UNIVERSITY OF CALIFORNIA SAN DIEGO

Design of Intuitive and Risk-Perception-Aware Robotic Navigation Algorithms

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy

in

Engineering Sciences
(Mechanical Engineering)

by

Aamodh Suresh

Committee in charge:

Professor Sonia Martínez, Chair
Professor Thomas Bewley
Professor Tania Morimoto
Professor Laurel Riek
Professor Michael Yip

2022

The Dissertation of Aamodh Suresh is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2022

DEDICATION

To my parents, my strongest pillars of support.

EPIGRAPH

कर्मण्येवाधिकारस्ते मा फलेषु कदाचन ।
मा कर्मफलहेतुर्भूर्मा ते सङ्गोऽस्त्वकर्मणि ॥

Karmanyevadhikaraste Ma Phaleshu Kadachana,
Ma Karmaphalaheturbhurma Te Sangostvakarmani

You have the right to work, but that does not entitle you to the fruits of labor. Don't let these fruits be the sole purpose of your actions, and also never be inclined to not do your duty.

—Lord Krishna, in the Mahabharatha

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

Firstly, I am immensely grateful to my advisor Prof. Sonia Martínez, whose guidance has been one of the key factors in making my PhD path optimal. I have gained numerous technical, professional and life skills from interacting and working with her. Her sense of discipline, rigor, and round-the-clock availability is something I deeply admire and seek to emulate in my career. I really appreciate her patience, encouragement and understanding, especially during trying times and dealing with reviewers. Her mentorship has definitely helped me evolve into a better researcher and person.

Next, I would like to thank my collaborators Dr. Angelique Taylor, Dr. Carlos Nieto-Granda and Prof. Laurel Riek for sharing their valuable time and expertise to help me comprehend topics beyond my original grasp. I am continually broadening my skills and knowledge by interacting with them. Special thanks to Prof. Jorge Cortés for giving me the opportunity to develop and explore my passion for practical robotics in both hardware and software. I am also grateful for both his and Prof. Martínez's trust and confidence in me to mentor undergraduates and manage the Multi-agent Robotics lab. This experience has greatly helped in uncovering the manager in me, which in prior I didn't know existed.

I am also very thankful to the rest of my doctoral committee: Prof. Tania Morimoto, Prof. Thomas Bewley and Prof. Michael Yip whose constructive inputs have guided me through the course of my research.

I would also like to thank Priyank, Parth, Pio, Vishaal, Aaron, Sven, Drew, Eduardo, Mayank, Javi, Dan, Tor and all my other lab mates over the years, for the invaluable company for both my academic and personal journey.

Lastly and most importantly, I am indebted to my family and friends for helping me navigate through the non-technical challenges (which were often larger) throughout my PhD. Especially to my parents, whose unconditional love and support, like a

trampoline, have cushioned every one of my falls and at the same time propelled me to even greater heights. Special thanks to Noor, whose unwavering companionship and encouragement has given me the belief system necessary to tackle the nuances of PhD life.

Chapter 3, in full, is a reprint of the material as it appears in the publication Human-swarm interactions for formation control using interpreters, A. Suresh and S. Martínez, International Journal of Control, Automation and Systems, 18, pp. 2131–2144, 2020. A preliminary version of the work appeared in the proceedings of the IFAC Cyber-Physical-Human Systems conference, Miami, USA, pp. 83–88, December 2018, as Gesture-based human-swarm interactions for formation control using interpreters, A. Suresh and S. Martínez. The dissertation author was the primary investigator and author of these papers.

Chapter 4, in full, is a reprint of the material as it appears in the publication Planning under non-rational perception of uncertain spatial costs, A. Suresh and S. Martínez, IEEE Robotics and Automation Letters, 6(2), pp. 4133–4140, 2021. The dissertation author was the primary investigator and author of this paper.

Chapter 5, in full, is a reprint of the material as it appears in the publication Risk-Perception-Aware Control Design under Dynamic Spatial Risks, A. Suresh and S. Martínez, IEEE Control Systems Letters, 6, pp. 1802–1807, 2022. The dissertation author was the primary investigator and author of this paper.

The material in Chapter 6, in full, is currently under preparation for publication as Risk-Aware Navigation in Human-Centered Environments, A. Suresh, A. Taylor, L. Riek, and S. Martínez. The dissertation author was the primary investigator and author of these papers.

VITA

2014 B. Tech. in Mechanical Engineering, National Insititute of Technology, Surathkal, India

2016 M, S. in Mechanical Engineering, Boston University, Boston, USA

2022 Ph. D. in Mechanical Engineering, University of California San Diego, USA

PUBLICATIONS

1. A. Suresh, A. Taylor, L. Riek, and S. Martínez , "Risk-Aware Navigation in Human-Centered Environments", In preparation.

2. A. Suresh, C. Nieto-Granda, and S. Martínez , "Perceived Entropy and its Application towards Robotic Exploration", In preparation.

3. A. Suresh and S.Martínez,"Risk-Perception-Aware Control Design Under Dynamic Spatial Risks", in IEEE Control Systems Letters, vol. 6, pp. 1802-1807, 2022, IEEE, doi: 10.1109/LCSYS.2021.3134942.

4. A. Suresh and S. Martínez,"Risk-Perception-Aware Control Design Under Dynamic Spatial Risks", in IEEE American Controls Conference, June 2022, To present as part of LCSS-ACC joint submission

5. A. Suresh and S. Martínez,"Planning Under Non-Rational Perception of Uncertain Spatial Costs", in IEEE Robotics and Automation Letters, vol. 6, no. 2, pp. 4133-4140, April 2021, IEEE, doi: 10.1109/LRA.2021.3067272.

6. A. Suresh and S. Martínez,"Planning Under Non-Rational Perception of Uncertain Spatial Costs", in IEEE International Conference on Robotics and Automation, May 2021, Presentation as part of RAL-ICRA joint submission

7. A. Suresh and S. Martínez," Human-swarm Interactions for Formation Control Using Interpreters", in International Journal of Control, Automation and Systems, vol. 18, no. 1, pp. 2131-2144, 2020, Springer, doi: 10.1007/s12555-019-0497-3

8. A. Suresh and S. Martínez," Gesture based Human-Swarm Interactions for Formation Control using interpreters", in IFAC-PapersOnLine, vol. 51, no. 34, pp. 83-88, 2019, Elsevier, doi: 10.1016/j.ifacol.2019.01.033

ABSTRACT OF THE DISSERTATION

Design of Intuitive and Risk-Perception-Aware Robotic Navigation Algorithms

by

Aamodh Suresh

Doctor of Philosophy in Engineering Sciences
(Mechanical Engineering)

University of California San Diego, 2022

Professor Sonia Martínez, Chair

As robots become more integrated into society, their reasoning and actions will invariably be evaluated by human decision makers. Thus, robots need to perceive, act, and reason like humans to maintain clarity, trust, and safety. In this thesis, we consider navigation problems, which consist of designing global planning and reactive control modules for single and multiple robots. While most current navigation strategies are robot-centric, here we take a human-centric approach and design navigation algorithms that are intuitive, risk-perception-aware, and possibly non-rational (as humans often are in risky situations).

First, we focus on intuition and consider a formation control problem for a distributed robotic swarm. We develop a novel Human-Swarm Interaction (HSI) framework using the notion of an interpreter, enabling the user to control a robotic swarm's shape and formation with intuitive hand gestures. The interpreter acts an intermediary, translating a high-level shape inputs to swarm specifications and vice versa. These specifications are then translated into commands, which are calculated and executed in a decentralized manner to depict the intended shape.

Next, we focus on a single robot deployed in environments that contain generic moving sources of risk (for example, human-like obstacles requiring certain social distancing). We develop planning (via RRT*) and control (via CBFs) algorithms, that take human-like non-rational risk perception of the environment into account. We use Cumulative Prospect Theory (CPT), a non-rational model from Behavioral Economics, to construct perceived risks in the environment, capable of depicting a wide spectra of risk profiles. We introduce three new metrics: "Expressiveness", "Inclusiveness," and "Versatility" to characterize the richness of a risk model. We prove that CPT is superior in all these categories when compared to other popular models such as Conditional Value at Risk (CVaR) and Expected Risk (ER).

This is further confirmed via simulations, which show that our approach can capture a richer set of meaningful paths, representative of different risk perceptions in an environment. We also observe that a learning algorithm using CPT can approximate the risk profile of arbitrary paths in an environment better than CVaR and ER. From a controls perspective, we prove that our CBF based approach result into larger feasible control set for a robot when using CPT.

Finally, we propose a novel user study design to understand human path planning in everyday risky and uncertain environments. Considering a COVID-19 pandemic grocery shopping scenario, we ask participants to choose paths with varying risks (proximity

to sick people) and time-urgency (path length). We reveal that participants in general are willing to take more risks and time-urgent paths, contrary to the popular assumption that humans are in general risk averse. Data analysis further shows that human decision making is better captured by CPT, as compared to CVaR and ER, thus validating our CPT approach to model non-rational risk perception in navigation problems.

Chapter 1

Introduction

Autonomous robots are increasingly being deployed in human occupied spaces (HOS) and/or communicate and exucute tasks given by humans. There is a need for them to reason and act as humans do. While robots are designed to operate with detailed specifications and instructions, humans naturally communicate using abstractions. As a consequence of bounded rationality, humans can get overloaded with feedback information provided by the robot.

Also, these HOS and other general environments where robots are deployed, often are risky and uncertain. Often, the activities of robots are evaluated and monitored by human decision makers (DMs). Interestingly, Phychophysics and Behavioral Economics researchers have independently studied the human perception of various modalities from physical sensing to monetary risks. Their conclusion is surprisingly similar: humans have a fundamental non-linear perception, leading to possibly non-rational decision making [Ste70, TK92]. So DMs could be non-rational while making decisions under risk and uncertainty, leading to alternative plans and action.

In this thesis, we develop robotic navigation algorithms keeping in mind the above concerns and situations. We consider both single robots and robotic swarms, for which we design risk-perception-aware and intuitive navigation algorithms (for a single robot) and intuitive Human-Swarm Interaction framework (for robotic swarms).

We will briefly introduce and motivate the contents of this thesis in the following sections.

1.1 Human Swarm Interaction

Recent technological advancements have propelled swarm robotics in academic as well as industrial settings. Previous work has mostly focused on developing autonomous coordination of multiple robots [OSFM06, JLM03, BCM09], while the interaction of humans and swarms has been less studied [KPC+16]. Thus, according to the latest Robotics Roadmap[1], a top priority in swarm robotics is the development of unifying HSI frameworks, the elucidation of rich set of HSI examples, and their comparison. In particular, there is a need to develop novel interfaces for users to communicate their intentions and make swarms easier to interpret. Simultaneously, a swarm may require high dimensional and complex control inputs which cannot be easily handled by a human [GF07]. Additionally, individual agents in swarms are often computationally and informationally constrained. Hence, they can benefit from the use of distributed control algorithms, also available for general large-scale systems in diverse areas such as power networks [SCGG17, CCM18], traffic control [PC19, GM18], robotics [FM04, CYZ+16] and epidemics control [RLM17].

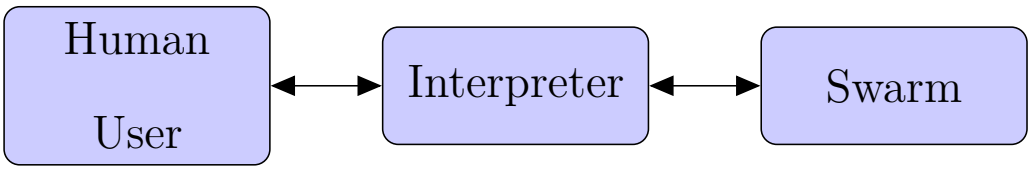


Figure 1.1. Interpreter as intermediary between user and swarm

Motivated by these aspects, we propose to build a novel supervisory interpreter (Figure 1.1) to bridge the human and the swarm, which is essential to ensure the

[1]Christensen, H. I., et al. "A roadmap for US robotics: from internet to robotics." (2016). http://jacobsschool.ucsd.edu/contextualrobotics/docs/rm3-final-rs.pdf

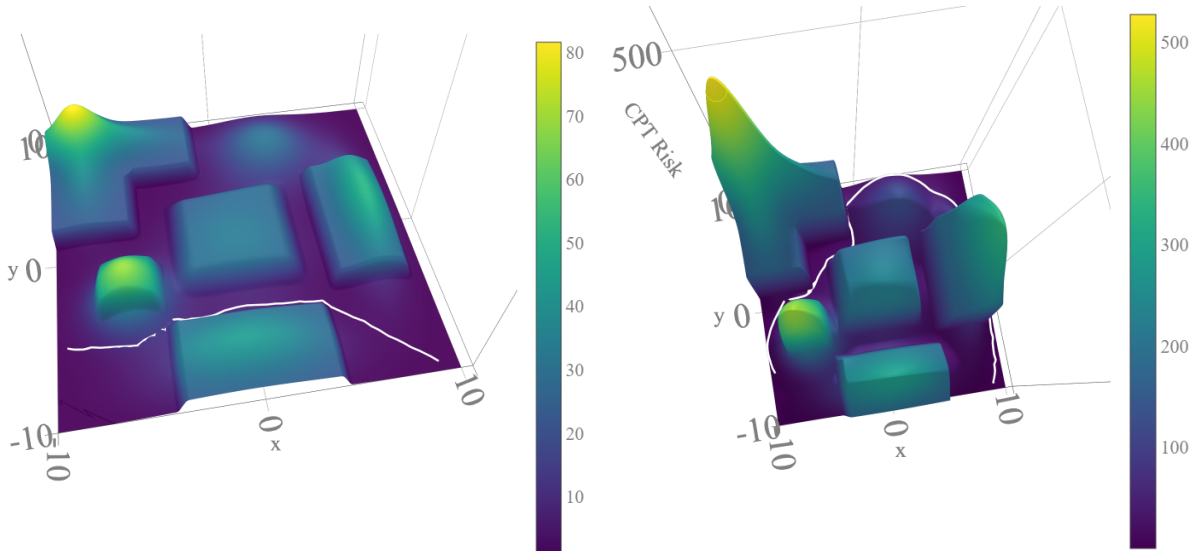
effectiveness of a HSI system. We consider the popular problem of formation control [CE15, OPA15, MMS10, KM16], where the human can draw shapes to depict desired swarm formations, instead of specifying the whole configuration. The interpreter also plans a set of sub-goal configurations in this abstract shape space, while considering the convergence properties of the swarm and translating them to swarm parameters. In this way, the user receives feedback in the abstract shape space, while the decentralized swarm need not spend precious computational resources to decode the human intention. The swarm then employs an effective distributed controller to reach the sub-goal configurations sequentially, eventually reaching the desired goal configuration.

1.2 Risk-Perception Aware Path Planning

Path planning is one of the most fundamental tasks an autonomous mobile robot needs to perform. These robots ranging from industrial manipulators to robotic swarms [CHL$^+$05, WBM18, SM20], are becoming less isolated and increasingly more interactive.

Arguably, most environments where these robots operate, have an associated spatial cost, which can lead to a robot's loss or damage. For example, an oily surface can cause a robot to slip and collide with a nearby obstacle, resulting in a crash. In more complex scenarios, a decision maker (DM) may be directly involved with the motion of an autonomous system, such as in robotic surgery, search and rescue operations, or autonomous car driving. Typically path planning treats obstacles as risk, which are perceived in an expected manner. However, in many such scenarios risk could arise from different sources and the risk perceived from these costs or losses could vary among different DMs. This motivates the consideration of richer models that are inclusive of non-rational perception of spatial costs in motion planning. With this goal, we aim to study how Cumulative Prospect Theory (CPT) [TK92] can be included into path

planning, and compare its paths with those obtained from other risk perception models like Conditional Value at Risk (CVaR) and Expected Risk (ER). Fig. 1.2 shows a preview



(a) Motion plan with expected risk

(b) Motion plan with CPT risks

Figure 1.2. Environment perception and sampling-based motion planning using a) Rational environment perception using expected risk, b) DM's Risk-Averse environment perception with the chosen path in white.

of how a nonlinear DM's perception of the environment influences the path produced to reach a goal. While Fig. 1.2a shows a rational perception of the environment using expected risk, Fig. 1.2b illustrates a non-linearly deformed and scaled surface that reflects the perception of a certain DM using CPT.

1.3 Risk-Perception-Aware Safety Critical Control Design

Safety is a desirable and necessary design constraint for any control system; specially when operated in a shared environment with a Decision Maker (DM). Arguably, most environments have associated spatial risks, whose source can vary from hard constraints (e.g. moving obstacles) to softer constraints (e.g. wind conditions). Different DMs can perceive risks differently, leading to notions of perceived risks and perceived safety from these risks.

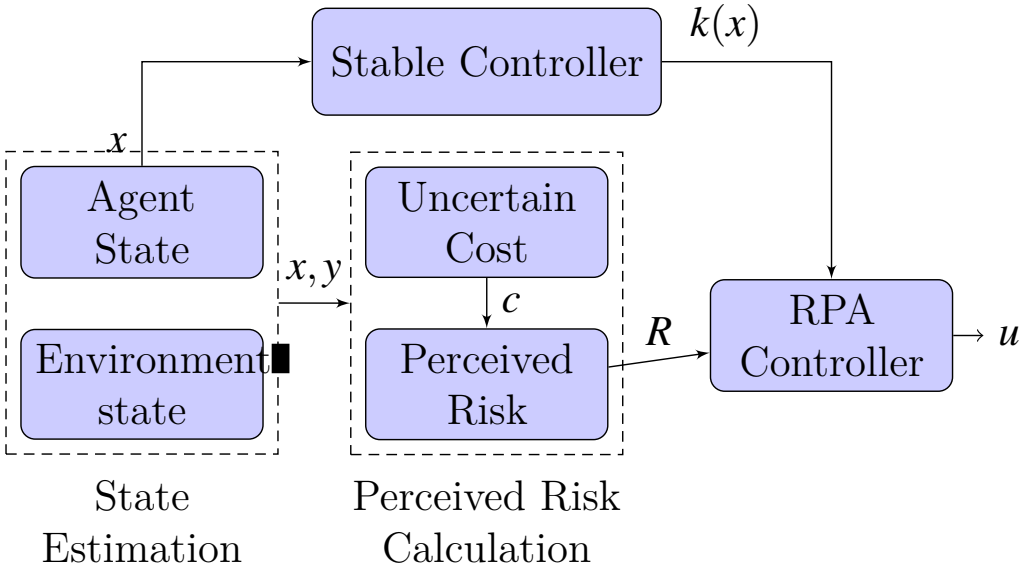


Figure 1.3. Proposed control framework. First, the agent localizes itself as x and estimates the current environment conditions y. From x, y, uncertain costmap c is determined and used to generate a perceived risk R. The risk-perception-aware (RPA) controller takes in the nominal stable state feedback control $k(x)$ and the perceived risk map R to generate safe and stable control u.

For example, this is evident from the Covid-19 era [WATD20] where people were given a risk threshold (social distancing guidelines) from dynamic risks (other moving humans) and they displayed a variety of behavior (trajectories) from being risk-insensitive to risk-averse. In such cases, existing methods assuming perfect knowledge or rational and coherent treatment (as in Expected Risk (ER) and Conditional Value at Risk (CVaR)) of risks may not suffice, which can lead to loss of trust or discomfort among DMs.

This motivates the need of richer and more inclusive modeling of risk perception to capture a variety of DMs and use them for safe control design. This work aims to bridge the gap between behavioral decision making and safety by using Cumulative Prospect Theory (CPT) as a Risk Perception Model (RPM), and Control Barrier Functions (CBFs) for safe control design. Our proposed framework and setting is visualized in Figure 1.3

1.4 Explainable AI user study design

We are increasingly seeing more robots being deployed in everyday HOS such as in shopping malls, museums, streets (as autonomous cars), etc. These environments are often crowded, and contain risk and uncertainty in terms of dynamic and chaotic human motion, noisy sensor measurements, and those from camera ego-motion. As robots become more integrated into such environments, they need to appropriately reason with these challenges and navigate in a safe and socially-acceptable manner.

One way of developing systems that navigate safely in HOS is by using models of human perception of risk. There are several models of risk perception, some of which have been introduced in Chapter 2. These models vary based on the degree of rationality assumptions made on the human decision maker (DM) subjected to risky choices. These assumptions treat human behavior ranging from completely rational and possibly risk averse (models like Expected Risk (ER), Conditional Value at Risk (CVaR), etc. [RU00]) to non-rational and possibly risk insensitive behavior (models like Cumulative Prospect Theory (CPT) [TK92]).

However, we know little about how these models compare to human perceptions of risk in a navigation setting. We are particularly interested in exploring methods to enable robots to reason with humans and explain their behaviors and actions, also known as Explainable Artificial Intelligence (AI) [XUD+19]. In other words, the AI has to "explain" itself, by opening up its reasoning to human scrutiny, resulting in better, faster, more accurate and more consistent decisions. This work will empower people to take corrective actions, if needed, based on the explanations machines give them. Furthermore, our work will enable robots and humans to develop mechanisms to adapt to each other, thereby improving interactions between them and increase robot safety.

1.5 Organization of thesis

In Chapter 2, we introduce notation used throughout the thesis and preliminary concepts that provide a broader fundamental understanding of the concepts used. In Chapter 3, we design an intuitive HSI framework for solving the formation control problem. In Chapter 4, we first introduce the notion of risk perception and develop path planning algorithms to plan paths in risky and uncertain environments. Next, in Chapter 5, we develop reactive control algorithms to avoid risky areas while trying to follow a nominal desired path. In Chapter 6, we describe a user study to understand human risk and time-urgency perception in everyday scenarios. Finally, we conclude the thesis in Chapter 7 and provide avenues for future work.

Chapter 2

Preliminaries

Here, we describe some basic notations used in the paper along with a concise description of Cumulative Prospect Theory. More details about CPT can be found in [Dha16].

2.1 Notation

We let $\mathbb{R}$ denote the space of real numbers, $\mathbb{Z}_{\geq 0}$ the space of positive integers and $\mathbb{R}_{\geq 0}$ the space of non negative real numbers. Also, $\mathbb{R}^n$ and $\mathbb{R}^{M\times n}$ denote the n-dimensional real vector space and $M \times n$ real matrices, respectively. We use $\mathbb{P}$ to denote the set of n dimensional polygonal shapes. We have $\mathbb{P} \subset \mathbb{R}^{V\times n}$, where V is any arbitrary number of vertices in a polygon. More formally, we consider the space of polygons $\mathbb{P} \subset \bigcup_{i=3}^{V} \mathbb{R}^{i\times 2}$, denoting the space of all possible polygons starting with triangles to some polygons with an arbitrarily high number of vertices V. Then we can define a shape $S \in \mathbb{P}$ which takes the coordinates of the vertices of a polygon.

In what follows, $1_M \in \mathbb{R}^M$ are column vector of ones, $\mathbf{I} \in \mathbb{R}^{M\times M}$ is the identity matrix, and $\mathbf{O} \in \mathbb{R}^{M\times n}$ denotes a matrix of zeros. Also, $\|.\|$ denotes the Euclidean norm. Given a matrix $A \in \mathbb{R}^{M\times M}$, its eigenvalues are denoted by $\{\lambda_1^A, \ldots, \lambda_M^A\}$, enumerated by their increasing real parts. The i^{th} row of a matrix A is denoted by A_i.

We have $\mathbb{R}^n$ and $\mathscr{C} \subset \mathbb{R}^n$ denote the n-dimensional real vector space and the

configuration space used for planning. We use $\circ$ for the composition of two functions f and g, that is $f(g(x)) = f \circ g(x)$. We model a tree by an directed graph $G = (V,E)$, where $V = \{1,\ldots,T\}$ denotes the set of sampled points (vertices of the graph), and $E \subset V \times V$, denotes the set of edges of the graph.

2.2 Graph Theory

Here, we introduce some basic Graph Theory notions which will be used in this thesis. Readers can refer [BCM09, GR01] for more details on Graph Theory. Consider a swarm of M agents in $\mathbb{R}^n$. Let $p_i(t), v_i(t) \in \mathbb{R}^n$ denote the position and velocity respectively of the i^{th} agent at time t. We use $p(t) \in \mathbb{R}^{M\times n}$ as the position of the whole swarm defined by $p(t) = [p_1(t)^\top,\ldots,p_M(t)^\top]^\top$. We model the communication among agents by means of an undirected ν-disk graph $\mathscr{G}_\nu = (V,E_\nu(p))$, where $V = \{1,\ldots,M\}$ denotes the set of agents (vertices of the graph), and $E_\nu(p) \subset V \times V$, denotes the set of edges. In particular, $(i,j) \in E_\nu(p)$ if and only if $\|p_i - p_j\| \leq \nu$. The entries a_{ij} of the associated adjacency matrix $A(p) \in \mathbb{R}^{M\times M}$ become 1 if $\|p_i - p_j\| \leq \nu$ and 0 otherwise. That is,

$$a_{ij} = \begin{cases} 1, & \text{if } \|p_i - p_j\| \leq \nu, \\ 0, & \text{otherwise.} \end{cases}$$

The neighbor set $\mathscr{N}_i$ for the i^{th} agent is given by $\mathscr{N}_i := \{j \mid a_{ij} = 1\}$. Associated with $\mathscr{G}_\nu$, we consider a weight-balanced weighting $W(t) \in \mathbb{R}^{M\times M}$, where $W(t)$ is the metropolis weight matrix corresponding to the communication graph $\mathscr{G}_\nu$; see [XB04].

$$w_{ij} = \begin{cases} 1/(1+\max\{d_i,d_j\}), & \text{if } (i,j) \in E_\nu(t), \\ 1 - \sum_{k\in\mathscr{N}_i}(1/(1+\max\{d_i,d_j\})), & \text{if } i = j, \\ 0, & \text{otherwise.} \end{cases} \tag{2.1}$$

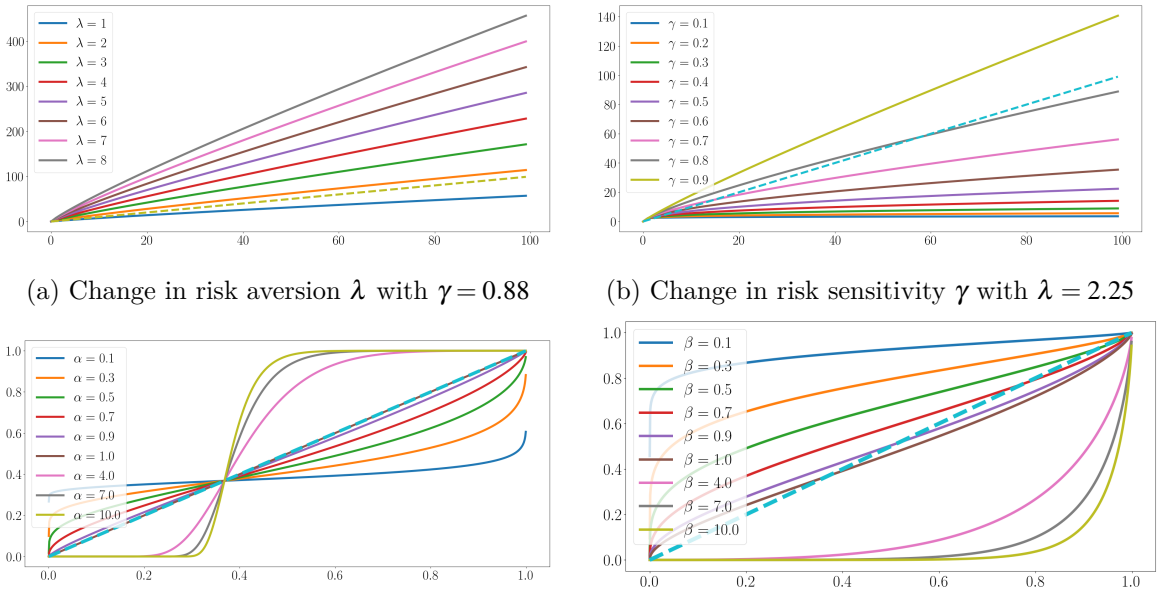
Since we consider an undirected graph the matrix, W is symmetric and doubly stochastic. The graph $\mathscr{G}_w$ is balanced as $1_M W = W 1_M^\top = 1_M$. We denote by $D \in \mathbb{R}^{M\times M}$

the diagonal degree matrix of $\mathscr{G}$ with d_i, the degree of node i, being the i^{th} diagonal entry of D. The Laplacian matrix $L \in \mathbb{R}^{M\times M}$ of the graph $\mathscr{G}_{\mathbf{v}}$ is given by $L = D - A$, and the normalized laplacian matrix is given by $L^N = D^{\frac{-1}{2}} L D^{\frac{-1}{2}}$. Similarly the weighted Laplacian matrix is given by $L^W = \mathbf{I} - W$. The connectivity properties of a graph are captured by the second smallest eigenvalue λ_2 of the Laplacian matrix L. We can also express connectivity in terms of λ_2^W and λ_2^N. We can say that the respective graph is connected if $\lambda_2^W, \lambda_2^N > 0$, and connectivity increases with increase in λ_2^W, λ_2^N.

2.3 Cumulative Prospect Theory

Cumulative Prospect theory (CPT) [TK92] is a Nobel prize winning theory, which tries to model human decision making under risk and uncertainty. CPT is a non-rational decision making model which incorporates non-linear perception of uncertain costs. Traditionally it has been used in scenarios of monetary outcomes such as lotteries [TK92] and the stock market [Dha16]. Let us suppose a DM is presented with a set of prospects $\{\rho^1, \ldots, \rho^k, \ldots, \rho^K\}$, representing potential outcomes and their probabilities, $\rho^k = \{(\rho_i^k, p_i^k)\}_{i=1}^M$. More precisely, there are M possible outcomes associated with a prospect k, given by $\rho_i^k \in \mathbb{R}_{\geq 0}$, for $i \in \{1, \ldots, M\}$, which can happen with a probability p_i^k. The outcomes are arranged in a decreasing order denoted by $\rho_M^k < \rho_{M-1}^k < \ldots < \rho_1^k$ with their corresponding probabilities, which satisfy $\sum_{i=1}^M p_i^k = 1$. The outcomes of prospect k may be interpreted as the random cost[1] of choosing prospect k. We define a utility function, $v : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ modeling a DM's perceived cost and $w : [0,1] \to [0,1]$ as the probability weighting function which represents the DM's perceived uncertainty. While previous literature have used various forms for these functions, here we will focus on

[1]CPT has an alternate perception model for random rewards [Dha16], which is not used here since we are interested in cost perception.



(a) Change in risk aversion λ with $\gamma = 0.88$

(b) Change in risk sensitivity γ with $\lambda = 2.25$

(c) Change in α with $\beta = 1$

(d) Change in β with $\alpha = 0.74$

Figure 2.1. Variation of risk aversion, risk sensitivity and uncertainty perception using CPT. (a)-(b) show risk perception with x-axis indicating the associated risk, ρ, and the y-axis showing the perceived risk, v. The dotted line indicates the line $v = \rho$. (c)-(d) show uncertainty perception with x-axis indicating probabilities p and y-axis showing their perception w, with the dotted line depicting $w = p$

CPT utility function v taking the form:

$$v(\rho) = \lambda \cdot \rho^{\gamma}, \tag{2.2}$$

where $0 < \gamma < 1$ and $\lambda > 1$. Tversky and Kahneman [TK92] suggest the use of $\gamma = 0.88$ and $\lambda = 2.25$ to parametrize an average human in the scenario of monetary lotteries, however this may not hold for our application scenario. The parameter λ represents the coefficient of cost aversion with greater values implying stronger aversion indicative of higher perceived costs, as indicated in Figure 2.1a. The parameter γ represents the coefficient of cost sensitivity with lower values implying greater indifference towards cost ρ which is indicated in Figure 2.1b.

We will be using the popular Prelec's probability weighting function [Dha16,

D.P98] indicative of perceived uncertainty, which takes the form:

$$w(p) = e^{-\beta(-\log p)^{\alpha}}, \alpha > 0, \beta > 0, w(0) = 0. \tag{2.3}$$

Figures 2.1c and 2.1d show changing uncertainty perception resulting from varying α and β respectively. By choosing low α and β values, one can get "uncertainty averse" behavior with $w(p) > p$, implying that unlikely outcomes are perceived to be more certain, as seen on Figures 2.1c and 2.1d. When $w(p) < p$, "uncertainty insensitive" behavior is obtained implying that the DM only considers more certain outcomes, which can be observed with high α and β values. These perceived probabilities are further considered in a cumulative manner. Cumulative functions $\Pi := \{\pi_1, ..., \pi_M\}$ is used to non-rationally modify the perception of the probabilities $p_i(x)$ in a cumulative fashion. Defining a partial sum function $S_j(p_1, \ldots, p_M) \triangleq \sum_{i=j}^{M} p_j$ we have

$$\pi_j = w \circ S_j(p_1, \ldots, p_M) - w \circ S_{j+1}(p_1, \ldots, p_M), \tag{2.4}$$

where we employ the weighting function w from (2.3).

These concepts illustrate the nonlinear perception of cost and uncertainty, a DM under consideration can be categorized by the parameters $\Theta = \{\alpha, \beta, \gamma, \lambda\}$. Using the non-linear parametric perception functions v and w, CPT calculates a value function $R^c(\rho)$, indicating the perceived risk value of the prospect ρ. The value function is given by:

$$R^c(x) \triangleq \sum_{j=1}^{M} (v \circ \rho_j(x))(\pi_j \circ p(x)) \tag{2.5}$$

This value $R^c(\rho)$ signifies the perceived risk by a DM characterized by the parameters Θ, when faced with an uncertain cost ρ.

2.4 Sampling based path planning with Rapidly Exploring Random Tree* (RRT*)

Algorithm 1: RRT*

Input: T, x_s, x_g ; Output : $G(V,E), P$
$V \leftarrow x_s$, $E \leftarrow \phi$, $J^c_{\text{cum}}(x_s) \leftarrow 0$;
for $i \in \{1,\ldots,T\}$ do
  $G \leftarrow (V,E)$; $x_{\text{rand}} \leftarrow \textit{Sample}()$;
  $x_{\text{nearest}} \leftarrow \textit{Nearest}(G, x_{\text{rand}})$; $x_{\text{new}} \leftarrow \textit{Steer}(x_{\text{nearest}}, x_{\text{rand}})$;
  if ObstacleFree($x_{\text{new}}, x_{\text{nearest}}$) then
    $V \leftarrow V \cup x_{\text{new}}$; $x_{\min} \leftarrow x_{\text{nearest}}$;
    $X_{\text{near}} \leftarrow \textit{Near}(G, x_{\text{new}}, \gamma_{\text{RRT}^*}, d)$; $c_{\min} \leftarrow J^c_{\text{cum}}(x_{\text{nearest}}) + J^c(x_{\text{nearest}}, x_{\text{new}})$;
    for $x_{\text{near}} \in X_{\text{near}}$ do
      $c' \leftarrow J^c_{\text{cum}}(x_{\text{near}}) + J^c(x_{\text{near}}, x_{\text{new}})$;
      if $c' < c_{min}$ then
        $x_{\min} \leftarrow x_{\text{near}}$; $c_{\min} \leftarrow c'$;
      end
    end
    $J^c_{\text{cum}}(x_{\text{new}}) \leftarrow c_{\min}$; $E \leftarrow E \bigcup (\{x_{\text{near}}, x_{\text{new}}\})$;
    for $x_{\text{near}} \in X_{\text{near}}$ do
      $c' \leftarrow J^c_{\text{cum}}(x_{\text{new}}) + J^c(x_{\text{new}}, x_{\text{near}})$;
      if $c' < J^c_{\text{cum}}(x_{\text{near}})$ then
        $x_{\text{par}} \leftarrow \textit{Parent}(x_{\text{near}}, G)$; $E \leftarrow (E \setminus (\{x_{\text{par}}, x_{\text{near}}\})) \bigcup (\{x_{\text{new}}, x_{\text{near}}\})$;
        $X_{\text{chld}} \leftarrow \textit{Children}(x_{\text{near}}, G)$;
        for $x_{\text{chld}} \in X_{\text{chld}}$ do
          $J^c_{\text{cum}}(x_{\text{chld}}) \leftarrow J^c_{\text{cum}}(x_{\text{chld}}) - J^c_{\text{cum}}(x_{\text{near}}) + c'$
        end
        $J^c_{\text{cum}}(x_{\text{near}}) \leftarrow c'$
      end
    end
  end
end
$P \leftarrow \textit{Path}(G, x_s, x_g)$;

Consider a configuration space $\mathscr{C} \subset \mathbb{R}^n$, where the robot can move around. Let there be obstacles present in the configuration space occupying the region $\mathscr{O} \subset \mathscr{C}$, through which the robot cannot travel. Then the free configuration $\mathscr{C}^F$ is given by $\mathscr{C} \setminus \mathscr{O}$. A path η from a start point $x \in \mathscr{C}$ to a goal point $y \in \mathscr{C}$ is defined as $\eta : [0,1] \rightarrow \mathscr{C} \eta(0) = x, \eta(1) = y$. A path η is feasible to take if it entirely lies in the free config-

uration space $\mathscr{C}^F$ Every path η also has an associated cost (usually distance metric) $J_{\text{cum}}(\eta) \in \mathbb{R}_{\geq 0}$.

The common task of path planning is to produce a feasible path η^* from a given start point x_s to goal point x_g, such that the cost $c(\eta)$ is minimized. RRT* [KF11] is a sampling based planning technique which produces a tree of feasible paths rooted at the start point x_s. So given a number of iterations T and a start point $x_s \in \mathscr{C}$, RRT* produces a graph $G(V,E)$, which represents a tree rooted at x_s whose nodes V are sample points in the configuration space and the edges E represent the path between the nodes in V. Let $J^c_{\text{cum}} : \mathscr{C} \to \mathbb{R}_{\geq 0}$ be a function that maps $x \in \mathscr{C}$ to the cumulative cost to reach a point x from the root x_s of the tree $G(V,E)$. This function is used to keep track of reaching each node in the tree G. RRT* is also asymptotically optimal, that is, if $T \to \infty$, then the output path $\eta \to \eta*$. The algorithm is summarized in Algorithm 1 and explained below.

- *Sample*(): Returns a pseudo-random sample $x \in \mathscr{C}$ drawn from a uniform distribution across $\mathscr{C}$.

- *Nearest*(G,x): Returns the nearest node according to the Euclidean distance metric from x in tree G.

- *Steer*(x_1,x_2) returns
$$\begin{cases} x_2, & \text{if } \|x_2 - x_1\| \leq d \\ x_1 + d\frac{x_2 - x_1}{\|x_2 - x_1\|}, & \text{otherwise.} \end{cases}$$

- ObstacleFree(x,y) : returns whether the line between x,y, entirely lies in the free configuration space $\mathscr{C}^F$

- *Near*(G,x,γ_{RRT^*},d): returns a set of nodes $X \in V$ around x, which are within a radius as given in [KF11].

- $Parent(x, G)$: Returns the parent node of x in the tree G.
- $Children(x, G)$: Returns the list of children of x in G.
- $Path(G, x_s, x_g)$: Returns the path from the nearest node to x_g in G to x_s.

2.5 Safety critical control with Control Barrier Functions (CBF)

Control barrier functions (CBF) apply set invariance principles to guarantee safety of an agent $x(t) \in \mathscr{X}$, assumed to be moving with control affine dynamics:

$$\dot{x}(t) = f(x) + \sum_{i=1}^{m} g^i(x)u^i = f(x) + Gu,\ u \in \mathbb{R}^m,\ G \in \mathbb{R}^{n \times m} \tag{2.6}$$

where $f : \mathscr{X} \to \mathscr{X}$ is the drift associated with the agent and $u_i \in \mathbb{R}$ is the i^{th} control input and there are m control inputs.

Let us consider a function $h : \mathscr{X} \to \mathbb{R}$ to encode safety. We define safe sets considering super level sets of the function h in our environment $\mathscr{X}$ as follows:

$$\mathscr{X}_{\text{safe}} = \{x \in \mathscr{X} | h(x) \geq 0\}, \tag{2.7a}$$

$$\partial \mathscr{X}_{\text{safe}} = \{x \in \mathscr{X} | h(x) = 0\}, \tag{2.7b}$$

$$\mathscr{X}^{\circ}_{\text{safe}} = \{x \in \mathscr{X} | h(x) > 0\}. \tag{2.7c}$$

Now let us consider the set of control inputs $K_{\text{cbf}}(x)$ defined by:

$$K_{\text{cbf}}(x) = \{u \in \mathscr{U} : \mathscr{L}_f h(x) + \mathscr{L}_g h(x)u \geq -\alpha(h(x))\}, \tag{2.8}$$

where α is an extended class $\mathscr{K}_\infty$ function. Now we can formally define CBF as follows:

Definition 1. (CBF) Let the sets $\mathscr{X}_{\text{safe}}$ and K_{cbf} be defined according to (2.7) and (2.8) respectively. Then h is a CBF if there exists an extended class $\mathscr{K}_\infty$ function α such that the set $K_{\text{cbf}}(x)$ is non-empty for all $x \in \mathscr{X}_{\text{safe}}$.

With the above definition we can state the following lemma:

Lemma 1. [ACE+19] Given the sets X_{safe} and K_{cbf} defined according to (2.7) and (2.8) respectively, let h be a CBF defined according to Definition 1 with $\frac{\partial h}{\partial x}(x) \neq 0$ for all $x \in \partial X_{\text{safe}}$. Then any control signal $u(x) \in K_{\text{cbf}}(x)$ for the system (2.6) ensures safety of the system.

So if we can find a CBF h satisfying (2.7) with the assumptions in Lemma 1, we can guarantee that the system will be safe.

Chapter 3

Human-Swarm Interactions for Formation Control using Interpreters

In this chapter, we develop a novel Human-Swarm Interaction (HSI) framework for formation control using the notion of an interpreter, enabling the user to control a robotic swarm's shape and formation using abstraction. The user conveys their intended commands by drawing shapes through arm gestures and motions which are recorded by an off-the-shelf wearable device. We propose a novel interpreter system, which acts as an intermediary between the user and the swarm to simplify the roles of both. The interpreter takes in high level input in the form of shapes drawn by the user, and translates it into swarm control commands by planning in the shape space using novel shape morphing dynamics (SMD), which is also used for user feedback. The proposed interpreter employs machine learning, estimation and optimal control techniques to translate the users intention into swarm control parameters. The dynamics of the swarm are realized by means of a novel decentralized formation controller based on distributed linear iterations and dynamic average consensus. Theoretical guarantees of convergence along with convergence rate of the proposed swarm controller are given. The resulting shape morphing dynamics are illustrated and discussed through simulations. The entire framework is demonstrated theoretically as well as experimentally in a 2D environment, with a human controlling a swarm of simulated robots in real time with the help of a

Graphical User Environment (GUI).

3.1 Literature Review

According to recent surveys on HSI [KPC+16] and human multi agent systems [Fra17], humans either take a supervisory [SF12], direct [SFKE15], shared [FSR+12], or environmental [WS16] control role in an HSI framework. Most of the HSI frameworks design have been user-centric and focused on direct control of swarms, either through teleoperation or proximal interaction; see e.g. [JGGD14, SFKE15]. Due to complicated swarm dynamics, the human will likely be quickly overwhelmed and would not make the best decisions [Sur16, SS16]. Previously authors have considered automated sequencing of swarm behaviors [NCS17]. However, this general formulation with a discrete search space and combinatorial algorithm can just deal with a few discrete behaviors at a time.

Previous work on using gestures for multi-robot control have focused on external sensors like cameras [MWVM13] and infrared sensors [POND14], which require setup time as well as good environment conditions. Some other works like [HOGD18] have tried to find the most separable gestures which can be used to control a robotic system. Regarding internal sensors for swarm control, smart watches [VSSF17] for dynamic gestures and the MYO armband [NCLS17] for static gestures have been used. Notably, [NCLS17] use the inbuilt static gestures which are inaccurate and non-intuitive for swarm specific interactions, whereas we use custom static and dynamic gestures employing just a single device. More recently, gesture based techniques along with speech, vision and motion have been used together to interact with small teams of robots in [AMLL+15] and [GGD16]. These works rely on proximal multi-modal interaction schemes which require complex hardware setup to interpret the human intention, which is not practical for large scale swarms.

With respect to formation control for large swarms, [RCN14] and [AMBR+12]

have only used predefined shapes and images as inputs for the swarm, which facilitates only supervisory control for a HSI system. However, these works lack of an interpreter, and the main focus is on display development to showcase known distributed algorithms.

3.2 Contributions

We propose a novel HSI framework in which we consider both a user and a decentralized swarm, with an interpreter acting as a bridge between the two. By means of it, the user can communicate their intentions naturally by drawing shapes, without having an in depth understanding of the swarm dynamics. At the same time, the swarm receives suboptimal control subgoals in their domain and need not spend resources and time to decode the user's intention. The paper presents contributions in the following aspects: (i) We present a novel planner embodied in the interpreter to devise control subgoals in the abstract shape domain that are efficient for a swarm. (ii) Through the principles of optimal switching control, we present a procedure to simulatenously plan in the continuous shape space and discrete parameter space (swarm communication range) to generate swarm subgoals, which to the best of our knowledge has not been done before. (iii) We present a novel discrete second-order distributed formation controller for the swarm that uses notions from the Jacobi Overrelaxation Algorithm and dynamic average consensus to guarantee the convergence of a (second-order integrator) swarm to a desired shape, scaling, rotation and displacement. Our controller relies only on position information of each agent and communication with their neighbors using variable communication radii, which provides a practical setting. (iv) On the human-interpreter interaction side, we formulate a novel intention decoder using Kalman Filtering and Hidden Markov Model (HMM) for simultaneous dynamic and static gesture decoding utilizing signals from Inertial Measurement Unit (IMU) and Electromyographic (EMG) sensors, respectively. (v) A final contribution is provided by the system integration of

diverse tools that serve to articulate our HSI framework.

We wish to clarify that, our proposed interface is motivated by a future application scenario in which a computer mouse cannot be used effectively, such as in Virtual Reality (VR) and Augmented Reality (AR) environments. However, the development of this AR/VR environment system and the use of our interface in it is currently beyond the scope of this paper. Similarly, a thorough human study of the interface and its comparison with other platforms is also out of the scope of this work. Here, we mainly focus on the development of improved signal processing algorithms that can set the initial stage for future work. Also, this manuscript significantly extends our preliminary version in [SM19] by including an extended literature review, discussion on gesture decoding, full analytical proofs of the swarm controller, a new discussion section on the shape morphing dynamics, additional details on our GUI and planner, and additional results for accuracy testing of intention decoder and the proposed planner.

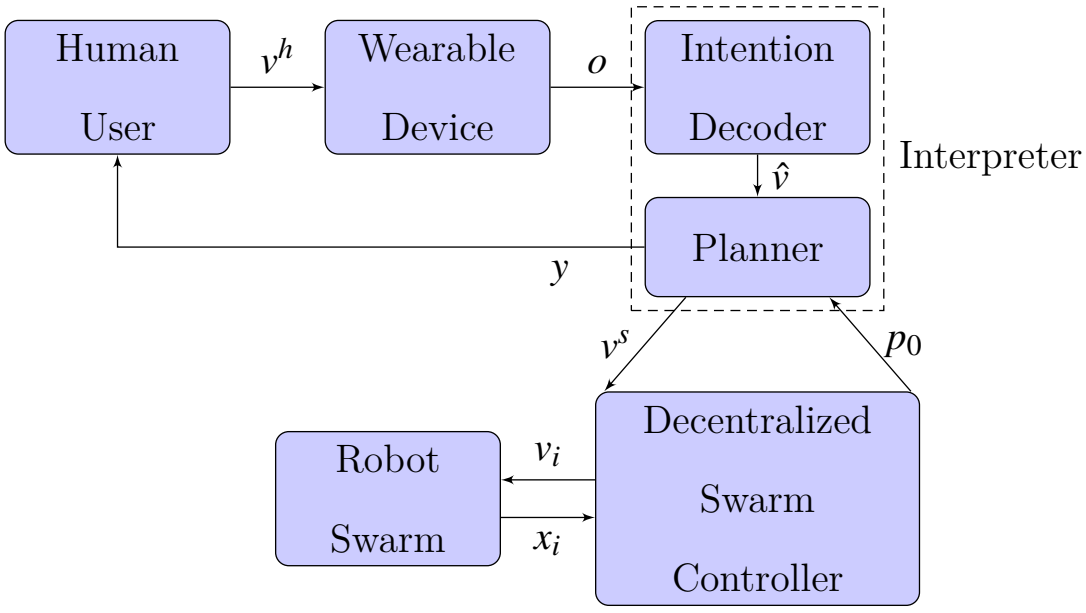


Figure 3.1. Proposed HSI framework. The user communicates their intent v^h through the Myo armband which produces observations o. The decoder estimates the user intent $\hat{v}$ from o. The planner using $\hat{v}$, plans a set of intermediate goals v^s. The decentralized controller in agent i reaches v_i^s by computing the velocities v_i. The interpreter uses the swarm state $p(t)$ to give feedback $y(t)$ to the human.

3.3 Proposed Framework and Problem Formulation

Here, we first introduce the various timescales involved in the interactions, and propose a new HSI framework, while providing a description of its components. Later, we identify the various problems to be solved to implement this framework.

Timescales Involved. We assume that the interactions between the human, interpreter and the swarm, and the dynamic update of the swarm, may occur at time scales that go from coarser to finer resolution. In our framework we allow the human user to operate at the slowest timescale keeping in mind of their cognitive complexity, whereas the swarm operates at the fastest time scale keeping in mind of stability. The interpreter operates at an intermediate time scale communicating with both the human and the swarm. More formally, human and interpreter may interact at discrete times that are a multiple of τ_{h}, the interpreter and the swarm interact at multiples of $\tau_{\mathrm{int}} < \tau_{\mathrm{h}}$, while the swarm operates at $\tau_{\mathrm{s}} < \tau_{\mathrm{int}}$. In what follows, we identify $T \equiv T\tau_{\mathrm{h}} \geq 0$ (resp. $l \equiv l\tau_{\mathrm{int}}$, and $t \equiv t\tau_{\mathrm{s}}$) and we distinguish these integers as belonging to $T \in \mathbb{Z}^{\mathrm{h}}_{\geq 0} \equiv \mathbb{Z}_{\geq 0}$ (resp. $l \in \mathbb{Z}^{\mathrm{int}}_{\geq 0} \equiv \mathbb{Z}_{\geq 0}$, and $t \in \mathbb{Z}^{\mathrm{s}}_{\geq 0} \equiv \mathbb{Z}_{\geq 0}$.) We use the time variable t for the wearable device as it operates at a fast rate, similar to the swarm. Proposed Framework. The user intentions are translated by the interpreter and in turn communicated to the swarm. The user's intention is captured by a wearable device called the MYO armband[1]. By means of it, the user specifies a desired formation shape $S \in \mathbb{P}$, centroid $\mathrm{c} \in \mathbb{R}^2$, orientation $\theta \in \mathbb{R}$, and scaling $\mathrm{s} \in \mathbb{R}$ for the swarm. These parameters make up the desired human intention v which the interpreter decodes as $\hat{v}$, where $v, \hat{v} : \mathbb{Z}^{\mathrm{int}}_{\geq 0} \to \mathbb{P} \times \mathbb{R}^2 \times \mathbb{R} \times \mathbb{R}$. The wearable receives the human intention $v(T)$ as Electromyography (EMG) signals $o^{\mathrm{emg}}(\tau)$ and Inertial Measurement Unit (IMU) signals $o^{\mathrm{imu}}(\tau)$, where $\tau \in [(T-1)\tau_{\mathrm{h}}, T\tau_{\mathrm{h}}]$. The interpreter first uses a decoder (Section 3.4.1) to translate human intentions $v(T)$ into $\hat{v}(T)$. Then it translates the shape $S(T)$ to desired relative agent positions $z^d(T) \in \mathbb{R}^{M \times n}$ which best

[1] https://www.myo.com/

depicts the swarm shape. The swarm also has an operation mode $\mu(t) \in \{1,\ldots,m\}$ corresponding to m different communication ranges for each agent of the swarm. We have the notion of swarm operating cost involving $\mu(t)$ as a trade-off between network connectivity and network maintenance costs. We also introduce shape morphing Dynamics (SMD), which represents easily understandable swarm dynamics by the Human.

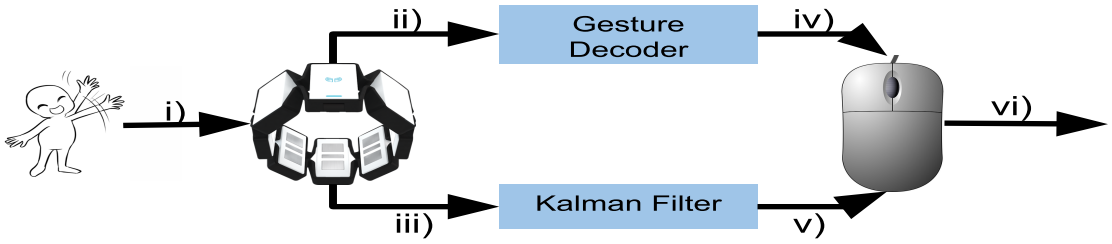


Figure 3.2. User intention decoder system. i) The user conveys their intention $v(t)$. ii) and iii) The wearable captures EMG $o^{\text{emg}}(t)$ and IMU $o^{\text{imu}}(t)$ signals, which are sent to gesture decoder and Kalman filter respectively. iv) The decoder provides gestures which are mapped to mouse clicks and scrolls.v) The Kalman filter is used to assign mouse position. vi) Mouse updates are sent to the GUI.

Now, Given a desired formation $z^d(T)$ and the current state $p(0)$, the interpreter then determines the set of switching intermediate goals $V^s = \{v^s(1),...,v^s(N)\}$ with $v^s(l) = \{z(l), \mathrm{s}(l), \mathrm{c}(l), \theta(l), \mu(l)\}$, $l \in \{1,\ldots,N\}$ and N being the time horizon for switching. These intermediate goals V^s follow the SMD and are optimal with respect to the swarm operating costs. These parameters constitute the high-level commands that the swarm receives and executes via a distributed algorithm. Figure 3.1 illustrates the work-flow of our proposed framework. Thus, we need to solve the following problems to complete our framework:

Problem 1. (Human Intention Decoder). Given the observations $o^{\text{imu}}(t)$ and $o^{\text{emg}}(t)$ from the Myo armband, design a decoder to get the desired human intention $\hat{v}(T)$.

Problem 2. (Behavior Specifier). Given the desired human intention $\hat{v}$, design an algorithm to produce the goal behavior V^s which can be understood by the swarm.

Problem 3. (Planning Algorithm). Given the goal behavior $V^s(T)$, generate the set of optimal intermediate behavior subgoals $\{v^s(l)\}$ with $l \in \{1,\ldots,N\} \cap \mathbb{Z}^{\text{int}}_{\geq 0}$, and N denoting

the time horizon, and $N\tau_{\text{int}} \leq T\tau_{\text{h}}$ which follow Shape-Morphing Dynamics and minimize swarm operating costs.

Problem 4. (Distributed Swarm Controller). Given the command $v^s(l)$, for some $l \in \mathbb{Z}^{\text{int}}_{\geq 0}$, design a distributed algorithm to drive the swarm to the intermediate shape $z(l)$ with scaling $\mathrm{s}(l)$, rotation $\theta(l)$ and centroid $\mathrm{c}(l)$ using operation mode $\mu(l)$ from some initial position $p(l-1)$.

Problem 5. (User Interface Design and Feedback). Develop a Graphical user interface (GUI) for the human to communicate their intention v to the interpreter and receive feedback about the decoded intention $\hat{v}$ and the state of the swarm.

3.4 Technical Approach

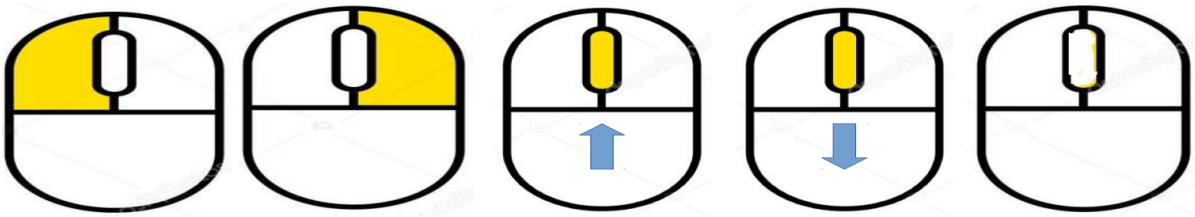
The following subsections describe the proposed solutions to the problems of Section 3.3

3.4.1 Intention Decoding

The user conveys their intention v through gestures and arm movement which are recorded by the Myo armband as EMG signals. There are 8 spatial EMG sensors on the Myo armband which generate EMG signals $o^{\text{emg}}(t) \in \mathbb{R}^8$ at every time t. We only consider the planar angular velocity and orientation signals $o^{\text{imu}}(t) \in \mathbb{R}^4$ at time t from the 9DOF IMU. The intention decoder deciphers discrete gestures $o^{\text{gs}}(t) \in \{0,1,2,3,4\}$ from EMG signals $o^{\text{emg}}(t)$ and state of the arm $\mathbf{arm}(t) \in \mathbb{R}^4$ consisting of planar arm position $\mathbf{arm}^p(t) \in \mathbb{R}^2$ and planar arm velocity $\mathbf{arm}^v(t) \in \mathbb{R}^2$ from IMU signals $o^{\text{imu}}(t)$. The gestures o^{gs} and $\mathbf{arm}$ are translated to mouse movement and mouse clicks, which provide feedback of the decoded intended gesture $\hat{v}$ to the user according to Figure 3.2.

We use HMM, see [Rab89], a common probabilistic machine learning technique to decode gestures from the EMG signals. We use discrete states which are the gestures

(a) Gestures used: (from left to right) Fist, spread, wave up, wave down and normal



(b) Mouse functionalities: (from left to right) Left click, right click, scroll up, scroll down, normal

Figure 3.3. Gestures used for mouse control. a) Show the various gestures used and b) indicate the corresponding mouse functionalities.

$o^{\text{gs}}(t) \in \{0,1,2,3,4\}$ and continuous observations related to EMG signals which are modeled as a multivariate Gaussian distribution. The wearable produces an 8-dimensional spatial EMG signal $o^{\text{emg}}(t) \in \mathbb{R}^8$. We use the mean $\bar{o}^{\text{emg}} \in \mathbb{R}^8$ and standard deviation $\tilde{o}^{\text{emg}} \in \mathbb{R}^8$ of the signals over $1s$ window and $0.2s$ frame shift as input observations giving us $o := (\bar{o}^{\text{emg},\top}, \tilde{o}^{\text{emg},\top})^\top \in \mathbb{R}^{16}$. We collect the training data o for 1 minute, during which the user performs all 5 gestures. The gestures are implemented in a fixed order in a 3 second interval for each gesture without stopping.

To this data we employ the Baum-Welch algorithm to train the HMM model parameters. Details related to the Baum-Welch algorithm implementation can be found in our previous work [Sur16].

We then use the standard forward algorithm to perform live decoding of the gestures similar to our previous work in [Sur16]. We use a standard discrete-time Kalman filter [TBF05] to decode the arm state $\text{arm}(t)$ from the IMU signals $o^{\text{imu}}(t)$.

We use a discrete, linear time-invariant model to describe the dynamics of the

mouse state, $m^p(t)$ and $m^v(t)$, based on Newton's second law:

$$\begin{bmatrix} m^p(t+1) \\ m^v(t+1) \end{bmatrix} = \begin{bmatrix} 1 & \eta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} m^p(t) \\ m^v(t) \end{bmatrix} + \begin{bmatrix} \eta^2/2 \\ \eta \end{bmatrix} m^a(t) + w^p(t), \tag{3.1}$$

where $m^a(t)$ is the input acceleration given by the planar angular orientation of the arm which is under the user's control, η is the update time constant and $w^p(t)$ is the Gaussian process noise. In this way, the acceleration of the mouse pointer is controlled by changing the arm orientation, which is a more stable signal than the one provided by the accelerometer. The measurement model of the KF given by:

$$y^m(t) = r^{\text{arm}} \mathbf{I}_4 o^{\text{imu}}(t) + w^m(t), \tag{3.2}$$

where r^{arm} is the distance between the MYO armband to the tip of the user's finger, which can be measured or fixed approximately and w^m is the Gaussian measurement noise present in the gyroscope and magnetometer signals. Equations (3.1) and (3.2) are in the standard form to apply the KF to estimate the mouse state which is then used by the GUI program to control the mouse movement in the computer. We direct the interested reader to [TBF05] for KF framework and its examples. This enables the armband to potentially substitute for a computer mouse as a complete Human Computer interaction (HCI) device, which can be more practical in VR/AR environments, where a computer mouse could be cumbersome. In a more traditional setting, this also gives the user the opportunity to interact with the computer using both the mouse and the armband. Section 3.5.2 shows the results of our proposed intention decoder. The decoded intentions are sent to the GUI which is illustrated in Section 3.4.2

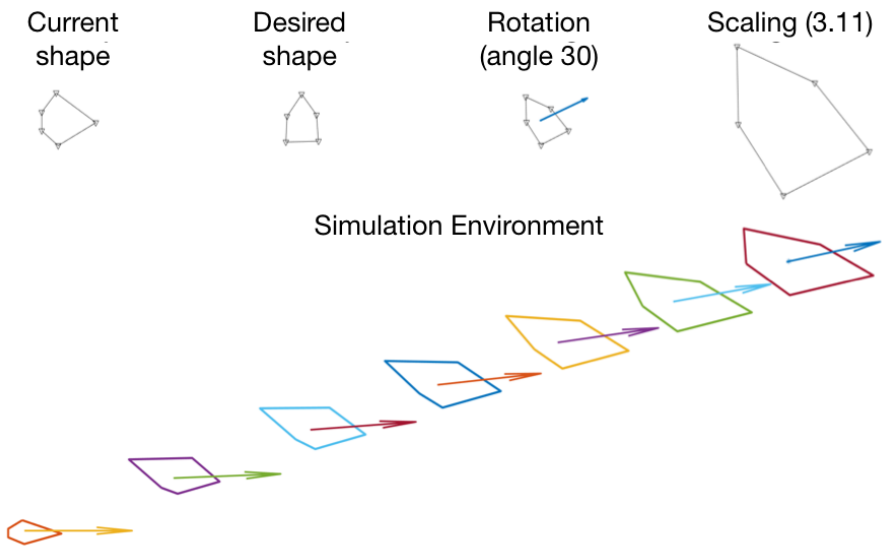


Figure 3.4. UI used to interact with the interpreter.

3.4.2 User Interface Design

We developed a GUI in MATLAB, where the user interacts using arm movements and gestures, which are mapped to mouse movements and mouse clicks according to Figure 3.2. Figure 3.4 illustrates a snapshot of the GUI during the planning phase having 5 different areas to specify the parameters: desired shape S^d, rotation θ^d, scaling s and centroid c^d.

In this manner the user communicates their desired intention which is sent to the interpreter that is described in Section 3.4.5.

3.4.3 Swarm Controller

Our swarm controller is designed to achieve the interpreter's intention $v^s(l) := \{z(l), \mathrm{s}(l), \mathrm{c}(l), \theta(l), \mu(l)\}$ at time $l\tau_{\mathrm{int}}$. Having second-order integrator dynamics for the agents, and the need of controlling the swarm centroid motivates our controller which extends [Cor09] (for first-order agents) with the dynamic consensus feedback interconnection of [ZM10].

With p_i, v_i being the position and velocity of the i^{th} agent, our second-order distributed swarm controller takes the form:

$$
\begin{aligned}
p_i(t+1) =& p_i(t) + v_i(t), && \text{(3.3a)}\\
v_i(t+1) =& -\alpha(p_i(t) + v_i(t)) + \\
& \frac{\alpha}{d_i(t)} \sum_{j \neq i} \{a_{ij}(t)(p_j(t) + v_i(t)) + \\
& \mathrm{s}(l) d_i(t)(z_i(l) - z_j(l)) R^{\theta}(t)\} - k^p(c_i(t+1) - \mathrm{c}(l)),\\
c_i(t+1) =& c_i(t) + \\
& \sum_{j \neq i} w_{ij}(c_j(t) - c_i(t)) + p_i(t) - p_i(t-1), && \text{(3.3b)}
\end{aligned}
$$

where $k^p, \alpha \in (0,1)$ are control gains and R^{θ} is the rotation matrix corresponding to θ. The variable $c_i(t) \in \mathbb{R}^n$ is the estimated center of the swarm by the i^{th} agent. Note that the w_{ij} are the Metropolis weights defined in Section 2. This algorithm, which applies to second-order systems, cancels out the drift observed in [Cor09] with the help of dynamic consensus, and drives the swarm to the desired centroid at time $l\tau_{\text{int}}$. The FODAC algorithm in [ZM10] in equation (3.3b) is used to distributively estimate the mean of time varying reference signal $p(t)$ which would give us the estimate of the swarm's centroid $c(t)$.

Using (3.3) the swarm achieves the desired interpreter's intention $v^s(l)$. After some calculations, with $X(t) = [p(t)^\top, v(t)^\top,\ c(t)^\top, q(t)^\top]^\top \in \mathbb{R}^{4M \times n}$ as the combined

state of the swarm, the state space form of our swarm controller is represented as:

$$X(t+1) = \mathbf{A}X(t) + \mathbf{F}, \tag{3.4}$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{I} & \mathbf{I} & \mathbf{O} & \mathbf{O} \\ -\alpha D_\mu^{-1} L_\mu - k^p \mathbf{I} & -\alpha D_\mu^{-1} L_\mu & -k^p W & k^p \mathbf{I} \\ \mathbf{I} & \mathbf{O} & W_\mu & -\mathbf{I} \\ \mathbf{I} & \mathbf{O} & \mathbf{O} & \mathbf{O} \end{bmatrix},$$

$$\mathbf{F} = [\mathbf{O}^\top,\ [\mathrm{s}\,\alpha D_\mu^{-1} L_\mu z R^\theta + k 1_M \mathrm{c}]^\top\ ,\ \mathbf{O}^\top\ ,\ \mathbf{O}^\top]^\top.$$

Here $q(t) = p(t-1)$ is a dummy state introduced to obtain a linear system in standard form. It is interesting to note that the swarm controller (3.4) consists of an autonomous component $\mathbf{A}$ and a controlled component $\mathbf{F}$ housing the desired interpreter's intention $v^s(l)$. So $v^s(l)$ can be communicated once at the beginning of the l^{th} iteration and the agents just need to adjust their positions and communicate locally with their neighbors to achieve the intermediate goal. Letting $Z^d(l) = [1_M \mathrm{c}(l)]^\top + [\mathrm{s}(l) z(l) R^\theta(l)]^\top$, the desired intention $X^d(l) \in \mathbb{R}^{4M \times n}$ in this state space is given by $X^d(l) = [Z^d(l)^\top, \mathbf{0}^\top, [1_M \mathrm{c}(l)]^\top, Z^d(l)^\top]^\top$. Now we will theoretically analyze the performance of the proposed swarm controller in the next section.

3.4.4 Swarm Controller Analysis

In this section we will analyze our proposed controller (3.4) to determine stability and convergence. We will look at the case when $\mathscr{G}_\mu(t)$ remains constant for $t \in [\tau^l(l-1), \tau^l(l)]$. That is, $\mathscr{G}_\mu(t) = \mathscr{G}_\mu(\tau^l(l-1))$ for $t \in [(l-1)\tau_{\text{int}}, l\tau_{\text{int}}]$. So this makes our system time-invariant in that interval. In this work, we will make use of the following assumptions on $\mathscr{G}_\mu(t)$:

Assumption 1 ((Connectivity)). The communication graph $\mathscr{G}_\mu(t)$ has at least one globally

reachable vertex at every time t.

Assumption 2 ((Constant graphs)). The communication graph $\mathscr{G}_\mu(t)$ remains constant for $t \in [(l-1)\tau_{\text{int}}, l\tau_{\text{int}}]$.

System (3.4) represents n copies of the same dynamics corresponding to n different dimensions. To simplify notation, we will analyze only one of the dimensions. After fixing μ and omitting it for simplicity, our swarm controller (3.4) can be reduced by combining the p and v dynamics to obtain:

$$p(t+1) = (\mathbf{I} - \alpha D^{-1} L)p(t) - k^p 1_M c(t) + \mathbf{F}_1(l), \tag{3.5a}$$

$$c(t+1) = Wc(t) + p(t) - q(t), \tag{3.5b}$$

$$q(t+1) = p(t). \tag{3.5c}$$

where $\mathbf{F}_1(l) = \mathrm{s}(l)\alpha D^{-1} L z(l) R(l) + k^p 1_M \mathrm{c}(l)$. System (3.5) is an interconnected system whose stability depends on the chosen gains α and k^p. We will use the discrete analogue of composite Lyapunov functions [Kha02] to design the gains that guarantee the stability of the interconnected system. With $\delta_1 = 1-(1-\alpha\lambda_2^N)^2$, $\delta_2 = 1-(1-\lambda_2^W)^2$ we can state the following theorem.

Theorem 1. (Stability of Swarm Controller). Under Assumption 1 (connectivity) and Assumption 2 (constant interconnection graph), with the control gains satisfying $k^p < \frac{\delta_1\delta_2}{2}$, the swarm globally uniformly asymptotically stabilizes to the desired state X_d under the swarm controller dynamics (3.4) from any initial condition.

The proof of Theorem 1 is presented in the Appendix. Next we will use the results of Theorem 1 to get an intuition of the role of graph connectivity (λ_2^N and λ_2^W) in the convergence of our swarm controller (3.4).

Corollary 1. The convergence rate of (3.4) is directly proportional to λ_2^N and λ_2^W of the communication graph.

The proof of Corollary 1 can be easily verified by looking at the convergence rate of the system in proof of Theorem 1. Using these results we will design a planning algorithm, which optimally determines the intermediate subgoals which will be described in Section 3.4.5.

Stability under time delays:

Next we will give some insight into our controller's performance under communication time delays. Let us consider our system (3.5) subjected to a communication delay $\tau_d \in \{0, 1, ..., \overline{\tau}\}$, where $\overline{\tau} < \infty$ is the maximum delay possible. Then we have :

$$p(t+1) = (\mathbf{I} - \alpha D^{-1} L) p(t - \tau_d) - k^p 1_M c(t) + \mathbf{F}_1(l), \tag{3.6a}$$

$$c(t+1) = W c(t - \tau_d) + p(t) - q(t), \tag{3.6b}$$

$$q(t+1) = p(t). \tag{3.6c}$$

Remark 1. (Robustness to time delay): We see that the interconnections in (3.6) are without delays as they are independent of the communication network and each agent updates them locally. As τ_d is bounded and there are no delays in interconnections, we can easily apply results from [XW08, RB05] to show consensus against time-varying bounded delays for each sub-system. Then we can analyze stability of (3.6) similar to Theorem 1 by considering each subsystem individually and then checking stability for interconnections. Hence, in the interest of space we skip rigorous analysis.

Now we will move on to the Interpreter, which is the next aspect of our framework.

3.4.5 The Interpreter

In this section, we describe the role of the interpreter in the framework. For ease of illustration, we consider the formulation in 2D space. The interpreter mainly consists

of two parts: the behavior specifier and the high level planner, which are illustrated in the following paragraphs. Problem 2: Behavior Specifier The Behavior specifier converts the desired human intention into parameters that can be comprehended by the swarm. The human user specifies the desired shape $S^d \in \mathbb{P}$ which takes the form of an arbitrary polygon, the desired centroid $\mathrm{c}^d \in \mathbb{R}^2$, scaling $\mathrm{s}^d \in \mathbb{R}$ and rotation $\theta^d \in \mathbb{R}$. The interpreter then decides the formation denoted by the relative positions of the agent $z^d \in \mathbb{R}^{M \times n}$, which would best illustrate the shape S^d given by the human. For simplicity, we use a uniform distribution in the interior of the shape S^d to obtain z^d, which is illustrated in Figure 3.5b. The human specifies the polygon by providing the vertices sequentially using the GUI from Section 3.4.2, which is shown on the left side of the Figure 3.5b. The corresponding formation density $\rho^M = M/\operatorname{area}(S^d)$ is calculated, where $\operatorname{area}(S^d)$ is the area of polygon S^d. We assume the density is large enough to fit M robots in the shape S^d. Note that, since the shape S^d is bounded, there exists a large enough box B such that $S^d \subseteq B$ and $M\frac{\operatorname{area}(B)}{\operatorname{area}(S)}$ is equal to a perfect square r^2, for some $r^2 \geq M$ and $r^2/\operatorname{area}(B) = M/\operatorname{area}(S) = \rho^M$. Using this density, robots are distributed uniformly in the bounding box B of the polygon S^d by creating a meshgrid. Finally, we discard the generated points not in the polygon and we arrive at the desired formation z^d of M points shown in the right half of Figure 3.5b.

The parameters z^d, S^d, c^d, s^d and θ^d are passed on to the Planner, which is described next.

Planner

While the planner does not deal with detailed, low-level motion plans for each robot, high-level (optimal) laws are provided for an abstraction of the system, and then used as a starting point for implementation later. At the lower layer, robots rely on distributed algorithms that aim to realize the output for the planner.

The Planner receives the decoded human intention in the form of desired forma-

tion S^d (or, equivalently, z^d), scaling s^d, rotation θ^d, and centroid c^d. The planner then constructs a set of intermediate way points $\{S(l), \mathrm{s}(l), \theta(l), \mathrm{c}(l)\}$, $\forall l \in \{1, \ldots, N\}$, where N denotes the number of intermediate steps in the plan to reach the final goal.

To do this, we employ an N-Horizon Discrete Switched Linear Quadratic Regulator (DSLQR) formulation. A particular DSLQR problem with a dynamical variable $h \in \mathbb{R}^d$ and time horizon $l \in \{1, \ldots, N\}$ can be formulated as follows:

$$\min J(u, \mu) = \sum_{l=0}^{N} (h(l)^\top Q_\mu h(l) + u(l)^\top R_\mu u(l)) + h(N)^\top Q_f h(N), \tag{3.7a}$$

$$\text{subject to } h(l+1) = \mathscr{A} h(l) + \mathscr{B} u(l), \tag{3.7b}$$

where $h(0) = h_0$. Here, the running cost consist of a switching LQ cost function, with parameterized matrices Q_μ and R_μ, depending on a mode μ. The function will be designed to enhance swarm performance while the linear constraint will be used to enforce an easy-to-interpret behavior by a human, which defines a shape morphing (SMD) dynamics.

Details and methodology of DSLQR systems can be found in [ZHA09]. We show next how we apply this approach in our particular setup and describe the matrices that we choose for our framework.

(i) Shape-Morphing Dynamics: We introduce the notion of shape morphing Dynamics (SMD) to denote a dynamical system that can be used to morph into any desired shape. Since the interpreter needs to provide feedback to the user, the planner needs to provide an abstraction of the complicated swarm dynamics in an Mn-dimensional space. These dynamics need to be slower than the swarm dynamics to enhance interpretability, and are hence implemented in the l timescale described in Section 3.3.

Here, we propose a simple linear dynamical system approach to model these dynamics, which takes into account the desired human intention $h^d = (S^d, \mathrm{s}^d, \theta^d, \mathrm{c})$.

We let $h = [S, \mathrm{s}, \theta, \mathrm{c}]^\top$ denote the state of the SMD system with $h(l) \in \mathbb{H}$, where $\mathbb{H} = \mathbb{P} \times \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}$. Then, the SMD takes the form:

$$h(l+1) = \mathscr{A} h(l) + \mathscr{B} u(l), \tag{3.8}$$

where matrices $\mathscr{A}, \mathscr{B} \in \mathbb{H} \times \mathbb{H}$ and control input $u \in \mathbb{H}$.

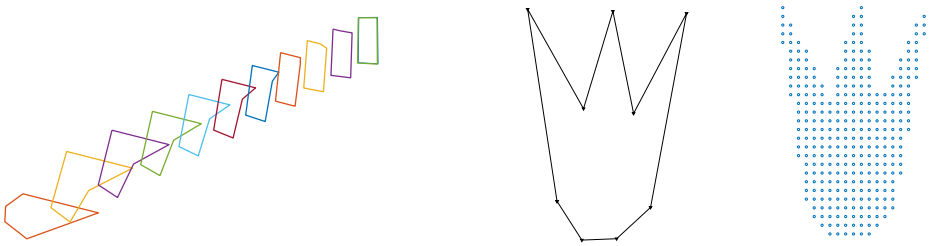
We use the N horizon Discrete LQR control technique to drive the SMD towards h^d starting from some initial configuration $h(0) = h_0$. By considering a change of variable $h^e(l) = h(l) - h^d$, we define a first term contributing to the problem cost functional as follows:

$$J_{\mathrm{SMD}}(u) = \sum_{l=0}^{N-1} (h^{e\top}(l) Q h^e(l) + u(l)^\top R u(l)) + h^e(N)^\top Q_f h^e(N). \tag{3.9}$$

where the matrices $Q, R, Q_f \in \mathbb{H} \times \mathbb{H}$ are positive definite and $u(l)$, $\forall l \in \{1, \ldots, N\}$ is a step change applied during the l^{th} time. So $u(l)$ is chosen such that the cost J_{SMD} is minimized. This is solved using the standard LQR approach, and the results are shown in Figure 3.5 for a $N = 10$ horizon problem. Intuitively, one can choose these matrices to satisfy $Q \prec R \prec Q_f$ to provide a more "interpretable" and natural transition dynamics. This condition implies that the priority is to reach the desired behavior h^d with small changes in the intermediate steps, which would make it look more natural and "interpretable" as seen in Figure 3.5. Further discussions on the choice of parameters and their implications are done in Section 3.5.3. Figure 3.5 shows the stages of transformation of a 5 sided polygon to a rotated and translated 4 sided polygon. The figure depicts a seemingly natural transition which can be easily interpreted by the user. The case of mismatch in the number of vertices in the initial and desired shapes is handled by

adding vertices appropriately on the perimeter of the shape that has fewer vertices.

(ii) Swarm performance costs. We just discussed how to generate intermediate shapes taking into account the SMD. Now we consider the swarm performance and communication cost to choose the operating mode ν in the general setup. The operating modes ν correspond to a subset of ν-disk graphs defined over the swarm when distributed over a shape. Since agent formations are chosen in a consistent manner as described in e.g. Figure 3.5b, the number of possible graphs over the agents for different ν is very much reduced and remains constant for scaled shapes. From now on, we consider this set is given by $\{\nu_1, \ldots, \nu_m\}$ by choosing appropriate communication radii. The proposed planner aims to provide high-level optimal plans for the swarm without taking each individual agent into consideration. This philosophy is consistent with a multi-level optimization approach to complex systems, by which high-level (optimal) laws are provided for an abstraction of the system, and then used as a starting point to a more detailed optimization. At the lower layer, robots can employ more involved controls as well as distributed algorithms, which are more robust to single robot failures. In this way, we are able to use abstraction for the intermediate interface to communicate with user, while not giving up robustness from distributed algorithms at the level of individual robots.



(a) SMD illustration (b) Formation Specifier

Figure 3.5. (a) SMD illustration for shape changing from rotated cone to a standing rectangle. The model parameters used are $\mathscr{A} = \mathscr{B} = Q = \mathbf{I}_h$, $R = 100\mathbf{I}_h$ and $Q_f = 1500\mathbf{I}_h$. (b) Left: The user specifies the desired shape S^d by providing v vertices (triangles). Right: the interpreter determines the relative positions z^d of $M = 500$ agents (blue dots) to represent the shape drawn by user.

Operating costs involved: To increase the speed of convergence and to facilitate quicker interpretation, we need to maximize the notion of connectivity involving the second smallest eigenvalue λ_2^N or λ_2^W of the respective Laplacian matrices L^N and L^W. This can be found from the determinant of the matrix $G \in \mathbb{R}^{(M-1)\times(M-1)}$ defined as $G = F^\top L^N F$ with $F \in \mathbb{R}^{M\times(M-1)}$, $F1_M = 0$ and $F^\top F = \mathbf{I}$. Since the determinant of a matrix is a product of its eigenvalues, connectivity determined by λ_2^N increases iff the determinant of G increases. So the connectivity cost $J_{\mathrm{CON}}(l)$ being in formation z and operation mode ν at time l is given by:

$$J_{\mathrm{CON}}(\nu, h) = -\kappa_1 \log\det(\kappa_2 G_\nu(l)). \tag{3.10}$$

To ensure J_{CON} remains well scaled and positive we introduce positive constants κ_1 and κ_2 respectively. Having a ν corresponding to a higher communication radius implies that we will be using more energy to communicate and maintain communication links. This is encoded as a communication cost $J_{com}(l)$ being in formation z and operation mode ν at time l. It is given by

$$J_{\mathrm{COM}}(\nu, h) = \kappa_3 \log(\nu_\nu^2 1_M^\top A_\nu(h) 1_M), \tag{3.11}$$

where $\nu(l)$ is the communication range at time l and κ_3 is a positive constant used for scaling.

Adding these costs together defines the total cost used by the planner as:

$$\begin{aligned} J(u,\nu,h) =& J_{\mathrm{SMD}}(u) + J_{\mathrm{CON}}(\nu,h) + J_{\mathrm{COM}}(\nu,h) \qquad (3.12)\\ J(u,\nu,h) =& \sum_{l=0}^{N-1} (\bar{h}^e(l)^\top \overline{Q}_{\nu(l)} \bar{h}^e(l) + \overline{u}(l)^\top \overline{R}\overline{u}(l) + \\ & \bar{h}^e(N)^\top Q_f \bar{h}^e(N). \end{aligned}$$

where $\overline{Q}_{\nu} = \begin{bmatrix} Q & 0 \\ 0 & J_{\mathrm{CON}}(\nu) + J_{\mathrm{COM}}(\nu) \end{bmatrix}$, $\bar{h}^e = \begin{bmatrix} h^e \\ 1 \end{bmatrix}$, $\overline{u} = \begin{bmatrix} u \\ 0 \end{bmatrix}$ and $\overline{R} = \begin{bmatrix} R & 0 \\ 0 & 1 \end{bmatrix}$.

Optimization Approach

Observe that a solution to the above problem (3.12) requires the evaluation of all possible graph combinations for different chosen controls u and starting configurations h. By choosing the graphs based on the communication radii, and considering a class of formations, we reduce significantly the number of possible graphs to evaluate. Also note that the variables ν belong to a discrete set, while the controls u are continuous, and also the cost function J is non-linear. This makes the resulting problem nonconvex and NP-hard.

Our approach to this problem is based on optimal control and the LQR method. The approach is summarized in Algorithm 2. Initially, we remove the dependency on h from (3.12) by an finding an optimal set of initial configurations $\{\tilde{h}(i)\ i \in 0,...,N\}$ using the LQR solution approach to solve J_{SMD}. This gives us $\{\tilde{u}(i)\ i \in 0,...,N\}$ from which we can construct $\{\tilde{h}\}$. Using the constructed $\{\tilde{h}\}$, we can initially remove the dependency of h in (3.12). After this, we employ the DSLQR formulation from [ZHA09] to obtain the optimal set of $u(l)$ and $\nu(l)$ which minimizes J with respect to the optimal initialization of h. Note that, by rederiving $u(l)$, we are obtaining a new sequence of h.

Algorithm 2: Planner

Input: $A, B, Q, R, \{v\}, h(0), h(N), J(u, v, h)$;

Output: $\{u(l)\}, \{v(l)\}, \{h(l+1)\}\ \forall l \in \{0, 1, .., N-1\}$;

for $l \in \{0, 1, .., N-1\}$ do

    $\tilde{u}(l) = LQR(\mathscr{A}, \mathscr{B}, Q, R, Q_f)$;

    $\tilde{h}(l+1) = SMD(\mathscr{A}, \mathscr{B}, \tilde{u}(l), \tilde{h}(l))$;

end

$J(u, v) \leftarrow J(u, v; \tilde{h})$;

for $l \in \{0, 1, .., N-1\}$ do

    $u(l), v(l) = DSLQR(\mathscr{A}, \mathscr{B}, Q, R, Q_f, J(u, v), \{v\})$;

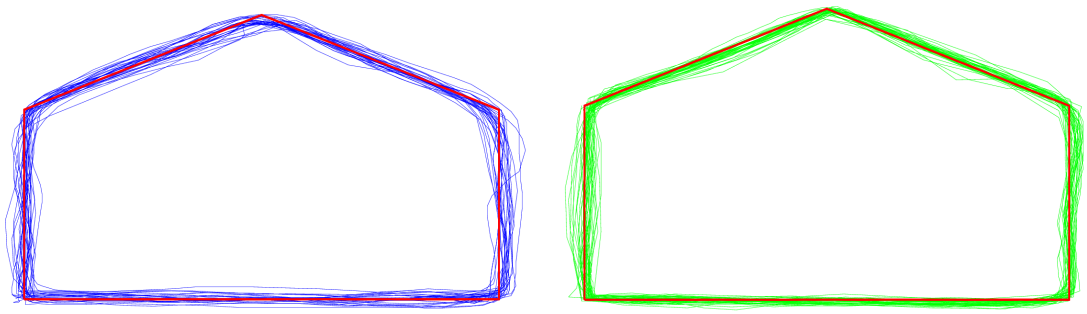
    $h(l+1) = SMD(\mathscr{A}, \mathscr{B}, u(l), h(l))$;

end

3.5 Implementation Results and Discussions

3.5.1 System Setup

The user can choose between MYO armband and mouse to interact with a GUI to control the formation of a simulated swarm in a two dimensional environment. The swarm controller developed in Section 3.4.3 essentially generates waypoints for the swarm to follow, we assume holonomic dynamics for the individual agents and assume they reach their respective waypoints. We do not focus on collision avoidance, which will we addressed in future work. We utilize the ROS kinetic framework with Python scripting language to interface with the MYO armband and control the mouse pointer. We use Matlab to create the GUI shown in Figure 3.4, which uses the mouse or the MYO armband as an input device. For the formation controller we set the control gain $\alpha = 0.15$ and proportional constant $k^p = 0.03$.



(a) Mouse movement with wearable (b) Mouse movement without wearable

Figure 3.6. Aggregate results of tracing a pentagon (Red). a) The user specifies the shape by using MYO armband. (Blue) b) The user specifies the shape by using the mouse (Green).

3.5.2 Preliminary Results on Intention Decoding

We performed tests to gauge the accuracy and speed of the proposed intention decoder. For the HMM model, some of our previous tests had given an accuracy levels of over 90% on an average [Sur16] for similar gestures and framework. On preliminary tests we observed similar results and hence, in the interest of space, we skip the accuracy test for the HMM model. For the effectiveness of the arm movement decoder, we compare the results of operating a mouse with and without the MYO armband. We use a single user and perform 5 trials. The user was given one minute and was tasked to continuously trace a pentagon (height= 6 units, width= 2 units and area = 10 units) which represents the human intention. Figure 3.6 represents the aggregate results over 5 trials. These trials were done in a Matlab environment, while the user observed the movement of mouse pointer along the polygon as feedback. It can be seen from the Figure 3.6 that the results are similar for both cases. Table 3.1 describes the absolute error involved in each of the trials. The total error is calculated by summing the shortest distance from each traced point to the polygon. The Average error is total error divided by the number of points traced. The speed is tested by the number of complete loops the user is able to make in the given time which is one minute. It can be seen that the errors involved are about the same with both interfaces, however the speed of using the mouse is higher

which is expected as user need time to adapt to the new interface.

Table 3.1. Absolute Error comparison of mouse and wearable.

	Mouse			Wearable		
Sl. no	Loops	Avg Error	Total Error	Loops	Avg Error	Total Error
1	7	0.026	122.57	5	0.038	179.26
2	8	0.028	129.40	5	0.037	174.74
3	9	0.031	147.02	7	0.048	222.07
4	8	0.031	148.92	7	0.05	235.27
5	9	0.035	161.50	5	0.029	132.72

Table 3.2. Percentage of points within a given buffer.

	Mouse(%)			Wearable(%)		
Sl. no	**0.05d**	**0.1d**	**0.2d**	**0.05d**	**0.1d**	**0.2d**
1	79.25	93.56	99.80	79.24	93.56	99.80
2	86.66	97.71	100.00	75.13	92.73	100.00
3	85.66	98.46	100.00	77.05	92.3 0	100.00
4	79.86	95.09	100.00	64.68	90.31	98.71
5	81.37	94.56	100.00	85.36	97.66	100.00

We also compared accuracy levels by computing percentage error which is illustrated in Table 3.2. We created a buffer around the perimeter of the polygon, and calculated the percentage of points within the buffer as compared to the total number of points traced in that particular trial. A traced point is within the buffer if its Euclidean distance to the polygon is less than the buffer units. These buffers where not shown to the user and were used for post computation only. We created three different buffers of sizes 0.05, 0.1, 0.2 units, which are indicated as the columns $0.05d$, $0.1d$, $0.2d$ of

Table 3.2 respectively. We can see that almost all ($\approx 100\%$) points traced by using both the mouse and the wearable is within a distance of 0.2 units. The performance within 0.1 units is also similar for both devices in terms of precision with all trials being $> 90\%$ accurate. In the sub 0.05 units category we see a decline in accuracy for both cases. In this category the mouse performs better than the myo armband on average, but we see that the peak accuracy levels are still comparable (85.36% for wearable v/s 86.66% for mouse).'

The preliminary results shown in Table 3.1 and Table 3.2 indicate that the performance of the wearable is quite close to the mouse, which is encouraging in AR/VR applications where the usage of mouse is more cumbersome compared to the wearable.

3.5.3 Shape Morphing Dynamics

In this section we illustrate our shape morphing dynamics formulation (3.8) and discuss the implications of varying the various parameters involved in the dynamics.
Q and R matrices. We know from LQR theory that higher the values in the Q matrix, the shape converges to the desired shape in fewer iterations. The matrix R however provides a tradeoff, by penalizing the change between consecutive shapes. These aspects are illustrated in Figure 3.7

(a) $r = 10^{-2}$ (b) $r = 10^{0}$ (c) $r = 10^{2}$ (d) $r = 10^{4}$

Figure 3.7. Results of varying the $R = r\mathbf{I}$ for SMD while keeping $Q = \mathbf{I}$ constant for $N = 10$ horizons.

$\mathscr{A}$ and $\mathscr{B}$ matrices. From standard notions of discrete linear control theory, we know that the eigenvalues of $\mathscr{A}$ should be within the unit circle to ensure stability

and the $\mathscr{B}$ matrix translates the control input u to the system dynamics. Figure 3.8 illustrates these concepts on our system.

(a) $r = 10^{-2}$ (b) $r = 10^{0}$ (c) $r = 10^{2}$ (d) $r = 10^{4}$

Figure 3.8. (a)-(b) Effect on SMD by varying $\mathscr{A}$ keeping $\mathscr{B} = \mathbf{I}, Q = \mathbf{I}, R = 10^{3}\mathbf{I}$ constant. (c)-(d) Effect on SMD by varying $\mathscr{B}$ keeping $\mathscr{A} = Q = R = \mathbf{I}$ constant for $N = 10$ horizons.

Now we will validate the proposed framework by running simulations of a swarm of 50 agents to reach the desired human intention shown in Figure 3.9.

Figure 3.9(a)-(d) indicate the desired human intention. Using $\mathscr{A} = \mathscr{B} = Q = \mathbf{I}_h$, $R = 100\mathbf{I}_h$, $Q_f = 1500\mathbf{I}_h$, $\kappa_1 = 10^6$, $\kappa_2 = 0.05$, $\kappa_3 = 2 \times 10^4$ the planner was implemented for a $N = 8$ horizon problem with $m = 3$ subsystems. The communication ranges are $\nu(l) \in \{10, 40, 150\}$, corresponding to the three operating modes. Figure 3.9e illustrates the intermediate shapes resulting from the 8 horizon planner, starting from the current intention (triangle on the left), to the desired intention (larger rotated quadrilateral) on the right. Figure 3.9f describes the evolution of the cost (3.12) and switching strategy in a backward horizon. The costs 1,2 and 3 correspond to the cost incurred by the system if they exclusively run on the respective operating modes $\nu = 10, \nu = 40$ or $\nu = 150$ throughout the execution. The optimal cost is shown in purple is calculated by solving (3.12) by using Algorithm 2.

Figure 3.9f shows that switching occurs in a timely manner to maintain minimum costs. In this way, switching occurs from 1^{st} mode to the 2^{nd} mode during the 2^{nd} timestep. During the 7^{th} timestep another switching occurs to the 3^{rd} operating mode to maintain minimum cost. This is coherent with the intuition of using larger communication radii for more sparse swarms. As the scaling increases with every timestep, the

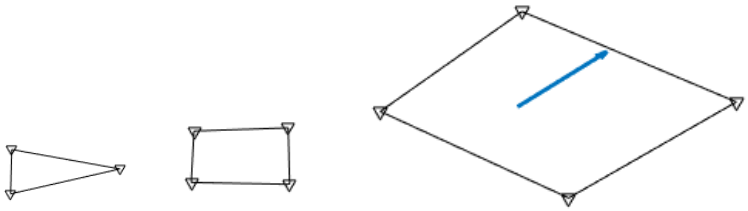
agents are forced further apart and the cost of using a smaller communication range $\nu = 10$ rapidly increases. Whereas, the cost of using the largest range $\nu = 150$ remains almost constant throughout because the connectivity and communication costs mostly remain the same. Figure 3.9e shows the execution of the swarm controller during the $l = 2$ horizon. Each of the red dots represent individual agents of the swarm. We evaluate the performance of the swarm controller (3.4) by measuring the error with respect to the intermediate formations and centroid at each time step t. The formation error and centroid error are measured as $e_l^f(t) = \|p(t) - s(l)z(l)R(\theta(l))\|$ and $e_l^c(t) = \|c(t) - c^d(l)\|$ respectively in reaching the l^{th} intermediate goal. The evolution of these errors(y-axis) with respect to time t(x-axis) is illustrated in Figures 3.9g and 3.9h. We see that the swarm successfully reaches every intermediate goal and finally reaches the desired human intention.

We also tested the DSLQR approach with different random initialization against the proposed LQR initialization. We found that the resulting cost from LQR initialization is far lesser (more than two times) than the random initializations in every horizon for over 25 trials with 5 horizons. This justifies the usage of LQR initialization for the DSLQR algorithm.

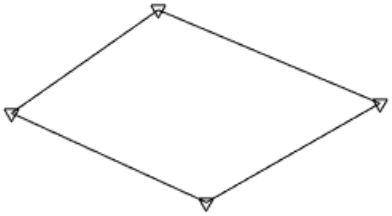
3.6 Summary

In this chapter we have proposed and successfully implemented a novel HSI framework for formation control using the concept of an interpreter. Using this framework the user communicates their intentions by drawing shapes, and the swarm successfully depicts the drawn shape with the aid of the interpreter. The effectiveness of this framework has been illustrated through theoretical stability analysis and simulations.

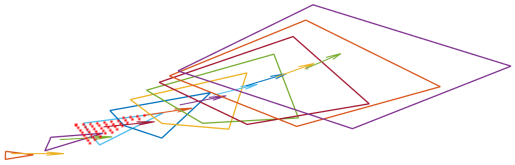
In the next chapters, we focus on designing behavioral perception based robot navigation algorithms in risky and uncertain environments.



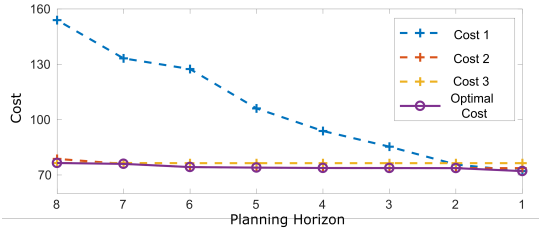
(a) Current shape (b) Desired shape (c) Desired rotation: $\theta^d = 50°$



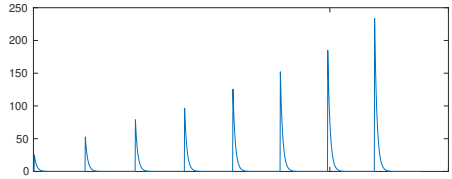
(d) Desired scaling: $s^d = 11.6$



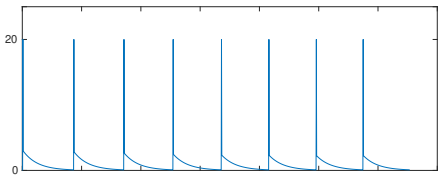
(e) Planning and Execution



(f) Switching cost throughout execution



(g) Formation error $e^f(t)$



(h) Centroid error $e^c(t)$

Figure 3.9. Results of executing a particular desired behavior communicated by the human.

This chapter, in full, is a reprint of the material as it appears in the publication Human-swarm interactions for formation control using interpreters, A. Suresh and S. Martínez, International Journal of Control, Automation and Systems, 18, pp. 2131–2144, 2020. A preliminary version of the work appeared in the proceedings of the IFAC Cyber-Physical-Human Systems conference, Miami, USA, pp. 83–88, December 2018, as Gesture-based human-swarm interactions for formation control using interpreters, A. Suresh and S. Martínez. The dissertation author was the primary investigator and author of these papers.

Chapter 4

Risk-perception-Aware Path Planning

This chapter investigates the design of risk-perception-aware motion-planning strategies that incorporate non-rational perception of risks associated with uncertain spatial costs. Our proposed method employs the Cumulative Prospect Theory (CPT) to generate a perceived risk map over a given environment. CPT-like perceived risks and path-length metrics are then combined to define a cost function that is compliant with the requirements of asymptotic optimality of sampling-based motion planners (RRT*). The modeling power of CPT is illustrated in theory and in simulation, along with a comparison to other risk perception models like Conditional Value at Risk (CVaR). Theoretically, we define a notion of expressiveness for a risk perception model and show that CPT's is higher than that of CVaR and expected risk. We then show that this expressiveness translates to our path planning setting, where we observe that a planner equipped with CPT together with a simultaneous perturbation stochastic approximation (SPSA) method can better approximate arbitrary paths in an environment. Additionally, we show in simulation that our planner captures a rich set of meaningful paths, representative of different risk perceptions in a custom environment. We then compare the performance of our planner with T-RRT* (a planner for continuous cost spaces) and Risk-RRT* (a risk-aware planner for dynamic human obstacles) through simulations in cluttered and dynamic environments respectively, showing the advantage of our

proposed planner.

4.1 Literature review

Traditional risk-aware path planning considers risk in the form of dynamic uncertainty [KBP12], collision time [SGW19], or sensing uncertainty [BB07]. Chance constrained approaches [LKH13, BOW11] are used to handle agent and environment uncertainty in a robust manner, however discrete polyhedral obstacles are considered which cannot incorporate continuous spatial costs. Stochastic dynamic programming [TB12] in used in dynamic environments to locally integrate planning and estimation without optimality guarantees. Moreover, in all the above works, how the risks and uncertainties are perceived or relatively weighted has been overlooked. A few recent works [SCMP19] contemplate risk perception models, but assume rational DMs and use coherent risk measures like Conditional Value at Risk (CVaR) [HKY19]. Unlike CPT's suggestions, these measures are built using certain axioms that assume rationality and linearity of the DM's risk perception [ADEH99]. CPT has been extensively used in engineering applications like traffic routing [GFBA10], network protection [HS19], stochastic optimization [JAF+18], and safe shipping [WLY18] to model non-rational decision making. However, CPT is yet to be applied in robotic planning and control.

Regarding planning algorithms themselves, RRT* [KF11] has been the basis for many motion planners due to its asymptotic optimality properties and its ability to solve complex problems [BHM18]. Risk [CM17] and uncertainty [ESBS16] have been an ingredient of motion planning problems involving a human, but have been mainly modeled in a probabilistic manner [SLVP18] with discrete obstacles. Very few of these works have considered modeling planning environments via continuous cost maps [DSC16, SE15], while, to the best of our knowledge, the simultaneous treatment of cost and uncertainty perception to model a DM's spatial risk profile has largely been ignored.

4.2 Contributions

Our contributions lie in three main areas: Firstly, we adapt CPT into path planning to model non-rational perception of spatial cost embedded in an environment. With this, we can capture a larger variety of risk perception models, extending the existing literature.

Secondly, we generate desirable paths using a sampling-based (RRT*-based) planning algorithm on the perceived risky environment. Our planner integrates a continuous risk profile and path length to calculate path cost, enabling us to plan in the perceived environment setting above.

Furthermore, the chosen cost satisfies the sufficient conditions for asymptotic optimality of the planner, leading to reliable and consistent paths according to a specified risk profile. We then compare our planner's performance with T-RRT* (continuous cost space planner) and Risk-RRT* (risk-aware planner) through simulations in cluttered and dynamic environments respectively. We show that our proposed planner can generate better paths in comparison.

Finally, we define the notion of "expressiveness" for a risk perception model and show that CPT's is higher than that of CVaR and expected risk. Furthermore using SPSA, we show that the expressiveness hierarchy translates to our path planning setting, where we observe that a planner equipped with CPT can better approximate arbitrary paths in an environment.

We clarify that here, we merely examine CPT based environment perception models for motion planning and leave the validation of these models with human user studies for future work.

4.3 Environment Setup

In this work, we consider spatial sources of risk embedded in the $\mathscr{C}$ space. Our starting point is an uncertain cost function $\rho(x)$ that aims to quantify objectively the (negative) consequences of being at a location x or adopting a certain decision under uncertainty at a point x.

For example, suppose a robot moves to a location x from x' where there is an obstacle with certain probability. Then, we can define a cost measurement as the possible damage to the robot by moving from x' to x under action a' applied at x'. A cost value $\rho(x',x,a')$ can be defined depending on i) the type of robot (flexible robot or rigid robot), ii) the probability of having an obstacle in the said location, and iii) the type of action applied at x' to get to x (e.g. slow/fast velocity). For simplicity, adopting a worst-case scenario, we may reduce the previous cost function to a function of the state $\rho(x) \equiv \max_{x',a'} \rho(x',x,a')$[1].

As another example, consider a drone navigating in a building which is ablaze. In this case, the cost function can be proportional to the temperature profile. As sensors are noisy, the temperature profile is uncertain, resulting into a noisy spatial cost value $\rho(x)$. Similarly, environmental conditions that affect the robot's motion may lead to under performance. When moving over an icy road, the dynamics of the robot may behave unpredictably, resulting in a temporary loss of control and departure from an intended goal state. In this case, the uncertain cost may be quantified as the state disturbance under a given action over a given period of time. For example, for a simple second-order and fully actuated vehicle dynamics with acceleration input a which is subject to a locally constant "ice" disturbance $d(x) \approx d$, in a small neighborhood of x, we have $(x' - x) = (a + d(x))\Delta t^2/2$ for a small time Δt. Thus, the difference with an intended state can be measured by the random variable $\rho(x) = \frac{\|d(x)\|\Delta t^2}{2}$, which encodes

[1] Instead of a max operation, one may use an expected operation wrt x', a'.

information about $d(x)$ and the unit time of actuation, Δt. Here $d(x)$ is uncertain, and can be modeled with prior data or measured with a noisy sensor as in the temperature profile case.

Prior knowledge in the form of expert inputs and data collected from sensors can be used to get information about the cost $\rho(x)$, environmental uncertainty, and the robot's capabilities. In this way, icy roads pose much lesser cost in the previous sense to a 4WD car with snow tires than a 2WD car with summer tires, hence the same cost at a given location could be scaled differently depending upon the robot's capabilities.

In this work, we will assume that the cost at a location $x \in \mathscr{C}$ has been characterized as a random variable $\rho(x)$ with a mean $\rho_\mu(x) \in \mathbb{R}_{\geq 0}$ and standard deviation $\rho_\sigma(x) \in \mathbb{R}_{\geq 0}$, for each $x \in \mathscr{C}$. We remark that this cost can be constructed from diverse criteria: From nature of location (For eg. operating table at hospital being mostly static but highly risky) to dynamic properties (For eg. velocity, state) of close-by obstacles. In particular, it is reasonable to approximate $\rho_\mu(x)$ via a "bump function," a concept extensively used in differential geometry. To fix ideas, consider the previous case where a vehicle moves through an "icy" environment, and assume $\Delta t = 1$. Then, a mean disturbance over a subset $A \subseteq \mathscr{C}$ should result approximately into a disturbance $\frac{\|d\|}{2}\ell = \rho^{\max}\ell$, where ℓ is the portion of the trajectory from x to x' that is inside the icy section A. As x is farther from x', the disturbance reduces its effect on x, and the value of ℓ should decrease to zero. In other words, there is a B such that $A \subseteq B$, where B is an enlarged region whose boundary delimits the uncertain cost area from the certain one (i.e. outside B the cost is zero with low uncertainty). The effect of ℓ is thus similar to that of a bump function defined with respect to A and B. Bump functions are infinitely smooth, take a positive constant value over A, which smoothly decreases and becomes zero outside B. There are many ways of defining bump functions on manifolds, such as via convolutions, which works in arbitrary dimensions as described the following. Let $\chi_D : \mathscr{C} \to \mathbb{R}$ denote the

indicator function of a subset $D \subseteq \mathscr{C}$, and, given A, define $f(x) = \frac{\rho^{\max}}{C} \exp(-\frac{1}{1-\|x\|^2})\chi_A(x)$, with $\int_A \exp(-\frac{1}{1-\|x\|^2}) = C$. Then, a bump function based on A and B can be given by the convolution $b(x) = \chi_B \star f(x)$, $x \in \mathscr{C}$. This function takes a value of 0 outside B, $\rho^{\max}$ inside A and a value between 0 and $\rho^{\max}$ at the points $x \in B \setminus A$. See Section 4.8 for an alternative choice of bump function.

In this work, the notions of "risk" and "risk perception" relate to the way in which the values of $\rho(x)$ are scaled and averaged in expectation. That is, risk is a moment of a given uncertain function (either $\rho(x)$ or a composition with ρ). For example, the risk of being at a location x can be measured via expected cost; that is $R^e(x) = \mathbb{E}(\rho(x))$, which may represent "expected damage to robot" with respect to uncertainty. However, there are other ways of weighting the $\rho(x)$ outcomes to define alternative risk functions, such as using CPT. With this is mind, we proceed to define the following three main problems and address them in Section 4.5, Section 4.6 and Section 4.7, respectively.

4.4 Problem Statement

Here we list the set of problems we address in this chapter:

Problem 6. (CPT environment generator). Given the configuration space $\mathscr{C}$ containing the uncertain cost ρ along with the DM's CPT parameters Θ, obtain a DM's (non-rational) perceived risk R^c consistent with CPT theory.

Problem 7. (Planning with perceived risk). Given a start and goal points x_s and x_g, compute a desirable path P from x_s to x_g in accordance with the DM's perceived risk R^c.

Problem 8. (CPT planner evaluation). Given a configuration space $\mathscr{C}$, and an uncertain cost ρ along with a drawn path P_d, evaluate the CPT planner as a model approximator to generate the perceived risk R^c representing the path P_d.

4.5 Risk perception using CPT

Here, we will generate a DM's perceived risk and address Problem 1. We consider an uncertain cost $\rho(x)$ is given at every point $x \in \mathscr{C}$, which we approximate via its first two moments, a mean value $\rho_\mu(x) \in \mathbb{R}_{\geq 0}$ and a standard deviation $\rho_\sigma(x) \in \mathbb{R}_{\geq 0}$. In what follows, we use a discrete approximation[2] of $\rho(x)$ by considering $M \in \mathbb{Z}_{\geq 0}$ bins, to obtain a set of possible cost values $\rho(x) \triangleq \{\rho_1(x), \ldots, \rho_M(x)\}$ such that $\rho_M(x) < \rho_{M-1}(x) < \ldots < \rho_1(x)$ with their corresponding probabilities $p(x) \triangleq \{p_1(x), \ldots, p_M(x)\}$, such that $\sum_{i=1}^{M} p_i(x) = 1 \ \forall x \in \mathscr{C}$. Further, we will assume that $p_i(x_1) = p_i(x_2) \equiv p_i, \quad \forall x_1, \ x_2 \in \mathscr{C}$,and $i \in \{1, ..., M\}$. In other words, even though cost values $\rho_i(x)$, $i \in \{1, \ldots, M\}$, may change from point to point in $\mathscr{C}$, the probabilities $p_i(x)$ remain the same for different x. Note that we can do this wlog by discretizing the continuous RV appropriately, see Algorithm 3. The function *discretize* finds $y_i(x) < y_{i+1}(x)$ such that $\mathbb{P}[y_i(x) \leq \rho(x) \leq y_{i+1}(x)] = p_{i+1}(x) - p_i(x)$.

Now, the expected Risk $R^e(x)$ at a point x is

$$R^e(x) \triangleq \sum_{i=1}^{M} \rho_i(x) p_i(x). \tag{4.1}$$

That is, from (4.1) we have an expected risk $R^e : \mathscr{C} \to \mathbb{R}_{\geq 0}$ defined over $\mathscr{C}$ which is shown in Figure 1.2a and corresponds to a standard or rational notion of risk.

Next, we use the CPT notions developed in Section 2 to provide a non-rational perception model of the cost $\rho(x)$. According to CPT [TK92], there is a notion of cumulative functions $\Pi := \{\pi_1, ..., \pi_M\}$ used to non-rationally modify the perception of the probabilities $p_i(x)$ in a cumulative fashion. Defining a partial sum function

[2]The discretization of the random cost function is used to be able to use CPT directly with discrete random variables. However, it is possible to generalize what follows to the continuous random variable case.

$S_j(p_1,\ldots,p_M) \triangleq \sum_{i=j}^{M} p_j$ we have

$$\pi_j = w \circ S_j(p_1,\ldots,p_M) - w \circ S_{j+1}(p_1,\ldots,p_M), \tag{4.2}$$

where we employ the weighting function w from (2.3).

With this, a DM's CPT risk $R^c : \mathscr{C} \to \mathbb{R}_{\geq 0}$ associated to the configuration space is given by:

$$R^c(x) \triangleq \sum_{j=1}^{M} (v \circ \rho_j(x))(\pi_j \circ p(x)). \tag{4.3}$$

We note that both functions R^e and R^c are differentiable, which is important for the good behavior of the planner and which will be used for the analysis in Section 4.6.

Algorithm 3: CPT Environment (CPT-Env)

Input: $\rho_\mu(x)$, $\rho_\sigma(x)$, Θ, $\{p_1,...,p_M\}$
Output : $R^c(x)$
for $i \in \{1,...,M\}$ do
    $y_i(x), y_{i+1}(x) \leftarrow discretize(p_i(x), p_{i+1}(x))$;
    $\rho_i(x) \leftarrow \frac{y_{i+1}(x) - y_i(x)}{2}$;
end
$w \circ p(x) \leftarrow e^{-\beta(-\log \circ p(x))^\alpha}$;
$v \circ \rho(x) \leftarrow \lambda(\rho(x))^\gamma$;
for $j \in \{1,...,M\}$ do
    $\pi_j \leftarrow w \circ S_j(p_1,\ldots,p_M) - w \circ S_{j+1}(p_1,\ldots,p_M)$;
end
$R^c(x) \leftarrow \sum_{j=1}^{M} (v \circ \rho_j(x))(\pi_j \circ p(x))$;

Given an uncertain spatial cost ρ with the first moment ρ_μ (Figure 4.2b) and second moment ρ_σ (Figure 4.2c) across an environment, the DM's perception can vary from being rational (i.e. using expected risk R^e in Figure 1.2a) to non-rational (i.e using CPT risk R^c). By varying Θ, CPT risk R^c can be tuned to represent risk averse (Figure 1.2b), risk indifferent (Figure 4.2d) perception, as well as uncertainty indifferent (Figure 4.2e) to uncertainty averse (Figure 4.2f) perception.

This process gives us the CPT perceived risk at a point x, the process is summarized in Algorithm 3. It can be seen that Algorithm 3 does not depend on the dimensionality of the $\mathscr{C}$ space, but on the discretization factor M. We will now use the perceived environment for planning in the next section.

4.6 Sampling-based Planning using perceived risk

Here, we will use CPT notions to derive new cost functions, which will be used for planning in the DM's perceived environment generated in Section 4.5. In traditional RRT* optimal planning is achieved using path length as the metric. In our setting, the notion of path length is insufficient as it does not capture the risk in $\mathscr{C}$. Thus, we define cost functions that a) take into account risk and path length of a path, and b) satisfy the requirements that guarantee the asymptotic performance of an RRT*-based planner.

Path costs functions

Let two points $x, y \in \mathscr{C}$ be arbitrarily close. A decrease in risk is a desirable trait, hence it is reasonable to add an additional term in the cost only if $R(y) - R(x) \geq 0$, which indicates an increase in DM's perceived risk by traveling from x to y. Consider the set of parameterized paths $\mathscr{P}(\mathscr{C}) \triangleq \{\eta : [0,1] \to \mathscr{C} \mid \eta(0) = x, \eta(1) = y\}$. First, we first define the cost $J^c : \mathscr{P}(\mathscr{C}) \to \mathbb{R}_{\geq 0}$ of a path $\eta \in \mathscr{P}(\mathscr{C})$. Consider a discretization of $[0,1]$ given by $\{0, t_1, t_2, \ldots, t_L = 1\}$ with $t_{\ell+1} - t_\ell = \Delta t$, for all ℓ. Then, a discrete approximation of the cost over η should be:

$$J^c(\eta) \approx \Delta t \sum_{\ell=1}^{L} \max\{0, \frac{R^c(\eta(t_{\ell+1})) - R^c(\eta(t_\ell))}{\Delta t}\} + \delta L(\eta),$$

where $L(\eta)$ denotes the arc-length of the curve η, and $\delta \in \mathbb{R}_{\geq 0}$ is a constant encoding an urgency versus risk tradeoff. The greater the δ value, the greater is the urgency and hence path length is more heavily weighted whereas, smaller δ indicates greater

prominence towards risk. The choice of δ will be discussed in Section 4.8. By taking limits in the previous expression, and due to the continuity and integrability of max, we can express $J^c(\eta)$ as:

$$\begin{aligned} J^c(\eta) &= \lim_{\Delta t \to 0} \Delta t \sum_{\ell=1}^{L} \max\{0, \frac{R^c(\eta(t_{\ell+1})) - R^c(\eta(t_\ell))}{\Delta t}\} + \delta L(\eta) \\ &= \int_0^1 \max\{0, \frac{d}{dt}(R^c(\eta(t))\}\mathrm{d}t + \delta L(\eta) = \\ &\int_0^1 \max\{0, (R^c)'(\eta(t)) \cdot \eta'(t)\}\mathrm{d}t + \delta L(\eta). \end{aligned} \tag{4.4}$$

From here, the cost of traveling from x to y is given by

$$J^c(x,y) \triangleq \min_{\eta \in \mathscr{P}(\mathscr{C}): \eta(0)=x, \eta(1)=y} J^c(\eta).$$

Similarly, the path cost using expected risk $J^e : \mathscr{P}(\mathscr{C}) \to \mathbb{R}_{\geq 0}$ can be obtained by replacing the CPT cost R^c in (4.4) with the expected risk R^e as calculated in (4.1).

Remark 2. (Monotonicity). It can be verified that the costs J^c and J^e satisfy monotonic properties in the sense that 1) they assign a positive cost to any path in $\mathscr{P}(\mathscr{C})$, and 2) given two paths η_1 and η_2, and their concatenation $\eta_2|\eta_1$, in the space $\mathscr{P}(\mathscr{C})$, it holds that $J^c(\eta_1) \leq J^c(\eta_1|\eta_2)$ (resp. $J^e(\eta_1) \leq J^e(\eta_1|\eta_2)$), (due to the additive property of the integrals) and 3) J^c (resp. J^e) are bounded over a bounded $\mathscr{C}$.

Proposed Algorithm

Now we have all the elements to adapt RRT* to our problem setting. Given $\mathscr{C}$, a number of iterations T and a start point $x_s \in \mathscr{C}$, we wish to produce graph $G(V,E)$, which represents a tree rooted at x_s whose nodes V are sample points in the configuration space and the edges E represent the path between the nodes in V. Let $J^c_{\text{cum}} : \mathscr{C} \to \mathbb{R}_{\geq 0}$ be a function that maps $x \in \mathscr{C}$ to the cumulative cost to reach a point x from the root x_s of the tree $G(V,E)$ using the CPT cost metric (4.4). Similarly we define $J^e_{\text{cum}} : \mathscr{C} \to \mathbb{R}_{\geq 0}$ for the expected cost function J^e.

Algorithm 4: CPT-RRT*

Input: T, x_s, x_g ; Output : $G(V,E), P$
$V \leftarrow x_s$, $E \leftarrow \phi$, $J^c_{\text{cum}}(x_s) \leftarrow 0$;
for $i \in \{1,\ldots,T\}$ do
  $G \leftarrow (V,E)$; $x_{\text{rand}} \leftarrow \mathit{Sample}()$;
  $x_{\text{nearest}} \leftarrow \mathit{Nearest}(G, x_{\text{rand}})$; $x_{\text{new}} \leftarrow \mathit{Steer}(x_{\text{nearest}}, x_{\text{rand}})$;
  $V \leftarrow V \cup x_{\text{new}}$; $x_{\min} \leftarrow x_{\text{nearest}}$;
  $X_{\text{near}} \leftarrow \mathit{Near}(G, x_{\text{new}}, \gamma_{\text{RRT}^*}, d)$;
  $c_{\min} \leftarrow J^c_{\text{cum}}(x_{\text{nearest}}) + J^c(x_{\text{nearest}}, x_{\text{new}})$;
  for $x_{\text{near}} \in X_{\text{near}}$ do
    $c' \leftarrow J^c_{\text{cum}}(x_{\text{near}}) + J^c(x_{\text{near}}, x_{\text{new}})$;
    if $c' < c_{min}$ then
      $x_{\min} \leftarrow x_{\text{near}}$; $c_{\min} \leftarrow c'$;
    end
  end
  $J^c_{\text{cum}}(x_{\text{new}}) \leftarrow c_{\min}$; $E \leftarrow E \bigcup (\{x_{\text{near}}, x_{\text{new}}\})$;
  for $x_{\text{near}} \in X_{\text{near}}$ do
    $c' \leftarrow J^c_{\text{cum}}(x_{\text{new}}) + J^c(x_{\text{new}}, x_{\text{near}})$;
    if $c' < J^c_{\text{cum}}(x_{\text{near}})$ then
      $x_{\text{par}} \leftarrow \mathit{Parent}(x_{\text{near}}, G)$;
      $E \leftarrow (E \setminus (\{x_{\text{par}}, x_{\text{near}}\})) \bigcup (\{x_{\text{new}}, x_{\text{near}}\})$;
      $X_{\text{chld}} \leftarrow \mathit{Children}(x_{\text{near}}, G)$;
      for $x_{\text{chld}} \in X_{\text{chld}}$ do
        $J^c_{\text{cum}}(x_{\text{chld}}) \leftarrow J^c_{\text{cum}}(x_{\text{chld}}) - J^c_{\text{cum}}(x_{\text{near}}) + c'$
      end
      $J^c_{\text{cum}}(x_{\text{near}}) \leftarrow c'$
    end
  end
end
$P \leftarrow \mathit{Path}(G, x_s, x_g)$;

Remark 3. (Additivity). The cumulative costs J^h_{cum} and J_{cum} are additive with respect to costs J^c and J^e in the sense that: for any $x \in V$ we have $J^c_{\text{cum}}(x) = J^c_{\text{cum}}(\mathit{Parent}(x)) + J^c(\mathit{Parent}(x), x)$ and similarly $J^c_{\text{cum}}(x) = J^e_{\text{cum}}(\mathit{Parent}(x)) + J^c(\mathit{Parent}(x), x)$.

The other functional components of our algorithm CPT-RRT* (Algorithm 4) are similar to RRT*, and we briefly outline it out here for the sake of completeness:

- $\mathit{Sample}()$: Returns a pseudo-random sample $x \in \mathscr{C}$ drawn from a uniform distribution across $\mathscr{C}$. Other risk-averse sampling schemes as in [DSC16] may be employed. However, such schemes lead to conservative plans, which may not be suitable for all risk profiles.

- $\mathit{Nearest}(G, x)$: Returns the nearest node according to the Euclidean distance metric from x in tree G.

- $\mathit{Steer}(x_1, x_2)$ returns
$$\begin{cases} x_2, & \text{if } \|x_2 - x_1\| \leq d \\ x_1 + d\frac{x_2 - x_1}{\|x_2 - x_1\|}, & \text{otherwise.} \end{cases}$$

- $\mathit{Near}(G, x, \gamma_{RRT^*}, d)$: returns a set of nodes $X \in V$ around x, which are within a radius as given in [KF11].

- $\mathit{Parent}(x, G)$: Returns the parent node of x in the tree G.

- $\mathit{Children}(x, G)$: Returns the list of children of x in G.

- $\mathit{Path}(G, x_s, x_g)$: Returns the path from the nearest node to x_g in G to x_s.

We note that in order to compute J^c for each path, we approximate the cost as the sum of costs over its edges, (x_1, x_2), and for each edge we compute the cost as the differences $\max\{0, R^c(x_2) - R^c(x_1)\} + \delta L(x_1, x_2)$, where the latter is just the length of the edge. Then, this approximation will approach the computation of the real cost in the

limit as the number of samples goes to infinity. The values R^c are evaluated according to Algorithm 3. Our proposed CPT-RRT* algorithm augments RRT* algorithm in the following aspects: we consider a general continuous cost profile which leads to no obstacle collision checking. We also consider both path length and CPT costs for choosing parents and rewiring with the parameter δ which serves as relative weighting between CPT costs and Euclidean path length.

Remark 4. (ER-RRT*). We can obtain the expected risk version of Algorithm 4 by replacing cost function J^c by J^e and following the same procedure as Algorithm 4.

Lemma 2. (Asymptotic Optimality). Assuming compactness of $\mathscr{C}$ and the choice of γ_{RRT^*} according to Theorem 38 in [KF11] , the CPT-RRT* algorithm is asymptotically optimal.

Proof. It follows from the application of Theorem 38 in [KF11], and the conditions required for the result to hold. More precisely, the cost functions are monotonic (which follows from Remark 2), it holds that $c(\eta) = 0$ iff η reduces to a single point (resp. the same for c), and the cost of any path is bounded. The latter follows from the compactness of $\mathscr{C}$ and continuity of the cost functions. In addition, the costs are also cumulative, due to the additivity property in Remark 3. Finally, the result also requires the condition of the zero measure of the set of points of an optimal trajectory. This holds because both costs include a term for path length. □

Simulation results of CPT-RRT* algorithm are presented in Section 4.8.1. Next we describe our proposed method to evaluate and compare risk perception models in our setting.

4.7 CPT-planner parameter adaptation

In this section, we theoretically compare CPT value function with other risk perception functions and we describe an algorithm that can adapt the CPT parameters

of the planner to approximate arbitrary paths in the environment. By doing so, we aim to evaluate the expressive power of the CPT risk perception model, both theoretically and in a motion planner by comparing its capability to approximate single and arbitrary paths in the environment versus other approaches using different risk perception models.

If successful, this method could be used as a first ingredient in a larger scheme aimed at learning the risk function of a human decision maker[3] using techniques such as inverse reinforcement learning (IRL). We recall that IRL requires either discrete state and action spaces or, if carried out over infinite-dimensional state and action spaces, a class of parameterized functions that can be used to approximate system outputs. Since our planning problem is defined over a continuous state and action space, the class of CPT planners for a parameter set could play the role of a function approximation class required to apply IRL. Then, as is done in IRL, a larger collection of path examples can used to learn the best weighted combination of specific CPT planners in the class. While certainly of interest, this IRL question is out of the scope of this work, and we just focus on analyzing the expressive power of the proposed class of CPT planners. Having a good expressive power is a necessary prerequisite for the class of CPT planners to constitute a viable function approximation class. Firstly, we will define the notion of expressiveness and compare the expressiveness risk perception models from a theoretical point of view. Next, we will describe an approach to compare expressiveness in a path planning setting using SPSA.

4.7.1 Expressiveness for a risk perception model.

Let ρ be a random cost variable with an associated probability distribution. Let R be a risk value function (with $R(\rho) \in \mathbb{R}_{\geq 0}$) which associates a real value to the random cost variable ρ. We can compare the expressiveness of two risk perception models by

[3] Just for offline planning, or in situations where the human does not update the environment online as new information is found.

comparing the range space of their respective risk value functions.

Definition 2. (Expressiveness). Consider two risk perception models $\mathscr{M}_1$ and $\mathscr{M}_2$ with corresponding classes of risk value functions $\mathscr{V}_1$ and $\mathscr{V}_2$ with respective range spaces $\mathscr{R}_1$ and $\mathscr{R}_2$. We say that $\mathscr{M}_1$ is more expressive ($\geqq$) than $\mathscr{M}_2$ if $\mathscr{R}_2 \subseteq \mathscr{R}_1$ for any given positive random variable ρ. That is,

$$\mathscr{M}_1 \geqq \mathscr{M}_2 \iff \{R_2(\rho)|R_2 \in \mathscr{V}_2\} \subseteq \{R_1(\rho)|R_1 \in \mathscr{V}_1\}$$

With this definition, we can compare expressivity of CPT with Conditional Value at Risk (CVaR) [ADEH99], also known as "expected shortfall", another popular risk perception model in the financial decision making community. CVaR uses a single parameter $q \in [0,1)$ representing the fraction of worst case outcomes to consider for evaluating expected risk of an uncertain cost ρ. We will use R^v_Q to denote the perceived risk by CVaR model with $q = Q$. So a $q \approx 1$ considers the worst case outcome of ρ and a $q = 0$ considers all the outcomes thus making the CVaR value equal to expected risk ($R^v_0 = R_E$). Now we will proceed to compare expressiveness of Expected Risk, CVaR and CPT with parametrized risk value function classes R^e, R^v and R^c respectively.

Proposition 1. Let us Consider Expected risk (ER), CVaR and CPT risk models with risk value function classes R^e, R^v and R^c defined accordingly. Then, Expected Risk is the least expressive of the three models, that is CVar $\geqq$ ER and CPT $\geqq$ ER for any given random variable ρ.

Proof. The function class R^e has a single function $\mathbb{E}(\rho)$ which gives the expected value of ρ. So the range set of R^e is a singleton, containing the expected value of ρ. It is easy to see that by choosing a function $R^v_0 \in R^v$ and $R^c_{\overline{\Theta}} \in R^c$ where $\overline{\Theta} = \{1,1,1,1\}$ we have $R^v_0(\rho) = R^e(\rho) = R^c_{\overline{\Theta}}$. Which implies that $\mathbb{E}(\rho) \in \{R^v_q(\rho)|q \in [0,1)\}$ and also $\mathbb{E}(\rho) \in \{R^c_\Theta(\rho)|R^c_\Theta \in R^c\}$, thus proving the expressive order. □

Next we will look at the relationship between the CVaR value of a random variable ρ and the expected value of another random variable $\kappa\rho$ where $\kappa > 0$ is a scaling factor.

Proposition 2. Let us consider CVaR value of a given random variable ρ, then there exists a $\kappa_q \geq 1$ such that $R^v_q(\rho) = \mathbb{E}(\kappa_q \rho)$ for all $q \in [0,1)$.

Proof. The range space of R^v is $[\mathbb{E}(\rho), b]$, where b is the worst case outcome of ρ. We also know that $\mathbb{E}(\kappa\rho) = \kappa\mathbb{E}(\rho)$. From this we can construct $\kappa_q = \frac{R^v_q(\rho)}{\mathbb{E}(\kappa\rho)}$ which shows that $R^v_q(\rho) = \mathbb{E}(\kappa_q \rho)$ □

From this we can compare the expressivity of CPT and CVaR models.

Proposition 3. Let us consider CVaR and CPT risk models with risk value function classes R^v and R^c defined accordingly. Then, we have the expressive order CPT $\geqq$ CVaR for any given random variable ρ.

Proof. Considering a subclass of CPT value functions $R^c_{\Theta*}$ where $\Theta* \in \{\Theta | \alpha = 1, \beta = 1, \gamma = 1, \lambda = \kappa_q\}$, we have CPT value $R^c_\theta = \mathbb{E}(\kappa_q \rho)$ with $\theta \in \Theta*$ and some constant κ_q defined according to previous proposition. From this we can say that $\{R^v_q(\rho) | q \in [0,1)\}$ $\subseteq \{R^c_\theta | \theta \in \Theta*\} \subseteq \mathscr{R}(R^c)$, which concludes the proof. □

Lemma 3. Let us Consider Expected risk (ER), CVaR and CPT risk models with risk value function classes R^e, R^v and R^c defined accordingly. Then, we have the following expressive order : CPT $\geqq$ CVaR $\geqq$ ER.

Proof. The proof immediately follows from the above three propositions. □

The above arguments imply that risk aversion can equivalently be modeled as an expected value of a scaled random variable, with greater scaling implying higher risk aversion. This is captured in both CPT (λ parameter) and CVaR (q parameter) models. Additionally, CPT also captures risk sensitivity and uncertainty sensitivity which makes

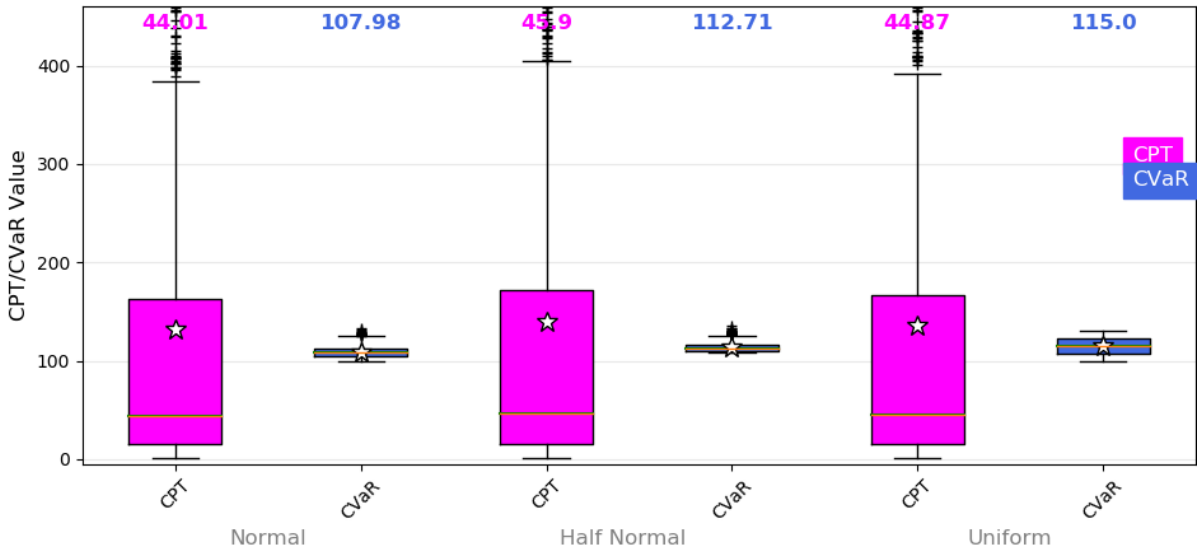


Figure 4.1. Boxplots showing the distribution of 1000 value samples of CPT and CVaR risk perception model for three different distributions.

it more expressive than CVaR. This can be visualized in Fig. 4.1. A thousand samples of $\Theta \in \mathbb{R}^4$ were drawn uniformly randomly for CPT, while for CVaR, q was sampled uniformly across $[0, 1]$. Three distributions: Normal, half Normal and Uniform, were considered with mean 100 and standard deviation of 10 for the first two and a range of $[70, 130]$ for the latter. The median values for each box plot is indicated on the top row. The mean value of the distribution is indicated as "stars", the black lines above and below the box represent the range, and + indicates outliers. It can be clearly seen that the range of values captured by CPT is greater, which is in accordance with the theoretical argument above. Next, we will propose a method to evaluate expressiveness in the context of path planning.

4.7.2 Comparing expressiveness in path planning

Let us suppose that we have an arbitrary example path P_d drawn in the environment. If the class of CPT planners is expressive enough, we should be able to find a set of parameters that is able to to exactly mimic this drawn path. Since an arbitrary path

P_d belongs to a very high dimensional space[4] and the planner parameters are typically finite, any amount of parametric tuning may not produce good approximations. This is what we evaluate in the following. We use the term $\mathrm{Ar}(P;P_d) \in \mathbb{R}_{\geq 0}$ to denote the area enclosed between the given path P_d and another path P. This value measures the closeness between P and P_d.

A path P produced by a CPT planner can be represented by the CPT parameters Θ. In order to find the closest possible path P^* to P_d we have to evaluate

$$\underset{P_\Theta, \Theta \in \mathscr{T}}{\operatorname{argmin}} \mathrm{Ar}(P_\Theta; P_d), \tag{4.5}$$

where P_Θ is the path produced by CPT-RRT* with CPT parameters Θ, and $\mathscr{T}$ is the set of all possible values of Θ. Directly evaluating (4.5) is computationally not feasible as the set $\mathscr{T}$ is infinite and resides in $4D$ space.

An alternative to (4.5) is to use parameter estimation algorithms to determine $\Theta^* \in \mathscr{T}$ which characterizes the path P^* with $\mathrm{Ar}(P_\Theta; P_d)$ as a loss/cost function. We note that neither $\mathrm{Ar}(P_\Theta; P_d)$ can be computed directly (without running CPT-RRT* first), nor the gradient of Ar wrt Θ is accessible. This limits the use of standard gradient descent algorithms to estimate Θ^*. To address this problem, we use SPSA [Spa03] with $\mathrm{Ar}(P_\Theta; P_d)$ as the loss function to estimate the parameters Θ^*. Next, we briefly explain the main idea and adaptation of SPSA to our setting and refer the reader to [Spa03] for more detailed treatment and analysis of the SPSA algorithm.

We start with an initial estimate Θ_0 and iterate to produce estimates Θ_k, $k \in \mathbb{Z}_{\geq 0}$ using the loss function measurements $\mathrm{Ar}(P_{\Theta_k}; P_d)$. The main idea is to perturb the estimate Θ_k according to [Spa03] to get Θ^+ and Θ^-, for the k^{th} iteration. These perturbations are then used to generate the perturbed paths $P_{\Theta^+}, P_{\Theta^-}$ using Algorithm 4.

[4] An arbitrary path can be modeled as a curve defined by a large number of parameters (possibly infinite).

With these perturbed paths, the loss function measurements $\mathrm{Ar}(P_{\Theta^+}; P_d), \mathrm{Ar}(P_{\Theta^-}; P_d)$ are evaluated and used to update our parameter Θ_k according to [Spa03]. To test the goodness of the updated parameter, we determine the corresponding path $P_{\Theta_{k+1}}$ and measure $\mathrm{Ar}(P_{\Theta_{k+1}}; P_d)$. If the area is within a tolerance $\kappa \in \mathbb{R}_{>0}$, that is, if $\mathrm{Ar}(P_{\Theta_{k+1}}; P_d) < \kappa$, the iteration stops and $P_{\Theta_{k+1}}$ is returned. We followed the guidelines from [Spa03] for choosing the parameters used in SPSA. The results of this adaptation are evaluated and compared with the results that employ other risk perception models in Section 4.8.

4.8 Results and Discussion

In this section we illustrate the results of the solutions to the problem statement proposed in Sections 4.5 and 4.6, considering a specific scenario having some risk and uncertainty profiles.

4.8.1 Environment Perception and Planning

We consider a hypothetical scenario where an agent needs to navigate in a room during a fire emergency. In this, the 2D configuration space for planning becomes $\mathscr{C} = [-10, 10] \times [-10, 10]$. The agent is shown a rough floor map (Figure 4.2a) with obstacles (which are thought to be ablaze) in the environment with a blot of ink/torn patch, making that region unclear and hard to decipher. This results in the spatial uncertain cost ρ with first moment (ρ_μ) represented by cost associated to obstacles and fire source and second moment (ρ_σ) represented by the uncertainty associated to the ink spot/tear.

The blue colored objects are the obstacles whose location is known to be within some tolerance (dark green borders) and the light orange ellipses illustrate that these objects have caught fire. The grey ellipse indicates a possible tear/ink spot on the map, which makes that particular region hard to read. The start and goal positions are in-

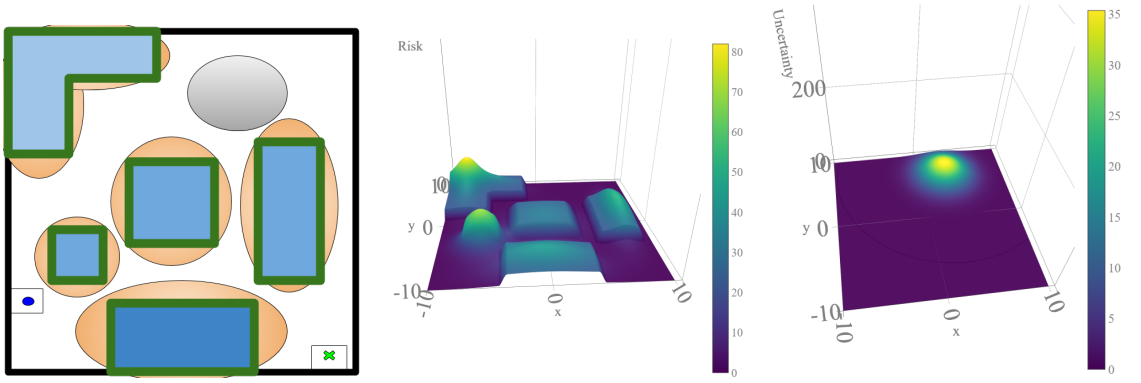
dicated as blue spot and green cross respectively. We use a scaled sum of bi-variate Gaussian distribution to model the sources of continuous cost (orange ellipses) with appropriate means and variances to depict the scenario in Figure 4.2a. We utilize bump functions from differential geometry to create smooth "bumps" depicting the discrete obstacles. One approach to do this is described in Section 4.3. An alternative procedure is briefly described as follows. Consider the maximum cost value imparted to the obstacles as $\rho^{\max} \in \mathbb{R}_{\geq 0}$ and let $a_1, a_2, b_1, b_2 \in \mathbb{R}_{\geq 0}$ be the inner (blue rectangle) and outer (dark green borders) measurements of the obstacles from the center $c = (c_1, c_2) \in \mathscr{C}$. Let $x = (x_1, x_2) \in \mathscr{C}$ be a point in the configuration space with f, g, h being real valued scalar functions given by $f(y) = e^{-\frac{1}{y}}, y \in \mathbb{R}_{>0}$ and $f(y) = 0$ otherwise, $g(y) = \frac{f(y)}{f(y)+f(1-y)}$ and $h(y) = 1 - g(\frac{y^2-a^2}{b^2-a^2})$. Then, $\rho_\mu(x)$ can be calculated by :

$$\rho_\mu(x) = \rho^{\max} h(x_1 - c_1) h(x_2 - c_2). \tag{4.6}$$

This procedure produces smooth "bumps" in the cost profile which are visualized in Figure 4.2b using $\rho^{\max} = 20$. This approach can be easily generalized to arbitrary high dimensions by simply multiplying upto $h(x_i - c_i)$ terms in (4.6) to create a bump function in the i^{th} dimension. To generate the second moment of cost ρ_σ, we use a scaled bivariate Gaussian distribution with appropriate means and variances to depict the ink spot/tear in Figure 4.2a. Now we will illustrate the results of implementing Algorithm 4 in this environment.

Simulations and discussions

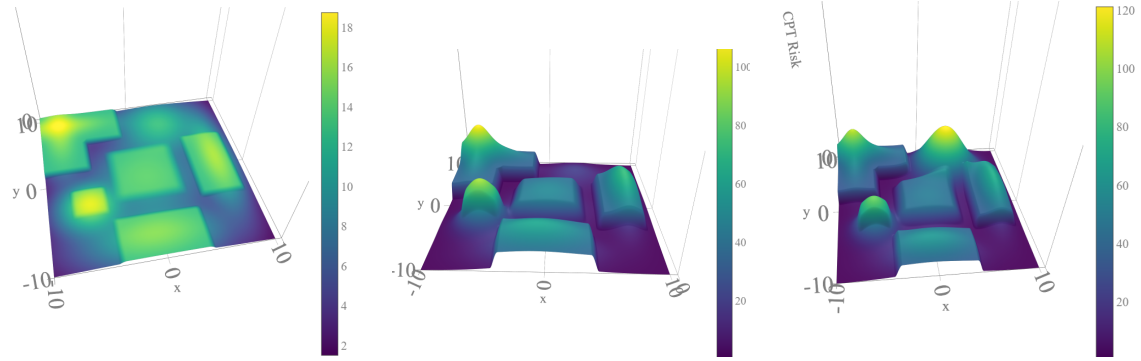
With the uncertain cost ρ with moments ρ_μ and ρ_σ from previous paragraph, we use a half Normal distribution and discretization factor $M = 20$ to generate the costs $\rho(x)$ and their corresponding $p(x)$ from Section 4.5. The results of using Algorithm 3 to every point in $\mathscr{C}$ to generate the perceived environment is shown in Figures 1.2 and



(a) Rough sketch of environment.

(b) Mean cost ρ_μ

(c) Uncertainty ρ_σ



(d) Risk indifferent profile with $\Theta = \{0.74, 1, 0.3, 2.25\}$

(e) Uncertainty indifferent profile with $\Theta = \{0.74, 3, 0.88, 2.25\}$

(f) Uncertainty averse profile $\Theta = \{0.74, 0.05, 0.88, 2.25\}$

Figure 4.2. Environment perception using CPT.

4.2. The level of risk at a point R^c or R^e is indicated by color map. Figure 1.2a shows a rationally perceived environment using expected risk R^e. Whereas, Figure 1.2b indicates a non-rational highly risk averse perception using CPT (R^c) with $\Theta = \{0.74, 2, 0.9, 10\}$ having a high λ value. A risk indifferent profile (Figure 4.2d) is generated by $\Theta = \{0.74, 1, 0.3, 2.25\}$ having a low risk sensitivity γ value. Similarly, uncertainty indifferent profile (Figure 4.2e) and uncertainty averse profile (Figure 4.2f) are generated by fixing α and having high and low β values respectively.

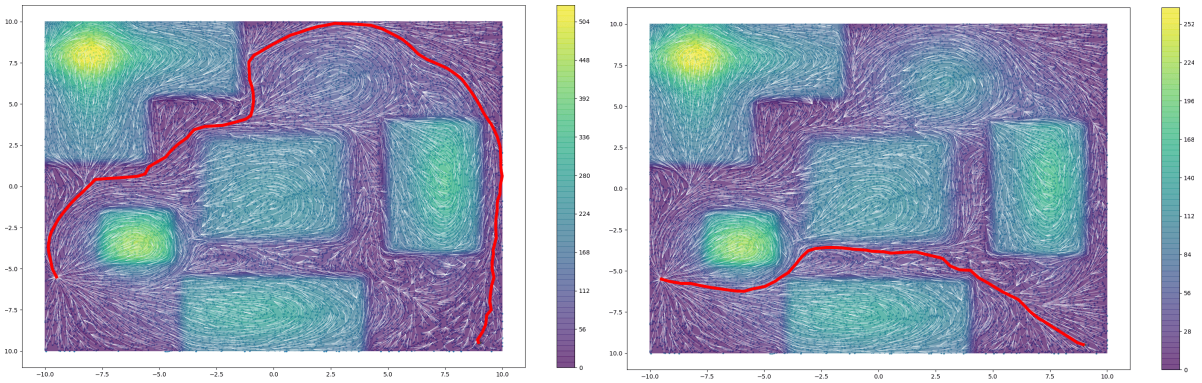
After the perceived environment is generated, Algorithm 4 is used to plan a path from the start point to the goal point shown in Figure 4.2a. We use $T = 20,000$ iterations for the CPT-RRT* algorithm with $\delta = 10^{-4}$. The same random seed was used for

all executions for consistency. The path planning results are illustrated in Figure 4.3. As expected, we see that the path depends on the perceived risk profile. Figure 4.3a indicates a circuitous path due to the highly risk averse perception, whereas Figure 4.3d indicates a shorter and more direct path for a rational DM using expected risk. Increasing the uncertainty sensitivity (lowering β) and reducing risk aversion (lowering λ) makes the planner avoid the highly uncertain ink spot/tear in the top-right region and take a more riskier path in the lower region as shown in Figure 4.3b. By having a medium risk aversion and lower uncertainty sensitivity (increasing β), the planner produces a different path through the medium risky and uncertain middle region as shown in Figure 4.3c.

We also demonstrate the capability of our planner in 3D space, which is illustrated in Fig. 4.4. We employ a cubic configuration space measuring 10 units, cluttered with randomly placed 50 cube obstacles of unit volume. The start position is $(1,1,1)$ and goal position is $(9,9,9)$. There is also a continuous source of risk (modeled as a scaled normal distribution as before) centered at $(3,3,3)$. We use $\Theta = \{0.74, 1, 0.9, 5\}$ (same as 4.3b) with $\gamma_{RRT*} = 500$, $d = 0.2$ and $\delta = 10^{-6}$. From Fig. 4.4, we see that CPT-RRT is able to find a reasonably smooth path avoiding obstacles and the risky area, in similar number of iterations as in the 2D case. This shows the capability of our planner in 3D space.

Solution quality

Figure 4.5 illustrates the empirical convergence and solution quality of the paths produced by our algorithm. We performed empirical convergence tests, by running CPT-RRT* 100 times with the same parameters and initial conditions and measuring the area between paths produced after every 500 iterations for a total of $80,000$ iterations. The results are shown in Figure 4.5a. We see that initially (< 10000 iterations) there are changes in the output path as the space is being explored and the output path is



(a) High cost and high uncertainty sensitivity ($\Theta = \{0.74, 1, 0.9, 10\}$)

(b) Medium cost aversion and high uncertainty sensitivity ($\Theta = \{0.74, 1, 0.9, 5\}$)

(c) Medium cost aversion and low uncertainty sensitivity ($\Theta = \{0.74, 2, 0.9, 5\}$)

(d) Expected Risk

Figure 4.3. Paths produced by CPT-RRT* under different perception models. White lines indicate the tree grown from the start position, red line indicates the optimal path to goal after $T = 20,000$ iterations. Background color map depicts the CPT costs in (a)-(c) and expected costs in (d)

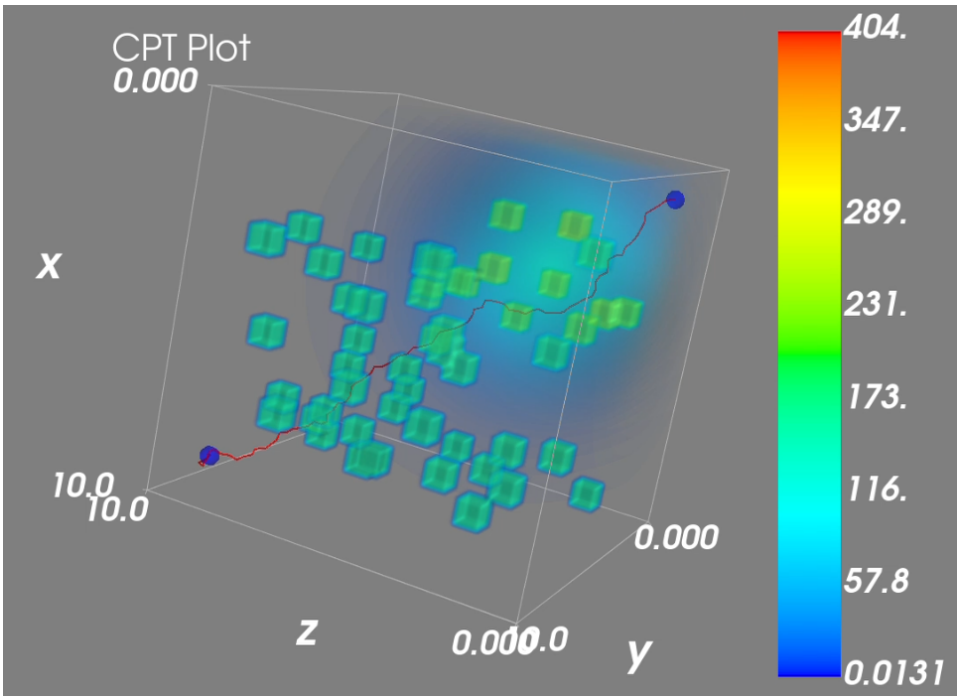
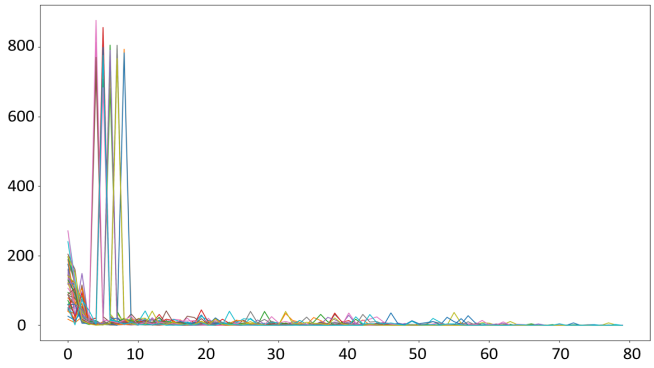


Figure 4.4. Path(Red line) by CPT-RRT* in 3D environment with 50 discrete random obstacles and a continuous risk source after 20,000 iterations.

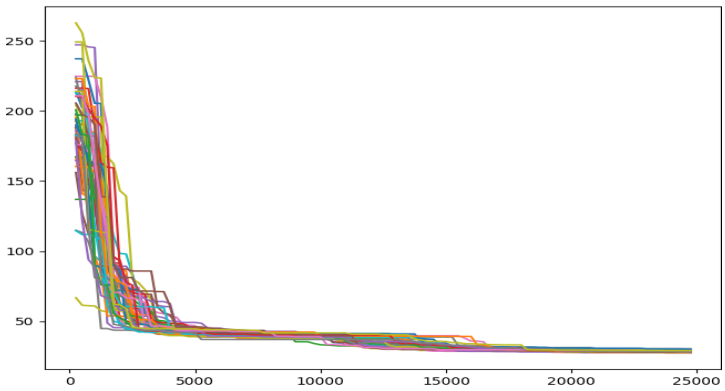
changing. After $10,000$ iterations we consistently see minimal path changes indicating that the algorithm is converging towards a desirable path. Then we also checked the solution quality of the path by computing the cost of the output path every 250 iterations as shown in Figure 4.5b for 100 trials consisting of $25,000$ iteration. We see that the there is a consistent decrease in path cost in all the trials throughout. We also note that after $10,000$ iterations the cost decrease starts to plateau, indicating that the algorithm is close to a high quality (low cost) solution. From these observations of Figure 4.5, we recommend upwards of $T = 10,000$ iterations to achieve smooth and consistent paths in our setting.

Comparison in narrow and cluttered environments

Here, we will illustrate and compare the performance between our RRT* framework and T-RRT* [DSC16] (another algorithm operating on continuous cost spaces) in a cluttered environment with narrow passages as shown in Figure 4.6. To construct this

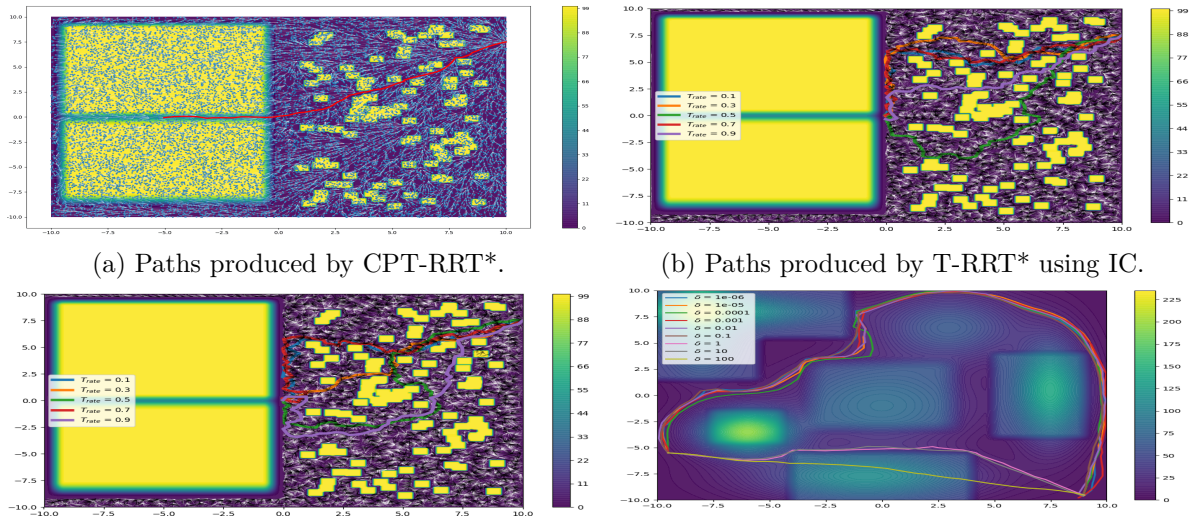


(a) Convergence over iterations



(b) Path cost over iterations

Figure 4.5. a) Empirical convergence analysis. The distance between paths after every 500 iterations (y-axis) with the number of iterations in thousand (x-axis). b) Cost of the output path (y-axis) every 250 iterations with the number of iterations (x-axis)



(a) Paths produced by CPT-RRT*.

(b) Paths produced by T-RRT* using IC.

(c) Paths produced by T-RRT* using MW.

(d) Paths produced by varying δ

Figure 4.6. Path comparison with T-RRT* and varying δ. (a)-(c)Paths produced in a cluttered environment using $T = 20,000$ iterations for CPT-RRT* and 20,000 nodes for T-RRT*. (d) Paths produced by CPT-RRT* by varying δ with $\Theta = \{0.74, 1, 0.9, 7.5\}$.

environment, we used 100 randomly placed small objects on the right half and two big objects separated by a narrow passage on the left half. Start point x_s is on the top right corner and the goal x_g is at the center of the narrow passage. Bump functions similar to previous paragraphs were used to construct a smooth spatial cost ρ from the obstacles. Since T-RRT* does not have risk perception capabilities, for a fair comparison we use the continuous cost ρ to implement both algorithms. In this way, we will be able to specifically compare the planning capabilities of both algorithms in the same continuous cost environment. We used $\gamma_{\text{RRT}^*} = 100$ and $d = 0.35$ for both algorithms. From Figure 4.6a we can see that our algorithm is able to sample and generate paths in the narrow passage, as well as avoid obstacles in a cluttered environment. In comparison, we can see that from T-RRT* employing integral cost (IC) in Figure 4.6b and minimum work (MW) in Figure 4.6c cannot generate paths in the narrow costly region fast enough irrespective of the T_{Rate} used due to the sampling bias towards low-cost regions. We used nodes instead of iterations for T-RRT* to maintain an equal number of nodes in the tree, as a node does not get added if it fails the transition test. Also, T-RRT* paths do

not appear to be as smooth as the paths from our framework, irrespective of the cost(IC or MW) used. We also note that, the cluttered and high cost environment induces a high failure rate of the transition test, resulting in longer run times of T-RRT* required to build the same number of nodes as our algorithm, especially for high T_{Rate} values.

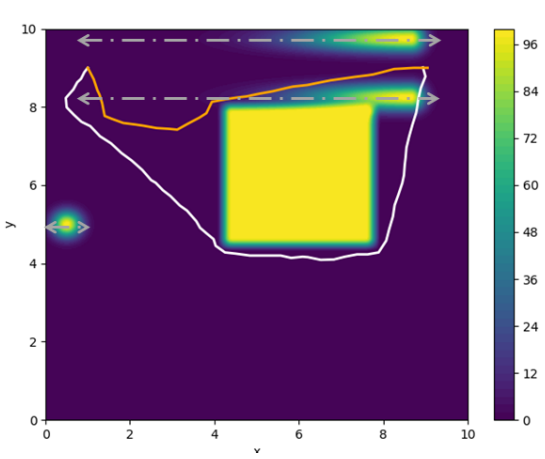
Comparison in dynamic environments

Here, we contrast the performance of CPT-RRT* and Risk-RRT* [CM17] (a risk aware planner) in a 10 by 10 environment area with static and moving obstacles as shown in Figure 4.7. To account for risk dynamics, we will be planning in the space-time domain, and assume knowledge of the dynamics (or a good estimate) of $\rho_\mu(t)$ and $\rho_\sigma(t)$, which will result in a time-varying perceived risk map $R^c(x,t)$. We also assume that each edge in the tree will be traversed in some time Δt. The underlying RRT* parameters employed by both algorithms were taken to be identical, with $\gamma_{RRT*} = 100$ and $d = 0.25$, while $\delta = 0.1$ for CPT-RRT*. Our starting point is the same parametric CCR Risk Map [CKT$^+$16] as in Risk-RRT*, which generates a continuous, and time-varying, cost map based on the pose and velocity of a moving human as shown in Fig. 4.7a (a snapshot). The human obstacles move back and forth within the indicated range (gray line), with top two obstacles moving d units in Δt time, while middle left obstacle moves at $0.1d$ units. Since the CCR map does not incorporate uncertainty, we will use it as the mean cost $\rho_\mu(t)$. We employ a scaled normal distribution on top of each source of dynamic risk (moving human) to denote $\rho_\sigma(t)$, representing uncertainty for each source. From $\rho_\mu(t)$ and $\rho_\sigma(t)$, we calculate $R^c(x,t)$ according to Algorithm 3 which is visualized in Fig. 4.7b. Note that with higher risk (moving obstacles as compared to static or no obstacles), the lighter the color in the map. We compare our algorithm with Risk-RRT* in two scenarios. At first, for fair comparison, and since Risk-RRT* does not consider uncertainty or risk perception models, we use directly the CCR cost map or $\rho_\mu(t)$ for planning. This corresponds to a rational DM model. The results are summarized

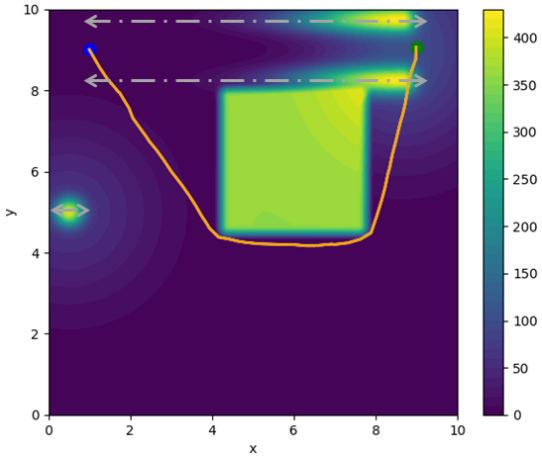
in Fig. 4.7c which show path length and cumulative risk returned by CPT-RRT* and Risk-RRT* using the CCR map in 50 trials. In general, we see a lower performance in Risk-RRT* due to its conservative approach in dealing with risk. First, since risk is not explicitly accounted for in the cost function of Risk-RRT* and "risk" is treated as an "obstacle" to avoid, the resulting path produced by Risk-RRT* is longer, even though its cost function optimizes path length. The length of the path from our planner is shorter, with comparable cumulative risk of the output paths of both planners computed as in (4.4). Furthermore, 12 out of the 50 trials in case of Risk-RRT* could not find a solution within 15,000 iterations. This seems to be a consequence of a higher sample rejection rate due to the tight free spaces created by the dynamic obstacles when close to other objects. This drawback is more pronounced when considering a risk-averse DM. Fig. 4.7b represents such DM who perceives that getting close to the dynamic obstacles is highly risky, as compared to the perception of a rational DM represented by Fig. 4.7a. In this way, the risk values in Fig. 4.7b are in the range $0-421$, which is much higher than those of the CCR Map in Fig. 4.7a (with ranges $0-100$). Due to higher risk values as given by this map, the sample rejection in Risk-RRT* is very high and could not find a feasible path in any of 50 trials, whereas CPT-RRT* consistently found a path similar to the one shown in Fig. 4.7b in all of the 50 trials.

Variation in δ

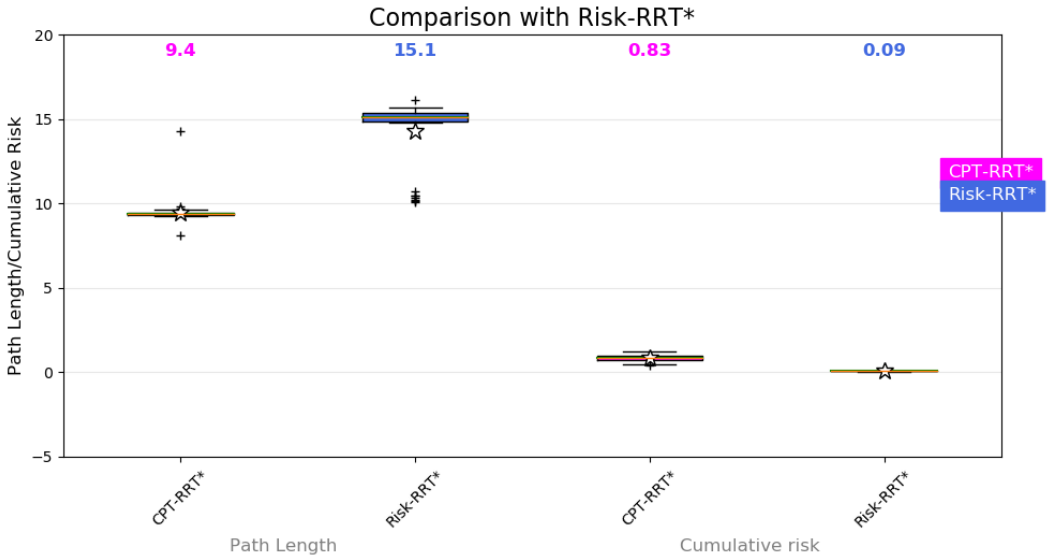
Using the previous environment (Figure 4.2a) and a cost and uncertainty averse profile (Figure 4.3a)), we run CPT-RRT* with δ varying from $\delta = 10^{-6}$ to $\delta = 10^{2}$ for $T = 15,000$ iterations. The results are shown in Figure 4.6d. We can see that when $\delta \geq 1$ the path output changes reflecting an increase in urgency over risk and thus choosing shorter paths. When δ is comparable to the risk values (in this case $\delta = 100$), we see that the paths no longer avoid the high risk area and can even go through the soft obstacles. From this study, we observe that δ needs to be rather small as compared to



(a) Paths returned by CPT-RRT*(Orange) and Risk-RRT*(White) in CCR map



(b) Paths returned by CPT-RRT*(Orange) in risk averse perception of CCR map with $\Theta = \{0.74, 1, 0.88, 6.25\}$



(c) Boxplots showing path length and cumulative risk returned by CPT-RRT* and Risk-RRT* using the CCR map in 50 trials (38 trials for Risk-RRT* as 12 did not succeed in finding a path) consisting of 15,000 RRT* iterations.

Figure 4.7. Comparison with Risk-RRT*. The CCR Map containing 3 moving humans and a stationary obstacles at initial time is shown in the background.

the given risk profile in order to ensure meaningful consideration of risk in the planning process. If explicit obstacle avoidance is a necessity, then a standard collision check can be performed prior to adding a node in the tree $\mathscr{G}$.

Overall, our adaptation of CPT to the planning setting produces paths that are logically consistent with a given risk scenario. Additionally, our planning framework can explore narrow corridors and cluttered environment and produce smooth paths quickly.

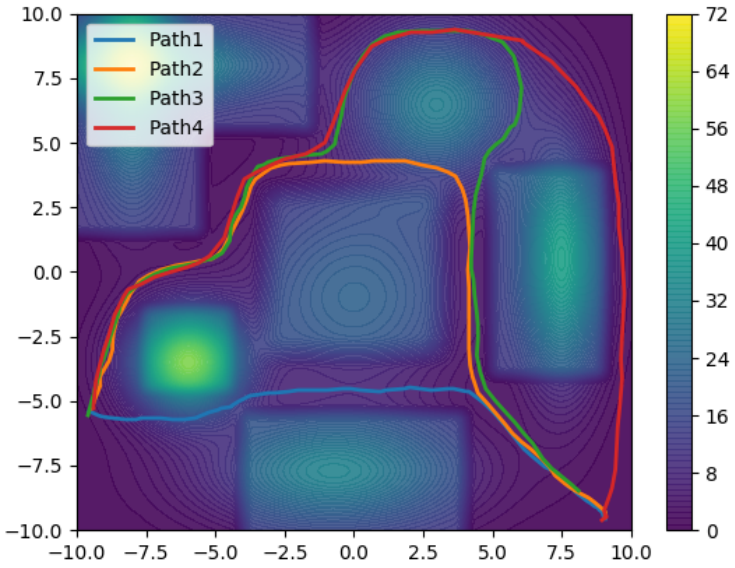
4.8.2 CPT planner expressive power evaluation

We now discuss the proposed SPSA framework in Section 4.7 to gauge the adaptability of CPT as a perception model to depict a drawn path P_d. To implement SPSA, we follow guidelines from [Spa03]. We consider a Bernoulli distribution of Δ_k with support $\{-1,1\}$ and equal probabilities, learning rate $a_k = \frac{0.4}{(1.6+k)^{0.601}}$ and perturbation parameter $c_k = \frac{0.97}{(1.6+k)^{0.301}}$. We choose $\Theta_0 = \{0.74, 1, 0.88, 2.25\}$ for CPT throughout the simulation, which are the nominal parameters from [TK92] and $q_0 = 0.5$ for the CVaR variant. We use the same environment as in Figure 4.2 for all the simulations. Four different paths $\{P_d^1, P_d^2, P_d^3, P_d^4\}$ are drawn by hand on the expected risk profile (Figure 4.3d) using a computer mouse as shown in Figure 4.8a. Path P_d^1 is similar to a path generated with expected risk perception (Figure 4.3d). Whereas, path P_d^4 and P_d^2 are similar to paths generated with high risk aversion (Figure 4.3a) and uncertainty insensitivity (Figure 4.3c) respectively. Path P_d^3 is more challenging to represent as it shows an initial aversion to risk and uncertainty and then takes a seemingly costlier turn at the top. We then use the SPSA approach described in Section 4.7 with a tolerance $\kappa = 15$ and a maximum of 10 SPSA iterations per trial. We use $T_k = 15000$ iterations and $\delta = 0.01$ to implement Algorithm 4 to determine P_Θ in order to determine the loss $\mathbf{Ar}$ during each SPSA iteration. For the CVaR variant, the planner (Algorithm 4) replaces R^c with R^v in order to use perceived risk according to CVaR while the rest of the RRT* framework

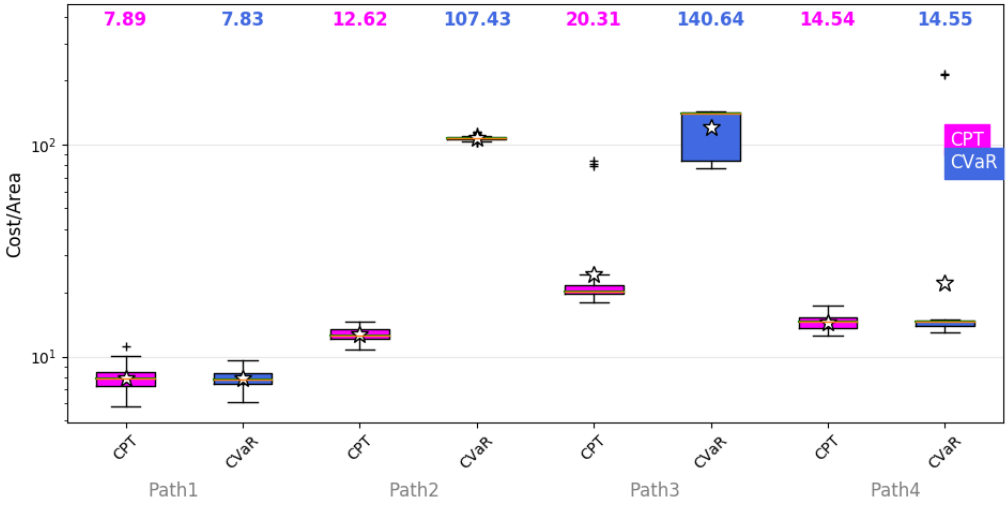
remains unchanged. At the end each trial we get the area (loss) Ar between the returned P_Θ and the drawn path P_d^x.

We represent the statistics of the returned cost Ar as boxplots as shown in Figure 4.8b. Each box plot represents the distribution of 50 cost/Area Ar values returned after each trial for each path and perception model. The Y-Axis represents the cost/Area Ar in a base 10 log scale. We calculate a few sample areas: $\text{Ar}(P_d^1, P_d^2) = 99.14, \text{Ar}(P_d^2, P_d^3) = 35.20$ and $\text{Ar}(P_d^3, P_d^4) = 73.41$ to give a quantitative idea of the measure Ar in this scenario to the reader. The median values for each box plot is indicated on the top row. The mean value of the distribution is indicated as "stars", the black lines above and below the box represent the range, and + indicates outliers. We observe that from Figure 4.8b, both Path P_d^1 and Path P_d^4 were captured equally well with CVar and CPT with low Ar values. Since both CPT and CVaR are generalizations of expected risk, paths close (like P_d^1) to paths generated from expected risk can be easily mimicked. Similarly, since CPT and CVaR are designed to capture risk aversion, paths close (like P_d^4) to risk averse paths (Figure 4.3a) can also be easily captured.

However, we see a contrast in performance for path P_d^2 and path P_d^3. CPT, on both occasions, is able to track the drawn paths reasonably well with low Ar values. Whereas CVaR has consistently higher (an order of magnitude) Ar values, indicating the inability to capture the risk perception leading to path P_d^2 and path P_d^3. This is due to the fact that CPT can handle uncertainty perception independently from the cost (as seen between Figure 4.2e and Figure 4.2f). This ability is needed to capture paths like P_d^2 and P_d^3 which is lacking in models like CVaR and expected risk. This shows the generalizability of CPT over CVaR with CPT having a richer modeling capability, thus validating the theoretical results from Lemma 3.



(a) Four paths $\{P_d^1, P_d^2, P_d^3, P_d^4\}$ are drawn in blue, orange, green and red respectively.



(b) Boxplots showing the cost(Area) returned after using SPSA with CPT and CVaR to capture risk profile of the drawn paths.

Figure 4.8. Result of using CPT and CVaR to model drawn paths.

4.9 Summary

In this Chapter, we have proposed a novel adaptation of CPT to model a DM's non-rational perception of a risky environment in the context of path planning. Firstly, using CPT, we provided a tuning knob to model various risk perceptions of an uncertain spatial cost. Next, we demonstrated a novel embedding of non-rational risk perception into a sampling based planner, the CPT-RRT*, to plan asymptotically optimal paths in perceived environments. Finally, we theoretically and empirically evaluated CPT as a good approximator to the risk perception of arbitrary drawn paths by comparing against CVaR, and show that CPT is a richer model approximator.

In the next chapter we will look at designing risk-perception-aware control strategies.

This chapter, in full, is a reprint of the material as it appears in the publication Planning under non-rational perception of uncertain spatial costs, A. Suresh and S. Martínez, IEEE Robotics and Automation Letters, 6(2), pp. 4133–4140, 2021. The dissertation author was the primary investigator and author of this paper.

Chapter 5

Risk-Perception-Aware Safe Control Design

This work proposes a novel Risk-Perception-Aware (RPA) control design using non-rational perception of risks associated with uncertain dynamic spatial costs. We use Cumulative Prospect Theory (CPT) to model the risk perception of a Decision Maker (DM) and use it to construct perceived risk functions that transform the uncertain dynamic spatial cost to deterministic perceived risks of a DM. These risks are then used to build safety sets which can represent risk-averse to risk-insensitive perception. Using these sets, we define novel notions of "inclusiveness" and "versatility" which can be employed to compare and evaluate any risk models in the context of RPA safety-critical controls. We then prove that CPT is the most "inclusive" and "versatile" model w.r.t. Conditional Value at Risk (CVaR) and Expected Risk (ER). Given a RPM, we construct a class of Control Barrier Functions (CBFs) and generate perceived-safety-critical controls using a Quadratic Program (QP) to guide an agent safely to a goal. For a class of truncated-Gaussian costs, we provide sufficient geometric conditions for the above QP to be feasible. We also prove that CPT-equipped RPA controller has both a larger feasible control set and more accurate stabilization w.r.t CVaR and ER models. We present simulations in a 2D environment to illustrate the proposed controller.

5.1 Literature Review

Safe control design has been tackled using various frameworks such as artificial potential functions [Kha90], barrier certificates [PJ04] and more recently, control barrier functions (CBFs) [ACE+19]. CBFs have gained popularity due to their Lyapunov-like properties, rigorous safety guarantees and ease of application. They have been successfully used in optimization [ACE+19], stabilization [OC19] and data-driven control frameworks [LSH21]. CBFs have traditionally been used for static obstacles. Recently, CBFs are employed to handle moving obstacles [CPG18] and multi-agent systems [GCE21, AXA21a], however they do not consider risk or uncertainty. Uncertainty is mainly handled using robustness measures [PJP07], stochastic control [Cla19], or chance constraints [KDFA20]. Very few works have considered the notion of risk perception explicitly in a control system [SY18, AXA21b]. The previous works use CVaR to quantify risk perception, which only captures rational risk-averse behavior. CPT on the other hand is a more expressive (see [SM21a]), non-linear and non-rational perception theory which is yet to be applied in the context of safety for a control system. Moreover, CPT has been successfully used in engineering applications like path planning [SM21a], traffic routing [GFBA10], and network protection [HS19].

5.2 Contributions

We first show how to employ non-rational risk-perception notions to express perceived safety for control systems, expanding the spectrum of risk perception models considered in the literature. We formally introduce the novel concepts of "inclusiveness" and "versatility" to compare and contrast the ability of different models to handle a variety of DMs' risk profiles. We consider two popular models, CVaR, and ER, and prove that CPT is both more "inclusive" and "versatile" than these. Next, we construct

a class of CBFs which guarantee safety according to a DM's perceived risk, corresponding to any RPM. Using a QP formulation, we obtain Risk-Perception-Aware (RPA) controls that can guide an agent to a desired goal while maintain perceived safety. We analyze feasibility conditions and stability properties of the proposed approach in terms of the three RPMs, and prove that CPT equipped RPA controller has both a larger feasible control set and more accurate stabilization. We illustrate our results through simulations and animations in a 2D environment.

5.3 Risk perception formalism and Problem Statement

Here, we introduce notation[1], formalize risk perception, and briefly describe CPT and CVaR (see [Dha16] and [RU00] for more details). Later, we describe our problem statement.

Risk Perception: By risk perception, we refer to the notion of attaching a value (risk) to a random cost output. Formally, let $\mathscr{S}$ be a discrete sample space endowed with a probability distribution $\mathbb{P}$. We model environmental cost via a discrete random variable (RV) $c : \mathscr{S} \to \mathbb{R}_{\geq 0}$, taking M values, $c_i \in \mathbb{R}_{\geq 0}$, $i \in \{1, \ldots, M\}$, and such that $p_i = \mathbb{P}(c = c_i)$, with $\sum_i^M p_i = 1$. We let $\mathscr{C}$ be the set of such RVs and $R : \mathscr{C} \to \mathbb{R}_{\geq 0}$ a value function which associates a value (risk) to a RV.

A value function R can be defined in many ways, resulting in different risk perceptions. Here, an RPM (or, briefly, a model) is characterized as a parameterized family $\mathscr{M} \triangleq \{R_\Theta | \Theta \in \mathbb{R}^l\}$ of value functions. In what follows, we consider three popular RPMs: Expected RiskER,CVaR[2] [RU00] and CPT [TK92]. CPT captures non-rational decision making, and was introduced in [Dha16, D.P98]. In CPT, outcomes are first

[1]The Euclidean norm in $\mathbb{R}^n$ is denoted by $\|.\|$. We use $\mathbb{E}$ as the expectation operator of a random variable. The set $\mathbf{B}^r(y) \triangleq \{x \in \mathscr{X} | \|x - y\| \leq r\}$ is a ball of radius r centered at y.

[2]The CVaR model uses a class of value functions parameterized by $q \in [0,1]$ to represent expectation over a fraction (q) of the worst-case outcomes. Thus the CVaR value with $q = 1$ is the worst-case outcome of c, c_M. While, with $q = 0$ CVaR value equals ER ($R_0^{\text{cv}} = R^{\text{e}}$).

weighed using a non-linear utility function $v : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, with $v(c) = \lambda c^{\gamma}$, modeling a DM's perceived cost. The parameters $\lambda \in [1, \infty), \gamma \in [0, 1]$ represent "risk aversion" and "risk sensitivity", respectively. In addition, a non-linear probability weighing function $w : [0, 1] \to [0, 1]$, given by $w(p) = e^{-\beta(-\log p)^{\alpha}}$ and $w(0) = 0$, is used to model uncertainty perception. Here, uncertainty sensitivity is tuned via the parameters $\alpha, \beta \in \mathbb{R}_{>0}$. CPT also suggests that probabilities are perceived via decision weights $\Pi_i \in [0, 1]$, which are calculated in a cumulative fashion with $\Pi_i = w(\sum_{j=i}^{M} p_j) - w(\sum_{j=i+1}^{M} p_j)$ and $\Pi_M = w(p_M)$.

With this, assigning the parameter q for CVaR and $\theta = \{\alpha, \beta, \gamma, \lambda\}$ for CPT, the value functions of ER (R^{er}), CVaR (R^{cv}) and CPT (R^{cpt}) of a DM are defined as:

$$R^{\text{er}}(c) \triangleq \mathbb{E}(c) = \sum_{i=1}^{M} c_i p_i, \tag{5.1a}$$

$$R_q^{\text{cv}}(c) \triangleq \mathbb{E}\left[c | c \geq \min\{d : \mathbb{P}(c \leq d) \geq q\}\right], \tag{5.1b}$$

$$R_{\theta}^{\text{cpt}}(c) \triangleq \sum_{j=1}^{M} v(c_j)\Pi_j. \tag{5.1c}$$

In CPT, θ can be varied to generate different value functions pertaining to various risk profiles of DMs (from risk-taker to risk-averse). We refer to [SM21a, Dha16] for more details on the parameter choices in CPT.

Risky Environment: Consider a compact state space $\mathscr{X} \subset \mathbb{R}^n$ containing dynamic spatial sources of risk at $y \in \mathscr{X}$ and an agent or robot at a state $x \in \mathscr{X}$. The relative state space is $\mathscr{Z} \triangleq \{\xi = y - x | x \in \mathscr{X}, y \in \mathscr{X}\}$. Our starting point is a discrete random field cost $c : \mathscr{Z} \to \mathbb{R}_{\geq 0}$, that aims to quantify objectively the (negative) consequences of being at $x \in \mathscr{X}$ relative to a known risk source at $y \in \mathscr{X}$. More precisely, $c(\xi)$ is a discrete RV which takes values $c_i(\xi) \in \mathbb{R}_{\geq 0}$, for $i \in \{1, \ldots, M\}$. We assume that c has associated mean and standard deviation functions $c_{\mu} : \mathscr{Z} \to \mathbb{R}_{\geq 0}$ and $c_{\sigma} : \mathscr{Z} \to \mathbb{R}_{\geq 0}$, respectively. We assume that c_{μ}, c_{σ} are continuously differentiable in their domains. Given c, an associated spatial-risk function is given by $R_c : \mathscr{Z} \to \mathbb{R}_{\geq 0}$, $R_c(\xi) \equiv R(c(\xi))$,

where R belongs to any of the previous RPMs defined in (5.1) above. When clear from the context, we will identify $R_c \equiv R \in \mathscr{M}$. The larger R_c is at ξ, the higher the perceived risk of being at $x \in \mathscr{X}$.

Dynamic Systems: We aim to control an agent modeled as a control-affine dynamic system:

$$\dot{x} = f_x(x,u) = f(x) + G(x)u, \tag{5.2}$$

where $u \in \mathbb{R}^m$, $G : \mathscr{X} \to \mathbb{R}^{n\times m}$, $f : \mathscr{X} \to \mathbb{R}^n$ and f and G are locally Lipschitz. We also consider a dynamic risk

$$\dot{y} = f_y(y),\ y \in \mathscr{X},\ f_y : \mathscr{X} \to \mathbb{R}^n, \tag{5.3}$$

with a locally Lipschitz f_y. We focus on moving obstacles as the source of risk, but the approach can be extended to other scenarios. We also assume that a asymptotically stable controller $k : \mathscr{X} \to \mathbb{R}^m$ has been designed to guide the agent to a goal state $x^* \in \mathscr{X}$ in the absence of risk sources.

Problem Statement: We wish to drive the agent to a goal $x^* \in \mathscr{X}$ safely, while avoiding risky areas. Formally, we define safety considering a perceived spatial risk function R_c:

Definition 3. (Perceived Safety) An agent moving under (5.2), and subject to an uncertain cost source c with dynamics (5.3), is said to be safe w.r.t. the perceived risk R_c iff $R_c(\xi(t)) \leq \rho$, $\forall t \geq 0$, for some tolerance $\rho \in \mathbb{R}_{>0}$.

We now state the problems we address in this work:

Problem 9. (RPA safe sets) Given a risky environment $\mathscr{X}$, endowed with an uncertain cost c, design perceived safety sets considering models from (5.1). Characterize and contrast the properties of these sets among the three models.

Problem 10. (RPA safe controls) Under previous conditions, design a controller u, nominally deviating from a stable state feedback controller k, such that the agent reaches the goal x^* safely (Definition 3) and examine feasibility of u.

5.4 Perceived Safety using various RPMs

Here, we address Problem 9 by defining two properties: "Inclusiveness" and "Versatility", which will be used to compare and contrast ER, CVaR and CPT models.

Fix a cost field c, and a safety level $\rho \in \mathbb{R}_{\geq 0}$, and define:

$$\mathscr{X}_{\text{safe}}(R;y) = \{x \in \mathscr{X} \,|\, R(y-x) \leq \rho\}, \tag{5.4a}$$

$$\mathscr{X}_{\text{risky}}(R;y) = \{x \in \mathscr{X} \,|\, R(y-x) > \rho\}, \tag{5.4b}$$

where $\mathscr{X}_{\text{safe}}(R;y)$ (or $\mathscr{X}_{\text{risky}}(R;y)$) indicates the set of all states x which are perceived safe (or risky) w.r.t a model $\mathscr{M}$ with value function R, when a risk source is at y. Given $\mathscr{M}$, let $\mathscr{R}_{\mathscr{M}}(c) \equiv \mathscr{R}_{\mathscr{M}} \subset \mathbb{R}_{\geq 0}$ denote[3] the set of all possible perceived risk values of c. Similarly, let the total safe set $\mathscr{Y}_{\mathscr{M}}(y,c) \equiv \mathscr{Y}_{\mathscr{M}}$ and total risky set $\overline{\mathscr{Y}}_{\mathscr{M}}(y,c) \equiv \overline{\mathscr{Y}}_{\mathscr{M}}$ denote[4] all the states in $\mathscr{X}$ that can be considered safe (resp. risky) according to a model $\mathscr{M}$ when a risk source is at y.

Definition 4. (Inclusiveness and Strict Inclusiveness). Consider two models $\mathscr{M}_1$ and $\mathscr{M}_2$, a threshold $\rho \in \mathbb{R}_{>0}$, a cost field c, and a risk source at $y \in \mathscr{X}$. Let the sets $\mathscr{Y}_1, \overline{\mathscr{Y}}_1$ and $\mathscr{Y}_2, \overline{\mathscr{Y}}_2$ be the total safe and risky sets respectively of $\mathscr{M}_1$ and $\mathscr{M}_2$ wrt y and a spatial cost c. We say that $\mathscr{M}_1$ is more inclusive than $\mathscr{M}_2$ ($\mathscr{M}_1 \rhd \mathscr{M}_2$) if either $\overline{\mathscr{Y}}_2 \subseteq \overline{\mathscr{Y}}_1$ and $\mathscr{Y}_2 \subsetneq \mathscr{Y}_1$ holds, or $\overline{\mathscr{Y}}_2 \subsetneq \overline{\mathscr{Y}}_1$ and $\mathscr{Y}_2 \subseteq \mathscr{Y}_1$ holds, for all $y \in \mathscr{X}$ and costs $c : \mathscr{X} \to \mathbb{R}_{\geq 0}$. If both the above conditions simultaneously hold, then $\mathscr{M}_1$ is strictly more inclusive than $\mathscr{M}_2$ ($\mathscr{M}_1 \blacktriangleright \mathscr{M}_2$).

[3] $\mathscr{R}_{\mathscr{M}}(c) \triangleq \{r \in \mathbb{R} | r = R_\Theta(c), \forall R_\Theta \in \mathscr{M}\}$.

[4] $\mathscr{Y}_{\mathscr{M}}(y,c) \triangleq \bigcup_{R \in \mathscr{M}} \mathscr{X}_{\text{safe}}(R;y)$, $\overline{\mathscr{Y}}_{\mathscr{M}}(y,c) \triangleq \bigcup_{R_c \in \mathscr{M}} \mathscr{X}_{\text{risky}}(R;y))$.

The above definition implies that, if $\mathscr{M}_1 \triangleright \mathscr{M}_2$, then $\mathscr{M}_1$ has the ability to perceive a wider range of both safe and risky sets for a given environment than $\mathscr{M}_2$.

Now we compare the inclusiveness of CPT, CVaR and ER via their respective value functions. We first identify conditions for strict inclusiveness (Lemma 6) and compare the range space of these models (Lemma 7), which can be found in the Appendix. Based on these, the following theorem compares inclusiveness among the three models.

Theorem 2. (Inclusiveness Comparison) Let c be a discrete random field. Consider ER, CVaR and CPT models with value functions R^{er}, R_q^{cv}, and $R^{\text{cpt}}{}_\theta$, respectively. For any threshold $\rho \in \mathbb{R}_{\geq 0}$ and risk source at $\bar{y} \in \mathscr{X}$, CPT is more inclusive than CVaR and ER. If cost outcomes $c_i > 1$, $\forall i$, then CPT is strictly more inclusive than CVaR and ER. Additionally if $c_\sigma(\bar{y} - x) > 0, \forall x \in \mathscr{X}$, then CVaR is more inclusive than ER.

Proof. From Lemma 7, $\mathscr{R}_{\text{er}} \subsetneq \mathscr{R}_{\text{cpt}}$ and $\mathscr{R}_{\text{CVaR}} \subseteq \mathscr{R}_{\text{cpt}}$. As in Lemma 7, take $R^{\text{cpt}}{}_{\theta_1} = \lambda c_\mu$, for some θ_1. Choosing $\lambda = \bar{b}$, with $\bar{b} > \frac{b}{c_\mu}$, we get $R^{\text{cpt}}{}_{\theta_1} > R_q^{\text{cv}}$, for any $q \in [0,1]$, and $R^{\text{cpt}}{}_{\theta_1} > R^{\text{er}}$. Thus, from Lemma 7, we have CPT $\triangleright$ CVaR and CPT $\triangleright$ ER. With $c_i > 1$ for all i, taking $\theta_1 = \{1,1,1,\lambda\}$, with $\lambda > 1$, we get $R^{\text{cpt}}{}_{\theta_1} > R_q^{\text{cv}}$ for any $q \in [0,1]$ and $R^{\text{cpt}}{}_{\theta_1} > R^{\text{er}}$. Now, take $\theta_2 = \{1,1,\gamma,1\}$, with $0 < \gamma < 1$, we have $R^{\text{cpt}}{}_{\theta_2} = \sum_i c_i^\gamma p_i$. Since $c_i, p_i > 0$, $\forall i$, then $R^{\text{cpt}}{}_{\theta_2} < \sum_i c_i p_i$, implying $R^{\text{cpt}}{}_{\theta_2} < c_\mu$ and $R^{\text{cpt}}{}_{\theta_2} < R_q^{\text{cv}}$, $\forall q \in [0,1]$. From Lemma 6, we get CPT $\blacktriangleright$ CVaR and CPT $\blacktriangleright$ ER.

Finally, note that by definition R_q^{cv} is monotonically increasing in q and lies between $c_\mu = R^{\text{er}}$ and $c_{\max}$. Since $c_\sigma > 0$, there exits a $q \in (0,1)$ such that $R_q^{\text{cv}} > R^{\text{er}}$, but there is no q s.t. $R_q^{\text{cv}} < R^{\text{er}}$. Hence from Lemma 6 and the first part of this result, we get CPT $\blacktriangleright$ CVaR $\triangleright$ ER. $\square$

Theorem 2 implies that CPT can produce a larger variety of safe and risky sets perceived from any given environment and a risk source, leading to richer risk perception. This is illustrated via simulations in Section 5.5 and Section 5.6.

5.4.1 Versatility of Risk Perception Models

We now define the concept of versatility of a model in the context of perceived safety.

Definition 5. (Versatility of a RPM). Consider a compact space $\mathscr{X}$, a risk source $\bar{y} \in \mathscr{X}$, and a discrete random field cost c, with range in $[c_{\min}, c_{\max}] \subseteq \mathbb{R}_{\geq 0}$. Let $\bar{I}$ be a compact interval. A model $\mathscr{M}$ is said to be $\bar{I}-$versatile if $\{x \in \mathscr{X} \,|\, c(\bar{y}-x) \leq c_\ell\} \subseteq \mathscr{Y}_{\mathscr{M}}$ for any $c_\ell \in \bar{I}$ for a given $\rho > 0$. If $\bar{I} \supseteq [c_{\min}, c_{\max}]$, then $\mathscr{M}$ is most versatile in $\mathscr{X}$.

The above definition implies that an RPM is $\bar{I}-$versatile, if it has a risk-perception function that perceives any states having costs less than c_ℓ as safe, $\forall\, c_\ell \in \bar{I}$. Further, $\mathscr{M}$ is most versatile when it contains risk-perception functions that capture a range of perceptions from most risk averse (only states having costs $c \leq c_{\min}$ are safe) to the least risk-sensitive (every state including states having the highest cost $c_{\max}$ as safe). Now we look at versatility of the three models.

Theorem 3. (Versatility Comparison) Consider a compact space $\mathscr{X}$, with a risk source $\bar{y} \in \mathscr{X}$, and associated discrete random field cost c with $c_{\min} \geq 1$ and a finite $c_{\max}$. CPT is the only most versatile model in $\mathscr{X}$ among CPT, ER and CVaR. With $\bar{I}_1 = [c_\mu, c_{\max}]$ and $\bar{I}_2 = \{c_\mu\}$, CVaR is $\bar{I}_1-$versatile and ER is $\bar{I}_2-$versatile.

Proof. First, we show CPT captures most risk averse perception, i.e. the set $\{x \in \mathscr{X} | c(\bar{y}-x) \leq c_{\min}\}$ is considered safe. By choosing θ as in Lemma 6 and $\lambda = \frac{\rho}{c_{\min}}$, this result follows from (5.4a) and Definition 5. Next, we show CPT can capture the least risk sensitive perception (the set $\{x \in \mathscr{X} | c(\bar{y}-x) \leq c_{\max}\}$ is considered safe). Let $\theta_2 = \{1, 1, \gamma, 1\}$, with $0 \leq \gamma \leq 1$ we have ${R^{\text{cpt}}}_{\theta_2}(c) = \sum_i c_i^\gamma p_i$. Choosing $\gamma < \frac{\log \rho}{\log c_{\max}}$, since $c_i \geq 1, p_i \geq 0$, $\forall i$, and $\rho \geq 1$, we get ${R^{\text{cpt}}}_{\theta_1} \leq \rho$. Thus, from (5.4a) and Definition 5, this result follows.

Now we show CPT is most versatile. With $\theta_1 = \{1, 1, 1, \lambda\}$ and θ_2. Observe that ${R^{\text{cpt}}}_{\theta_1}$ is continuous in λ and ${R^{\text{cpt}}}_{\theta_2}$ is continuous in γ. By the intermediate value theorem $\exists \lambda$ s.t. ${R^{\text{cpt}}}_{\theta_1} \in [c_\mu, c_{\max}]$, and a γ s.t. ${R^{\text{cpt}}}_{\theta_2} \in [c_{\min}, c_\mu]$. Hence, from above arguments, CPT is most versatile in $\mathscr{X}$. The last result with ER and CVaR trivially follows from the range spaces $\mathscr{R}_{\text{CVaR}}$ and $\mathscr{R}_{\text{er}}$ in the proof of Lemma 7. □

The above result illustrates that given any environment with a random cost c, out of the three models, CPT can perceive a larger range of costs as safe/risky and hence is "more versatile" among the others. Next, we will look at control design with perceived safety.

5.5 Control design with Risk-Perception-Aware-CBFs

Here, we address Problem 10 and design controls u for an agent subject to (5.2), to ensure perceived safety (Definition 3). To do this, we formally adapt CBFs (see [ACE+19]) to our setting.

Definition 6 (RPA-CBF). Consider an agent subject to (5.2), a dynamic source of risk (5.3), and a perceived risk R model. A $\mathscr{C}^1$ function $h_R \triangleq h(R) : \mathscr{Z} \to \mathbb{R}$ is an RPA-CBF for this system, if there is an extended class $\mathscr{K}_\infty$ function η_1 such that the control set K_{R} defined as

$$K_{\text{R}}(R) = \{u \in \mathscr{U} \,|\, \dot{h}_R(\xi) \geq -\eta_1(h_R(\xi)))\}, \tag{5.5}$$

is non-empty for all $\xi \in \mathscr{Z}$.

The existence of h_R according to Definition 6 implies that the super-level set $\{x \in \mathscr{X} \,|\, h_R(\xi) > 0\}$ is forward invariant under (5.2). We specify $h_R = h(R)$ via h given as

$$h(\xi) \triangleq \eta_2(\rho - R(\xi)), \tag{5.6}$$

where $\eta_2 : \mathbb{R} \to \mathbb{R}$ is a $\mathscr{C}^1$ extended $\mathscr{K}_\infty$ function. Since η_2 is non-decreasing, $h(R(\xi)) \geq 0$ implies $R(\xi) \leq \rho$ and thus, x is perceived safe ($x \in X_{\text{safe}}$) w.r.t. R when risk source is at y.

The RPA control input u can be now computed via:

$$u(x) = \underset{u}{\text{argmin}} \ \|u - k(x)\|^2 \tag{5.7a}$$

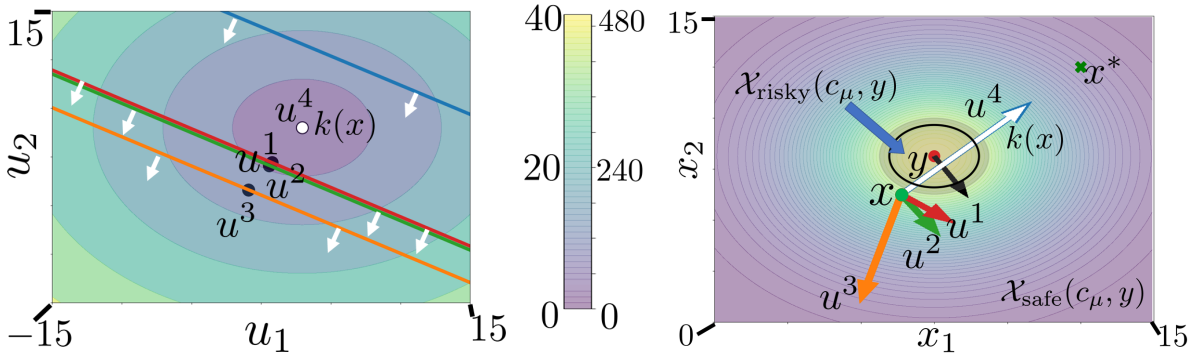
$$\text{s.t.} \quad \frac{d\eta_2}{dR}\left(\frac{\partial R}{\partial \xi}(\xi)\right) \cdot (f_y(y) - f_x(x,u)) \geq -\eta_1(h_R(\xi)). \tag{5.7b}$$

The above problem captures the notion of minimally modifying a stable controller to ensure safety of the system. Note that (5.7) is for a fixed time t thus, risk and its derivatives are only evaluated at the current x, y. Considering a fully actuated linear agent in 2D, we visualize the optimization landscape ($\mathscr{U}$) and the state space $\mathscr{X} \subset \mathbb{R}^2$ in Fig. 5.1. We note that depending on the choice of model, (5.7b) varies as shown in Fig. 5.1a. We use identity function for both η_1, η_2 in Fig.5.1. These functions affect constraint (5.7b) (lines in Fig. 5.1a) and can be designed according to practical considerations like frequency of implementation and performance near safe set boundary. From Fig. 5.1a, u_4 and $k(x)$ coincide as R^{cpt} with $\gamma = 0.8$ is insensitive enough to ignore the obstacle. On the flip side, u_3 is almost in the opposite direction of $k(x)$, indicating an extreme aversion. The values of u_1 and u_2 (corresponding resp. to R^{er}, risk neutral R^{cv}, and highly risk averse R^{cv}) produce almost similar and more moderate controls. This indicates CPT can produce a wider range of controls and a larger feasible control set. We theoretically analyze and justify this claim in the following paragraphs.

5.5.1 Feasibility analysis and comparison

We first construct finite outcomes of c from c_μ and c_σ called "truncated-Gaussian cost" which will be used for analysis. In this construction[5], we approximate c by means

[5]This truncation reassigns the probability mass s.t. $c(\xi) \in [c_\mu(\xi) - 3c_\sigma(\xi), c_\mu(\xi) + 3c_\sigma(\xi)]$ using an appropriate re-normalization constant.



(a) Optimization space $\mathscr{U}$ with objective $\|u-k(x)\|$ as a contour map.

(b) State space $\mathscr{X}$ with Expected risk R^{er} as a contour map.

Figure 5.1. (a) control space $\mathscr{U} \subset \mathbb{R}^2$ and (b) state space $\mathscr{X} \subset \mathbb{R}^2$ at a time instant t for fully actuated ($\dot{x} = u$) agent x (green dot) heading towards goal x^* (green cross) and obstacle y (red dot, imperfectly localized in grey disc) with $\dot{y}$ (black arrow). Contour maps (yellow means higher cost) indicate $\|u-k(x)\|$ in (a) and R^{er} in (b). Controls u^1(red: R^{er}), u^2(green: R^{cv} with $q=0.99$) and u^3, u^4(orange,blue: R^{cpt} with $\lambda = 5.0$ for u^3 and $\gamma = 0.8$ for u^4, with other parameters fixed at 1.0) are obtained by solving (5.7). In (a), the constraint (5.7b) is visualized as a line (color coded according to models) with respective optimal u as points (black dots). Area by white arrows indicate the feasibility region. In (b), optimal u are visualized as respective vectors. Considering R^{er} and obstacle at y, agent's safe ($\mathscr{X}_{\text{safe}}$) and risky ($\mathscr{X}_{\text{risky}}$) areas are shown separated by boundary $R^{\text{er}} = \rho$ (black circle).

of M discrete values c_i, $i \in \{1,\dots,M\}$, with probability calculated by $p_1 = F(c_1)$, and $p_i = F(c_i) - F(c_{i-1})$, for $i \in \{2,\dots,M\}$, where F is the CDF of c.

Now, we show conditions on u for the set K_{R} to be non-empty for a given risk function R. We first define a few constants to help us compare the feasibility conditions of the three models. Let $\phi_\xi \in [-\pi, \pi]$ be the relative angle between $\frac{\partial R}{\partial \xi}$ and $\dot{\xi}(u;x,y)$[6], and $c'_\mu = \frac{dc_\mu}{d\xi}$ and $c'_\sigma = \frac{dc_\sigma}{d\xi}$. Now define $k^{\text{er}}(\xi) = \eta_1(R^{\text{er}}(\xi))/\frac{d\eta_2}{dR^{\text{er}}}$, $k^{\text{cv}}_q(\xi) = \eta_1(R^{\text{cv}}_q(\xi))/\frac{d\eta_2}{dR^{\text{cv}}_q}$ and $k^{\text{cv}}_\sigma = \frac{\mathbb{P}(F^{-1}(q))}{q}$. Also let $k^{\text{cpt}}_\theta(\xi) = \eta_1(R^{\text{cpt}}{}_\theta(\xi))/\frac{d\eta_2}{dR^{\text{cpt}}{}_\theta}$, $k^{\text{cpt}}_\mu = \lambda\gamma\sum_{i=1}^{M} c_i^{\gamma-1}\Pi_i$ and $k^{\text{cpt}}_\sigma = \lambda\gamma\sum_{i=1}^{M}\left(3-\frac{6i}{M}\right)(c_i)^{\gamma-1}\Pi_i$. Consider $\eta^{\text{er}} = \frac{k^{\text{er}}(\xi)}{\|c'_\mu\|}$, $\eta^{\text{cv}} = \frac{k^{\text{cv}}_q(\xi)}{\|c'_\mu + k^{\text{cv}}_\sigma c'_\sigma\|}$ and $\eta^{\text{cpt}} = \frac{k^{\text{cpt}}_\theta(\xi)}{\|k^{\text{cpt}}_\mu c'_\mu + k^{\text{cpt}}_\sigma c'_\sigma\|}$. The following holds.

Proposition 4 (Feasibility). Let an agent and risk source be subject to (5.2) and (5.3), respectively. Consider a truncated Gaussian cost c . Given any x, y, if there is a u and

[6]The angle between two vectors $a, b \in \mathbb{R}^n$ is $\phi = \cos^{-1}\left(\frac{a\cdot b}{\|a\|\|b\|}\right)$. We assume $\frac{dc_\mu}{d\xi}$ and $\frac{dc_\sigma}{d\xi}$ are aligned. Hence $\frac{\partial R}{\partial \xi}$ is also aligned with $\frac{\partial c}{\partial \xi}$.

η_1 s.t.:

$$\|\dot{\xi}(u;x,y)\|\cos(\phi_\xi) \geq -\left(\frac{\eta_1(h_R(\xi))}{\frac{d\eta_2}{dR}\left\|\frac{\partial R}{\partial \xi}(\xi)\right\|}\right), \tag{5.8}$$

then h_R defined according to (5.6) is a valid RPA-CBF for any $\mathscr{C}^1$ and extended $\mathscr{K}_\infty$ function η_2, and (5.7) is feasible. Specifically, the RHS of the above inequality reduces to $-\eta^{\text{er}}$, $-\eta^{\text{cv}}$, and $-\eta^{\text{cpt}}$ for ER, CVaR and CPT, respectively.

Proof. For first part, rearranging terms in (5.8) we get:

$$\kappa\left\|\frac{\partial R}{\partial \xi}(\xi)\right\| \cdot \left\|f_y(y) - f_x(x,u)\right\|\cos(\phi_\xi) \geq -\eta_1(h_R(\xi)), \tag{5.9}$$

where $\kappa = \frac{d\eta_1}{dR}$. From (5.9) and the fact that LHS is time derivative of $h_R(\xi)$ (using the chain rule) we have $\dot{h}_R(x,u;y) \geq -\eta_1(h_R(\xi))$. The result follows from Definition 6. For the last part, the expressions are obtained by substituting the respective risk functions and evaluating the partial derivatives $\frac{\partial R}{\partial c_\mu}$ and $\frac{\partial R}{\partial c_\sigma}$ (part of $\frac{\partial R}{\partial \xi}$). For ER, these calculations are trivial. For CVaR and CPT, the derivatives w.r.t c_μ and c_σ are easy to evaluate with truncated Gaussian costs. □

From (5.8), the RHS is independent of u and the LHS is independent of R and the model. This separation makes it easier to compare feasibility conditions of various models.

Remark 5 (Satisfying Feasibility). Constraint (5.7b) is linear in u as $f_x(x,u)$ is affine in u from (5.2). Thus (5.7) has a closed form solution, making it computationally efficient. Furthermore, (5.8) gives the sufficient conditions on u to satisfy (5.7b) assuming truncated Gaussian costs. So if $x(0) \in \mathscr{X}_{\text{safe}}$ and if (5.8) can be satisfied throughout with an appropriate u and η_1, then the agent's entire trajectory will be perceived safe.

Remark 6 (Uncertainty perception among RPMs). The ER model is insensitive to uncertainty as $\frac{\partial R^{\text{er}}}{\partial \sigma} = 0$. In this way, CVaR is averse to uncertainty as $\frac{\partial R_q^{\text{cv}}}{\partial \sigma} \geq 0$ for all q. With

CPT, θ can be tuned to get both uncertainty insensitive and uncertainty averse behavior, additionally, it can also produce uncertainty liking behavior (when $\frac{\partial R^{\text{cpt}}{}_{\theta}}{\partial \sigma} \leq 0$)[7].

Next, we compare the size of control sets K_{R} respectively generated by each of the RPMs.

Proposition 5 (Feasibility set comparison). Assume conditions of Proposition 4 hold. Then, the feasibility sets defined according to (5.5) satisfy: $K_{\text{er}} \subseteq K_{\text{cpt}}$ and $K_{\text{cv}} \subseteq K_{\text{cpt}}$.

Proof. The comparison is established on the model's respective feasibility conditions (5.8). Since the LHS in (5.8) remains the same for any model and parameter choice, it is sufficient to show that $\eta^{\text{er}} \leq \eta^{\text{cpt}}$ and $\eta^{\text{cv}} \leq \eta^{\text{cpt}}$. In that respect, uncertainty perception (behavior of $\frac{\partial R}{\partial \sigma}$) is particularly important. These inequalities follow from the choice of $\theta = \theta_1$ in the proof of Theorem 2 and CPT's more adaptable uncertainty perception from Remark 1. □

It is interesting to note that although CVaR is more inclusive than ER as proved in Theorem 2, it does not immediately translate into CVaR having a larger control feasibility set. This due to the additional $k_{\sigma}^{\text{cv}} \frac{dc_{\sigma}}{d\xi}(\xi)$ term in the denominator of (5.8) for CVaR. We elaborate in the following remark:

Remark 7. Consider the control feasibility sets K_{ER} and K_{CVaR} respectively for ER and CVaR, defined according to (5.5). Then, depending on the choice of q and construction of c_{σ} we can obtain either $K_{\text{ER}} \subseteq K_{\text{CVaR}}$ or $K_{\text{CVaR}} \subseteq K_{\text{ER}}$. Although we have $k_q^{\nu}(\xi) >$

[7]The first two properties follow by choosing θ as in Theorem 2. The latter property can be obtained by tuning the uncertainty perception parameters α and β. Since the chosen distribution is symmetric, we can examine the relation between Π_i and Π_{M-i} for $i \in \left(0, \frac{M}{2}\right)$. If we have $\Pi_i < \Pi_{M-i}$ (for example when w is concave) or $\Pi_i > \Pi_{M-i}$ (when w is convex), then we have $\frac{\partial R^{\text{cpt}}}{\partial c_{\sigma}} > 0$, or $\frac{\partial R^{\text{cpt}}}{\partial c_{\sigma}} < 0$, respectively. A concave w ($\alpha = 1, \beta < 1$) implies that unlikely outcomes are viewed to be more probable compared with the more certain outcomes. This results into an "uncertainty averse behavior", which is reflected in the positive sign of $\frac{\partial R^{\text{cpt}}}{\partial c_{\sigma}}$. Conversely, a convex w ($\alpha = 1, \beta > 1$) leads to an "uncertainty liking behavior" with $\frac{\partial R^{\text{cpt}}}{\partial c_{\sigma}} < 0$.

$k^e(\xi)$ from Theorem 2, there isn't conclusive proof to suggest $K_{\text{ER}} \subseteq K_{\text{CVaR}}$ due to the additional $k_\sigma^v \frac{dc_\sigma}{d\xi}(\xi)$ term in the denominator of (5.8).

Stability Analysis: Next, let us look at the stability properties of the proposed controller u in (5.7). It is clear that if the nominal controller $k(x)$ also satisfies the safety constraint (5.7b), then $u = k(x)$ and the stability properties of $k(x)$ transfer over to u. First we look into the models and determine how they affect the deviation from $k(x)$. Later, we treat the controller u as a perturbed version of $k(x)$ and analyze accordingly.

Let $\delta = k(x) - u(x)$ be the perturbation to the nominal controller $k(x)$ and δ^{er}, δ_q^{cv} and $\delta^{\text{cpt}}{}_\theta$ be the respective perturbations of ER, CVaR and CPT with corresponding parameter choices. Then we have the following:

Proposition 6 (Stability Comparison). Under the assumptions of Proposition 4, choose u as in (5.7). Assume $\|\delta^{\text{er}}\|$, $\|\delta_q^{\text{cv}}\|$ and $\|\delta^{\text{cpt}}{}_\theta\|$ are bounded. Then for any given states x, y, and choice of q, there exists a θ such that:

1. $\|\delta^{\text{cpt}}{}_\theta\| \leq \|\delta^{\text{er}}\|$ and $\|\delta^{\text{cpt}}{}_\theta\| \leq \|\delta_q^{\text{cv}}\|$.
2. The agent stabilizes inside $\mathbf{B}^{\varepsilon^{\mathscr{M}}}(x^*)$ asymptotically for all models $\varepsilon^{\text{cpt}} \leq \varepsilon^{\text{cv}}$ and $\varepsilon^{\text{cpt}} \leq \varepsilon^{\text{er}}$ holds.

Proof. For 1), apply Proposition 5 and the fact that $K_{\text{er}} \subseteq K_{\text{cpt}}$ and $K_{\text{cv}} \subseteq K_{\text{cpt}}$.

For 2), employ an ISS argument to construct the $\mathbf{B}^{\varepsilon^{\mathscr{M}}}(x^*)$ for each model considering the unforced system with $u = k(x)$ in (5.2) and $P(x) = G(x)\delta$ being the forcing term after applying RPA controls u from (5.7). From ISS, since the radius of the stability ball is proportional to the upper bound on $\|P(x)\|$, the result immediately follows from the first part. □

Proposition 6 implies that, with an appropriate θ, CPT can not only produce the least perturbation among the three models, but can also stabilize to the smallest ball around x^*.

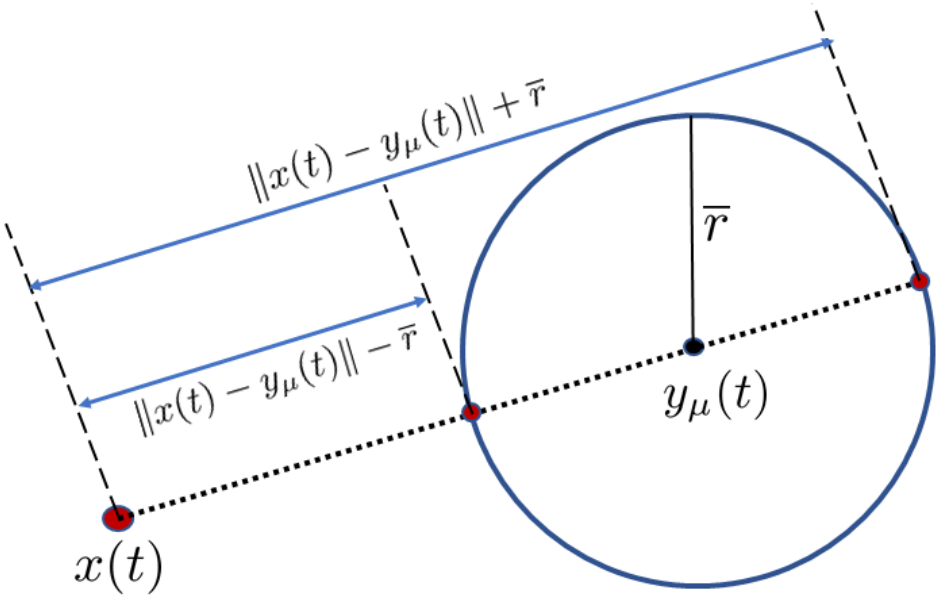


Figure 5.2. Illustration of simulation setting and measuring DTE for an agent at $x(t)$, facing an obstacle which is localized imperfectly in a circle of radius $\bar{r}$ and centered at $y_\mu(t)$.

5.6 Simulation Results

Here, we demonstrate the results from Section 5.4 and 5.5, considering a 2D environment with moving obstacles. Uncertain Cost: We consider a single agent with unicycle dynamics and a single obstacle whose known dynamics evolve in the space $\mathscr{X} \subset \mathbb{R}^2$. We assume the obstacle is imperfectly localized in a ball of radius $\bar{r}$ centered at $y_\mu \in \mathscr{X}$, i.e , $y \in \mathbf{B}^{\bar{r}}(y_\mu)$[8]. With this, the relative vector $\xi \in \mathbf{B}^{\bar{r}}(y_\mu - x)$. We use the notion of "distance to endangerment (DTE)", $d : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$, $d(\xi) \triangleq \|\xi\|$ to construct the uncertain cost c. From this, we obtain $d \in [\|x - y_\mu\| - \bar{r}, \|x - y_\mu\| + \bar{r}]$ (visualized in Figure 5.2). Define the cost $c(\xi) = k_1 \mathrm{e}^{-k_2 d(\xi)^2}$, representing the cost of being at x, knowing the obstacle $y \in \mathbf{B}^{\bar{r}}(y_\mu)$, with constants $k_1, k_2 > 0$. Assume that c is distributed as a truncated Gaussian (Section 5.4) with $c_\mu(\xi) = k_1 \mathrm{e}^{-k_2 d_\mu^2(\xi)}$ and $c_\sigma(\xi) = c_\mu(\bar{r}) p^{\mathscr{N}}(\xi, \mathbf{I})$[9], where $d_\mu = \|x - y_\mu\|$ and $\mathbf{I}$ is the 2D identity matrix. We construct uncertain cost

[8] W.l.o.g. this assumption also allows us to consider obstacles with a size.

[9] $p^{\mathscr{N}}(\mu, \Sigma)$ is the pdf of a bi-variate Normal distribution with mean μ and covariance Σ

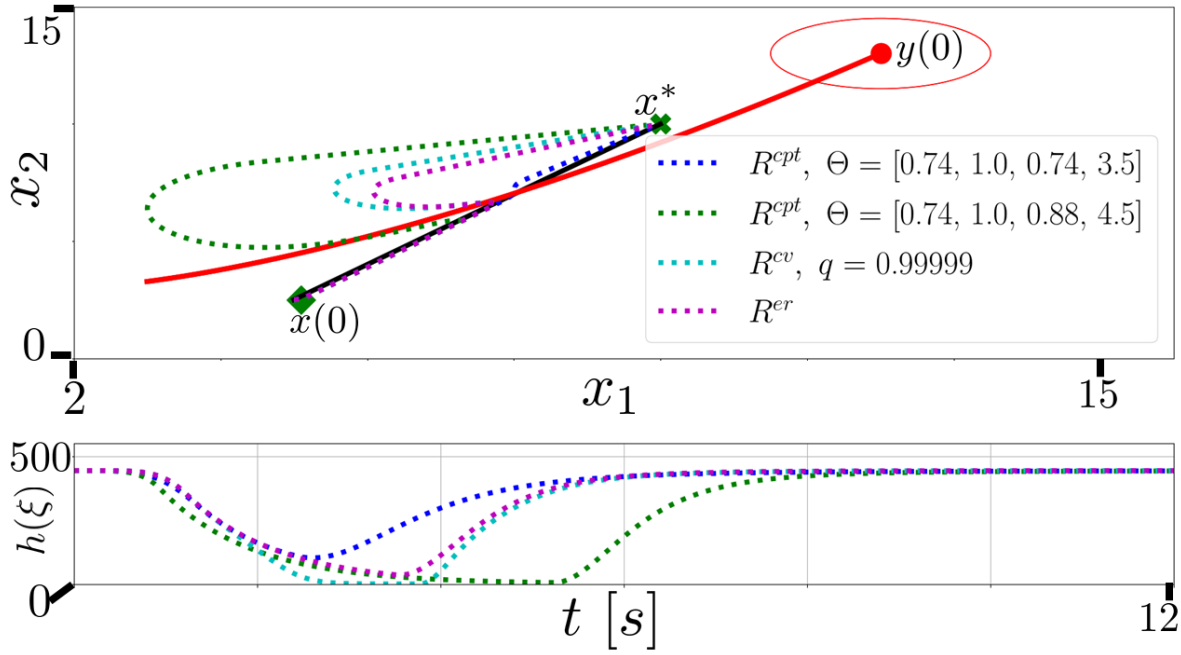


Figure 5.3. RPA-CBF demonstration through trajectories (top) and their corresponding $h(\xi)$ evolution (bottom). We consider 2D space $[2,17] \times [0,15] \subset \mathscr{X} \subset \mathbb{R}^2$, agent starting at $x(0) = (5,2)^\top$ (green diamond) and its goal at $x^* = (10,10)^\top$ (green cross). Risk source moves from $(13,13)^\top$ (red dot) to $(2,3)^\top$. Red ellipse represents the localization uncertainty $\mathbf{B}^{\bar{r}}(y_\mu)$. Without the risk source, agent takes nominal path (black line) using $k(x)$. Four risk functions are considered : CPT insensitive ($\theta = \{0.74, 1.0, 0.74, 3.5\}$), CPT averse ($\theta = \{0.74, 1.0, 0.88, 4.5\}$), CVaR averse ($q = 0.99999$), and ER. We simulate agent's path by applying u from (5.7) with above functions from $t = 0s$ to $t = 12s$, at $500Hz$. The agent takes blue path (CPT insensitive), green path (CPT averse), cyan path (CVaR averse) and magenta path (ER) to reach goal, with corresponding $h(\xi)$ shown below. For a detailed video showing more paths and risk dynamics see "https://youtu.be/lZ8s1QVtxwk".

outcomes according to Section 5.4 and calculate $R^{\text{cpt}}, R^{\text{cv}}, R^{\text{er}}$ appropriately. We use $\rho = c_\mu(\bar{r})$ to denote the risk threshold.

RPA-CBF Controller: We use the costs defined in the previous paragraph with $\bar{r} = 1.5$ and $k_1 = 200, k_2 = 0.01$. We use identity functions for η_1 and η_2 and respective risk functions in (5.1) to construct the RPA-CBF in (5.6). If obstacle and vehicle follow along straight paths, a collision can occur and perceived safety (Definition 3) can be violated. To handle unicycle dynamics we use the projected point method to control a virtual point $p \in \mathbb{R}^2$, a distance l along the direction of its heading (refer [SM21b]) for details. We use a standard proportional controller for $k(x) = kx$ with a constant $k = (0.6, 0.6)$. We note

that one can always appropriately tune the reference value ρ by l units to ensure safety w.r.t. x. The results of varying λ, γ and q are shown in Fig. 5.3. The results of using various models to generate u from (5.7) are shown in Fig. 5.3. The agent most likely will collide with the obstacle (red dot localized uncertainly in red ellipse) if it follows the nominal path (black line) from applying controls $k(x)$, thus making it unsafe. By using the controller u from (5.7), the agent swerves away from the obstacle corresponding to its risk perception and manages to reach the goal while maintaining $h \geq 0$ throughout, implying that perceived safety is maintained according to Definition 3. We see that the range of paths produced by CVaR lies between the magenta (ER) and cyan (CVaR extreme risk aversion) paths whereas, CPT is able to generate a wider range of paths (between green and blue and beyond) by tuning the risk aversion and risk sensitivity parameter than CVaR thus capturing a greater variety of risk perception, which follows the theoretical arguments from Theorem 2 and Theorem 3. We also see that the agent also reaches the goal due to the inherent stability properties of the nominal controller $k(x)$. See "https://youtu.be/lZ8s1QVtxwk" for a detailed video showing more paths and dynamics. Interestingly, we also found that CPT is more computationally efficient than CVaR in this setting. Time taken for 6000 evaluations[10] (corresponding to one path in Fig. 5.3) of (5.7) is $4.5s$ for CVaR, $2.9s$ for CPT and $1.9s$ for ER. We think this is because CVaR evaluates inverse cdf for every calculation whereas, CPT uses less costlier function evaluations[11]. Proper complexity analysis and comparison between various RPMs and their dependence on the probability distribution properties is still an open problem to solve. For another implementation with multiple moving obstacles we direct the reader to [SM21b]. This implementation uses principles from [GCE21] to construct a single RPA-CBF resulting from multiple sources of risk.

[10]Implemented in python with quadprog without visualization on a laptop with i7-8550U CPU @ 1.80GHz and 16GB ram

[11]Using truncated Gaussian costs, decision weights Π remain the same for a fixed θ, making CPT calculations more efficient.

5.7 Summary

In this chapter, we have proposed a novel integration of CPT into a safety-critical control scheme. This enables to generate risk-perception-aware controls (according to a DM's risk profile) in an environment embedded with dynamic uncertain costs. Thus, opening new avenues to incorporate behavioral decision theory into safety-critical controls.

In the next chapter, we look at user study design to better understand human risk perception in everyday environments.

This chapter, in full, is a reprint of the material as it appears in the publication Risk-Perception-Aware Control Design under Dynamic Spatial Risks, A. Suresh and S. Martínez, IEEE Control Systems Letters, 6, pp. 1802–1807, 2022. The dissertation author was the primary investigator and author of this paper.

Chapter 6

Risk-Aware Navigation in Human-Centered Environments

In this chapter, we propose a novel user study design to understand human decision making for path planning in risky and uncertain environments. We explore how robots can model human perception of risk for risk-aware navigation. By observing and understanding human risk preferences and conveying its risk preference to humans, robots can adjust their risk perception models appropriately. These models will then enable robots to detect high-risk areas in human-occupied environments and navigate in in a socially-acceptable manner. Although there are many theoretical models of risk, it is important to validate these models with human perception of risk, particularly with a large and diverse sample size to capture a wide range of preferences. This will reveal which models are well-suited for robot navigation in everyday settings, particularly for systems that communicate their reasoning to people.

Another important aspect is to assess the reliability of humans' self perception of risk and time-urgency to navigate in everyday scenarios. Often, humans tend to decide and act in a different manner, compared to their self perception of their own actions. So it is important to evaluate the effectiveness of questionnaires and self-reported measures of risk and time-urgency in the context of risk-aware navigation in everyday scenarios.

6.1 Literature Review

Risk is a very relevant and urgent notion that is used to design navigation algorithms in robotics. In order for AI to reason and make decisions in risky situations, various models have been employed to quantify risk. As we have seen before Conditional Value at Risk (CVaR) is one such popular model from financial risk management, which has been directly employed in robotics [CMP+21, STZT20]. CVaR based methods capture risk aversion (i.e., to 'play it safe') by employing linear and rational notions of decision-making under risk, which cannot capture the non-linear and non-rational decision-making that humans exhibit [Ste70, Ste57, Dha16].

To alleviate this shortcoming, Cumulative Prospect theory (CPT) [TK92] based methods have been recently proposed [SM21a, SM22] to plan and navigate in risky environments. These are more general methods designed to capture non-rational human decision-making along with standard risk notions. Theoretically, it has been shown that CPT is more "versatile", "inclusive" [SM22], and expressive [SM21a] than CVaR and Expected Risk (ER), thus capturing a wider range of diverse risk profiles of humans.

Recently, user studies found that CPT is indeed more representative of human decision-making in a traffic intersection scenario [KBT+20]. Researchers have also conducted a study to learn the utility function of human decision makers in a resource management setting [dSCMD20]. These studies utilized time-urgent and resource-constrained decision-making scenarios with money as a modality. A more general study pertaining to everyday scenarios using more abstract modalities remains under-explored.

Commonly used risk modalities like money [MP], time [GFBA10] or collision probabilities [HY20] is not very natural or explainable to humans, especially considering everyday social navigation scenarios. Recent studies have found that humans are often sub-optimal in planning paths in everyday scenarios [RDL18]. However, they assume that the human is either "noisy-rational" or they do not have the correct environment

models to make optimal decisions. These arguments are from a robot's perspective which acts in an expected manner and also expects the human to do so. However, from a human-centric and explainable-AI perspective, the robot's "expected" behavior might lead to mistrust and confusion [KPAK13]. So there a need for robots to be equipped with models, that can describe human's perception of the environment. This will lead to better understanding human decision making and consequently a better explainable AI in robots.

There has been great interest in using risk models in the robotics field to design safe intelligent systems for human-centered environments. For instance, recent work has explored fall risk assessment as a robot navigates with humans [NYMH20, KM20, BPMU18], risk of localization and mapping systems [HAS19, AHJS20], and risk for search and rescue operations [SAZ+21]. In our work, we are interested in risk models for human-aware robot navigation, particularly by considering human's perception of risks [KPAK13].

Much of prior work on risk-aware robot navigation focuses on obstacle avoidance which includes avoiding walls, objects, and people in the environment. These methods can be characterized as using probabilistic, dynamical systems, and reinforcement learning methods. For instance, some prior work employing probabilistic techniques use a Gaussian process risk map to model risk [HSP+20, GMH+19]. Examples of dynamical systems approaches model risk as an entropic risk measure which generate multiple trajectory forecast for multiple agents in crowds [NIG+20]. Reinforcement learning methods have employed deep reinforcement learning techniques to learn an uncertainty-aware policy that can update it's measure of risk without fine-tuning or retraining [CDK+21].

6.2 Contributions

Although great progress has been made to model risk for uncertain, human-occupied environments, there is a lack of work that validates human perception of risk under different conditions, compares theoretical models of risk with human perception of risk, and explore how consistent human perception of risk and time urgency is with standard survey responses. This work explores these gaps. Our main contributions are as follows:

1. Reveal trends in humans' path choices in risky environments
2. Reveal comparison trends between humans' path choices and that of popular risk models
3. Reveal the relationship between humans' self risk assessment, self time-urgency assessment, and their path choices in risky and time-urgent scenarios.

Our work also reveals insights into human navigation preferences compared to risk-aware models. Furthermore, equipped with these insights we can aim to design better models of human risk that enable more human-aware navigation models. These models can enable robots to operate safely and adapt to human preferences in real-world environments.

6.3 Research Questions

We explore how people's decision-making during risky scenarios compares to theoretical models of human perceptions of risk. We introduce a new study design that explores human-decision making in a navigation task with different risky scenarios. Also, we explore how consistent humans' self-assessment of risk and time-urgency is with their

risk-taking and time-urgent behavior. In particular, we are interested in addressing the following research questions:

RQ 1. How do humans make decisions under risky situations?

RQ 2. How do the human decisions compare to the ones from standard risk models in everyday risky situation?

RQ 3. How does human's self risk assessment and self time-urgency evaluation compare to their path choices in risky and time-urgent scenarios?

Next, we describe our study methodology used to address the above research questions.

6.4 Study Methodology

We conducted an IRB-approved (approval code: 201638) human-subjects online study on Qualtrics[1] survey platform. We recruited 80 participants affiliated with a university campus through university list-serves and via word of mouth. The participants consisted of 27 females, 48 males, and 3 that preferred not to answer this question with ages ranging from 21-32 (mean = 25.6, SD = 2.5).

This study focused on human-decision making under different risk scenarios. We consider a COVID-19 pandemic grocery shopping scenario, where the risk is characterized by the close proximity to "sick" people. Participants are in the shoes of an "Instacart Shopper" who needs to go from the entrance to the milk section. Time-urgency is characterized by the need to complete shopping quickly in order to get better ratings. This scenario is illustrated in Figure 6.1. There are three paths to choose from: Path A, Path B, and Path C. Each of these paths have varying intensity of risk and time urgency characteristics (discussed in detail in Section 6.4.1). The participants indicate their most

[1]https://www.qualtrics.com/

Figure 6.1. Grocery store environment used in the user studies. The participants have three paths (A, B, and C) to choose from, to go from the entrance (shown as the 'Start' in blue) to the milk section (shown as the 'Goal' in green). The supermarket is crowded with people having various levels of sickness ranging from 'Nil' to 'Severe'.

preferred and least preferred path for each scenario (constituents of one trial). Next, we will describe the study design in detail.

6.4.1 Study Design

We designed an online study that takes participants through several modules including a consent form, demographic survey, study description, demonstration, practice rounds, nine study trials, and post-study questionnaires. The consent form provided a brief description of the study and described how there are minimal risks for participating in this study. After giving informed consent, the participants filled out a demographic survey that collected information about their age, gender, occupation, and area of expertise. Afterwards, we described the study to participants.

We started by discussing the goal of the study which is to investigate how people

plan paths in risky situations. This work will enable us to design robot navigation systems that plan human-like paths as well as better adapt to peoples' motion in everyday environments. To study how humans select paths, we designed a study scenario that incorporates different levels of risk to enable us to study human decision-making. In each scenario, there is a trade-off between time and risk, where the riskiest path is also the shortest, while the longest path is the least risky in terms of encountering "sick" people.

Now, we discuss our thought process used to create the trial scenarios.

Scenario Creation

The users' are presented with three choices of paths to choose from including paths A, B, and C (see Figure 6.1). Path A is the longest, path B is the shortest, and path C has a length that is in between path A and path B. We used the situation of "being coughed at by sick people" to elicit risk for each path in every scenario. This risk was described by four decision variables: "time taken", "number of sick people", "level of sickness", and "chance of being coughed on" (see Table 6.1). The time taken varies from 5-20 minutes, number of sick people from 0 to 2, level of sickness from 0 to 3 and chance of being coughed at was expressed as a percentage for each sick person encountered. For level of sickness, we used the following terminology: 0-Nil, 1-Mild, 2-Moderate and 3-Severe (see Figure 6.1). We purposefully kept the consequences abstract, in order to extract realistic risk perception from participants. The variables are summarized in Table 6.1. Then, we asked participants to select their most favorite and least favorite path.

To engage participants in our study, we constructed nine different scenarios to explore how they make decisions under risky situations. Keeping the research questions in mind, we use the following rationale to construct the scenarios:

Table 6.1. Description of decision variables and their ranges for each path in every scenario.

No.	Decision Variables	Range of values presented
1	Time Taken	Path A : 20 mins Path B : 5 mins Path C : 10 mins
2	Number of Sick people	Path A : 0-1 Path B : 2-3 Path C : 1-2
3	Level of Sickness	0-3 for each path
4	Chance of being coughed at	0-100 % for each path

- There are a total of nine trials with different values of decision variables, aimed to capture a wide range of scenarios.
- In each trial, the shortest path (Path B) has the most risk and uncertainty, while the longest path (Path A) has the least risk and uncertainty.
- The participants choose most and least preferred path, thus giving us their preference order.
- The risk variables for each trial is designed such that the preferred choice (A or B or C) w.r.t. expected behavior (using expected risk) is varied across scenarios.
- There are 2 scenarios whose risk variables are such that all paths are preferred equally likely from an expected risk point of view.
- We group the nine trials into three levels of uncertainty (w.r.t. number of sick people). This helps us in understanding how the choices vary with increase in uncertainty.

The participants engaged in 9 trials where the three risk variables (2-4 in Table 6.1) on each path are varied to elicit different decision-making in risky situations. We used four parameters in total (one for time and three for risk) to characterize each path in every scenario.

In the next stage of the study, participants are given a demonstration of a study trial. The participants were presented with a demonstration scenario of the grocery store, which describes the four decision variables pictorially, in sentences and through a summary table. These modalities were selected after a few rounds of pilot studies which indicated different people prefer different modalities to describe the scenarios. Then, they selected their most and least favorite path. Based on their most preferred path choices, we randomly selected a risk outcome and displayed the final results (e.g., 'You encountered no sick people'). Next, participants engaged in three practice rounds with three different scenarios and selected their most and least favorite path.

After the participants engaged in practice rounds, they were exposed to the same nine study trials (within-subject study design) with different "risk" in each scenario. To remove ordering effects and influence of regret, we randomized the ordering of the trials across all participants.

6.4.2 Post-Study Questionnaire

At the end of the study, we administered two post-study questionnaires.The first questionnaire is called the General Risk Propensity Scale (GRiPS) [ZHN19] which measures participants' self risk taking abilities. In other words, it evaluates how risk averse or risk loving they think they are in their daily lives (See Section 6.5). GRiPS is a self-report measure (see Table 6.2) of general risk and pro-social behavior consisting of 8-items which participants answer on a Likert scale from 1 (Strongly Disagree) to 5 (Strongly Agree).

The second questionnaire is called the "Time Urgency Scale" [LRTC91] which measures participants' self assessment of how time-urgent and urgent they think they are in everyday scenarios. It is a self-report measure (See Table 6.3) of general time-related behavior consisting of 6-items (as commonly used [MN11]) which participants

Table 6.2. The 8-item the General Risk Propensity Scale (GRiPS) [ZHN19] that we administered to participants after engaging in our study.

GRIPS Survey Questions
1. Taking risks makes life more fun
2. My friends would say that I'm a risk taker
3. I enjoy taking risks in most aspects of my life
4. Taking risks is an important part of life
5. I commonly make risky decisions
6. I am a believer of taking chances
7. I would take a risk even is it meant I might get hurt
8. I am attracted, rather than scared, by risk

Table 6.3. The 6-item the Time Urgency Scale [LRTC91] that we administered to participants after engaging in our study.

Time Urgency Survey Questions
1. I find myself hurrying to get places even when there is plenty of time.
2. I often work slowly and leisurely.
3. People that know me well agree that I tend to do most things in a hurry.
4. I tend to be quick and energetic at work.
5. I often feel very pressed for time.
6. My spouse or a close friend would rate me as definitely relaxed and easy going.

answer on a Likert scale from 1 (Strongly Disagree) to 5 (Strongly Agree).

In addition, after the study trials and questionnaires were administered, we asked participants how they relatively weighed (as a %) each of the four variables in making their decision.

To summarize, the trial data helped us in answer RQ 1. RQ 2 can be assessed from analyzing the trial data along with the questionnaire responses.

6.5 Evaluation

Here we describe the metrics and methodology used to evaluate data from the user study described earlier.

6.5.1 Data Collection

From the study we obtain the following numeric data from each participant:

1. The participant's most preferred path for each trial $M \triangleq \{m_1, m_2, ..., m_9\}$. For any trial j, we set $m_j \triangleq 0$ for path A, $m_j \triangleq 1$ for path B, and $m_j \triangleq 0.5$ for path C. We use this coding strategy as Path A is the least risky and least urgent, whereas Path B is the most riskiest and most urgent, while Path C is in between.

2. The participants' least preferred path for each trial denoted as $L \triangleq \{l_1, l_2, ..., l_9\}$. We use a similar coding strategy as above where $l_j \triangleq 0$ for path A, $l_j \triangleq 1$ for path B, and $l_j \triangleq 0.5$ for path C.

3. The participants' GRIPS survey (Table 6.2) responses $R \triangleq \{r_1, r_2, ..., r_8\}$.

4. The participants' Time Urgency survey (Table 6.3) responses $T \triangleq \{t_1, t_2, ..., t_6\}$.

5. The participants' relative consideration (in %) of each decision variable (Table 6.1) $V \triangleq \{v_1, v_2, ..., v_4\}$, following the numbering in Table 6.1.

6.5.2 Metrics

We employ metrics to evaluate the participants' perception of risk (subjective metrics) and their risk-taking behavior (objective metrics).

Subjective Metrics:

These metrics evaluate the participant's self perception of time-urgency and risk. We use the data from the two standardized questionnaires to define the following:

- We obtain the risk score $\overline{R}$ as the average response (normalized between 0 and 1) from the GRIPS survey responses R (as commonly used [NQN+20]). Where a $\overline{R} = 1$ indicates a risk liking perception vs $\overline{R} = 0$ indicating a risk averse perception.

- We obtain the time urgency score $\overline{T}$ as the average response (normalized between 0 and 1) from the Time-urgency survey responses T. Where a $\overline{T} = 1$ indicates being time urgent, whereas $\overline{T} = 0$ indicates being time relaxed.

From the two questionnaires, we obtain the participant's perception of their own risk-taking and time-urgent behavior in their daily lives.

Objective Metrics:

These metrics illustrate the participant's risk and time urgency characteristics during the user study. Here, we consider three different risk perceptions to compare the users' decision making. First, we consider "expected behavior" using Expected Risk (ER), from which we get M^{exp} and L^{exp}, which are the MPP and LPP choices made respectively treating risk in an expected manner. The next behavior we observe is "risk aversion" using CVaR, where we similarly get M^{averse} and L^{averse}. Finally, we consider "risk insensitive" behavior using CPT with M^{insens} and L^{insens} as the MPP and LPP choices respectively. We note that from our previous theoretical results [SM21a], we have shown that CPT is the most inclusive model and can capture all three perceptions (risk insensitive, risk averse, expected). CVaR can capture expected and risk averse perception, whereas ER only captures the expected behavior.

1. We measure the average most preferred path score $\overline{M}$ for each participant from M. This indicates the average risk the participant is willing to take as well as the time urgency the participant faces while planning.

2. We measure the average least preferred path score $\overline{L}$.

3. We measure deviation from expected behavior $J^{exp} = \sum_{i=1}^{9} |M - M^{exp}|$, where a larger value indicated a greater deviation from expected perception.

4. Similarly, we measure deviation from risk aversion $J^{averse} = \sum_{i=1}^{9} |M - M^{averse}|$, where a larger value indicated a greater deviation from risk averse perception.

5. Similarly, we use deviation from risk insensitivity $J^{insens} = \sum_{i=1}^{9} |M - L^{insens}|$, where a larger value indicated a greater deviation from risk insensitive perception.

6. We can then evaluate the models by taking the minimum of the deviations over the behaviors that the model can capture. So for ER the model deviation is given by $J^{ER} = J^{exp}$, for CVaR we get $J^{CVaR} = \min\{J^{exp}, J^{averse}\}$ and for CPT we have $J^{CPT} = \min\{J^{exp}, J^{averse}, J^{insens}\}$.

7. We calculate a risk similarity score $R^{sim} = \overline{M} - \overline{R}$ to show us if people choose paths according to the GRIPS survey responses. The measure R^{sim} varies from -1 to 1, where a larger positive value indicates that the participant chose paths in a riskier fashion than they indicate in the survey. Whereas, a larger negative value indicates that the participants are more risk averse in the study than they claim in the survey.

8. Similarly, we calculate a time urgency similarity score $T^{sim} = \overline{M} - \overline{T}$ to show us if people choose paths according to the time urgency survey responses. As above, the measure T^{sim} varies from -1 to 1, where a larger positive value indicates that the participant chose shorter and more urgent paths than they indicate in the survey. Whereas, a larger negative value indicates that the participants are more time relaxed in the study than they claim in the survey.

6.6 Results and Discussion

We use metrics defined in the previous section to answer the research questions posed in Section 6.3. We provide descriptive statistics of the relevant variables and metrics including the mean, median, standard deviation and 95% confidence interval. To study correlation between two variables, we calculate the Pearson's correlation coefficient, along with null-hypothesis significance testing with threshold p-value = 0.05.

Table 6.4. Descriptive statistics of relevant variables to address RQ 1

No.	Variable	Mean	Median	95% confidence interval of mean	Standard Deviation	Min	Max
1	$\overline{M}$	0.60	0.60	0.56 to 0.66	0.20	0.00	1.00
2	$\overline{L}$	0.40	0.40	0.35 to 0.47	0.30	0.00	0.94
3	J^{exp} for MPP	0.40	0.40	0.35 to 0.40	0.10	0.11	0.61
4	J^{averse} for MPP	0.30	0.30	0.31 to 0.36	0.10	0.00	0.61
5	J^{insens} for MPP	0.30	0.30	0.28 to 0.36	0.20	0.00	0.89
6	J^{CVaR} for MPP	0.30	0.30	0.31 to 0.36	0.10	0.00	0.61
7	J^{CPT} for MPP	0.20	0.20	0.21 to 0.26	0.10	0.00	0.44
8	J^{exp} for LPP	0.40	0.40	0.37 to 0.42	0.10	0.06	0.67
9	J^{averse} for LPP	0.40	0.40	0.34 to 0.44	0.20	0.00	1.00
10	J^{insens} for LPP	0.30	0.40	0.31 to 0.43	0.30	0.00	0.89
11	J^{CVaR} for LPP	0.30	0.30	0.27 to 0.34	0.20	0.00	0.61
12	J^{CPT} for LPP	0.20	0.30	0.22 to 0.30	0.20	0.00	0.61

6.6.1 Study and compare path choices with RPMs

In order to compare risk models with human decision-making, we provide descriptive statistics of the relevant variables, which is summarized in Table 6.4. The independent variables are the behavior-based path choices for MPP (M^{exp}, M^{averse}, L^{insens}) and LPP (L^{exp}, L^{averse}, L^{insens}). The dependent variables are participants path choices M and L for each trial.

Now, we provide a discussion for each variable in Table 6.4, row by row. Recall that MPP for a j^{th} trial with $m_j = 0$ is path A with lowest risk and most time relaxed. Whereas, $m_j = 1$ is path B with highest risk and most time-urgent and Path C is in between in both risk and time-urgency with $m_j = 0.5$.

1. Average MPP $\overline{M}$: The mean, median and confidence interval are over 0.5. This indicates a preference towards Path B which is more risky and time urgent. We also see that the extremes are 0 (most risk averse and time relaxed) and 1 (most risk insensitive and time-urgent), which shows that there were participants in the complete opposite end of the spectra.

2. Average LPP $\overline{L}$: The mean, median and confidence interval are below 0.5. This indicates a preference towards Path A which is less risky and time relaxed. We see that the max < 1 indicating that there was no participant who unanimously disliked path B, whereas the min is 0 indicating there was at least one participant who disliked Path A for all trials.

3. Deviation from expected behavior J^{exp} for MPP: Row 3 shows a higher deviation in almost all columns as compared to risk-averse J^{averse} and risk insensitive J^{insens} behavior. Thus indicating that participants' preference was not very frequently according to ER. In fact, the min > 0, which indicates that not a single participant chose paths according to expected risk in all their trials.

4. Deviation from risk averse J^{averse} and risk insensitive J^{insens} behavior for MPP: The statistics for risk averse and risk insensitive behavior is almost similar, with both having min 0, indicating that there are participants that completely choose according to the respective risk perception.

5. Deviation from risk models J^{ER}(same as J^{exp}), J^{CVaR}, J^{CPT} for MPP: Since J^{CPT} is lesser than both J^{CVaR} and J^{ER}(same as J^{exp}) in all columns, it is clear that CPT as a model can better capture decision making. This corroborates with the theoretical results that CPT is more "expressive" [SM21a] and "inclusive" [SM22] than CVaR and ER. This is because it can produce a full range of risk perception from most insensitive to most averse. Also, J^{CVaR} is lesser than J^{ER} as CVaR can capture risk aversion better than ER, which people have exhibited in these trials.

6. Deviations J^{exp}, J^{averse}, and J^{insens} for LPP: The deviations show similar characteristics in mean, median and confidence interval, with J^{insens} showing slightly lesser mean. Both J^{insens} and J^{averse} have a min 0, while min of $J^{exp} > 0$. This indicates that similar to MPP, not a single participant chose LPPs in an expected manner,

Table 6.5. Descriptive statistics of relevant variables to address RQ 2

No.	Variable	Mean	Median	95% confidence interval of mean	Standard Deviation	Min	Max
1	$\overline{R}$	3.40	3.30	3.18 to 3.55	0.90	1.00	5.00
2	$\overline{T}$	3.10	3.00	2.90 to 3.25	0.80	1.33	4.83
3	R^{sim} for MPP	0.10	0.00	-0.05 to 0.09	0.30	-0.84	0.94
4	R^{sim} for LPP	0.00	0.00	-0.08 to 0.07	0.30	-0.91	0.94
5	T^{sim} for MPP	0.10	0.10	0.04 to 0.15	0.30	-0.58	0.67
6	T^{sim} for LPP	0.10	0.10	0.00 to 0.14	0.30	-0.58	0.67

whereas some participants showed characteristic risk averse and risk insensitive perception for all trials. Also, max of $J^{averse} = 1$, showing that there was at least one participant who didn't exhibit risk aversion at all for all the choices.

7. Deviations J^{ER}(same as J^{exp}), J^{CVaR}, J^{CPT} for LPP: The trend is similar to that from MPP. J^{CPT} is lesser than both J^{CVaR} and J^{ER}(same as J^{exp}), and J^{CVaR} is lesser than J^{ER}, in all columns. Thus, showing CPT capturing better decision making than CVaR and ER.

Next we will look at addressing RQ 3.

6.6.2 Compare Path choice characteristics with survey responses

In order to compare risk models with human decision-making, we provide descriptive statistics of the relevant variables, which is summarized in Table 6.5. The independent variables Survey responses R and T. The dependent variables are participants path choices M and L for each trial. We will also perform correlation studies between the dependent and independent variables to provide to deeper insight. Now, we provide a discussion for each variable in Table 6.5, row by row. Recall that MPP for a j^{th} trial with $m_j = 0$ is path A with lowest risk and most time relaxed. Whereas, $m_j = 1$ is path B with highest risk and most time-urgent and Path C is in between in both risk and time-urgency with $m_j = 0.5$.

1. Average Survey responses $\overline{R}$: The mean, median and confidence interval are all over 3, indicating that on an average majority of the participants perceive that they are inclined towards taking risks. But the standard deviation is quite high at 0.9 indicating less neutral perception. There are also extreme cases (with min 1 and max 5), implying that there are some people who think they are extreme risk takers and also some who are scared and averse to take risks.

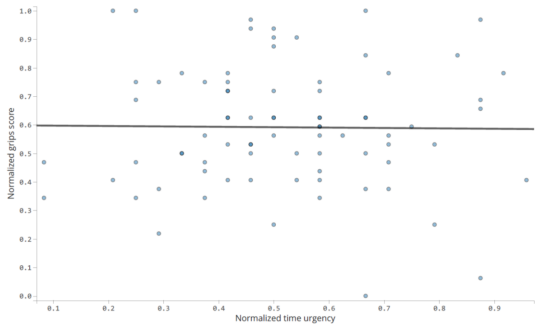
2. Average Survey responses $\overline{T}$: Here, the perception is more balanced with the mean, median and confidence interval close to 3.0. For this survey as well there are extreme responses with min 1.33 and max 4.83 and high standard deviation of 0.9. Hence, there are a people variety of different perception from highly time relaxed to highly time urgent.

3. Risk similarity score R^{sim} for MPP and LPP: On an average R^{sim} is balanced with the mean, median and confidence interval close to 0. However, there are extreme responses with min -0.84 and max 0.94 and high standard deviation of 0.3. Hence, there are a people variety of different perception and the GRIPS may not fully represent their decision making in he study.

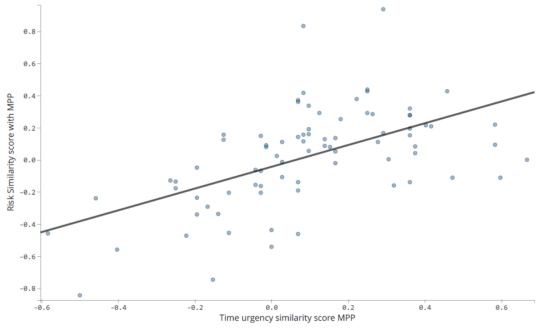
4. Time urgency similarity score T^{sim} for MPP and LPP: This metric shows more biased characteristics than R^{sim}. For both LPP and MPP, the mean, median, and confidence interval are positive, implying that the people act more urgently than they indicate in the survey. Although the standard deviation is high at 0.3, the min and max are well within the extremes, indicating that time urgency has a more moderate perception range.

Correlation

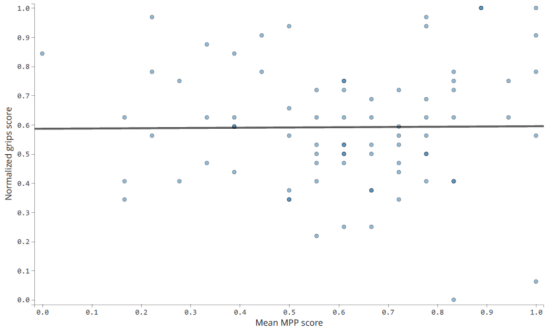
Here, we try to identify trends between GRiPS and Time-Urgency survey responses and path choices by measuring correlation and performing linear regression.



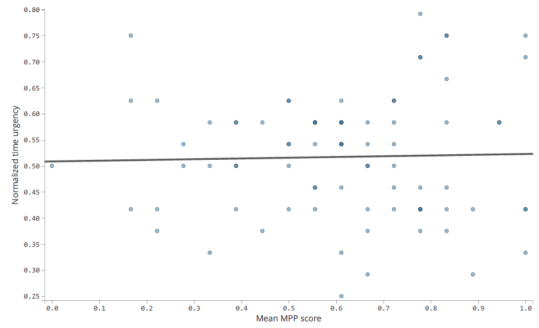
(a) Normalized GRIPS response $\overline{R}$ v/s Normalized time urgency response $\overline{T}$



(b) Risk similarity score R^{sim} v/s time urgency similarity score T^{sim}



(c) Normalized GRIPS response $\overline{R}$ v/s Mean MPP score $\overline{M}$



(d) Normalized time urgency response $\overline{T}$ v/s Mean MPP score $\overline{M}$

Figure 6.2. Scatter plots and correlation between various variables derived from user data in the study

The results are described in Table 6.6 and is partly visualized in Figure 6.2, which are discussed below.

Again, recall that MPP for a j^{th} trial with $m_j = 0$ is path A with lowest risk and most time relaxed. Whereas, $m_j = 1$ is path B with highest risk and most time-urgent and Path C is in between in both risk and time-urgency with $m_j = 0.5$.

1. There was an insignificant interaction between the average survey responses for risk $\overline{R}$ and time urgency $\overline{T}$ with p = 0.912 and an effect size close to 0 (see row 1 in Table 6.6); thus, these variables are not correlated. This reveals that participants' self risk evaluation and time-urgency perception from the survey are not related to each other and these two entities are assessed differently by individuals.

2. There was a significant interaction between the average survey responses for risk

Table 6.6. Correlation statistics of relevant variables to address RQ 2

No.	Variable X	Variable Y	P-Value	Effect Size (Pearson's r)	95% confidence interval of effect size
1	$\overline{T}$	$\overline{R}$	0.912	-0.012	-0.232 to 0.208
2	T^{sim} for LPP	R^{sim} for LPP	<0.001	0.617	0.460 to 0.737
3	T^{sim} for MPP	R^{sim} for MPP	<0.001	0.540	0.364 to 0.679
4	$\overline{M}$	$\overline{R}$	0.936	0.009	-0.211 to 0.228
5	$\overline{L}$	$\overline{R}$	0.655	-0.051	-0.268 to 0.171
6	$\overline{M}$	$\overline{T}$	0.738	0.038	-0.183 to 0.256
7	$\overline{L}$	$\overline{T}$	0.268	-0.125	-0.336 to 0.098

similarity score R^{sim} and time urgency similarity score T^{sim} for MPP and LPP with and $p < 0.05$ and a effect size <0.5 (see row 2 and 3 in Table 6.6); thus, these variables are correlated. This implies that people who acted more/less riskier than they indicated in the GRIPS survey, also acted correspondingly more/less time urgent than they indicated in the time-urgency survey. This can arise because of the study construction, where paths which are shorter (time urgent) are also riskiest and vice versa.

3. There was an insignificant interaction between the normalized GRIPS response $\overline{R}$ with MPP and LPP scores $\overline{M}/\overline{L}$ with $p = 0.936$ and $p = 0.655$, and effect size close to 0 (see row 4 and 5 in Table 6.6); thus, these variables are not correlated. This reveals that participants' self risk evaluation and their path choice not related to each other. So just relying on the GRIPS survey to depict participants' risk perception may not be effective in such situations.

4. There was an insignificant interaction between time urgency response $\overline{T}$ with MPP and LPP score $\overline{M}/\overline{L}$ with $p = 0.738$ and $p = 0.268$, and effect size close to 0 (see row 6 and 7 in Table 6.6); thus, these variables are not correlated. This reveals that participants' self time-urgency evaluation and their path choice not related to each other. So just relying on the time-urgency survey to depict participants'

time perception may not be effective in such situations.

Next, we use the Paired t-test for the pairs of variables in Table 6.6 to see any comparison trends. The results are summarized in Table 6.7. We will discuss the results in the following paragraphs.

1. We find that responses for risk $\overline{R}$ tends to be slightly larger than time urgency $\overline{T}$ with $p < 0.05$ and a small effect size of 0.249. (see row 1 in Table 6.7); therefore, these variables are correlated. From the GRiPS and Time Urgency surveys, participants have the propensity to take more risks in comparison to be more time-urgent.

2. We find that risk similarity score R^{sim} tends to be slightly smaller than T^{sim} for MPP and LPP with $p < 0.05$ and a small effect size of 0.249 (see row 2 and 3 in Table 6.7); thus, these variables are correlated. This means that w.r.t. the corresponding surveys, participants behaved with time-urgency more than their riskier behavior.

3. There was an insignificant interaction between the normalized GRiPS response $\overline{R}$ with MPP $\overline{M}$ with p $= 0.544$, and effect size close to 0 (see row 4 and 5 in Table 6.7). However with LPP scores $/\overline{L}$, $\overline{R}$ showed a positive correlation with $p < 0.001$ and a medium effect size $d > 0.5$. This reveals no correlation between participants' self risk evaluation and their MPP choicee. But, considering LPP choices, participants' LPP is less likely riskier than they indicated. Thus, although their MPP may not match the survey responses, they tend to not prefer safer paths than their indicated risk appetite.

4. We found time urgency response $\overline{T}$ that is slightly less than the MPP score $\overline{M}$ at p $= 0.06$ and slightly larger than the LPP score $/\overline{L}$ with p $= 0.08$ and effect size

Table 6.7. Paired t-test statistics of relevant variables to address RQ 2

No.	Variable X	Variable Y	P-Value	Effect Size (Cohen's d)	Difference Between Means (Y - X)	95% confidence interval of difference
1	$\overline{T}$	$\overline{R}$	0.029	0.249	0.070	0.010 to 0.140
2	T^{sim} for LPP	R^{sim} for LPP	0.029	0.249	-0.070	-0.140 to -0.010
3	T^{sim} for MPP	R^{sim} for MPP	0.029	0.249	-0.070	-0.140 to -0.010
4	$\overline{M}$	$\overline{R}$	0.544	0.068	-0.020	-0.090 to 0.050
5	$\overline{L}$	$\overline{R}$	<0.001	0.503	0.18	0.100 to 0.260
6	$\overline{M}$	$\overline{T}$	0.006	0.318	-0.090	-0.160 to -0.030
7	$\overline{L}$	$\overline{T}$	0.008	0.302	0.110	0.030 to 0.190

close of 0.318 and 0.302 respectively (see row 6 and 7 in Table 6.7). This reveals that participants' prefer to choose slightly more time-urgent paths than indicated by their survey responses. Whereas, they tend to not prefer more time-relaxed paths than their indicated risk appetite.

6.7 Summary

In this chapter, we have proposed a novel user study design to understand human decision making for path planning in a risky and uncertain environments. The summary of our findings from the user study is as follows:

1. Participants tend to prefer riskier and time urgent paths, while they least prefer safer and time relaxed paths.

2. Participants tend to show risk averse and risk insensitive behavior more often than expected behavior with respect to risk.

3. CPT risk model captures participants' decision making better than CVaR and ER, while ER performs the worst of the three models.

4. On an average the survey responses effectively captured participants' risk and time urgency behavior from the trials. However, they may not be fully indicative of

people's risk and time urgency preferences as there was a large standard deviation in the similarity scores.

5. Participants tend to prefer safer paths the least compared to their indicated risk appetite for the survey.

6. There was no significant correlation between participants' risk propensity and time urgency indicated in the respective surveys.

6.8 Limitations and Future Work

Here we will discuss some of the main limitations of our work and then mention possible future work for improvement.

Due to the online nature of the study, participants' path choices may not be fully representative of their actions in real world setting. Also, there might have been an inadvertent sampling bias resulting in a relatively younger age group of the participant population. A more in-person study in a real world supermarket or similar setting may be needed to further validate our claims.

Although, we had a large sample size in terms of number of people, we only collected limited data (only 9 trials) per participant to minimize fatigue. More data points are needed to explicitly compare and characterize decision making between participants and risk models. This limitation can be again alleviated by conducting in-person user studies where data can be collected in a natural and continuous manner (entire paths), which can then be used to perform more rigorous comparisons between various risk models.

The material in this chapter, in full, is currently under preparation for publication as Risk-Aware Navigation in Human-Centered Environments, A. Suresh, A. Taylor, L. Riek, and S. Martínez. The dissertation author was the primary investigator and author of these papers.

Chapter 7

Conclusions

In this thesis, we have developed and evaluated intuitive and risk-perception-aware navigation algorithms to enable robots to perceive, reason and act more appropriately in risky and uncertain human centered environments .

In Chapter 3, we addressed the notion of intuition by developing a human-swarm-interaction framework. We have combined diverse tools from control theory, network science, machine learning, signal processing, optimization and robotics to create this multi-disciplinary framework. Firstly, we have demonstrated the effectiveness of human interaction using this framework, whose accuracy and speeds are comparable to standard interaction devices. Next, we have proposed and utilized a unique notion of shape morphing dynamics along with switching systems to simultaneously plan intermediate shapes for the swarm to depict and choose swarm parameters, indicating the domain of both the user and the swarm. We have also developed, analyzed and illustrated a novel decentralized formation controller capable of reaching any shape, centroid, rotation and scaling in the $2-D$ space. Lastly, we have integrated the framework by developing a GUI environment which interacts with user by means of gestures, and rest of the framework is encapsulated in the GUI using matlab simulations.

In Chapter 4, we have proposed a novel adaptation of CPT to model a DM's non-rational perception of a risky environment in the context of path planning. Firstly,

we adapt CPT into path planning to model non-rational perception of spatial cost embedded in an environment. With this, we can capture a larger variety of risk perception models, extending the existing literature. Secondly, we generate desirable paths using a sampling-based (RRT*-based) planning algorithm on the perceived risky environment. Our planner integrates a continuous risk profile and path length to calculate path cost, enabling us to plan in the perceived environment setting above. Furthermore, the chosen cost satisfies the sufficient conditions for asymptotic optimality of the planner, leading to reliable and consistent paths according to a specified risk profile. We then compare our planner's performance with T-RRT* (continuous cost space planner) and Risk-RRT* (risk-aware planner) through simulations in cluttered and dynamic environments respectively. We show that our proposed planner can generate better paths in comparison. Finally, we define the notion of "expressiveness" for a risk perception model and show that CPT's is higher than that of CVaR and expected risk. Furthermore using SPSA, we show that the expressiveness hierarchy translates to our path planning setting, where we observe that a planner equipped with CPT can better approximate arbitrary paths in an environment.

In Chapter 5, we proposed a risk-perception-aware safety-critical control scheme to guide a robot in dynamic risky and uncertain environments. We first showed how to employ non-rational risk-perception notions to express perceived safety for control systems, expanding the spectrum of risk perception models considered in the literature. We formally introduced the novel concepts of "inclusiveness" and "versatility" to compare and contrast the ability of different models to handle a variety of DMs' risk profiles. We considered two popular models, CVaR, and ER, and prove that CPT is both more "inclusive" and "versatile" than these. Next, we constructed a class of CBFs which guarantee safety according to a DM's perceived risk, corresponding to any RPM. Using a QP formulation, we obtained Risk-Perception-Aware (RPA) controls that can guide an agent

to a desired goal while maintain perceived safety. We analyzed feasibility conditions and stability properties of the proposed approach in terms of the three RPMs, and proved that CPT equipped RPA controller has both a larger feasible control set and more accurate stabilization. We illustrated our results through simulations and animations in a 2D environment.

In Chapter 6, we proposed a novel user study design to understand human decision making for path planning in a risky and uncertain environments. We explored how robots can model how humans perceive risk in terms of risk-aware navigation. We revealed trends in humans' path choices in risky environments and compared correlation between humans' path choices and that of popular risk models. We also revealed the relationship between humans' self risk assessment, self time-urgency assessment, and their path choices in risky and time-urgent scenarios. We provided insights for designing Explainable AI systems for robotic navigation in everyday human occupied spaces. Furthermore, with these findings, we can aim to design better models of human risk and time-urgency perception that can enable better human-aware navigation in robots. Thus, enabling robots to operate safely and adapt to human preferences in real-world environments.

Appendix A

HSI Proofs

A.1 Preliminaries for proof of Theorem 1

Let us first define the following quantities : $x = (x_1^\top, x_2^\top, x_3^\top)^\top$, $x_1 = p$, $x_2 = c$, $x_3 = q$, $\mathbf{F}_2 = \mathbf{F}_3 = \mathbf{0}, \in \mathbb{R}^M$, $f_1(x_1) = (I - \alpha D^{-1}L)x_1$; $f_2(x_2) = Wx_2$; $f_3(x_3) = \mathbf{0}$ $g_1(x) = -k^p x_2$; $g_2(x) = x_1 - x_3$; $g_3(x) = x_1$; and $\mathbb{F} = [\mathbf{F}_1^\top, \mathbf{F}_2^\top, \mathbf{F}_3^\top]^\top$.

With these definitions (3.5) can be represented as :

$$x_k(t+1) = f_k(x_k(t)) + g_k(x(t)) + \mathbf{F}_k, \quad \forall k \in \{1,2,3\}, \tag{A.1}$$

where f_k is the system dynamics of the k^{th} system, g_k is the interconnection to the k^{th} system and $\mathbf{F}_k$ is the drift of the k^{th} system. Now $f_1(x_1) + \mathbf{F}_1$ resembles the shape stabilizing JOR algorithm in [Cor09] with some additional centroid drift $k^p 1_M \mathbf{c}$. From [Cor09] we know this system converges to the desired shape with some centroid translation. Henceforth, we will ignore the drift $\mathbb{F}$ while analyzing the overall system stability. As we see next, stability is established by first analyzing the convergence rates of each of the subsystems defined by f_k, and by identifying suitable conditions on the interconnections g_k, for $k \in \{1,2,3\}$. To this end, we define the Lyapunov function $V(x_k) = \frac{1}{2} x_k^\top x_k$, defined over x_k for $k \in \{1,2,3\}$.

Lemma 4. The subsystem $x_1(t+1) = f_1(x_1(t)) - 1_M 1_M^\top] x_1(t)$ is globally uniformly asymp-

totically stable at $x_1 = \mathbf{0}$.

Proof. Considering $A = I_M - \alpha D^{-1}L$, we have that the eigenvalues $\lambda^A \in (0,1]$ and 1 is a simple eigenvalue with right eigenvector 1_M, which shows $x_1(t+1) = Ax_1(t)$ is globally stable. We can perform a similarity transformation on A to get $A_s = I_M - \alpha L^N$ where $L^N = D^{\frac{-1}{2}} L D^{\frac{-1}{2}}$ is the symmetric normalized Laplacian of the graph. We perform a Hotelling deflation [Saa03] on A_s using the largest eigenvalue to get $\overline{A} = A_s - D^{\frac{-1}{2}} 1_M 1_M^\top D^{\frac{-1}{2}}$. In this way, we have deactivated the largest eigenvalue of A_s and now we have $\lambda^{\overline{A}} \in [0, 1-\alpha\lambda_2^N]$ where λ_2^N is the second smallest eigenvalue of the normalized Laplacian L^N. We will proceed by analyzing the stability properties of $\overline{A}$ which is similar to analyzing the stability of $x_1(t+1) = [(I_M - \alpha D^{-1}L) - 1_M 1_M^\top] x_1(t)$.

With $\Delta V(x_1) = V(x_1(t+1)) - V(x_1(t))$ and $Q = \overline{A}^\top \overline{A} - I_M$ we have $\Delta V(x_1) = x_1^\top Q x_1 < 0$. The above observation follows from the fact that $\overline{A}$ is symmetric and $\lambda^{\overline{A}} \in (0, 1-\alpha\lambda_2^N]$, hence the eigenvalues $\lambda^Q \in (-1, (1-\alpha\lambda_2^N)^2 - 1]$, which makes Q negative definite. From Lyapunov theory we have that $x_1(t+1) = [(I_M - \alpha D^{-1}L) - 1_M 1_M^\top] x_1(t)$ is globally uniformly asymptotically stable about the origin. From the theory of symmetric quadratic forms we also have the following inequality

$$\Delta V(x_1) \leq -(1 - (1-\alpha\lambda_2^N)^2) \|x_1\|^2, \tag{A.2}$$

which gives us a convergence rate for the $x_1(t+1) = [(I_M - \alpha D^{-1}L) - 1_M 1_M^\top] x_1(t)$ dynamics.

□

Now we will analyze the second subsystem. The Matrix W has 1 as the simple eigenvalue with eigenvector 1_M. The matrix $\overline{W} = W - \frac{1_M^\top 1_M}{M}$ is Schur stable and $\lambda^{\overline{W}} \in (-\frac{n-2}{n}, 1-\lambda_2^W)$, where $\lambda_2^W \in [0,1]$ is the second smallest eigenvalue associated with the weighted graph $\mathcal{G}_\text{w}$. Hence we will analyze the convergence of the system $x_2(t+1) =$

$\overline{W}x_2(t)$, which will give us the convergence rate for system f_2.

Lemma 5. The system $x_2(t+1) = \overline{W}x_2(t)$ is globally uniformly asymptotically stable to the origin, and the convergence rate of system f_2 is proportional to $(1-(1-\lambda_2^W)^2)$.

Proof. With $\Delta V(x_2) = V(x_2(t+1)) - V(x_2(t))$ and $Q_2 = \overline{W}^\top \overline{W} - I_M$ we have $\Delta V(x_2) = x_2^\top(t) Q_2 x_2(t) < 0$. This follows from the fact that the eigenvalues $\lambda_{Q_2} \in ((\frac{n-2}{n})^2 - 1, (1-\lambda_2^W)^2 - 1)$, which makes Q_2 negative definite. Hence according to Lyapunov theory $x_2(t+1) = f_2(x_2(t)) - \frac{1_M^\top 1_M}{M} x_2(t)$ is globally uniformly asymptotically stable to the origin. In addition,

$$\Delta V_2(x_2) \leq -(1-(1-\lambda_2^W)^2)\|x_2\|^2, \tag{A.3}$$

which finally gives us the convergence rate for the $f_2(x_2)$ dynamics.

□

The analysis of the third subsystem, $x_3(t+1) = 0$, is trivial. Now, let us define the following constants: $\delta_1 = 1-(1-\alpha\lambda_2^N)^2$, $\delta_2 = 1-(1-\lambda_2^W)^2$, $\delta_3 = 1$, $\gamma_{11} = \gamma_{13} = \gamma_{22} = \gamma_{32} = \gamma_{33} = 0$, $\gamma_{12} = k^p$, $\gamma_{21} = \gamma_{23} = \gamma_{31} = 1, \beta_k = 1$, and $\phi(x_k) = \|x_k\|$, $\forall k \in \{1,2,3\}$.

Now we are ready to state the stability of System (3.4).

Proof of Theorem 1

Proof. The system (3.4) can be equivalently represented in the form (3.5). Now let us first consider driftless system (3.5). The positive definite Lyapunov functions $V_k(x_k) \equiv V(x_k)$ and the interconnection functions $g_k(x)$ satisfy the conditions of (A.4) for all $t \geq 0$. From Lemma 4 and Lemma 5 for each subsystem $k,l \in \{1,2,3\}$ we have:

$$\Delta V(x_k) \leq -\delta_k \phi^2(x_k), \tag{A.4a}$$

$$\left\| \frac{\partial V(x_k)}{\partial x_k} \right\| \leq \beta_k \phi(x_k), \tag{A.4b}$$

$$\|g_k(t,x)\| \leq \sum_{l=1}^{3} \gamma_{kl} \phi(x_k). \tag{A.4c}$$

Now if we consider a diagonal matrix $\mathbf{diag}(\delta) \in \mathbb{R}^{3\times 3}$ with entries $(\delta_1, \delta_2, \delta_3)$, a column vector $\beta = (\beta_1, \beta_2, \beta_3)^\top$ and a matrix $\Gamma = (\gamma_{kl}) \in \mathbb{R}^{3\times 3}$, we can define a Matrix $S \in \mathbb{R}^{3\times 3}$ as follows

$$S = \mathbf{diag}(\delta) - \beta\Gamma. \tag{A.5}$$

The Matrix S is an M-matrix, which is characterized by non-positive off diagonal entries and positive leading principal minors. The first leading principal minor is positive from the definition of the constants and the connectivity Assumption 1. For the second leading principal minor to be positive we require $k^p < \delta_1\delta_2$. For the third leading principal minor $(\det(S))$ to be positive we require $k^p < \frac{\delta_1\delta_2}{2}$.

Now, we choose k^p accordingly such that S defined according to (A.5) is an M matrix. Now from [Kha02] (cf. Theorem 9.2) we can conclude that the interconnected system (3.5) is globally stable.

As the interconnections are asymptically stable we can infer the following. Firstly, the subsystem (3.5a) reaches the desired centroid due to the shifting term $k^p 1_M c_d$. Additionally, assumptions in executing the FODAC algorithm in [ZM10] are satisfied due to the current assumption, and the fact that the first order differences of the reference signal are asymptotically stable from Lemma 4. Thus, the centroid estimate $\hat{c}(t)$ converges to $p(t)$. Thus, the overall system converges to the desired state X^d. □

Appendix B

Chapter 5 Additional results

B.1 Inclusiveness Condition

Lemma 6 (Inclusiveness condition). Consider a threshold $\rho \in \mathbb{R}_{\geq 0}$, a risk source at $\bar{y} \in \mathscr{X}$, and two models $\mathscr{M}_1, \mathscr{M}_2$ with range spaces $\mathscr{R}_1, \mathscr{R}_2$, respectively. If $\mathscr{R}_2 \subseteq \mathscr{R}_1$, and if there exists an $R_1 \in \mathscr{M}_1$ such that $R_1 > R_2$ or $R_1 < R_2$ for any $R_2 \in \mathscr{M}_2$, and any c, then $\mathscr{M}_1 \rhd \mathscr{M}_2$. In addition, if there are $R_1^a, R_1^b \in \mathscr{M}_1$ such that $R_1^a > R_2^a$ and $R_1^b < R_2^b$, $\forall R_2^a, R_2^b \in \mathscr{M}_2$, and any c, then $\mathscr{M}_1 \blacktriangleright \mathscr{M}_2$.

Proof. Fix c. Since $\mathscr{R}_2 \subseteq \mathscr{R}_1$, $\forall R_2 \in \mathscr{M}_2$, there is $R_1 \in \mathscr{M}_1$ s.t. $R_1(\bar{y} - x) = R_2(\bar{y} - x)$, $\forall x \in \mathscr{X}$. Thus, $\mathscr{Y}_2 \subseteq \mathscr{Y}_1$ and $\overline{\mathscr{Y}}_2 \subseteq \overline{\mathscr{Y}}_1$. Assume $\exists \overline{R}_1, \widetilde{R}_1 \in \mathscr{M}_1$ s.t. $\overline{R}_1(\bar{y} - x) > R_2(\bar{y} - x)$ or $\widetilde{R}_1(\bar{y} - x) < R_2(\bar{y} - x)$ hold for all $R_2 \in \mathscr{M}_2$. This implies either $\overline{\mathscr{Y}}_2 \subsetneq \overline{\mathscr{Y}}_1$ or $\mathscr{Y}_2 \subsetneq \mathscr{Y}_1$. Inclusiveness follows from Definition 4. In parallel, $\mathscr{M}_1 \blacktriangleright \mathscr{M}_2$. □

B.2 Range Comparison

Lemma 7. (RPM range space comparison) Consider CPT, CVaR and ER risk models, with associated range sets $\mathscr{R}_{\text{cpt}}, \mathscr{R}_{\text{CVaR}}$, and $\mathscr{R}_{\text{er}}$. Then, $\mathscr{R}_{\text{cpt}} \supsetneq \mathscr{R}_{\text{CVaR}} \supseteq \mathscr{R}_{\text{er}}$, holds $\forall c$.

Proof. Fix c. Note that $\mathscr{R}_{\text{er}} = \{c_\mu\}$. By choosing $R^{\text{cpt}}{}_{\overline{\theta}}$ with $\overline{\theta} = \{1, 1, 1, 1\}$ and $R_0^{\text{cv}} \in$ we have $R_0^{\text{cv}} = R^{\text{er}} = R^{\text{cpt}}{}_{\overline{\theta}}$, $\forall c$. Note that only if $c_\sigma = 0$ then $R_q^{\text{cv}} = c_\mu = R^{\text{er}}$ for all q.

When $c_\sigma \neq 0$, with any other valid choice of parameters q in CVaR we obtain $R_q^{\text{cv}} \notin \mathscr{R}_{\text{er}}$. We can find $\theta \neq \overline{\theta}$ such that ${R^{\text{cpt}}}_\theta \notin \mathscr{R}_{\text{er}}$, $\forall c$. Hence, $\mathscr{R}_{\text{er}} \subseteq \mathscr{R}_{\text{CVaR}}$ and $\mathscr{R}_{\text{cpt}} \supsetneq \mathscr{R}_{\text{er}}$.

For CVaR, $R_0^{\text{cv}} = \{c_\mu\}$ and $R_1^{\text{cv}} = \{b\}$, where $b \in \mathbb{R}$ is the worst-case outcome of c. Since R_q^{cv} increases in q, $\mathscr{R}_{\text{CVaR}} \subseteq [c_\mu, b]$. Choosing $\theta_1 = \{1,1,1,\lambda\}$, for $\lambda \geq 1$, leads to ${R^{\text{cpt}}}_{\theta_1} = \lambda \sum_i c_i p_i = \lambda c_\mu$. Taking $\lambda \in [1, \bar{b}]$, with $\bar{b} > \frac{b}{c_\mu}$, we get $\mathscr{R}_{\text{cpt}} \supset [c_\mu, b]$; hence, $\mathscr{R}_{\text{cpt}} \supsetneq \mathscr{R}_{\text{CVaR}}$. □

Bibliography

[ACE+19] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada. Control Barrier Functions: Theory and Applications. In European Control Conference, pages 3420–3431, 2019.

[ADEH99] P. Artzner, F. Delbaen, J.M. Eber, and D. Heath. Coherent measures of risk. Mathematical Finance, 9(3):203–228, 1999.

[AHJS20] Guillermo Duenas Arana, Osama Abdul Hafez, Mathieu Joerger, and Matthew Spenko. Localization safety validation for autonomous robots. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 6276–6281. IEEE, 2020.

[AMBR+12] J. Alonso-Mora, A. Breitenmoser, M. Rufli, R.Siegwart, and P. Beardsley. Image and animation display with multiple mobile robots. International Journal of Robotics Research, 31(6):753–773, 2012.

[AMLL+15] J. Alonso-Mora, S. H. Lohaus, P. Leemann, R. Siegwart, and P. Beardsley. Gesture based human - multi-robot swarm interaction and its application to an interactive display. In IEEE Int. Conf. on Robotics and Automation, pages 5948–5953, 2015.

[AXA21a] M. Ahmadi, X. Xiong, and A. D. Ames. "guaranteed obstacle avoidance for multi-robot operations with limited actuation: A control barrier function approach. IEEE Control Systems Letters, 5(1):127–132, 2021.

[AXA21b] M. Ahmadi, X. Xiong, and A. D. Ames. Risk-averse control via cvar barrier functions: Application to bipedal robot locomotion. IEEE Control Systems Letters, 6(1):878–883, 2021.

[BB07] B. Burns and O. Brock. Sampling-based motion planning with sensing uncertainty. In IEEE Int. Conf. on Robotics and Automation, pages 3313–3318, 2007.

[BCM09] F. Bullo, J. Cortés, and S. Martínez. Distributed Control of Robotic Networks. Applied Mathematics Series. Princeton University Press, 2009.

[BHM18] B. Boardman, T. Harden, and S. Martínez. Limited range spatial load balancing in non-convex environments using sampling-based motion planners. Autonomous Robots, 42(8):1731–1748, 2018.

[BOW11] L. Blackmore, M. Ono, and B. C. Williams. Chance-constrained optimal path planning with obstacles. IEEE Transactions on Robotics, 27(6):1080–1094, 2011.

[BPMU18] Joaquin Ballesteros, Jose Manuel Peula, Antonio B Martinez, and Cristina Urdiales. Automatic fall risk assessment for challenged users obtained from a rollator equipped with force sensors and a rgb-d camera. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 7356–7361. IEEE, 2018.

[CCM18] C.-Y. Chang, J. Cortés, and S. Martínez. Scheduled-asynchronous distributed algorithm for optimal power flow. IEEE Transactions on Control of Network Systems, 6(1):261–275, 2018.

[CDK+21] J. Choi, C. R. Dance, J. Kim, S. Hwang, and K. Park. Risk-conditioned distributional soft actor-critic for risk-sensitive navigation. arXiv preprint arXiv:2104.03111, 2021.

[CE15] J.P.D.L. Croix and M. Egerstedt. A control lyapunov function approach to human-swarm interactions. In American Control Conference, pages 4368–4373, Chicago, USA, 2015.

[CHL+05] H. Choset, S. Hutchinson, K. M. Lynch, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun. Principles of Robot Motion: Theory, Algorithms and Implementations. The MIT Press, 2005.

[CKT+16] W. Chi, H. Kono, Y. Tamura, A. Yamashita, H. Asama, and M. Q. Meng. A human-friendly robot navigation algorithm using the risk-rrt approach. In IEEE Int. Conf. on Real-time Computing and Robotics, pages 227–232, 2016.

[Cla19] A. Clark. Control Barrier Functions for Complete and Incomplete Information Stochastic Systems. In American Control Conference, pages 2928–2935, 2019.

[CM17] W. Chi and M. Q. Meng. Risk-RRT*: A robot motion planning algorithm for the human robot coexisting environment. In Int. Conf. on Advanced Robotics, pages 583–588, 2017.

[CMP+21] A. Choudhry, B. Moon, J. Patrikar, C. Samaras, and S. Scherer. Cvar-based flight energy risk assessment for multirotor uavs using a deep energy model. arXiv preprint arXiv:2105.15189, 2021.

[Cor09] J. Cortés. Global formation-shape stabilization of relative sensing networks. In American Control Conference, pages 1460–1465, St. Louis, MO, 2009.

[CPG18] Y. Chen, H. Peng, and J. Grizzle. Obstacle avoidance for low-speed autonomous vehicles with Barrier Function. IEEE Transactions on Control Systems Technology, 26(1):194–206, 2018.

[CYZ+16] S. Cheng, L. Yu, D. Zhang, L. Huo, and J. Ji. Consensus of second-order multi-agent systems using partial agents' velocity measurements. Nonlinear Dynamics, 86(3):1927–1935, 2016.

[Dha16] S. Dhami. The Foundations of Behavioral Economic Analysis. Oxford University press, 2016.

[D.P98] D.Prelec. The probability weighing function. Econometrica, 66(3):497–527, 1998.

[DSC16] D. Devaurs, T. Simeon, and J. Cortes. Optimal Path Planning in Complex Cost Spaces with Sampling-based Algorithms. IEEE Transactions on Automation Sciences and Engineering, 13(2):415–424, 2016.

[dSCMD20] Paulo E U de Souza, Caroline P C Chanel, Melody Mailliez, and Frédéric Dehais. Predicting Human Operator's Decisions Based on Prospect Theory. Interacting with Computers, 32(3):221–232, 2020.

[ESBS16] B. Englot, T. Shan, S. D. Bopardikar, and A. Speranzon. Sampling-based min-max uncertainty path planning. In IEEE Int. Conf. on Decision and Control, pages 6863–6870, 2016.

[FM04] J. A. Fax and R. M. Murray. Information flow and cooperative control of vehicle formations. IEEE Transactions on Automatic Control, 49(9):1465–1476, 2004.

[Fra17] A. Franchi. Human-Collaborative Schemes in the Motion Control of Single and Multiple Mobile Robots, pages 301–324. Springer International Publishing, 2017.

[FSR+12] A. Franchi, C. Secchi, M. Ryll, H. H. Bulthoff, and P. Robuffo Giordano. Shared control: Balancing autonomy and human assistance with a group of quadrotor uavs. IEEE Robotics and Automation Magazine, Special Issue on Aerial Robotics and the Quadrotor Platform, 19:57–68, 2012.

[GCE21] P. Glotfelter, J. Cortés, and M. Egerstedt. Nonsmooth approach to controller synthesis for Boolean specifications. IEEE Transactions on Automatic Control, 66(11):5160–5174, 2021.

[GF07] V. Gazi and B. Fidan. Coordination and control of multi-agent dynamic systems: Models and approaches. In Swarm Robotics, Lecture Notes in Computer Science, pages 71–102. Springer, Berlin, Germany, 2007.

[GFBA10] S. Gao, E. Frejinger, and M. Ben-Akiva. Adaptive route choices in risky traffic networks: A prospect theory approach. Transportation Research Part C: Emerging Technologies, 18(5):727–740, 2010.

[GGD16] B. Gromov, L. M. Gambardella, and G. A. Di Caro. Wearable multi-modal interface for human multi-robot interaction. In IEEE Int. Symposium on Safety, Security, and Rescue Robotics, pages 240–245, 2016.

[GM18] E. Gravelle and S. Martínez. Distributed dynamic lane reversal and rerouting for traffic delay reduction. International Journal of Control, 91(10):2355–2365, 2018.

[GMH+19] Hongliang Guo, Zehui Meng, Zefan Huang, Leong Wei Kang, Ziyue Chen, Malika Meghjani, Marcelo Ang, and Daniela Rus. Safe path planning with gaussian process regulated risk map. In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 2044–2051. IEEE, 2019.

[GR01] C. D. Godsil and G. F. Royle. Algebraic Graph Theory, volume 207 of Graduate Texts in Mathematics. Springer, New York, 2001.

[HAS19] Osama Abdul Hafez, Guillermo Duenas Arana, and Matthew Spenko. Integrity risk-based model predictive control for mobile robots. In 2019 International Conference on Robotics and Automation (ICRA), pages 5793–5799. IEEE, 2019.

[HKY19] A. Hakobyan, G. C. Kim, and I. Yang. Risk-aware motion planning and control using CVaR-constrained optimization. IEEE Robotics and Automation Letters, 4(4):3924–3931, 2019.

[HOGD18] R. Hendrix, P. Owan, J. Garbini, and S. Devasia. Context-specific seperable gesture selection for control of a robotic manufacturing assistant. In IFAC Conf. on Cyber Physical and Human Systems, volume 51, pages 89–96, 2018.

[HS19] A. R. Hota and S. Sundaram. Game-Theoretic Protection against Networked SIS Epidemics by Human Decision-Makers. IFAC Papers Online, 51(34):145–150, 2019.

[HSP+20] Z. Huang, W. Schwarting, A. Pierson, H. Guo, M. Ang, and D. Rus. Safe path planning with multi-model risk level sets. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 6268–6275. IEEE, 2020.

[HY20] A. Hakobyan and I. Yang. Wasserstein distributionally robust motion control for collision avoidance using conditional value-at-risk. arXiv preprint arXiv:2001.04727, 2020. Available at https://arxiv.org/abs/2001.04727.

[JAF+18] C. Jie, P. L. A., M. Fu, S. Marcus, and C.Szepesvari. Stochastic optimization in a Cumulative Prospect Theory framework. IEEE Transactions on Automatic Control, 63(9):2867–2882, 2018.

[JGGD14] N. Jawad, A. Giusti, L.M. Gambardella, and G.A. Di Caro. Human-swarm interaction using spatial gestures. In IEEE/RSJ Int. Conf. on Intelligent Robots & Systems, pages 3834–3841, 2014.

[JLM03] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. IEEE Transactions on Automatic Control, 48(6):988–1001, 2003.

[KBP12] H. Kurniawati, T. Bandyopadhyay, and N. M. Patrikalakis. Global motion planning under uncertain motion, sensing, and environment map. Autonomous Robots, 33(3):1–18, 2012.

[KBT+20] Minae Kwon, Erdem Biyik, Aditi Talati, Karan Bhasin, Dylan P Losey, and Dorsa Sadigh. When humans aren't optimal: Robots that collaborate with risk-aware humans. In 2020 15th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pages 43–52. IEEE, 2020.

[KDFA20] M. J. Khojasteh, V. Dhiman, M. Franceschetti, and N. Atanasov. Probabilistic safety constraints for learned high relative degree system dynamics. In Proceedings of Learning for Dynamics and Control, pages 781–792, 2020.

[KF11] S. Karaman and E. Frazzoli. Sampling-based algorithms for optimal motion planning. International Journal of Robotics Research, 30(7):846–894, 2011.

[Kha90] O. Khatib. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots, pages 396–404. Springer New York, 1990.

[Kha02] H. Khalil. Nonlinear Systems. Prentice Hall, 2002.

[KM16] V. Krishnan and S. Martínez. Self-organization in multi-agent swarms via distributed computation of diffeomorphisms. In Mathematical Theory of Networks and Systems, Minneapolis, MN, USA, July 2016.

[KM20] K. Koide and J. Miura. Collision risk assessment via awareness estimation toward robotic attendant. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 11011–11016. IEEE, 2020.

[KPAK13] Thibault Kruse, Amit Kumar Pandey, Rachid Alami, and Alexandra Kirsch. Human-aware robot navigation: A survey. Robotics and Autonomous Systems, 61(12):1726–1743, 2013.

[KPC+16] A. Kolling, W. Phillip, N. Chakraborty, K. Sycara, and M. Lewis. Human interaction with robot swarms: A survey. IEEE Transactions on Human-Machine Systems, 46(1):9–26, 2016.

[LKH13] B.D. Luders, S. Karaman, and J.P. How. Robust sampling-based motion planning with asymptotic optimality guarantees. In AIAA Conf. on Guidance, Navigation and Control, August 2013.

[LRTC91] F. J. Landy, H. Rastegary, J. Thayer, and C. Colvin. Time urgency: The construct and its measurement. Journal of Applied Psychology, 76(5):644–657, 1991.

[LSH21] B. T. Lopez, J. J. E. Slotine, and J. P. How. Robust adaptive Control Barrier Functions: An adaptive and data-driven approach to safety. IEEE Control Systems Letters, 5(3):1031–1036, 2021.

[MMS10] E. Montijano, S. Martínez, and C. Sagués. De-RANSAC: Robust consensus for robot formations. In Network Science and Systems in Multi-Robot Autonomy, Workshop at the IEEE International Conference on Robotics and Automation 2010, 2010.

[MN11] Susan Mohammed and Sucheta Nadkarni. Temporal diversity and team performance: The moderating role of team temporal leadership. Academy of Management Journal, 54(3):489–508, 2011.

[MP] A. Majumdar and M. Pavone. How should a robot assess risk? towards an axiomatic theory of risk in robotics. arXiv:1710.11040.

[MWVM13] V. M. Monajjemi, J. Wawerla, R. Vaughan, and G. Mori. HRI in the sky: Creating and commanding teams of UAVs with a vision-mediated gestural interface. In IEEE/RSJ Int. Conf. on Intelligent Robots & Systems, pages 617–623, 2013.

[NCLS17] S. Nagavalli, M. Chandarana, M. Lewis, and K. Sycara. Multi-operator gesture control of robotic swarms using wearable devices. In Int. Conf. on Advances in Computer-Human Interactions, pages 25–33, 2017.

[NCS17] S. Nagavalli, N. Chakraborty, and K. Sycara. Automated sequencing of swarm behaviors for supervisory control of robotic swarms. In IEEE Int. Conf. on Robotics and Automation, pages 2674–2681, Singapore, 2017.

[NIG+20] H. Nishimura, B. Ivanovic, A. Gaidon, M. Pavone, and M. Schwager. Risk-sensitive sequential action control with multi-modal human trajectory forecasting for safe crowd-robot interaction. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 11205–11212. IEEE, 2020.

[NQN+20] Muhammad Asif Nadeem, Muhammad Ali Jibran Qamar, Mian Sajid Nazir, Israr Ahmad, Anton Timoshin, and Khurram Shehzad. How investors attitudes shape stock market participation in the presence of financial self-efficacy. Frontiers in Psychology, 11, 2020.

[NYMH20] Roya Sabbagh Novin, Amir Yazdani, Andrew Merryweather, and Tucker Hermans. Risk-aware decision making in service robots to minimize risk of patient falls in hospitals. arXiv preprint arXiv:2010.08124, 2020.

[OC19] P. Ong and J. Cortés. Universal formula for smooth safe stabilization. In IEEE Int. Conf. on Decision and Control, pages 2373–2378, Nice, France, December 2019.

[OPA15] K.-K Oh, M.-C. Park, and H.-S Ahn. A survey of multi-agent formation control. Automatica, 53:424–440, 2015.

[OSFM06] R. Olfati-Saber, J. A. Fax, and R. M. Murray. Consensus and cooperation in multi-agent networked systems. Proceedings of the IEEE, 95(1):215–233, 2006.

[PC19] A. Pietrabissa and L. R Celsi. Discrete-time selfish routing converging to the wardrop equilibrium. IEEE Transactions on Automatic Control, 64(3):1288–1294, 2019.

[PJ04] S. Prajna and A. Jadbabaie. Safety Verification of Hybrid Systems using Barrier Certificates. In Hybrid systems: Computation and Control, pages 477–492, 2004.

[PJP07] S. Prajna, A. Jadbabaie, and G. J. Pappas. A framework for worst-case and stochastic safety verification using barrier certificates. IEEE Transactions on Automatic Control, 52(8):1415–1428, 2007.

[POND14] G. Podevijn, R. O'Grady, Y. S. G. Nashed, and M. Dorigo. Gesturing at subswarms: Towards direct human control of robot swarms. In Towards Autonomous Robotic Systems, pages 390–403, 2014.

[Rab89] L. R. Rabiner. A tutorial on Hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE, 77(2):257–286, 1989.

[RB05] W. Ren and R. W. Beard. Consensus seeking in multiagent systems under dynamically changing interaction topologies. IEEE Transactions on Automatic Control, 50(5):655–661, 2005.

[RCN14] M. Rubenstein, A. Cornejo, and R. Nagpal. Programmable self-assembly in a thousand-robot swarm. Science, 345:795–799, 2014.

[RDL18] Sid Reddy, Anca Dragan, and Sergey Levine. Where do you think you're going?: Inferring beliefs about dynamics from behavior. Advances in Neural Information Processing Systems, 31, 2018.

[RLM17] E. Ramírez-Llanos and S. Martínez. A distributed dynamics for virus-spread control. Automatica, 76:41–48, 2017.

[RU00] R. T. Rockafellar and S. Uryasev. Optimization of conditional value-at-risk. Journal of risk, 2(1):21–42, 2000.

[Saa03] Y. Saad. Iterative methods for sparse linear systems. SIAM, 2003.

[SAZ+21] Vikram Shree, Beatriz Asfora, Rachel Zheng, Samantha Hong, Jacopo Banfi, and Mark Campbell. Exploiting natural language for efficient risk-aware multi-robot sar planning. IEEE Robotics and Automation Letters, 6(2):3152–3159, 2021.

[SCGG17] V. Suraci, L. R. Celsi, A. Giuseppi, and A. Di Giorgio. A distributed wardrop control algorithm for load balancing in smart grids. In Mediterranean Conf. on Control and Automation, pages 761–767, 2017.

[SCMP19] S. Singh, Y. Chow, A. Majumdar, and M. Pavone. A framework for time-consistent, risk-sensitive model predictive control: Theory and algorithms. IEEE Transactions on Automatic Control, 64(7):2905–2912, 2019.

[SE15] T. Shan and B. Englot. Sampling-based minimum risk path planning in multiobjective configuration spaces. In IEEE Int. Conf. on Decision and Control, pages 814–821, 2015.

[SF12] K. Savla and E. Frazzoli. A dynamical queue approach to intelligent task management for human operators. Proceedings of the IEEE, 19:672–686, 2012.

[SFKE15] T. Setter, A. Fouraker, H. Kawashima, and M. Egerstedt. Haptic interactions with multi-robot swarms using manipulability. Journal of Human-Robot Interaction, 4(1):78–95, 2015.

[SGW19] J. Song, S. Gupta, and T.A. Wettergren. T*: Time-optimal risk-aware motion planning for curvature-constrained vehicles. IEEE Robotics and Automation Letters, 4(1):33–40, 2019.

[SLVP18] E. Schmerling, K. Leung, W. Vollprecht, and M. Pavone. Multimodal Probabilistic Model-based planning for Human-Robot Interaction. In IEEE Int. Conf. on Robotics and Automation, pages 1–9, 2018.

[SM19] A. Suresh and S. Martínez. Gesture-based human-swarm interactions for formation control using interpreters. IFAC Papers Online, 51(34):83–88, 2019.

[SM20] A. Suresh and S. Martínez. Human-swarm interactions for formation control using interpreters. International Journal of Control, Automation and Systems, 18:2131—2144, 2020. DOI: 1007/s12555-019-0497-3.

[SM21a] A. Suresh and S. Martínez. Planning under non-rational perception of uncertain spatial costs. IEEE Robotics and Automation Letters, 6(2):4133–4140, 2021.

[SM21b] A. Suresh and S. Martínez. Risk-Perception-Aware Control Design under Dynamic Spatial Risks. arXiv preprint arXiv:2109.04570, 2021.

[SM22] A. Suresh and S. Martínez. Risk-Perception-Aware Control Design under Dynamic Spatial Risks. IEEE Control Systems Letters, 6:1802 – 1807, 2022.

[Spa03] J. C. Spall. Introduction to Stochastic Search and Optimization. New York, 2003.

[SS16] A. Suresh and M. Schwager. Brain-swarm interface (bsi): Controlling a swarm of robots with brain and eye signals from an eeg headset. arXiv preprint arXiv:1612.08126v1, 2016.

[Ste57] S. S. Stevens. On the psychophysical law. Psychological Review, 64(3):153–181, 1957.

[Ste70] S. S. Stevens. Neural events and the psychophysical law. Science, 170(3962):1043–1050, 1970.

[STZT20] Vishnu D Sharma, Maymoonah Toubeh, Lifeng Zhou, and Pratap Tokekar. Risk-aware planning and assignment for ground vehicles using uncertain perception from aerial vehicles. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 11763–11769. IEEE, 2020.

[Sur16] A. Suresh. Body swarm interface (bosi): controlling robotic swarms using human bio-signals. Master's thesis, Boston University, 2016.

[SY18] S. Samuelson and I. Yang. Safety-aware optimal control of stochastic systems using Conditional Value-at-Risk. In American Control Conference, pages 6285–6290, 2018.

[TB12] N. E. Du Toit and J. W. Burdick. Robot motion planning in dynamic, uncertain environments. IEEE Transactions on Robotics, 28(1):101–115, 2012.

[TBF05] S. Thrun, W. Burgard, and D. Fox. Probabilistic Robotics. Intelligent Robotics and Autonomous Agents. The MIT Press, 2005.

[TK92] A. Tversky and D. Kahneman. Advances in Prospect theory: Cumulative representation of uncertainty. Journal of Risk and Uncertainty, 5(4):297–323, 1992.

[VSSF17] V. Villani, L. Sabattini, C. Secchi, and C. Fantuzzi. Natural interaction based on affective robotics for multi-robot systems. In Int. Symposium on Multi-Robot and Multi-Agent Systems (MRS), pages 56–62, 2017.

[WATD20] S. N. Williams, C. J. Armitage, T. Tampe, and K. Dienes. Public perceptions and experiences of social distancing and social isolation during the covid-19 pandemic: a uk-based focus group study. BMJ Open, 10(e039334):1–8, 2020.

[WBM18] C. Wu, A.M. Bayen, and A. Mehta. Stabilizing traffic with autonomous vehicles. In IEEE Int. Conf. on Robotics and Automation, pages 1–7, 2018.

[WLY18] L. Wang, Q. Liu, and T. Yin. Decision-making of investment in navigation safety improving schemes with application of Cumulative Prospect Theory. Journal of Risk and Reliability, 232(6):710–724, 2018.

[WS16] Z. Wang and M. Schwager. Kinematic multi-robot manipulation with no communication using force feedback. In IEEE Int. Conf. on Robotics and Automation, pages 427–432, May 2016.

[XB04] L. Xiao and S. Boyd. Fast linear iterations for distributed averaging. Systems and Control Letters, 53:65–78, 2004.

[XUD+19] Feiyu Xu, Hans Uszkoreit, Yangzhou Du, Wei Fan, Dongyan Zhao, and Jun Zhu. Explainable ai: A brief survey on history, research areas, approaches and challenges. In CCF international conference on natural language processing and Chinese computing, pages 563–574. Springer, 2019.

[XW08] F. Xiao and L. Wang. Consensus protocols for discrete-time multi-agent systems with time-varying delays. Automatica, 44(10):2577–2582, 2008.

[ZHA09] W. Zhang, J. Hu, and A. Abate. On the value functions of the discrete-time switched LQR problem. IEEE Transactions on Automatic Control, 54(11):2669–2674, 2009.

[ZHN19] Don C Zhang, Scott Highhouse, and Christopher D Nye. Development and validation of the general risk propensity scale (grips). Journal of Behavioral Decision Making, 32(2):152–167, 2019.

[ZM10] M. Zhu and S. Martínez. Discrete-time dynamic average consensus. Automatica, 46(2):322–329, 2010.