Human Guidance Behavior Decomposition and Modeling

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

Andrew James Feit

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Advisor: Bérénice Mettler

DECEMBER, 2017

Acknowledgements

There are many people that have earned my gratitude for their contribution to my time in graduate school. I would like to specifically acknowledge my fellow graduate students in the interactive guidance and control lab: Abhishek Verma, Bin Li, Kuo-Shih Tseng, Jon Andersh, and Monique Hladun. They have provided feedback, lab infrastructure, and company during my time in this program without which I could not have completed this work. My advisor, Dr. Bérénice Mettler for allowing me to take my own research directions, providing direct feedback, and working to help me become a better thinker through better writing. In addition, this dissertation would not have been possible without funding from the National Science Foundation and the Office of Naval Research. Finally, I thank my family and my friends, including those in the Minnesota rock climbing community, for being my support structure and for being always ready for adventure.

Abstract

Trained humans are capable of high performance, adaptable, and robust first-person dynamic motion guidance behavior. This behavior is exhibited in a wide variety of activities such as driving, piloting aircraft, skiing, biking, and many others. Human performance in such activities far exceeds the current capability of autonomous systems in terms of adaptability to new tasks, real-time motion planning, robustness, and trading safety for performance. The present work investigates the structure of human dynamic motion guidance that enables these performance qualities. This work uses a first-person experimental framework that presents a driving task to the subject, measuring control inputs, vehicle motion, and operator visual gaze movement. The resulting data is decomposed into subspace segment clusters that form primitive elements of action-perception interactive behavior. Subspace clusters are defined by both agent-environment system dynamic constraints and operator control strategies. A key contribution of this work is to define transitions between subspace cluster segments, or subgoals, as points where the set of active constraints, either system or operator defined, changes. This definition provides necessary conditions to determine transition points for a given task-environment scenario that allow a solution trajectory to be planned from known behavior elements. In addition, human gaze behavior during this task contains predictive behavior elements, indicating that the identified control modes are internally modeled. Based on these ideas, a generative, autonomous guidance framework is introduced that efficiently generates optimal dynamic motion behavior in new tasks. The new subgoal planning algorithm is shown to generate solutions to certain tasks more quickly than existing approaches currently used in robotics.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Humans and animals have a natural ability for motion guidance because it is our primary method of interacting with the world. Despite having this intuitive ability, we don't yet have a complete understanding of the computational processes occurring in human brains that enable these behaviors. Human guidance includes the motion of one's own body, as well as controlling vehicles from a first-person perspective such as driving a car or flying an aircraft. In addition, we use these skills to control vehicles or other systems such as remote-control vehicles or robotic arms from a third-person perspective. While performing these motion tasks in competitive or safety-critical scenarios, humans often exhibit high performance in terms of travel-time, speed, or effort. They exhibit versatility and adaptability to a wide range of task requirements, and robustness to uncertainty in system dynamics and constraints. Throughout these tasks, humans maintain safety by trading risk vs. performance in response to specific task requirements. Investigating these characteristics of human performance is important for understanding human cognitive processes and how machines can be designed to best interact with human operators.

Human brains process large amounts of information, but they achieve this high performance despite having physical and computational limitations. For example, humans have limited working memory, restricting the planning search space and the number of environment objects that can be considered simultaneously [1]. In addition, senses,

such as vision, proprioception, and inertial by the vestibular system provide only partial observability of the agent-environment state. Humans must integrate sensory inputs to gain situational awareness [2]. Finally, humans are subject to similar information processing constraints as artificial systems when approaching a planning task. For example, humans do not posses a shortcut to solve NP-complete combinatorial optimization problems. Despite these limitation, athletes and pilots achieve performance significantly beyond the capabilities of current autonomous or robotic systems. This performance difference suggests that computational guidance systems can attain significant improvements by employing human-inspired strategies.

Recent research in artificial intelligence has shown that deep learning and deep neural networks (DNN) generate promising results in learning complex tasks such speech and character recognition, medical diagnosis [3], and autonomous driving [4]. It is known that DNNs are able to achieve high-performance using relatively few training examples by capturing the structure in a learning problem [5]. A remaining challenge is that the resulting structural elements cannot be extracted from the DNN in order to communicate how it works to human operators, and to verify that it will produce correct outputs in response to a specified input domain. Recent work by Kong and Mettler have taken a systems-based, embedded-agent approach to investigating human motion guidance tasks. [6, 7, 8, 9]. This approach provides a method for identifying the structural elements humans use to generate motion, and is the principle background for the present research. The present work identifies structure in observed human perception-action behavior, and defines models that take advantage of this structure to generate behavior in guidance tasks. Understanding these processes that allow humans to learn and use task structure would provide a significant advancement for autonomous vehicle performance and human-machine system interaction.

1.2 General Background

Two ideas in human cognition are used to approach the guidance problem. First, Simon introduced the concept of satisficing [10] to describe behavior that may not be optimal with respect to a utility function, but satisfies constraints. This idea suggests that humans simplify the planning and guidance task to obtain sub-optimal, constraint

satisfying solutions efficiently. Simon identifies three characteristics of tasks that make optimization difficult. First, the reward for taking specific actions may be uncertain. Second, the set of available actions may not be fully known at each instant in time, and third, computation of the true cost or value of taking an action may be too complex. As a result, humans must employ heuristics, prune the decision tree, or approximate the true value function to make decisions efficiently. The second idea Simon presented is that, as in the study of other physical phenomenon, invariants in the agent-environment system are key to understanding human behavior [11]. Examples of invariants include exact numerical quantities, such as conservation of energy, or qualitative ideas such as the germ theory of disease stating that many illnesses are caused by a microorganism. Mettler and Kong extend the concept of invariants to motion control in two ways. First, they present experimental results showing that humans exploit invariants in system dynamics and constraints to reduce problem complexity [12]. Invariants in guidance are equivalence classes across the task domain that decompose a problem into a smaller set of common subtasks. Mettler and Kong also extend the concept of invariants along with the concept of causal state from computational mechanics [13] to investigate structural patterns in observed human behavior. Interaction patterns [14, 12, 8, 6] describe action-perception relationships that generate common elements of behavior across multiple parts of a task-environment domain. The identification of common subtasks is a form of structure learning [15] that simplifies the decision domain.

From a control theory perspective, guidance behavior can be defined by a value function and guidance policy. [12] and [16] show that human guidance behavior is also described by value and policy functions, which define a spatial invariant of motion. [17] and [6] suggest that humans generate this motion by deploying sequences of sensory-motor interaction patterns. Interaction patterns extend the motion-primitive automaton concept introduced by [18] by introducing guidance primitives. Guidance primitives are discrete agent-environment interaction elements, and define perceptual-dynamic invariants of behavior. An agent can deploy a series of guidance primitives chosen from a library of learned elements. Each element optimizes closed-loop perception and action behavior performance.

The decomposition of observed human behavior into primitive elements is based on the idea of structure learning [15], which captures three concepts that are important to

understanding human behavior: coordinating multiple control effectors into a smaller number of meta-controls, identifying the important connections between meta-controls and the accomplishment of task goals, and identifying repeating similar subtasks across a domain for which the same control strategy is applicable. Structure in a task can be defined in terms of the information that an agent's actions depend on in subsets of the task domain.

Human perception also plays a critical role in understanding human motion guidance behavior. Based on information theory, a human can be modeled as an embedded agent within a perception-action cycle. In this model, perceptual measurements and focus of attention form a closed-loop system together with action policies, in which optimizing performance and robustness corresponds to optimizing information flow in this loop. Concepts from cognitive science on predictive processing suggest that humans use a generative model to both perceive the environment and to plan actions. Observation of human gaze motion during a motion guidance task can be used to identify perceptual guidance relationships and identify predictive elements of perception.

1.3 Research Statement

The present work focuses on decomposing observed human motion guidance behavior into primitive structural elements, modeling the elements of behavior, and implementing algorithms for deploying motion planning and guidance behavior in new task scenarios. This work focuses on first-person human motion guidance tasks, such as driving a car, helicopter, or bicycle, and employs a new simulation and data collection framework to investigate this type of task. This work makes the following key contributions:

1.3.1 Perceptual Guidance Behavior Decomposition

Previous work decomposed 3rd-person motion behavior data by identifying planning subgoals and partitions [12], and mapping a spatial guidance policy that describes the aggregate set of guidance behavior [19]. The present work extends that by decomposing the an aggregate set of first person guidance behavior into dynamic mode elements based on system constraints, and identifying the specific perceptual guidance relationships that define agent motion. In addition, an information-based approach is used to model

the sparse structural relationships between agent perceptions and actions that define elements of motion behavior.

1.3.2 Mode Transition Properties

This research extends previous work that introduced subgoals [14] by providing specific necessary conditions that define optimal subgoal locations. The conditions specify subgoal locations relative to task constraint topology based on constrained optimal control principles. In addition, this work shows that a motion guidance policy can be described as a composition of dynamic mode elements, and defines models for each dynamic element as a relationship between agent perceptions and actions.

1.3.3 First-person Perceptual Functions

Previous work identified segments of gaze behavior corresponding to perceptual functions in third-person remote-control helicopter control [20]. The present work investigates gaze behavior during a first-person task and identifies analogous perceptual functions. The identified function provide insights into how humans plan and perceive during guidance task deployment.

1.3.4 Integrated Model

This work concludes by simulating integrated human guidance behavior. This work introduces a subgoal planning algorithm that generates constrained optimal control trajectories with higher performance and lower computational cost than other state of the art planners such as RRT*. The subgoal planner is able to quickly generate dynamic motion between subgoals by using a guidance-primitive automaton that plans a sequence of dynamic perception-action interaction elements that together define a solution trajectory.

1.4 Outline

- **Chapter 2:** Background and Related Work. This chapter summarizes prior work on constrained motion guidance, the structure of human motion guidance, research

on human gaze, perception, decision making, and information theory as it relates to the embedded human agent.

- **Chapter 3:** Experimental Framework. A key contribution of this work is the experimental framework used to record first-person perceptual guidance behavior. This chapter describes the simulated vehicle control task setup used to collect human behavior data used in this research.

- **Chapter 4:** Goals and Hypotheses. This chapter describes the detailed problem setup and the embedded-agent model that provides a platform for further analysis. This chapter introduces the overall approach for data analysis and modeling, along with the main hypotheses of this work.

- **Chapter 5:** Information-based Behavior Decomposition. In this chapter, human motion guidance strategies are investigated as principle components of information transfer between actions, states, and perceptual measurements. This analysis suggests that human motion consists of modes of behavior that are both temporally and functionally distinct.

- **Chapter 6:** Constraint-based Behavior Decomposition. Together with chapters 7 and 8, this chapter is part of the core contribution of this work. This chapter describes the identification of control modes in the observed human behavior based on the set of active constraints and information-based relationships between free variables. This decomposition extends the concepts in the previous chapter, recognizing that constraint transitions are critical to defining modes of guidance behavior. In addition, this chapter investigates operator gaze motion, and how it corresponds to identified guidance control modes.

- **Chapter 7:** Human Behavior Modeling. This is the second part of the core contribution, which formulates models that describe primitive elements of behavior and how these elements arise in a given task-environment scenario. The concept of constraint transitions are defined from optimal control theory to specify subgoal necessary conditions and describe how guidance behavior can be generated at each functional level.

- **Chapter 8:** Autonomous Behavior Deployment. As the final part of the core contribution, this chapter describes how an autonomous agent embedded in a task-environment scenario can identify a set of applicable primitive behavior elements and deploy them to generate a solution trajectory. Results are shown demonstrating how this is a practical approach to autonomous system guidance, modeling aspects of performance and computation found in human behavior. Planning solutions exhibit better performance and lower computational costs than results generated by a benchmark RRT* planner implementation.

- **Chapter 9:** Conclusion. This chapter summarizes results and suggests future research directions.

Chapter 2

Background

2.1 Overview

This chapter provides a summary of prior publications that frame the present research problems. Because this work seeks to understand elements of human behavior from an autonomous control perspective, it is influenced by guidance and control literature, as well as cognitive science and studies on human perception. This chapter first summarizes approaches to autonomous guidance and control, relating them to the human guidance task. Next, this chapter reviews research that has investigated patterns in human motion behavior and provide an understanding about how humans organize their interaction with the world. Third, prior work is introduces that investigates the elements of human perception, from eye motion to perceptual processing in the brain, providing insights into how humans filter essential information for a task from a noisy chaotic world. Finally, work on information theory and human interaction with the world is introduced as a method of tying together control theory, human motion behavior, and human perceptual mechanism into an integrated model of human motion behavior.

2.2 Autonomous Guidance

Autonomous guidance includes a wide range of approaches for choosing controls that direct the motion of an agent to complete a task within an environment domain [21]. This section both provides an overview of the latest approaches to robotic motion guidance,

and introduces methods that provide a theoretical basis for evaluating motion guidance strategies. Optimal control and planning algorithms provide a formal framework for investigating the motion guidance problem, and highlight the trade-offs and technical issues that must be accounted for by any autonomous or human system. These autonomous guidance approaches can each be evaluated in terms if biological feasibility – is it possible that a human brain uses elements of each method to perform in motion guidance – to understand what functional processes must be occurring within a human mind.

2.2.1 Optimal Control

Optimal control is concerned with methods for computing a continuous control sequence that drives a dynamic system from a start state to a goal state while minimizing an objective function [22]. A general solution to this minimization problem is obtained using a Hamiltonian function, based on the properties of function minima. For certain types of system dynamics and objective functions, analytical solutions can be found, such as the linear quadratic regulator (LQR) or finite-time minimum-energy control [23]. These solutions are mathematically elegant, but it may be infeasible for a biological system to perform the calculations needed to directly arrive at such an optimal solution. Nonetheless, optimal control approaches serve as an ideal baseline for comparison with other methods.

2.2.2 Constrained Optimal Control

Real-world guidance problems faced by humans typically involve constraints on allowable solution trajectories and nonlinear system dynamics. Closed-form solutions to the Hamiltonian function are generally not possible when arbitrary constraints are present or when system dynamics are nonlinear [24]. Properties of the Hamiltonian formulation however may be used to understand properties of optimal solutions. When constraints and system dynamics are linear, linear programming may be used to compute a solution [25], but this may be an infeasible approach for a biological brain. In addition, humans deal with nonlinear constraints with ease. For general system dynamics and constraint

boundaries, a dynamic programming approach must be used [26]. Dynamic programming algorithms however are typically implemented by iterating through a large number of discretized system states, which may be unlikely to exist within a biological system.

2.2.3 Graph Search

Dynamic programming involves recursion over finely discretized time and system states, incurring a high computational cost to closely approximate an optimal solution. Graph search approaches reduce computational complexity by using a course discretization, based on prior knowledge of the problem domain. A graph is constructed as a connected structure of nodes and edges within the system state space, and an optimal solution is found using a search algorithm such as A* or D* [27, 28]. Approaches such as RRT [29], and road-map methods such as Voronoi [30] and visibility graph [31] seek to build efficient graphs of the problem domain, optimized for a specific heuristic such as safety or path length. RRTs discretize the domain randomly, but are able to trade computation time for solution quality. RRT* [32] is modified to update tree connections at each step to ensure that the tree converges towards an optimal solution. RRT*-SMART [33] suggests a smart sampling method that forms beacon points, similar to subgoals presented here, which focus random sampling relative to obstructions.

Graph search and road-map approaches model some human goals, but do not implicitly consider system dynamics or more complex cost functions. To make the solutions dynamically feasible, they may be smoothed in post-processing using known vehicle dynamic capabilities [34]. The smoothing process reduces optimality however if it does not consider the global cost function. With respecto to biological feasibility, limitations in human working memory [35, 1] mean that large graphs are unlikely to be solved perfectly in advance; a human may not have the complete environment knowledge required.

2.2.4 Potential Field

Artificial potential field methods are commonly used in robotics path planning, and are based on the definition of a global, spatial cost function based on the start location, goal location, and obstructions [36]. Sources are placed within the problem domain so as to repel system trajectories away from the start location and environment constraints. A

sink is placed at the goal to attract system trajectories toward it. One shortcoming of this method is that some obstacle configurations induce local minima in the potential field, incorrectly directing solution trajectories to these points. One method of avoiding local minima is an approach called forward-chaining [37, 38], which generates subgoal-like points at locations adjacent to obstructions that are likely to result in a local minima. Forward-chaining is an ad-hoc approach to integrating global planning with motion guidance. Forward-chaining is similar to the subgoal approach presented in this work.

2.2.5 Receding Horizon

Path planning in an uncertain environment faces an inherent problem of combining uncertain prior global environment knowledge with locally sensed knowledge to generate a trajectory that maximizes expected value and minimizes risk. Receding horizon approaches deal with this by setting a time horizon, based approximately on sensor range [39, 40, 41]. A path within the horizon is planned exactly using an optimal control approach. This solution is concatenated with an approximate solution based on a global cost-to-go function representing prior environment knowledge. Mettler et. al [42] further refines the global cost function as a state-dependent, spatial value function (SVF). An SVF reduces computational complexity by collapsing higher-order states into an optimized function over only configuration states. These two path segments are joined at an active waypoint, lying on the horizon boundary and chosen to minimizes the total path cost. With this approach, an agent can converge to an optimal solution after multiple trials as a global value function is learned [43]. This approach models the relationship between short and long term memory in humans [35, 1] by separating local and global planning. In addition, this approach matches both the goals of human planning, and also aspects of the algorithmic implementation. Local optimal planning can still be computationally expensive however, since it does not take advantage of previously computed solutions to subtasks.

2.2.6 Maneuver-Automaton

[44] and [45] introduce the maneuver automaton (MA), which is a finite-state approximation of system dynamics. These behavior types are inspired by observed human pilot behavior during acrobatic flight [46]. A MA generates complex trajectories in terms of a sequence of motion primitive elements. Motion primitives are chosen from a library of known behaviors consisting of, for example, trim and maneuver elements. [47] evaluates this approach in the application of aerobatic rotorcraft control.

2.2.7 Machine Learning

Machine learning describes a set of algorithms with seek to model an input-output relationship by training on a set of example data. For a guidance task, a machine learning algorithm can learn a guidance policy, specifying an agent action as a function of agent state, based on example motion data provided by either a human or from self-guided exploration of a task domain. Examples of learning algorithms include neural networks [27] and Gaussian process regression [48]. A problem with learning for motion guidance is that the learned function, $a = f(z)$ giving actions a in response to every perception z an agent may encounter over its lifetime is intractable and would require a huge amount of example training data to enumerate. In contrast, humans are able to extrapolate behavior in a wide range of tasks with sparse prior experience.

Option Policy Learning

Recent developments in machine learning provide methods to increase learning rate based on the sparse structure present in many tasks. Sutton et al. [49] introduce the concept of options in Markov decision process (MDP) problems. An option is a sequence of actions that are defined by an initial set, an option policy, and a termination condition. Options reduce the representation size of an MDP task policy because similar options may be reused in different parts of the total task domain. Sutton at al show that policy learning rate is increased when options are considered.

Deep Learning

Deep learning is an approach to machine learning that utilizes deep neural networks (DNN), which are neural networks consisting of multiple hidden layers. DNNs have recently become popular due to advancements in both computational capability, and advancements in algorithms that efficiently back-propagate example data to update parameters in the hidden layers of the network during learning. Recent research indicates that DNNs are able to learn using less example data by taking advantage of the sparse structure in input-output relationship [5].

2.2.8 Task Structure

While option policies and deep learning approaches can take advantage of structure in a guidance task, it is not clear how the structure they use relates to properties of the task geometry. The following are theoretical approaches that reveal structure in specific problems.

Sliding-Mode Control

Sliding-mode control is a strategy used to design a robust controller for nonlinear systems. In this approach, a trajectory or series of trajectories are designed, which route the system state to the desired equilibrium point. The trajectory is specified as a function of system state, i.e. $c(x) = \mathbf{0}$ [50]. A sliding mode acts like a constraint, and collapses system dynamics to a lower-order model confined to the sliding mode surface. A feedback controller maintains the system on the sliding mode, with the requirement that the system converges to the sliding mode more quickly than it converges to the equilibrium state, driving the use of bang-bang, or high gain feedback laws. In this approach the control task is simplified by designing convergent, hierarchical trajectory segments that each aim to regulate a subset of system states, and are similar to MDP option policies or maneuver-automaton approaches. Importantly, well-defined conditions exist that ensure the finite-time stability of a sliding mode controller [51] over a range of states and system parameters, which makes this a feasible approach for humans to achieve robust behavior.

Homotopy Classes

Algebraic geometry can be used to describe the topology of guidance trajectories relative to constraints. A homotopy class contains a set of all trajectories that can be continuously morphed into each other without passing through a constraint [52]. When an environment contains a discrete, disjoint set of obstructed subsets of the task domain, then trajectories fall into a discrete number of homotopy classes. Determining the possible homotopy classes is therefore required to relate continuous trajectory planning with discrete route decision making. Bhattacharya et al. introduce a 1-Form integral over a candidate trajectory relative to constraint geometry that determines its homotopy class in any number of spatial dimensions [53]. This method is then applied to graph search based robot path planning [54]. For humans, the concept of a discrete set of possible routes is very intuitive, so it is likely that homotopy classes are important in reducing task complexity by separating the discrete task planning problem from continuous motion control.

Partitions

Kong and Mettler investigated general patterns in 2D optimal optimal trajectories, considering both environment constraint topology and vehicle system dynamics [14]. Solution trajectories exhibit partition and subgoal behavior. Trajectories within each partition can be related through symmetry transformation. These patterns can be used to simplify the planning task by reducing the set of unique motion elements needed to generate a trajectory solution.

2.2.9 State Estimation and Localization

The autonomous guidance and planning approaches above have so far assumed that the agent has access to the system state, and information about the environment. In reality, agents must make noisy measurements using available sensors, and typically experience the environment from a first-person perspective in which they do not have access to global environment knowledge. Engineered systems widely approach perception as a state estimation problem [55, 56]. When interacting with environment constraints, the agent must solve a landmark-based navigation problem, in which prior knowledge of

measurable environment features (i.e. a map) is available [57, 58, 59]. Humans use primarily vision for this task, with very little depth measurement capability. When the agent is also simultaneously estimating the layout of the environment, it is a bearing-only SLAM problem [60]. Computational issues arise with these approaches if the set of environment landmarks is large, therefore, the agent must focus on a subset of landmarks with sufficient measurement value. In addition, the agent may need to re-plan their route if there are insufficient visual cues available along the original path to successfully implement their control policy. Frintrop investigated filtering of visual cues using a saliency metric to identify landmarks that are easily measured across multiple visual frames [61]. Lerner [62] recognizes that the value of selected landmarks depends on the specific guidance task. While humans are unlikely to perform Kalman filter calculations internally, qualities of optimal filtering methods may be used to determine bounds of the information obtainable by a human in a specific sensory task-environment scenario.

2.3 Human Motion Guidance

This section reviews research in cognitive science, artificial intelligence, and dynamics that observe and investigate characteristics of human behavior. In contrast to the previous section, this work seeks to identify patterns and properties that might be transferred to autonomous approaches to achieve human-inspired robust and adaptive guidance performance.

2.3.1 Human Guidance and Interaction Patterns

To investigate specific properties of human guidance behavior, the present work is based on work by Mettler and Kong that proposes a language for describing structure in human motion task behavior based on the concept of an interaction pattern. The authors first propose that this behavior can be analyzed in terms of a language of schemas, or agent-environment action elements that are assembled to create complex behavior[9]. The authors then present an experimental approach to observing these behavioral patterns in a human subjects and comparing them to patterns expected in a rational planning agent. Kong and Mettler extend this by more precisely defining the

formal language [63]. Patterns are identified in a task domain as symmetries and invariances in agent-environment interaction. Mettler et al next introduce an experimental framework used to observe interaction patterns in human vehicle control [64]. The research infrastructure allows a human subject to control a remote control helicopter from a third-person perspective. The system records helicopter motion, operator commands, and operator gaze motion via eye tracking goggles. The system also has the capability to incorporate autonomous feedback control of the helicopter, and simulated vehicle or environment elements. The authors then formally define equivalence relations that identify interaction patterns within a set of observed motion behavior [12]. First, subgoal and guidance equivalences are used to extract subgoals and trajectory segments. Trajectory segments can then be considered in aggregate as a library of motion elements, or interaction patterns. Trajectories segments are also dynamically clustered to identify different types of dynamic behavior, such as accelerating, trim, and target closure.

2.3.2 Hierarchical Model of Human Guidance

Another contribution by Mettler and Kong is to formalize the structure of interaction patterns as a hierarchical model which relates planning, guidance, and tracking levels of motion behavior [7]. This work emphasizes that the flow of information between hierarchical levels, such as planning and motion control constitute knowledge contained in behavioral elements. In addition, the hierarchical model bridges the gap between discrete decision making and continuous motion control. The authors then analyze an aggregate set of trajectory segments, extracted using the subgoal equivalence, to map guidance behavior [19]. Aggregate guidance behavior is described by cost-to-go and velocity-vector maps. This paper also maps dynamic quantities, such as acceleration and turn rate to see how they are distributed over the guidance domain relative to optimal behavior expected in a rational agent. Mettler et al combines many of these concepts into an integrated, systems model of human guidance and perception [6]. This work describes how interaction patterns, based on invariances and symmetries in agent-environment dynamics form interacting guidance solutions at each level of the hierarchical model. Based on observation of experimental human guidance behavior, perceptual behavior elements are defined which form a reciprocal feedback loop with motion guidance elements.

2.3.3 Human Skill Analysis

Li at al apply the hierarchical model of interaction patterns to analyze skill in human pilots [65] and for a tele-operated surgery task [66]. A data-driven dynamic clustering approach is used to identify interaction patterns, with results showing that greater distinction between modes corresponds to greater operator skill. This trend is important to both understand properties of skilled human behavior, and to validate interaction patterns as a model for how humans achieve high-performance behavior.

2.4 Human Perception

The previous related work emphasizes the important of perception in human interaction with their environment. This section reviews prior work specifically on human visual perception. Marr provides a general framework for understanding human perception, processing and representation of sensory information, known as the tr-level hypothesis [67]. It states that any information processing mechanism, human or machine, can be understood at three levels. First, the computational theory describes the goal of the computation in terms of logic or math. Second, is the representation of the algorithm - what is the representation for the input and output, and what steps are required to implement the computational theory? Finally, is the hardware implementation. What type of physical mechanism is required to execute the algorithm. This model can be used to understand human visual processing, and to evaluate perceptual algorithms for biological feasibility.

2.4.1 Eye motion

Investigating perceptual behavior begins with understanding how the eyes move. Since human eyes have a narrow fovea, eye motion reveals how the brain focuses attention in the surrounding environment. Humans must actively direct visual attention to locations of interest in order to achieve sufficient situational awareness for motion guidance tasks [68]. Work by Land [69] investigated how eyes move relative to a task, suggesting they move pro-actively to seek out information needed to plan the next action. Gaze-tracking devices record eye motion as a subject completes various tasks [70]. Data reveals that gaze motion consists of three types: fixation, smooth-pursuit, and saccades [71].

2.4.2 Focus of Visual Attention

Prior work investigating human visual attention introduced a visual saliency model to predict human gaze motion around a static image [72]. The saliency model is based on the hypothesis that human attention is directed to the regions of an image that are sufficiently different, or stand out in some way. The saliency model has been validated using gaze location data for subjects viewing a variety of images. Saliency that is based only on image content is termed "bottom-up" direction of attention. When a subject views a scene for the purpose of performing a specific perceptual task, domain knowledge is also important [73, 74] and constitutes "top-down" direction of attention. Attention is directed to regions of the image most relevant to the task. For example, during an object search task, visual attention is directed to parts of the image most likely to contain the target object, based on prior knowledge about how objects are likely to be positioned relative to each other. Johnson et al observe gaze behavior in subjects while performing a specific motion task - operating a driving simulator [75]. They propose that gaze at a specific location provides a measurement for the purpose of reducing uncertainty in a control variable, such as speed or following distance. A model is proposed which determines the desired measurement based on variable uncertainty and priority to the task. They found that this model approximates human gaze behavior for the driving task.

2.4.3 Perception of Motion

Prior work investigated ways that humans visually perceive motion. This includes optical flow, which is the motion of textured surfaces across the visual field. If the geometry of the surrounding environment is known, the optimal flow field can identify motion of the observer [76]. Other visual effects were studied in terms of how the brain detects motion at an early stage in visual processing. Looming—the visual effect of an object moving closer in the visual scene—is used to detect and control self motion [77, 78]. The theory of Tau guidance suggests that humans and animals detect time-to-closure of gaps in the agent-environment state [17, 79]. They then use this information to control motion.

2.4.4 Useable Cue Environment

In the field of aircraft handling qualities evaluation, empirical studies have evaluated the relationship between the quality of visual cues available, the level of control augmentation (response type) and the control performance attainable by the pilot. Usable cue environment (UCE) specifies the required level of control augmentation, given a degraded visual environment (DVE) required to achieve level-1 handing qualities performance [80, 81]. A crucial point is that this approach suggests that cue quality can be quantified in terms of the resulting control performance.

2.4.5 Human Perception in Motion Guidance

Andersh et al. take a closer look at the perceptual behavior of a human subject operating a remote-control helicopter from a third-person perspective [20]. This work seeks to observe both operator gaze and robot trajectories, to understand the specific function that types of gaze behavior play in vehicle control. Gaze behavior is consider at each level within the hierarchical human guidance model. Results show that gaze motion behavior falls into discrete functional categories such as pursuit of the helicopter and saccades to measure a gap to the target.

Verma at al investigate human gaze behavior while performing first-person navigation in a new environment [82]. The hypothesis is that as a subject learns a task, they learn subgoal locations, and that visual perception of subgoal locations is critical to learning environment layout and navigating around obstructions. Results shows that subjects spend a significant portion of gaze-time fixated at the subgoal locations which are optimal given their current environment knowledge. Verma and Mettler also observes how subject vehicle guidance characteristics, such as speed and turning ability influence exploration and navigation in the environment [83]. Results show that subjects ignore possible routes that are dynamically unlikely (i.e. routes requiring sharp turns at high speed).

2.5 Embedded Agent and Information

2.5.1 Embodiment and Perceptual Guidance

Work in cognitive science in the area of embodiment [84, 85] suggests that humans and animals make use of their physical situation in the environment as part of behavior and motion generation. In other words, they offload processing and information storage into the structure of the environment to reduce computational demands. The concept of perceptual guidance builds on this to suggest that humans generate motion by maintaining simple relationships between visual cues in the environment [17]. This provides a potentially much simpler method of generating control actions, which may not be optimal, but satisfy the constraints for a particular task.

Tau guidance extends this by defining a specific quantity Tau, which is the instantaneous time to closure of a perceptual gap [86]. The rate of change of Tau, $\dot{\tau}$, describes the type of motion: $\dot{\tau} > -0.5$ results in stopping short, $\dot{\tau} = -0.5$ results in stopping at the goal, and $\dot{\tau} < -0.5$ results in continuing motion past the goal. Empirical studies have shown that a wide range of biological motion can be described in this way. To form more complex motions, multiple perceptual gaps are closed simultaneously, for example the distance and angle to a goal point. In this case, Tau parameters may be coupled linearly [87], e.g. $\tau_1 = k\tau_2$, to coordinate motion.

2.5.2 Graphical Models

Graphical models are a tool for representing the causal relationships between random variables in a stochastic system. The interaction of an agent with its environment over time can be modeled by a graph, and used to understand the primary modes of information transfer. When estimating the correlations between a high number of variables, it is useful to project the covariance matrix into a sparse domain - i.e determine only the strongest variable relationships. Sparse learning methods provide this capability to define useful graphical models [88, 89, 90].

2.5.3 Subspace Clustering

Subspace clustering is based on the idea that correlations or relationships between variables arise when behavior occurs within a subspace of the full task state space, and has previously been used to identify specific types of motion or behavior in video sequences [91]. Temporal subspace clustering extends this approach to explicitly recognize that points close together in time are likely to belong to the same subspace [92].

2.5.4 Perception Action Cycle and Information

Shannon first introduces information, measured in bits, as a quantification of communication channel capacity [93]. Recent work models human behavior using coupled dynamic systems and information theory, which provides insight into modeling human motivation and behavior in uncertain environments. Warren observes that the interaction of coupled environment-agent systems describe human behavior while performing specific tasks [94]. Properties of the dynamic systems, such as equilibrium points, equilibrium trajectories, bifurcations, repellors, and attractors, can be related to elements of human behavior. Tishby and Polani model coupled perception and action dynamics as a graphical information-flow model [95]. The information required to complete a guidance task is related to Shannon's definition of information in a communication channel. A Bellman-like equation is presented which describes optimal perception-action behavior by minimizing the information processing required along a guidance trajectory. Polani also investigates the information processing requirements of biological systems during motion control tasks [96]. It is observed that embodied guidance strategies can significantly reduce information processing requirements of a task, and demonstrates the motivation for humans to use these types of strategies.

Other researchers use the perception-action model to investigate costs or utility functions that humans may use to make decisions and generate actions. Lungarella looks at human motion costs by using a robot simulation to show that how an organism is embedded in its environment influences the structure of information flow between sensory perception and motor actions [97]. This is further evidence of the role of embedded cognition in human guidance. Drogowitsch applies the information-based perception-action model to measure the cost of information processing in humans and monkeys

[98]. Based on the relationship between task performance and thinking time allowed, the cost of processing new information vs. time is modeled. Laughlin relates information processing in sensory tasks to physical energy consumption in brain neurons [99]. This provides evidence that minimization of information processing is a real physical constraint for biological systems that drives motivation in motion tasks. Schwartenbeck investigates human motivation in decision making in terms of minimizing a quantity defined as surprise [100]. The analysis shows that minimizing long-term surprise leads to behaviors such as exploration and exploitation. Another quantity related to surprise is quantified as empowerment, which is is maximum mutual information between actions and perceptions across all action choices [101]. Klyubin shows that empowerment describes a wide range of human behavior. While empowerment may describe long-term human motivation, short-term behavior is defined by a near-term goal. Edge et al. define goal-weighted empowerment as a metric for short-term human behavior [102].

2.5.5 Structure Learning

An information theory approach to behavior defines a guidance policy as a joint probability distribution over perceptual inputs and agent actions, $p(A, Z)$. In complex and cluttered real world tasks with many perceptual measurements and many available actions, this distribution may be intractable to learn or evaluate efficiently. To overcome this complexity, it is beneficial for an agent to identify a sparse set of relationships within this distribution that approximate optimal behavior. Structure learning is one approach to identifying these sparse relationships. Braun et al. [15] describes structure learning as learning to learn, or extracting invariants across task episodes that allow knowledge to be shared.

Another aspect of structure learning involves recognizing subtasks across a domain for which the same structure applies. For example, humans may perform motion tasks that are the same, but transformed by a symmetry transformation, such as rotation or reflection. The structure is the same in each case, but the specific transformation is different. To identify structure, Van Dijk et al. [103] describes an information-based method for identifying subgoals in a guidance task modeled as an Markov decision process (MDP). The idea is that while an agent is guiding to a specific subgoal, their control actions depend on the relative subgoal state, also referred to as relevant goal

information (RGI). This informational view on subgoals is related to the information bottleneck concept [104].

2.6 Discussion

This chapter summarized prior work in understanding human motion guidance and perception behavior, autonomous and optimal control approaches to motion guidance, and statistical tools such as graphical modeling and subspace clustering that may be used to investigate patterns in human behavior. The next chapter introduces an experimental framework that records human motion behavior, providing data on which to apply these approaches.

Chapter 3

Experimental Framework

3.1 Introduction

3.1.1 Motivation

To investigate guidance and perception behavior, an experimental system is used to observe this behavior in human subjects. The primary goal of this framework is to present a first-person guidance task to the subject, and observe the action and perception relationships generated by the subject to complete the task. The first-person perspective requires a subject to learn at both the guidance and planning levels. At the planning level to determine feasible routes based on visual cues, and at the guidance level to learn optimal relationships between control actions and visual cue motion. These learning tasks are challenges also faced by autonomous systems with first-person sensors. Understanding how humans perform guidance from this perspective is therefore directly applicable to investigating human-inspired approaches to autonomous guidance.

3.1.2 Approach

Experiments consisted of both simulated and real first-person guidance tasks. A simulated environment allows for precise control of the available visual cues used by the subject, a precise and repeatable environment configuration, and consistent vehicle dynamic response. A real guidance task is used to validate simulation results, and observe human reactions to the complexity in a natural scene environment.

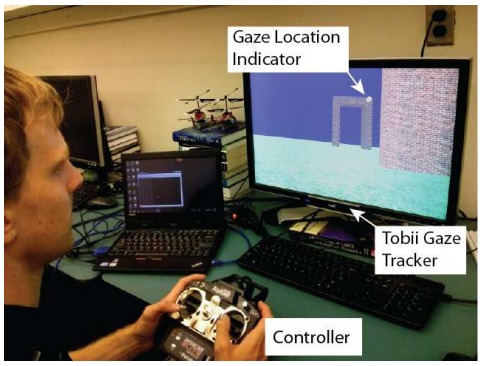
During both simulated and real first-person experiment trials, a subject uses a controller to move a vehicle through an environment. The objective of each trial is to move from a specified start position to the goal corridor in minimal time while avoiding obstructions in the environment. While navigating through the scene, a gaze tracking device records the gaze direction of the subject. This data is used to determine which portions of the environment the subject is focusing on during specific phases of the task. Experiments are designed to provide data for the following types of analysis:

Planning Behavior

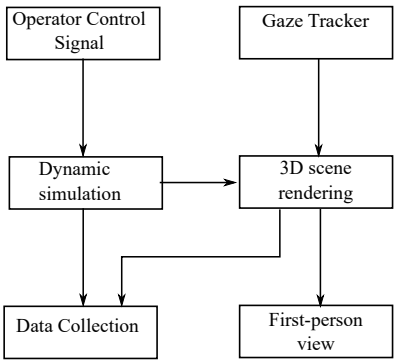
Previous work by Mettler and Kong investigated third-person human motion guidance behavior. Based on properties of optimality, they identified subgoals that define partitions within the planning domain. Subgoals and partitions encode the underlying planning structure learned by an agent in a specific environment domain. The goal of this experimental framework is to validate the use of subgoals by humans in dynamic, real-life, first-person motion guidance tasks. In addition, the observation of gaze behavior is expected to provide information about the perceptual functions that humans use to plan in a first-person task.

Guidance Behavior

Previous work by Mettler and Kong show that optimal guidance behavior can be described in terms of a spatial value function (SVF), and that aggregate sets of behavior across partitions of a task domain may be described by a common spatial policy. The goal of the present experiments is to collect an aggregate set of first-person guidance behavior and first validate the SVF model. Second, the set of guidance behavior will be further decomposed based on the hypothesis that humans compose guidance behavior from discrete patterns of interaction. Each interaction pattern is defined by a relationship between actions and perceived sensory information. Furthermore, interaction patterns are individually optimized for performance and risk.



(a) Simulation setup.



(b) System architecture.

Figure 3.1: First-person experimental framework.

3.2 Simulation Experiments

3.2.1 Simulation System

The simulation system is implemented on a Linux PC, using the robot operating system (ROS) for communication between modules. The 3D environment model is constructed using SketchUp modeling software. The simulation module reads the environment model file and renders the first-person view of the environment using open scene graph (OSG) libraries, based on current vehicle position and orientation. A dynamics module reads in the subject controller input signals, and updates the position and orientation of the vehicle at each time-step based on defined vehicle dynamics. A data collection module records vehicle position, heading, velocity, operator control signals, and gaze position to a file at each time step during experiment trials.

Subject gaze position is measured by a Tobii gaze tracking device attached to the simulation monitor. The gaze tracker uses cameras to track the position of the subjects pupils as they look at the screen, and is calibrated to accurately translate pupil position to corresponding screen coordinates. The gaze tracker is connected to a separate computer, which reads gaze position and sends it as a ROS message to the simulation computer. The simulation module on the simulation system computer converts the gaze position on the screen to a gaze vector in the simulated environment, and projects it onto the environment model surface. If the subject gaze vector is directed towards a modeled surface in the environment, the system records the 3D intersection point.

The simulation system includes functions to reset the vehicle to a specified start location and begin data collection. A configuration file defines specified start and goal locations. The system keeps track of trajectory time from when the vehicle begins moving during a trial to when it reaches the goal, and displays trajectory time on the screen for the subject when each trial is complete.

3.2.2 Procedure

For each trial, the vehicle begins at an initial position within the field, $\mathbf{x}_k = \{x_0, y_0, h_0\}$, where h_0 is a constant initial height for all start locations, set at 2 meters in the present experiment. A task configuration file provides a series of initial positions, $\{\mathbf{x}_1, \mathbf{x}_2, \dots\}$ for the experiment. The task for the subject is to use control inputs to move the vehicle from the start location through the goal doorway without colliding with environment obstacles. Task performance is measured by travel time to the goal, experimental subjects are instructed to attempt to minimize time during each trial. Travel time is displayed in the corner of the screen after the goal threshold is reached to give the subject immediate feedback on their performance. At each start point, the subject is instructed to repeat the trial until they reach a consistent level of performance (travel time).

3.2.3 Vehicle Dynamics

The vehicle dynamics module simulates a system model replicating an ideal unicycle-type response. The system is constrained to 2D motion: $z = h_0 = \text{constant}$. The system has two control inputs, consisting of forward and lateral acceleration. The dynamics are defined in Eqn. 3.2:

$$\begin{aligned} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} &= \begin{bmatrix} v\cos\psi \\ v\sin\psi \\ \min(u_{lat}/v, \omega_{max}) \end{bmatrix} \\ \dot{v} &= k_{acc} * u_{lon} - k_{drag} * v \end{aligned} \tag{3.1}$$

In Eqn. 3.2, u_{lat} is limited based on lateral acceleration ($a_{lat} = v u_{lat}$), $u_{lon} \in [0, u_{max}]$, and k_{drag} is a drag coefficient that provides speed stability. k_{acc} is an acceleration gain that defines the maximum acceleration possible. These vehicle dynamics were selected to

provide both an easily-controlled vehicle for the subjects that also incorporates essential elements present in real vehicles. The system requires no disturbance rejection, is stable, and contains minimal integrators between operator control inputs and system states, which could require additional training by participants to successfully control. The lateral acceleration limit represents either a limit on aircraft bank-angle during turns, or a limit in lateral traction of a wheeled when turning on slippery surfaces, and requires subjects to plan turns in advance to successfully navigate the course.

3.2.4 Environment

The 3D environment model consists of a flat ground plane and a goal corridor. The goal corridor provides a 2 meter wide by 5 meter long hallway that the subject must move the vehicle through to successfully complete a trial. Such a goal corridor restricts the range of directions in which the vehicle can be moving in when it reaches the goal position. In addition, the environment contains rectangular obstacles. Obstacles are 5 meters high, with varying lengths and widths depending on the specific course design. Fig. 3.2 depicts an example view within the simulation environment of the goal doorway and part of an obstacle on the left side of the image.

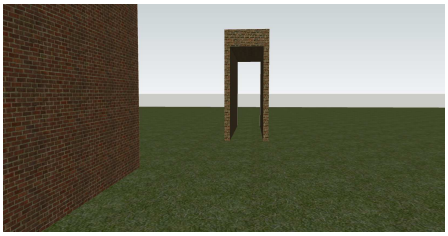
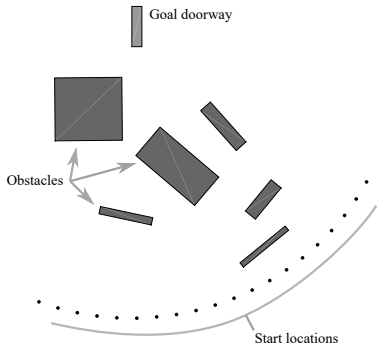


Figure 3.2: Example first-person simulation goal-doorway.

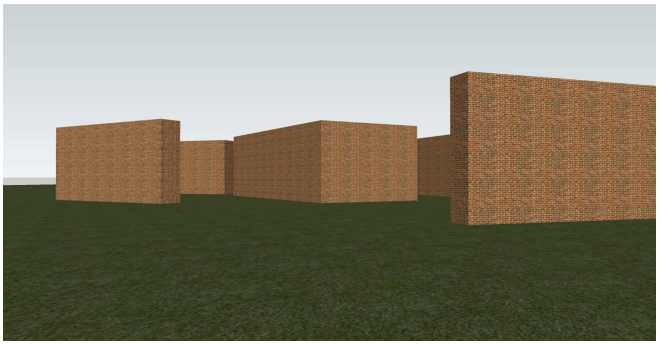
3.2.5 Uniform Course

Previous third-person experiments investigated patterns across multiple solution trajectories in a constrained environment [8]. First-person simulation experiments are designed to identify these patterns in more complex environments.

Two types of uniform obstacle courses were used. A course with a uniform distribution of rectangular obstructions was used for initial tests, with vehicles beginning at



(a) Top-view of uniform obstacle course.



(b) Typical first-person view.

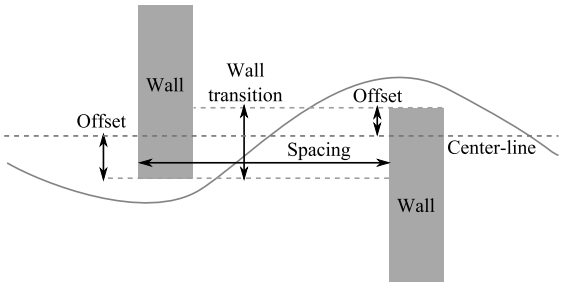
Figure 3.3: Random uniform course configuration.

an array of positions along the edge of the course. After initial trials, it was found that behavior in the uniform course did not exhibit a wide range of turning behavior. To improve upon this, a second random uniform course was created. A series of courses were randomly generated, then evaluated and selected based on the range of turning behavior exhibited in a simulated set of solution trajectories in that course. The resulting random-uniform course is shown in Fig. 3.3. This course pushes the operator to use a variety of motions as they approach and perceive obstructions from different angles. It is designed to reveal emerging behavior patterns in an ensemble of motion and perception trajectories.

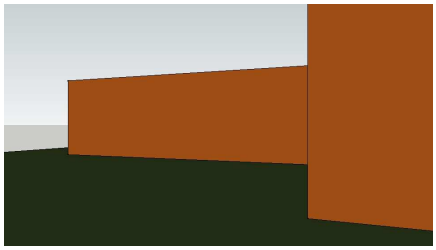
3.2.6 Slalom Course

To focus on motion guidance and perception behavior, a slalom course was created (Fig. 3.4). This course is designed to require the agent to use a repeating pattern of obstacle avoidance maneuvers. This type of course removes the planning and decision making aspect of vehicle guidance, and allows the agent to become proficient at a single repeating guidance task. The course consists of 38 walls extending in alternating directions away from the course centerline, with a spacing of 11 meters between subsequent walls in the downrange direction, as shown in Fig. 3.4c. Walls are offset toward the centerline by distances of 0, 0.375, and 0.75 meters, in random order. Randomized offsets reduce the ability of a subject to predict the course layout, driving the subject to visually

perceive each wall location. Wall spacing and offset geometry is shown in Fig. 3.4a. For this task, textures of environment surfaces were removed, as shown in Fig. 3.4b. This intentionally creates a sparse set of visual cues that limit distractions normally present in the real world.



(a) Wall offset and spacing.



(b) First-person view of slalom course.

(c) Entire slalom course layout.

Figure 3.4: Slalom course configuration.

Chapter 4

Goals and Hypotheses

4.1 Goals and Hypotheses

The primary goal of this work is to to identify and model human motion guidance functions that enable dynamic, high performance, robust, and adaptive behavior in a wide range of tasks. Function identification includes decomposing observed guidance behavior into elements of specific perceptual guidance relationships. This model includes a generative capability that deploys human-like behavior in new task-environment scenarios. The model combines functional elements corresponding to capabilities exhibited in human behavior, including discrete task planning, motion trajectory planning, perceptual guidance strategies, and focus of sensory perception.

4.1.1 Hypotheses

The perceptual guidance investigation is based on the following two overall hypotheses:

First, is that humans reduce task complexity by identifying and learning structure within a task-environment scenario. Structure provides a framework for decomposing a problem into a set of primitive elements of behavior that may be repeated throughout a task. Structure is identified by equivalence relations that result in a partitioning of the problem space into equivalence classes. Each equivalence class uses the same solution behavior.

Second, human behavior is motivated by a combination of maximizing overall task performance (in terms of some metric such as energy or time) and by minimizing risk

(in terms of mutual information afforded by the relationships between available actions and perceptual measurements). Risk and performance of overall behavior is optimized by selecting and refining constituent primitive elements.

4.1.2 Specific Goals

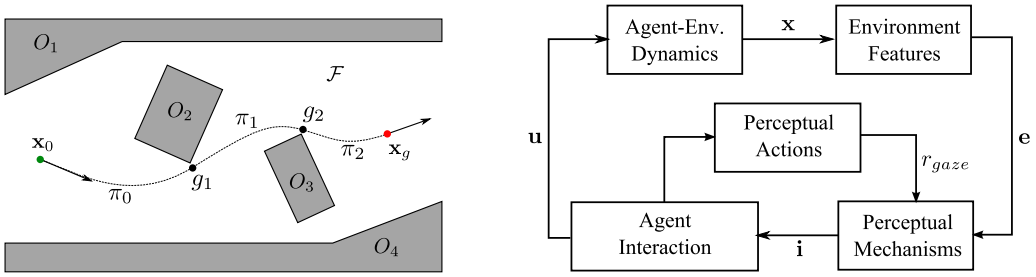
Planning Behavior At the discrete planning level, the goal is to identify formal conditions specifying optimal transition points between subtasks as a function of a motion guidance policy and environment geometry. The goal is to apply these conditions to identify subgoals in fully or partially known environments, converting the continuous control task into a discrete decision-making problem.

The specific hypothesis is that humans represent a planning task as a set of connected subgoals. As a representation, subgoals both contain the relevant task information contained in perceptual measurements, and specify the actions an agent must take to reach a guidance objective. This hypothesis predicts that subgoals have an optimal control theoretical basis, and that observed human behavior is consistent with this representation.

Guidance Behavior At the continuous motion guidance level, the goal is to identify a model characterizing first-person spatial guidance behavior. This a includes spatial value function (SVF) model and a motion-primitive maneuver automaton (MPMA) model describing motion behavior as a sequence of discrete modes. Each control mode consists of specific perceptual guidance strategies. Perceptual guidance strategies are lawful relationships between actions and perceived quantities that result in the observed spatial guidance or motion primitive behavior. In addition, the goal is to apply these models to generate human-inspired motion behavior, and to use this framework to formally define stability and robustness criteria.

The specific hypothesis is that humans deploy continuous motion guidance behavior as a sequence of control modes. This hypothesis predicts that control modes can be identified from continuous behavior as subspace segment clusters, and that unique perception-action relationships can be identified defining each control mode.

Perceptual Attention Identify a model describing visual attention during perceptual guidance, spatial guidance, and task planning functions. Each guidance function



(a) Guidance problem overview depicting start and goal states $\mathbf{x}_0$ and $\mathbf{x}_g$, subgoals g_1 and g_2, and guidance elements π_0, π_1, and π_2.

(b) Agent-environment system components.

Figure 4.1: Agent-environment system.

specifies requirements on information needed by an agent to select actions. These requirements can be used to determine the value of specific cues at a point during the guidance process. The goal is to use these informational requirements to identify a model for visual perception, generating a set of key visual cues at a given point during the guidance process.

The specific hypothesese are that, first, gaze location provides information about the specific environmental information relevant to an agent at that point in guidance. This hypothesis predicts that measurements of specific cues attended to during a guidance task contain the required perceptual information. The second hypothesis is that the active perceptual guidance control mode specifies what perceptual information must be obtained by the agent. This hypothesis predicts that gaze behavior will change depending on the current control mode, and that the active control mode depends on the visual information available at a point in a task.

4.2 Functional Behavior Model

4.2.1 Overview

To investigate human guidance and perception behavior, an agent is considered to be embedded in their environment, forming a closed-loop agent-environment system as shown in in Fig. 4.1b. The environment state $\mathbf{e} \in \mathcal{E}$ describes environment objects O_i

Symbol	Description
$\mathbf{x} \in \mathcal{X}$	Agent state.
$\mathbf{e} \in \mathcal{E}$	Environment state.
$\mathbf{i} \in \mathcal{I}$	Environment information.
$\mathbf{u} \in \mathcal{U}$	Control action.
$\mathbf{r}_{gaze} \in \mathcal{R}$	Gaze direction.

Figure 4.2: Agent-environment system signals.

relative to the agent. The function $\mathbf{e} = h(\mathbf{x})$ describes the relationship between agent state and environment state, consisting of a transformation into the agent's first-person reference frame. Perceptual mechanisms describe information $\mathbf{i}$ that an agent extracts from the environment state using a perceptual function, $\mathbf{i} = g(\mathbf{e}, \mathbf{r}_{gaze})$. Perceptual actions capture agent actions $\mathbf{r}_{gaze}$, including eye and head motion, that changes the available perceptual information the agent can obtain from the environment.

4.2.2 Agent-Environment Dynamics

Agent-environment dynamics describe how system state changes in response to control inputs, $\dot{\mathbf{x}} = f(\mathbf{u}, \mathbf{x})$. The agent-environment state consists of a workspace $\mathcal{W}$ and higher-order state, $\mathcal{V}$, such that $\mathcal{X} = \mathcal{W} \times \mathcal{V}$. The workspace $\mathcal{W} = \mathcal{O}_E \cup \mathcal{F}$, contains free space ($\mathcal{F}$) and environment objects ($\mathcal{O}_E = \bigcup_{i=1}^{k} O_i$). Motion of the system through space is constrained by the dynamics of the vehicle or body, $\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t))$, with control input sequence $\mathbf{u}(t) \in \mathcal{U} \subset \mathbf{R}^m$, for time $t \in [t_0, t_f] \subset \mathbf{R}$. $\mathbf{x}(t) : [t_0, t_f] \rightarrow \mathcal{X}$ denotes a solution to the system. A solution trajectory begins at state $\mathbf{x}_0 = \mathbf{x}(t_0)$, and ends at $\mathbf{x}_g = \mathbf{x}(t_f)$. The set ${}^{\leftarrow}_{s}{}_{u}(\mathbf{x}_0) = \{\mathbf{x}(t)\,,\, [t_0, t_f]\}$ contains all points on the trajectory.

System dynamics are hierarchically partitioned so as to separately consider the discrete task sequence, continuous reference trajectories, and disturbances in higher-order dynamics [7]:

$$\text{Task transition:} \quad g_{k+1} = \Phi(g_k, \pi_k) \tag{4.1}$$

$$\text{Kinematics:} \quad \dot{\mathbf{x}}_p(t) = \mathbf{v}_{\mathbf{ref}}(t) \tag{4.2}$$

$$\text{Dynamics:} \quad \dot{\mathbf{v}} = f(\mathbf{v}, \mathbf{u}) \tag{4.3}$$

In this hierarchical model, a subgoal $g_k \in \mathcal{X}$ is a task state representing an intermediate goal on the solution trajectory $\overleftarrow{s}_{x_g}$, as illustrated by points $\{g_1, g_2\}$ in Fig. 4.1a. Transitions between subgoals are specified by $\Phi(g_k, \pi_k)$, depending on the current subgoal, g_k, and a guidance element π_k. A solution trajectory consists of a sequence of trajectory segments, $\overleftarrow{s} = \{\pi_1, \ldots, \pi_n\}$. A guidance element π_k represents a reference trajectory, forming a path between start and end task states $\mathbf{x_p}(0)$ and $\mathbf{x}(T)$ as specified by a reference velocity, $\mathbf{v}_{ref}(t)$ for $t \in [0, T]$. Guidance elements π_0, π_1, and π_2 are labeled in Fig. 4.1a. Finally, vehicle velocity $\mathbf{v}(t)$ is modeled by dynamics $f(\mathbf{v}, \mathbf{u})$, evolving based in response to control inputs.

4.2.3 Agent Interaction

Agent interactions describe how an agent chooses control actions $\mathbf{u}$ in response to perceptual information, $\mathbf{u} = k(\mathbf{i})$. The human guidance task is modeled as a constrained optimal control problem as depicted in Fig. 4.1a. The objective is to determine a control sequence $\mathbf{u}(t)$ that guides the system from an initial state, $\mathbf{x}_0 \in \mathcal{F}$ to a goal state, $\mathbf{x}_g \in \mathcal{F}$, while satisfying system dynamics and spatial constraints, and minimizing trajectory cost. The cost of a trajectory $\overleftarrow{s}_u(\mathbf{x}_0)$ resulting from control sequence $\mathbf{u}(t)$ is defined by the functional:

$$J(\overleftarrow{s}) = \int_{t_0}^{t_f} j(\mathbf{x}(t), \mathbf{u}(t))\mathrm{d}t \tag{4.4}$$

In Eqn. 4.4, $j(\cdot)$ is the differential path cost.

This guidance task is modeled as hierarchically partitioned elements, describing different levels of environmental interaction:

$$\text{Subgoal planning:} \qquad \pi_k = \gamma(g_k, x_g, \tilde{G}_k) \tag{4.5}$$

$$\text{Spatial guidance:} \qquad v_{ref} = \pi_k(\mathbf{x}_p, g_k) \tag{4.6}$$

$$\text{Dynamic tracking:} \quad \mathbf{u} = f^{-1}(v_{ref}) + k(v - v_{ref}) \tag{4.7}$$

At the subgoal planning level, the interaction involves selecting a guidance element π_k based on the current system subgoal, g_k, the task goal state, x_g, and prior knowledge about candidate subgoals in the environment, $\tilde{G}_k$. At the spatial guidance level, the

interaction involves specifying a reference velocity sequence $v_{ref}(t)$ based on the agent current position $\mathbf{x}_p$ and the current subgoal, g_k. Finally, the tracking control level describes the regulation of higher-order system dynamics so as to maintain system velocity at the reference velocity. Tracking control is typically modeled using an inverse-dynamics, feed-forward component, $f^{-1}(v_{ref})$, and a feedback component, $k(\mathbf{i}, v_{ref})$.

4.2.4 Gaze Direction

Gaze direction is described by a vector, r_{gaze}, going from the human agent's eye into the environment. It is assumed that human visual perception occurs primarily near the ocular fovea, and therefore gaze direction offers a measurement of agent attentional focus within the environment, and across the available perceptual cues.

4.2.5 Task Structure

The hierarchical levels of agent-environment interaction can be generalized in terms of task structure. This section describes how observed human guidance data is decomposed based on this structure to identify the primitive elements of behavior. Task structure can be described through the following equivalence relations:

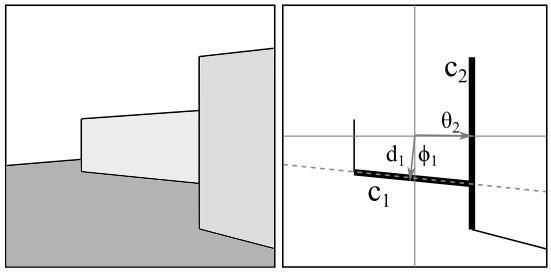
The common subgoal equivalence relates two trajectories, $\overleftarrow{s}_1 \sim_S \overleftarrow{s}_2$ that begin at different spatial locations, meet at a common subgoal, g_i, and continue along the same route from g_i to the final goal, x_g. This equivalence is based on the principle of optimality, stating that optimal trajectories consist of optimal sub-trajectories. Because trajectory subgoals define obstacle avoidance behavior, trajectories that share a common subgoal have the same set of active spatial constraints from that point forward.

The guidance equivalence relates two subgoal partitions, $g_i, \sim_G g_j$, for which environment constraints and system dynamics are equivalent. This equivalence defines classes of trajectory segments that represent the same motion guidance problem, and for which the same set of guidance solution trajectories are applicable. A set Π_{g_i} represents solution trajectories beginning in partition P_i and reaching g_i. Two subgoals are equivalent if a spatial transformation maps the set of guidance solutions, Π_{g_i} onto Π_{g_j}. A spatial transformation is defined by $\Psi : \mathcal{H} \times \mathcal{X} \to \mathcal{X}$, where $h \in \mathcal{H}$ is an admissible rigid-body transformation in the Lie group $\mathcal{H}$. The guidance equivalence is expressed

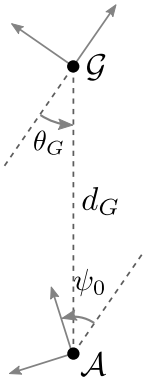
as [18]:

$$\Psi_h(\Pi(g_i)) = \Pi(\Psi_h(g_j)) \tag{4.8}$$

4.2.6 Perceptual Mechanisms



(a) Visual cues.



(b) Guidance geometry

Figure 4.3: First-person guidance.

Perceptual mechanisms describe how cue positions in the environment correspond to quantities perceived by the agent. Perceived quantities include direct measurements, such as the position or velocity of features in the visual field, as well as quantities that are inferred from direct measurements. The measurement model $\mathbf{i} = h(\mathbf{e})$ specifies the perceptual information available from an environment state. In the first person simulation analysis, $\mathbf{e}$ is delineated into a set of discrete cues $\{c_i, \ldots, c_n\}$ that are described by line segments in the workspace. Fig. 4.3a depicts an example set visual cue measurements available in this environment. Each line segment $c_i = \{\theta_i \ \phi_i\}$ can be described by orientation ϕ_i and angular distance from the center of the visual field, θ_i. Inferred measurements include Tau of perceived gaps computed from the divergence of optical flow, and agent angular rate computed from translational optical flow. Because vehicle motion is constrained to the x-y plane, translational optical flow only occurs horizontally in response to vehicle rotation, and is the mean angular rate of vertical edge cues, $\dot{\bar{\theta}}_{vert}$. Tau of each vertical cue is computed based on diveregence, $\tau_i = (\dot{\theta}_i - \dot{\bar{\theta}}_{vert})/\theta_i$.

4.2.7 First-Person Task Definition

Based on this functional model, the human guidance task is formulated in terms of first-person perceptual cues and relevant goal information. Two frames of reference describe agent-environment interaction, $\mathcal{G}$ and $\mathcal{A}$. $\mathcal{G}$ is centered at the goal location, and oriented with the velocity vector at the goal state. The agent must be aware of this reference frame in order to demonstrate a consistent guidance policy based on state relative to the subgoal. $\mathcal{A}$ is a reference frame centered at the agent position, and oriented with the agent's heading. The agent must understand how this frame relates to $\mathcal{G}$ in order to relate perceived cues in the visual field to relevant goal information. In the experimental task presented here, the workspace is a 2D space, $\mathcal{W} \in \mathbf{R}^2$, and is defined in polar coordinates as $\mathbf{x}_G = [\theta_G \; d_G]$ as illustrated in Fig 4.3b.

Based on concepts from optimal control theory, a dynamic motion trajectory consists of a deployment of a sequence of primitive guidance elements, $\Gamma = \{\pi_1, \ldots, \pi_n\}$, Where each π_i is a motion guidance element connecting spatial subgoals, as described in [105]. A spatial subgoal is a transition point between trajectory subsegments that are either free or constrained (not mixed). Each guidance element is then decomposed into a sequence of perceptual guidance elements, $\pi_i = \{d_1, \ldots, d_m\}$, which transition at dynamic subgoals. Dynamic subgoals are points where the set of active dynamic constraints, constraint on velocity states or control inputs, changes.

4.3 Conclusion

This section outlined the goals and hypothesis of the human behavior investigation presented in this work. Goals are based on decomposing observed behavior into primitive elements of interactive behavior at the planning, guidance, and perception levels. This decomposition is based on a closed-loop embedded model of agent task-environment interaction. Primitive elements are defined by a hierarchical set of equivalence classes. The next section will describe specifically how observed data is decomposed and show the resulting primitive interactive elements.

Chapter 5

Information-based Behavior Decomposition

5.1 Introduction

While human senses provide a limited window into the world, an overwhelming number of visual features present themselves in cluttered environments. Only a small subset of these features are necessary or useful for motion guidance, so successful task completion hinges on filtering the feature data and isolating relevant measurement. Humans filter and isolate measurements by focusing attention; both by directing senses, e.g. eye movement, and internal mental focus within a cognitive representation. The ability to focus attention and extract relevant cues from the sensory stream is critical to efficient situational awareness and motion guidance.

This chapter explores the problem of how humans deploy perceptual behavior during motion tasks. Perceptual behavior includes direction of visual attention (gaze), filtering visual information into feature measurements, and relationships between measurements and motion behavior (perceptual guidance). To approach this problem, information and control theory are used to determine the value of specific cue measurements relative to performing a guidance task, such as travel time or risk. This approach is then used to evaluate human performance relative to theoretically optimal behavior. Since real-world environments are cluttered, the optimal information computation may not be tractable in real-time, and may not be biologically feasible for the human brain.

5.1.1 Hypothesis

In this chapter a guidance strategy is defined by a mode of Shannon information transfer [93] between signals, such as feature measurements, vehicle state, and agent actions. Actions, states, and measurements are modeled as random vectors A, X, and Z respectively. Relationships between these signals result in information transfer. The hypothesis explored in this chapter is that modes of information transfer between signals can be observed in experimentally recorded guidance behavior. In addition, these observed modes indicate active perception and guidance strategies used by the subject, revealing how they interact with a task environment scenario.

5.2 Approach

Guidance strategies are investigated from action, state, and cue measurements data collected during the slalom course simulation described in Chapter 3. Agent action consists of vehicle heading and velocity, and system state consists of vehicle position, both of which are directly recorded during the simulation. Cue measurements are not explicitly recorded, but can be recovered from the agent position relative to environment obstacles.

5.2.1 Feature Extraction

Environment measurements consist of features in the visual field. The human visual system is known to extract several types of feature measurements from the environment, such as changes in objects size (looming) [77], optical flow [76], parallax [106], motion, etc. This chapter considers a simple environment that contains only linear edge cues as illustrated in Fig. 4.3a in Chapter 4. A measurement of cue c_i consists of orientation ϕ_i and distance d_i from center of the visual field $z(c_i) = [d_i, \theta_i]$.

At each point in time, a set of p cues are available for measurement within the visual field, $C = \{c_1, \ldots, c_p\}$. Cue measurements at a time-step are a function of the agent position, environment configuration, and projection of the environment onto the visual field. In this work, the environment configuration and projection are assumed constant, so the vector of cue measurements is described by a measurement model $z = h(\mathbf{x})$.

5.2.2 Guidance Strategies

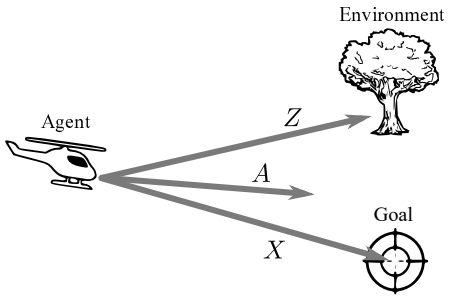


Figure 5.1: Visual cue, state, and action.

Model-based Guidance

A guidance policy can be represented by a spatial optimal velocity vector function (VVF), $v_{ref} = V^*(x_p)$ and cost-to-go (CTG) function, $J^*(x_p)$. Evaluating these functions requires knowledge of agent configuration x_p, which may not be directly observable from cue measurements. An agent must estimate an internal representational system state based on available measurements and prior knowledge of environment dynamics. Model-based guidance such as this involves three random variables, $Z \rightarrow X \rightarrow A$, representing the information flow between environment measurements, internal model state and actions.

Note that it is unlikely that the internal representational model state X use by humans is defined in terms of engineering units in Euclidean coordinates. Since instantaneous time to gap closure, τ and $\dot{\tau}$ have been used to describe perceptual guidance [17, 87], humans may also represent their internal state model in terms of τ.

Non-representational Guidance

Work investigating embodied or ecological approaches to human behavior suggests humans do not necessarily represent a complete internal state in this way, but instead use patterns within a guidance policy to simplify control [96]. This non-representational approach involves only two variables, $Z \rightarrow A$, describing the information flow directly from cue measurements to agent actions. Tau theory [17, 87] suggests a non-representational

approach in which humans perceive the time-to closure of perceptual gaps in the visual field. This can apply to object interception tasks such as grasping or catching a ball, as well as navigation tasks that involving moving toward or around obstructions.

5.2.3 Information Transfer

Signal Relationships

Environment measurements, system states, and agent actions are interrelated during guidance as illustrated in Fig. 5.1. Measurements are related to agent state by a measurement model, $z = h(x)$, which is based on spatial the relationship between agent and cue positions in the environment. Similarly, a model-based guidance policy describes the relation between actions and states, $v = k(x)$, and a perceptual guidance policy describes the relationship between actions and cue measurements, $v = k(z)$. Guidance policies exist internally within an agent and must be empirically determined. The relationships between actions, states, and measurements define the information transfer between the variables at a point along a trajectory.

Mutual Information

Mutual information quantifies the information that one random variable contains about a second in bits. Mutual information between random variables X and Y, $I(X;Y)$ is defined in terms of the marginal entropy $H(X)$ and joint entropy $H(X,Y)$ as:

$$I(X;Y) = H(X) + H(Y) - H(X,Y) \tag{5.1}$$

Mutual information is expressed in terms of correlation when random variables are normally distributed:

$$I(X;Y) = -\frac{1}{2}\ln\left(1 - \left(\frac{\sigma_{xy}}{\sigma_x\sigma_y}\right)^2\right) \tag{5.2}$$

Where σ_{xy} is the covariance between X and Y and σ_x and σ_y are standard deviations. This formulation of mutual information assumes that measurement and process noise are normally distributed.

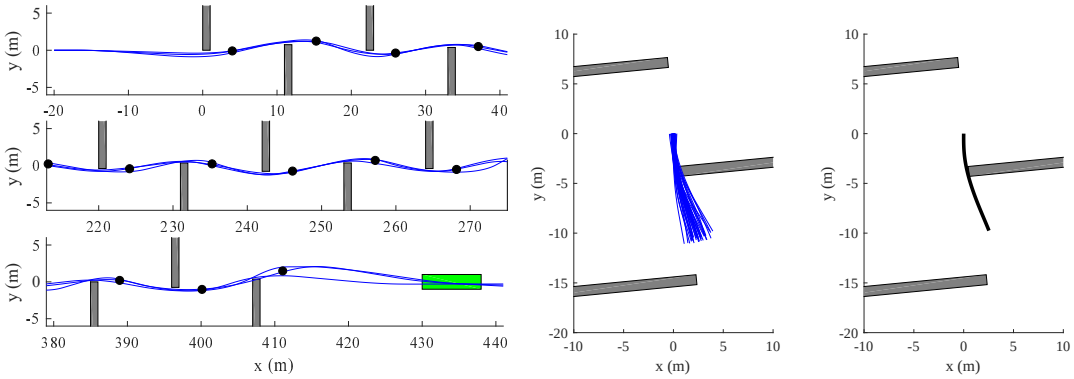
Mutual information can be used to investigate the information flow between signals at each point along a guidance trajectory, in order to reveal active guidance strategies. A signal relationship—such as a measurement model or guidance policy—together with the mutual information between random variables indicates information flow between them. However, multiple samples of each signal are needed to compute correlations at an instance in time on a trajectory. The next section presents the experimental system used to provide this data. Multiple trials of a similar subtasks are considered in aggregate, providing the required sample of trajectory signals.

5.2.4 Gaze and Cue Attendance

Human vision consists of foveal and peripheral regions [107]. Visual acuity is highest in the foveal region, and because of this, humans must direct their gaze towards features, in order to see fine detail or perform identification tasks such as reading. The peripheral region, while not as good at perceiving fine detail, is good at detecting salient features, patterns and motion, which are important for guidance. Visual motion perception in the peripheral region presents a challenge because gaze tracking measurements provide the location of only the center of the subject's visual field. Since surfaces in the environment do not have texture, subjects cannot perceive optical flow in peripheral vision. They may however still perceive motion of line-segment cues outside of the fovea. This analysis will consider the measurements available from all cues within the visual field—the computer simulation display—to identify guidance strategies. Gaze measurements are used to indicate where the subject is most likely focusing attention.

5.3 Data Analysis

Fig. 5.2a shows the complete set of recorded data for one subject navigating from a start location, around the sequence of 12 walls, and passing through the goal doorway. The objective of analysis is to compute statistics between actions, states, cue measurements, and subject gaze behavior during this task. Multiple trials, and multiple object-avoidance maneuvers within each trial provide a set of behavior samples that can be used to compute statistics among action, state, and cue measurements data. Guidance and perception patterns are identified by computing covariances among these



(a) Trajectory data. (b) Segments. (c) Average.

Figure 5.2: Aggregate data for similar subtasks.

signals at a points in time along a trajectory.

An additional objective is to observe statistical relationships between human gaze motion and guidance strategies. Since gaze only attends a single cue at a time and frequently jumps between multiple cues, it is necessary to observe several trials of a task to capture aggregate cue attendance behavior.

5.3.1 Subgoals

Subgoal locations are chosen to divide the set of trajectories into segments as shown in Fig. 5.2a. In previous work [16] subgoals were identified in first-person behavior data, based on properties introduced by Kong and Mettler [12]. Subgoals emerge as the result of equivalence relations between solution trajectories. The subgoal equivalence relates trajectories that meet at a subgoal and remain together until they reach the goal. This relation is not applicable to the slalom data, since all trajectories follow the same general course. The guidance primitive equivalence relates trajectories through a group action. Trajectories are equivalent if a group action transforms one to the other, which occurs when both are generated from the same guidance policy. Previous work [105] gives necessary conditions that must be met at valid subgoal locations. Subgoals must lie at points that are both on constraint boundaries, and where the optimal velocity vector is tangent to the constraint boundary. At non-smooth constraint boundary locations, "tangent" means that the optimal velocity vector is not directed into or out

of the constraint boundary. In the simulation data, subgoals are identified as the point where trajectories pass the closest to each obstruction. This satisfies necessary conditions. In addition, it results in trajectory segments that satisfy the guidance primitive equivalency.

5.3.2 Transformations

Trajectory segments between subgoals represent paths around walls with varying offset distances. Segments with similar offsets are grouped together as equivalent guidance primitive elements. This equivalence is defined by a guidance primitive transformation ($\sim_g$) that relates elements within this offset group. The transformation consists of a translation, rotation, and reflection which superimposes the subgoal location and velocity vector of equivalent elements. A trajectory element S_{g_k} connects subgoals g_{k-1} and g_k, and contains a sequence of trajectory points, $\{s_1, \ldots, s_n\}$. A trajectory point at time-step i contains state, action, control inputs, cue measurements, gaze position, and a time reference, $s_i = [x_i, v_i, u_i, z_i, g_i, t_i]$. An element is defined for time-step indices $i \in [i_0, i_g]$ between the bounding subgoals. The transformed elements are plotted along with a representative set of environmental obstructions in Fig. 5.2b. This data is used to analyze motion guidance and perceptual behavior across a large number of guidance element samples.

5.3.3 Average Trajectory

A reference point is needed at each time step to compute trajectory signal covariances and gaze attention behavior from the aggregate set of elements. An average trajectory S_j^* is computed from the set of guidance elements in wall offset group j, as depicted in Fig. 5.2b and plotted in Fig. 5.2c. The average trajectory is computed by referencing sub-trajectory data to an average time-vector, $t_{ave} = [t_0, 0]$, where t_0 is the median initial time-to-go of the constituent elements. Continuous trajectory states such as position and velocity are averaged together across constituent trajectories at each average time-to-go index k.

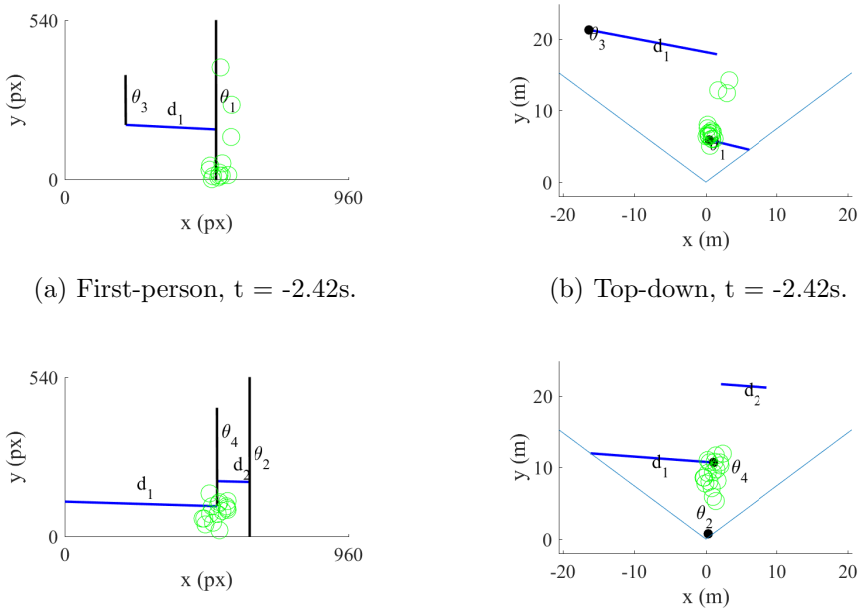
$$x_{ave}(t_{ave}(k)) = \frac{1}{n}\sum_{i=1}^{n} x_i(t_{ave}(k) \tag{5.3}$$

In Eqn. 5.3, x_i is a signal in the i-th constituent trajectory. Only constituent elements that exist at time $t_{ave}(k)$ are included in the average, i.e. shorter segments are excluded. Gaze data is stored as a set of gaze points consisting of the union of individual gaze points from each constituent trajectory.

$$g_{ave}(t_{ave}(k)) = \bigcup_{i=1}^{n} g_i(t_{ave}(k)) \tag{5.4}$$

Note that gaze position is recorded in the global frame, so actual gaze position is preserved even though the observer position on the average trajectory is different from the actual observer position on the constituent element. Because of this, relative gaze position and angles are distorted if they are measured from the average trajectory position.

5.3.4 Cue Attendance



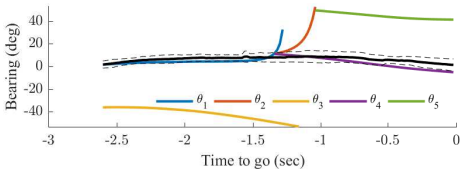
(a) First-person, t = -2.42s.

(b) Top-down, t = -2.42s.

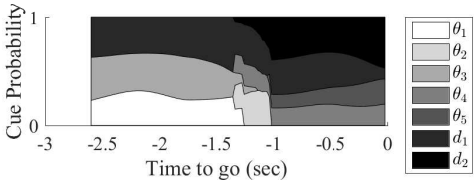
(c) First-person, t = -1.02s.

(d) Top-down, t = -1.02s.

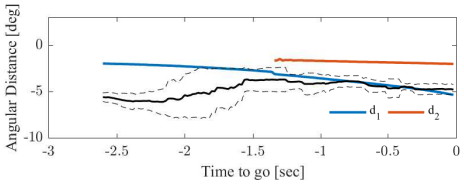
Figure 5.3: Visible cues and gaze points.



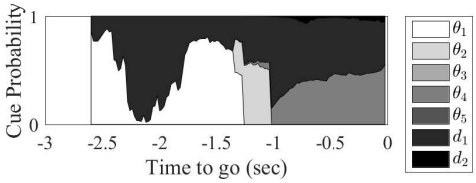
(a) S1: vertical edge bearing measurements.



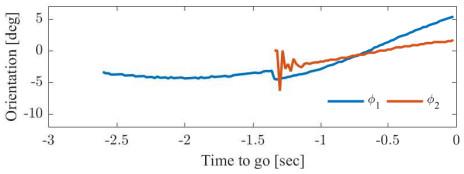
(b) S1: uniform cue attendance distribution.



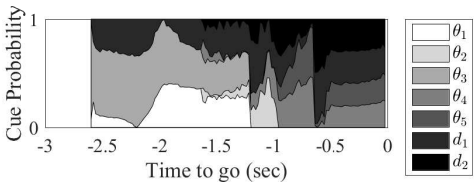
(c) S1: horizontal edge distance measurements.



(d) S1: observed cue attendance.



(e) S1: horizontal edge orientation measurements.



(f) S1: information-based cue attendance.

Figure 5.4: Measurements and gaze behavior along the average trajectory.

Cue Measurements

The set of cue measurements relative to agent position is computed from obstructions at each time-step, based on the set of visible cues. Cue visibility is computed in a 2-D top-down coordinate frame (Figs. 5.3b and 5.3d). Vertical wall edges are considered from left to right. Whenever the edge of a new wall is encountered, edge distance is used to determine if it is in front of the previous wall. This results in a set of visible vertical edges, and horizontal edge line-segments connecting them. Finally, line endpoints are projected into the first-person view. The extents of each line segment are checked and clipped as needed to fit within the first-person field of view (Figs. 5.3a and 5.3c). For each line segment within the field of view, bearing, distance and orientation are extracted as measurements, and plotted in Figs. 5.4a, 5.4c and 5.4e.

Vertical edge bearing measurements follow two primary patterns. First, cues near the edge of the field of view diverge outward as they are passed. For example, the far left

corner θ_3 diverges negative to the left side of the screen. Between -1.5 and -1.0 second, the corners on the left side of the close wall θ_1 and θ_2 diverge to the right. Second, cues become visible from behind other cues, and their measurements appear to divide at these points. For example, cues θ_2 and θ_4 emerge from behind cue θ_1 at about -1.5 seconds, and cue θ_5 appears from behind θ_2 at -1.0 seconds.

Horizontal distance and orientation measurements show less motion in general—distances and orientations stay within a narrow range for this course. Horizontal edge 2 (d_2 and ϕ_2) appears at about -1.5 seconds. Note that the orientation measurement ϕ_2 is noisy immediately after it appears. The noise is due to numerical errors, since initially only a very small portion of edge 2 is visible. As the visible portion of edge 2 becomes longer, the orientation measurement becomes more steady.

Cue Attendance Distribution

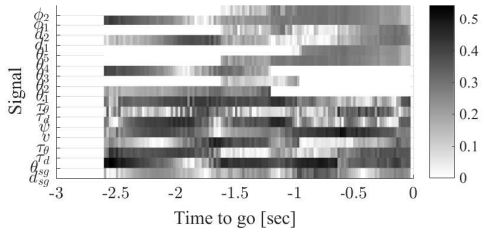
Cue attendance probability is computed as a normal distribution around a cue location, based on the expected angular size of the foveal region. A 2-sigma bound of 5 degrees is used to model the foveal acuity distribution. Variations in actual acuity ,with cue position relative to the fovea, are accounted for, with the probability of cue attendance resulting in less information extraction from that cue.

The cue attendance probability of each gaze point at each time-step gives a distribution of attendance across cues, $p(c_i, t)$. For each cue, attendance probabilities are summed over gaze points to give a total cue attendance weight. The cue attendance distribution is plotted in Fig. 5.4d for S1. For comparison, Fig. 5.4b depicts a uniform distribution of attendance across all visible cues. The result highlights the difference in the human subject's perceptual strategy during this task, from a reference strategy of attending all regions of the visual field with equal probability.

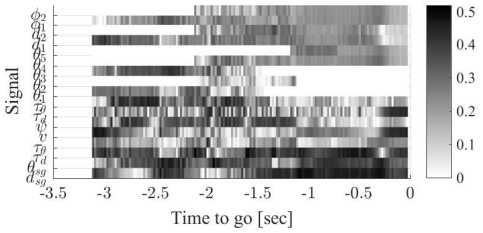
5.3.5 Perceptual Guidance Strategy Analysis

Principle Component Analysis

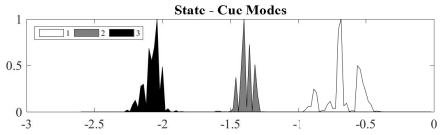
Figs. 5.5a and 5.5b map the primary principle signal correlation eigenvector vs. time on the average trajectory. High points on the map represents information transfer between signals at that time. Changes in the principle eigenvector over time indicate transitions



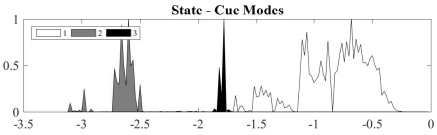
(a) S1: Primary signal component.



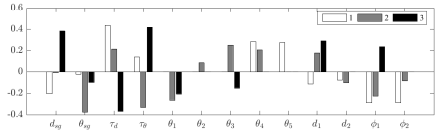
(b) S3: Primary signal component.



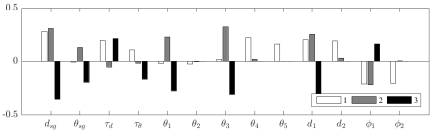
(c) S1: state-cue modes vs. time.



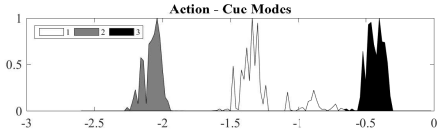
(d) S3: state-cue modes vs. time.



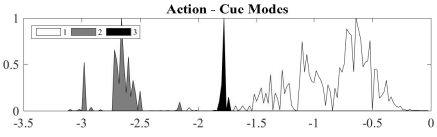
(e) S1: state-cue modes.



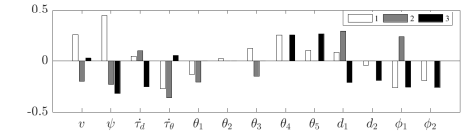
(f) S3 state-cue modes.



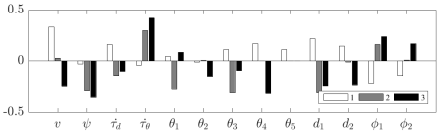
(g) S1 action-cue modes vs. time.



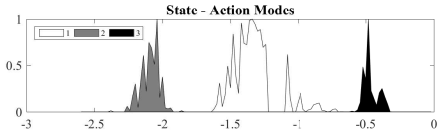
(h) Subject 3: action-cue modes vs. time.



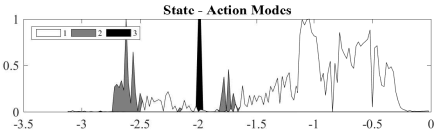
(i) S1 action-cue modes.



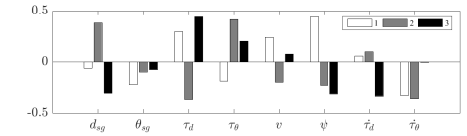
(j) S3 action-cue modes.



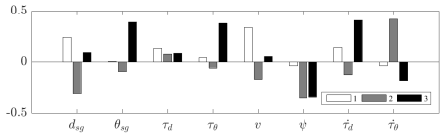
(k) S1 state-action modes vs. time.



(l) S3 state-action modes vs. time.



(m) S1 state-action modes.



(n) S3 state-action modes.

Figure 5.5: Guidance strategies for subjects 1 (left) and 3 (right).

between information transfer modes.

Primary Mode Clustering

A subspace clustering approach is used to identify specific guidance strategies. Signals are first partitioned into groups of actions, states, and cue measurements. Each combination of two groups identifies an information transfer or guidance strategy. State-cue modes are reflected in $P_1 = \{X_\tau, X_z\}$ and indicate information about agent state contained in cue measurements. Primary components represent the strongest relationships between cues and states, which may indicate the cue measurements a subject would favor to gain state information. These relationships provide situational awareness to the agent. Action-cue modes are reflected in $P_2 = \{X_a, X_z\}$ and indicate the information about agent actions contained in cue measurements. Primary components represent the strongest relationships between perceived cue measurements and actions, indicating modes of perceptual guidance. Action-state modes are reflected in $P_3 = \{X_a, X_\tau\}$ and indicate the information about agent actions contained in agent state. Primary components represent the strongest relationships between agent actions and states, indicating use of Tau-guidance, or other model-based guidance policies.

To understand the evolution of active information transfer modes along the trajectory, the sequence of principle eigenvectors for each group combination are clustered across time. Three eigenvector clusters are identified, x_1, x_2, and x_3. Clusters are plotted in Figs. 5.5e, 5.5i, and 5.5k for S1 and in Figs. 5.5f, 5.5j, and 5.5l for S3.

Primary Mode Intensity

Once primary modes are identified in each information transfer group, the intensity of each mode is computed at each time on the average trajectory. Mode intensity is computed as density of a normal distribution with statistics based on the scatter of points in a cluster. The intensity of the three primary modes is plotted vs. time to go in Figs. 5.5c, 5.5g, and 5.5m for S1, and in Figs. 5.5d, 5.5h, and 5.5n for S3. Mode intensity reveals organization in information transfer and guidance strategy modes across time, i.e., if specific modes are used at specific times during the trajectory.

5.4 Results and Discussion

5.4.1 Interpretation of Guidance Modes

This section describes the results shown in Fig. 5.5 in terms of identified guidance modes. Correlated signals are compared to expected results given by model, tau, and perceptual guidance based approaches. This section focuses on results from S1.

State-cue Modes

(Figs. 5.5c, 5.5e) All three group combinations for S1 exhibit three peaks of intensity. Each state-cue mode shows correlation between system states and some combination of visible cues indicating channels over which the agent may gain situational awareness. Peak 1 is from -2.25 to -2 seconds, peak 2 from -1.5 to -1.25 seconds, and peak 3 from -1 to -0.5 seconds. Peak 1 indicates information transfer between d_{sg}, and both τ states, and the visible cue measurements at that time. Peak 2 indicates information transfer between θ_{sg}, τ states, and visible cues. Peak 3 indicates information transfer primarily from τ_d.

Action-cue Modes

(Figs. 5.5g, 5.5i) Each action-cue mode shows information transfer between visual cues and agent actions, indicating possible perceptual guidance strategy. Both peak 1 and peak 2 show information transfer between visible cues and both actions. Peak 3 shows only information transfer between cues and Ψ, indicating that turn rate may be the primary perceptual guidance mode during that phase.

Action-state Modes

(Figs. 5.5m, 5.5k) Each state-action mode shows information between agent states and action, indicating a possible model-based guidance strategy. For correlation in this group combination to represent an actual guidance mode, the agent must also have good situational awareness (state-cue infomation transfer) for the state variables involved. From a controls perspective, situational awareness means that the involved agent states are a good representation of true state values.

Peak 1 indicates information transfer between actions v and ψ and all states, indicating a possible Tau or model-based strategy. Peak 2 shows much higher information transfer between θ_{sg} and ψ, indicating that turning control may become more important during this phase. Peak 3 indicates a correlation betwee $\dot{\tau}$ and d_{sg} and τ_d, indicating a possible Tau guidance mode in the distance gap.

Summary of Interpretation

The above analysis is summarized in Table 5.1:

Table 5.1: Guidance strategy summary.

Time (sec)	v-mode	ψ-mode
-2.3 to -2.0	None	Tau
-1.5 to -0.5	Tau	Percept
-0.5 to 0.0	Tau	Percept

5.4.2 Mode Characteristics

Model-based vs. Nonrepresentational

Mode analysis suggests that subjects employ a combination of model-based and nonrepresentational guidance strategies during the trajectory segment. Subject may also employ different strategies in different states, such as turning/bearing guidance vs. speed/distance guidance. The plots in Fig. 5.5 show only the three primary modes for each group combination—other modes are present, and are needed to develop a complete picture of agent behavior. Additional investigation is required to understand why a subject uses a specific set of guidance strategies, and how this is related to environment structure and system dynamics.

Perception-action Coupling

State-cue information transfer modes illustrate the perception-action information balance. In order to achieve adequate situational awareness, the agent must generate motion which results in high correlation between cues and system states. Good SA

depends on agent motion in addition to perceptual behavior, a key tenant of embodiment theory. This highlights the closed-loop interaction between perception and action; results suggests that cue structure influences agent actions to maximize information flow. Intuition of situational awareness suggests that cue-state information transfer is important, even if a perceptual guidance strategy is being used. SA may be required to choose and deploy perceptual guidance modes in a task scenario.

Tau Guidance

State-action mode analysis indicates that a tau-guidance mode may be used at the end of the trajectory for speed control. During this phase, $\dot{\tau_d}$ and τ_d, and v are correlated. Tau-guidance requires that vehicle control inputs are not saturated during a gap closure profile. Tau-guidance may occur only at the end of the trajectory element if control inputs are saturated earlier in the element. Once control limits come below saturation limits, they can be modulated to close the gap based on constant $\dot{\tau}$.

5.5 Discussion

This chapter presents a method for identifying primary information transfer modes in human perception and guidance behavior. Information transfer modes correspond to guidance and perception strategies, such as perceptual guidance, tau guidance or state measurement. Dominant guidance modes are plotted vs. time, showing clear phases where distinct modes are active. Gaze measurements indicate how a subject distributes attention across visible environment cues during a task. The cue attendance distribution is compared to active perceptual modes to both validate whether a mode may be active, and understand how gaze behavior relates to active guidance modes. Results show a poor match between predicted cue distribution based on active perceptual modes, indicating that gaze motion serves other functions, and cue information is likely obtained in the peripheral vision region.

Perception-Action Coupling

These results highlight the closed-loop coupling of perception and action. The value of visual features, in terms of situational awareness, depends largely on agent motion. Cue

measurements contain more information when they are highly correlation with agent states and actions. This result has application to human-machine interface design. Humans may be able to maintain higher situational awareness during autonomous or augmented modes if the system maintains high correlation between visual cues and vehicle motion. This can be achieved not only by adding artificial cues, but also by modifying the vehicle motion profile relative to existing environmental cues.

Gaze Attention and Visual Information

Fig. 5.4f above shows an estimated gaze cue attendance distribution for S1, based on maximizing information gain about system states. This distribution is notable, in that it is different from the observed distribution of S1's actual cue attendance. This suggests that actual gaze direction is not as important as originally thought for measurement of visual cues in guidance. Data from previous work [16] shows that a significant portion of gaze behavior appears to trace the anticipated future vehicle trajectory rather than attend cues.

Chapter 6

Constraint-based Behavior Decomposition

6.1 Guidance Behavior Decomposition

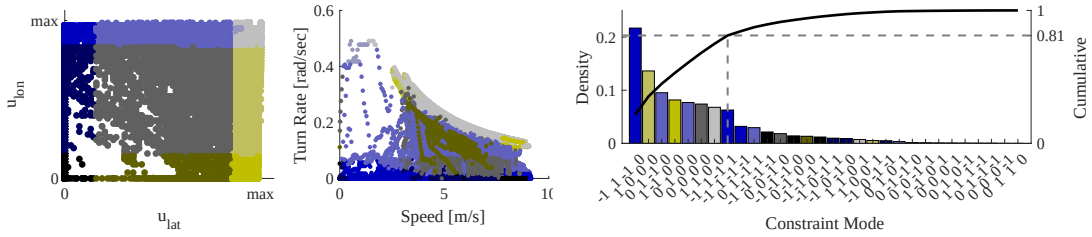
6.1.1 Overview

Behavior is decomposed based on two hypotheses on the nature of human dynamic motion: first, that humans use a small number of dynamic control modes to generate the range of guidance behavior throughout a task. A control mode corresponds to a unique set of agent-environment interaction patterns. This hypothesis predicts that mode transitions occur in typical sequences that can be modeled as a motion-primitive maneuver automaton [42, 18]. The second hypothesis is that dynamic control modes occur along lower-dimension, subspace manifolds of the full agent-environment system. Manifold dimension can be reduced in the input space by defining meta-controls [15], in latent states by using a hierarchical task representation [7], or in perception by using visual quantities such as Tau [17].

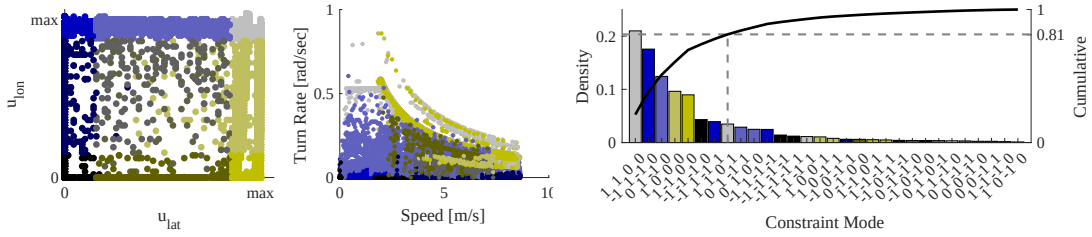
This decomposition is a type of temporal subspace clustering problem [108, 92], in which the hypothesis is that each time sample in the behavior sequence belongs to one of a discrete set of subspace manifolds. In this work, the system state belongs to the product space of perceptual measurements and control actions, emphasizing the hypothesis that subspace clusters describe perceptual guidance interaction patterns used

by the agent.

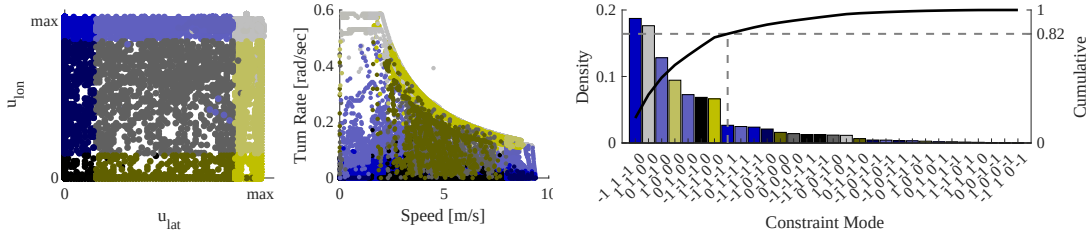
Behavior decomposition consists of two steps: first identifying subspace manifolds across the entire behavior sequence and second, clustering the identified manifolds into a small number of dynamic control modes that describe the majority of subject behavior. A key contribution of this work is to identify subspace manifolds based on transitions in the set of active constraints at each time sample, along with relationships between unconstrained signals [109].



(a) Subject 1.



(b) Subject 2.



(c) Subject 3.

Figure 6.1: Dynamic state scatter and relative density.

6.1.2 Constraint Class Identification

To investigate active constraints during guidance behavior, a dynamic state is defined as the union of control inputs, $\{u_{lat}, u_{lon}\} \in \mathcal{U}$ and configuration-state rates, $\{v, \omega\} \in \mathcal{V}$, $\mathcal{D} = \mathcal{V} \cup \mathcal{U} = \{u_{lat}, u_{lon}, v, \omega\}$. System dynamics specify constraints on these four

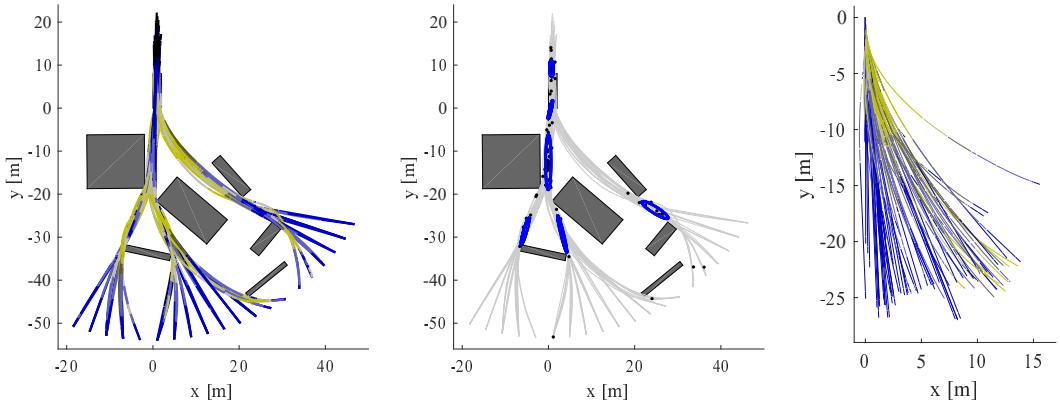
states. Each dynamic state along a trajectory, $\mathbf{d}(t) \in \mathcal{D}$ satisfies dynamic constraints when $\mathbf{c}(\mathbf{d}) < \mathbf{0}$. The dynamic constraint state for a trajectory point is defined as $\mathbf{c}(\mathbf{x}) = [c(u_{lat})), c(u_{lon}), c(\omega)c(v)]$, where $c(x)$ is the constraint state of an individual parameter value x and is defined as:

$$c_i(x) = \begin{cases} 1, & \text{if } d_i - d_i^{max} \approx 0 \\ -1, & \text{if } d_i^{min} - d_i \approx 0 \\ 0, & \text{otherwise} \end{cases} \tag{6.1}$$

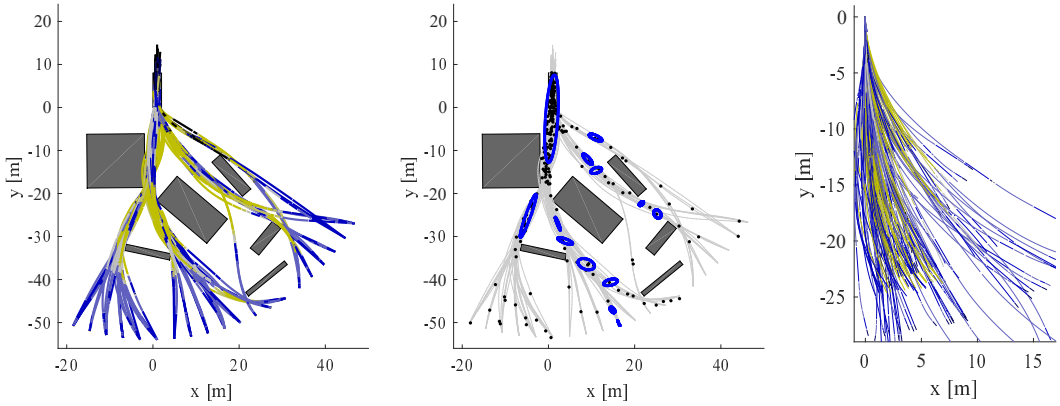
Three values are used ($\{-1, 0, 1\}$) to explicitly account for minimum and maximum constraint activation for a signal being mutually exclusive. Fig. 6.1 shows a scatter of trajectory points in both a control domain and dynamic state domain, as well as a histogram depicting relative frequency of each constraint class. Color on both plots indicates the constraint state of the control inputs for that point (u_{lat} in yellow and u_{lon} in blue). This constraint state definition transforms a trajectory $\mathbf{x}(t)$ for $t \in [0, T]$ to a constraint-state trajectory, $\mathbf{c}(t) \in \{-1, 0, 1\}^4 \times T$. The left column of Fig. 6.2 shows the constraint state along each recorded trajectory.

6.1.3 Spatial Subgoal Identification

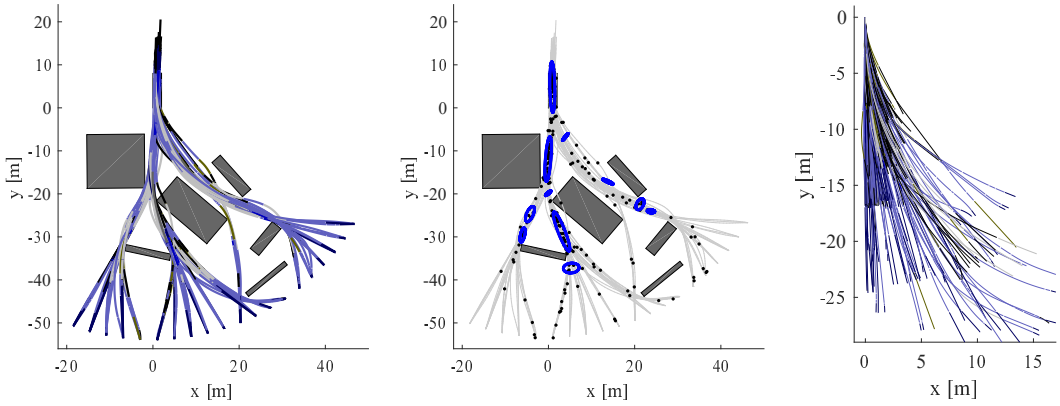
Subgoals are introduced in prior work as being based on the common subgoal equivalence [12] and as a property of optimal guidance solutions [105, 110]. Subgoals are important for task planning because they represent transitions between regions of similar active spatial constraints. A key point is that spatial subgoals are invariant over a task domain. A complementary approach to subgoal identification is based on the idea of relevant goal information (RGI) [103] and MDP policy options [49]. This RGI approach suggests that an agent uses information about their position relative to a subgoal to select control actions. This concept specifies that subgoals occur at trajectory points where there is an abrupt change in the perceived information an agent uses to specify control actions, and a corresponding abrupt change in agent actions. Prior work has shown that human motion is described as a sequence of perceptual gap closures in the sense of Tau-theory [17, 87]. Under this model, reaching a subgoal corresponds to a gap closure, and an abrupt RGI transition to the next subgoal gap.



(a) Subject 1.



(b) Subject 2.



(c) Subject 3.

Figure 6.2: Trajectory constraint classes and subgoal candidate clusters.

Using this idea of RGI transitions, subgoal locations can be estimated from the constraint class sequence along each trajectory. The hypothesis is that subgoals occur at the transitions from rectilinear to turning motion. This is shown by considering an agent approaching a subgoal, where the time to reach the subgoal is τ, the perceived relevant goal information gap is z_{RGI}, and the control action $a = k(z_{RGI})$ is a function of relevant goal information. As $\tau \to 0$, $z_{RGI} \to 0$, and $a \to a_0 = k(0)$. At the instant that $\tau = 0$, z_{RGI} will transition from zero to some non-zero value that depends on the RGI for the next subgoal, and a will transition from a_0 to some other value, causing the vehicle to begin turning. These transitions from rectilinear to turning motion are identified from the constraint state as a transition in d_1 or d_2 from zero to either -1 or 1, and indicate subgoal candidate locations. The common subgoal equivalence predicts that these constraint transition points correspond to the invariant set of common subgoals.

The middle column of Fig. 6.2 shows that subgoal candidate points are closely clustered, and occur at points where multiple trajectories join together.

6.1.4 Guidance Behavior Identification

The identified subgoal locations can be used to divide trajectories into segments. The hypothesis is that because subgoals locations depend on spatial task constraints, the set of segments connecting subgoals are spatially unconstrained, and only depend on dynamic system constraints. Previous work observed that symmetries in system dynamics define equivalence relations, mapping the set of behavior segments into a common dynamic domain [19]. Based on these symmetry transformations, each segments is shifted and rotated so that it terminates at the origin, with velocity in the positive y-axis. The right column of Fig. 6.2 shows the resulting transformed trajectory segments, showing a common set of guidance behavior that is used across the constrained task domain. In addition, these results are significant because the set of segments are consistent with respect to the Markov property, i.e. trajectories do not intersect, and velocity is only a function of position relative to the goal. Prior work showed that this aggregate set of guidance behavior can be modeled by a spatial cost-to-go function [19]. This invariance across a task domain observed in human behavior is significant both because it is known to be a property of optimal behavior [14] and because it provides a reduction in motion planning complexity [10].

6.1.5 Semantic Control Modes

The set of trajectory segments are spatially unconstrained, but are still subject to dynamic state constraints. The objective is to identify a set of functional control modes, m_i for $i \in [1, n]$ that describe the aggregate set of guidance trajectories. Each trajectory is then described as a sequence of functional modes. Control modes are obtained by clustering together constraint classes that are functionally similar in terms of dynamic constraint state and relationships between free variables. Functional control modes have semantic meaning, such as "turning" or "linear" motion that are relevant to the subject in the context of the guidance task.

Graphical Modeling

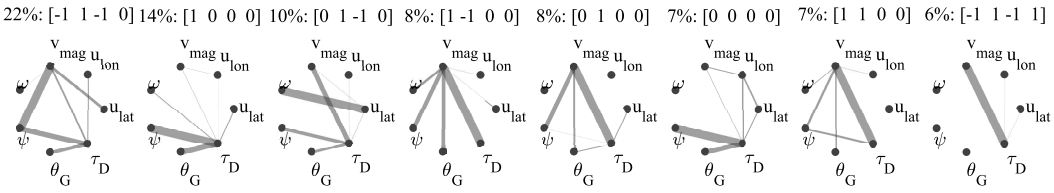
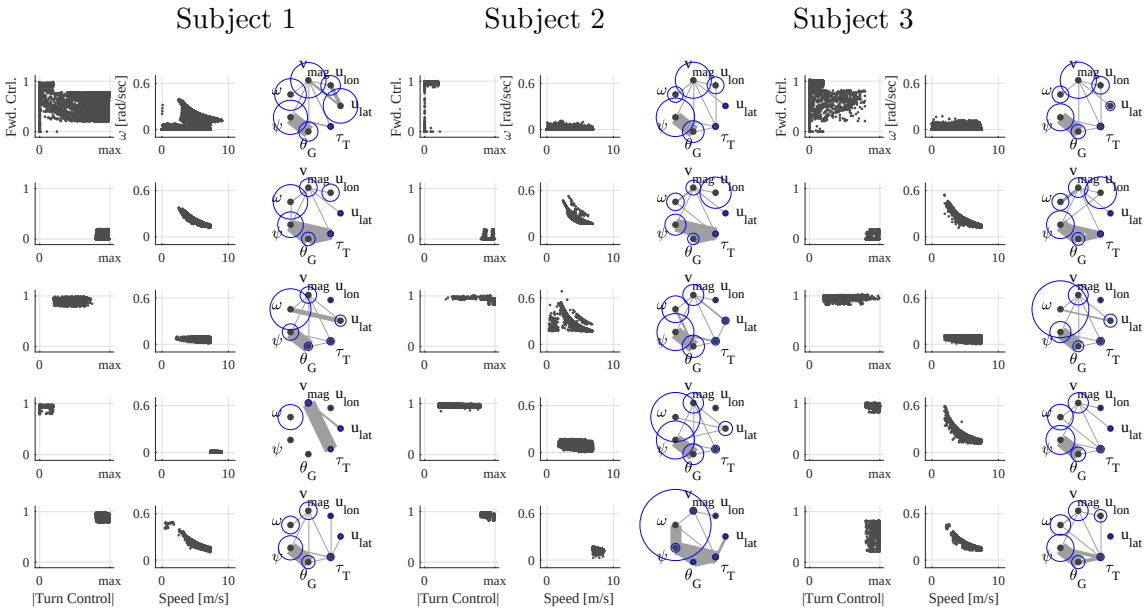
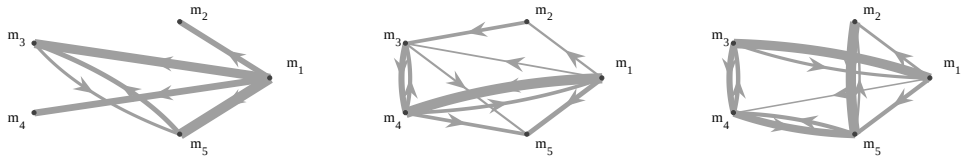


Figure 6.3: Sparse MRFs depicting constraint class signal relationships.

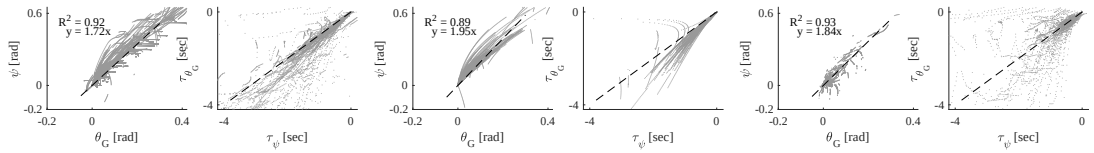
Functional equivalence between two segments of behavior involves similarity in the relationships between free variables during that segment. In this section, graphical modeling is used to determine the signal relationships present within segments belonging to each constraint mode. These relationships include known dynamic and kinematic relationships in the vehicle-environment model, as well as perceptual guidance relationships implemented by the human operator. A Markov random field (MRF), with graph edge weights defined as the inverse covariance of the adjacent random variables, models the joint probability distribution between guidance signals. A sparse MRF is estimated from the set of data belonging to each constraint class using the Matlab SLEP package [90]. Fig. 6.3 depicts graphical models describing the eight most common constraint classes.



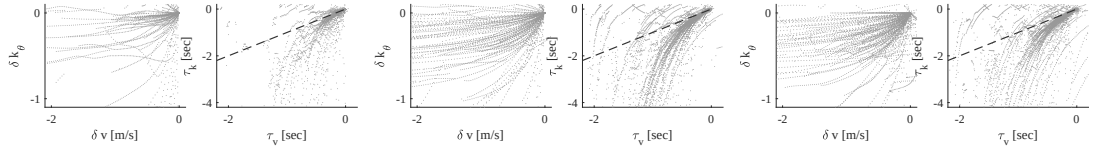
(a) Functional mode point scatters and signal relationships.



(b) Mode transition graph.



(c) Turning behavior.



(d) Non-turning behavior.

Figure 6.4: Functional mode clusters and Tau coupling.

Control Mode Clusters

Next, constraint classes are clustered to identify a small number of functional control modes. Functional similarity is formulated as a combination of MRF graph edge similarity and constraint mode similarity. Edge similarity is quantified as the Jaccard index of graph edge sets. Constraint mode similarity is the L_1-norm of the difference between constraint states:

$$S_m(i,j) = min(E_i, E_j)/max(E_i, E_j) - w * ||c_i - c_j||_1 \tag{6.2}$$

Eqn. 6.2 is the similarity metric, with weight w defining the trade-off between edge similarity and constraint mode similarity. Constraint modes are hierarchically clustered using this metric, resulting in five control modes. Fig. (6.4) shows the control and dynamic state clusters for each mode, and the corresponding MRF showing mode signal relationships.

Control Mode Transitions

Using these identified functional mode definitions, each trajectory is transformed into a sequence of modes. The hypothesis is that motion behavior is based on an ideal, common, repeating transition mode sequence, for which observed mode sequences are a noisy measurement. Fig. 6.4b shows the observed mode transition probabilities for each subject. While edges connect different mode numbers in different subjects, they often represent similar functional transitions. For all subjects, mode 1 includes rectilinear, accelerating motion, and in the transition graph, appears as a source, with many transitions occurring from this to other modes, suggesting that subjects begin by accelerating without turning. Note that some transition sequences do not appear in the graphs because they are not possible. For example, subjects 1 and 2, cannot transition directly between modes 2 and 5, because the control signal must pass through other modes. Other transitions occur frequently in all subjects. From mode 1, subjects most often transition into mode 3 or 4, which is a high-speed turning mode. Subjects less frequently transition into modes 2 or 5, which are either maximum-speed rectilinear motion, or maximum-turning. These functional transitions will be used to infer a common mode-transition model in the next chapter.

6.1.6 Control Modalities

The identified functional modes each show mixtures of common perception-action signal relationships. To identify these common related signal groups, or control modalities, groups of related signals are identified in each control mode. Signal relationships fall into two categories: vehicle dynamic modalities that are defined by the vehicle system model, and agent control modalities. Agent control modalities are perceptual guidance relationships that the operator uses to generate guidance behavior.

Table 6.1 describes four control modalities: steering, speed-turnrate, a turnrate-control, and braking-control. The first and last of these are perceptual guidance relationships that are implemented by the operator. The second and third modalities are system dynamic relationships, indicating which constraints are currently active. Fig. 6.4c contains scatter plots, showing a common relationship between ψ and θ_G, which also appears as edges connecting these signals in the graphical models in Fig. 6.4a. This relationships is consistent with a τ-coupling between agent heading and goal bearing gap. Fig. 6.4d contains a scatter of velocity vs. steering ratio, suggesting that as a subject approaches the transition from rectilinear to turning behavior, they close simultaneous gaps in speed and in steering ratio.

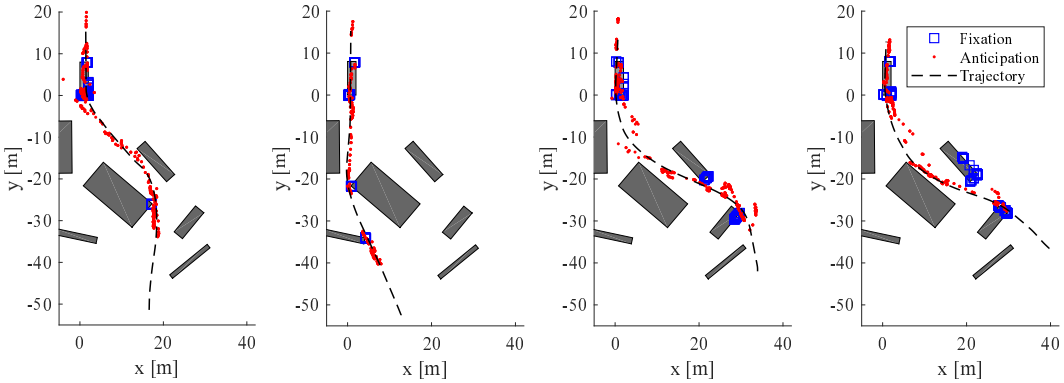


Figure 6.6: Smooth-pursuit gaze behavior clusters.

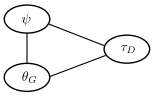
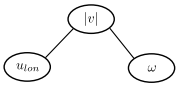
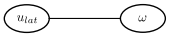
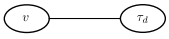
Modality	Relationships	Type
	$\tau_\psi = k_\psi \tau_d$ $\tau_\theta = k_\theta \tau_d$ $\psi_{ref} = k_\theta \theta_G$	Agent
	$u_{lon} = v_{ref}$ $+k(v - v_{ref})$ $v_{ref} = a_y^{max}/\omega_{ref}$	System
	$u_{lat} = \omega_{ref}$	System
	$\tau_v = k_v \tau_d$ $\tau_k = k_k \tau_d$ $\delta v = k_b \delta k_\theta$	Agent

Table 6.1: Control modalities.

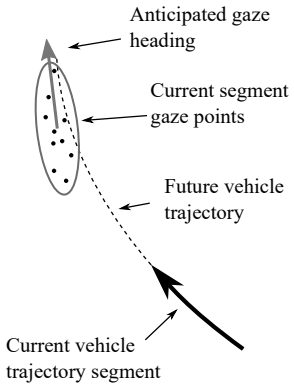
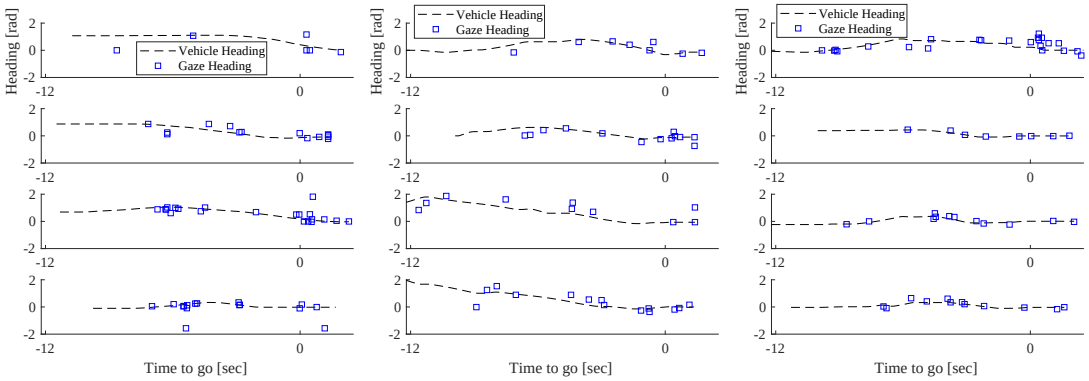


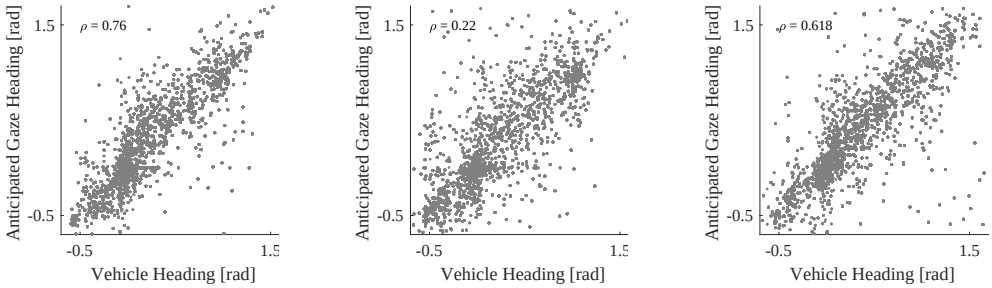
Figure 6.5: Anticipated gaze heading.

6.2 Perceptual Behavior Decomposition

The gaze-based approach to perceptual behavior decomposition identifies gaze interaction patterns (GIP) based on observed eye motion. Gaze behavior is decomposed in terms of known motion types: fixation, smooth-pursuit, and saccade [70]. Prior work



(a) Gaze and trajectory heading vs. time.



(b) Gaze vs. trajectory heading.

Figure 6.7: Anticipatory Gaze Behavior.

studying third-person motion guidance [20] shows that eye motion is associated with two perceptual functional modes: measuring a gap to the goal, and tracking current vehicle position.

6.2.1 Gaze Classification

Subject gaze behavior in the first-person guidance task is first decomposed based on established gaze function types: fixation, smooth-pursuit, and saccade using a Markov-based classification algorithm [111]. The result is a sequence of gaze motion segments, separated by saccades: $\mathcal{P} = \{p_1, \ldots, p_n\}$ as shown in Fig. 6.6.

6.2.2 First-person Gaze Functions

In Fig. 6.6 it is apparent that segments of gaze behavior attend to different features, suggesting that they serve different functions to the agent. Based on this observation the hypothesis for first-person data is that gaze segments serve two perceptual functions: cue-fixation and trajectory anticipation. Cue-fixation segments are are coincident with environment obstacle boundaries and have low global velocity. Trajectory anticipation gaze segments do not attend to obstacle boundaries, and scan along the future spatial path of the agent. Fig. 6.6 illustrates four trajectories with gaze points mapped onto the 2D space. Gaze points are classified as cue-fixation or anticipation based on the distance from an obstacle.

To determine how well gaze anticipates behavior, the principal covariance direction of each gaze cluster is compared with the velocity direction at the nearest corresponding point along each trajectory as shown in Fig. 6.7a. Fig. 6.7 shows the heading time-history of three example trajectories, along with the heading of each gaze cluster. Fig. 6.7b shows the correlation ($R^2 = 0.57$) across all trials for this subject, suggesting that subjects are using a predictive cognitive function to reliably anticipate the future trajectory. A key point is that trajectory anticipation in the visual field must reconcile the transformation between the first-person visual field domain and the spatial domain. This anticipation provides evidence that humans easily perform this transform as a cognitive function.

Chapter 7

Behavior Modeling

7.1 Introduction

In the previous chapter, recorded human motion and gaze data was decomposed into interaction patterns that describe sets of functionally-equivalent behavior. The identified interaction patterns include subgoal selection and planning, generation of unconstrained motion between subgoals, and perceptual-guidance control modes based on constraint classes. In this chapter, a generative behavior model is formulated that is inspired by the interaction patterns observed in human behavior. The generative model specifies interaction patterns at each level of the hierarchical guidance framework in response to a given task-environment scenario. The objective for the model is that it should generate behavior that achieves characteristics of human performance, including robustness to uncertainty, adaptability to a range of tasks, and computational efficiency for real-time deployment.

This chapter begins by formulating the motion guidance task as a constrained optimal control problem. This formulation shows that constraint transition points can be identified, and used to decompose a task into motion guidance elements, analogous to the behavior decomposition in the last chapter. Each of these elements can then be described by control function, relating agent actions to perceived environment information. The resulting elements definitions are then shown to satisfy finite-time stability criteria. Finally, this chapter describes a perceptual attention model based on the information required by the agent to deploy each interaction pattern element. This chapter

concludes with a summary and implications of these models for autonomous guidance and other applications.

7.2 Theoretical Approach

The concept of satisficing suggests that humans use interaction patterns that are easily deployed – in terms of information processing requirements, perceptual requirements, and risk – in place of optimizing a classical motion performance metric. In contrast, an optimal control formulation provides guaranteed performance and stability. The generative model in this chapter is built by first formulating the task at each hierarchical guidance level as a constrained optimal control problem. Properties of this formulation then lead to heuristics that specify a reduced, finite set of feasible interaction patterns that have defined stability criteria.

Throughout this section, the environment and system dynamics are assumed to be deterministic. While this is not realistic for navigation in natural environments, it serves to identify principles that will later be applied to guidance in uncertain environments.

7.2.1 Optimal Control Formulation

The unconstrained optimal control problem is posed as a minimization of a cost functional over the set of admissible control input series'. A Hamiltonian function is used to solve it, formulated based on properties of function minima:

$$H(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{u}, t) = \boldsymbol{\lambda}^T(t) f(\mathbf{x}, \mathbf{u}, t) - j(\mathbf{x}, \mathbf{u}, t) \tag{7.1}$$

In Eqn. 7.1, $\boldsymbol{\lambda}$ is the costate vector, relating changes in system states to changes in cost. The following conditions hold for the optimal solution:

$$\frac{\partial H}{\partial x_i(t)} = -\dot{\lambda}_i(t) \qquad \frac{\partial H}{\partial u_j} = 0 \tag{7.2}$$

The optimal control sequence or function is obtained by solving Eqn. (7.2) for $\mathbf{u}(t)$. The first condition expresses the rate of decrease of system cost along solution trajectories. The second condition states that the optimal control sequence is an extrema of the cost

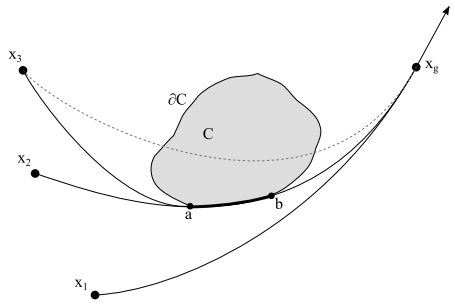


Figure 7.1: Free and mixed control solutions.

function. Analytical solutions are only possible in particular cases; including linear quadratic regulator (LQR) feedback control and finite-time minimum energy feed-forward control [22, 23].

7.2.2 Constrained Optimal Control

Constrained optimal control theory is used to ground the motion guidance problem formulation, and more formally understand properties that determine task structure. The objective of this problem is to find the minimum-cost trajectory, with the condition that solution trajectories must satisfy spatial constraints in the environment. The constrained optimal control problem is expressed by the Lagrange function:

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{u}, t) = \boldsymbol{\lambda}^T(t) f(\mathbf{x}, \mathbf{u}, t) - j(\mathbf{x}, \mathbf{u}, t) + \phi c(\mathbf{x}) \tag{7.3}$$

In Eqn. (7.3) $\phi \geq 0$ and $\phi c(\mathbf{x}) = 0$. ϕ is the Lagrange multiplier, accounting for constraint satisfaction in the optimization problem. Similar to the unconstrained case, solutions must satisfy the following conditions:

$$\frac{\partial \mathcal{L}}{\partial x_i(t)} = -\dot{\lambda}_i(t) \qquad \frac{\partial \mathcal{L}}{\partial u_j} = 0 \tag{7.4}$$

Based on this formulation, solutions $\mathbf{x}(t)$ for $t \in [0, T]$ take one of three forms:

1. Free: $\forall t \in [0, T], c(\mathbf{x}(t)) > 0$
2. Constrained: $\forall t \in [0, T], c(\mathbf{x}(t)) = 0$
3. Mixed: $\exists t \in [0, T], g(\mathbf{x}(t)) > 0 \vee \exists t \in [0, T], g(\mathbf{x}(t)) = 0$

As illustrated in Fig. 7.1, trajectory $\overleftarrow{s}(\mathbf{x}_1, \mathbf{x}_g)$ is free, because for all points $\mathbf{x} \in \overleftarrow{s}(\mathbf{x}_1, \mathbf{x}_g)$, $c(\mathbf{x}) > 0$. Both trajectories $\overleftarrow{s}(\mathbf{x}_2, \mathbf{x}_g)$ and $\overleftarrow{s}(\mathbf{x}_3, \mathbf{x}_g)$ are mixed because they consist of free and constrained segments. For example, trajectory $\overleftarrow{s}(\mathbf{x}_2, \mathbf{x}_g)$ consists of three segments: $\overleftarrow{s}(\mathbf{x}_2, \mathbf{a}) \cup \overleftarrow{s}(\mathbf{a}, \mathbf{b}) \cup \overleftarrow{s}(\mathbf{b}, \mathbf{x}_g)$. Segments $\overleftarrow{s}(\mathbf{x}_2, b)$ and $\overleftarrow{s}(\mathbf{b}, \mathbf{x}_g)$ are free, because all included points satisfy the constraint. The segment $\overleftarrow{s}(\mathbf{a}, \mathbf{b})$ is constrained, because for all points $\mathbf{x} \in \overleftarrow{s}(\mathbf{a}, \mathbf{b})$, $c(\mathbf{x}) = 0$, indicating that the trajectory segment coincides with the constraint boundary. Free and constrained segments within a mixed trajectory meet at transition points, such as points $\mathbf{a}$ and $\mathbf{b}$ in Fig. 7.1.

The key point illustrated by this formulation is that constraint transition points are critical to the structure of a control task. If constraint transition points can be determined in advance, a solution can be composed of segments that are either unconstrained or have a constant set of active constraints. Solutions for each segment type are easier to solve, and may have similarities across the domain. This concept is relied upon for modeling both planning and guidance interaction patterns.

7.3 Planning Model

7.3.1 Overview

The planning behavior decomposition in the previous chapter identified a set of subgoals, $g_k \in \mathcal{G}$ and a set of guidance elements, $\pi_k \in \Pi$, that describe the invariant planning and guidance structure across an environment domain. A planning model is a functional relationship, $H : (\mathcal{O}_E, x_g, \Pi) \rightarrow G$, mapping the set of environment objects $\mathcal{O}_E$, a goal state x_g, and a set of guidance elements Π to a set of subgoals, $G \subset \mathcal{G}$ that the agent may use for planning. When an agent approaches an unknown environment, the function H identifies potential subgoal candidates based on perception of environment geometry and on knowledge of subgoal locations from prior task experience. To generate behavior, an agent uses a policy, $\pi_k = \gamma(\mathbf{x}_p, \mathbf{x}_g, \tilde{G}_k)$ to select a guidance element π_k using subgoal knowledge $\tilde{G}_k$.

This section first extends the definition of subgoals by relating them to constraint transition points through the concept of bounding trajectories. Next, transition point properties are used to state necessary conditions for feasible motion planning subgoal

candidates. Finally, properties of subgoals are used to outline a subgoal planning procedure.

7.3.2 Planning Task Properties

Composite Trajectory

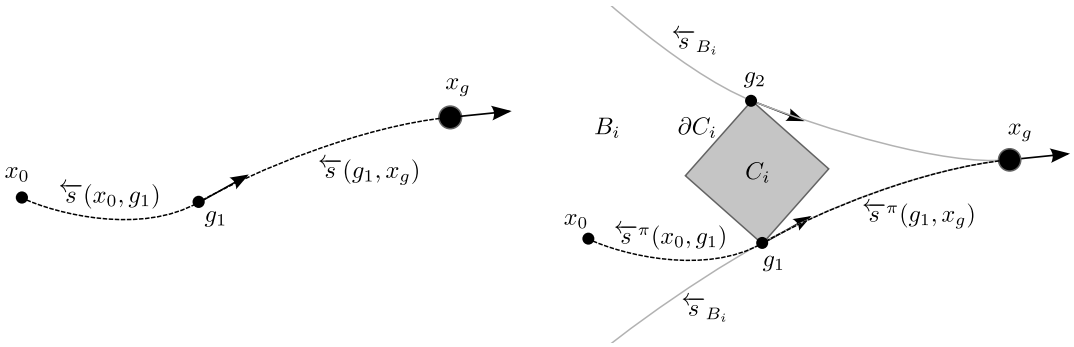
A subgoal is a state along a trajectory, $g_i \in \overleftarrow{s}(\mathbf{x}_0, \mathbf{x}_g)$, dividing the trajectory into two segments $\overleftarrow{s}(\mathbf{x}_0, g_1)$ and $\overleftarrow{s}(g_1, \mathbf{x}_g)$ as shown in Fig. 7.2a. Based on the guidance equivalence relation, all points $\mathbf{x} \in \overleftarrow{s}(\mathbf{x}_0, g_1)$ may be transformed by $\Psi : \mathbb{R}^4 \rightarrow \mathbb{R}^4$ so that $\Psi(g_1) = \mathbf{x}_g$. The resulting transformed segment $\Psi(\overleftarrow{s}(\mathbf{x}_0, g_1))$ is not equal to $\overleftarrow{s}(g_1, \mathbf{x}_g)$, but is described by the same spatial guidance policy, $\pi(\mathbf{x}_0)$. A composite trajectory is described by a series of subgoals and a guidance policy. The guidance policy mapping defines a free trajectory between a pair of subgoals: $\overleftarrow{s}^{\pi}_i(g_{(i-1)}, g_i) = \Pi_{g_i}(g_{i-1})$. A series of subgoals, $\Gamma(g_0, g_n) = \{g_0, g_1, \ldots, g_n\}$, form a composite trajectory as the union of trajectory segments between subgoals, where $g_0 = \mathbf{x}_0$ is the start and $g_n = \mathbf{x}_g$ is the ultimate goal:

$$\overleftarrow{s}(\mathbf{x}_0, \mathbf{x}_g) = \bigcup_{i \in [1,n]} \overleftarrow{s}^{\pi}_i(g_{(i-1)}, g_i) \tag{7.5}$$

Because each subgoal belongs to both the preceding and subsequent trajectory segment, the composite trajectory is smooth. When a series of two or more trajectory segments are joined, the total cost of the composite trajectory is the sum of the individual segment costs, $J(\overleftarrow{s}(\mathbf{x}_0, \mathbf{x}_g)) = \sum_{i=1}^{n} J(\overleftarrow{s}_i(g_{i-1}, g_i)) = J(\Gamma(\mathbf{x}_0, \mathbf{x}_g))$. Such a composite trajectory is piecewise-optimal if it is composed of optimal trajectory segments, $s_i(g_{i-1}, g_i) = s^*(g_{i+1}, g_i)$. The total cost of a piecewise-optimal composite trajectory is greater than or equal to the cost of a single optimal trajectory between the start and end points, i.e. $\sum_{i=1}^{n} J(\overleftarrow{s}^*_i) \geq J(s^*(g_0, g_n))$, with the equality condition holding when all subgoals lie on the optimal trajectory, $g_i \in G \subset s^*(g_0, g_n)$.

Constraint Structure

A constrained region $B_i(\mathbf{x}_g, \pi, O_i)$ contains all points $\mathbf{x}_0$ such that a trajectory from $\mathbf{x}_0$ to $\mathbf{x}_g$ using policy π, $\overleftarrow{s}^{\pi}(\mathbf{x}_0, \mathbf{x}_g)$, does not satisfy O_i, as illustrated in Fig. 7.2b. The boundary of $B_i(x_g, \pi, O_i)$ is defined as $S_{B_i}(\mathbf{x}_g, \pi, O_i)$, and consists of points $\mathbf{x}$ such that



(a) A trajectory divided by a subgoal.

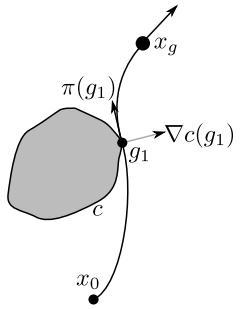
(b) Obstruction avoidance.

Figure 7.2: Example solution trajectories.

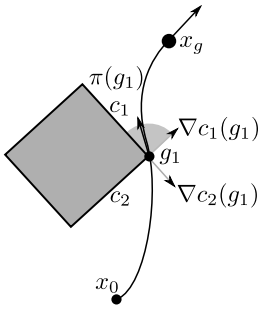
the trajectory $\overleftarrow{s}^{\pi}(\mathbf{x}, \mathbf{x}_g)$ intersects only the constraint boundary ∂O_i, i.e.:

- $\exists \mathbf{x} \in \overleftarrow{s}^{\pi}(\mathbf{x}, \mathbf{x}_g) \mid c_i(\mathbf{x}) = 0$
- $\forall \mathbf{x} \in \overleftarrow{s}^{\pi}(\mathbf{x}, \mathbf{x}_g) \mid c_i(\mathbf{x}) \geq 0$

A trajectory $\overleftarrow{s}^{\pi}(\mathbf{x}_0, \mathbf{x}_g)$ is a bounding trajectory $\overleftarrow{s}_{B_i}(\mathbf{x}_g, \pi, O_i)$ if $\mathbf{x}_0 \in S_{B_i}(\mathbf{x}_g, \pi, O_i)$, i.e. if $\mathbf{x}_0$ is on the boundary of the constrained region. Conversely, all trajectories starting outside of the constrained regions, $\overleftarrow{s}^{\pi}(\mathbf{x}_0, \mathbf{x}_g)$ for $\mathbf{x}_0 \notin B_i$, satisfy O_i.



(a) Continuous constraint boundary. Subgoal at point where policy action $\pi(\mathbf{x})$ is perpendicular to constraint gradient $\nabla c(\mathbf{x})$



(b) Piecewise-continuous constraint boundary. Subgoal at point where action $\pi(\mathbf{x})$ falls within range (grey wedge) bounded by $\nabla c_1(\mathbf{x})$ and $\nabla c_2(\mathbf{x})$

Figure 7.3: Example subgoal placement.

The constrained optimal control problem is now reformulated in terms of subgoals and bounding trajectories. The goal is to determine a sequence of subgoals, $\Gamma(g_0, g_n) =$

$\{g_0, g_1, \ldots, g_n\}$ that specify a piece-wise optimal minimum-cost composite trajectory that satisfies constraints. We first consider the case of a single obstruction, O_1. The constrained optimal solution trajectory in this case contains a single subgoal, $\Gamma(\mathbf{x}_0, \mathbf{x}_g) = \{\mathbf{x}_0, g_1, \mathbf{x}_g\}$, as illustrated in Fig. 7.2b. The subgoal state $g_1 \in \mathcal{X}$ is chosen such that total trajectory cost is minimized, and each segment satisfies constraints:

$$\begin{aligned} g_1 &= \arg\min_{g \in \mathcal{X}} J^*_g(\mathbf{x}_0)) + J^*_{\mathbf{x}_g}(g)) \\ \text{such that } &\overleftarrow{s}(\mathbf{x}_0, g) \notin \mathcal{O}_E \wedge \overleftarrow{s}(g, \mathbf{x}_g) \notin \mathcal{O}_E \end{aligned} \tag{7.6}$$

Eqn. 7.6 is a minimization over a continuous state-space domain, $\mathcal{X}$. To solve this efficiently, properties of the constrained optimal control problem define two necessary conditions that reduced the subgoal candidate domain, $\mathcal{G} \subset \mathcal{X}$.

Condition 1. *Transition points between trajectory segments only occur on constraint boundaries.*

$$\mathcal{G} \subseteq \partial\mathcal{O}_E \times \mathcal{X}_v \tag{7.7}$$

Proof. Consider a piecewise-optimal composite trajectory $\Gamma_{g_1}(\mathbf{x}_0, \mathbf{x}_g) = \{\mathbf{x}_0, g_1, \mathbf{x}_g\}$, with transition point g_1 that is not on a constraint boundary, $c(g_1) > 0$. If the piecewise optimal trajectory is also optimal, $J(s^*(\mathbf{x}_0, \mathbf{x}_g)) = J(\Gamma_{g_1}(\mathbf{x}_0, \mathbf{x}_g))$, then the the composite trajectory is equal to a single optimal trajectory, $s^*(\mathbf{x}_0, g_1) \cup s^*(\mathbf{x}_0, g_1) = s^*(\mathbf{x}_0, \mathbf{x}_g)$. In this case, the transition point g_1 is not required to specify the trajectory, and can be discarded. If the composite trajectory is not optimal, then the total cost is greater than the cost of a single optimal trajectory: $J(\Gamma_{g_1}(\mathbf{x}_0, \mathbf{x}_g)) > J(s^*(\mathbf{x}_0, \mathbf{x}_g))$. In this case, the transition point can be moved to a new location, $g_1 + \delta\mathbf{x}$, so as to reduce the cost of the composite trajectory: $\exists \delta\mathbf{x} \mid J(\Gamma_{g_1+\delta\mathbf{x}}(\mathbf{x}_0, \mathbf{x}_g)) < J(\Gamma_{g_1}(\mathbf{x}_0, \mathbf{x}_g))$. g_1 can be adjusted in this way until one of two cases occur. The first is that the transition point meets the single optimal trajectory, $g_1 + \delta\mathbf{x} \in s^*(\mathbf{x}_0, \mathbf{x}_g)$, in which case any additional adjustment would increase the total trajectory cost. In this case the composite trajectory can be replaced by the optimal single trajectory as described above. The second case is that the transition point reaches a constraint boundary, $g_i + \delta\mathbf{x} \in \partial\mathcal{O}_E$. Any additional movement of g_i would either increase the composite trajectory cost, or result in a composite trajectory that does not satisfy the constraint. As a result,

a transition point g_i either lies on a constraint boundary, or can be discarded. □

Condition 2. *The optimal velocity vector $\pi_{x_g}(g_i)$ at a transition point g_i must not be directed into or out of the constraint boundary.*

$$\begin{aligned} \nabla c_i \cdot \pi(g_i) &\geq 0 \\ \nabla c_i \cdot -\pi(g_i) &\geq 0 \end{aligned} \tag{7.8}$$

Proof. This condition is based on the requirement that the subgoal lies on a trajectory $g_i \in \overleftarrow{s}(\mathbf{x}_0, \mathbf{x}_g)$ that is unconstrained, i.e. $\overleftarrow{s}(\mathbf{x}_0, \mathbf{x}_g) \notin \mathcal{O}_E$. For a subgoal g_i to be admissible, trajectories both terminating at g_i and beginning from g_i must be free. Trajectories passing through g_i are free if the following conditions hold for arbitrarily small $\epsilon > 0$:

$$\begin{aligned} c_i(g_i) &= 0 \\ c_i(g_i \pm \epsilon \cdot \pi(g_i)) &> 0 \end{aligned} \tag{7.9}$$

Substituting Eqn. 7.8 into Eqn. 7.9 results in $c(g_i \pm \epsilon \cdot \pi(g_i)) = c(g_i) \pm \epsilon \cdot (\nabla c_i \cdot \pi(g_i))$, and it follows that since $c(g_i) = 0$, Eqn. 7.9 is satisfied when 7.8 holds. □

When a constraint boundary ∂O_i is continuous at point g_i, Eqn. 7.8 reduces to:

$$\mathcal{G} = \{g_i \mid \nabla c_i(g_i) \cdot \pi(g_i) = 0\} \tag{7.10}$$

Constraint boundaries consist of a set of continuous boundary segments, i.e. $O_i = \{c_{i1}, \ldots, c_{in}\}$, such as in the polygonal obstructions shown in Fig. 4.1a. In this case, the constraint O_i is satisfied if at least one boundary segment constraint is satisfied. If a subgoal lies at a convex corner where two segments, c_{i1} and c_{i2} meet, then a subgoal is admissible over a range of velocity directions:

$$\begin{aligned} \nabla c_{i1} \cdot \pi(g_i) \geq 0 \vee \nabla c_{i1} \cdot \pi(g_i) \geq 0 \\ \nabla c_{i2} \cdot -\pi(g_i) \geq 0 \vee \nabla c_{i2} \cdot -\pi(g_i) \geq 0 \end{aligned} \tag{7.11}$$

Example subgoal locations are illustrated in Fig. 7.3. Combining the two necessary

conditions results in the following definition of admissible subgoals:

$$\mathcal{G} = \{g_i \mid g_i \in \partial\mathcal{O}_E \wedge \nabla c_i(g_i) \cdot \pi_{\mathbf{x}_g}(g_i) = 0\} \tag{7.12}$$

The set $\mathcal{G}$ is a subset of the state space domain that contains extrema of the optimization problem, Eqn. 7.6. For a spatial domain $\mathcal{W} \in \mathbf{R}^2$, Eqn. 7.12 gives a discrete, finite set of subgoal candidates, significantly reducing the complexity of the planning problem.

7.3.3 Subgoal Properties

Eqn. 7.12 is equivalently stated as the following, which emphasizes the dependence of $\mathcal{G}$ on the constraint O_i, and the goal state $\mathbf{x}_g$:

$$G(\mathbf{x}_g, O_i) = \{\mathbf{x} \mid \mathbf{x} \in S_{B_i}(\mathbf{x}_g, O_i) \cap \partial O_i\} \tag{7.13}$$

Fig. 7.2b illustrates the constrained optimal control problem with a single obstruction. In this example problem, the direct trajectory from $\mathbf{x}_0$ to $\mathbf{x}_g$ using the optimal policy, $\overleftarrow{s}(\mathbf{x}_0, \mathbf{x}_g) = \Pi_{\mathbf{x}_g}(\mathbf{x}_0)$ is constrained, i.e. $\exists \mathbf{x} \in \overleftarrow{s}(\mathbf{x}_0, \mathbf{x}_g) \mid \mathbf{x} \in O_i$. Since the direct trajectory is constrained, a composite solution must be found consisting of two free segments joined at a subgoal. For this problem, the set of admissible subgoals are $\mathcal{G} = \{g_1, g_2\}$ based on conditions in Eqn. 7.13. The optimal subgoal is chosen from $\mathcal{G}$ that minimizes the total trajectory cost based on Eqn. 7.6. In Fig. 7.2b, g_1 is depicted as the optimal subgoal, and the optimal solution trajectory is illustrated as $\overleftarrow{s}^*_1(\mathbf{x}_0, g_1) \cup \overleftarrow{s}^*_2(g_1, \mathbf{x}_g)$.

These subgoal properties result in a partitioning of the task domain, as observed by Kong and Mettler [14] for optimal Dubin's path solutions. An obstacle O_i, goal state $\mathbf{x}_g$, and guidance policy π define a constrained region $B_i(\mathbf{x}_g, \pi, O_i)$ and a set of subgoals $g_j \in \mathcal{G}$. The constrained region is further divided into partitions, such that all optimal trajectories beginning within a partition $P(g_j) \subset B_i$ converge to subgoal g_j on the optimal path to the goal $\mathbf{x}_g$. A partition $P_j(g_j)$ is defined by bounding trajectories $\overleftarrow{s}_{B_i}$, and a separatrix, T_G, which is a set of points for which the total trajectory cost is equal for two or more different subgoals $g_i, g_j \in G$:

$$T_{B_i} = \{\mathbf{x} \mid J_{g_i}(\mathbf{x}) + J_{\mathbf{x}_g}(g_i) = J_{g_j}(\mathbf{x}) + J_{\mathbf{x}_g}(g_j)\} \tag{7.14}$$

Separatrices T_B are repelling manifolds, dividing initial states that move toward different subgoals. Bounding trajectories S_B are attracting manifolds, such that initial states on either side of the manifold result in nearly the same optimal trajectory. Subgoal and partition properties define equivalence classes associated with the equivalence relations introduced by Kong and Mettler [14]. Partitions specify a single optimal subgoal for each initial configuration. The set of initial configurations in a partition $\mathbf{x}_0 \in P_i$ belong to the same subgoal equivalence class through subgoal g_i. By this equivalence, determining the partition that an initial state belongs to fully specifies the remaining trajectory to the goal.

In environments with multiple obstructions, the solution trajectory may require multiple subgoals. The optimization problem becomes that of selecting an optimal sequence of subgoals, $\Gamma_\pi^* = \{g_1^*, g_2^*, \ldots, g_n^*\}$ that define a minimum cost path to the goal using trajectory segments generated by policy π.

$$\Gamma_\pi^* = \underset{\Gamma_i \subset \mathcal{G}}{\arg\min}\, J_\pi(\Gamma_i)) \tag{7.15}$$

The principle of optimality [112] states that any sub-trajectory $\overleftarrow{s}(g_i, g_{i+k})$ for $g_i, g_{i+k} \in \Gamma$ must be an optimal trajectory between endpoint subgoals g_i and g_{i+k}. Hence, the optimal subgoal sequence can be defined recursively:

$$\Gamma_\pi^*(g_0, g_n) = \{g_0, \ldots, g_{n-2}, \underset{g_{n-1} \in \mathcal{G}}{\arg\min}\, J_\pi^*\left(\Gamma_\pi^*(g_0, g_{n-1}), g_n\right), g_n\} \tag{7.16}$$

Eqn. 7.16 reduces the problem size by one, by solving a single-subgoal problem as presented in Eqn. 7.6. In practice, a solution Γ_π^* is found using dynamic programming.

As in road-map motion planning approaches, subgoals define partitions that discretize the task. In order to generate solutions using a graph-search algorithm, nodes must satisfy the Markov property, requiring that each next optimal state depends only on the present state [27]. The Markov property requires that the next optimal subgoal cannot depend on the previous subgoal on the trajectory, meaning that a set of solution trajectories must form a tree-structure. The Markov property is equivalent to two properties of subgoals. First, that partitions are hierarchically included, each partition is a subset of a partition that is closer to the final goal. Second, the Markov condition satisfies

the subgoal equivalence observed in human guidance behavior, stating that trajectories that meet at a subgoal remain together until they reach the final goal.

7.4 Guidance Model

7.4.1 Overview

A guidance behavior model is a functional relationship, $\Pi : (\mathcal{I}, x_g, f()) \rightarrow \pi_k$, mapping perceptual information $\mathcal{I}$, a goal state x_g and system dynamics $f()$ to a set of guidance elements $\pi_k \in \Pi$. When an agent approaches a new task, they must learn the set of feasible guidance elements for the task. This section focuses on modeling the individual guidance elements π_k that specify agent action sequences over the domain of perceptual information, $\pi_k : \mathcal{I} \rightarrow \mathcal{A} \times T$. Based on the hierarchical partitioning of system dynamics described in Chapter 4, guidance dynamics model the continuous motion of the agent in response to actions as:

$$\dot{\mathbf{x}}_p = f_g(\mathbf{x}, \mathbf{a}) = \mathbf{v}_{ref} \tag{7.17}$$

In Eqn. 7.17, the agent guidance action is $\mathbf{v}_{ref} \in \mathcal{A}$ and is in the range of velocities reachable from the domain of allowable system states and control inputs: $\mathcal{A} := \{\mathbf{v}_{ref} = f(\mathbf{x}, \mathbf{u}) | \mathbf{u} \in \mathcal{U} \wedge \mathbf{x} \in \mathcal{X}\}$. $\mathcal{I}$ is the domain of relevant goal information (RGI) [15]. RGI consists of values of states that are constrained by the goal set. For example, a spatial subgoal g_k may specify agent position and heading, (θ_G, d_G, ψ), which constitute RGI. The goal set may not specify agent speed, $|v|$, so it is not RGI.

7.4.2 Spatial Policy Model

In lieu of computing a full control sequence in advance for each initial state, optimal control sequences $\mathbf{u}^*(t)$ throughout a system domain can be expressed as a control policy function, giving a control input as a function of system state and desired goal state: $\mathbf{u}^* = k_{x_g}(\mathbf{x})$. The quality of a control policy is defined by an objective function, $J_{x_g}(\mathbf{x})$, expressing the cost of a trajectory from state $\mathbf{x}$ to the goal $\mathbf{x}_g$ resulting from the control sequence. Approaches to this problem include analytical finite-time control solutions, such as minimum-energy control [23], that apply to linear system dynamics. When system dynamics are nonlinear, a dynamic programming approach is required,

incurring high computational cost [112]. In addition, the resulting policy may have high-dimension domain, for example a rotorcraft position controller must determine four control actuator values based on 14 or more states.

Optimal Policy

The spatial policy can be used to reduce control policy dimension, and can be defined in terms of an optimal control formulation. Given an initial configuration $\mathbf{x}_p \in \mathcal{W}$ and a goal state $\mathbf{x}_g \in \mathcal{X}$, the initial velocity $\mathbf{v}_{ref}$ at $\mathbf{x}_p$ can be found that minimizes the objective function $J(\overleftarrow{s}^*(\mathbf{x}_0, \mathbf{x}_g))$, with $\mathbf{x}_0 = \{\mathbf{x}_p, \mathbf{v}_{ref}\}$. Optimal spatial VVF and CTG functions are:

$$\begin{aligned} \pi^*_{\mathbf{x}_g}(\mathbf{x}_p) &= \underset{\mathbf{v}_{ref} \in X_v}{\arg\min}\, J_{\mathbf{x}_g}(\mathbf{x}_p, \mathbf{v}_{ref}) \\ J^*_{\mathbf{x}_g}(\mathbf{x}_p) &= \underset{\mathbf{v}_{ref} \in X_v}{\min}\, J_{\mathbf{x}_g}(\mathbf{x}_p, \mathbf{v}_{ref}) \end{aligned} \tag{7.18}$$

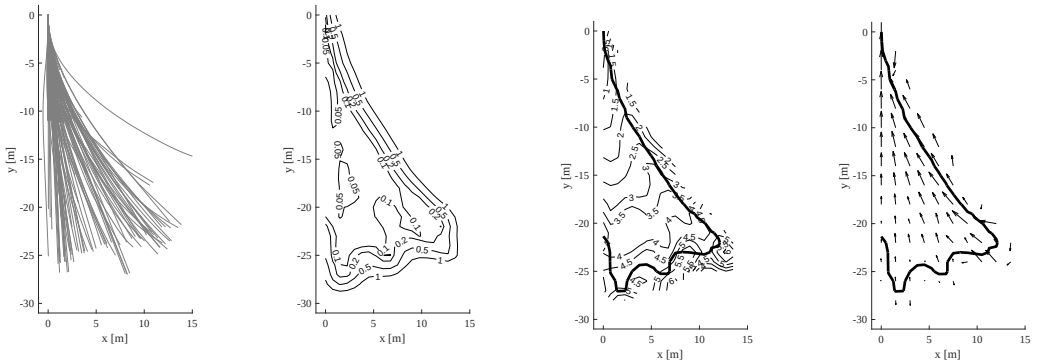
The VVF and CTG specify optimal trajectories within the configuration space of system dynamics, reducing the dimensionality of the optimal control policy.

Nominal Policy

Humans however may not always generate guidance behavior that is optimal with respect to typical control costs such as settling time or energy [10]. Based on the idea of bounded rationality, a sub-optimal solution may be faster or easier to compute when an optimal policy is unknown, too complex, or too uncertain. A nominal guidance policy, $\pi(\mathbf{x}, \mathbf{x}_g) : \mathcal{W} \times \mathcal{X} \rightarrow \mathcal{X}$, generates a sub-optimal reference trajectory given an initial configuration state and a goal state.

Machine Learning Model

One approach to identifying a human nominal guidance policy is to model the task as a Markov decision process (MDP) for which utility and policy functions can be learned based on example data ([113]). For motion guidance, utility and policy functions consist of CTG and VVF functions. Direct utility estimation is used to identify approximate functions $\hat{V}^*(x_p)$ and $\hat{\pi}^*(x_p)$ that best model actual behavior of the human subject.



(a) Guidance segments. (b) CTG std. dev. [sec]. (c) CTG map [sec]. (d) VVF map.

Figure 7.4: Learned guidance policy.

Reinforcement learning is not used here, but may be applied in future work to determine a policy that is optimal for a specified cost function based on human example behavior. In the present work, time-to-go is considered as the cost function for the following reasons: 1) subjects were asked to minimize travel time during the task, 2) [17] suggests that time is a quantity that humans intuitively use to plan and make decisions about motion, and 3) previous work has successfully modeled human guidance cost using time-to-go ([12]). The training data set consists of points along each trajectory, $s_i(k) = \{\mathbf{x}_p(k), \mathbf{x}_v(k), t(k)\}$, for time-step k and trajectory i as shown in Fig. 7.4a. The example input domain consists of $\mathbf{x_{in}} = \{\mathbf{x}_p(k)\}$ for both spatial CTG and guidance policy functions. The example output is $\mathbf{y_{ctg}} = \{t(k)\}$ for spatial CTG, and $\mathbf{y_{vv}} = \{\mathbf{x}_v(k)\}$ for guidance policy.

Gaussian process regression approximates a function based on a set of squared exponential kernel functions:

$$k(x_i, x_j) = \sigma_f^2 \exp\left[\frac{-(x_i - x_j)^2}{2l^2}\right] + \sigma_n^2 \delta(x_i, x_j) \tag{7.19}$$

A matrix K is defined such that $K_{ij} = k(x_i, x_j)$, vector K_* such that element $K_{*i} = k(x_*, x_i)$, and $K_{**} = k(x_*, x_*)$, where $i, j \in [1, m]$ for m input example points, and x_* is the test input. The best estimate of the approximated function at x_* is given by $\bar{y}_* = K_* K^{-1} \mathbf{y}$, and the covariance at that point is given by $var(y_*) = K_{**} - K_* K^{-1} K_*^T$. $\mathbf{y}$ is the example output set. K is invertible if all example inputs x are distinct. The

resulting approximated CTG function is shown in Fig. 7.4c, the cost covariance in Fig. 7.4b, and the VVF function is shown in Fig. 7.4d [114].

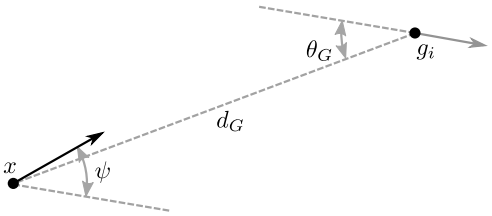
7.4.3 Control Mode Model

Overview

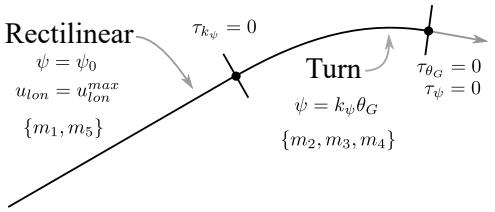
A spatial value function (SVF) guidance policy model representation has limitations however: first, an agent using an SVF is limited to generating actions within the learned function domain, because the SVF does not define actions outside of that region. The second limitation is that large spatial domains associated with tasks such as driving or piloting require impractical amounts of memory and a huge amount of prior experience to adequately cover. In contrast, humans generate motion behavior in large domains and with a small number of example trials, suggesting that they extrapolate behavior using functional principles learned from prior task experience.

Behavior extrapolation is the application of a strategy to a new task based on the notion that the task is functionally similar to some other, previously encountered task, for which a strategy is already known. The concept of interaction patterns for guidance defines an approach to extrapolation, in which a new task is first classified into a known interaction pattern class based on an equivalence relation that defines the functional similarity between tasks. Then, guidance strategies are applied as a function of specific task properties within that interaction pattern class.

This section proposes a hybrid control model that better approximates the robust, adaptive learning and extrapolation capabilities observed in humans based on the concepts of interaction patterns and extrapolation. A hybrid control model is similar to a motion primitive maneuver automaton (MPMA) described in prior work [47, 44, 46, 115], but each mode defines an agent-environment interaction behavior element, rather than only a motion profile. agent-environment interaction elements, or guidance primitives, define relationships between perceptual quantities and system control inputs that hold while each mode is active. A control mode m_i combines attentional elements, $\mathbf{e}_{attn} = h(\mathbf{x})$, perceptual functions, $\mathbf{i} = \eta_i(\mathbf{e}_{attn})$, and agent interactions, $\mathbf{u} = k_i(\mathbf{i})$. By combining these system elements, a control mode closes the loop around agent-environment dynamics, defining an autonomous guidance behavior pattern.



(a) Relevant Goal Information.



(b) Control mode sequence.

Optimal Control Formulation

A hybrid control guidance model contains two parts: a library of perceptual guidance control modes $m_k \in \mathcal{M}$ and a policy to determine an optimal control mode sequence, $\pi_{\mathcal{M}}$. The policy specifies a sequence of control modes in response to a guidance task: $\{m_1, m_2, \ldots, m_n\} = \pi_{\mathcal{M}}(x_p, \mathcal{O}_E, \mathcal{M})$. Determining a sequence of modes requires the agent to simultaneously optimize transition points between modes and optimize each mode definition. The following three hypotheses are used to make this problem tractable: first, control mode transition points occur where the set of active dynamic constraints changes. This hypothesis is based on properties of the constrained optimal control formulation above that was used to define spatial subgoals. This property simplifies the guidance task by reducing the size and dimensionality of the set of candidate mode transition points. The second hypothesis is that guidance elements consist of typical mode sequences that occur repeatedly. This hypothesis is based on the guidance equivalence relation, [7], stating that guidance behavior segments are equivalent within each subgoal partition across the task domain. The third hypothesis is that control modes include signal relationships that specify agent actions in response to perceived quantities. This hypothesis is based on prior work on the concept of perceptual guidance, and on observed behavior decomposition results showing that control modes occur within agent-environment system subspace manifolds.

Tau Model of Perceptual Guidance

The guidance behavior model is constructed as a sequential closure of gaps in perceived relevant goal information, typical across guidance elements. Perceived relevant goal information, $\mathbf{i}$ is defined as $\mathbf{i} = \{\theta_G, d_G, \Psi\}$, as depicted in Fig. 7.5a. Each gap closure is described by Tau theory [17]. The hybrid control model consists of two modes, turning

and rectilinear, as depicted in Fig. 7.5b.

Turning Model During turning, vehicle heading converges to the subgoal heading ($\psi \rightarrow \psi_G$) and the goal bearing error converges to zero ($\theta_G \rightarrow 0$). These two gap closures can be coordinated by coupling the Tau of each gap through a constant, i.e. $\tau_\psi = k\tau_{\theta_G}$, where $\tau_\psi = \psi/\dot{\psi}$ and $\tau_{\theta_G} = \theta_G/\dot{\theta_G}$. Substituting, this coupling results in the relationship:

$$\frac{\psi}{\theta_G} = k\frac{\dot{\psi}}{\dot{\theta_G}} \tag{7.20}$$

This relationships is consistent with the steering modality identified in the previous chapter modeled by the linear perceptual guidance relationship. Based on this, agent turning behavior is modeled as:

$$\psi = k^*_{steer}\theta_G \tag{7.21}$$

Rectilinear Model Prior to turning, the agent may travel with rectilinear motion. During this mode vehicle heading is constant ($\dot{\psi} = 0$), and goal bearing error is decreasing ($\dot{\theta_G} < 0$). As a result, the steering ratio $k_{steer} = \psi/\theta_G$ decreases during this mode. The agent then transitions from rectilinear to steering when the steeting ratio reaches the desired turning-mode value, $k_{steer} \rightarrow k^*_{steer}$. The error between the actual desired turning-mode steer ratio can be considered as a Tau gap closure:

$$\begin{aligned} \delta k = &\quad (k_{steer} - k^*_{steer}) \\ \tau_k = &\quad \delta k/\dot{\delta k} \end{aligned} \tag{7.22}$$

The agent transitions from rectilinear to turning behavior then $\tau_k = 0$.

Speed Control Model The agent must also control vehicle speed, which is not explicitly defined by the turning and rectilinear modes. During the turning mode, vehicle speed determines the turn-rate limit. As a result the agent must regulate speed to follow the turn mode behavior model in Eqn. 7.21. Based on vehicle dynamics, the reference speed during turning is: $v_{ref} = u^{max}_{lat}/\dot{\psi}$.

During rectilinear motion, traveling at the maximum speed provides the shortest travel time, however the agent may need to regulate speed during rectilinear motion in anticipation of initiating a turn. As an agent approaches the transition from rectilinear to turning, they can compute a speed gap as: $\delta v = v - u^{max}_{lat}/\dot{\psi}_0$, where $\dot{psi}_0$ is the

anticipated initial turn-rate of the upcoming the turning mode. Anticipatory braking can be coordinated by coupling $\tau_v = \delta v / \delta \dot{v}$ with τ_k, $\tau_v = k\tau_k$.

7.5 Stability

7.5.1 Stability Criteria

With satisficing, human solutions need not always be optimal, but system stability is important to guarantee that a system converges to the target state. Stability is defined with respect to a non-empty neighborhood, $\mathcal{V} \in \mathbb{R}^n$, containing the target state, $\mathbf{x}_g$. A system is asymptotically stable in the sense of Lyapunov if there exists a smooth function $V : V \rightarrow R_{\geq 0}$ such that $V(\mathbf{x}_0) > 0$ for all $\mathbf{x}_0 \in \mathcal{V} - \{\mathbf{x}_g\}$, $V(\mathbf{x}_g) \geq 0$, and $\dot{V}(\mathbf{x}) < 0$ for all $\mathbf{x} \in \mathbf{V}$ [50]. Asymptotic stability guarantees that a system reaches the target state as time goes to infinity. For the motion guidance task, it is additionally required that the system reaches the target in a bounded time interval, $t \in [0, T]$. Moulay and Perruquetti present Lyupanov-based criteria for finite-time stability [51]:

$$T(\mathbf{x}_0) = \int_{V(\mathbf{x}_0)}^{0} \frac{d\xi}{\dot{V}(\overleftarrow{s}_{\mathbf{x}_g}(\mathbf{x}_0, \theta(\xi))} < +\infty \tag{7.23}$$

In Eqn. 7.23, $T(\mathbf{x}_0)$ is the settling-time, or time-to-go of initial state $\mathbf{x}_0$. $\theta : V(\mathbf{x}) \rightarrow t$ is inverse cost, mapping Lyuponov function value to settling time. Eqn. 7.23 is difficult to evaluate if dynamics are not defined analytically or contain uncertainty. In either case, finite-time stability may still be shown using a bounding function, $g(\xi) \in L^1([0, \sup_{\mathbf{x} \in \mathcal{V}} V(\mathbf{x})])$. The system is finite-time stable if for all $\mathbf{x} \in \mathcal{V} - \{\mathbf{x}_g\}$, and all $\xi \in [0, V(\mathbf{x})]$:

$$\frac{-1}{\dot{V}((\overleftarrow{s}_{\mathbf{x}_g}(\mathbf{x}_0, \theta(\xi)))} \leq g(\xi) \tag{7.24}$$

Based on Eqn. 7.24, it is shown in [51] that settling time is bounded when $\dot{V}(\mathbf{x}) \leq -c(V(\mathbf{x}))^{\alpha}$, for $c > 0$, $\alpha \in [0, 1]$. When this condition holds, settling time of an initial state $\mathbf{x}_0 \in \mathcal{V}$ is bounded by:

$$T(\mathbf{x}_0) \leq \frac{V(\mathbf{x}_0)^{1-\alpha}}{c(1-\alpha)} \tag{7.25}$$

This criteria is applied later to evaluate stability requirements for specific planning and guidance approaches.

7.5.2 Planning Stability

The stability of a constrained subgoal planning problem, $\langle \mathbf{x}_0, \mathbf{x}_g, \mathcal{O}_E, \Pi \rangle$ is considered with respect to the evolution of a solution sequence of subgoals, $\Gamma = \{g_0, g_1, \dots\}$ with $g_0 = \mathbf{x}_0$. A Lyupanov function, $V(g_k) = J^*_\pi(\Gamma(g_k, g_n)) = \sum_{i=k}^{n-1} J^*_\pi(g_i, g_{i+1})$ is defined as the total cost incurred by the sequence of guidance elements connecting each pair of subgoals, $\langle g_i, g_{i+1} \rangle$ in sequence Γ. For the discrete-time process, $\dot{V}$ is expressed as $\dot{V}(g_k) = \Delta V / \Delta T = (V(g_k) - V(g_{k-1})/(T(g_k) - T(g_{k-1})$, which is the rate at which the cost decreases during the subgoal transition g_{k-1} to g_k. Based on this, Eqn. 7.23 is extended to the discrete time system:

$$T(g_0) = \sum_{g_k \in \Gamma} \frac{\Delta V(g_k)}{\Delta V(g_k)/\Delta T(g_k)} = \sum_{g_k \in \Gamma} \Delta T(g_k) < +\infty \tag{7.26}$$

Based on Eqn. 7.26, finite-time planning stability requires that both $\Delta T(g_k) < +\infty$ and $V(g_k) < +\infty$ for all $g_k \in \Gamma$. Practically, these two conditions are met if the guidance policy is finite-time stable for each pair of subgoals in the plan, and if the plan reaches the goal using a finite number of subgoals. Guidance policy stability is addressed later in the paper.

7.5.3 Guidance Mode Stability

Finite-time stability criteria determine whether a system converges to a goal state in finite time. The initiation set for each mode is defined by the guidance primitive automaton mode transition criteria described in the next section.

Planning stability requires that trajectories between each pair of subsequent subgoals can be completed in finite time. This requires finite time stability of the guidance policy. To show this, the turning perceptual guidance strategy shown in Table 8.1 is expressed

in terms of distance to the goal d and bearing error θ_g:

$$\begin{aligned} \dot{d} &= -v\cos(k\theta_g) \\ \dot{\theta}_g &= -\frac{v}{d}\sin(k\theta_g) \end{aligned} \tag{7.27}$$

The following Lyupanov function is chosen for $\mathbf{x}_p = [d, \theta_g]$:

$$\begin{aligned} V &= d + \theta_g \\ \dot{V} &= \frac{dV}{d\mathbf{x}_p}\dot{\mathbf{x}}_p = -v(\cos(k\theta_g) + \frac{1}{d}\sin(k\theta_g)) \end{aligned} \tag{7.28}$$

Based on Eqn. 7.24, the system is finite-time stable when $\dot{V} \leq -c(V(\mathbf{x}_p))^\alpha$, resulting in:

$$-v(\cos(k\theta_g) + \frac{1}{d}\sin(k\theta_g)) \quad \leq \quad -c\,(d + \theta_g)^\alpha$$

Taking $\alpha = 0$, the system is finite-time stable when $c = v_{min} \min\left[\cos(k\theta_g) + (1/d)\sin(k\theta_g)\right] > 0$. This is conservatively satisfied when $v_{min} > 0$ and $0 < \theta_g < \pi/2k$, which corresponds to $|\theta_g| \leq 115$ degrees. Furthermore, settling time is bounded by $T(\mathbf{x}_0) \leq V(\mathbf{x}_0)/c = (d + \theta_g)/c$, which is an admissible planning heuristic. Such a conservative bound allows stability to be robustly ensured over a range of guidance policies that may result from modeling errors, system failures, or environmental uncertainty.

7.6 Perceptual Attention Model

This section describes a perceptual attention model based on observed human guidance behavior elements in the previous chapter and on the guidance model described above. Perceptual attention is a map or filter, indicating the subset of perceptual information that is relevant to agent motion control. As described in Chapter 5, the importance of a perceptual measurement Z with respect to an action A depends on mutual information, $I(Z; A)$. In this section, relevant perceptual information is determined by the perception-action relationships specified by the guidance model.

In prior work, Mettler et al. [6] describe a systems-based approach to human guidance behavior modeling, including a perceptual attention model. The attention model describes perceptual requirements at each hierarchical level: planning, guidance and tracking, and divides them into three categories: attention elements, perceptual functions, and control functions. In the perceptual attention model, control functions represent the decisions agents make about taking actions **a** in response to perceptual information, driving perceptual requirements. Perceptual functions define the information **i** needed for the agent to implement a control function. A perceptual function maps visual information into quantities directly used by control functions. Finally, attention elements, $\mathbf{e}_{attn}$ are environment and task locations to which the agent must attend to extract the required perceptual information. An agent's sensory function maps environment locations to regions of the visual field that contain relevant perceptual information.

7.6.1 Planning Perceptual Functions

The planning-level control function is implemented as the subgoal planning model described above that selects a sequence of subgoal states. Choosing a subgoal requires information about the utility of a path through that state. Path utility includes both feasibility, in terms of spatial constraint satisfaction, and path cost. To maximize utility, the agent must select a subgoal that is optimal over the predicted costs and feasibilities of potential paths.

Predicting path feasibility requires comparing a projected path to perceived obstacle locations in the spatial domain. A predicted path $\tilde{s}_i = \rho(g_i, x_g, \tilde{\mathcal{O}}_E)$ is a function of the subgoal g_i, the ultimate goal x_g, and the visible environment objects, $\tilde{\mathcal{O}}_E$, where the tilde indicates that it is an internal representation by the agent. At a time k during a guidance task, the agent may perceive multiple paths through feasible subgoals, $\tilde{s}_i \in \tilde{S}_k$. The corresponding agent attention element consists of the set of projected paths and the visible environment objects: $\left(\tilde{S}_k, \tilde{\mathcal{O}}_{E,k}\right)$. For first-person visual perception, the agent must transform this set into the visual field to direct their visual attention. The perceptual function associated with these attentional elements is $\tilde{G}_k = \eta(\tilde{S}_k, \tilde{\mathcal{O}}_{E,k})$, where $(g_i, J_i) \in \tilde{G}_k$ is the agents perception of cost and feasibility of paths through each subgoal candidate.

For humans, this results in the previous chapter show gaze motion consistent with

a predictive attentional function. For a computational system, the agent can estimate path feasibility by simulating trajectories that pass through each subgoal candidate. The resulting trajectory path can be compared with obstacle boundaries to determine feasibility.

7.6.2 Guidance and Motion Automaton Perceptual Functions

In the guidance level control function, the agent determines a reference velocity as a function of their position relative to a subgoal: $\mathbf{v}_{ref} = \pi_k(\mathbf{x}_p, g_k)$ as described in Chapter 4. In the guidance model presented above, this process is abstracted through a hybrid maneuver automaton, in which the agent closes a sequence of perceptual Tau gaps. The Tau model describes each motion guidance element by a common gap closure sequence.

Based on this Tau model the agent has three control functions, each defining specific perceptual functions and attentional elements:

1. During rectilinear motion, the agent closes a distance gap until turn initiation. During this phase, the agent must perceive τ_k, indicating the time to turn initiation, which is based on ψ and θ_G. This perceptual function indicates that the subgoal bearing and heading, (ψ, θ_G) are an important attentional element.

2. During turning, the agent maintains constant $k_{steer} = \psi/\theta_G$. In this phase, the control function is different, but uses the same perceptual information, suggesting that the agent focus on the same attentional elements in the environment.

3. During both rectilinear and turning motion the agent must modulate speed.

    a. During turning, the agent maintains a turn-rate that satisfies the steering ratio, and must modulate speed in response to $\dot{\psi}$ to satisfy the lateral acceleration constraint. This speed modulation indicates that the agent must employ a perceptual function measuring speed and angular velocity. Both speed and angular velocity can be computed from the optical flow in the visual field of textured objects in the environment.

    b. During rectilinear motion, in order to brake in anticipation of turning, the agent must predict the turn rate and corresponding maximum speed at the point of turn entry. This predictive requirement suggests that points along

Level	**Control Function**	**Perception Function**	**Attention Element**
	$\mathbf{a} = k(\mathbf{i})$	$\mathbf{i} = \eta(\mathbf{e}_{attn})$	$\mathbf{e}_{attn}$
Subgoal Planning	$\pi_k = \gamma(g_k, x_g, \tilde{G}_k)$	$\tilde{G}_k = \eta(\tilde{S}_k, \tilde{\mathcal{O}}_E)$	$\tilde{S}_k, \tilde{\mathcal{O}}_E$
Motion Guidance	$\mathbf{v}_{ref} = \pi_k(d_G, \theta_G, \psi)$	$(d_G, \theta_G, \psi) = \eta(g_k)$	g_k
Turning	$\psi_{ref} = k_\theta^* * \theta_G$	$\theta_G = \eta(g_k)$	g_k
Rectilinear	$\psi_{ref} = \psi_0 \ : \ k_\theta < k_\theta^*$	$k_\theta = \theta_G / \psi$	g_k

Figure 7.6: Overview of perceptual attention model.

the predicted trajectory near the point of turn entry are an important attentional element.

7.7 Discussion

In this chapter, a behavior model is defined that describes action-perception interactions at each level of the guidance hierarchical model. The concept of subgoals is introduced along which necessary conditions that provide a method for breaking a general spatial guidance task into a series of unconstrained guidance tasks, for which solutions can be found using a spatial guidance policy. Next, necessary conditions are extended from spatial dynamic constraints to specify transition points between dynamic control modes. The resulting behavior segments are clustered into functional control modes that occur along discrete sub-manifolds within the full system state space. Finally, extensions of planning and guidance models predict the perceptual functions and attentional elements important for each task. This perceptual analysis suggests that the specific control functions used by the agent serve as a filtering approach to obtain relevant guidance information from the full set of perceptual information available in the environment. These perceptual functions are in duality with agent actions, in which available perceptual information acts to filter the control strategies that are feasible for an agent to deploy.

Chapter 8

Behavior Deployment

8.1 Overview

The previous chapter described models for each functional guidance level, each specifying applicable interaction patterns in response to a task-environment scenario. This section defines the algorithms that generate autonomous behavior by deploying sequences of interaction patterns at each functional level. Breaking down motion behavior into sequences of interaction pattern elements converts the continuous trajectory optimization into a discrete planning problem. A planning algorithm computes the sequence of interaction patterns generating a trajectory that reaches the goal while maximizing a utility metric.

This chapter first describes the algorithms used to generate a sequence of guidance behavior at each functional level. Next, an integrated system architecture is introduced showing how these approaches are combined across functional levels to generate autonomous, dynamic trajectories in known environments. Finally, a modified system is suggested for guidance in an unknown environment in which the agent is exploring and learning the environment as they are navigating.

8.2 Planning

The previous chapter presented two elements that allow the constrained optimal control problem to be formulated as a graph search problem. The first element is the set of

necessary conditions that specify a discrete, finite set of subgoal candidates used to construct an optimal trajectory. The second element is Eqn. 7.16, which defines a recursive approach for determining an optimal subgoal sequence. This chapter presents a solution to determining an optimal subgoal sequence using a graph search algorithm. In this approach, graph nodes consist of admissible subgoal candidates, $\mathcal{G}$, and the guidance policy Π is used to determine feasible edges between nodes. An optimal graph search algorithm computes a sequence of subgoals, $\Gamma = \{\mathbf{x}_0, g_1, \ldots, g_n\}$ that specify a piecewise-optimal solution trajectory.

In a typical forward-expansion graph search application, edge costs are estimated using heuristics [116], for example when nodes represent spatial locations the heuristic may be distance. The heuristic is then used to find a set of successor nodes with minimum cost edges connecting them to the current node. In the guidance application, edge cost also depends on the velocity vector of the destination node, which is not known in advance for any node except for the final goal. To account for this backwards dependence, the backwards A* [117] graph search algorithm is used to determine subgoal sequence solutions beginning at the final goal state and expanding connected nodes towards the start.

8.2.1 Graph Search

Backwards A* search begins at the goal and works backward, keeping a list of open nodes, $\mathcal{G}_{open}$, that are currently under consideration as the previous node in the optimal sequence. Initially, only the goal state $\mathbf{x}_g$ is in the open list, and is designated as the current node. At each iteration, if the current node is not the start location, then the current node is expanded backwards. Backwards expansion consists of finding neighboring nodes for which a trajectory to the current node is feasible. This is performed by the **getNeighbors()** function described in the next section. Each neighboring node is either added to the open list, or the cost of that node is updated if a lower cost edge has been found. Next, the algorithm chooses the node in the open list with the lowest estimated total cost and designates it as the new current node. The estimated total cost of each open node is $f(g_i) = h(\mathbf{x}_0, g_i) + g(g_i, \mathbf{x}_g)$, where $h(\mathbf{x}_0, g_i)$ is the estimated cost of traveling from the start to the subgoal, based on a heuristic, such as distance between nodes. $g(g_i, \mathbf{x}_g)$ is the actual cost of traveling from subgoal g_i to the goal $\mathbf{x}_g$,

using the guidance policy Π. If the current node is the start location, then search is complete. The A* algorithm is shown in Fig. 2.

The subfunction **getNeighbors(g)**, called by backwards A*, returns a list of subgoals that are able to reach subgoal g using a free nominal trajectory, along with the total cost associated with each subgoal. **getNeighbors(g)** is shown in Fig. 1. The function **predictTrajectory($\mathbf{x}_{start}$,$\mathbf{x}_{goal}$, π^*)** simulates the trajectory $\overleftarrow{s}(\mathbf{x}_{start}, \mathbf{x}_{goal})$ using guidance policy $\dot{\mathbf{x}} = \pi^*(\mathbf{x})$ to determine trajectory cost (travel time). Another helper function, **isFree($\overleftarrow{s}$)** determines $I = \overleftarrow{s} \cap \mathcal{O}_E$ and returns a logical $I = \emptyset$, indicating that the trajectory $\overleftarrow{s}$ is free.

Predicting trajectories during calls to **getNeighbors** requires the majority of computational time during SGP execution. Two strategies based on satisficing were considered to reduce the number of paths that must be simulated. First, the function uses a distance heuristic to estimate the total cost of a path through each subgoal and computes actual costs in order of increasing estimated cost. A limit is specified such that **getNeighbors** only considers n_{limit} neighboring admissible subgoals. When $n_{limit} \geq N_{tot}$, all subgoals are considered, and the algorithm is optimal. When $n_{limit} < N_{tot}$, computation time is reduced, but optimal subgoals may be missed if the heuristic significantly underestimates actual cost. For experiments presented in this chapter $n_{limit} = 5 < N_{tot}$. The second strategy prunes successor nodes based on incremental cost difference. After the first (lowest-cost) neighbor is found, **getNeighbors** only explores a new neighbor if, based on the distance heuristic, the cost of the new neighbor is within tolerance ϵ of the previously explored neighbor. When $\epsilon = \infty$, subgoals are never pruned, and optimality is preserved. For experiments in this paper, $\epsilon = 10.0sec$.

Conditions for admissible subgoal candidates are defined in Eqn. 7.12, based on constraint boundaries and velocity vector direction. When **getNeighbors** is called, it attempts to identify two types of subgoal candidates. First, obstruction vertices are considered. In example cases presented in this paper, obstructions are convex, with piecewise linear boundaries. As shown in Fig. 7.3b, obstruction vertices allow for a range of possible velocity vector directions that satisfy the necessary conditions. As a result, admissible subgoals almost always occur at obstruction vertices. Subgoals may also occur at points along continuous constraint boundary segments where the velocity vector specified by the nominal policy $v = \pi_g(\mathbf{x})$ is parallel to the constraint

```
Data: g_node, G_all, E_all, O_E, Π
Result: G_neighbors
H = ∅;
G_edges =
 findEdgeSubgoals(E_all, g_node);
foreach g_i ∈ G_all ∪ G_edges do
    h = h(g_start, g_i) + h(g_i, g_node) +
     cost(g_node);
    H = H ∪ {h_i, g_i};
end
sort H by decreasing h_i;
G_neighbors = ∅;
while |G_neighbors| < n_limit do
    {h_i*, i*} = min(H);
    if |G_neighbors| > 1 and
     h_i* − h_last < ε then
        ←s_test =
         predictTrajectory(g_i*, g_node, Π);

        if isFree(←s_test, O_E) then
            g_i*.cost = g_node.cost +
             length(←s_test);
            G_neighbors =
             G_neighbors ∪ g_i*;
            if g_i ∉ G_all then
                G_all = G_all∪g_i
            end
        end
    end
end
```

Figure 8.1: getNeighbors()

```
Data: x_goal,x_start O_E, G_all, E_all, Π
Result: Γ_x_g
G_open = {x_goal};
while x_start ∉ G_open do
    g_node = arg min_{g∈G_open} cost(g);
    G_neighbors =
     getNeighbors(g_node);
    foreach g_i ∈ G_neighbors do
        if cost(g_i) < cost(G_all [i])
         then
            G_all [i]) = g_i;
            g_node.next = g_i;
            if g_i ∈ G_open then
                G_open(i).cost =
                 g_i.cost;
            end
        end
    end
end
g_i = x_0;
Γ_x_g = {g_i};
while g_i.next ≠ ∅ do
    g_i = g_i.next;
    Γ_x_g = {Γ_x_g, g_i};
end
```

Figure 8.2: backwards_Astar()

boundary, as in Fig. 7.3a. To accommodate for this case, **getNeighbors** also checks each obstruction edge (**getEdgeSubgoals()**) to determine if a point along it satisfies Eqn. 7.12 and should be included as a subgoal candidate. This subgoal identification process transforms the spatial constraint representation of the environment into a more biologically feasible, graphical connected state representation.

8.3 Guidance

Guidance behavior is deployed in two capacities, predictive and generative. The predictive guidance function models the way humans anticipate future trajectories, as suggested by their gaze behavior, to determine the feasibility of a candidate guidance element. The subgoal guidance algorithm uses the **predictTrajectory** (see Fig. 1) function to predict the cost and feasibility of guidance elements. The generative function is used during vehicle guidance to determine a real-time reference trajectory. The reference trajectory is used by the tracking level to generate control inputs.

8.3.1 Guidance Automaton

Both predictive and generative functions are implemented as a guidance automaton model. The automaton model generates motion as a sequence of perceptual guidance control modes, drawn from a library containing rectilinear (m_r), braking (m_b), and turning (m_t) elements. As shown in the previous chapter, guidance modes occur in this typical sequence ($\{m_r, m_b, m_t\}$), although for a specific guidance trajectory, any of these modes may be marginalized, i.e. a segment may consists of $\{m_r, m_t\}$ or $\{m_r\}$. At a spatial location in the guidance domain relative to the subgoal, the guidance task is to determine the current optimal control mode and mode parameters. For example, if an agent begins moving toward a goal state using the rectilinear mode, the agent must specify the initial heading, ψ_0, and the future point where the agent will transition into a braking or turning mode. The guidance modes, with corresponding parameters, guidance policy, and termination condition, are listed in Table 8.1.

From an initial position, x_0, the set of possible dynamic guidance sequences, $\mathcal{D}$ is parameterized by initial heading, ψ_0, as shown in figure 4.3b. The maximum value of ψ_0 occurs when the agent uses a single steering element to reach the goal. The minimum

Control Mode	Parameter	Guidance Policy	Termination
m_r Rectilin.	ψ_0	$\psi = \psi_0$ $u_{lon} = \max$	$\tau_k - \tau v = 0$
m_b Brake	-	$\psi = \psi_0$ $u_{lon} = \min$	$\tau_k = 0$
m_t Turn	k_{steer}	$\psi = k_{steer}\theta_G$	$\tau_\theta = 0$

Table 8.1: Guidance Control Modes.

value, $\psi_0 = 0$, occurs when the agent proceeds directly towards the goal, in that case requiring the agent to come to a complete stop at the goal, and rotate in-place to reach the goal heading. For most initial conditions, the minimum-cost path results from ψ_0 somewhere between these two extremes, so the agent must choose $\psi_0 \in [0, \psi_{max}]$ to maximize expected utility. Given ψ_0 chosen by the agent, guidance proceeds as follows:

1. Trim mode until $\tau_k - \tau v = 0$.
2. Brake mode until $\tau_k = 0$.
3. Turn mode until $\tau_\theta = 0$.

8.4 Integrated Model

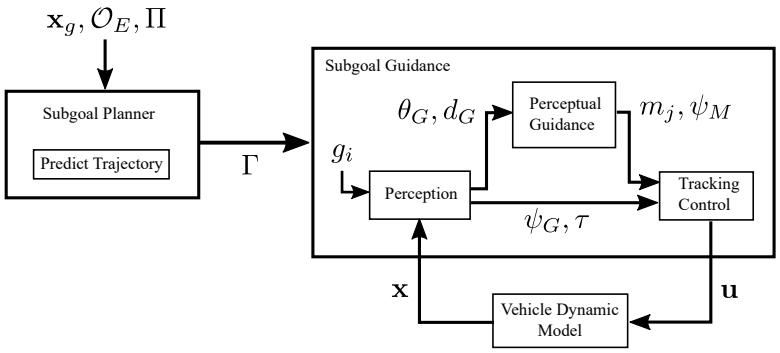


Figure 8.3: Subgoal guidance system architecture.

This section describes perceptual guidance behavior deployment in the case that the agent has perfect global domain knowledge. The perfect-knowledge case is not

realistic, but represents an ideal limit illustrating system component functions. Fig. 8.3 illustrates the integrated guidance system, with each function block described in the following sections.

8.4.1 Subgoal Planner

The subgoal planner implements the **backwards_Astar** algorithm described above, taking as input the final goal state x_g, the set of environment objects $\mathcal{O}_E$, and a guidance policy Π. The planner determines an optimal sequence of subgoals, $\Gamma = \{g_1, g_2, \ldots, g_k\}$, leading to the ultimate goal state. When the environment is exactly known in advance, subgoal planning occurs only once at the beginning of the task.

8.4.2 Subgoal Guidance

The subgoal guidance function takes as input a predetermined sequence of subgoal states, $\Gamma = \{g_1, g_2, \ldots, g_k\}$, and generates control inputs, $\mathbf{u}(t)$ that guide the vehicle system sequentially to each subgoal. This function contains three subsystems: perception, perceptual guidance, and tracking control.

Perception

The subgoal guidance perception function extracts relevant goal information from the environment state, as specified by the perceptual attention model in Chapter 5. The perception function takes as inputs the current subgoal state, g_i, and the current agent state, $\mathbf{x}(t)$. Included implicitly in the perception function is the first-person transformation needed to obtain the environment state, $\mathbf{e} = g(\mathbf{x}, \mathcal{O}_E)$. As shown in Fig. 4.1b, the perception function extracts relevant information from the environment state, $\mathbf{i} = h(\mathbf{e})$. In this implementation, relevant information consists of goal bearing error, distance to the goal, and heading relative to the goal, $\{\theta_G, d_G, \psi_G\}$

Perceptual Guidance

The perceptual guidance function implements the guidance automaton model to generate reference actions specifying the desired system trajectory. The function first identifies the optimal current dynamic mode, $m_j \in \{m^{rl}, m^{brake}, m^{turn}\}$ as a function

of goal bearing and heading, them implements the corresponding perceptual guidance mode as described in Table 8.1.

Tracking Control

The tracking control function generates system control inputs, $\mathbf{u} = \{u_{lat}, u_{lon}\}$ that regulate the system along the reference trajectory. Tracking control is based on feedback and feed-forward components that regulate heading error:

$$u_{lat} = k \cdot (\psi_M - \psi_G) + u_{lat}^{ff}(\theta_G) \tag{8.1}$$

A human operator would learn the feed-forward steering control, u_{lat}^{ff} as they gain experience with the vehicle system dynamics as described by Mettler et al [47]. A computational system could either learn this function, or compute it analytically based on system kinematics.

At high vehicle speeds system dynamics limit turn rate, preventing the vehicle from properly tracking the reference trajectory. The tracking controller compensates for this using a feedback controller that reduces vehicle speed when the lateral control input is close to the saturation limit. The parameter a indicates how close the lateral control input is to the saturation limit:

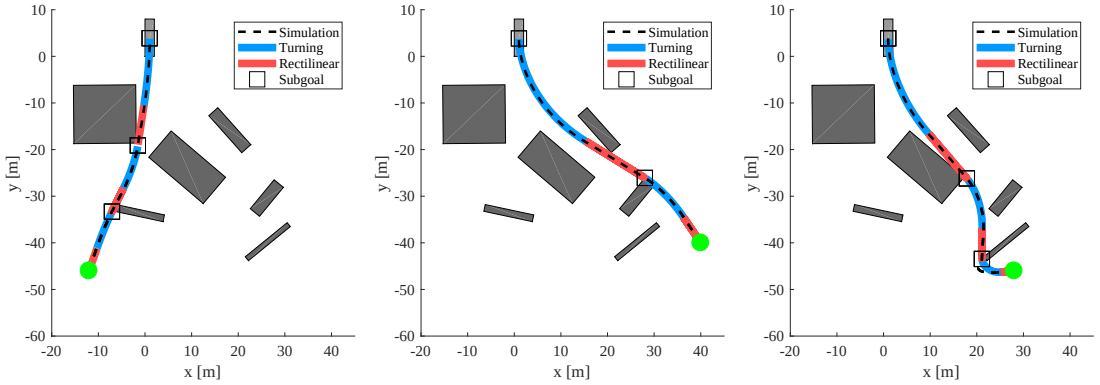
$$a(u_{lat}) = \begin{cases} 0, & \text{if } \|u_{lat}\| \leq 0.8 \\ 5 \cdot (\|u_{lat}\| - 0.8), & \text{if } 0.8 < \|u_{lat}\| \leq 1.0 \\ 1, & \text{if } \|u_{lat}\| > 1.0 \end{cases} \tag{8.2}$$

The following feedback rule then modulates speed based on the parameter a:

$$u_{lon} = 1 - k_{speed} \cdot a \tag{8.3}$$

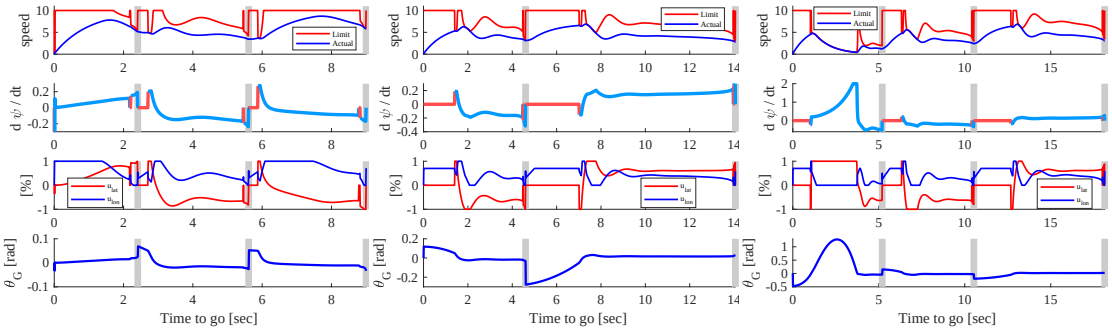
8.4.3 Results

Figs. 8.5 and 8.4 illustrate time history data and the resulting trajectories generated by the integrated model in a constrained task environment. Wide colored lines in Fig. 8.4 show the predicted trajectory generated by the guidance function. The black dashed



(a) Trial 1. (b) Trial 2. (c) Trial 3.

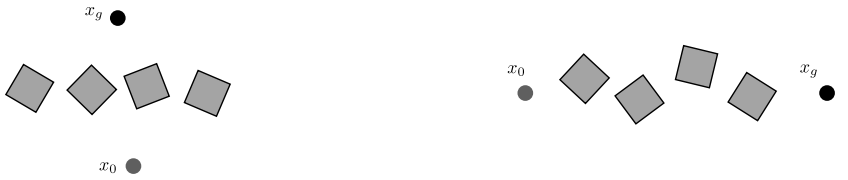
Figure 8.4: Simulation results: trajectories.



(a) Trial 1. (b) Trial 2 (c) Trial 3.

Figure 8.5: Simulation results: time histories.

line shows the actual vehicle trajectory generated by the tracking controller and vehicle dynamics. Note that near the beginning of trial 3, the tracking controller accelerates too quickly, resulting in a tracking error during the first turn. Line color indicates control mode, with blue indicating turning and red indicating rectilinear modes. Subgoals are indicated by black squares. Fig. 8.5 shows time history plots of vehicle speed, turn-rate, control inputs, and goal bearing. Subgoals are indicated by vertical gray bars. In the top set of plots, vehicle speed is shown together with the speed limit resulting from the lateral acceleration constraint, which becomes lower while the vehicle is turning. The second set of plots shows turn rate, the line is blue during rectilinear motion and red during turning. The third row of plots shows lateral and longitudinal



(a) Parallel obstacles. Single choice between many subgoals.

(b) Serial obstacles. Many sequential choices between few subgoals.

Figure 8.6: Obstacle configurations.

control inputs, and illustrates the trade-off between controls during turning caused by the lateral acceleration limit.

8.5 Computational Complexity

The size of a graph search problem is quantified by the search depth d, which is the number of nodes in the solution path, and average branching factor b, which is the number of neighbor nodes considered at each expansion step. In SGP, b is the number of neighbor subgoals that can reach the current node using a free trajectory. A* search with no heuristic has exponential time complexity, $O(b^d)$ [113]. A search heuristic may be defined in terms of goal position relative to the start, $h(\mathbf{x}) = h(\mathbf{x} - \mathbf{x}_0)$ that approximates the true cost of a path, $h^*(\mathbf{x})$. When a heuristic has bounded error $\|h(\mathbf{x}) - h^*(\mathbf{x})\| \leq \log h^*(\mathbf{x})$, search complexity is reduced to polynomial time. For a perfect heuristic, i.e. $h(\mathbf{x}) = h^*(\mathbf{x})$, search takes linear time, $O(d)$.

The search complexity of SGP depends on obstacle topology of the specific task. For a task with m obstacles each with n subgoals, both search depth and branching factor may be as high as mn, the total number of nodes. If obstacles are aligned side-by-side (i.e. in parallel) as in Fig. 8.6a, then the planner must expand all nodes at once. In this case search depth is $d = 1$, and the branching factor $b = nm$, so time complexity is $O(mn)$. If obstacles are in series between the start and goal as in Fig. 8.6b, there are $d = m$ choices between a constant $b = n$ subgoals, so complexity is polynomial, $O(m^n)$.

Solution depth and branching factor can be used to compare the time complexity of SGP with other graph search approaches. The subgoal necessary conditions guarantee that any SGP solution trajectory is composed of the minimum number of subgoals,

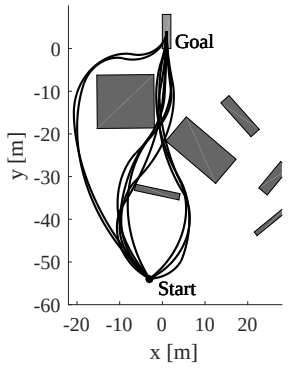
thereby minimizing the path depth d. Because the minimum number of subgoals are used, SGP will expand the minimum number of neighboring subgoals that provide complete problem domain coverage at each step, minimizing the branching factor b. In contrast, graph search methods that identify neighboring nodes using a sampling approach may either over-sample, or not provide full domain coverage, which will lead to either a sub-optimal solution or higher computation time.

8.6 Experimental Evaluation

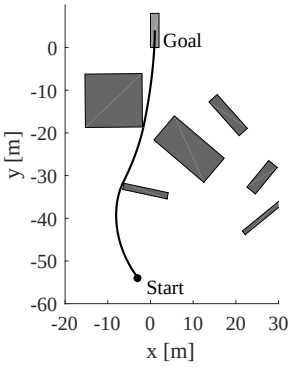
8.6.1 Evaluation Approach

Subgoal planning (SGP) performance is validated by comparing results with solutions generating using a rapidly exploring random tree (RRT*) benchmark implementation [32, 29, 118], as well as with observed human performance. In addition RRT*, like SGP, uses a guidance policy to generate dynamically feasible trajectories between nodes. Finally, it can be modified (by choosing the number of samples, k) to trade-off solution quality for planning time, which allows it to be tuned to provide similar solution quality or computational time as SGP. RRT* does not focus nodes using heuristics, so it is expected to result in lower-performance solutions and require more computational time than SGP, but RR* is still useful to provide a comparison for SGP results. Path planning results are compared in two categories, planning computation time and path performance. For comparison, both SGP and RRT* will use the perceptual guidance policy described in the previous section to generate kinematically feasible trajectories and compute costs between subgoals or nodes.

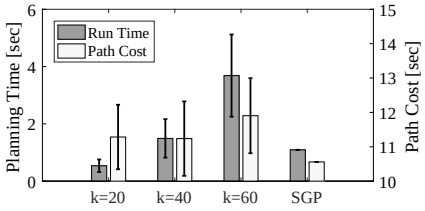
RRT* time complexity is $O(k \log k)$ for k random samples [32]. To compare RRT* complexity with SGP, the required k must be related to constraint topology quantified by m and n. The average subgoal density on a uniform obstacle field can be approximated as mn/l^2, with representative task distance l. For k uniformly distributed RRT* samples, the average node density is k/l^2. This suggests that RRT* must use at least $k \geq mn$ samples to obtain a solution, with a time complexity of $O(mn \log mn)$. Compared to SGP, RRT* may be slower when obstacles are encountered in parallel as in Fig. 8.6a ($O(mn)$ vs. $O(mn \log mn)$), however RRT* could perform better than SGP when obstacles are encountered serially as in Fig. 8.6b, ($O(m^n)$ vs. $O(mn \log mn)$). If



(a) RRT* path planning result: 20 node samples, 10 trajectories



(b) SGP path planning results.



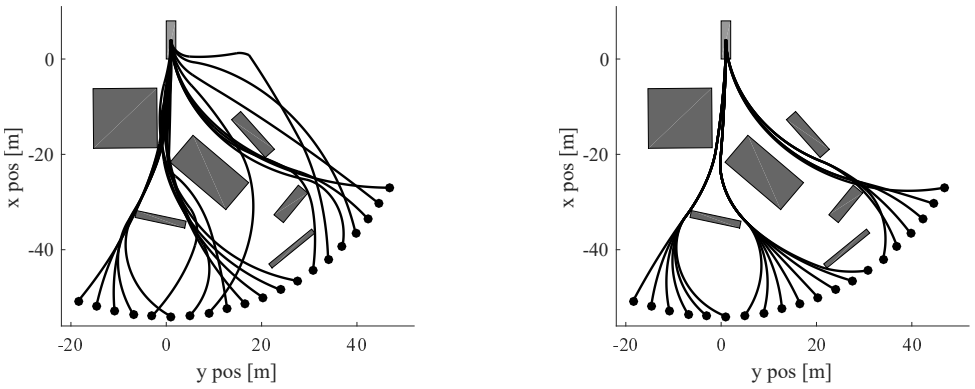
(c) SGP vs. RRT* performance comparison.

Figure 8.7: RRT* vs. SGP performance, single starting point.

SGP subgoals are concentrated in a specific region of the task domain (not uniformly distributed), then RRT* will require many more than mn nodes to find an admissible solution.

8.6.2 Planning Performance Comparison

Planning computation time is evaluated by generating a batch of solutions from a single starting point within the uniform-obstruction course. Fig. 8.7a shows a set of 10 RRT* solution trajectories, each computed using 20 samples. Fig. 8.7b shows the SGP solution trajectory. In Fig. 8.7, RRT* path cost (travel time) is plotted vs. planning time for $k = [10, 20, 30, 40]$ samples, along with the SGP path cost and planning time. RRT* results indicate that increasing the number of samples increases average planning time. The trend of path cost vs. number of samples is not clear, although when $k = 40$ the mean path cost is less than at lower sample numbers. At $k = 20$, RRT* takes approximately the same planning time as SGP, but generates solutions with lower average performance. The RRT* vs. SGP computation time comparison shows that SGP generates a solution of equal or better cost with less planning time than the RRT* benchmark. Note that many variations of RRT are possible which may allow it to achieve higher performance for specific environment or task scenarios. A key point is that SGP generates a deterministic, optimal path in similar planning time.



(a) RRT* example trajectories.

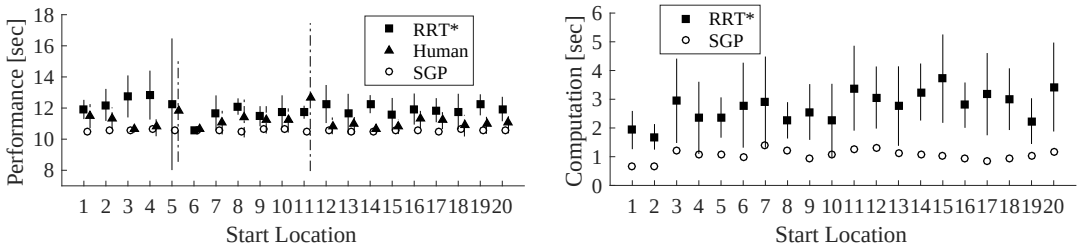
(b) SGP example trajectories.

Figure 8.8: RRT* and SGP example trajectories.

To evaluate performance, a path is generated using SGP and RRT* from each starting location in the same uniform-obstacle course used for human experiments. Fig. 8.8b shows the set of paths generating using SGP. To capture stochastic RRT* performance, ten paths are generated at each start location, each using $k = 20$ samples. Since RRT* is not always successful in finding a path, solutions are attempted until 10 solutions are found. Fig. 8.8a shows the lowest-cost trajectory generated using RRT* from each start location. Fig. 8.9a plots the resulting costs for SGP, RRT*, and human generated paths vs. start location. Both methods compute trajectories using the perceptual guidance steering policy. Vehicle speed is specified as $v = \min(v_{max}, a_{y,max}/\omega)$, but does not take into account the longitudinal acceleration limit. Humans path costs are higher because the experimentally simulated vehicle is subject this limit. Because of this, SGP results serve as a lower limit for path cost from each start location. Fig. 8.9b plots the computation time for SGP and RRT* methods vs. start time.

8.6.3 Additional Planning Examples

Next, a U-shaped course is used to evaluate robust planning performance (Fig. 8.10). RRT has difficulty planning in this type of environment because it relies on sampling points within the narrow passage. Potential field methods also have difficulty because the obstacle generates a local minima in the global value function, preventing solutions



(a) Path time: RRT*, SGP, and human results. (b) Run time: RRT* and SGP results.

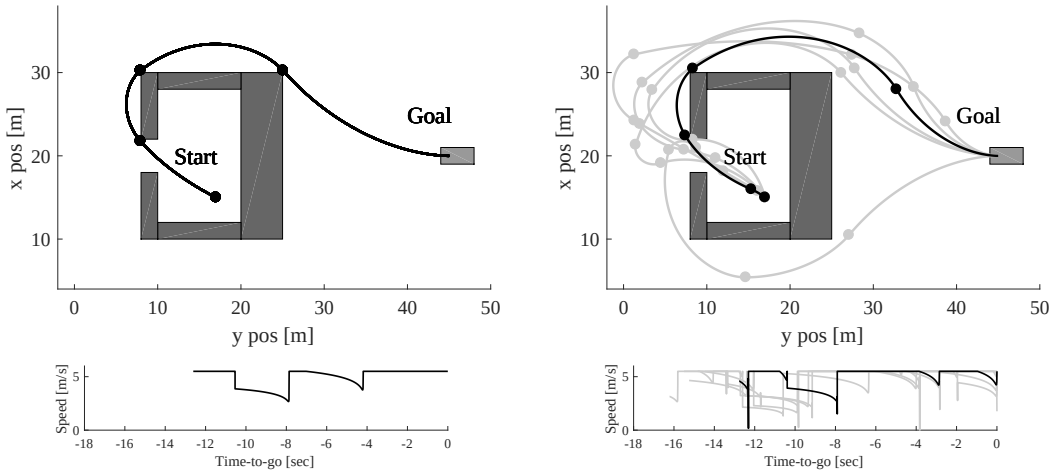
Figure 8.9: RRT* vs. SGP performance, multiple paths.

Table 8.2: SGP vs. RRT* performance.

	Uniform Course			Trap Course		
Planner	Comp. [sec]	Perf. [sec]	Succ. [%]	Comp. [sec]	Perf. [sec]	Succ. [%]
RRT*	2.72	9.27	50%	2.88	15.58	6.0%
SGP	0.996	7.64	100%	1.69	12.61	100%
Human	-	10.90	100%	-	-	-

from reaching the goal. Workarounds have been proposed to generate solutions in environments with narrow passageways, such as informed-RRT* [119] and forward-chaining [38]. These methods use a heuristic to specify intermediate points or nodes which allow the planner to avoid local minima. In constrast, SGP provides exact conditions for subgoals, and does not require special tuning to generate optimal trajectory in these cases. Table 8.2 summarizes the average computation time, path cost, and success rate for SGP vs. RRT* in the trap and uniform courses. Success rate is based on the number of solution attempts required to obtain 10 solutions.

SGP solutions are demonstrated on two additional courses: the two-block world and hallway world. The two-block world contains two skew-angled rectangular blocks placed to the left of the goal. This course shows typical obstacle avoidance in an exterior, unenclosed environment. Velocity vector field and cost-to-go plots are shown in Fig. 8.11, summarizing solutions over the entire domain. The hallway-world demonstrates path planning in an interior environment. The main part of the course is completely enclosed by walls formed from adjacent rectangular obstructions. Note that obstructions in this course contain concave corners. These vertices are immediately excluded as subgoal candidates since there is no valid trajectory passing through them that satisfies



(a) SGP solution trajectory and speed profile. Mean speed is 4.98 m/s, path distance is 62.75 m, and path time is 12.61 seconds. The SGP routes has a lower average speed than the best RRT* solution, but finds a shorter path resulting in a lower path time.

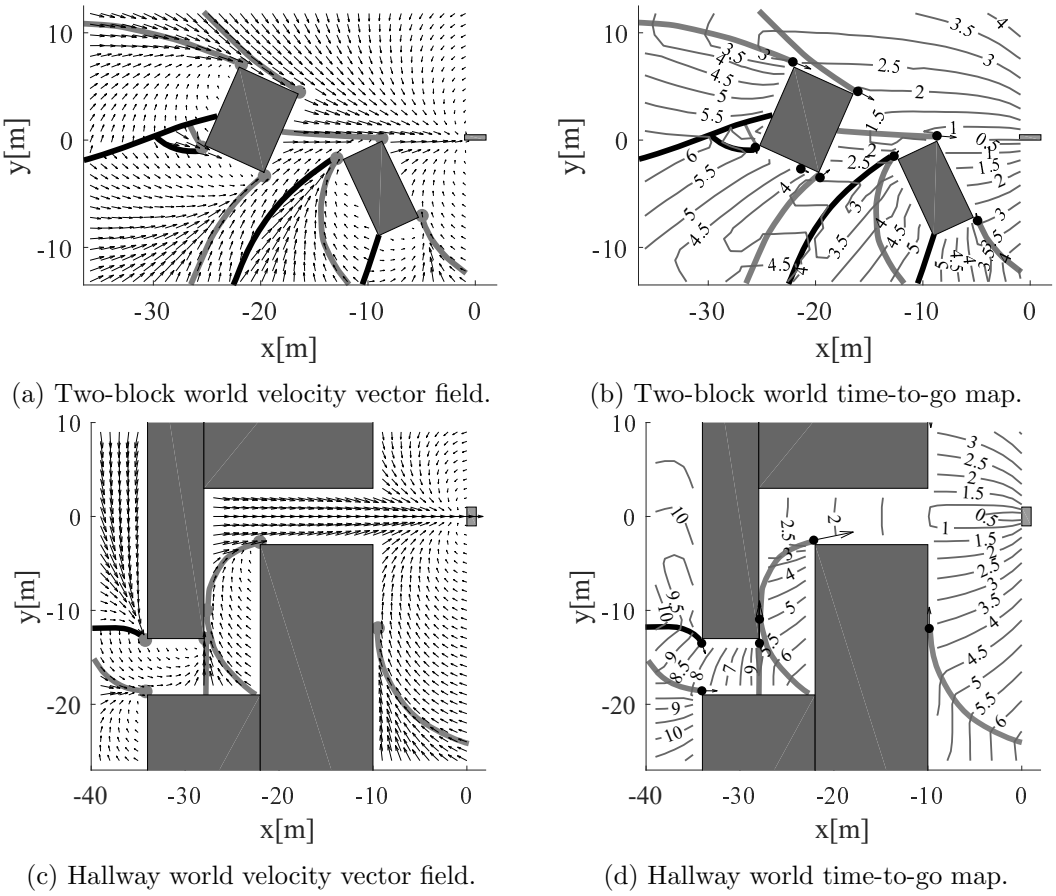
(b) RRT* was run 100 times, resulting in six successful solutions. For the fastest route, mean speed is 5.01 m/s, path distance is 63.97 m, and path time is 12.77 seconds. Average path time across all successful routes is 15.58 seconds.

Figure 8.10: Trap-course solutions.

constraints. In the hallway world, Fig. 8.11d depicts subgoals placed along continuous (linear) constraint boundaries at locations tangent to the guidance policy by **getEdgeSubgoals()**.

8.7 Deployment: First-person

The simulation of first-person guidance behavior plans and deploys dynamic behavior elements using only perceptual information available in a first-person agent perspective. Instead of planning once at the beginning of the task, subgoals are re-planned as new environment constraint information becomes available. Fig. 8.12 depicts the environment perception and subgoal planning process, consisting of the following sub-functions, as introduced by Verma and Mettler [43]:



(a) Two-block world velocity vector field.

(b) Two-block world time-to-go map.

(c) Hallway world velocity vector field.

(d) Hallway world time-to-go map.

Figure 8.11: Subgoal planning test solutions.

8.7.1 Environment Perception

In the planning function, the agent perceives the environment to identify subgoal candidates and feasible connections between subgoals. Based on the necessary conditions for subgoals, the agent identifies points along environment constraint boundaries that are expected to be tangent to optimal guidance trajectories. When constraints are assumed to be polygonal, subgoals are most likely to occur at obstacle corners. An agent perceives obstacles corners in a first-person perspective as the visible obstacle boundaries. Fig. 8.13 shows an example first-person view of the environment on the left, and on the right shows the top-down view of visible obstacles, as well as perceived subgoal candidates indicated by arrows. The set $\mathcal{G}_p$ contains the perceived subgoal candidates

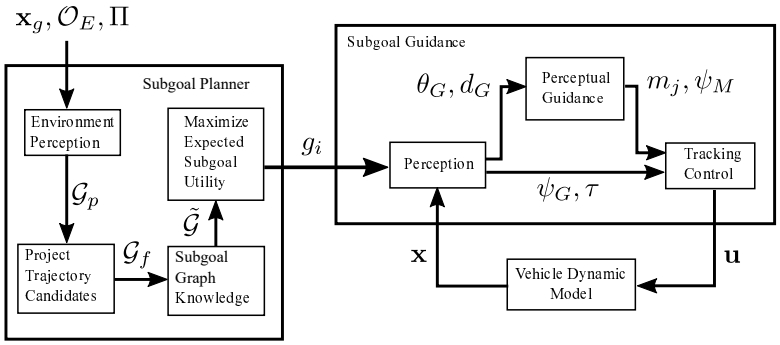


Figure 8.12: Subgoal guidance system: first-person perception.

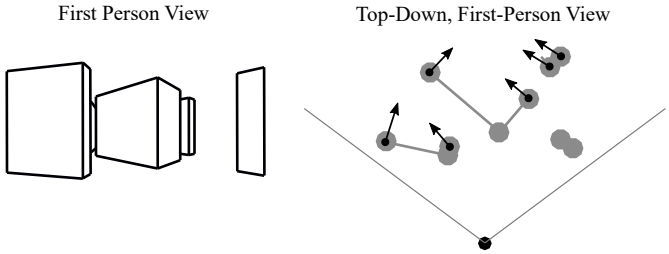


Figure 8.13: Environment perceptual information.

at an agent state $\mathbf{x}_k$.

8.7.2 Path Projection

The previous chapters on perceptual decomposition and modeling suggest that the agent determines the feasibility and utility of each subgoal candidate by predicting guidance trajectories through each visible subgoal to the final goal. Trajectory feasibility from a visible subgoal to the final goal is likely not perceivable from a first-person perspective, so the agent must estimate the distribution of costs. The estimated cost distribution is based on prior knowledge of environment objects, and meta-information about likely environment structure, such as the size and spacing of obstacles. For example, when humans navigate inside a new buiding, they may make many assumptions about the geometry of walls and common forms of interior spaces. Computationally, this distribution can be estimated using a sampling approach. Broadly, the agent would sample random subgoal locations in the unknown space, then estimate path feasibility and path cost based on available prior knowledge. The set $\mathcal{G}_F$ contains the subgoal candidates

that are determined to be feasible as the result of path projection. A subgoal is defined as feasible if the expected subgoal utility meets a minimum threshold – i.e. subgoals that result in paths that intersect obstructions would have very low utility.

Knowledge Integration

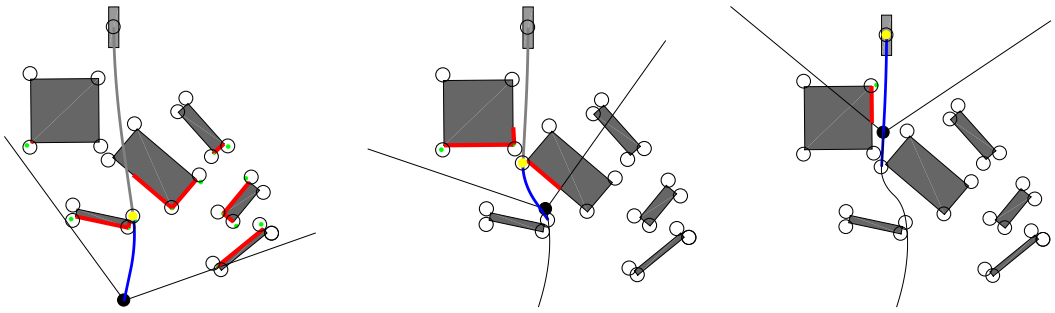
Knowledge integration involves merging the set of feasible subgoals with prior subgoal knowledge. When perceived subgoals are deterministic, the merge consists of updating graph connection knowledge. When subgoals are uncertain, subgoal location and cost is updated to the best estimate, and subgoal uncertainty is updated as a combination of prior and perceptual uncertainty. The result is an updated subgoal graph representation, $\tilde{\mathcal{G}}$. The subgoal graph representation contains the expected utility of each subgoal, and the choice of next-subgoal that is expected to maximize utility at each point.

Spatial Reasoning

Spatial reasoning, in the context of constrained motion planning and guidance, is defined as estimating the feasibility and utility of a dynamic trajectory segment without being able to visually project the candidate trajectory from the agent's current location to the goal state. The previous hypothesis about perception during guidance is that it is used to directly predict or perform an action from the agent's current state. Spatial reasoning involves using perceptual information from the environment gained in a different context to make predictions about possible actions in the current context.

This type of spatial reasoning consists of two primary functions. First, when an agent projects a trajectory from their current location to a subgoal state for which some portion of the projected trajectory not visible. In this case, the agent may use prior knowledge of environment occupancy to estimate the feasibility and utility of the projected path. In this situation, the agent is using perceptual information that was obtained during a previous task, before the agent knew it was going to be making the current specific prediction. The second case is when the agent is able to see and project a feasible trajectory between two other subgoals, neither of which the agent is currently at. In this case, the agent recognizes the utility of a segment for a future task trial.

Spatial reasoning may allow the agent to infer additional connections in the subgoal graph, and update estimates of expected subgoal utility.



(a) Prior to first subgoal. (b) Prior to second subgoal. (c) Prior to final goal.

Figure 8.14: First-person subgoal guidance example behavior.

Expected Utility Maximization

The agent selects the next subgoal, $g_i \in \tilde{\mathcal{G}}$ that maximizes expected path utility. After initially planning, subgoals are reconsidered as needed. As the agent moves along the path, they gain information about the environment through perception, and may both perceive new subgoals, or perceive information that modifies the expected utility of an already-known subgoal. If the subgoal graph representation, $\tilde{\mathcal{G}}$ changes, then the agent reconsiders and may change the target subgoal. In order to prevent limit cycles, the agent may only change subgoals if a new subgoal exceeds the current subgoal utility by a certain threshold (i.e. the sunk cost fallacy observed in human behavior).

8.7.3 Results: First-Person Perception

Fig. 8.14 shows first-person guidance behavior at three points along a solution trajectory. In Fig. 8.14a, the agent begins guidance at the location indicated by the black dot. The black lines extending from the agent location indicate the perceptual field of view. Red lines are obstacle boundaries within the field of view. Black open circles indicate all possible subgoal locations. The yellow dot indicates the subgoal the agent has chosen to maximize expected utility, with the blue line showing the projected path to that subgoal. The blue path depends on the expected velocity at the yellow subgoal, which is based on the prediction of the path from the yellow subgoal to the final subgoal, shown as a gray line. Note that the gray line intersects the obstacle, but the agent is not yet aware of the obstacle since it is beyond the agent's field of view.

Fig. 8.14b shows the agent at a point just after passing the first subgoal. At this point, the agent sees that the originally predicted path to the final goal is obstructed, so it selects and switches to a new intermediate subgoal. This sudden change in subgoal selection causes a change in reference heading, and results in a path tracking error as illustrated by the difference between agent heading and the blue trajectory in this figure. Finally, Fig. 8.14c shows the agent after passing through the second subgoal. The agent then progresses to the final goal as planned.

8.8 Discussion

8.8.1 Uncertain Environments

A fundamental issue in uncertain or unknown environments is that information flows from the goal state towards the start; each subgoal state depends on the next subgoal. This is the reverse of the direction in which the agent experiences the environment. This disparity causes two primary issues, first, the agent must make planning decisions with limited subgoal information. Information both perceived during prior runs and extrapolated from environment strongly influence agent actions. Verma et al [43] investigate environment perception, representation and learning as critical processes for motion guidance. Second, when information is limited, incorrect decisions are likely. In order to prevent catastrophic outcomes an agent must take into account information constraints in the planning process.

In uncertain environments, subgoal costs are random variables, characterized by a distribution $p(J(g_k))$. A utility function, $U : J(g_k)) \rightarrow \mathbb{R}^+$, defines a positive value to the agent of a task state based on the cost-to-go. The resulting decision policy is expressed as the maximization of expected utility of:

$$g_{k+1} = \gamma(g_k, e_k) = \underset{g_{k+1} \in G}{\arg\max}\, E\left[U(p(J(g_{k+1})))\right] \tag{8.4}$$

Stability, robustness and performance are characterized by the utility distribution of each subgoal. The utility function may be nonlinear to emphasize relevant decision making characteristics such as diminishing return of low cost paths, or avoidance of extremely high path costs [27, 120]. If the utility function $U(\cdot)$ is strictly monotonically

decreasing with cost-to-go, then the resulting finite-time stability condition is that the utility gain from each subgoal must be bounded from below by a constant $\epsilon > 0$:

$$E\left[U(J(g_k) - J(g_{k-1}))\right] \geq \epsilon \tag{8.5}$$

Eqn. 8.5 defines an information constraint on subgoals to ensure stability. If no known subgoal candidates satisfy this constraint, the agent may generate new subgoal candidates that guarantee a minimum utility. For example, a subgoal may be added that provides an option of stopping before a possible obstacle collision occurs. This safety guarantee would be similar to the approach used by Schouwenaars et al. [121] with a mixed-integer path planning optimization.

If a task will be repeated multiple time, an agent can seek of combination of maximizing performance on the current trial based on available information (exploitation), and gaining information about the task in order to improve performance in the future (exploration). Decision entropy, or informational regret [95] may be computed across the likelihoods that each subgoal is optimal, $H_{dec} = \sum_{g_i \in G} p_{min}(g_i|\mathbf{x}) \log p_{min}(g_i|x|)$, where $p_{min}(g_i|\mathbf{x})$ is the probability that choosing subgoal g_i minimizes path cost, conditioned on the current agent state $\mathbf{x}$. Decision information can also be quantified by empowerment [101], which is the maximum mutual information between actions and perception for a specific action policy. This relationship suggests that exploration can be modeled as maximizing empowerment. An agent will explore by choosing routes that allow the agent to gain information that decreases the entropy of decisions during future runs.

8.8.2 Sensory Process Integration

During a real navigation task, an agent gathers information about the environment during task execution. The agent must decide on a subgoal sequence based on estimates of subgoal locations and perception of constraint boundaries. To alleviate the computational complexity of continual environment perception, receding-horizon approaches [41] combine near and long-term planning. In addition, receding horizon planning provides a model for exploratory vs. exploitative behavior when a task is repeated for multiple trials [43]. In previous work, the environment was modeled by a cell-occupancy and cost

map over the task domain. As the agent moves through the environment, occupancy and cost is updated to improve future planning trials. This approach however requires a large amount of memory to keep a high-resolution map of large environments.

The present work on subgoal guidance suggests a way to combine receding horizon approaches with prior work investigating how humans' use interaction patterns and primitive behavior elements to represent and act on the environment. Instead of a cell occupancy map, an agent can represent the task domain as a set of subgoals connected by guidance primitives. Perception of the environment during a task identifies subgoal candidates (i.e. obstacle corners) and checks projected guidance element trajectories for constraint satisfaction. This perceptual process directly updates subgoal knowledge of positions, cost distributions, and feasible connections. Because of the focus on feasible routes, a subgoal representation can be spatially much more sparse than local sensing resolution, while still capturing essential guidance knowledge. Future work will test the validity of subgoals as an environment representation for planning and learning in large, uncertain task domains.

Chapter 9

Conclusion

9.1 Discussion

This section discusses subgoal planning implementation issues involved with dynamic tasks, higher-dimension configuration spaces, and sensory process integration.

9.1.1 Dynamic Planning Tasks

The subgoal planning approach is most applicable to tasks involving a high level of dynamic maneuvering. Verma and Mettler [122] introduces the maneuver-length scale ratio, quantifying the relationship between system maneuver capability and environment or constraint geometry. Vehicles with a high maneuverability relative to distances between obstacles (e.g. passenger aircraft cross-country routing) can adequately plan routes in the spacial domain using visibility graph or way-point planning. Agile vehicles operating at high speed and close to obstructions, such as UAVs flying in an urban environment, must account for the vehicle dynamics at the trajectory planning level. Subgoal planning provides dynamically optimized trajectories with reduced computational complexity over other full-state planning approaches. Subgoal conditions could be applied to any robotic motion of autonomous planning task to bridge the gap between discrete task and continuous motion planning.

9.1.2 High Dimension Spatial Domain

The present SGP motion planning implementation has been formulated for a 2D spatial domain ($\mathcal{W} \in \mathbb{R}^2$). In $\mathbb{R}^2$, necessary conditions specify a discrete set of subgoal candidate points, i.e obstacle vertices. A task in a higher-dimension configuration space can result in a subgoal candidate set containing continuous subsets. For example, in a navigation problem in $\mathcal{W} \in \mathbb{R}^3$ with a single spherical obstacle, the set of subgoal candidates consists of a continuous circular manifold of points where the optimal velocity vector is tangent to the sphere surface. In this case, the planner must choose one or more discrete subgoal candidates from each continuous subset while exploring neighbor nodes. Determining optimal subgoal candidates in this case involves higher computational cost. Nevertheless, a satisficing approach may be applied that picks a sub-optimal subgoal state. Future work will investigate subgoal planning for higher-dimension spaces in more detail.

9.2 Summary

This paper presents an optimal control formulation of subgoal guidance strategies inspired by human guidance behavior. Despite polynomial time complexity, SGP generates lower-cost solutions more quickly in the example planning domains presented above. SGP is able to reduce complexity by choosing subgoal locations based on environment constraints to avoid oversampling. In addition, unlike previous roadmap methods, SGP places subgoals based on the relationship between constraints and vehicle dynamics so as to generate dynamically optimal solutions. Stability conditions for deterministic planning are presented, and extended to general conditions for robustness and performance for a stochastic planning process.

The subgoal planning algorithm achieves its performance by exploiting structural properties of the guidance task, consisting of equivalence classes, to efficiently generate optimal control solutions. The resulting algorithm generates solutions that mimic human behavior, and achieve improved performance and robustness over existing approaches. Subgoal planning provides a general method of discretizing continuous environments based on system dynamics and environment constraints. SGP identifies discrete minimal-cost subgoal candidates in a continuous task domain using conditions based on

constraint topology. The A* graph-search algorithm then computes an optimal subgoal sequence among subgoal candidates.

The subgoal necessary conditions presented here, along with the concepts of partitions and guidance elements, are specific examples of task structure [15] in human behavior, and constitute a language of primitive elements, together with a grammar for optimally combining them into solutions. Future work will investigate the extension of this approach to systems with higher-dimensional configuration spaces, uncertain environments, and to the sensory processes involved in real-time motion planning tasks. Finally, future work will investigate the application of subgoal planning concepts to provide efficient autonomous guidance solutions and intuitive human-machine interactive systems.

9.3 Insights

9.3.1 Specific Insights

This analysis provide two specific insights into human perceptual guidance behavior. First, is that humans use and learn hierarchical task structure in order to interact with the environment. Agent task-environment interaction patterns drive requirements on information that must be perceived from the environment. As a result, the task-environment interaction patterns must be understood in order to understand and model human perceptual behavior during motion guidance tasks. The hierarchical behavior decomposition is a pre-requisite for the cue-based or task-based approach to understanding human perception.

Second, is that the characteristics of actual human visual behavior drive what information is most easily accessed through gaze. As a result, observing actual gaze behavior provides information about what an agent may be able to perceive, and how this influences actions the agent takes. For example, quantities such as Tau to reach an object, rotation, or translation are easily extracted from optical from of environment objects in the visual field. As a result, it is likely that humans use these quantities as part of perceptual guidance strategies. As another example, gaze motion is observed to periodically exhibit a trajectory anticipation type of motion. This indicates that a prediction of the future path of the vehicle is part of the agent's cognitive process in

addition to purely reactive perceptual guidance functions.

9.3.2 Applications

The understanding provided by the decomposition and modeling of human interaction patterns leads to several applications in human-machine interfaces and autonomous motion guidance systems. Autonomous systems can use the concepts of perceptual guidance to achieve several enhancements. First, the decomposition of guidance into interaction patterns allow a system to suboptimal, or satisficing [10] solutions at any of the levels to trade performance for computational feasibility. Second, perceptual guidance strategies, by providing a link between perceived quantities and actions, can be evaluated in terms of risk vs. performance, and use to modulate or maintain safety of an autonomous system. Finally, perceptual guidance strategies can be used as a perceptual filter to identify features in the environment most important to the current task and filter out irrelevant background information.

In human-machine systems, the identification of interaction patterns provide a way for an augmentation system to interact using a human operator's natural language. The augmentation system can generate assistive actions using the same perceptual guidance strategies that the human operator would use, so that operator maintains situational awareness, and can take over control smoothly if needed.

9.4 Future Work

This work either falls short of reaching desired goals, or opens questions in several research directions. Topics of future research are outlined in the following sections.

9.4.1 Planning

This work defines a subgoal planning algorithm that works in fully-known task-environment scenarios. The ultimate objective is to apply these insights into a receding-horizon planner that generates satisficient behavior in real-time in response to imperfect perceptual information about an uncertain environment. This type of real-time task incporates several addition issues, including the nature of an uncertain and dynamic agent environment representation, exploration vs. exploitation, and how human behavior is

motivated.

9.4.2 Guidance Behavior

While this work defines formal conditions that specify spatial subgoals, the expectation is that these underlying ideas can be extended to specify dynamic mode transition points and identify optimal dynamic modes. In the present work, this concept is used to show that the identified dynamic modes are consistent with this concept, but future work is needed to show that a certain set of dynamic modes result as solutions to an optimal control problem.

9.4.3 Perceptual Guidance

This work identifies perceptual guidance relationships in human behavior, and suggests that these perceptual guidance modes may optimize utility or an informational quantity such as empowerment, linking them to perception. It remains as future work to show that the identified perceptual guidance relationships satisfy some expected human motivational factor such as empowerment, and to show that perceptual guidance strategies such as those identified here result as solutions to an optimization problem.

9.4.4 Perceptual Attention

The present work identified two perceptual gaze functions, including an anticipatory function which is consistent with the idea that humsn rely on a predictive processing approach to generating actions. Future work is required to understanding the switching process between gaze functions. In addition, humans perceive a significant amount of visual information that is not indicated by exact gaze location, for example motion perception in the peripheral vision region. The experiments in the present work are relatively simple, and require only basic visual information to perform successfully. A topic of future work would be to gain a more complete understanding of all perceptual information, such as looming, optical flow, etc, and action behavior in more complex, high performance tasks.

References

[1] Alan Baddeley. Working memory and conscious awareness. In *Theories of memory*, pages 11–20. Lawrence Erlbaum Associates, 1992.

[2] Joshua Borah, Laurence R. Young, and Renwich E. Curry. Optimal estimator model for human spatial orientationa. *Annals of the New York Academy of Sciences*, 545(1):51–73, 12 1988.

[3] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[4] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

[5] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *Information Theory Workshop (ITW), 2015 IEEE*, pages 1–5. IEEE, 2015.

[6] Bérénice Mettler, Zhaodan Kong, Bin Li, and Jonathan Andersh. Systems view on spatial planning and perception based on invariants in agent-environment dynamics. *Frontiers in neuroscience*, 8:439, 2015.

[7] Berenice Mettler and Zhaodan Kong. Hierarchical model of human guidance performance based on interaction patterns in behavior. *arXiv preprint arXiv:1311.3672*, 2013.

[8] Bérénice Mettler. Structure and organizational principles of agile behavior: challenges and opportunities in cognitive engineering. *Cog. Crit*, 3:1–21, 2011.

[9] Zhaodan Kong and Bernard Mettler. An investigation of spatial behavior in agile guidance tasks. In *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, pages 2473–2480. IEEE, 2011.

[10] Herbert A Simon. Theories of bounded rationality. *Decision and organization*, 1:161–176, 1972.

[11] Herbert A Simon. Invariants of human behavior. *Annual review of psychology*, 41(1):1–20, 1990.

[12] Zhaodan Kong and Bérénice Mettler. Modeling human guidance behavior based on patterns in agent–environment interactions. *Human-Machine Systems, IEEE Transactions on*, 43(4):371–384, 2013.

[13] Cosma Rohilla Shalizi and James P Crutchfield. Computational mechanics: Pattern and prediction, structure and simplicity. *Journal of statistical physics*, 104(3-4):817–879, 2001.

[14] Zhaodan Kong and Bernard Mettler. On the general characteristics of 2d optimal obstacle-field guidance solution. In *Decision and Control, 2009*, pages 3448–3453. IEEE, 2009.

[15] Daniel A Braun, Carsten Mehring, and Daniel M Wolpert. Structure learning in action. *Behavioural brain research*, 206(2):157–165, 2010.

[16] Andrew Feit and Berenice Mettler. Experimental framework for investigating first-person guidance and perception. In *Systmes, Man, and Cybernetics*. IEEE, 2015.

[17] David N Lee et al. A theory of visual control of braking based on information about time-to-collision. *Perception*, 5(4):437–459, 1976.

[18] Emilio Frazzoli, Munther A Dahleh, and Eric Feron. Maneuver-based motion planning for nonlinear systems with symmetries. *Robotics, IEEE Transactions on*, 21(6):1077–1091, 2005.

[19] Bérénice Mettler and Zhaodan Kong. Mapping and analysis of human guidance performance from trajectory ensembles. *Human-Machine Systems, IEEE Transactions on*, 43(1):32–45, 2013.

[20] Jonathan Andersh, Bin Li, and Bérénice Mettler. Modeling visuo-motor control and guidance functions in remote-control operation. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 4368–4374. IEEE, 2014.

[21] C Goerzen, Zhaodan Kong, and Bernard Mettler. A survey of motion planning algorithms from the perspective of autonomous uav guidance. *Journal of Intelligent and Robotic Systems*, 57(1-4):65–100, 2010.

[22] Michael Athans and Peter L Falb. *Optimal control: an introduction to the theory and its applications*. Courier Corporation, 2013.

[23] Joao P Hespanha. *Linear systems theory*. Princeton university press, 2009.

[24] Francesco Borrelli, Alberto Bemporad, and Manfred Morari. *Constrained Optimal Control and Predictive Control*. 2009.

[25] Alberto Bemporad, Manfred Morari, Vivek Dua, and Efstratios N Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, 2002.

[26] Francesco Borrelli, Mato Baotić, Alberto Bemporad, and Manfred Morari. Dynamic programming for constrained optimal control of discrete-time linear hybrid systems. *Automatica*, 41(10):1709–1721, 2005.

[27] Stuart Russell and Peter Norvig. Artificial intelligence: A modern approach. *Learning*, 2(3):4, 2005.

[28] Maxim Likhachev, David I Ferguson, Geoffrey J Gordon, Anthony Stentz, and Sebastian Thrun. Anytime dynamic a*: An anytime, replanning algorithm. In *ICAPS*, pages 262–271, 2005.

[29] Steven M LaValle. Rapidly-exploring random trees a new tool for path planning. 1998.

[30] Priyadarshi Bhattacharya and Marina L Gavrilova. Roadmap-based path planning-using the voronoi diagram for a clearance-based shortest path. *Robotics & Automation Magazine, IEEE*, 15(2):58–66, 2008.

[31] Tomás Lozano-Pérez and Michael A Wesley. An algorithm for planning collision-free paths among polyhedral obstacles. *Communications of the ACM*, 22(10):560–570, 1979.

[32] Sertac Karaman and Emilio Frazzoli. Optimal kinodynamic motion planning using incremental sampling-based methods. In *49th IEEE conference on decision and control (CDC)*, pages 7681–7687. IEEE, 2010.

[33] Jauwairia Nasir, Fahad Islam, Usman Malik, Yasar Ayaz, Osman Hasan, Mushtaq Khan, and Mannan Saeed Muhammad. Rrt*-smart: A rapid convergence implementation of rrt. *International Journal of Advanced Robotic Systems*, 10, 2013.

[34] Carlo L Bottasso, Domenico Leonello, and Barbara Savini. Path planning for autonomous vehicles by trajectory smoothing using motion primitives. *IEEE Transactions on Control Systems Technology*, 16(6):1152–1168, 2008.

[35] Nelson Cowan, Emily M Elliott, J Scott Saults, Candice C Morey, Sam Mattox, Anna Hismjatullina, and Andrew RA Conway. On the capacity of attention: Its estimation and its role in working memory and cognitive aptitudes. *Cognitive psychology*, 51(1):42–100, 2005.

[36] Charles W Warren. Global path planning using artificial potential fields. In *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, pages 316–321. IEEE, 1989.

[37] Yoram Koren and Johann Borenstein. Potential field methods and their inherent limitations for mobile robot navigation. In *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, pages 1398–1404. IEEE, 1991.

[38] Graeme Bell and Michael Weir. Forward chaining for robot and agent navigation using potential fields. In *Proceedings of the 27th Australasian conference on*

Computer science-Volume 26, pages 265–274. Australian Computer Society, Inc., 2004.

[39] Bernard Mettler and Olivier Toupet. Receding horizon trajectory planning with an environment-based cost-to-go function. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, pages 4071–4076. IEEE, 2005.

[40] James A Primbs, Vesna Nevistić, and John C Doyle. A receding horizon generalization of pointwise min-norm controllers. *Automatic Control, IEEE Transactions on*, 45(5):898–909, 2000.

[41] Bernard Mettler and Zhaodan Kong. Receding horizon trajectory optimization with a finite-state value function approximation. In *American Control Conference, 2008*, pages 3810–3816. IEEE, 2008.

[42] B Mettler, N Dadkhah, and Z Kong. Agile autonomous guidance using spatial value functions. *Control Engineering Practice*, 18(7):773–788, 2010.

[43] Abhishek Verma and Bérénice Mettler. Computational investigation of environment learning in guidance and navigation. *Journal of Guidance, Control, and Dynamics*, pages 1–19, 2016.

[44] Emilio Frazzoli, Munther A Dahleh, and Eric Feron. A hybrid control architecture for aggressive maneuvering of autonomous helicopters. In *Decision and Control, 1999. Proceedings of the 38th IEEE Conference on*, volume 3, pages 2471–2476. IEEE, 1999.

[45] Emilio Frazzoli, Munther A Dahleh, and Eric Feron. Real-time motion planning for agile autonomous vehicles. *Journal of Guidance, Control, and Dynamics*, 25(1):116–129, 2002.

[46] V Gavrilets, Bernard Mettler, and E Feron. Human-inspired control logic for automated maneuvering of miniature helicopter. *Journal of Guidance, Control, and Dynamics*, 27(5):752–759, 2004.

[47] B Mettler, M Valenti, T Schouwenaars, E Frazzoli, and E Feron. Rotorcraft motion planning for agile maneuvering. In *Proceedings of the 58th Forum of the American Helicopter Society, Montreal, Canada*, volume 32, 2002.

[48] Mark Ebden. Gaussian processes for regression: A quick introduction. *The Website of Robotics Research Group in Department on Engineering Science, University of Oxford*, 2008.

[49] Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1):181–211, 1999.

[50] Hassan Khalil. *Nonlinear Systems*. Prentice Hall, 2002.

[51] Emmanuel Moulay and Wilfrid Perruquetti. Lyapunov-based approach for finite time stability and stabilization. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 4742–4747. IEEE, 2005.

[52] Edwin H Spanier. *Algebraic topology*, volume 55. Springer Science & Business Media, 1994.

[53] Subhrajit Bhattacharya, David Lipsky, Robert Ghrist, and Vijay Kumar. Invariants for homology classes with application to optimal search and planning problem in robotics. *Annals of Mathematics and Artificial Intelligence*, 67(3-4):251–281, 2013.

[54] Subhrajit Bhattacharya, Maxim Likhachev, and Vijay Kumar. Topological constraints in search-based robot path planning. *Autonomous Robots*, 33(3):273–290, 2012.

[55] Dan Simon. *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.

[56] Brian DO Anderson and John B Moore. *Optimal filtering*. Courier Corporation, 2012.

[57] Anthony Lazanas and J-C Latombe. Landmark-based robot navigation. *Algorithmica*, 13(5):472–501, 1995.

[58] Thomas Lemaire, Simon Lacroix, and Joan Sola. A practical 3d bearing-only slam algorithm. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 2449–2454. IEEE, 2005.

[59] Raj Madhavan and Hugh F Durrant-Whyte. Natural landmark-based autonomous vehicle navigation. *Robotics and Autonomous Systems*, 46(2):79–95, 2004.

[60] Patric Jensfelt, Danica Kragic, John Folkesson, and Mårten Björkman. A framework for vision based bearing only 3d slam. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 1944–1950. IEEE, 2006.

[61] Simone Frintrop, Patric Jensfelt, Henrik Christensen, et al. Attentional landmark selection for visual slam. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 2582–2587. IEEE, 2006.

[62] Ronen Lerner, Ehud Rivlin, and Ilan Shimshon. Landmark selection for task-oriented navigation. *Robotics, IEEE Transactions on*, 23(3):494–505, 2007.

[63] Zhaodan Kong and Bernie Mettler. Foundations of formal language for humans and artificial systems based on intrinsic structure in spatial behavior. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 3093–3100. IEEE, 2011.

[64] Bérénice Mettler, Navid Dadkhah, Zhaodan Kong, and Jonathan Andersh. Research infrastructure for interactive human-and autonomous guidance. *Journal of Intelligent & Robotic Systems*, 70(1-4):437–459, 2013.

[65] Bin Li and Bérénice Mettler. Application of trajectory segmentation techniques for operator skill evaluation. In *American Control Conference (ACC), 2013*, pages 2680–2686. IEEE, 2013.

[66] Bin Li, Bérénice Mettler, and Timonthy M Kowalewski. Towards data-driven hierarchical surgical skill analysis. *arXiv preprint arXiv:1503.08866*, 2015.

[67] David Marr and A Vision. A computational investigation into the human representation and processing of visual information. *WH San Francisco: Freeman and Company*, 1982.

[68] Michael Land, Neil Mennie, Jenny Rusted, et al. The roles of vision and eye movements in the control of activities of daily living. *Perception-London*, 28(11):1311–1328, 1999.

[69] Michael F Land. Eye movements and the control of actions in everyday life. *Progress in retinal and eye research*, 25(3):296–324, 2006.

[70] Mary Hayhoe and Dana Ballard. Eye movements in natural behavior. *Trends in cognitive sciences*, 9(4):188–194, 2005.

[71] Dario D Salvucci and Joseph H Goldberg. Identifying fixations and saccades in eye-tracking protocols. In *Proceedings of the 2000 symposium on Eye tracking research & applications*, pages 71–78. ACM, 2000.

[72] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 20(11):1254–1259, 1998.

[73] Vidhya Navalpakkam and Laurent Itti. An integrated model of top-down and bottom-up attention for optimizing detection speed. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2049–2056. IEEE, 2006.

[74] Vidhya Navalpakkam and Laurent Itti. Optimal cue selection strategy. In *Advances in neural information processing systems*, pages 987–994, 2005.

[75] Leif Johnson, Brian Sullivan, Mary Hayhoe, and Dana Ballard. Predicting human visuomotor behaviour in a driving task. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 369(1636):20130044, 2014.

[76] William H Warren and Daniel J Hannon. Direction of self-motion is perceived from optical flow. *Nature*, 336(6195):162–163, 1988.

[77] D Regan and KI Beverley. Looming detectors in the human visual pathway. *Vision research*, 18(4):415–421, 1978.

[78] D Regan and A Vincent. Visual processing of looming and time to contact throughout the visual field. *Vision research*, 35(13):1845–1857, 1995.

[79] GD Padfield. The tau of flight control. *The Aeronautical Journal*, 115(1171):521–556, 2011.

[80] Gareth D Padfield. *Helicopter flight dynamics*. John Wiley & Sons, 2008.

[81] Roger H Hoh. Acah augmentation as a means to alleviate spatial disorientation for low speed and hover in helicopters. In *American Helicopter Society International Meeting on Advanced Rotorcraft and Disaster Relief, Gifu Japan*, 1998.

[82] Abhishek Verma, Andrew Feit, and Bérénice Mettler. Investigation of human first-person guidance strategy from gaze tracking data. In *Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on*, pages 1066–1072. IEEE, 2015.

[83] Abhishek Verma and Bérénice Mettler. Analysis of human guidance and perceptual behavior in navigation of unknown environments. In *Forum 72*. American Helicopter Society, 2016.

[84] Andrew D Wilson and Sabrina Golonka. Embodied cognition is not what you think it is. *Frontiers in psychology*, 4, 2013.

[85] J Gibson. The ecological approach to human perception, 1979.

[86] David N Lee, Cathy M Craig, and Madeleine A Grealy. Sensory and intrinsic coordination of movement. *Proceedings of the Royal Society of London B: Biological Sciences*, 266(1432):2029–2035, 1999.

[87] David N Lee. Guiding movement by coupling taus. *Ecological psychology*, 10(3-4):221–250, 1998.

[88] Lingzhou Xue, Hui Zou, et al. Regularized rank-based estimation of high-dimensional nonparanormal graphical models. *The Annals of Statistics*, 40(5):2541–2571, 2012.

[89] Liang Sun, Rinkal Patel, Jun Liu, Kewei Chen, Teresa Wu, Jing Li, Eric Reiman, and Jieping Ye. Mining brain region connectivity for alzheimer's disease study via sparse inverse covariance estimation. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1335–1344. ACM, 2009.

[90] Jun Liu, Shuiwang Ji, Jieping Ye, et al. Slep: Sparse learning with efficient projections. *Arizona State University*, 6:491, 2009.

[91] Ehsan Elhamifar and Rene Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2765–2781, 2013.

[92] Behnam Gholami and Vladimir Pavlovic. Probabilistic temporal subspace clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3066–3075, 2017.

[93] Claude E Shannon and Warren Weaver. The mathematical theory of information. 1949.

[94] William H Warren. The dynamics of perception and action. *Psychological review*, 113(2):358, 2006.

[95] Naftali Tishby and Daniel Polani. Information theory of decisions and actions. In *Perception-action cycle*, pages 601–636. Springer, 2011.

[96] Daniel Polani. An informational perspective on how the embodiment can relieve cognitive burden. In *Artificial Life (ALIFE), 2011 IEEE Symposium on*, pages 78–85. IEEE, 2011.

[97] Max Lungarella and Olaf Sporns. Mapping information flow in sensorimotor networks. *PLoS computational biology*, 2(10):e144, 2006.

[98] Jan Drugowitsch, Rubén Moreno-Bote, Anne K Churchland, Michael N Shadlen, and Alexandre Pouget. The cost of accumulating evidence in perceptual decision making. *The Journal of Neuroscience*, 32(11):3612–3628, 2012.

[99] Simon B Laughlin. Energy as a constraint on the coding and processing of sensory information. *Current opinion in neurobiology*, 11(4):475–480, 2001.

[100] Philipp Schwartenbeck, Thomas FitzGerald, Raymond J Dolan, and Karl Friston. Exploration, novelty, surprise, and free energy minimization. *Frontiers in psychology*, 4, 2013.

[101] Alexander S Klyubin, Daniel Polani, and Chrystopher L Nehaniv. Empowerment: A universal agent-centric measure of control. In *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, volume 1, pages 128–135. IEEE, 2005.

[102] Robert Edge. Predicting player churn in multiplayer games using goal-weighted empowerment. Technical report, Citeseer, 2013.

[103] Sander G van Dijk and Daniel Polani. Grounding subgoals in information transitions. In *2011 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, pages 105–111. IEEE, 2011.

[104] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[105] Andrew Feit, Abhishek Verma, and Berenice Mettler. A human-inspired subgoal-based approach to constrained optimal control. In *Systems, Man and Cybernetics, 2015*. IEEE, 2015.

[106] Brian Rogers and Maureen Graham. Motion parallax as an independent cue for depth perception. *Perception*, 8(2):125–134, 1979.

[107] Helmut Altner, Josef Dudel, Otto-Joachim Grüsser, Ursula Grüsser-Cornehls, Rainer Klinke, Manfred Zimmermann, Marguerite A Biederman-Thorson, and Robert F Schmidt. *Fundamentals of Sensory Physiology*. Springer Science & Business Media, 2012.

[108] Sheng Li, Kang Li, and Yun Fu. Temporal subspace clustering for human motion segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4453–4461, 2015.

[109] Andrew Feit and Berenice Mettler. First-person perceptual guidance behavior decomposition using active constraint classification. *arXiv preprint arXiv:1710.06943*, 2017.

[110] Andrew Feit and Berenice Mettler. Subgoal planning algorithm for autonomous vehicle guidance. *Forthcoming*, 2017.

[111] Bin Li, Bérénice Mettler, and Jonathan Andersh. Classification of human gaze in spatial guidance and control. In *Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on*, pages 1073–1080. IEEE, 2015.

[112] Richard Bellman. Dynamic programming. *Princeton University Press*, 89:92, 1957.

[113] Stuart Russell and Peter Norvig. Artificial intelligence: A modern approach. 2003.

[114] Andrew Feit and Berenice Mettler. Extraction and deployment of human guidance policies. In *Cyberphysical and Human Systems, 1st Conference on*. IFAC, 2016.

[115] Tom Schouwenaars, Bernard Mettler, Eric Feron, and Jonathan P How. Robust motion planning using a maneuver automation with built-in uncertainties. In *American Control Conference, 2003. Proceedings of the 2003*, volume 3, pages 2211–2216. IEEE, 2003.

[116] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[117] Dave Ferguson, Maxim Likhachev, and Anthony Stentz. A guide to heuristic-based path planning. In *Proceedings of the international workshop on planning under uncertainty for autonomous systems, international conference on automated planning and scheduling (ICAPS)*, pages 9–18, 2005.

[118] Steven M LaValle and James J Kuffner Jr. Randomized kinodynamic planning. *The International Journal of Robotics Research*, 20(5):378–400, 2001.

[119] Jonathan D Gammell, Siddhartha S Srinivasa, and Timothy D Barfoot. Informed rrt*: Optimal sampling-based path planning focused via direct sampling of an

admissible ellipsoidal heuristic. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 2997–3004. IEEE, 2014.

[120] Daniel Bernoulli. Exposition of a new theory on the measurement of risk. *Econometrica: Journal of the Econometric Society*, pages 23–36, 1954.

[121] Tom Schouwenaars, Jonathan How, and Eric Feron. Receding horizon path planning with implicit safety guarantees. In *American Control Conference, 2004. Proceedings of the 2004*, volume 6, pages 5576–5581. IEEE, 2004.

[122] Abhishek Verma and Bérénice Mettler. Scaling effects in guidance performance in confined environments. *Journal of Guidance, Control, and Dynamics*, pages 1–12, 2016.