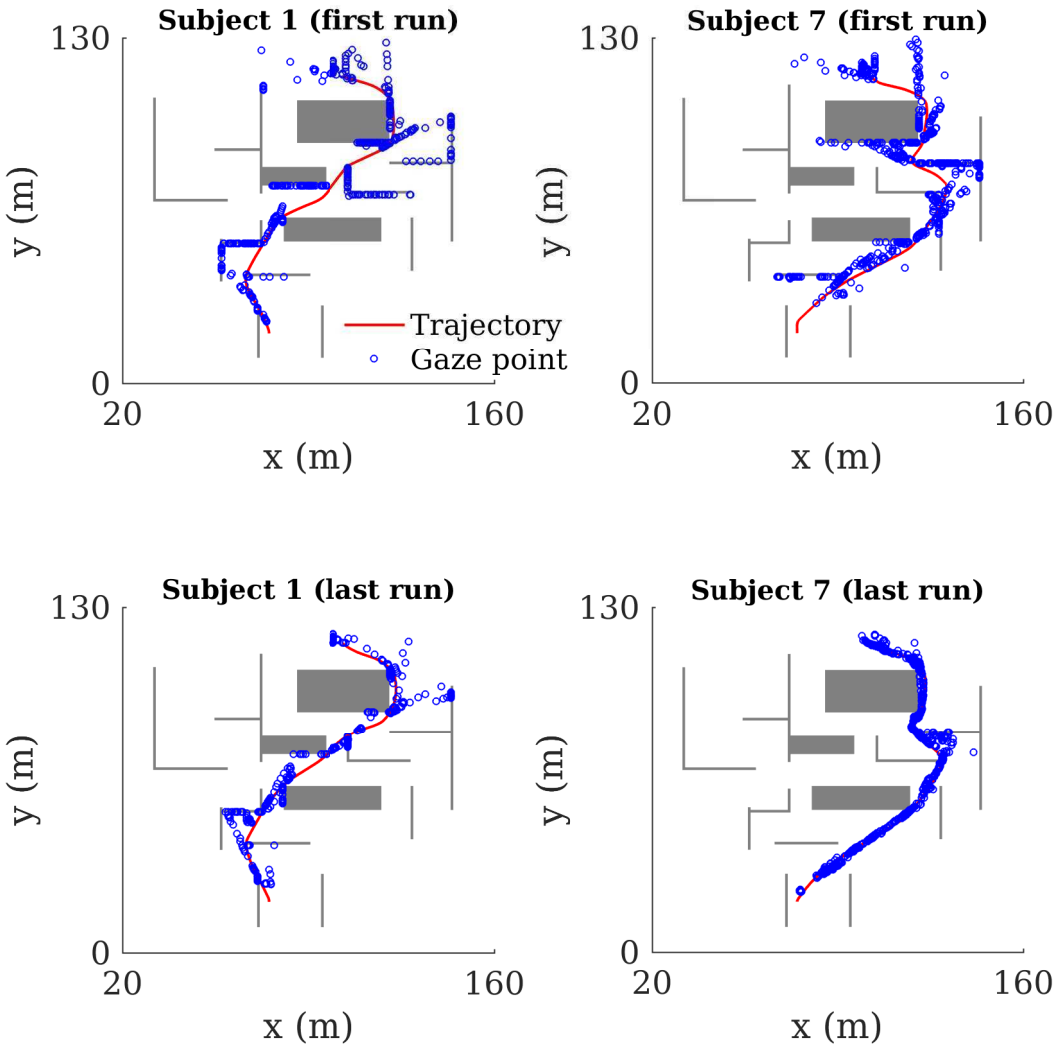


Figure 6.5: Gaze trajectories for first and last runs on best routes of subjects # 1 and # 7.

s is the time-constant for the vehicle command-to-speed model). For subject # 1, the frequency of high-speeds increases from 38.2 % to 54.0 % from starting to final runs. For subject # 7, the frequency increases from 27.7 % to 41.1 %. Figure 6.6(b) shows the mean minimum distance (r_{min}) from obstacle corners for starting and final runs for the both subjects. In final runs, mean r_{min} for subjects # 1 and # 7 are 0.2 m and 0.9 m, respectively. These results suggest that Subject # 7 shows higher obstacle avoidance behavior than subject # 1.

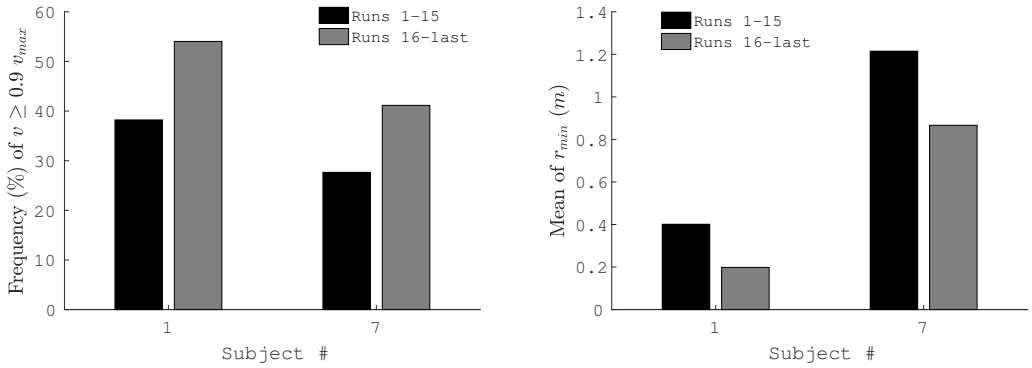


Figure 6.6: (a) Frequency of high-speeds near corners and (b) Mean r_{min} for starting (1-15) and final (16-last) runs for subjects # 1 and # 7.

6.2.1 Gaze

Figure 6.5 shows gaze trajectories for first and last runs on best routes of subjects # 1 and # 7. Figure 6.7 shows the frequency of gaze within 1 m of obstacle corners, i.e., subgoal heuristics, for the runs shown in Fig. 6.5. Visual attention in starting runs is scattered (e.g., regularly scanning sideways) for both subjects. In the last run, subject # 1 primarily (28.9 % of total time) focuses gaze near obstacle corners. Subject # 7 attends to obstacle corners with almost half the frequency (13.8 % in the last run) of subject # 1, and he/she focuses gaze at future points on the path. An explanation for such gaze behavior of subject # 7 is that the subject is occupied with stabilizing the vehicle on a reference path due to his/her novice control skills, which is showed later in the analysis of guidance primitives.

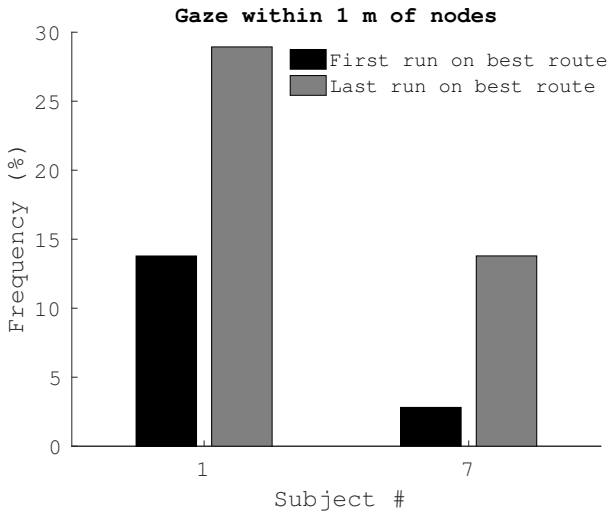


Figure 6.7: Frequency of gaze within 1 m of corners in first and last runs on best routes of subjects # 1 and # 7.

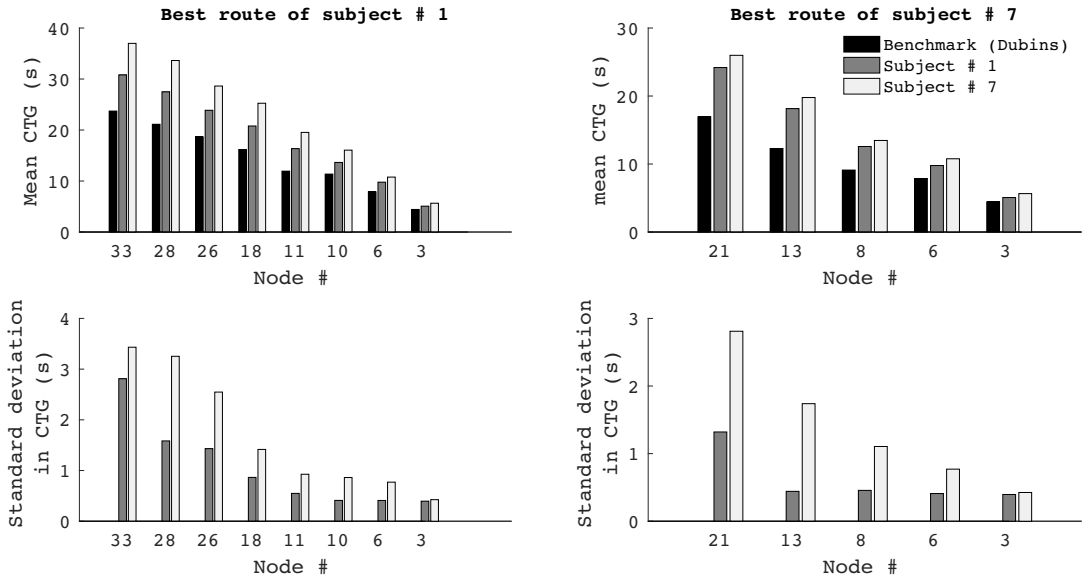


Figure 6.8: Benchmark, mean, and standard deviation of CTG for subjects # 1 and # 7 at nodes on their best routes.

6.2.2 CTG at Subgoals

Figure 6.8 shows the benchmark CTG and mean and standard deviation of CTG for subjects # 1 and # 7 at nodes on their best routes. The average gap between the benchmark CTG and subject # 1's mean CTG is 26.5 %. For subject # 7, the gap is 49.3 %. Mean standard deviation in CTG's at nodes for subjects # 1 and # 7 are 5.4 and 7.3 %, respectively. Subject # 1 shows better convergance in CTG at subgoals (nodes) than subject # 7.

6.3 Guidance Primitives (Quantitative Analysis)

Figures 6.9 and 6.10 show trajectory segments in the corner frame for runs 1-15 and 16-last for subjects # 1 and # 7, respectively. Time-window T is 2τ where $\tau = 1.13\ s$ is the time-constant for the vehicle command-to-speed model. The trajectories are divided into five clusters ($\pi_i, i \in [1\ 5]$) using hierarchical clustering (Eqs. 5.25 and 5.26). In runs 1-15, clusters are numbered in decreasing order of frequencies. In runs 16-last, clusters are numbered according to their similarity with the clusters in runs 1-15. The similarity between two clusters is measured as the average distance between all pairs of trajectory segments in the clusters (Eq. 5.26). Figure 6.11 shows the trajectory segments in the global environment for both the subjects.

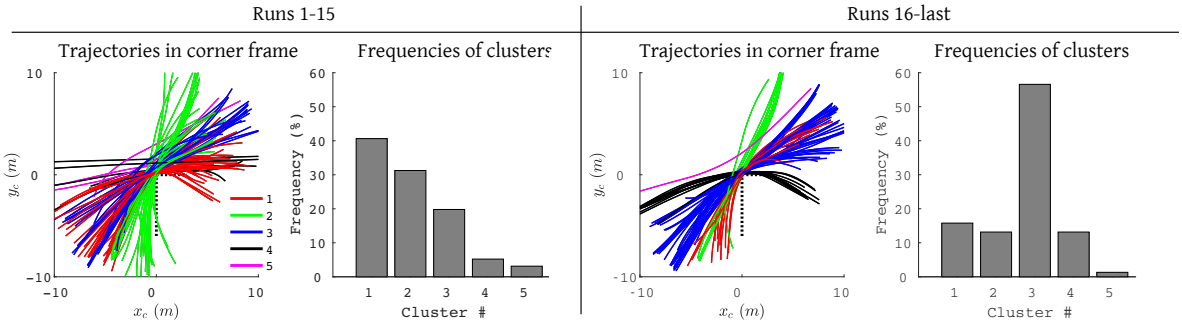


Figure 6.9: Subject # 1: trajectories in corner frame and clusters' frequencies for runs 1-15 and 16-last.

The frequencies of clusters in runs 1-15 and 16-last for subjects # 1 and # 7 are shown in Figs. 6.9 and 6.10, respectively. For subject # 1, clusters are not distinct in

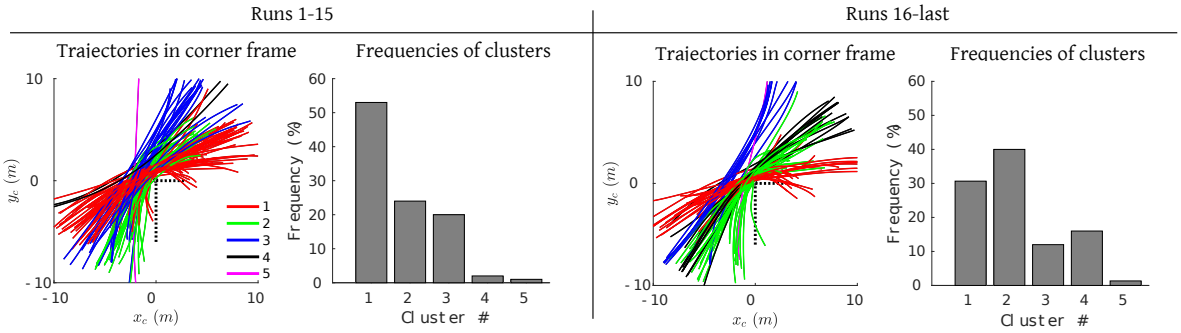


Figure 6.10: Subject # 7: trajectories in corner frame and clusters' frequencies for runs 1-15 and 16-last.

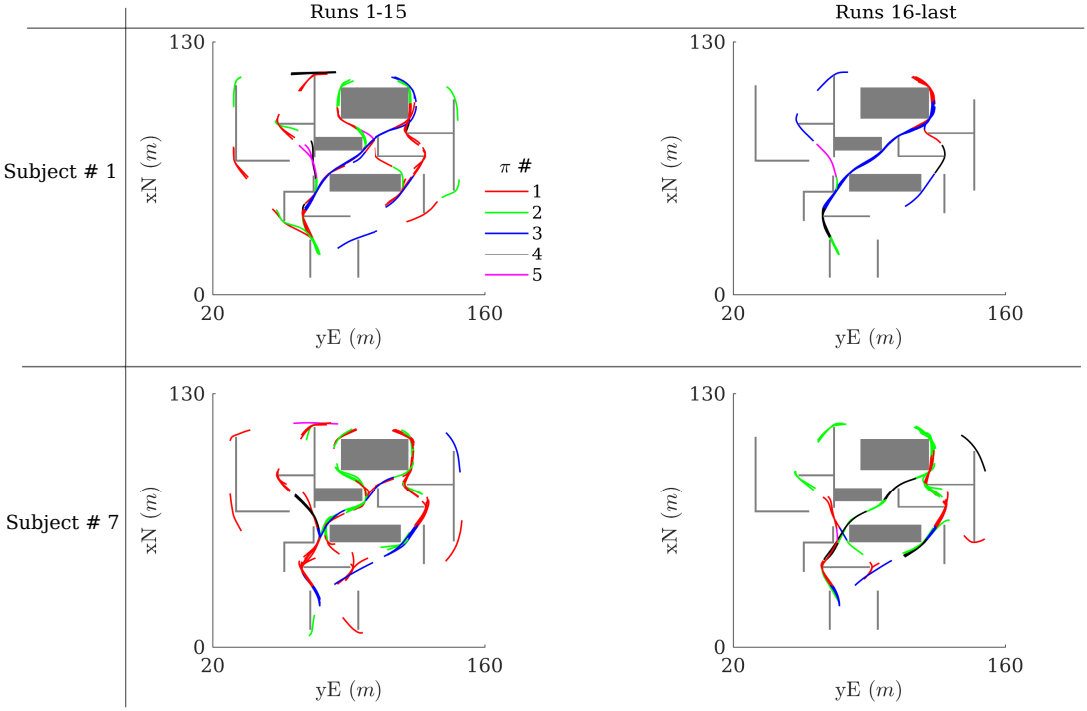


Figure 6.11: Subjects # 1 and 7: trajectory clusters 1-5 in global environment for runs 1-15 and 16-last.

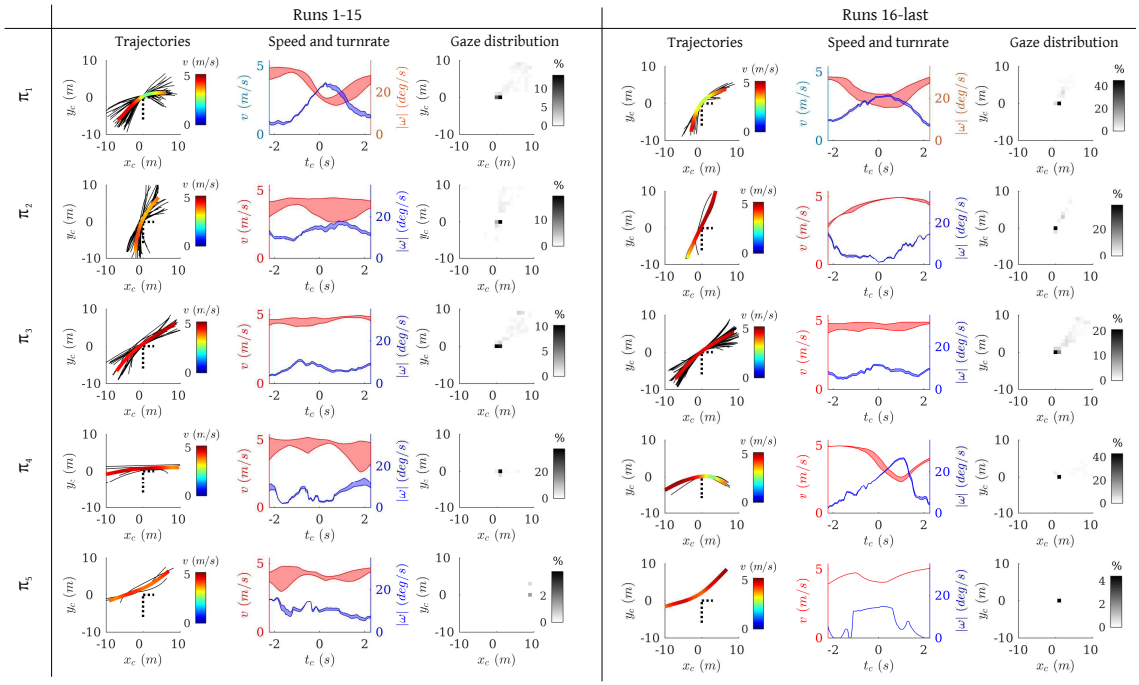


Figure 6.12: Subject # 1: trajectories, speed, turnrate, and gaze distribution for clusters 1 to 5 for runs 1-15 and 16-last.

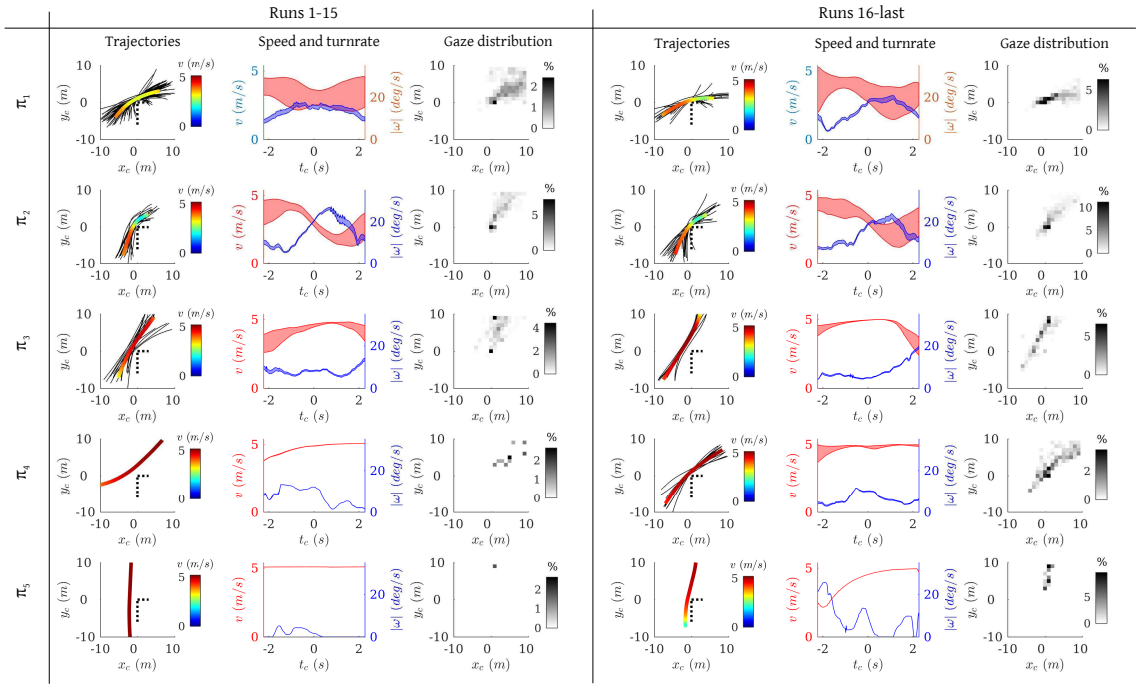


Figure 6.13: Subject # 7: trajectories, speed, turnrate, and gaze distribution for clusters 1 to 5 for runs 1-15 and 16-last.

runs 1-15. The behavior follows more distinct clusters in runs 16-last (see trajectories in Fig. 6.9). In runs 1-15, there are three dominant clusters with frequencies of 40.6, 31.3, and 19.8 %, respectively. In runs 16-last, there is one dominant mode with the frequency of 56.6 %. For subject # 7, trajectories in runs 16-last are spread across clusters. Subject # 1 has a guidance primitive library (Π) with better differentiated behaviors than subject # 7.

Figures 6.12 and 6.13 show trajectories, mean trajectory (colored based on mean speed value), time-histories of mean speed and turnrate, and gaze distribution for the clusters for runs 1-15 and 16-last for subjects # 1 and # 7, respectively. For a cluster, overall mean speed V and uncertainty in speed profile U_v are computed as follows:

$$\begin{aligned} V &= \frac{\int_{-T}^{T} w v_m dt_c}{\int_{-T}^{T} w dt_c}, \\ U_v &= \frac{\int_{-T}^{T} w \sigma_v dt_c}{\int_{-T}^{T} w dt_c}, \\ w &= 1 - \frac{|t_c - T|}{2T}, \end{aligned} \tag{6.1}$$

where v_m and σ_v are mean and standard deviation in speed, respectively. V and U_v for subjects # 1 and # 7 for the clusters (guidance primitives: $\pi_i, i \in [1\ 5]$) for runs 1-15 and 16-last are shown in table 6.1. The table also shows the V and U_v for the guidance primitive library Π, which are weighted sum of V and U_v for clusters π_i's based on their frequencies, in runs 1-15 and 16-last for the both subjects. The mean speed for subject 1 in runs 16-last is 4.3 m/s with the standard deviation of 0.2 m/s, which are 3.7 m/s and 0.5 m/s, respectively, for subject # 7.

Subject # 1 shows consistent (repeatable) control behavior unlike subject # 7. This observation supports that subject # 1 has consolidated the behavior in his/her memory. Also, the behavior consolidated in subject # 1's memory is effective and safe, which are supported by high speeds used by the subject (Figs. 6.6(a) and Fig. 6.12) and close distances to corners (Fig. 6.6(b)), respectively.

Gaze distribution in Figs. 6.12 and 6.13 are computed using gaze data from $t_c = -T$ to $t_c = 0$ because corner is not visible beyond $t_c = 0$. In runs 1-15, subject # 1 focuses gaze near corners with the frequency of 10-20 %. In runs 16-last, the frequency increases to 20-40 %, which is almost four times the frequency (5-10 %) of subject # 7. Subject

7 looks at future points on the path instead of focusing at corners, which is consistent with observations in Fig. 6.5.

Subject # 1 who achieves lower flight-time and better differentiated and converged guidance primitives than subject # 7, focuses gaze at corners. There are two possible reasons for subject # 1's focus at corners. One reason is based on bottom-up visual processing, i.e., corners are salient visual features. Another reason is top-down planning strategy where corners are heuristics for subgoals. Subject # 1's gaze focus at corners in π_1 is 40-45 % whereas it is almost the half (20 %) in π_3. The trajectories in π_1 involve higher turning of vehicle around the corner than the trajectories in π_3. This observation supports that the attention at corners is not only due to saliency but also because corners serve as subgoal heuristics.

Table 6.1: Overall mean (V) and uncertainty (U_v) of speed profile for clusters # 1 to # 5 (guidance primitives: $\pi_i, i \in [1\ 5]$) and all clusters together (guidance primitive library Π) for subjects # 1 and # 7 for runs 1-15 and 16-last.

Runs		π_1	π_2	π_3	π_4	π_5	Π
1-15	Sub. # 1: V (U_v) m/s	3.4(0.3)	3.8(0.6)	4.6(0.2)	4.6(0.3)	4.1(0.3)	3.8(0.4)
16-last	Sub. # 1: V (U_v) m/s	3.3(0.4)	4.6(0.0)	4.6(0.2)	3.8(0.2)	4.4(0.0)	4.3(0.2)
1-15	Sub. # 7: V (U_v) m/s	3.2(0.8)	3.0(0.3)	4.4(0.2)	4.8(0.0)	5.1(0.0)	3.4(0.5)
16-last	Sub. # 7: V (U_v) m/s	3.4(0.8)	3.2(0.6)	4.8(0.1)	4.8(0.1)	4.3(0.0)	3.7(0.5)

6.3.1 Specific Insights about Human Spatial Behavior

Proficient subjects demonstrate highly repeatable control behavior over vehicle dynamics and its interaction with the spatial environment. These subjects exhibit clearly formed interaction patterns. The interaction patterns allow subjects to focus their attention on the high-level elements of the task such as subgoals needed to elaborate plans and process relevant environment elements. In contrast, unskilled subjects are mostly focused on basic vehicle controls. Therefore they allocate most of their attention to the low-level functions such as stabilizing the vehicle along a path and avoiding collision.

The interaction patterns aid planning and ultimately learning, because the largely automated performance of guidance behavior enable filtering the information that is

relevant to the execution but is not relevant to the larger task specification, and extract information elements that are relevant to learning the task at hand. This suggest that the interaction patterns are assimilated in procedural memory similar to other sensory-motor patterns studied in human and animal motor control.

Chapter 7

Subgoal-Graph Framework for Human Environment Learning: Simulation Validation

This chapter uses simulations to validate the subgoal-graph representation and decision-making algorithm that are applied in the analysis of human learning of unknown environments in guidance tasks in Chapter 5. The chapter presents an autonomous guidance system for navigation in unknown environments based on the subgoal-graph representation and decision-making algorithm. The autonomous guidance system is simulated for the same guidance task (e.g., maze environment, minimum-time criteria) as used in the human experiments. The results from successive runs in the maze environment are analyzed for the emergence of guidance primitives (interaction patterns). Finally, the chapter presents a brief discussion of advantages of the subgoal-graph autonomous guidance system and relations of the presented guidance system to existing path planning methods.

7.1 Assumptions and Modeling

This section first presents the assumptions made in the autonomous guidance system. Second, the section presents a modeling of the onboard sensor and system's memory

for the task. Third, the section briefly revises 1) the modeling of environment cues for subgoal candidates, which is presented earlier in Section 5.2.5 in Chapter 5 and 2) the concept of subgoal graph. Finally, it presents the vehicle dynamic model used in simulations.

7.1.1 Assumptions

The task environment is two-dimensional and made of polygonal obstacles. The task is to find the fastest (minimum-time) route between pre-specified start and goal states over successive trials. The environment (e.g., obstacle-field) is unknown before the trials start. The system uses a laser scanner on-board vehicle, which gives a depth map or a visible boundary in a specified range, as shown in Fig. 7.1. R_{fov} and θ_{fov} are radius and angle, respectively, of field of view of the depth sensor. In simulations presented in this chapter, sensor parameters R_{fov} and θ_{fov} are determined using gaze data from human experiments presented in chapters 5 and 6, which is discussed later in this chapter.

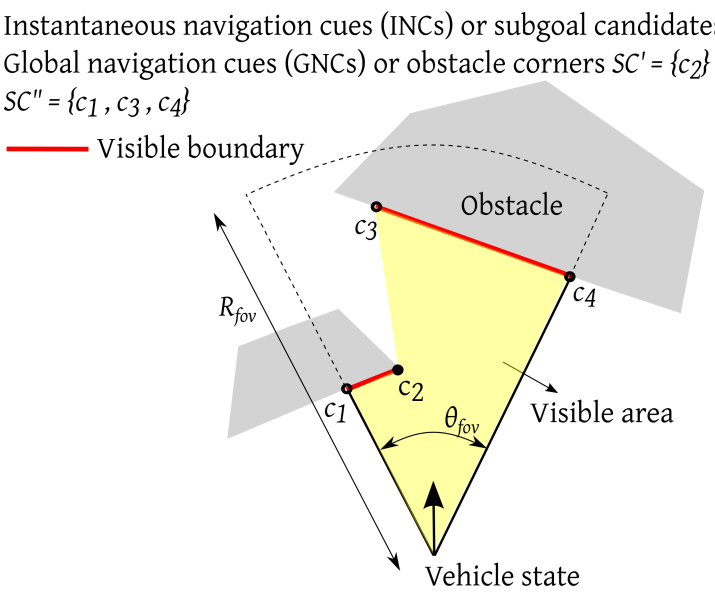


Figure 7.1: Example: instantaneous navigation cues (INCs) or subgoal candidates (SCs).

Human Gaze vs. Depth Sensor

The environment sensing with a depth sensor (e.g., laser scanner) is in general uniform in the sensor's field of view. In contrast, humans focus their gaze (or visual attention) at a point location and actively steer their gaze based on the specific task. A method to model the point attention is to use a weight function that increases the uncertainty in environment sensing at a location (e.g., probability of the location being occupied by obstacles) in proportion to the distance between the location and the gaze point [183]. In this dissertation, the sensing is assumed to be uniform in sensor's field of view.

Perfect Sensing and Path-Tracking

For simulations presented in this chapter, the depth sensor is assumed to be perfect. A perfect sensor provides accurate obstacle boundary in visible space (Fig. 7.1). The path-tracking or control implementation is assumed to have zero noise. The assumptions of perfect sensor and path tracking are made to remove confounding factors arising from sensing and tracking errors. The focus of simulations presented in this chapter is to validate environment learning framework based on the subgoal-graph memory structure presented in Chapter 5.

7.1.2 Task Memory

The guidance system's memory has two components: long-term memory and working memory. The long-term memory stores the goal state and subgoal-graph (subgoals, CTG at subgoals, connectivity among subgoals). The working memory represents the information used in current planning, which are: 1) connected subgoals, 2) visible subgoals, and 3) visible obstacle boundary and space. At any instant of time, the system recalls a subset of the subgoal-graph, which is connected to the preceding subgoal, retrieved from the long-term memory. The system also uses visible subgoal candidates or subgoals in planning (decision-making). The autonomous system remembers only the visible environment (obstacle boundary). Online trajectory planning plans a safe trajectory based on the visible space.

7.1.3 Environment Cues for Subgoal Candidates

For an optimal control (e.g., Dubins) solution, subgoals coincide with the obstacle corners [10]. Therefore, the corners are described as global navigation cues (GNCs). The instantaneous navigation cues (INCs) are the end points of the visible obstacle boundary, as shown in Fig. 7.1. An INC c_i is either an obstacle corner (GNC) or not. INCs are instantaneous subgoal candidates represented by set SC. For the example shown in Fig.7.1, SC is $\{c_1,\ c_2,\ c_3,\ c_4\}$. $SC' \subset SC$ are INCs that are corners. $SC'' \subset SC$ are INCs that are not corners. For the example shown in the figure, SC' and SC'' are $\{c_2\}$ and $\{c_1, c_3, c_4\}$, respectively.

If the vehicle travels towards an INC $c_i \in SC''$, the INC shifts as shown in Fig. 7.2. Eventually, the INC coverges to a GNC (obstacle corner). This is described as tracking an INC to a GNC.

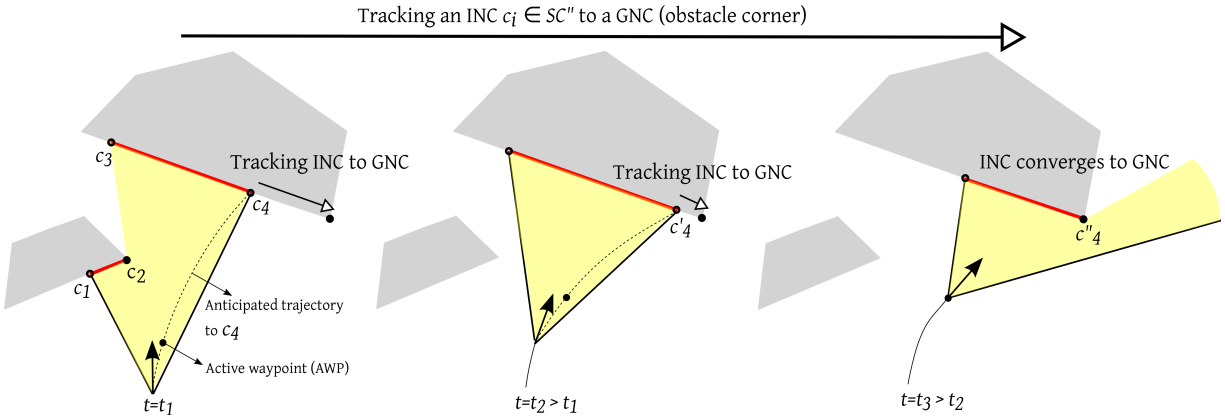


Figure 7.2: Example: tracking an INC $c_i \in SC''$ to a GNC (obstacle corner).

7.1.4 Subgoal Graph

In the proposed autonomous guidance system, knowledge of the task environment is represented by a graph of subgoals G. A subgoal g_k coincides with a corner. For avoiding collisions, a subgoal is placed at a safe distance d_{safe} from the corner along the bisector of two walls meeting at the corner, as shown in Fig. 7.3. Terms subgoal and node are used interchangeably.

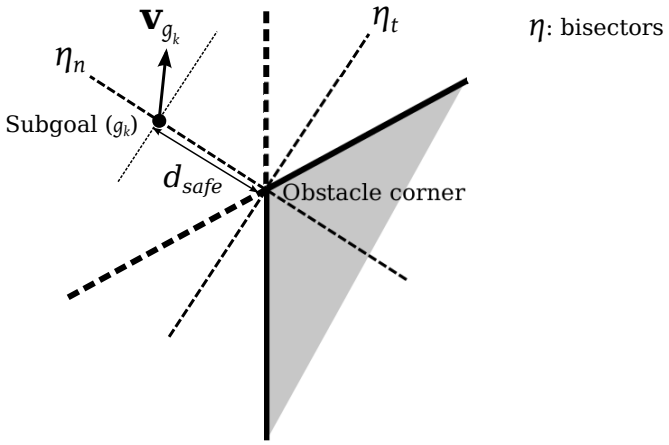


Figure 7.3: Subgoal at safe distance d_{safe} from obstacle corner.

Prior to run 1, the system knows the goal state g_0 but does not know the environment layout (obstacles). After a trial is completed (goal is reached), the graph knowledge is updated using the same method applied in human data processing (see Section 5.3.2). A trajectory is presented as a sequence of subgoals. The connection matrix (Q), cost-to-go from a subgoal g_k to the goal ($CTG_{k_{list}}$), and incremental cost between two subgoals g_k and g_i ($DC_{ki_{list}}$) are updated at the end of each trial. At a subgoal g_k, known (KN), unknown (UKN) and connected (CN_k) nodes are defined as in Section 5.3.2.

7.1.5 Vehicle Model

The vehicle used in simulations presented in this chapter is a point-mass with a discrete-time linear state-space model as follows:

$$\begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix}_k + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \begin{bmatrix} a_x \\ a_y \end{bmatrix}_k , \tag{7.1}$$

where Δt is the time-step. The velocity and acceleration constraints are as follows:

$$\begin{aligned} {v_x}^2 + {v_y}^2 &< {v_{max}}^2 \\ {a_x}^2 + {a_y}^2 &< {a_{max}}^2, \end{aligned} \tag{7.2}$$

where v_{max} and a_{max} are maximum speed and maximum acceleration, respectively. v_{max} and a_{max} for simulations presented in this chapter are determined based on the dynamic envelope of vehicle used in human experiments, which is discussed later in the chapter.

7.2 Autonomous Guidance System

This section presents the autonomous guidance system for navigation in unknown environments. The task is to learn minimum-time routes from specified start to goal locations over successive runs. The guidance system operates in two stages as shown in Fig. 7.4: exploration and consolidation. In the exploration stage, the system learns subgoal graph network (G and Q) of the task environment over successive runs. The guidance system explores subgoals that are visible, i.e., in sensor range and field of view. Thus, visibility of subgoals influence what portion of subgoal network is explored. In a related work [179], authors use visibility graph as heuristics at planning level to validate hierarchical framework for human guidance behavior presented in [8]. The exploration stage is terminated when no new subgoal sequence from start to goal is found. The termination condition is discussed later in this section.

After the exploration stage terminates, the system uses the learned subgoal graph, an optimal graph search method, and a heuristic cost function based on straight-line distances between subgoals to extract subgoal sequences (routes) that are candidates for time-optimal route. In the consolidation stage, the candidate subgoal sequences are arranged in an order of increasing cost. The system learns optimal cost-to-go on each candidate route (subgoal sequence). The system repeats the first route over successive runs and in each run it updates state (e.g. velocity) at subgoals based on cost-to-go data from preceding runs. When the cost-to-go is converged on the first route, the system switches to the second route and repeats the same process. If a pair of consecutive subgoals, i.e., an edge in subgoal graph, is common in two routes and the subgoal states for the pair are already learned on one route, the system exploits the learned information. The consolidation stage ends when optimal cost-to-go is learned on all candidate routes (subgoal sequences).

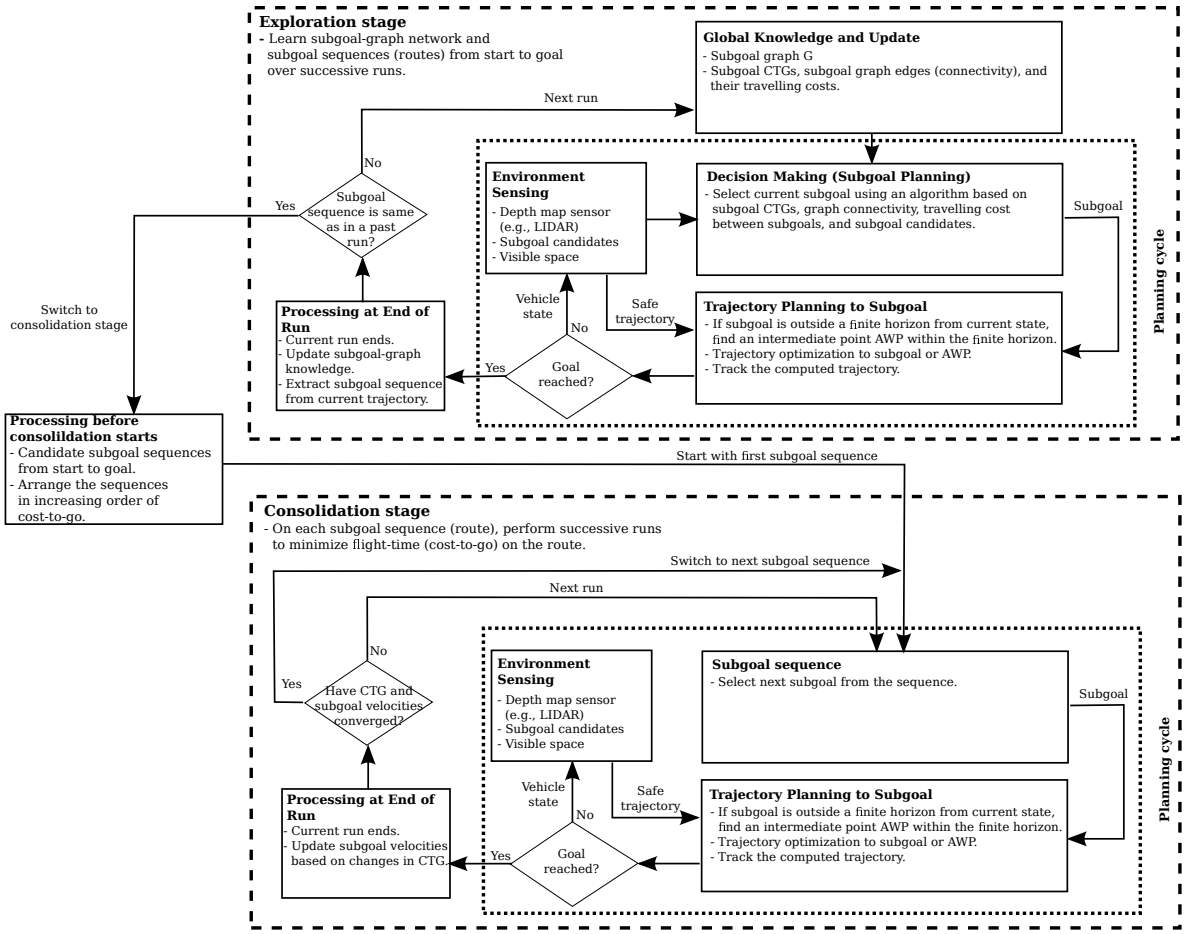


Figure 7.4: Subgoal-graph based autonomous guidance system: exploration and consolidation stages.

7.2.1 Exploration Stage

In the exploration stage, the guidance system consists of a real-time planning cycle as shown in Fig. 7.4. The cycle has three components: 1) environment-sensing, 2) decision-making (subgoal selection), and 3) trajectory planning. The central component is the decision-making rule that determines subgoal for trajectory planning. In trajectory planning, the system plans an active waypoint (AWP) in the visible area and plans a trajectory to the AWP. Each component is described in detail next.

Environment Sensing

The onboard depth sensors give visible obstacle boundary. The subgoal candidates SC (or INCs) are the endpoints of the visible boundary.

Decision-Making (Subgoal Planning/Switching)

The decision-making component plans subgoal for trajectory planning. For decision-making, there are two scenarios: a) the vehicle has reached at a subgoal g_k and has to decide the next subgoal g_{k+1}, b) the current subgoal is hindered by known (visible) obstacle boundary and a new subgoal has to be found.

In the scenario (a), the system checks if there is a connected subgoal from g_k, i.e., $|CN_k| > 0$. In case of $CN_k = \{\}$, if the straight line joining the current vehicle position and the goal does not intersect the visible obstacle boundry, the goal is the next subgoal. If the goal is hindered by the visible obstacle boundary, the system chooses next subgoal from SC. If $|SC'| > 0$, the system selects a c_i from SC'. For a $c_i \in SC'$, state $\mathbf{c_i}$ is position c_i and velocity at c_i. The a priori speed at c_i is assumed to be zero or minimum speed allowed because the next subgoal is unknown from c_i. The a priori velocity direction is parallel to the bisector η_t (see Fig. 7.3). The system chooses c_i that minimizes the total cost as follows:

$$\min_{c_i \in SC'} {CTC_{\mathbf{c_i}}}^p + {CTG_{\mathbf{c_i}}}^p, \tag{7.3}$$

where ${CTC_{\mathbf{c_i}}}^p$ and ${CTG_{\mathbf{c_i}}}^p$ are a priori estimates of cost-to-come to $\mathbf{c_i}$ from the current state and the cost-to-go from the $\mathbf{c_i}$ to the goal state. The values are based on straight-line distance and maximum speed. If $SC' = \{\}$, the system selects a c_i from SC'' using

the optimization as follows:

$$\min_{c_i \in SC''} {CTC_{\mathbf{c_i}}}^p + {CTG_{\mathbf{c_i}}}^p. \tag{7.4}$$

A $c_i \in SC''$ is a temporary subgoal because the cue (subgoal) position shifts on the boundary. The cue position shifts because the visible portion of edge expands as vehicle approaches towards the cue as shown in Fig. 7.2. Eventually, the cue converges to a corner (GNC) that is a stable subgoal.

If case of $|CN_k| > 0$ in the scenario (a), if there is a visible corner (subgoal candidate) that is not explored from the current subgoal, i.e., $|SC'/CN_k| > 0$, the system selects a $c_i \in SC'/CN_k$ that minimizes a priori estimate of cost-to-go to the goal, which is the following:

$$\min_{c_i \in SC'/CN_k} {CTC_{\mathbf{c_i}}}^p + {CTG_{\mathbf{c_i}}}^p, \tag{7.5}$$

If the node c_i is explored in past runs, the term ${CTG_{\mathbf{c_i}}}^p$ is replaced by $f\left(CTG_{i_{list}}\right)$. The a priori cost values ${CTC_{\mathbf{c_i}}}^p$ and ${CTG_{\mathbf{c_i}}}^p$ are based on straight line distance and are lower bounds for the optimal cost. As $|CN_k| > 0$, the current node g_k has been explored in past runs and $CTG_{k_{list}}$ that represent cost-to-go to the goal from g_k is not empty. Therefore, the algorithm rejects nodes c_i's in Eq. 7.5 that gives total cost more than the minimum cost-to-go achieved from the current node g_k, which is written as the following constraint for Eq. 7.5:

$${CTC_{\mathbf{c_i}}}^p + {CTG_{\mathbf{c_i}}}^p < \min\left(CTG_{k_{list}}\right). \tag{7.6}$$

Condition 7.6 discards to explore solutions that can not give a lower cost than the minimum cost achieved so far. Thus the condition limits graph search, which is similar to branch and bound [152].

The further process is if no $c_i \in SC' \setminus CN_k$ satisfies the constraint 7.6 and therefore no subgoal is found from Eq. 7.5. In this case, the system checks if there are INCs, i.e., $|SC''| > 0$. With learning, the system memorizes only GNCs (corners/subgoals) and not the INCs. The INCs can be any point on obstacle boundaries and storing them therefore will require a larger memory than required for GNCs. If an INC is selected as subgoal, it eventually converges to a GNC. In later phase of exploration, INCs can be ignored as they may converge to GNCs that are already explored in past runs. The

system remembers how many times INCs have been selected as subgoals from a node g_k, which is represented by n_k^{INC}. N_k represents the maximum number of times the system is allowed to select an INC as subgoal from a node g_k.

If the condition $n_k^{INC} < N_k$ is not satisfied, the system selects next subgoal from CN_k, which is as follows:

$$\min_{g_i \in CN_k} f\left(DC_{ki_{list}}\right) + f\left(CTG_{i_{list}}\right), \tag{7.7}$$

where function f can be maximum, minimum, median, mean, mean of last few (e.g., three) trials, etc. For example, maximum and minimum functions represent conservative and greedy approaches, respectively. Median function models that the system relies on the cost value with highest probability. Mean function models the overall average cost. Assuming that the system has a limit on memory and it can store experiences from only a certain number of past trials, f can be mean of last few trials. In this dissertation, f is the minimum function.

Note that Eq. 7.7 represents exploitation as the system selects a subgoal that has been explored from the current subgoal in previous trials, i.e., $g_k \in CN_k$. In starting runs, the system will select solutions using equations 7.3, 7.4, or 7.5. Eventually, after the subgoal-graph is learned, the decision-making algorithm does not find a new subgoal that can give a better solution than the best one learned so far. Therefore, the algorithm starts exploiting (Eq. 7.7).

In the scenario (b) when current subgoal g_k is hindered by visible obstacle boundary, a new subgoal has to be found. The new subgoal is selected from SC. The process is the same as described in the second paragraph (Eqs. 7.3 and 7.4). If $|SC'| > 0$, the new subgoal is selected from SC' using Eq. 7.3. If a node (corner) $c_i \in SC'$ is tried in previous runs, the $CTG_{\mathbf{c_i}}{}^p$ in Eq. 7.3 is the CTG at the subgoal $g_j \in G$ that corresponds to the corner c_i. If $SC' = \{\}$, the new subgoal is selected from SC'' using Eq. 7.4.

Trajectory Planning

After the decision-making process computes a subgoal, the vehicle has to plan a trajectory to the subgoal from its current state. There are three steps involved in the trajectory planning: 1) compute time-optimal active waypoint (AWP) [23] within a finite horizon from the vehicle current state using a heuristic cost map that is computed

offline and stored in long-term memory, 2) apply a numerical optimization method to compute time-optimal trajectory to the AWP using state-space model (7.1) of vehicle dynamics, and 3) execute first N_s steps of the trajectory. AWP is an intermediate point to reach the subgoal, and is placed within a finite horizon and visible space as shown in Fig. 7.5. The execution of first N_s steps is a receding horizon planner [9]. If N_s is small, the frequency of trajectory update (recomputation) will be high.

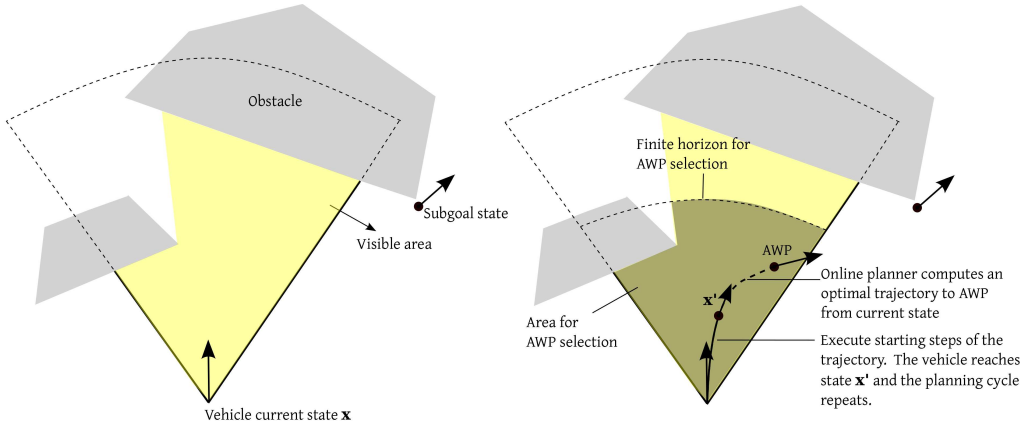


Figure 7.5: Illustration: AWP in a finite horizon from vehicle current state.

The trajectory optimization from the current state $\mathbf{x}$ to the AWP is a two point boundary value problem [19, 23] as presented in Chapter 3. The vehicle used in simulations presented in this chapter has nonlinear constraints (Eq. 7.2). For a vehicle with nonlinear constraints, there are in general no analytical solutions to two point boundary value problems. An approach is to use numerical methods as the simulations presented in Chapter 3 uses the CPLEX [175] for online trajectory optimization. For the vehicle model used in this chapter, the input and state vectors are $[a_x \ a_y]$ and $[x \ y \ v_x \ v_y]$, respectively. The discrete-time state-space model is given by Eq. 7.1.

The AWP minimizes the total cost-to-go to the subgoal from the current state, which is the sum of cost-to-come (CTC) to the AWP from the current state and cost-to-go (CTG) to the subgoal g_k from the AWP. The optimization for AWP selection is formulated as follows:

$$\min_{\mathbf{x}_p, \mathbf{v}} CTC(\mathbf{x}, \mathbf{x}_p, \mathbf{v}) + CTG(\mathbf{x}_p, \mathbf{v}, g_k), \tag{7.8}$$

where $\mathbf{x}_p$ is a spatial position in the finite horizon and visible space (Fig. 7.5), and $\mathbf{v}$

is the velocity at $\mathbf{x}_p$. The $CTG(\mathbf{x}_p, \mathbf{v}, g_k)$, i.e., the cost-to-go to the subgoal g_k from the AWP state $[\mathbf{x}_p\ \mathbf{v}]$, is approximated based on the straight line distance between AWP and subgoal, speeds at AWP and subgoal, and maximum acceleration a_{max}. The $CTC(\mathbf{x}, \mathbf{x}_p, \mathbf{v})$, i.e., the cost-to-come to the AWP state from the current state $\mathbf{x}$, is uploaded from offline computed maps in vehicle body-frame. The CTC maps are computed using the CPLEX for all combinations of discretized $\mathbf{x_p}$ and $\mathbf{v}$ in the finite horizon in vehicle body-frame, for all vehicle speeds in the discretized space. In online planning, an appropriate CTC map based on vehicle current speed is uploaded, and translated and rotated based on vehicle current location and heading angle (velocity direction), respectively. For safety, d_{AWP} that is the distance between AWP and closest obstacle along the velocity direction at AWP has to satisfy the following condition:

$$d_{AWP} \geq |\mathbf{v}|^2\ 2a_{max} + d_{safe}. \tag{7.9}$$

Eq. 7.9 is used as a constraint to reject AWP candidates in Eq. 7.8.

Termination Condition for Exploration Stage

In an environment with finite number of obstacles, there are a finite number of distinct (not homotopic) routes between any two points. To achieve completeness [184], a brute and direct approach is to learn all subgoal sequences from start to goal. In this dissertation, the task is to find minimum-time routes. Therefore, the guidance system uses a graph pruning approach as shown by Eq. 7.6. The equation discards to explore subgoal connections (graph edges) that can not give a lower cost than the best cost achieved in previous runs. This pruning approach is similar to branch and bound [152].

Random sampling such as RRT [33] is a popular approach for exploring solutions in a probem space. In the guidance system presented in this chapter, no random factor is used to decide exploration vs. exploitation in selection of subgoals.

At the beginning of the exploration stage, the subgoal graph includes only the goal state as the obstacle field is unknown. At the end of each run in the exploration stage, the trajectory to the goal and corresponding subgoal sequence are used to update the graph knowledge such as subgoal connectivity Q'. Eventually, the subgoal selection (decision-making) algorithm does not find any new subgoal and decides to exploit known/learned subgoals as shown by Eq. 7.7. If the travelled subgoal sequence in a run is a repeated

one, i.e., the sequence is identical to the one in a previous run, the exploration stage is terminated.

7.2.2 Consolidation Stage

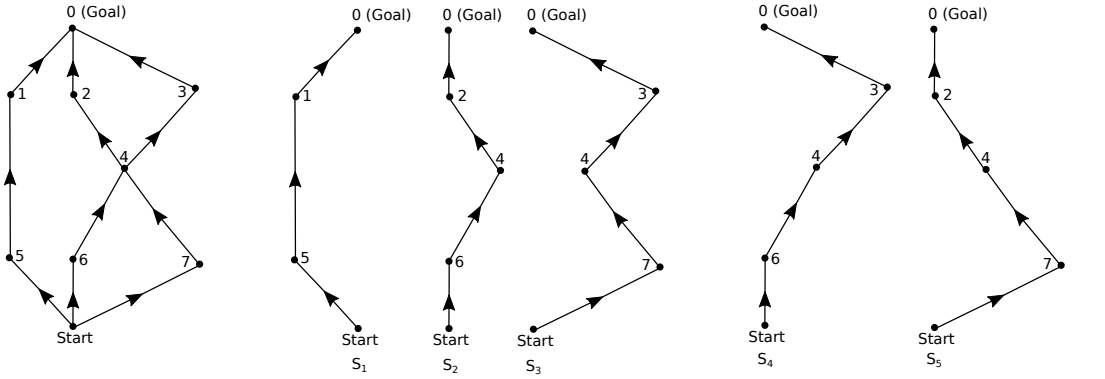
In the exploration stage, the system learns the subgoal graph and therefore, implicitly learns the different routes, i.e., subgoal sequences, from the specified start to the goal. However, during exploration, flight-times of routes are not minimized because of the following two reasons: 1) speeds at subgoals are assigned to be zero or minimum and 2) trajectory segments between consecutive subgoals are not always direct. The system assigns zero or minimum speed at subgoals as the environment beyond subgoals is in general unknown in starting runs. This is a conservative strategy but is consistent with the observed behavior from the speed data of the starting runs in human experiments (Chapters 5 and 6). In the exploration stage, when INCs ($c_i \in SC''$) are selected as subgoals, the system successively updates subgoals as INCs move on obstacle boundaries and eventually coincide with GNCs (see Fig. 7.2). This subgoal-shifting behavior makes trajectories longer, which result in higher flight-times.

In the consolidation stage, subgoal positions are known and subgoal sequence executed in a run is specified before the run starts. This resolves the issue of indirect trajectories between consecutive subgoals. The first issue, i.e., zero or minimum speeds at subgoals, is resolved by finding cost(time)-optimal velocities at subgoals. The subgoal velocities are iterated and updated based on changes in cost-to-go over successive runs.

Candidate Subgoal Sequences for Consolidation Stage

Figure 7.6(a) shows an example of subgoal graph learned in the exploration stage. Figure 7.6(b) shows the subgoal sequences (routes) S_1, S_2, and S_3 taken in the exploration stage. One can extract all subgoal sequences from start to goal from the learned subgoal graph. There may be subgoal sequences that are never taken in the exploration state. For example, Fig. 7.6(c) shows the sequences (routes) S_4 and S_5 that are not taken in the exploration stage. Extracting all subgoal sequences from start to goal is a combinatorial problem and is exponentially complex in time. In this dissertation,

candidate subgoal sequences for the consolidation stage are extracted using a method of perturbation along an optimal subgoal sequence, which is described next.



(a) Learned subgoal graph at the end of exploration stage

(b) Subgoal sequences (routes) taken in the exploration stage

(c) Subgoal sequences (routes) that are not taken in the exploration stage

Figure 7.6: Example: subgoal graph and subgoal sequences learned in the exploration stage of the subgoal-graph based autonomous guidance system in unknown environments.

Path length of a subgoal sequence is the total straight-line distance between subgoals, and is represented by L_s. The straight-line distance between subgoals is a heuristic for time-to-go. The system applies Dijkstra's algorithm on the learned subgoal graph to compute an optimal, i.e., shortest path, subgoal sequence from start to goal. ${L_s}^*$ is the path length of optimal subgoal sequence. Next, the system finds subgoal sequences that have path length within a threshold of the path length of the optimal subgoal sequence, e.g., $L_s \leq 1.1{L_s}^*$. This dissertation uses a method of perturbation along the optimal subgoal sequence. N_g is the number of subgoals between start and goal in the optimal subgoal sequence S = Start $\rightarrow$ $g_1 \rightarrow$ ... $g_{k-1} \rightarrow g_k \rightarrow g_{k+1}$ ... $\rightarrow g_{N_g} \rightarrow g_0$. ${S_k}^l = g_k \rightarrow g_{k+1} \rightarrow$ ... $\rightarrow g_{k+l-1}$ is a subsequence from subgoal g_k ($k \in 1, ..., N_g$) and of length l ($l \leq N_g - k + 1$). The system iteratively increases k from 1 to N_g and l from 1 to $N_g - k + 1$. For each ${S_k}^l$, the system searches an alternate subsequence between g_{k-1} and g_{k+l} using Dijkstra's algorithm and subgoal graph knowledge. The alternate subsequence is rejected if the total path length from start to goal gets more than $1.1{L_s}^*$.

Invalid Subgoal Sequences for Consolidation Stage

Figure 7.7 shows an example of invalid subgoal sequence for the consolidation stage. A subgoal sequence S = Start $\rightarrow$ ... $g_j \rightarrow g_k \rightarrow g_i$ ... g_0 is invalid if the angle α between any pair of consecutive edges $g_j \rightarrow g_k$ and $g_k \rightarrow g_i$ is smaller than π. Otherwise, the sequence is valid. The angle α is on the side that does not have obstacle corner associated with the subgoal g_k, as shown in Fig. 7.7. The invalid subgoal sequences from the exploration stage are converted to valid subgoal sequences using the learned subgoal graph connectivity Q'. It is checked using Q' if a valid subsequence $g_j \rightarrow .... \rightarrow g_i$ exists. For example in Fig.7.7(c), there exists $g_j \rightarrow g_{k'} \rightarrow g_i$ that is valid and replaces the invalid subsequence $g_j \rightarrow g_k \rightarrow g_i$ for the consolidation stage. If no valid subsequence is found, the subgoal sequence is rejected for processing in the consolidation stage.

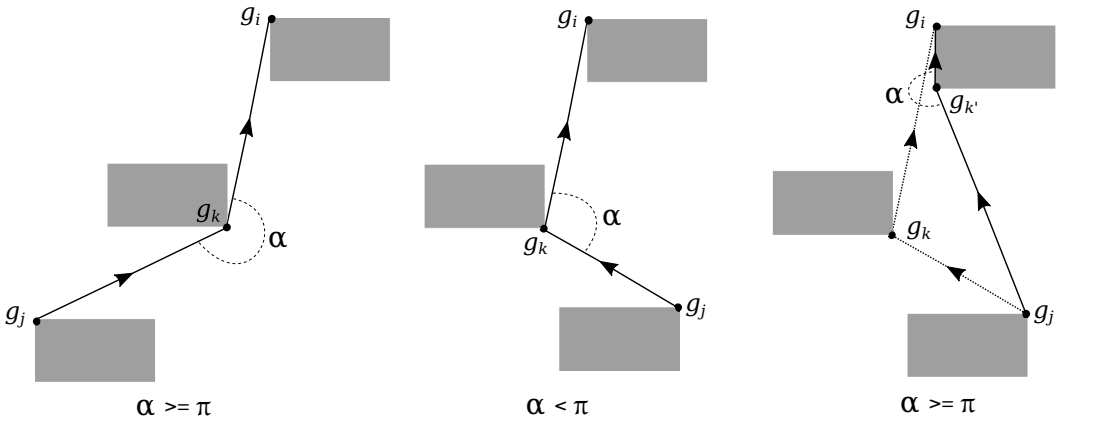


Figure 7.7: Illustration: invalid subgoal sequence.

Order Subgoal Sequences

The candidate subgoal sequences are arranged based on the number of common edges (pairs of consecutive subgoals) and path lengths. The subgoal sequences are numbered as S_1 to S_{N_s}, where N_s is the number of sequences. S_1 is the shortest path length subgoal sequence. S_2 is the subgoal sequence that has maximum number of common edges with S_1. If there are two or more subgoal sequences that have same number of

common edges with S_1, the subgoal sequence with smallest path length is chosen as S_2. Then, S_3 is selected based on maximum number of common edges with S_2 and smallest path length. This process goes on until all candidate subgoal sequences are numbered.

The cost(time)-to-go based on straight-line distance is a lower bound for true cost(time)-to-go. A subgoal sequence $S_I (I \in [1\ N_s])$ is iterated until the cost-to-go on the route converges, which is described in the following section. Then the system switches to the next subgoal sequence S_{I+1} as shown in Fig. 7.4.

Optimization of Cost-to-go of a Subgoal Sequence

For a subgoal sequence S_I = Start $\rightarrow$ ... g_j $\rightarrow$ g_k $\rightarrow g_i$ ... g_0, speeds at subgoals are set as zero or minimum speed in the exploration stage, which results in higher cost(time)-to-go. The velocity direction at subgoals are parallel to the bisector η_t (see Fig. 7.3). $\mathbf{v}_{g_k}$ represents the velocity at a subgoal g_k. For a planar motion, $\mathbf{v}_{g_k}$ is $[v_{g_k}\ \psi_{g_k}]$, where v_{g_k} and ψ_{g_k} are subgoal speed and velocity angle, respectively.

Velocities at the start and goal are given. Velocities at subgoals between the start and goal are variables and have to be optimized as follows:

$$\min_{\cdots\ ,\ \mathbf{v}_{g_j},\ \mathbf{v}_{g_k},\ \mathbf{v}_{g_i},\ \cdots} CTG_{S_I}, \tag{7.10}$$

where CTG_{S_I} is the cost-to-go of a trajectory through the subgoal sequence S_I. CTG_{S_I} is the total sum of costs of trajectory segments between consecutive subgoals as follows:

$$CTG_{S_I} = \sum\ DC_{ki}, \tag{7.11}$$

where DC_{ki} is the cost-to-go of trajectory segment going from g_k to g_i (Fig. 7.8). Cost DC_{ki} depends on subgoal velocities $\mathbf{v}_{g_k}$ and $\mathbf{v}_{g_i}$:

$$DC_{ki} = f(\mathbf{v}_{g_k}, \mathbf{v}_{g_i}). \tag{7.12}$$

Similarly, cost DC_{jk} depends on $\mathbf{v}_{g_j}$ and $\mathbf{v}_{g_k}$:

$$DC_{jk} = f(\mathbf{v}_{g_j}, \mathbf{v}_{g_k}). \tag{7.13}$$

Velocity at subgoal g_k, i.e., $\mathbf{v}_{g_k}$, affects DC_{jk} and DC_{ki} that are costs of consecutive segments $g_j \rightarrow g_k$ and $g_k \rightarrow g_i$, respectively. In simulations presented in this chapter,

subgoal velocity directions are assumed to be parallel to the bisector η_t (see Fig. 7.3) because the direction η_t is the safest. Therefore, the variable to be optimized is v_{g_k} that satisfies the following condition:

$$\frac{\partial\left(DC_{jk} + DC_{ki}\right)}{\partial v_{g_k}} = 0. \tag{7.14}$$

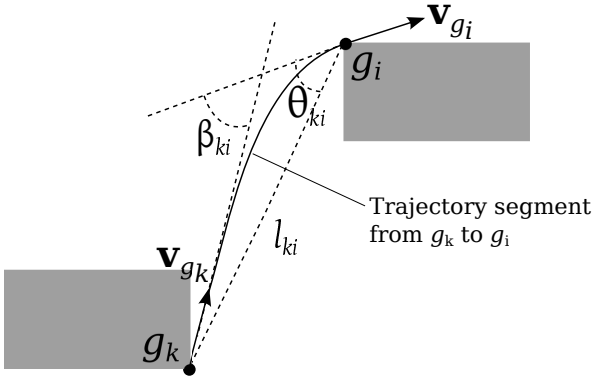


Figure 7.8: Subgoal velocities $\mathbf{v}_{g_k}$ and $\mathbf{v}_{g_i}$ and trajectory segment from subgoal g_k to g_i.

The consolidation stage executes successive runs on a subgoal sequence to find cost(time)-optimal speeds (Eq. 7.14) at subgoals. For a sequence S_I, the first run is simulated with subgoal speeds [... , $v^1_{g_j}$, $v^1_{g_k}$, $v^1_{g_i}$, ...] that are computed using a heuristic function described later in this section. The heuristic function is a lower bound for optimal speeds. Travel-time in general reduces for higher speeds. The speeds are increased stepwise over successive runs until flight-time stops decreasing and starts increasing.

The speeds [... , $v^1_{g_j}$, $v^1_{g_k}$, $v^1_{g_i}$, ...] are not necessarily optimal, and have to be optimized over successive runs. [... , C_{g_j}, C_{g_k}, C_{g_i}, ...] are binary variables that are initially zero. When the optimal value of v_{g_k} is found, C_{g_k} becomes one. The optimal velocity at a subgoal g_k depends on the child subgoal g_i. If subgoal edge $g_k \rightarrow g_i$ has occured in an already consolidated subgoal sequence S_1 to S_{I-1}, the subgoal velocity at g_k, i.e., $v^1_{g_k}$, is set the value learned in consolidation of previous subgoal sequences.

At the end of first run, costs of consecutive segments are extracted from the trajectory data and represented as [... , DC^1_{jk}, DC^1_{ki}, ...]. The second run is simulated with

increased values [... , $v^2_{g_j}$, $v^2_{g_k}$, $v^2_{g_i}$, ...] that are as follows:

$$v^2_{g_k} = \min(v^1_{g_k} + \Delta v,\ v_{max})\ \forall\ g_k \in \mathrm{S}_I \setminus \{\text{Start},\ \text{Goal}\}, \tag{7.15}$$

where Δv is the incremental change in speed. Costs from the second run are [... , DC^2_{jk}, DC^2_{ki}, ...]. Now the speeds [... , $v^3_{g_j}$, $v^3_{g_k}$, $v^3_{g_i}$, ...] for the third run are updated as follows:

$$v^3_{g_k} = \begin{cases} \min(v^2_{g_k} + \Delta v,\ v_{max}), & \text{if } C_{g_k} = 0 \text{ and } \left(DC^2_{jk} + DC^2_{ki}\right) < \left(DC^1_{jk} + DC^1_{ki}\right) \\ v^1_{g_k} \text{ (speed is converged)}, & \text{if } C_{g_k} = 0 \text{ and } \left(DC^2_{jk} + DC^2_{ki}\right) \geq \left(DC^1_{jk} + DC^1_{ki}\right) \\ v^2_{g_k} \text{ (speed is converged)}, & \text{if } C_{g_k} = 1 \end{cases} \tag{7.16}$$

Binary variables C_{g_k}'s are updated from zero to one if the corresponding speeds reach v_{max} or converge. Successive runs for S_I are simulated until speeds at subgoals ($g_k \in \mathrm{S}_I$) converge and C_{g_k}'s become one. Then, the consolidation stage switches to the next subgoal sequence S_{I+1}, and repeats the iteration process in Eq. 7.16. The number of runs ($N^{runs}_{\mathrm{S}_I}$) required for consolidating a subgoal sequence S_I is bounded as follows:

$$\begin{aligned} N^{runs}_{\mathrm{S}_I} &\leq \frac{v_{max} - \min(\dots, v^1_{g_j},\ v^1_{g_k},\ v^1_{g_i},\ \dots)}{\Delta v} \\ &\leq \frac{v_{max}}{\Delta v}. \end{aligned} \tag{7.17}$$

The prior speeds [... , $v^1_{g_j}$, $v^1_{g_k}$, $v^1_{g_i}$, ...] used in the first run for iterative consolidation of a subgoal sequence are computed as follows. Figure 7.8 shows a trajectory segment from g_k to g_i. The objective is to find $\mathbf{v}_{g_k}$ and $\mathbf{v}_{g_i}$ such that the cost DC_{ki} is minimized:

$$\min_{\mathbf{v}_{g_k},\ \mathbf{v}_{g_i}} DC_{ki}. \tag{7.18}$$

As the subgoal velocity directions are fixed to be parallel to the bisector η_t, Eq. 7.18 reduces to as follows:

$$\min_{v_{g_k},\ v_{g_i}} DC_{ki}. \tag{7.19}$$

If the speed v_{g_i} is known, Eq. 7.19 reduces to as follows:

$$\min_{v_{g_k}} DC_{ki}. \tag{7.20}$$

This chapter uses a heuristic solution of Eq. 7.20, which is as follows:

$$\begin{aligned} v_{g_k} &= k_a k_b k_c, \\ k_a &= \min(v_{max}, \sqrt{2a_{max}\max(l_{ki} - d_{safe}, 0)}), \\ k_b &= \min(\max(1 - \frac{|\beta_{ki}|}{\pi/2}, 0), 1), \\ k_c &= \min(\max(1 - \frac{|\theta_{ki}|}{\pi/2}, 0), 1). \end{aligned} \tag{7.21}$$

The velocity at goal (g_0) is given. Speeds [... , $v^1_{g_j}$, $v^1_{g_k}$, $v^1_{g_i}$, ...] are computed using Eq. 7.21 and going backwards from the goal.

7.3 Simulations

The subgoal-graph guidance system presented in the previous section is simulated in the maze environment used for human guidance experiments presented in Chapters 5 and 6. This section presents the simulation results and investigates the emergence of guidance primitives as a result of environment learning over successive runs. The section compares the guidance primitives from autonomous simulations with the guidance primitives extracted from human data presented in Chapter 6.

7.3.1 Task Definition and Parameters

The task environment is as shown in Fig. 5.7. The start and goal states are the same as in the human guidance experiments. In autonomous experiments, the goal location is known to the system. In the human experiments, subjects were described that the goal is an archway located Northbound from the start. The obstacles are a priori unknown. The system has to find minimum-time route to the goal from the specified start location. Successive trials are simulated to evaluate the task environment learning process.

The parameters of the simulations are given in table 7.1. The vehicle used in the simulations has the same $v_{max} = 5.2$ m/s as in human experiments. For the vehicle used in human experiments, turnrate is inversely proportional to speed (see Fig. 5.4). Maximum turnrates at maximum and minimum speeds are 37.5 deg/s and 14.5 deg/s, respectively. The maximum acceleration a_{max} in simulations in this chapter is 2.3 m/s^2

Table 7.1: Parameters for subgoal-graph guidance system simulations

Parameter	Value
R_{fov}	20 m
θ_{fov}	60^0
d_{safe}	3 m
v_{max}	5.2 m/s
a_{max}	2.3 m/s^2
$dmin_{stop} = v_{max}^2/2a_{max}$	5.9 m
Finite horizon for AWP selection	$2dmin_{stop}$=11.8 m
Δv	1 m/s
Δt	0.2 s
ϵ_p	1.05 m
ϵ_v	0.05 m/s
N_s	2

that is based on the average turnrate ((37.5+14.5)/2 = 26 deg/s) and v_{max}. Parameters (Δt, ϵ_p, and ϵ_v) of online trajectory optimization with CPLEX are the same as in simulations presented in Chapter 5.

The simulations in this chapter uses a point-mass model (Eq. 7.1 and 7.2) for online trajectory optimization to AWPs and subgoals. The model does not account for vehicle heading and therefore the heading is free to have any value at zero speed. In presented simulations, therefore, the minimum speed at AWPs and subgoals is set as a small but non-zero value (1 m/s). Vehicle turnrate in human experiments is limited, which ensures vehicle heading and field of view rotates smoothly. In current simulations, a trajectory to AWP may have sharp changes in heading at zero speeds. Therefore, an AWP candidate in Eq. 7.8 is rejected if trajectory computed from online solver CPLEX does not satisfy the following condition at all time-steps:

$$|\psi(t + \Delta t) - \psi(t)| \leq \Delta\psi_{max}, \tag{7.22}$$

where $\psi(t)$ is the heading at time t, and $\Delta\psi_{max}$ is the maximum allowed change in heading in time Δt. In simulations presented later in this chapter, $\Delta\psi_{max}$ is $\theta_{fov}/2 = 30^0$.

Sensor Parameters

Sensor field of view θ_{fov} is 60^0 that is same as in human experiments. Sensor range R_{fov} is determined based on human gaze data distribution. Figure 7.9 shows distributions of gaze distance r_g for all human subjects. The figure also shows mean gaze distance. The maximum mean distance is around 20 m for subject # 2. Therefore, sensor range R_{fov} in simulations is set 20 m.

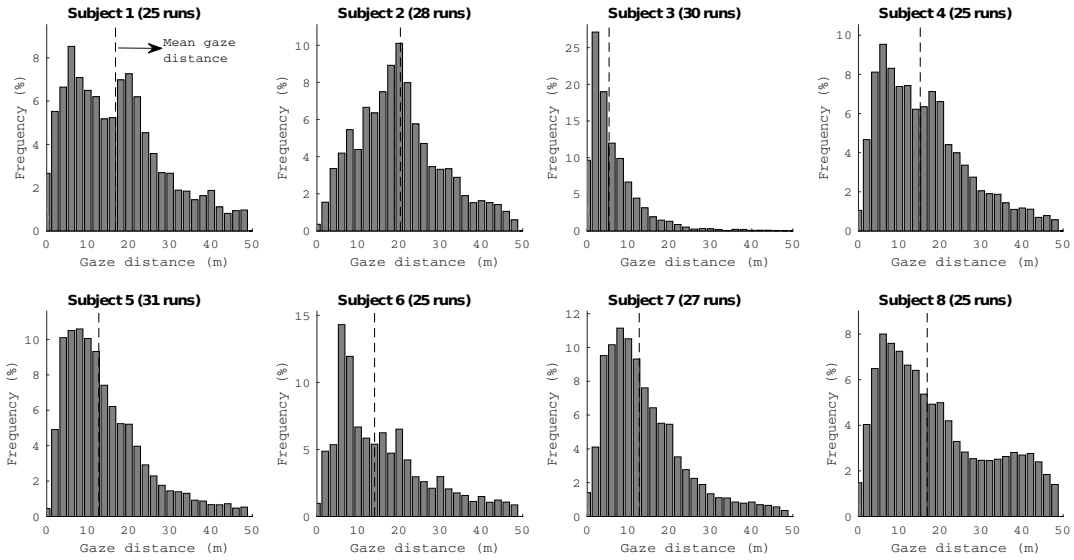


Figure 7.9: Human environment learning experiments: distribution of gaze distance data.

7.3.2 Results

Simulations are run for the parameters shown in table 7.1. Figure 7.10 shows the runs from the exploration stage. The system terminates the exploration stage after run 7 as the subgoal sequence in run 7 is a repeated one from run 2. Flight times are shown in Fig. 7.11. The system finds four distinct routes that are not homotopic to each other.

Figure 7.12(a) shows the subgoal graph learned in the exploration stage. Figure 7.12(b) shows the three subgoal sequences (S_1, S_2, and S_3) that are extracted from the learned graph using Dijkstra's shortest path search algorithm and the perturbation

method along the optimal solution. The consolidation stage executes the three subgoal sequences. Figures 7.13 and 7.11 show the trajectories and flight-times, respectively, for runs 8-25 from the consolidation stage. The best flight-time (38.4 s) is achieved on subgoal sequence S_1.

Dynamic Behavior: Speed

Figure 7.14 shows speed trajectories for the first run in exploration stage and the last run in consolidation stage on the three subgoal sequences (S_1, S_2, and S_3). In runs in the exploration stage, the system slows down frequently as the environment and subgoals are not known. This behavior is conservative as observed in human experiments in Chapters 5 and 6. In the consolidation stage, the system learns subgoals and time-optimal velocities at subgoals, and high speeds are used frequently.

Learning of Speeds at Subgoals

Figure 7.15 shows speeds at subgoals (nodes) for the four sequences S_1, S_2, and S_3 in the consolidation stage. The figure shows the learning of time-optimal speeds at subgoals over successive runs. The speeds are updated iteratively using cost information achieved over successive runs (Eq. 7.16). Flight-times on subgoal sequences decrease significantly at the end of consolidation stage. For example, flight-time on S_1 decreases from 58.6 s in the exploration stage to 40.0 s in the consolidation stage.

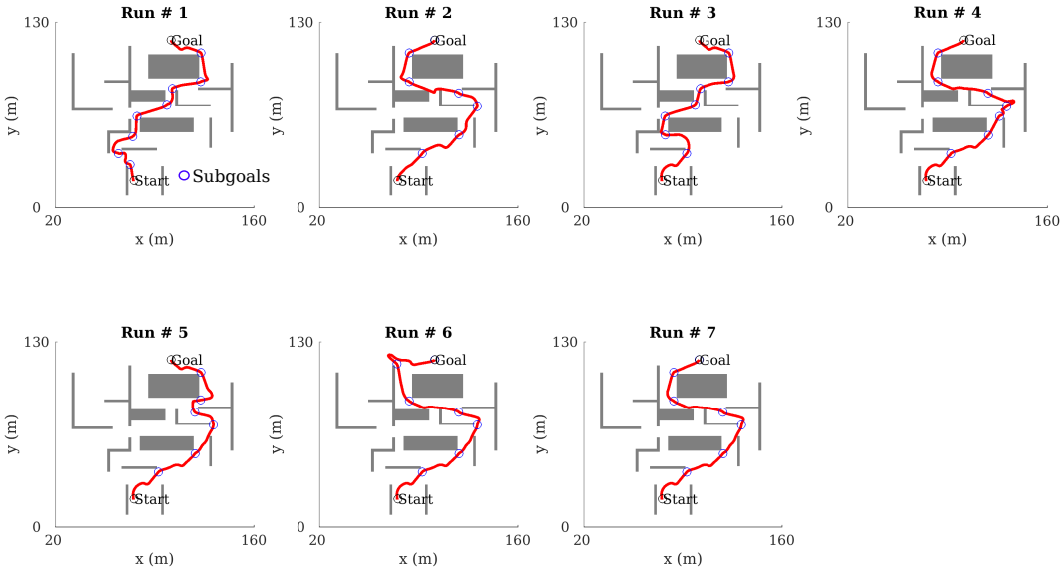


Figure 7.10: Subgoal-graph guidance simulations: trajectories for runs 1-7 from the exploration stage.

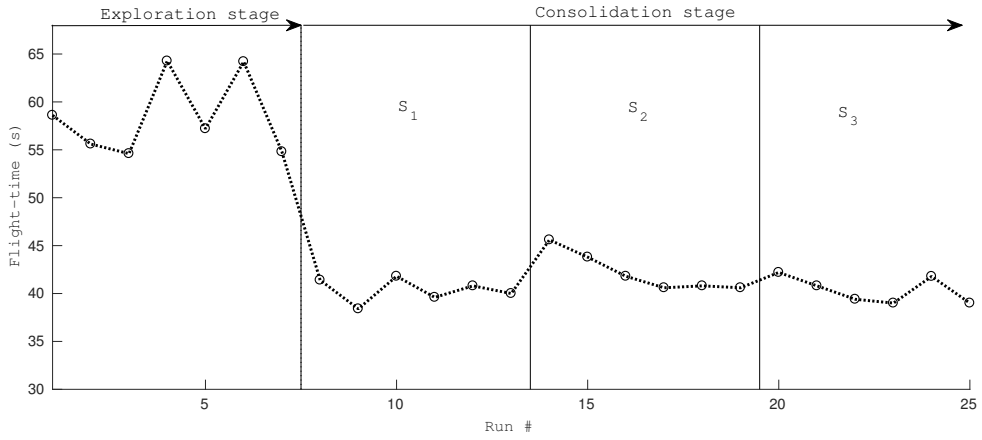
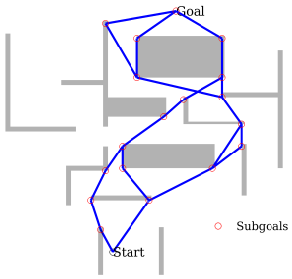
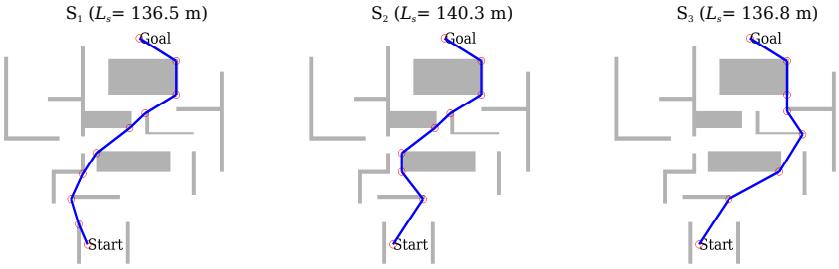


Figure 7.11: Subgoal-graph guidance simulations: flight-times for runs 1-7 (exploration stage) and 8-25 (consolidation stage).



(a) Subgoal graph learned in exploration stage.



(b) Candidate subgoal sequences extracted from the learned subgoal graph, which are to be processed in consolidation stage.

Figure 7.12: Subgoal-graph guidance simulations: (a) subgoal graph learned in the exploration stage and (b) subgoal sequences S_1, S_2, and S_3 extracted from the learned subgoal graph to be executed in the consolidation stage.

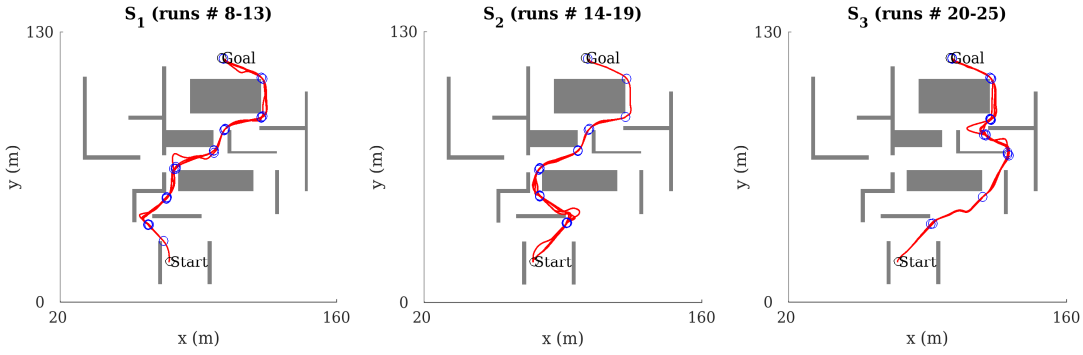


Figure 7.13: Subgoal-graph guidance simulations: trajectories for runs 8-25 from the consolidation stage.

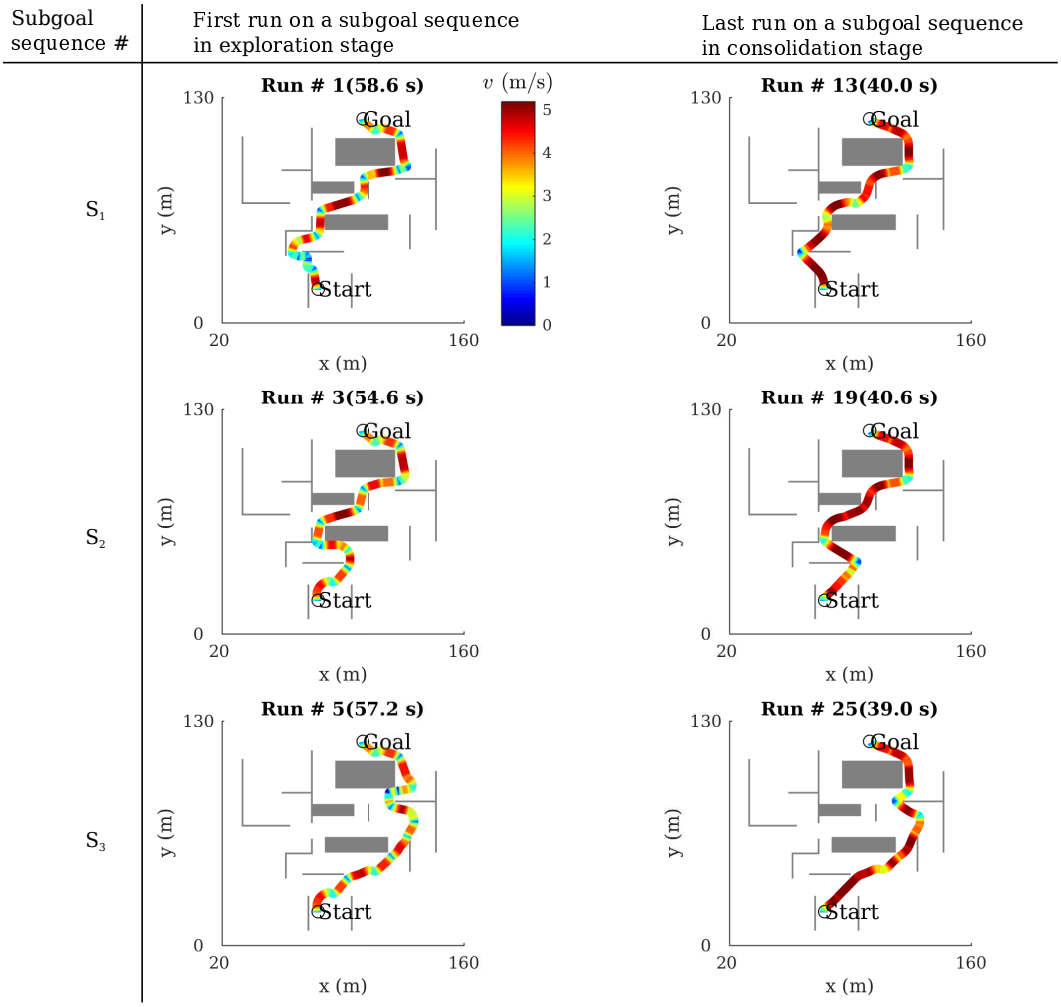


Figure 7.14: Subgoal-graph guidance simulations: speed trajectories for the first run in exploration stage and the last run in consolidation stage on subgoal sequences S_1, S_2, and S_3.

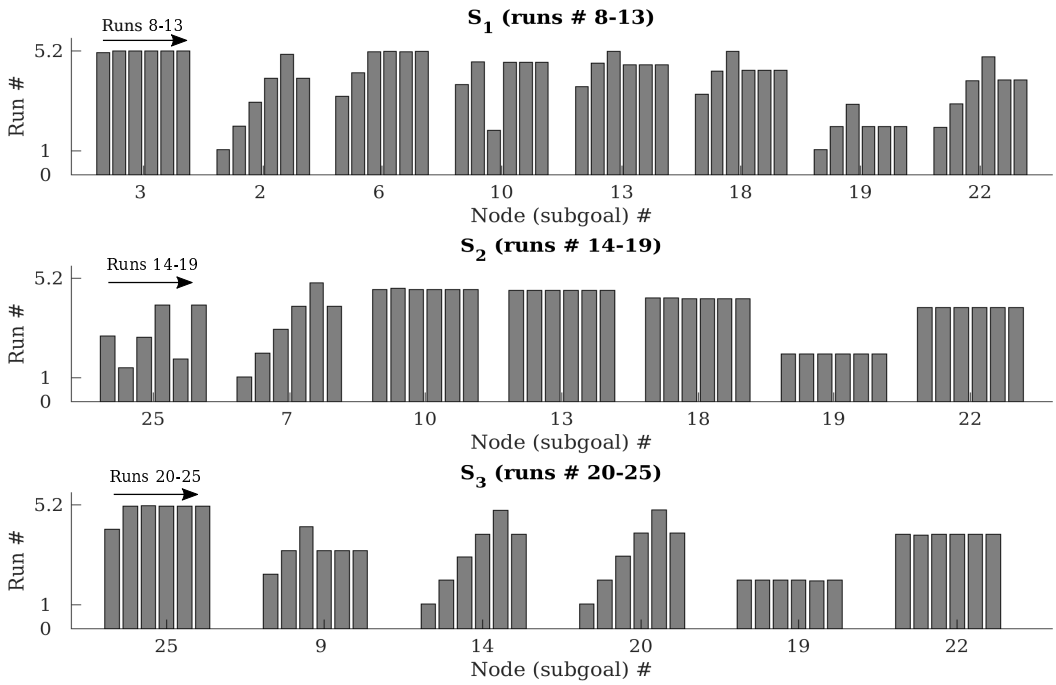


Figure 7.15: Subgoal-graph guidance simulations: speed at subgoals for successive runs on subgoal sequences S_1, S_2, and S_3.

Extracting Guidance Primitives

This section applies the clustering method presented in Chapter 5 to extract guidance primitives from trajectory data from subgoal-graph autonomous guidance simulations. A trajectory cluster represents a guidance primitive π. Figures 7.16 and 7.17 show trajectory segments around corners from runs in exploration and consolidation stages, respectively. The number of clusters for trajectories in exploration and consolidation stages are four and three, respectively. The number of clusters are decided based on the change in cumulative spread of clusters (CS_π) as the number of clusters (N_π) increases (see Fig. 7.18). The cumulative spread of clusters CS_π is computed as follows:

$$CS_\pi = \sum_{I=1}^{I=N_\pi} \sum_{s_i, s_j \in \pi_I, i \neq j} d_s^{ij}, \tag{7.23}$$

where d_s^{ij} is the distance (Eq. 5.25) between two trajectory segments s_i and s_j in I^{th} cluster π_I (see Chapter 5 for more details).

In the exploration stage, the four clusters are numbered in decreasing order of frequencies. In the consolidation stage, the three clusters are numbered according to their similarity (Eq. 5.26) with the clusters from the exploration stage. Figure 7.19 shows speed mean and variance for clusters in exploration and consolidation stages. Speed in the exploration stage reduces near corners more than it does in the consolidation stage. The speed variance near corners is higher in the consolidation stage because the subgoal speeds in starting runs are set to be lower values (Eq. 7.21). The subgoal speeds incrementally update to higher values until the speed and flight-time converge.

Figure 7.20 shows trajectory clusters from simulations's consolidation stage and human subject # 1's runs 16-last. The clusters from simulations and human data are matched based on distance d_π^{IJ} as defined by Eq. 5.26. Fig. 7.20 also shows the speed map for average trajectory and frequency of each cluster. Trajectory clusters from simulations and human data are significantly different. For instance, the human subject's most frequent cluster (frequency of 56.6 %) is identical to the least frequent cluster (frequency of 16.0 %) in the simulation results. The frequency of the most frequent cluster # 1 in simulations is 47.2 %, which is identical to human subject's cluster # 2 (frequency of 13.2 %). The cumulative distance d_π^{IJ} between the most frequent cluster in simulations and the corresponding identical cluster of the human

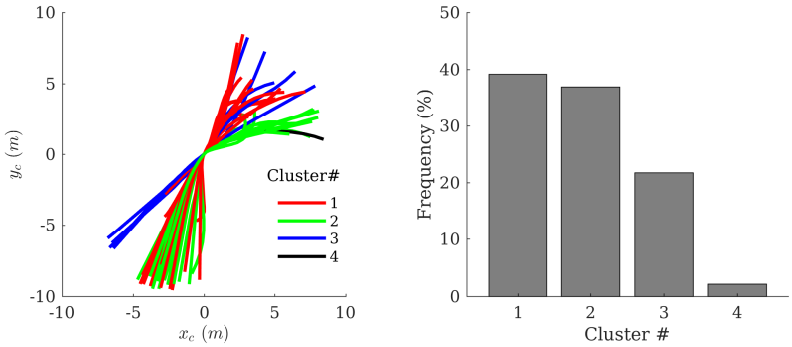


Figure 7.16: Subgoal-graph guidance simulations: trajectory clusters in corner frame for trajectory data from runs 1-7 (exploration stage).

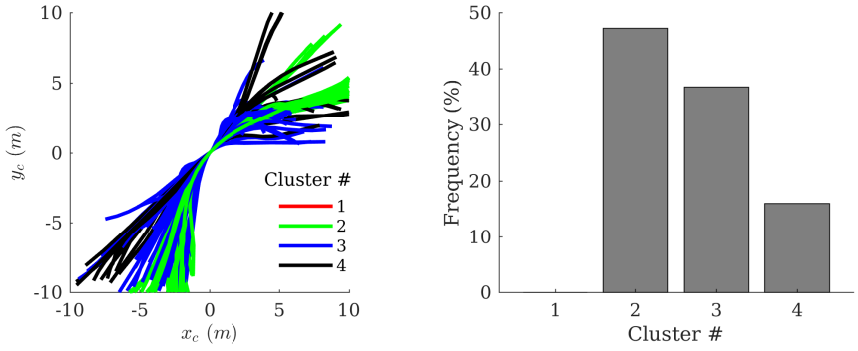
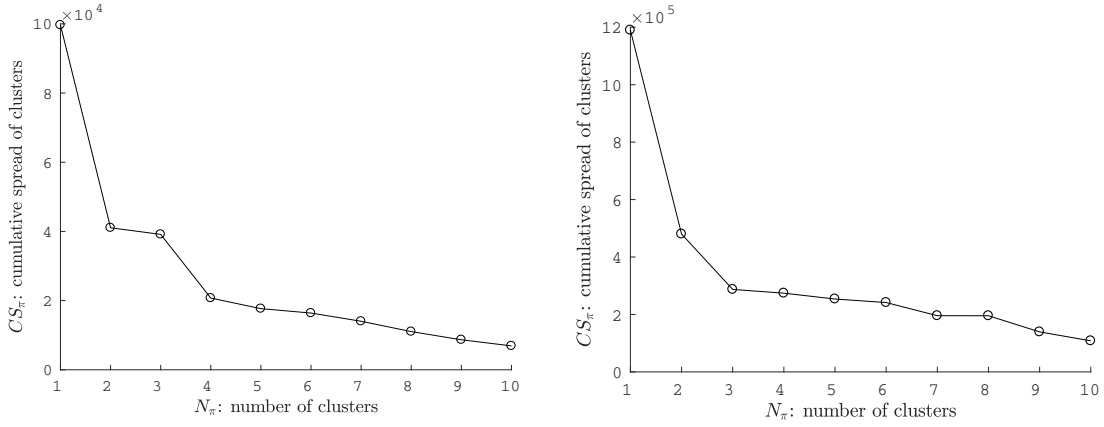


Figure 7.17: Subgoal-graph guidance simulations: trajectory clusters in corner frame for trajectory data from runs 8-25 (consolidation stage).



(a) Exploration stage (b) Consolidation stage

Figure 7.18: Number of clusters (N_π) vs. cumulative spread of clusters (CS_π) for trajectories in corner frame as shown in Fig. 7.16 and 7.17.

subject is the maximum of 197.1 m.

Figure 7.21 shows the trajectory clusters for simulations and human subject # 1 in the global task environment. Identical clusters are shown in same colors. Most frequent clusters for both simulations and human subject, which have frequencies of 47.2 % and 56.6 %, respectively, are on the route S_1 (Fig. 7.12). In simulations, the safety distance is set as 3 m (table 7.1). Whereas the subject # 1 has mean minimum distance (r_{min}) of 0.2 m from obstacle corners (see Fig. 6.6). Therefore, the simulation trajectories on route S_1 are not as straight as those are for the human subject. This observation is also evident in trajectory cluster π_1 (frequency of 47.2 %) for simulations vs. cluster π_3 (frequency of 56.6 %) for subject # 1.

Table 7.2 shows variances of the trajectory clusters shown in Fig. 7.20. The variance σ_π of a trajectory cluster π is computed as follows:

$$\sigma_\pi = \frac{1}{|\pi|} \sum_{i=1}^{i=|\pi|} {d_s^{im}}^2, \tag{7.24}$$

where d_s^{im} is the distance of a trajectory $s_i \in \pi$ from the mean trajectory of the cluster (e.g., mean trajectories are shown in Fig. 7.20). The distance d_s^{im} is computed using Eq. 5.25. Trajectory resolutions for simulations and the human subject are 0.2 s (see

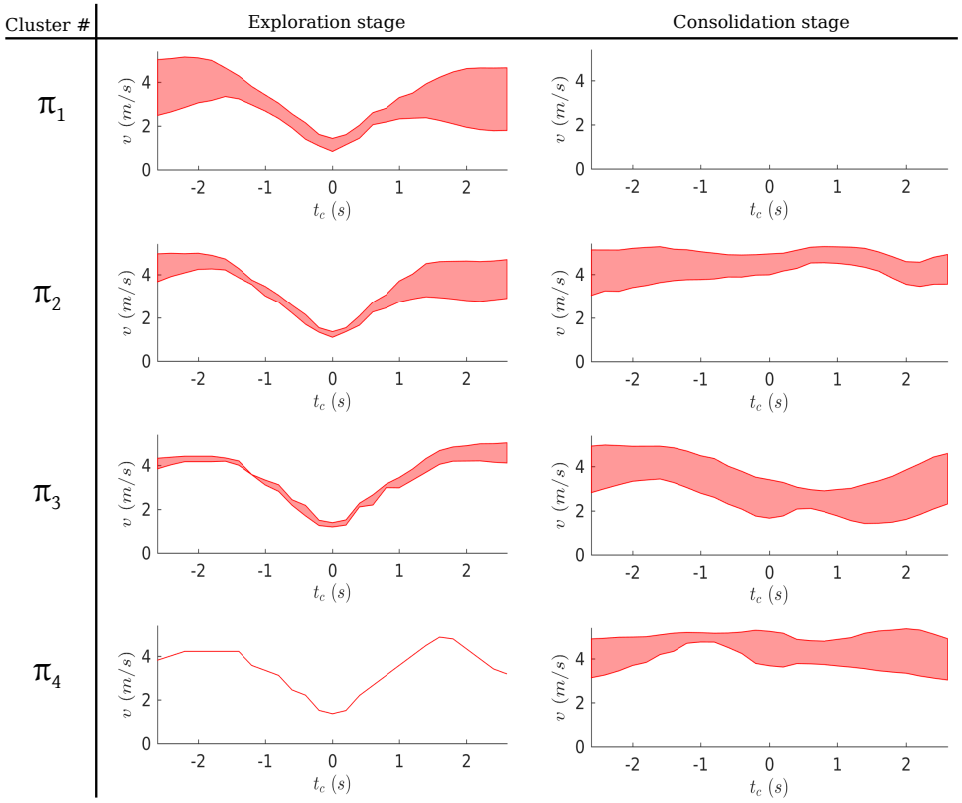


Figure 7.19: Speed mean and variance trajectories for clusters for runs 1-7 (exploration stage) and 8-25 (consolidation stage).

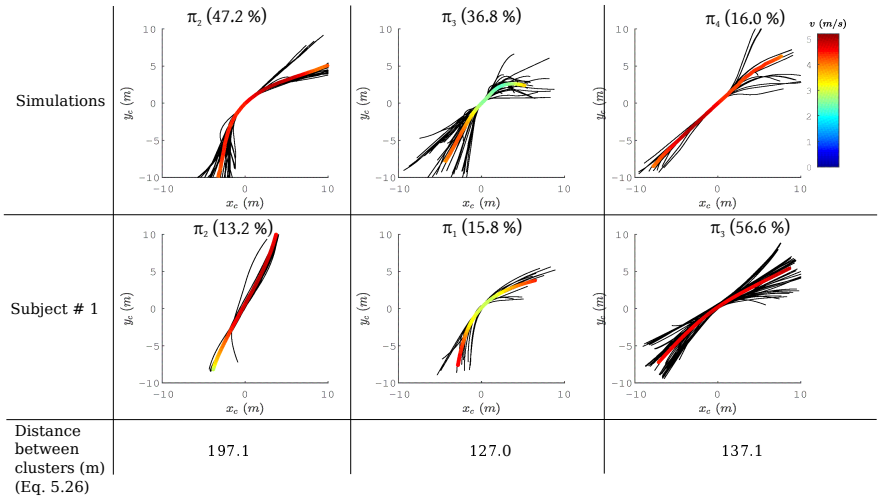


Figure 7.20: Clusters from simulations's consolidation stage and human subject # 1's runs 16-last are matched based on distance d_{π}^{IJ} (Eq. 5.26). Identical clusters are placed in same column. Speed map for average trajectory for each cluster is also shown.

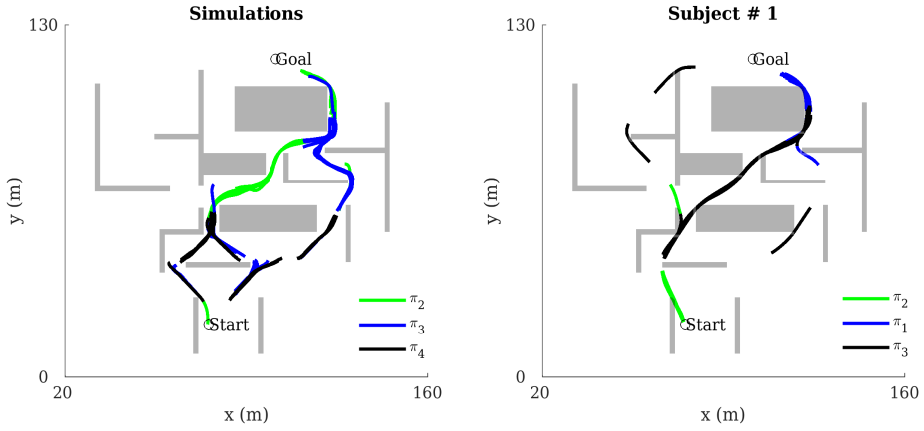


Figure 7.21: Identical clusters from simulations's consolidation stage and human subject # 1's runs 16-last are shown in the global task environment. Identical clusters are shown in same colors.

table 7.1) and 0.02 s (see section 5.2.5), respectively. Therefore, for a scaled comparison between variances of simulations and human subject's clusters, d_s^{im} for human trajectories are scaled down by a factor of $0.2/0.02 = 10$.

Table 7.2: Variances of trajectory clusters from simulations's consolidation stage and human subject # 1's runs 16-last. The identical clusters between simulations and the subject are in same rows.

Simulations	Subject # 1
π_2 (32.4 m)	π_2 (19.1 m)
π_3 (71.7 m)	π_1 (51.0 m)
π_4 (69.2 m)	π_3 (59.3 m)

Table 7.2 shows that the human subject's trajectory clusters have smaller variances in comparison to the clusters from simulations. Fig. 7.20 shows that the human subject executes a trajectory cluster for an intended behavior. For example, the subject uses π_1 for high angle turns around corners, and π_2 and π_3 for passing corners in almost straight lines at different angles. In simulations, the trajectory clusters do not show such organization of behavior. For example, the trajectories seem to diverge after they pass corners. The consistency of human trajectory clusters support that the clusters represent interaction patters in their interaction with the task environment. A future work is to investigate and capture the direction change in human interaction patterns.

Simulation vs. Human Performance

Table 7.3 summarizes the comparison between simulation results and best human subject (# 1). Both the simulation system and the human subject perform twenty-five runs, and choose same route S_1 (Fig. 7.12) as their best routes. The human subject achieves better flight-time (31.0 s) than simulation (38.4 s) on the route S_1. The simulation system explores four distinct (non-homotopic) routes whereas the subject explores seven routes (Fig. 5.2). The frequency of the subject # 1's most dominant guidance primitive in final runs is 56.6 % while the frequency of the second most frequent guidance primitive is 15.8 %. For the simulation results, frequencies of the most and second most frequent guidance primitives are 47.2 % and 36.8 %, respectively. It is to note

that there are some basic differences between human subject's experiments and simulations. Two of those differences are the following: 1) vehicle dynamics in simulations is an approximation of the dynamic model used in human experiments, 2) simulations use numerical trajectory optimization whereas humans use perceptual guidance policies such as tau-control [2, 3].

Table 7.3: Subgoal-graph guidance system simulation's vs. human subject # 1's performance

Performance metric	Simulation	Human subject # 1
Best flight-time (s)	38.4	31.0
Best route (routes as named in Fig. 7.12)	S_1	S_1
Runs	25	25
Number of explored routes	4	7
Frequency of most (and second most) frequent guidance primitive (%)	47.2 (36.8)	56.6 (15.8)

7.4 Discussion

This section first discusses the advantages of the subgoal-graph guidance system over the SVF system [9, 23] presented in Chapter 3. Next, the section presents a brief comparison with existing boundary tracking and topology based path planning methods (e.g., Voronoi diagrams), and highlights contributions of the subgoal-graph guidance system presented in this dissertation.

7.4.1 Subgoal-Graph vs. Occupancy Map and SVF Based System

The autonomous guidance system (Fig. 3.3) presented in Chapter 3 is based on metric representation of the environment. The environment is modeled as an occupancy map over a grid. The guidance policy is represented as spatial value function (SVF) map over the grid. The memory and computational requirements increase as the task domain size increases, the grid size reduces, or the size of motion primitive library increases.

The occupancy map representation of spatial environments does not seem realizable for humans.

The subgoal-graph guidance system, presented in this chapter, models the global environment as a subgoal graph. The task environment memory structure based on subgoal graph G combines connection matrix Q, cost-to-go CTG at subgoals, and incremental cost DC between connected subgoals. This representation requires less memory than a global occupancy map and SVF. For example for the test environment used in simulations, total number of convex corners that are potential subgoal candidates is 54. Therefore, the size of subgoal-graph memory structure is a few metrices with maximum size of 54×54. On the contrary for occupancy map representation, if grid size is 1 m, the space is represented as a matrix of the size of 140×130. The memory requirements increase for higher grid resolutions. Studies in the past (e.g., [7, 8]) have supported the hypothesis that human pilots use subgoals to abstract a task space in guidance tasks.

Sparse Representation of Task Space

The subgoal-graph guidance system only relies on known/learned subgoal graph G and visible obstacle boundary. In contrast, the SVF guidance system is based on an occupancy map representation of task space. The simulation results for the subgoal graph system show that the system finds a near-optimal solution using the sparse representation of task space.

7.4.2 Comparison with Existing Path Planning Methods

Feit and Mettler [180] presented a constrained optimal control formulation for the concept of subgoals in path planning and guidance tasks. The authors presented an algorithm that computes a path as a sequence of subgoals between start and goal states. The algorithm assumes full knowledge of obstacles in the environment. It checks if a path is free or obstructed by obstacles. In the latter case, the algorithm introduces new subgoals on the path, which are points on obstacle boundaries tangent to the distorted path. In this dissertation, the subgoal-graph guidance system does not assume full knowledge of obstacles. The system learns subgoal graph and builds graph knowledge over successive trials.

Boundary Tracking

Boundary tracking/following is an early approach used in mobile robots, in which a robot moves along an obstacle boundary keeping a minimum distance from it. As Mataric [185] stated, "The avoiding behavior is simply a survival mechanism while boundary following is the basis of the robot's perception of the world." Bug algorithm [186, 187] is a tactile boundary tracking/following method in unknown obstacle fields, which assumes zero sensing range. Past studies have shown that even human pilots exhibit boundary tracking behavior when they are close to a boundary (e.g., physical boundary as ground or control boundary as maximum speed limit) [188]. Pilots' boundary tracking behavior result in pilot induced oscillations. The subgoal-graph guidance system, presented in this chapter, selects an end point of visible boundary as instantaneous or temporary subgoal and moves towards it, which can be described as a form of boundary following. As the system approaches the end point of the visible boundary, the boundary expands until a corner is seen, i.e., tracking an INC to GNC as shown in Fig. 7.2. The corner serves as a steady subgoal and the system reaches to it.

Roadmap or Topological Path Planning

Topological representation of task space requires less memory and computation than metric grid maps. The grid map approach becomes computationally intractable for high-dimensional configuration spaces. Roadmap or topological map representation of a problem space reduces the path planning into a graph search problem [24]. Some of graph search algorithms are Dijkstra [27], A* [28], D* [29], etc. For example, three types of roadmap or topological path planning approaches are: probabilistic roadmaps (e.g., [189, 190, 191, 192]), visibility graphs (e.g., [193, 194, 31, 195, 196]), and Voronoi graphs (e.g., [197, 198, 30, 199, 200, 201]).

In probabilistic roadmap methods (PRMs), random samples in configuration space are used as nodes in a graph representation of problem space. When goal is found, an optimal path between start and goal configurations is computed using a graph search algorithm (e.g., A*). PRMs have been successfully implemented in manipulator problems that have in general high-dimensional and complex configuration space [191, 24]. PRMs are often used in obstacle fields that are known in advance [192]. The subgoal-graph

guidance system presented in this chapter is based on subgoals that are invariants in agent-environment interactions in goal-directed guidance and navigation tasks. Prior work [7, 8] has showed that subgoals are used by human pilots to organize their spatial planning.

Visibility graph based path planning is often applied in environments that have polyhedral obstacles [193]. The vertices of obstacles are nodes in the visibility graph. Two vertices are connected if they are visible to each other. In Voronoi roadmap methods, a robot finds paths that are equidistant from surrounding obstacles. The graph nodes are placed at intersections of such paths. Both visibility and Voronoi graph based path planning have been applied in mobile robots and UAVs (e.g., [31, 30]). The subgoal graph used in this dissertation is a subset of visibility graph in a polygonal obstacle field. A subgoal is associated with the optimal state (e.g., velocity) at the subgoal position. Also, the subgoal-graph guidance system presented in this chapter uses no memory of obstacle field in learning the subgoal graph. The system only knows the learned subgoal graph and visible obstacles. The graph connectivity is extracted at the end of each trial using the executed path that is remembered along with the corners (subgoal candidates) passed by.

7.4.3 Limitations

First limitation of the subgoal-graph guidance system presented in this chapter is that the system assumes 2-D polygonal obstacles. In real-world environments, obstacles are in general 3-D and not always polygonal. Second, the system assumes static environment. Third, the environment sensing and path tracking (control implementation) are assumed to be accurate.

7.4.4 Generalization

This dissertation uses the subgoal-graph representation and a guidance primitive extraction method to evaluate human environment learning in guidance tasks. The evaluation framework can be extended and applied to other spatial guidance and control tasks such as surgery because the elements, e.g., interaction patterns, used for modeling human guidance are common to other spatial tasks. For example for surgical tasks, Li

et. al. [202] showed that expert surgeons exploit interaction patterns to organize spatial behavior.

Chapter 8

Conclusions and Future Directions

8.1 Conclusions

This dissertation starts with an evaluation and validation of learning processes in the sensory-predictive guidance system [9, 23] that uses a receding horizon trajectory optimization associated with the spatial value function (SVF). The SVF guidance system is simulated for a baseline case with ideal sensing and zero noise in the system. The task is to find a time-optimal trajectory between specified start and goal states. Successive runs are simulated with each run using the SVF learned in the preceding run. The learned SVF describes the information necessary to determine the optimal guidance behavior over geographical space. The presented framework enables learning simultaneous to operation in unknown or partially known environments. This approach is more efficient than learning the environment and subsequently using this information for online trajectory planning. For some real-world applications such as exploring an unknown territory, learning and subsequent planning may not be feasible. Learning and subsequent task operation would require mapping the whole environment, as it is generally not possible to predict which part of the environment would be significant to the task performance. The guidance system presented in this study computes the trajectory using up-to-date environment information obtained by combining the information from onboard sensors with existing information. A key benefit is that the local information is

integrated in the global context and therefore the guidance system can provide reactive behaviors that are consistent with the long-term goal.

In the simulations of the SVF guidance system, the learning process is observed and analyzed at several levels such as: environment knowledge, spatial features in the SVF, and dynamic/control behavior. The guidance system updates the occupancy probability map for the environment based on sensory data. As the environment knowledge alone is not sufficient for optimal spatial guidance, the system assimilates the environment knowledge into the SVF, i.e., spatial guidance behavior. Spatial features (subgoals, attracting and repelling manifolds) emerge in the SVF. These spatial features account for how vehicle dynamics interact with the environment characteristics (layout, scale, etc.). To test the learned SVF, a test case in which the guidance system does not rely on environment sensing and uses the learned SVF is simulated. The guidance system performs close to the optimal as the spatial features in the learned SVF allow the vehicle to navigate around the obstacles in a fashion compatible with the vehicle dynamics. Analysis of the acceleration profile shows that the vehicle uses less turning in latter runs. The guidance policy used in latter runs is more straight-line movement and less turning. For learning at the control level, this dissertation investigates the velocity and cost maps in the vehicle body frame. Patterns emerge in the control behavior in body frame. A segmentation in the behavior in the body frame is observed based on the vehicle speed. The vehicle flies straight at high speeds. The lower the vehicle speed the higher the turning curvature is.

In the second part, this dissertation extends prior concept of the interaction patterns to formulate hypothesis and an analysis framework for environment learning in goal-directed guidance tasks in unknown obstacle fields. The dissertation presents a graph framework based on subgoals that are patterns in sensory-motor behavior in interaction with the spatial environment and task elements to analyze human environment learning in agile guidance tasks. The graph representation of task environment enables a formal assessment of human learning of the following three elements: task environment structure (subgoal graph), optimal behavior (cost-to-go) across graph, and sensory-motor primitives. The framework uses an optimal graph search method to evaluate human decision-making (subgoal selection) in navigation tasks. The presented model allows testing an operator's rationality and accuracy of the model. Finally, the framework

uses a clustering method to extract guidance primitives and study their emergence as a result of task environment learning over multiple trials.

The analysis framework is applied on human data collected from guidance experiments in a simulated quasi three-dimensional environment. The subjects had no priori knowledge of the obstacle-field. Each subject performed multiple trials between pre-specified start and goal locations to find minimum-time routes. The data analysis revealed that control skill level of an operator effects learning of unknown task environments. For an operator with reliable control over vehicle dynamics, better differentiated and concentrated control and perceptual policies, i.e., guidance primitives, emerge as interaction patterns. The well formed guidance primitives due to reliable control skills relieve an operator's attention from low-level task elements such as stabilizing the vehicle. The operator can focus his/her attention at high-level task elements such as subgoals for planning and learning.

In the third (last) part, the dissertation extends the subgoal graph framework used for analysis of human environment learning, and presents an autonomous guidance algorithm for goal-directed navigation in unknown environments. The subgoal graph based autonomous guidance system is simulated and results show that the system successfully finds a near-optimal route between pre-specified start and goal locations. The system uses a sparse representation, i.e., subgoal graph and visible obstacle boundary, of the environment.

8.2 Future Directions

This section briefly presents future research directions.

8.2.1 Autonomous Guidance

The subgoal graph guidance system presented in this dissertation uses a state-space dynamic model and numerical optimization for online trajectory planning to AWPs. A future direction is to use a perceptual guidance policy such as Tau-gap guidance [3] to generate human-like guidance behavior.

8.2.2 Human Guidance

A potential area of future research is to study how human operators transfer and use their knowledge about the environment and task elements if they start from a different location or they navigate in a different environment with similar structures such as layout and gap between obstacles. The hypothesis is that subjects will exploit the interaction patterns they learned in previous tasks. Such studies can provide an understanding about how human operators transfer their knowledge among problems that have similar task structure.

References

[1] Gibson, J. J., "The Ecological Approach to Visual Perception," *Blackwell Publishing*, Boston: Houghton Mifflin, 1979, pp. 119–136.

[2] Lee, D. N., "A Theory of Visual Control of Braking Based on Information about Time-to-Collision," *Perception*, Vol. 5, No. 4, 1976, pp. 437–459. doi: 10.1068/p050437

[3] Lee, D. N., "Guiding Movement by Coupling Taus," *Ecological Psychology*, Vol. 10, No. 3-4, 1998, pp. 221–250. doi: 10.1080/10407413.1998.9652683

[4] Lee, D. N. and Kalmus, H., "The Optic Flow Field: The Foundation of Vision [and discussion]," *Philosophical Transactions of the Royal Society B: Biological Sciences*, Vol. 290, No. 1038, 1980, pp. 169–179. doi: 10.1098/rstb.1980.0089

[5] Simon, H. A., "Invariants of Human Behavior," *Annual Review of Psychology*, Vol. 41, 1990, pp. 1–20. doi: 10.1146/annurev.ps.41.020190.000245

[6] Kong, Z. and Mettler, B., "An Investigation of Spatial Behavior in Agile Guidance Tasks," *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, Anchorage, Alaska, 9-12 October, 2011, pp. 2473–2480. doi: 10.1109/ICSMC.2011.6084049

[7] Kong, Z. and Mettler, B., "Modeling Human Guidance Behavior Based on Patterns in Agent-Environment Interactions." *Human-Machine Systems, IEEE Transactions on*, Vol. 43, No. 4, July 2013, pp. 371–384. doi:10.1109/TSMC.2013.2262043

[8] Mettler, B., Kong, Z., Li, B., and Andersh, J., "Systems View on Spatial Planning and Perception based on Invariants in Agent-Environment Dynamics," *Frontiers in Neuroscience*, Vol. 8, No. 439, Jan. 2015. doi: 10.3389/fnins.2014.00439

[9] Mettler, B., Dadkhah, N., and Kong, Z., "Agile Autonomous Guidance Using Spatial Value Functions," *Control Engineering Practice*, Vol. 18, No. 7, July 2010, pp. 773–788. doi:10.1016/j.conengprac.2010.02.013

[10] Kong, Z. and Mettler, B., "On the General Characteristics of 2D Optimal Obstacle-Field Guidance Solution," *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, Shanghai, December 15–18, 2009, pp. 3448–3453. doi: 10.1109/CDC.2009.5400828

[11] Kong, Z. and Mettler, B., "Foundations of Formal Language for Humans and Artificial Systems Based on Intrinsic Structure in Spatial Behavior," *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Francisco, California, 25-30 September, 2011, pp. 3093–3100. doi: 10.1109/IROS.2011.6094903

[12] Mettler, B. and Kong, Z., "Mapping and Analysis of Human Guidance Performance From Trajectory Ensembles," *Human-Machine Systems, IEEE Transactions on*, Vol. 43, No. 1, September 2012, pp. 32–45. doi: 10.1109/TSMCA.2012.2207110

[13] Warren, W. H., "The Dynamics of Perception and Action," *Psychological Review*, Vol. 113, No. 2, April 2006, pp. 358–389. doi: 10.1037/0033-295X.113.2.358

[14] Andersh, J., Li, B., and Mettler, B., "Modeling Visuo-Motor Control and Guidance Functions in Remote-Control Operation," *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, 2014, pp. 4368–4374. doi: 10.1109/IROS.2014.6943180

[15] Downs, R. M. and Stea, D. (Eds.) "Image and Environment: Cognitive Mapping and Spatial Behavior," *Transaction Publishers*, 1973, page 9.

[16] Feit, A. and Mettler, B., "Experimental Framework for Investigating First Person Guidance and Perception," *Systems, Man, and Cybernetics (SMC), 2015 IEEE*

International Conference on, HongKong, October 9–12, 2015, pp. 974–980. doi: 10.1109/SMC.2015.177

[17] Bryson, A. E., "Optimal Control-1950 to 1985," *IEEE Control Systems*, Vol. 16, No. 3, June 1996, pp. 26–33. doi: 10.1109/37.506395

[18] Schouwenaars, T., Mettler, B., Feron, E., and How, J., "Hybrid Model for Trajectory Planning of Agile Autonomous Vehicles," *Journal of Aerospace Computing, Information, and Communication* Vol. 1, No. 12, 2004, pp. 629–651. doi: 10.2514/1.12931

[19] Bellingham, J., Richards, A., and How, J. P., "Receding Horizon Control of Autonomous Aerial Vehicles," *American Control Conference, 2002. Proceedings of the 2002*, Vol. 5, 2002, pp. 3741–3746. doi: 10.1109/ACC.2002.1024509

[20] Mettler, B. and Bachelder, E., "Combining On- and Offline Optimization Techniques for Efficient Autonomous Vehicle's Trajectory Planning," *AIAA Guidance, Navigation, and Control Conference 2005*, San Francisco, California, Vol. 1, 15-18 August, 2005, pp. 499–511. doi: 10.2514/6.2005-5861

[21] Mettler, B. and Toupet, O., "Receding Horizon Trajectory Planning with an Environment-Based Cost-to-go Function," *Proceedings of the 44th IEEE Conference on Decision and Control*, Seville, Spain, 12-15 December, 2005, pp. 4071–4076. doi: 10.1109/CDC.2005.1582799

[22] Mettler, B. and Kong, Z., "Receding Horizon Trajectory Optimization with a Finite-State Value Function Approximation," *2008 American Control Conference*, Seattle, WA, 11-13 June, 2008, pp. 3810–3816. doi: 10.1109/ACC.2008.4587087

[23] Dadkhah, N. and Mettler, B., "Sensory Predictive Guidance in Partially Known Environment," *AIAA Guidance, Navigation, and Control Conference*, Portland, Oregon, 8-11 August, 2011. doi: 10.2514/6.2011-6293

[24] Goerzen, C., Kong, Z., and Mettler, B., "A Survey of Motion Planning Algorithms from the Perspective of Autonomous UAV Guidance," *Journal of Intelligent and Robotic Systems*, published online 17 November 2009; Vol. 57, No. 1-4, 2010, pp. 65–100. doi: 10.1007/s10846-009-9383-1

[25] Verma, A. and Mettler, B., "Computational Investigation of Environment Learning in Guidance and Navigation," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 2, 2017, pp. 371-389. doi: 10.2514/1.G001889

[26] Ferguson, D., Likhachev, M., and Stentz, A., "A Guide to Heuristic-Based Path Planning," *Proceedings of the international workshop on planning under uncertainty for autonomous systems, international conference on automated planning and scheduling (ICAPS)*, 2005, pp. 9–18.

[27] Dijkstra, E. W., "A Note on Two Problems in Connexion with Graphs," *Numerische Mathematik*, Vol. 1, No. 1, December 1959, pp. 269–271. doi: 10.1007/BF01386390

[28] Hart, P. E., Nilsson, N. J., and Raphael, B., "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Transactions on Systems Science and Cybernetics*, Vol. 4, No. 2, July 1968, pp. 100–107. doi: 10.1109/TSSC.1968.300136

[29] Stentz, A., "Optimal and Efficient Path Planning for Partially-Known Environments," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '94)*, San Diego, CA, 8-13 May, 1994, Vol. 4, pp. 3310–3317. doi: 10.1109/ROBOT.1994.351061

[30] Bortoff, S. A., "Path planning for UAVs," *American Control Conference*, Vol. 1, No. 6, Chicago, Illinois, September 2000, pp. 364–368. doi:10.1109/ACC.2000.878915

[31] Bellingham, J., Tillerson, M., Richards, A., and How, J. P., "Multi-Task Allocation and Path Planning for Cooperating UAVs," *Cooperative Control: Models, Applications and Algorithms*, Vol. 1, No. 3, 2004, pp. 23–41. doi: 10.1007/978-1-4757-3758-5-2

[32] Fox, D., Burgard, W., and Thrun, S., "The Dynamic Window Approach to Collision Avoidance," *IEEE Robotics and Automation Magazine*, Vol. 4, No. 1, March 1997, pp. 23–33. doi: 10.1109/100.580977

[33] LaValle, S. M. and Kuffner, J. J., "Randomized Kinodynamic Planning," *The International Journal of Robotics Research*, Vol. 20, No. 5, May 2001, pp. 378–400. doi: 10.1177/02783640122067453

[34] Michels, J., Saxena, A., and Ng, A. Y., "High Speed Obstacle Avoidance Using Monocular Vision and Reinforcement Learning," *Proceedings of the 22nd international conference on Machine learning (ICML '05)*, 2005, pp. 593–600. doi: 10.1145/1102351.1102426

[35] Abbeel, P., Coates, A., Quigley, M., and Ng, A. Y., "An Application of Reinforcement Learning to Aerobatic Helicopter Flight," *Proceedings of the Advances in Neural Information Processing (NIPS '07)*, Vol. 19, 2007.

[36] Richards, R. and Boyle, P., "Combining Planning and Learning for Autonomous Vehicle Navigation," *AIAA Guidance, Navigation, and Control Conference*, Toronto, Cananda, 2-5 August, 2010. doi: 10.2514/6.2010-7866

[37] Mayne, D. Q., Rawlings, J. B., Rao, C. V., and Scokaert, P. O. M., "Constrained Model Predictive Control: Stability and Optimality," *Automatica*, Vol. 36, No. 6, June 2000, pp. 789–814. doi: 10.1016/S0005-1098(99)00214-9

[38] Borrelli, F., Bemporad, A., Fodor, M., and Hrovat, D., "An MPC/Hybrid System Approach to Traction Control," *IEEE Transactions on Control Systems Technology*, Vol. 14, No. 3, May 2006, pp. 541–552. doi: 10.1109/TCST.2005.860527

[39] Richards, A. and How, J. P., "Model Predictive Control of Vehicle Maneuvers with Guaranteed Completion Time and Robust Feasibility," *Proceedings of the 2003 American Control Conference*, Vol. 5, June 2003, pp. 4034–4040. doi: 10.1109/ACC.2003.1240467

[40] Richards, A. and How, J., "Decentralized Model Predictive Control of Cooperating UAVs," *Decision and Control, 43rd IEEE Conference on*, Vol. 4, December 2004, pp. 4286–4291. doi: 10.1109/CDC.2004.1429425

[41] Schouwenaars, T., Mettler, B., Feron, E., and How, J., "Hybrid Architecture for Full-Envelope Autonomous Rotorcraft Guidance," In *American Helicopter Society 59th Annual Forum*, Phoenix, Arizona, May 6-8, 2003, Vol. 371.

[42] Mettler, B. and Bachelder, E., "Combining On-and Offline Optimization Techniques for Efficient Autonomous Vehicle's Trajectory Planning," *AIAA Guidance,*

Navigation, and Control Conference, San Francisco, California, 15-18 August, 2005. doi: 10.2514/6.2005-5861

[43] Toupet, O. and Mettler, B., "Design and Flight Test Evaluation of Guidance System for Autonomous Rotorcraft," *AIAA Guidance, Navigation, and Control Conference*, Keystone, Colorado, 21-24 August, 2006. doi: 10.2514/6.2006-6201

[44] Dadkhah, N., Korukanti, V. R., Kong, Z., and Mettler, B., "Experimental Demonstration of An Online Trajectory Optimization Scheme Using Approximate Spatial Value Functions," *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, Shanghai, 15-18 December, 2009, pp. 2978–2983. doi: 10.1109/CDC.2009.5400429

[45] Ramalingam, G. and Reps, T., "An Incremental Algorithm for a Generalization of the Shortest-Path Problem," *Journal of Algorithms*, Vol. 21, No. 2, September 1996, pp. 267–305. doi: 10.1006/jagm.1996.0046

[46] Leonard, J. J. and Durrant-Whyte, H. F., "Simultaneous Map Building and Localization for an Autonomous Mobile Robot," *Intelligent Robots and Systems '91. 'Intelligence for Mechanical Systems, Proceedings IROS '91. IEEE/RSJ International Workshop on*, Vol. 3, November 1991, pp. 1442–1447. doi: 10.1109/IROS.1991.174711

[47] Jafri, S. R. U. N. and Chellali, R., "A Distributed Multi Robot SLAM System for Environment Learning," *Robotic Intelligence In Informationally Structured Space (RiiSS), 2013 IEEE Workshop on*, Singapore, April 2013, pp. 82–88. doi: 10.1109/RiiSS.2013.6607933

[48] Weiss, S., Scaramuzza, D., and Siegwart, R., "Monocular-SLAM-Based Navigation for Autonomous Micro Helicopters in GPS-Denied Environments," *Journal of Field Robotics*, published online 12 October 2011; Vol. 28, No. 6, November 2011, pp. 854–874. doi: 10.1002/rob.20412

[49] Meyer, J. A. and Filliat, D., "Map-Based Navigation in Mobile Robots:: II. A Review of Map-Learning and Path-Planning Strategies," *Cognitive Systems Research*, Vol. 4, No. 4, December 2003, pp. 283–317. doi: 10.1016/S1389-0417(03)00007-X

[50] Kuipers, B., "Representing knowledge of Large-Scale Space," *MIT Artificial Intelligence Laboratory*, Technical Report, AI-TR-418, Cambridge, MA, July 1977. (Doctoral Thesis. MIT Mathematics Department)

[51] Mataric, M. J., "Environment Learning Using a Distributed Representation," *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, Vol. 1, Cincinnati, Ohio, 13-18 May, 1990, pp. 402–406. doi: 10.1109/ROBOT.1990.126009

[52] Mataric, M. J., "Integration of Representation into Goal-Driven Behavior-Based Robots," *IEEE Transactions on Robotics and Automation*, Vol. 8, No. 3, June 1992, pp. 304–312. doi: 10.1109/70.143349

[53] Thrun, S., "Learning Metric-Topological Maps for Indoor Mobile Robot Navigation," *Artificial Intelligence*, Vol. 99, No. 1, February 1998, pp. 21–71. doi: 10.1016/S0004-3702(97)00078-7

[54] Ranganathan, A. and Dellaert, F. "Online Probabilistic Topological Mapping," *The International Journal of Robotics Research,*, Vol. 30, No. 6, 2011, pp. 755–771. doi: 10.1177/0278364910393287

[55] Tuskin, A., "The Nature of the Operator's Response in Manual Control, and its Implications for Controller Design," *Journal of the Institution of Electrical Engineers - Part IIA: Automatic Regulators and Servo Mechanisms,*, Vol. 94, No. 2, 1947, pp. 190–206. doi: 10.1049/ji-2a.1947.0025

[56] McRuer, D. T. and Krendel, E. S., "The Human Operator as a Servo System Element," *Journal of the Franklin Institute*, Vol. 267, No. 5, 1959, 381–403. doi: 10.1016/0016-0032(59)90091-2

[57] Bekey, G. A., Burnham, G. O., and Seo, J., "Control Theoretic Models of Human Drivers in Car Following," *Human Factors The Journal of the Human Factors and Ergonomics Society*, Vol. 19, No. 4, 1977, pp. 399–413. doi: 10.1177/001872087701900407

[58] McRuer, D. T. and Jex, H.R., " A Review of Quasi-Linear Pilot Models," *IEEE Transactions on Human Factors in Electronics*, Vol. HFE-8, No. 3, 1967, 231–249. doi: 10.1109/THFE.1967.234304

[59] McRuer, D. T. and Krendel, E. S., "Mathematical Models of Human Pilot Behavior," *DTIC Document, Techical Report*, 1974.

[60] McRuer, D. T., Allen, R. W., Weir, D. H., and Klein, R. H., "New Results in Driver Steering Control Models," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, Vol. 19, No. 4, 1977, pp. 381–397. doi: 10.1177/001872087701900406

[61] Kleinman, D. L., Baron, S., and Levison, W. H., "An Optimal Control Model of Human Response Part I: Theory and Validation," *Automatica*, Vol. 6, No. 3, 1970, pp. 357–369. doi: 10.1016/0005-1098(70)90051-8

[62] Hess, R. A., "Structural Model of the Adaptive Human Pilot," *Journal of Guidance, Control, and Dynamics*, Vol. 3, No. 5, 1980, pp. 416–423. http://dx.doi.org/10.2514/3.56015

[63] Hess, R. A., "Pursuit Tracking and Higher Levels of Skill Development in the Human Pilot," *Systems, Man, and Cybernetics, IEEE Transactions on*, Vol. 11, No. 4, April 1981, pp. 262–273. doi: 10.1109/TSMC.1981.4308673

[64] Hess, R. A. and Modjtahedzadeh, A., "A Control Theoretic Model of Driver Steering Behavior," *IEEE Control Systems Magazine*, Vol. 10, No. 5, August 1990, pp. 3–8. doi: 10.1109/37.60415

[65] Zeyada, Y. and Hess, R., "Modeling Human Pilot Cue Utilization with Applications to Simulator Fidelity Assessment," *Journal of aircraft*, Vol. 37, No. 4, 2000, pp. 588–597. doi: 10.2514/2.2670

[66] Hess, R. A., "A Preliminary Study of Human Pilot Dynamics in the Control of Time-Varying Systems," *AIAA Modeling and Simulation Technologies Conference*, 2011. doi: 10.2514/6.2011-6554

[67] Johannsen, G. and Rouse, W. B., "Mathematical Concepts for Modeling Human Behavior in Complex Man-Machine Systems," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, Vol. 21, No. 6, 1979, pp. 733–747. doi: 10.1177/001872087912210610

[68] Pentland, A. and Liu, A., "Modeling and Prediction of Human Behavior," *Neural Computation*, Vol. 11, No. 1, 1999, pp. 229–242. doi: 10.1162/089976699300016890

[69] Oliver, N. and Pentland, A. P., "Graphical Models for Driver Behavior Recognition in a SmartCar," *In Proceedings of the IEEE Intelligent Vehicles Symposium 2000*, Dearborn, MI, October 3–5, 2000, pp. 7–12. doi: 10.1109/IVS.2000.898310

[70] Liu, A. and Salvucci, D., "Modeling and Prediction of Human Driver Behavior," *In Proceedings of the 9th International Conference on Human-Machine Interactions*, New Orleans, LA, August 5–10, 2001, pp. 1479–1483.

[71] Suzuki, S., Sakamoto, Y., Sanematsu, Y., and Takahara, H., "Analysis of Human Pilot Control Inputs Using Neural Network," *Journal of aircraft*, Vol. 43, No. 3, 2006, pp. 793–798.

[72] Salvucci, D., Boer, E., and Liu, A., "Toward an Integrated Model of Driver Behavior in Cognitive Architecture," *Transportation Research Record: Journal of the Transportation Research Board*, Vol. 1779, 2001, pp. 9–16. doi: 10.3141/1779-02

[73] Bellet, T. and Tattegrain-Veste, H., "A Framework for Representing Driving Knowledge," *International Journal of Cognitive Ergonomics*, Vol. 3, No. 1, 1999, pp. 37–49. doi: 10.1207/s15327566ijce0301_3

[74] Song, B., Delorme, D., and VanderWerf, J., "Cognitive and Hybrid Model of Human Driver," *In Proceedings of the IEEE Intelligent Vehicles Symposium 2000*, Dearborn, MI, October 3–5, 2000, pp. 1–6. doi: 10.1109/IVS.2000.898309

[75] Walker, G. H., Stanton, N. A., and Young, M. S., "An On-Road Investigation of Vehicle Feedback and Its Role in Driver Cognition: Implications for Cognitive Ergonomics," *International Journal of Cognitive Ergonomics*, Vol. 5, No. 4, 2001, pp. 421–444. doi: 10.1207/S15327566IJCE0504_4

[76] Macadam, C. C., "Understanding and Modeling the Human Driver," *Vehicle System Dynamics*, Vol. 40, No. 1-3, 2003, pp. 101–134. doi: 10.1076/vesd.40.1.101.15875

[77] Bellet, T., Bailly-Asuni, B., Mayenobe, P., and Banet, A., "A Theoretical and Methodological Framework for Studying and Modelling Drivers' Mental Representations," *Safety Science*, Vol. 47, No. 9, November 2009, pp. 1205–1221. doi: 10.1016/j.ssci.2009.03.014

[78] Fajen, B. R., Warren, W. H., Temizer, S., and Kaelbling, L. P., "A Dynamical Model of Visually-Guided Steering, Obstacle Avoidance, and Route Selection," *International Journal of Computer Vision*, Vol. 54, No. 1, 2003, pp. 13–34. doi: 10.1023/A:1023701300169

[79] Fajen, B. R. and Warren, W. H., "Behavioral Dynamics of Steering, Obstable Avoidance, and Route Selection.," *Journal of Experimental Psychology: Human Perception and Performance*, Vol. 29, No. 2, 2003, pp. 343–362. doi: 10.1037/0096-1523.29.2.343

[80] Khatib, O., "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," *The International Journal of Robotics Research*, Vol. 5, No. 1, 1986, pp. 90–98. doi: 10.1177/027836498600500106

[81] Patla, A. E., Tomescu, S. S., and Ishac, M. G. A., "What Visual Information is Used for Navigation Around Obstacles in a Cluttered Environment?," *Canadian Journal of Physiology and Pharmacology*, Vol. 82, No. 8-9, 2004, pp. 682–692. doi: 10.1139/y04-058

[82] Hamner, B., Singh, S., and Scherer, S., "Learning Obstacle Avoidance Parameters from Operator Behavior," *Special Issue on Machine Learning Based Robotics in*

Unstructured Environments, Journal of Field Robotics, Vol. 23, No. 11-12, December 2006, pp. 1037–1058. doi: 10.1002/rob.20171

[83] Huang, H., Chamberlain, L., and Murray, R. M., "An Experimental Platform for Motion Estimation and Maneuver Characterization in High Speed Off-Road Driving," *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, April 18–22, 2005, pp. 3090–3095. doi: 10.1109/ROBOT.2005.1570585

[84] Burns, C. R., Wang, R. F., and Stipanovi, D. M., "A Study of Human and Receding Horizon Controller Performance of a Remote Navigation Task with Obstacles and Feedback Delays," *PALADYN Journal of Behavioral Robotics*, Vol. 2, No. 1, 2011, pp. 44–63. doi: 10.2478/s13230-011-0015-7

[85] Rosenbaum, D. A., "Human Motor control (Second Edition)," *Academic press*, 2009, pp. 161–168.

[86] Witt, J. K., "Actions Effect on Perception," *Current Directions in Psychological Science*, Vol. 20, No. 3, 2011, pp. 201–206. doi: 10.1177/0963721411408770

[87] Lee, D. N. and Aronson, E., "Visual Proprioceptive Control of Standing in Human Infants," *Perception & Psychophysics*, Vol. 15, No. 3, 1974, pp. 529-532. doi: 10.3758/BF03199297

[88] Lee, D. N. and Lishman, J. R., "Visual Proprioceptive Control of Stance," *Journal of Human Movement Studies*, Vol. 1, No. 2, 1975, pp. 87–95.

[89] Gibson, J. J., "The Perception of the Visual World," *Houghton Mifflin*, 1950.

[90] Thomson, J. A., "How Do We Use Visual Information to Control Locomotion?," *Trends in Neurosciences*, Vol. 3, No. 10, October 1980, pp. 247–250. doi: 10.1016/S0166-2236(80)80076-2

[91] Thomson, J. A., "Is Continuous Visual Monitoring Necessary in Visually Guided Locomotion?," *Journal of Experimental Psychology: Human Perception and Performance*, Vol. 9, No. 3, 1983, pp. 427–443. doi: 10.1037/0096-1523.9.3.427

[92] Jeannerod, M. and Prablanc., C., in "Du Controle Moteur a l' Organisation du Geste," *(Hecaen, H. and Jeannerod, M., eds)*, Masson, Paris, 1978, pp. 261–289.

[93] Sharp, R. H. and Whiting, H. T. A., "Exposure and Occluded Duration Effects in a Ball-Catching Skill," *Journal of Motor Behavior*, Vol. 6, No. 3, 1974, pp. 139–147. doi: 10.1080/00222895.1974.10734990

[94] Andersen, G. J. and Sauer, C. W., "Optical Information for Car Following: The Driving by Visual Angle (DVA) Model," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, Vol. 49, No. 5, 2007, pp. 878–896. doi: 10.1518/001872007X230235

[95] Land, M. F. and Lee, D. N., "Where We Look When We Steer," *Nature*, Vol. 369, No. 6483, 1994, pp. 742–744. doi: 10.1038/369742a0

[96] Kandil, F. I., Rotter, A., and Lappe, M., "Driving is Smoother and More Stable When Using the Tangent Point," *Journal of Vision*, Vol. 9, No. 1, 2009, pp. 1–11. doi: 10.1167/9.1.11

[97] Mars, F., "Driving Around Bends with Manipulated Eye-Steering Coordination," *Journal of Vision*, Vol. 8, No. 11, 2008, pp. 1–11. doi: 10.1167/8.11.10

[98] Wilkie, R. M., Kountouriotis, G. K., Merat, N., and Wann, J. P., "Using Vision to Control Locomotion: Looking Where You Want to Go," *Experimental Brain Research*, Vol. 204, No. 4, 2010, pp. 539–547. doi: 10.1007/s00221-010-2321-4

[99] Salvucci, D. D and Gray, R., "A Two-Point Visual Control Model of Steering," *Perception*, Vol. 33, No. 10, 2004, pp. 1233-1248. doi: 10.1068/p5343

[100] Neumann, H. and Deml, B., "The Two-Point Visual Control Model of Steering-New Empirical Evidence," *Springer Berlin Heidelberg*, In V. G. Duffy (Ed.), Digital Human Modeling, 2011, pp. 493–502. doi: 10.1007/978-3-642-21799-9_55

[101] Vansteenkiste, P., Cardon, G., D'Hondt, E., Philippaerts, R., and Lenoir, M., "The Visual Control of Bicycle Steering: The Effects of Speed and Path Width.," *Accident Analysis & Prevention*, Vol. 51, 2013, pp. 222–227. doi: 10.1016/j.aap.2012.11.025

[102] Johnson, L., Sullivan, B., Hayhoe, M., and Ballard, D., "Predicting Human Visuomotor Behaviour in a Driving Task," *Philosophical Transactions of the Royal Society B: Biological Sciences*, Vol. 369, No. 1636, 2014. doi: 10.1098/rstb.2013.0044

[103] Kujala, T. and Salvucci, D. D., "Modeling Visual Sampling on In-Car Displays: The Challenge of Predicting Safety-Critical Lapses of Control," *International Journal of Human-Computer Studies*, Vol. 79, 2015, pp. 66–78. doi: 10.1016/j.ijhcs.2015.02.009

[104] Mackenzie, A. K. and Harris, J. M., "Eye Movements and Hazard Perception in Active and Passive Driving," *Visual Cognition*, Vol. 23, No. 6, 2015, pp. 736–757. doi: 10.1080/13506285.2015.1079583

[105] Crundall, D. E. and Underwood, G., "Effects of Experience and Processing Demands on Visual Information Acquisition in Drivers," *Ergonomics*, Vol. 41, No. 4, 1998, pp. 448–458. doi: 10.1080/001401398186937

[106] Underwood, G., Chapman, P., Bowden, K., and Crundall, D., "Visual Search while Driving: Skill and Awareness during Inspection of the Scene," *Transportation Research Part F: Traffic Psychology and Behaviour*, Vol. 5, No. 2, 2002, pp. 87–97. doi: 10.1016/S1369-8478(02)00008-6

[107] Elliott, M. H., "The Effect of Change of Reward on the Maze Performance of Rats," *University California Public Psychology*, Vol. 4, 1929, pp. 19–30.

[108] Tolman, E. C. and Honzik, C. H., "Introduction and Removal of Reward, and Maze Performance in Rats," *University of California Publications in Psychology*, Vol. 4, 1930, pp. 257–275.

[109] Tolman, E. C., "Cognitive Maps in Rats and Men," *Psychological Review*, Vol. 55, No. 4, July 1948, pp. 189–208. doi: 10.1037/h0061626

[110] Olton, D. S. and Samuelson, R. J. "Remembrance of Places Passed: Spatial Memory in Rats," *Journal of Experimental Psychology: Animal Behavior Processes*, Vol. 2, No. 2, 1976, pp. 97–116. doi: 10.1037/0097-7403.2.2.97

[111] Olton, D. S., "Mazes, Maps, and Memory.," *American Psychologist*, Vol. 34, No. 7, 1979, pp. 583–596. doi: 10.1037/0003-066X.34.7.583

[112] Norman, D. A., "Categorization of Action Slips.," *Psychological Review*, Vol. 88, No. 1, 1981, pp. 1–15. doi: 10.1037/0033-295X.88.1.1

[113] Stevens, A. and Coupe, P., "Distortions in Judged Spatial Relations," *Cognitive Psychology*, Vol. 10, No. 4, October 1978, pp. 422–437. doi: 10.1016/0010-0285(78)90006-3

[114] Thorndyke, P. W., "Distance Estimation from Cognitive Maps," *Cognitive Psychology*, Vol. 13, No. 4, October 1981, pp. 526–550. doi: 10.1016/0010-0285(81)90019-0

[115] Hirtle, S. C. and Jonides, J., "Evidence of Hierarchies in Cognitive Maps," *Memory & Cognition*, Vol. 13, No. 3, May 1985, pp. 208–217. doi: 10.3758/BF03197683

[116] McNamara, T. P., "Mental Representations of Spatial Relations," *Cognitive Psychology*, Vol. 18, No. 1, January 1986, pp. 87–121. doi: 10.1016/0010-0285(86)90016-2

[117] Foo, P., Warren, W. H., Duchon, A., and Tarr, M. J., "Do Humans Integrate Routes Into a Cognitive Map? Map-Versus Landmark-Based Navigation of Novel Shortcuts," *Journal of Experimental Psychology: Learning, Memory, and Cognition*, Vol. 31, No. 2, March 2005, pp. 195–215. doi: 10.1037/0278-7393.31.2.195

[118] Christaller, T., "Cognitive Robotics: a New Approach to Artificial Intelligence," *Artificial Life and Robotics*, Vol. 3, No. 4, December 1999, pp. 221–224. doi: 10.1007/BF02481184

[119] Jefferies, M. E. and Yeap, W. K., "Robot and Cognitive Approaches to Spatial Mapping," *Springer Berlin Heidelberg*, Vol. 38, 2008. doi: 10.1007/978-3-540-75388-9

[120] Chakravorty, S. and Junkins J. L., "Motion Planning in Uncertain Environments with Vision-Like Sensors," *Automatica*, Vol. 43, No. 12, December 2007, pp. 2104–2111. doi: 10.1016/j.automatica.2007.04.022

[121] Vasudevan, S., Gchter, S., Nguyen, V., and Siegwart, R., "Cognitive Maps for Mobile Robots-An Object Based Approach," *Robotics and Autonomous Systems*, Vol. 55, No. 5, May 2007, pp. 359–371. doi: 10.1016/j.robot.2006.12.008

[122] Manning, J. R., Lew, T. F., Li, N., Sekuler, R., and Kahana, M. J., "MAGELLAN: A Cognitive Map-Based Model of Human Wayfinding," *Journal of Experimental Psychology: General*, Vol. 143, No. 3, 2014, pp. 1314–1330. doi: 10.1037/a0035542

[123] Chase, W. G., "Spatial Representations of Taxi Drivers," *The Acquisition of Symbolic Skills*, 1983, pp. 391–405. doi: 10.1007/978-1-4613-3724-9_43

[124] Gillner, S. and Mallot, H. A., "Navigation and Acquisition of Spatial Knowledge in a Virtual Maze," *Journal of Cognitive Neuroscience*, Vol. 10, No. 4, July 1998, pp. 445–463. doi: 10.1162/089892998562861

[125] Spiers, H. J. and Maguire, E. A., "The Dynamic Nature of Cognition During Wayfinding," *Journal of Environmental Psychology*, Vol. 28, No. 3, 2008, pp. 232–249. doi: 10.1016/j.jenvp.2008.02.006

[126] Wang, R. F. and Brockmole, J. R, "Human Navigation in Nested Environments," *Journal of Experimental Psychology: Learning, Memory, and Cognition*, Vol. 29, No. 3, 2003, pp. 398–404. doi: 10.1037/0278-7393.29.3.398

[127] Jeffery, K. J., Jovalekic, A., Verriotis, M., and Hayman, R., "Navigating in a Three-Dimensional World," *Behavioral and Brain Sciences*, Vol. 36, No. 5, 2013, pp. 523–543. doi: 10.1017/S0140525X12002476

[128] Gagnon, S. A., Bruny, T. T., Gardony, A., Noordzij, M. L., Mahoney, C. R., and Taylor, H. A. "Stepping Into a Map: Initial Heading Direction Influences Spatial Memory Flexibility," *Cognitive Science*, Vol. 38, 2014, pp. 275–302. doi: 10.1111/cogs.12055

[129] May, M., Peruch, P., and Savoyant, A., "Navigating in a Virtual Environment With Map-Acquired Knowledge: Encoding and Alignment Effects," *Ecological Psychology*, Vol. 7, 1, 1995, pp. 21–36. doi: 10.1207/s15326969eco0701_2

[130] Golledge, R. G., "Path Selection and Route Preference in Human Navigation: A Progress Report," *Spatial Information Theory A Theoretical Basis for GIS, International Conference COSIT '95*, Vol. 988, 1995, pp. 207–222. doi: 10.1007/3-540-60392-1_14

[131] Hochmair, H. and Frank, A. U., "Influence of Estimation Errors on Wayfinding-Decisions in Unknown Street Networks Analyzing the Least-Angle Strategy," *Spatial Cognition and Computation*, Vol. 2, No. 4, December 2000, pp. 283–313. doi: 10.1023/A:1015566423907

[132] Hartley, T., Maguire, E. A., Spiers, H. J., and Burgess, N., "The Well-Worn Route and the Path Less Traveled: Distinct Neural Bases of Route Following and Wayfinding in Humans," *Neuron*, Vol. 37, No. 5, March 2003, pp. 877–888. doi: 10.1016/S0896-6273(03)00095-3

[133] Jueptner, M., Stephan, K. M., Frith, C. D., Brooks, D. J., Frackowiak, R. S. J., and Passingham, R. E., "Anatomy of Motor Learning. I. Frontal Cortex and Attention to Action," *Journal of Neurophysiology*, Vol. 77, No. 3, 1997, pp. 1313–1324.

[134] Jueptner, M., Frith, C. D., Brooks, D. J., Frackowiak, R. S. J., and Passingham, R. E., "Anatomy of Motor Learning. II. Subcortical Structures and Learning by Trial and Error," *Journal of Neurophysiology*, Vol. 77, No. 3, 1997, pp. 1325–1337.

[135] Darken, R. P. and Sibert, J. L., "Navigating Large Virtual Spaces," *International Journal of Human-Computer Interaction*, Vol. 8, No. 1, 1996, pp. 49–71. doi: 10.1080/10447319609526140

[136] Ruddle, R. A., Payne, S. J., and Jones, D. M., "Navigating Buildings in "Desk-Top" Virtual Environments: Experimental Investigations Using Extended Navigational Experience," *Journal of Experimental Psychology: Applied*, Vol. 3, No. 2, 1997, pp. 143–159. doi: 10.1037//1076-898X.3.2.143

[137] Waller, D., Loomis, J. M., Golledge, R. G., and Beall, A. C., "Place Learning in Humans: The Role of Distance and Direction Information," *Spatial*

Cognition and Computation, Vol. 2, No. 4, December 2000, pp. 333–354. doi: 10.1023/A:1015514424931

[138] Kato, Y. and Takeuchi, Y., "Individual Differences in Wayfinding Strategies," *Journal of Environmental Psychology*, Vol. 23, No. 2, June 2003, pp. 171–188. doi: 10.1016/S0272-4944(03)00011-2

[139] Kelly, J. W., McNamara, T. P., Bodenheimer, B., Carr, T. H., and Rieser, J. J., "The Shape of Human Navigation: How Environmental Geometry is Used in Maintenance of Spatial Orientation," *Cognition*, Vol. 109, No. 2, 2008, pp. 281–286. doi: 10.1016/j.cognition.2008.09.001

[140] Vilar, E., Rebelo, F., and Noriega, P., "Indoor Human Wayfinding Performance Using Vertical and Horizontal Signage in Virtual Reality," *Human Factors and Ergonomics in Manufacturing and Service Industries*, Vol. 24, No. 6, 2014, pp. 601–615. doi: 10.1002/hfm.20503

[141] Bailenson, J. N., Shum, M. S., and Uttal, D. H., "Road Climbing: Principles Governing Asymmetric Route Choices on Maps," *Journal of Environmental Psychology*, Vol. 18, No. 3, 1998, pp. 251–264. doi: 10.1006/jevp.1998.0095

[142] Bailenson, J. N., Shum, M. S., and Uttal, D. H., "The Initial Segment Strategy: A Heuristic for Route Selection," *Memory & Cognition*, Vol. 28, No. 2, 2000, pp. 306–318. doi: 10.3758/BF03213808

[143] Bruny, T. T., Mahoney, C. R., Gardony, A. L., and Taylor, H. A., "North is Up(Hill): Route Planning Heuristics in Real-World Environments.," *Memory & Cognition*, Vol. 38, No. 6, 2010, pp. 700–712. doi: 10.3758/MC.38.6.700.

[144] Bruny, T. T., Collier, Z. A., Cantelon, J., Holmes, A., Wood, M. D., Linkov, I., and Taylor, H. A., "Strategies for Selecting Routes through Real-World Environments: Relative Topography, Initial Route Straightness, and Cardinal Direction.," *PLoS One*, Vol. 10, No. 5, 2015. doi: 10.1371/journal.pone.0124404

[145] Vreeswijk, J., Thomas, T., Berkum, E. V., and Arem, B. V., "Drivers' Perception of Route Alternatives as Indicator for the Indifference Band.," *Transportation*

Research Record: Journal of the Transportation Research Board, Vol. 2383, 2013, pp. 10–17.

[146] Simon, H. A., "Models of Man," *New York, Wiley & Sons*, 1957.

[147] Simon, H. A., "Rational Choice and the Structure of the Environment," *Psychological Review*, Vol. 63, No. 2, March 1956, pp. 129–138. doi: 10.1037/h0042769

[148] Cowan, N., "Evolving Conceptions of Memory Storage, Selective Attention, and Their Mutual Constraints within the Human Information-Processing System," *Psychological Bulletin*, Vol. 104, No. 2, September 1988, pp. 163–191. doi: 10.1037/0033-2909.104.2.163

[149] Knudsen, E. I., "Fundamental Components of Attention," *Annual Review of Neuroscience*, Vol. 30, July 2007, pp. 57–78. doi:10.1146/annurev.neuro.30.051606.094256

[150] Miller, G. A., "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information," *Psychological Review*, Vol. 63, No. 2, March 1956, pp. 81–97. doi:10.1037/h0043158

[151] Chase, W. G. and Simon, H., "Perception in chess," *Cognitive Psychology*, Vol. 4, No. 1, January 1973, pp. 55–81. doi:10.1016/0010-0285(73)90004-2

[152] Land, A. H. and Doig, A. G., "An Automatic Method for Solving Discrete Programming Problems," *ECONOMETRICA*, Vol. 28, No. 3, 1960, pp. 497–520.

[153] Huys, Q. J. M., Eshel, N., O'Nions, E., Sheridan, L., Dayan, P., and Roiser, J. P., "Bonsai Trees in Your Head: How the Pavlovian System Sculpts Goal-Directed Choices by Pruning Decision Trees," *PLOS Computational Biology*, Vol. 8, No. 3, March 2012. doi: 10.1371/journal.pcbi.1002410

[154] Endsley, M. R., "Toward a Theory of Situation Awareness in Dynamic Systems," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, Vol. 37, No. 1, March 1995, pp. 32–64. doi: 10.1518/001872095779049543

[155] Summerfield, C. and Tsetsos, K., "Building Bridges between Perceptual and Economic Decision-Making: Neural and Computational Mechanisms," *Frontiers in Neuroscience*, Vol. 6, 2012. doi: 10.3389/fnins.2012.00070

[156] Towal, R. B., Mormann, M., and Koch, C., "Simultaneous Modeling of Visual Saliency and Value Computation Improves Predictions of Economic Choice," *Proceedings of the National Academy of Sciences*, Vol. 110, No. 40, 2013, pp. E3858–E3867. doi: 10.1073/pnas.1304429110

[157] Shimojo, S., Simion, C., Shimojo, E., and Scheier, C., "Gaze Bias Both Reflects and Influences Preference," *Nature Neuroscience*, Vol. 6, No. 12, December 2003, pp. 1317–1322. doi: 10.1038/nn1150

[158] Krajbich, I., Armel, C., and Rangel, A., "Visual Fixations and the Computation and Comparison of Value in Simple Choice," *Nature Neuroscience*, Vol. 13, 2010, pp. 1292–1298. doi: 10.1038/nn.2635

[159] Krajbich, I. and Rangel, A., "Multialternative Drift-Diffusion Model Predicts the Relationship Between Visual Fixations and Choice in Value-Based Decisions," *Proceedings of the National Academy of Sciences*, Vol. 108, No. 33, August 2011, pp. 13852–13857. doi: 10.1073/pnas.1101328108

[160] Sakellaridi, S., Christova, P., Christopoulos, V. N., Vialard, A., Peponis, J., and Georgopoulos, A. P., "Cognitive Mechanisms Underlying Instructed Choice Exploration of Small City Maps," *Frontiers in Neuroscience*, Vol. 9, No. 60, March 2015. doi: 10.3389/fnins.2015.00060

[161] March, J. G., "Exploration and Exploitation in Organizational Learning," *Organization Science*, Vol. 2, No. 1, February 1991, pp. 71–87. doi: 10.1287/orsc.2.1.71

[162] Cohen, J. D., McClure, S. M., and Yu, A. J., "Should I Stay or Should I Go? How the Human Brain Manages the Trade-Off Between Exploitation and Exploration," *Philosophical Transactions of The Royal Society B*, Vol. 362, No. 1481, 2007, pp. 933–942. doi: 10.1098/rstb.2007.2098

[163] Gittins, J.C. and Jones, D.M., "A Dynamic Allocation Index for the Sequential Design of Experiments," *In Progress in Statisics (ed. J. Gani)*, Amsterdam, The Netherlands: North-Holland, 1974, pp. 241–266.

[164] Gittins, J. C., "Bandit Processes and Dynamic Allocation Indices," *Journal of the Royal Statistical Society. Series B (Methodological)*, Vol. 41, No. 2, 1979, pp. 148–177.

[165] Lee, M. D., Zhang, S., Munro, M., and Steyvers, M., "Psychological Models of Human and Optimal Performance in Bandit Problems," *Cognitive Systems Research*, Vol. 12, No. 2, June 2011, pp. 164–174. doi: 10.1016/j.cogsys.2010.07.007

[166] Auer, P., "Using Confidence Bounds for Exploitation-Exploration Trade-offs," *Journal of Machine Learning Research*, Vol. 3, No. 3, 2002, pp. 397–422.

[167] Reverdy, P., Wilson, R. C., Holmes, P., and Leonard, N. E., "Towards Optimization of a Human-Inspired Heuristic for Solving Explore-Exploit Problems," *Proceedings of the IEEE Conference on Decision and Control*, Maui, HI, 2012, pp. 2820-2825.

[168] Mettler, B., "Structure and Organizational Principles of Agile Behavior: Challenges and Opportunities in Cognitive Engineering," *Journal of Cognitive Critique*, Vol. 3, 2011.

[169] Frazzoli, E., Dahleh, M. A., and Feron, E., "Real-Time Motion Planning for Agile Autonomous Vehicles," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 1, 2002, pp. 116–129.

[170] Bellman, R. E. and Dreyfus, S. E., "Applied Dynamic Programming," *RAND Corporation*, 1962.

[171] Sutton, R. S. and Barto, A. G., "Reinforcement Learning: An Introduction," *MIT press*, Cambridge, MA, 1998.

[172] Marlow, S. Q. and Langelaan, J. W., "Local Terrain Mapping for Obstacle Avoidance Using Monocular Vision," *Journal of the American Helicopter Society*, Vol. 56, No. 2, April 2011, pp. 22007-1-22007-14. doi: 10.4050/JAHS.56.022007

[173] Verma, A. and Mettler, B., "Scaling Effects in Guidance Performance in Confined Environments," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 7, 2016, pp. 1527-1538. doi: 10.2514/1.G001415

[174] Dadkhah, N. and Mettler, B., "System Identification Modeling and Flight Characteristics Analysis of Miniature Coaxial Helicopter," *Journal of the American Helicopter Society*, Vol. 59, No. 4, October 2014, pp. 36–51. doi: 10.4050/JAHS.59.042011

[175] IBM ILOG CPLEX Optimizer, http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/

[176] Mettler, B., Kong, Z., Goerzen, C., and Whalley, M., "Benchmarking of Obstacle Field Navigation Algorithms for Autonomous Helicopters," *Annual Forum Proceedings - American Helicopter Society International*, Vol. 3, Phoenix, Arizona, May 2010, pp. 1936–1953.

[177] Dubins, L. E., "On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents," *American Journal of Mathematics*, Vol. 79, No. 3, July 1957, pp. 497–516. doi: 10.2307/2372560

[178] Schouwenaars, T., How, J., and Feron, E., "Receding Horizon Path Planning with Implicit Safety Guarantees," *American Control Conference, 2004. Proceedings of the 2004*, Vol. 6, 2004, pp. 5576–5581.

[179] Li, B. and Mettler, B., "Investigation of Hierarchical Architecture of Human Guidance Behavior for Skill Analysis," *Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on*, HongKong, Oct. 9–12, 2015, pp. 1081–1088. doi: 10.1109/SMC.2015.194

[180] Feit, A., Verma, A., and Mettler, B., "A Human-Inspired Subgoal-Based Approach to Constrained Optimal Control," *IEEE CDC 2015, IEEE Conference on Decision and Control*, Osaka, Japan, Dec. 15–18, 2015, pp. 1289–1296. doi: 10.1109/CDC.2015.7402389

[181] Verma, A., Feit, A., and Mettler, B., "Investigation of Human First-Person Guidance Strategy from Gaze Tracking Data," *Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on*, HongKong, Oct. 9–12, 2015, pp. 1066–1072. doi: 10.1109/SMC.2015.192

[182] Verma, A. and Mettler, B., "Human Learning of Unknown Environments in Agile Guidance Tasks," 2017, arXiv:1710.07757.

[183] Verma, A. and Mettler, B., "Analysis of Human Guidance and Perceptual Behavior in Navigation of Unknown Environments," *Conference: American Helicopter Society Forum 72*, Florida, 2016.

[184] LaValle, S. M., "Sampling-based Motion Planning. Planning Algorithms," *New York: Cambridge University Press*, 2006.

[185] Mataric, M. J., "A Distributed Model for Mobile Robot Environment-Learning and Navigation," *Master's Thesis, Cambridge, MA*, 1990.

[186] Lumelsky, V. J. and Skewis, T., "Incorporating Range Sensing in the Robot Navigation Function," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 20, No. 5, 1990, pp. 1058–1069. doi: 10.1109/21.59969

[187] Wei, S. and Zefran, M., "Smooth Path Planning and Control for Mobile Robots," *Networking, Sensing and Control, 2005. Proceedings. 2005 IEEE*, 2005, pp. 894–899. doi: 10.1109/ICNSC.2005.1461311

[188] Padfield, G. D., "The Tau of Flight Control," *The Aeronautical Journal*, Vol. 115, No. 1171, 2011, pp. 521–556. doi: 10.1017/S0001924000006187

[189] Overmars, M. H., "A Random Approach to Motion Planning," *Technical Report RUU-CS-92-32*, Department of Computer Science, Utrecht university, Utrecht, The Netherlands, 1992 .

[190] Kavraki, L. E., Kolountzakis, M. N., and Latombe, J.-C., "Analysis of Probabilistic Roadmaps for Path Planning," *IEEE Transactions on Robotics and Automation*, Vol. 14, No. 1, February 1998, pp. 166–171. doi: 10.1109/70.660866

[191] Kavraki, L. E. and Latombe, J.-C., "Probabilistic Roadmaps for Path Planning," *in Practical Motion Planning in Robotics: Current Approaches and Future Directions, John Wiley*, 1998, pp. 33–53.

[192] Rantanen, M. T. and Juhola, M., "Using Probabilistic Roadmaps in Changing Environments," *Computational Animation & Virtual Worlds*, published online 18 June 2013; Vol. 25, 2014, pp. 17–31. doi: 10.1002/cav.1528

[193] Lozano-Prez, T. and Wesley, M. A., "An Algorithm for Planning Collision-Free Paths among Polyhedral Obstacles," *Communications of the ACM*, Vol. 22, No. 10, 1979, pp. 560–570. doi: 10.1145/359156.359164

[194] Oommen, B., Iyengar, S., Rao, N., and Kashyap, R., "Robot Navigation in Unknown Terrains using Learned Visibility Graphs. Part I: The Disjoint Convex Obstacle Case," *IEEE Journal on Robotics and Automation*, Vol. 3, No. 6, 1987, pp. 672–681. doi: 10.1109/JRA.1987.1087133

[195] Huang, H.-P. and Chung, S.-Y., " Dynamic Visibility Graph for Path Planning," *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, Vol. 3, 2004, pp. 2813–2818. doi: 10.1109/IROS.2004.1389835

[196] Rashid, A. T., Ali, A. A., Frasca, M., and Fortuna, L., "Path Planning with Obstacle Avoidance based on Visibility Binary Tree Algorithm," *Robotics and Autonomous Systems*, Vol. 61, No. 12, 2013, pp. 1440–1449. doi: 10.1016/j.robot.2013.07.010

[197] Choset, H., Walker, S., Eiamsa-Ard, K., and Burdick, J., "Sensor-Based Exploration: Incremental Construction of the Hierarchical Generalized Voronoi Graph," *The International Journal of Robotics Research*, Vol. 19, No. 2, 2000, pp. 126–148. doi: 10.1177/02783640022066789

[198] Bhattacharya, P. and Gavrilova, M. L., "Voronoi Diagram in Optimal Path Planning," *In Voronoi Diagrams in Science and Engineering, 2007. ISVD'07. 4th International Symposium on*, 2007, pp. 38–47. doi: 10.1109/ISVD.2007.43

[199] Sud, A., Andersen, E., Curtis, S., Lin, M., and Manocha, D., "Real-Time Path Planning in Dynamic Virtual Environments using Multiagent Navigation Graphs.," *IEEE Transactions on Visualization and Computer Graphics*, Vol. 14, No. 3, 2008, pp. 526–538. doi: 10.1109/TVCG.2008.27

[200] Kim, J., Zhang, F., and Egerstedt, M., "A Provably Complete Exploration Strategy by Constructing Voronoi Diagrams," *Autonomous Robots*, Vol. 29, No. 3, 2010, pp. 367–380. doi: 10.1007/s10514-010-9200-5

[201] TAN, P. and CAI, Z., "Modelling and Planning of Mobile Robot Navigation Control in Unknown Environment," *Computational Intelligence and Communication Networks (CICN), 2015 International Conference on*, 2015, pp. 1532–1536. doi: 10.1109/CICN.2015.292

[202] Li, B., Mettler, B., and Kowalewski, T., "Towards Data-driven Hierarchical Surgical Skill Analysis," *in 5th Annual Workshop on Modeling and Monitoring of Computer Assisted Interventions*, Boston, MA, 2014.

Appendix A

Appendix

A.1 Autonomous Guidance Simulation (Chapter 4)

Following are the parameters for simulations presented in Chapter 4.
Matrices A and B in Eq. (3.11) are (see [174] for details):

$$A = \begin{bmatrix} -0.0561 & -2.3351 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.0649 & 0.6445 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.0099 & 0.1715 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.0561 & -2.3351 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0649 & 0.6445 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0099 & 0.1715 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0.1377 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.4493 \end{bmatrix} \tag{A.1}$$

$$B = \begin{bmatrix} 2.3351 & 0 & 0 \\ 0.3555 & 0 & 0 \\ 0.0284 & 0 & 0 \\ 0 & 2.3351 & 0 \\ 0 & 0.3555 & 0 \\ 0 & 0.0284 & 0 \\ 0 & 0 & 0.0623 \\ 0 & 0 & 0.5507 \end{bmatrix} \tag{A.2}$$

Horizontal and vertical speed levels for motion primitives used in Chapter 4 are:

$$\begin{aligned} n_v = 6 \;\; ; \;\; v_l &= [0.61 \;\; 0.87 \;\; 1.06 \;\; 1.22 \;\; 1.37 \;\; 1.50] \text{ m/s} \\ n_{vz} = 2 \;\; ; \;\; v_{zl} &= [-0.5 \;\; -0.25 \;\; 0 \;\; 0.25 \;\; 0.5] \text{ m/s} \end{aligned} \tag{A.3}$$