

Modeling Supervisor Safe Sets for Improving Collaboration in Human-Robot Teams

Dexter Scobee

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2018-55

<http://www2.eecs.berkeley.edu/Pubs/TechRpts/2018/EECS-2018-55.html>

May 11, 2018



Copyright © 2018, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

Acknowledgement

Most of the content of this report is adapted from the paper "Modeling Supervisor Safe Sets for Improving Collaboration in Human-Robot Teams," available at <https://arxiv.org/abs/1805.03328>. That work is the result of a collaborative effort with co-lead author David McPherson, who built the experimental platform in addition to his theoretical contributions, Joseph Menke, who contributed key insights in analyzing algorithm performance, Allen Yang, who helped frame the project as part of a broader context, and, of course, our research advisor Shankar Sastry, who provided guidance throughout our research.

This work is supported by the Office of Naval Research under the Embedded Humans MURI (N00014-13-1-0341) as well as a Philippine-California Advanced Research Institutes (PCARI) grant.

Modeling Supervisor Safe Sets for Improving Collaboration in Human-Robot Teams

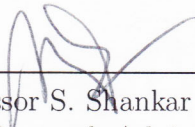
by Dexter Ryan Richard Scobee

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

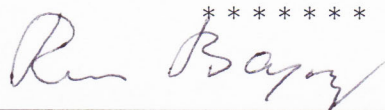
Committee:



Professor S. Shankar Sastry
Research Advisor

05/10/2018

(Date)



Professor Ruzena Bajcsy
Second Reader

May 4th, 2018

(Date)

Contents

1	Introduction and Background	6
2	Supervisor Safe Set Control	9
2.1	Preliminaries: Reachability for Safety	9
2.2	Noisy Idealized Supervisor Model	11
2.3	Learning Safe Sets from Supervisor Interventions	12
2.4	Team Control with Learned Safe Sets	14
3	Experimental Design for User Validation	18
3.1	Procedure	18
3.2	Independent Variables	21
3.3	Dependent Measures	21
3.3.1	Objective Measures	21
3.3.2	Subjective Measures	21
3.4	Subject Allocation	22
4	Analysis and Discussion	23
4.1	H1: False Positive Reduction over Standard	23
4.2	H2: Preference over Conservative	24
4.3	Model Validity	26
5	Conclusion	28

List of Figures

1.1	Perceived Safety	7
2.1	Illustration of Sets from Reachability Analysis	11
2.2	Zero Level Sets for a Library of Dynamics Functions	15
2.3	Example Intervention Data and Value Estimation	16
3.1	Safe Sets from User Study	19
3.2	Experimental Task Screen	20
4.1	Supervisory False Positives by Safe Set	24
4.2	Learned vs. Baseline Safe Sets	25
4.3	Empirical Distribution of Supervisor Interventions	27

List of Tables

4.1 Predicted vs. Observed False Positives 26

Abstract

When a human supervisor collaborates with a team of robots, the human’s attention is divided, and cognitive resources are at a premium. We aim to optimize the distribution of these resources and the flow of attention. To this end, we propose the model of an idealized supervisor to describe human behavior. Such a supervisor employs a potentially inaccurate internal model of the the robots’ dynamics to judge safety. We represent these safety judgements by constructing a *safe set* from this internal model using reachability theory. When a robot leaves this safe set, the idealized supervisor will intervene to assist, regardless of whether or not the robot remains objectively safe. False positives, where a human supervisor incorrectly judges a robot to be in danger, needlessly consume supervisor attention. In this work, we propose a method that decreases false positives by learning the supervisor’s safe set and using that information to govern robot behavior. We prove that robots behaving according to our approach will reduce the occurrence of false positives for our idealized supervisor model. Furthermore, we empirically validate our approach with a user study that demonstrates a significant ($p = 0.0328$) reduction in false positives for our method compared to a baseline safety controller.

Chapter 1

Introduction and Background

As automation becomes more pervasive throughout society, humans will increasingly find themselves interacting with autonomous and semi-autonomous systems. These interactions have the potential to multiply the productivity of humans workers, since it will become possible for a single human to supervise the behavior of multiple robotic agents. For example, a single human driver could manage a fleet of self-driving delivery robots, but the driver would only take full control for the “last mile,” guiding the robots to precisely deposit packages in environments where autonomous navigation may not be reliable. Human experts regularly serve as failsafe supervisors on factory assembly floors staffed with robotic arms [1]. Air traffic controllers soon will have to manage completely autonomous drones flying through their airspace alongside existing traditional mixed-autonomy planes and their auto-pilots [2].

While a human may be able to successfully exert direct control over a single robot, it becomes intractable for a human to directly control teams of robots (in fact, humans often benefit from automated assistance when controlling even a single robot, as discussed in the literature on assistive teleoperation [3, 4]). In order to manage the increased complexity of multi-robot teams, the human must be able to rely on increased autonomy from the robots, freeing the human to focus their attention only on those areas where they are most needed. Our goal is to model what grabs the supervisor’s attention in order to modify robot behavior to reduce the occurrence of distractions.

This project is inspired by work like A. Bajcsy et al [5] and Jain et al [6] that learn from supervisor interventions in a “coactive” learning framework. These works apply Learning from Demonstration techniques to the more challenging domain where the given data is just a correction from a trajectory rather than a full trajectory. The authors of [5] posed this correction challenge in

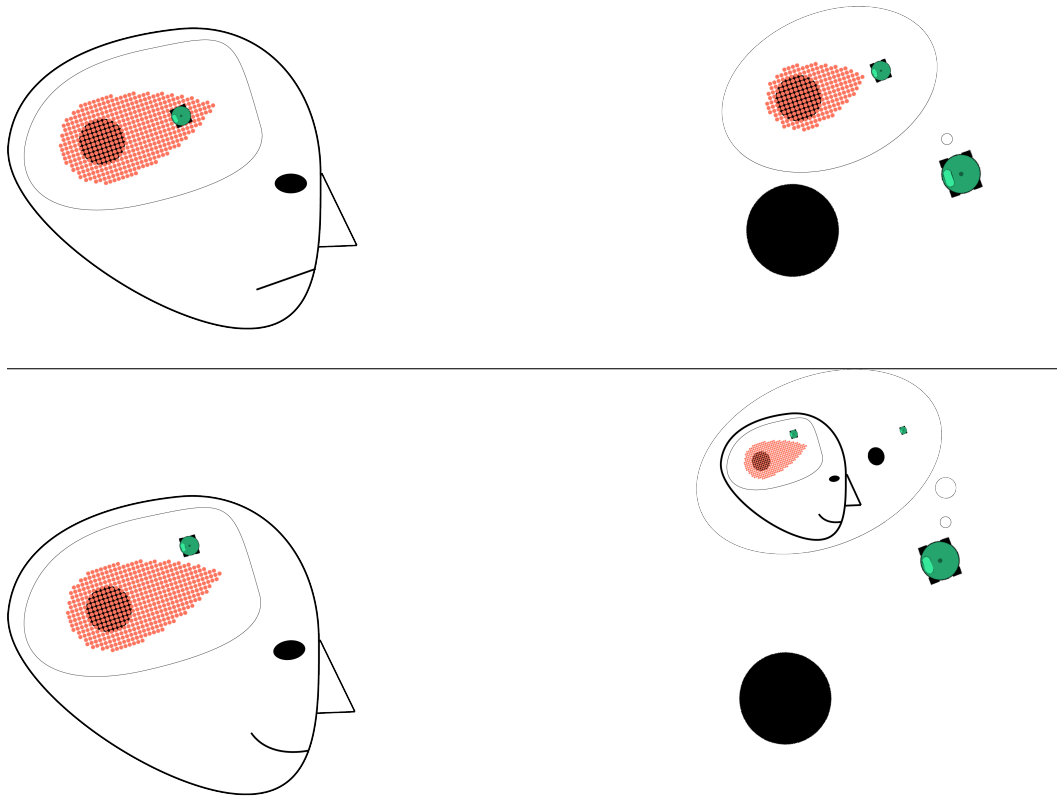


Figure 1.1: Top: if a robot’s behavior does not take into account a human supervisor’s notion of safety, the misaligned expectations can degrade team performance. Bottom: When a robot acts according to a human supervisor’s expectations, the supervisor can more easily predict the robot’s behavior.

model-based framework that interprets the human’s signals as resulting from an optimization problem. This inverse optimization framework has also been used in Inverse Reinforcement Learning [7, 8] which applies Inverse Optimal Control (as conceived of by Kalman [9]) to interpreting human trajectories. Our work applies the inverse optimization framework to learn from the supervisor’s decisions to intervene.

Results in cognitive science suggest that humans observing physical scenes can be modeled as performing a noisy “mental simulation” to predict trajectories [10, 11]. The use of mental dynamical models is also supported by work in neuromechanical motor control which asserts that human action mastery involves building a dynamical model of the human body [12]. Robinson et al. posited that this dynamics learning extends not just to direct control of

the body but to external systems with which the human interacts, such as human-cyber-physical systems [13]. We posit that human supervisors utilize this same cognitive dynamic simulation to predict robot safety and intervene accordingly. Specifically, we theorize that the intervention behavior is driven by an internal “safe set” which we can attempt to reconstruct by observing supervisor interventions.

Safe sets are conceived from the Formal Methods notion of “Viability”. A set of states is “viable” if for every state in the set there exists a dynamic trajectory that stays within the set for all time. Reachability analysis calculates the largest viable set that doesn’t include any undesirable state configurations (e.g. collisions with obstacles, power overloads, etc). Since the set is viable, it is possible to guarantee that the dynamic system will always stay within the set and therefore stay safely away from the undesirable states. For this reason, viability kernels are often referred to as “safe sets”. Reachability can be used for robust path planning in dynamically changing environments [14] or working around multiple dynamic agents [15], and recent results have leveraged the technique to bound tracking error in order to generate dynamically feasible paths using simple planning algorithms [16].

Hoffman et al. used the safety guarantees of reachability analysis to engineer a multi-drone team that could automatically avoid collisions [17]. Similarly, Gillula could guarantee safety for learning algorithms by constraining their explorations to stay within the safe set [18]. Extending this, Akametalu and Tomlin [19] were able to guarantee safety while simultaneously learning and expanding the safe set. All of these controllers supervise otherwise un-guaranteed systems and intervene to maintain safety whenever the system threatens to leave the viable safe set. In this paper, we explore how this intervention behavior is similar to human supervision, and apply this to representing human safety concerns as safe sets in the state space.

Chapter 2

Supervisor Safe Set Control

Based on the success of cognitive dynamical models for explaining humans' understanding of physical systems, we posit that human operators may have some notion of reachable sets which they employ to predict collisions or avoid obstacles. We propose a noisy idealized model to describe the behavior of the human supervisor of a robotic team, and we develop a framework for estimating the human supervisor's mental model of a dynamical system based on observing their interactions with the team. We then propose a control framework that capitalizes on this learned information to improve collaboration in such human-robot teams.

2.1 Preliminaries: Reachability for Safety

Consider a dynamical system with bounded input u and bounded disturbance d , given by

$$\begin{aligned} \dot{x} &= f(x, u, d), \\ x \in \mathbb{R}^n, \quad u \in \mathcal{U} \subset \mathbb{R}^{n_u}, \quad d \in \mathcal{D} \subset \mathbb{R}^{n_d}, \end{aligned} \tag{2.1}$$

where \mathcal{U} and \mathcal{D} are compact. We let \mathcal{U} and \mathcal{D} denote the sets of measurable functions $\mathbf{u} : [0, \infty) \rightarrow \mathcal{U}$ and $\mathbf{d} : [0, \infty) \rightarrow \mathcal{D}$, respectively, which represent possible time histories for the system input and disturbance. Given a choice of input and disturbance signals, there exists a unique continuous trajectory $\xi : [0, \infty) \rightarrow \mathbb{R}^n$ from any initial state x which solves

$$\begin{aligned} \dot{\xi}(t) &= f(\xi(t), \mathbf{u}(t), \mathbf{d}(t)), \text{ a.e. } t \geq 0, \\ \xi(0) &= x, \end{aligned} \tag{2.2}$$

where $\xi(\cdot)$ describes the evolution of the dynamical system [20].

Obstacles in the environment can be modeled as a “keep-out” set of states $\mathcal{K} \subset \mathbb{R}^n$ that the system must avoid. We define the safety of the system with respect to this set, such that the system is considered to be safe at state $\xi(0) = x$ over time horizon T as long as we can choose $\mathbf{u}(\cdot)$ to guarantee that there exists no time $t \in [0, T]$ for which $\xi(t) \in \mathcal{K}$. The task of maintaining the system’s safety over this interval can be modeled as a differential game between the control input and the disturbance. Consider an optimal control signal $\mathbf{u}(\cdot)$ which attempts to steer the system away from \mathcal{K} and an optimal disturbance $\mathbf{d}(\cdot)$ which attempts to drive the system towards \mathcal{K} . By choosing any Lipschitz payoff function $l : \mathbb{R}^n \rightarrow \mathbb{R}$ which is negative-valued for $x \in \mathcal{K}$ and positive for $x \notin \mathcal{K}$, we can encode the outcome of this game via a value function $V(x, t)$ characterized by the following Hamilton-Jacobi-Isaacs variational inequality [21]:

$$\min \begin{cases} l(x) - V(x, t), \\ \frac{\partial V}{\partial t}(x, t) + \max_{u \in \mathcal{U}} \min_{d \in \mathcal{D}} \frac{\partial V}{\partial x}(x, t) \cdot f(x, u, d) \end{cases} = 0 \quad (2.3)$$

$$V(x, T) = l(x).$$

The value function $V(x, t)$ that satisfies the above conditions is equal to $\min_{\tau \in [t, T]} l(\xi^*(\tau))$ for the trajectory with $\xi^*(t) = x$ driven by an optimal control $\mathbf{u}(\cdot)$ and an optimal disturbance $\mathbf{d}(\cdot)$. We can therefore find the set of states $\mathcal{R}_T = \{x \in \mathbb{R}^n : V(x, 0) < 0\}$ from which we cannot guarantee the safety of the system on the interval $[0, T]$, also known as the backward-reachable set of \mathcal{K} over this interval. That is, for all initial states $x \in \mathcal{R}_T$ and feedback control policies $\mathbf{u}(t) = g(\xi(t))$, there exists some disturbance $\mathbf{d}(\cdot) \in \mathcal{D}$ such that $\xi(t) \in \mathcal{K}$ for some $t \in [0, T]$.

If there exists a non-empty controlled-invariant set Ω that does not intersect \mathcal{K} , then we deem this set Ω a “safe set” because there exists a feedback policy that guarantees that the system remains in Ω , and thus out of \mathcal{K} , for all time. It follows from their properties that Ω is the complement of \mathcal{R}_T , and the relationship between \mathcal{K} , \mathcal{R}_T , and Ω is visualized in Fig. 2.1. Within a safe set Ω , the value function becomes independent of t as $T \rightarrow \infty$ [21]. Because we focus on the case where the system is initialized to some safe state $\xi(0) \in \Omega$ and we aim to maintain $\xi(t) \in \Omega$ for all $t \in [0, \infty)$, we simplify notation by defining the terms $V(x) \triangleq \lim_{T \rightarrow \infty} V(x, \cdot)$ and $\mathcal{R} \triangleq \mathcal{R}_\infty$.

One approach to guaranteeing the safety of the system is to apply a “minimally invasive” controller which activates on the zero level set of $V(x)$ [18]. This approach allows complete flexibility of control as long as $\xi(t) \in \text{interior}(\Omega)$, and applies the optimal control to avoid \mathcal{K} when $\xi(\cdot)$ reaches the boundary of

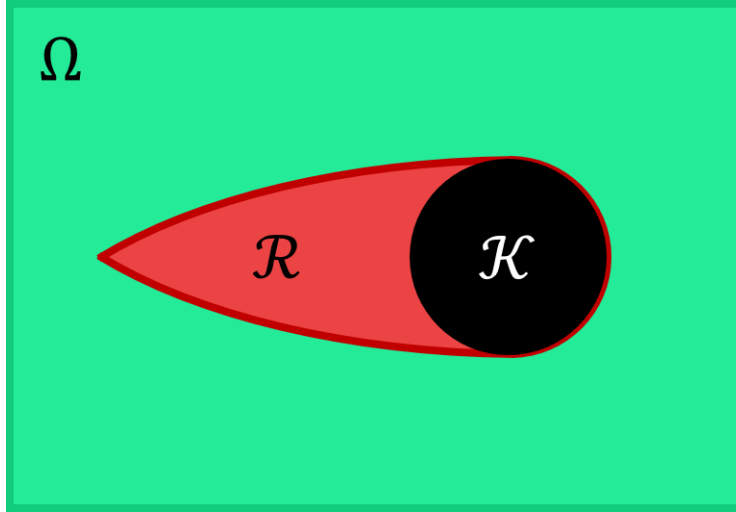


Figure 2.1: Illustration of the relationship between a keep-out set \mathcal{K} , the derived backward-reachable set \mathcal{R} , and the resulting safe set Ω . Note that $\mathcal{K} \subseteq \mathcal{R}$, and Ω is equal to the complement of \mathcal{R} . This illustration approximates the result obtained using the Dubins car dynamics given in (3.1).

Ω . We refer the interested reader to [18, 21] for a more thorough treatment of reachability and minimally invasive controllers.

2.2 Noisy Idealized Supervisor Model

We define an idealized model of the supervisor of a robotic team whose responsibility it is to ensure that no robots collide with obstacles represented by the keep-out set \mathcal{K} . The idealized supervisor behaves as a minimally invasive controller as described in Section 2.1. However, while the robotic team members' true dynamics are given by the function $f(x, u, d)$ as in (2.1), the supervisor possesses an internal model of the robots' dynamics given by $f_S(x, u, d)$, which is not necessarily equal to the true dynamics. Following the differential game characterized by (2.3), the supervisor also possesses an internal value function $V_S(\cdot)$ and safe set Ω_S which they use to evaluate the safety of each state x in the environment. We allow for the possibility that the supervisor adds some amount of margin μ to their internal safe set, such that $\Omega_S = \{x \in \mathbb{R}^n : V_S(x) \geq \mu\}$. Therefore, the idealized supervisor will always intervene when a robotic team member reaches the μ level set of $V_S(\cdot)$, rather than the zero level set of the true $V(\cdot)$. We further specify that the idealized supervisor is *conservative*: $\forall x \in \mathbb{R}^n, V(x) \leq 0 \implies V_S(x) \leq \mu$. This condi-

tion implies that the supervisor will never let a robot teammate leave the true safe set Ω since $\Omega_S \subseteq \Omega$. Additionally, we propose a noisy version of this idealized supervisor: the noisy idealized supervisor will intervene when they observe a robot reach the $\mu + w$ level set of $V_S(\cdot)$, where w is drawn from $\mathcal{N}(0, \sigma_S^2)$ whenever a supervisor makes a safety judgement.

2.3 Learning Safe Sets from Supervisor Interventions

We choose to model the human supervisor of a robotic team as approximating the behavior of the idealized supervisor model presented in Section 2.2. That is, the human supervisor will allow the robots to perform their task however they choose, but intervene whenever they *perceive* that a robot is approaching an obstacle \mathcal{K} in the state space. Given this model, we can interpret the points at which the human intervenes as corresponding to the unknown μ level set of some value function $V_H(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}$, which characterizes the human’s mental safe set Ω_H . Our goal is to use observations of human interventions to derive an estimated value function $\hat{V}_H(\cdot)$ and $\hat{\mu}$ which describe the observed behavior and induce an estimated $\hat{\Omega}_H$. We approach this task by deriving a Maximum Likelihood Estimator (MLE) of the human’s mental safe set. If we assume that a human supervisor always intends to intervene at the μ level set of $V_H(x)$, but their ability to precisely intervene at this level is subject to Gaussian noise, either from observation error or variability in reaction time, then we can consider the value at an intervention point x_i as being drawn from a normal distribution centered at μ (that is, $V_H(x_i) \sim \mathcal{N}(\mu, \sigma^2)$).

Given a proposed value function $\hat{V}_H(\cdot)$ and a set of intervention points $\{x_1, x_2, \dots, x_p\}$ with corresponding values $\{\hat{V}_H(x_1), \hat{V}_H(x_2), \dots, \hat{V}_H(x_p)\}$, we wish to estimate the most likely μ and σ^2 to explain these interventions. Gaussian distributions induce the following probability density function for a single observation $\hat{V}_H(x_j)$

$$f\left(\hat{V}_H(x_j) \mid \mu, \sigma^2\right) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\left(\hat{V}_H(x_j) - \mu\right)^2}{2\sigma^2}\right) \quad (2.4)$$

which leads to the following probability density for a set of p independent

observations

$$\begin{aligned}
f\left(\hat{V}_H(x_1), \dots, \hat{V}_H(x_p) \mid \mu, \sigma^2\right) &= \prod_{j=1}^p f\left(\hat{V}_H(x_j) \mid \mu, \sigma^2\right) \\
&= \left(\frac{1}{2\pi\sigma^2}\right)^{\frac{p}{2}} \exp\left(-\frac{\sum_{j=1}^p \left(\hat{V}_H(x_j) - \mu\right)^2}{2\sigma^2}\right).
\end{aligned} \tag{2.5}$$

The likelihood of any estimated parameter values $\hat{\mu}$ and $\hat{\sigma}^2$ being correct, given the observations and the proposed value function $\hat{V}_H(\cdot)$, is expressed as $\mathcal{L}\left(\hat{\mu}, \hat{\sigma}^2 \mid \hat{V}_H(\cdot)\right) = f\left(\hat{V}_H(x_1), \dots, \hat{V}_H(x_p) \mid \hat{\mu}, \hat{\sigma}^2\right)$. It can be shown that the values of the unknown parameters μ and σ^2 that maximize the likelihood function are given by

$$\hat{\mu}^* = \frac{1}{p} \sum_{j=1}^p \hat{V}_H(x_j) \quad \text{and} \quad \hat{\sigma}^{*2} = \frac{1}{p} \sum_{j=1}^p \left(\hat{V}_H(x_j) - \hat{\mu}^*\right)^2, \tag{2.6}$$

which are simply the mean and variance of the set of observations.

Notice that the estimates given by (2.6) are computed with respect to a given value function $\hat{V}_H(\cdot)$. If we were to assume that the human supervisor has a perfect model of the system dynamics, then we could simply set $\hat{V}_H(\cdot)$ to equal the true $V(\cdot)$ of the system in (2.1), and $\hat{\mu}^*$ would be the maximum likelihood estimate for the level at which the supervisor will intervene. However, it is unlikely that a human supervisor's notion of the dynamics will correspond exactly to this model, and we would like to maintain the flexibility of estimating value functions that are not strictly derived from (2.1). To this end, we define the maximum likelihood of $\hat{V}_H(\cdot)$ being the $V_H(\cdot)$ that produced our observations as $\mathcal{L}^*(\hat{V}_H(\cdot)) = \max_{\hat{\mu}, \hat{\sigma}^2} \mathcal{L}(\hat{\mu}, \hat{\sigma}^2 \mid \hat{V}_H(\cdot))$. The value of $\mathcal{L}^*(\hat{V}_H(\cdot))$ is obtained by substituting the estimates from (2.6) into the probability density function from (2.5). That is, $\mathcal{L}^*(\hat{V}_H(\cdot)) = f\left(\hat{V}_H(x_1), \dots, \hat{V}_H(x_p) \mid \hat{\mu}^*, \hat{\sigma}^{*2}\right)$.

We seek the most likely value function to explain our observations, which will be the value function $\hat{V}^*(\cdot)$ with the greatest maximum likelihood $\mathcal{L}^*(\hat{V}^*(\cdot))$ (the maximum over maxima)

$$\hat{V}^*(\cdot) = \arg \max_{V(\cdot) \in \mathcal{V}} \mathcal{L}^*(V(\cdot)), \tag{2.7}$$

where \mathcal{V} is the set of all possible value functions.

In order to make this optimization tractable, we can restrict ourselves to a set of value functions $\{V_\theta(\cdot)\}_{\theta \in \mathbb{R}^m}$ corresponding to a family of dynamics

functions $\{f_\theta(\cdot, \cdot, \cdot)\}_{\theta \in \mathbb{R}^m}$ parameterized by $\theta \in \mathbb{R}^m$, making the optimization in question

$$\hat{V}^*(\cdot) = \arg \max_{\theta \in \mathbb{R}^m} \mathcal{L}^*(V_\theta(\cdot)). \quad (2.8)$$

In practice, we may not be able to find an expression for the gradient of $\mathcal{L}^*(V_\theta(\cdot))$ with respect to θ , since the value function is derived from the dynamics $f_\theta(\cdot, \cdot, \cdot)$ via the differential game given by (2.3). The lack of a gradient expression restricts the use of numerical methods to solve the problem as presented in (2.8). In these cases, we can compute a representative library of b value functions $\{V_i(\cdot)\}_{i=1}^b$ corresponding to a set of b representative parameter values $\{\theta_i\}_{i=1}^b$ (see Fig. 2.2 for an example library). The optimization then reduces to choosing the most likely value function from among this library

$$\hat{V}^*(\cdot) = \arg \max_{i \in \{1, \dots, b\}} \mathcal{L}^*(V_i(\cdot)). \quad (2.9)$$

In order to ensure that the learned safe set is conservative, we can extend our MLE to a Maximum A Posteriori (MAP) estimator by incorporating our prior belief that, regardless of the safe set that the supervisor uses to generate interventions, they do not want the robots to be unsafe with respect to the true dynamics. In this case, we maintain a uniform prior $P(\theta)$ that assigns equal probability to all $V_\theta(\cdot)$ whose zero sublevel sets are supersets of the zero sublevel set of the true $V(\cdot)$, and zero probability to all other $V_\theta(\cdot)$. In other words, we assume that the supervisor does not overestimate the agility of the robots, and in practice we can enforce this condition by choosing the library in (2.9) to only contain appropriate value functions. Moreover, regardless of the choice of $\hat{V}_H(\cdot)$, we assume that the supervisor intends to intervene before reaching the zero level set of $\hat{V}_H(\cdot)$, which always includes the boundary of \mathcal{K} . If we choose a prior $P(\mu)$ that assigns zero probability to all non-positive μ and uniform probability elsewhere, it can be shown that the MAP estimates are obtained by letting $\hat{\mu}^*$ equal $\max\{\hat{\mu}^*, 0\}$ and otherwise proceeding as before. Fig. 2.3 provides an example of this algorithm estimating a safe set from human supervisor intervention data.

2.4 Team Control with Learned Safe Sets

We propose that safe sets learned according to the approach in Section 2.3 can be used to create effective control laws for the robotic members of human-robot teams. Recall our model of the human supervisor of a robotic team: the

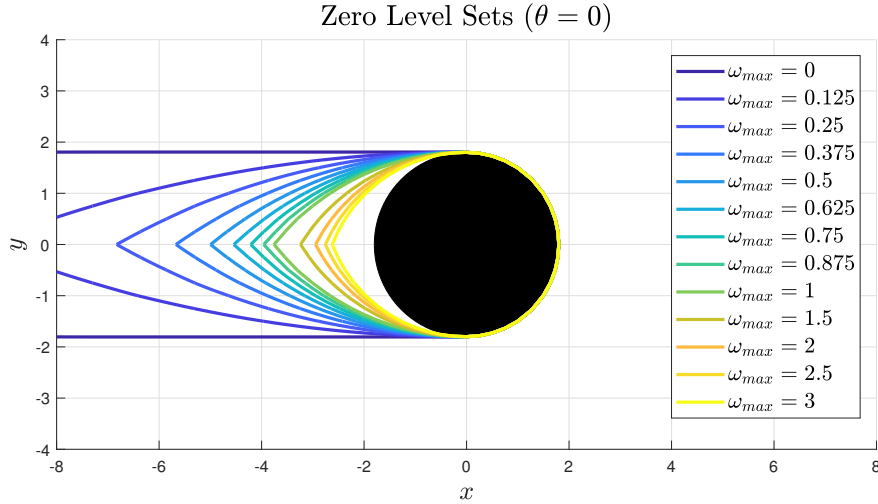


Figure 2.2: Two dimensional slices of the zero level sets of the value functions $V_i(\cdot)$ from the library used for the experiment described in Chapter 3. We used a family of Dubins car dynamics (see (3.1)) parametrized by ω_{max} . Notice that as ω_{max} decreases (the modeled control authority is decreased), the level sets extend farther away from the obstacle, indicating that a robot is expected to turn earlier to guarantee safety.

supervisor must rely on each robot’s autonomy to complete the majority of their tasks unassisted, but the supervisor may intervene to correct a robot’s behavior when necessary (such as by avoiding an imminent collision with the keep-out set \mathcal{K}). We put forth that in the scenario where the human intervenes to prevent a collision, they do so because they observe that a robot has violated the boundaries of their mental safe set Ω_H .

Now, consider a team of robots navigating an unknown environment, and which are able to avoid any obstacles that they detect. One approach to safely automating this team is to have each robot behave according to a minimally invasive control law: the robots are allowed to follow trajectories generated by any planning algorithm, so long as they remain within Ω , the reachable set computed using the baseline dynamics model (2.1) with associated value function $V(\cdot)$. Whenever these robots detect an obstacle, they add it to the

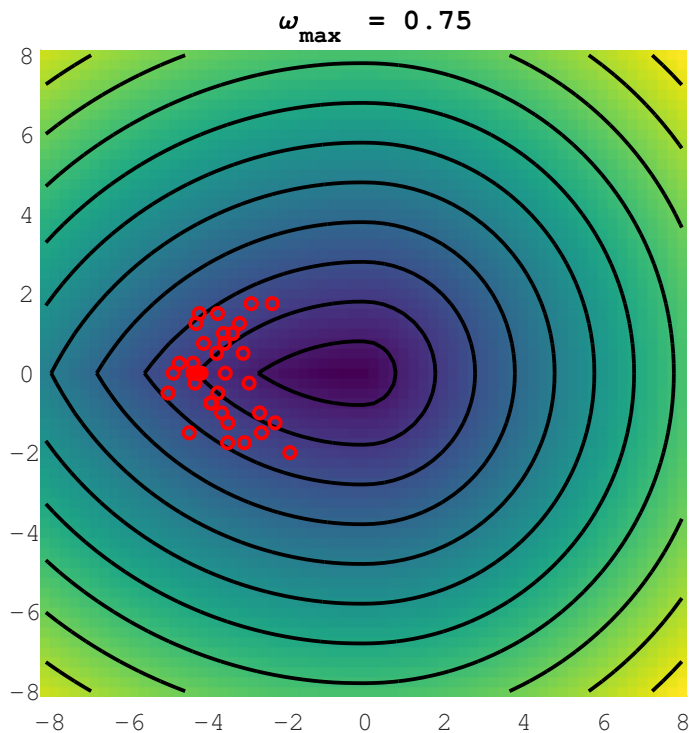


Figure 2.3: An example data set from the experiment described in Chapter 3. The red circles represent the location of supervisor interventions, and the colored background represents the learned value function $V(\cdot)$ with contour lines shown in black. In this case, the learning algorithm chose a dynamics model parametrized by $\omega_{max} = 0.75$.

keep-out set \mathcal{K} , thus modifying Ω and $V(\cdot)$. If a robot reaches the boundary of Ω , it applies the optimal control to avoid \mathcal{K} until it has cleared the obstacle. However, it is possible that a robot does not detect an obstacle, and a human supervisor must intervene to ensure robot safety.

As stated above, the human supervisor will intervene when a robot reaches the boundary of Ω_H , not the boundary of Ω . This discrepancy leads to the possibility that the supervisor will intervene when the robot reaches some state x , even if the robot would have avoided the obstacle without intervention. These situations arise whenever $V_H(x) \leq \mu$ but $V(x) > 0$. These “false positive” interventions represent unnecessary work for the human supervisor, and we seek to eliminate them in order to improve the human’s experience and the team’s overall performance.

We propose using a safe set $\hat{\Omega}_H$ learned from previous observations of supervisor interventions, as outlined in Section 2.3, as a substitute for Ω in the robots’ minimally invasive control law. By estimating the human’s internal safe set, we take advantage of the following property:

Property. *For an idealized supervisor collaborating with a team of robots as described in Section 2.4, if the robots avoid detected obstacles \mathcal{K} by applying an optimally safe control at the boundary of safe set Ω_S , then if the supervisor plans to intervene because they observe $\xi_i(t) \in R_S$ for robot i , the supervisor can infer that robot i has not detected an obstacle and any supervisor intervention will not be a false positive.*

Proof. The proof of this property follows constructively from the definitions of safe set, idealized supervisor, and false positive. If robot i had correctly detected an obstacle and adjusted its representation of Ω_S accordingly, then it would have applied the optimal control to remain within the supervisor’s safe set. Therefore, if the supervisor is able to observe that robot i has left Ω_S , it must be the case that the robot has not detected the obstacle. False positives are defined to be supervisor interventions that occur when a robot has detected an obstacle but the supervisor still intervenes. In this case, the supervisor can correctly infer that robot i has not detected an obstacle, so any intervention at this point cannot be a false positive. ■

For an idealized supervisor, as $\hat{\Omega}_S$ becomes an arbitrarily good approximation of Ω_S , the number of false positive interventions will approach zero. For a *noisy* idealized supervisor, the supervisor will intervene whenever $V_S(x) + w \leq \mu$ where $w \sim \mathcal{N}(0, \sigma_S^2)$. This noise will continue to produce false positives, even with a perfect fit $\hat{\Omega}_S = \Omega_S$, if the robots apply the optimally safe control at the μ -level set of Ω_S . Instead, the level set α where the optimally safe control is applied can be raised arbitrarily high to drive the false positive rate to zero. For example, $\alpha = \mu + 2\sigma_S$ is sufficient to avoid over 97% of intervention states used for learning, in expectation. We test the efficacy of our approach through the human-subjects experiment described in Chapter 3.

Chapter 3

Experimental Design for User Validation

Our goal in understanding and modeling the supervisor’s conception of safety is to improve team performance by decreasing cognitive overload. Although we have based our human modeling on the cognitive science literature, we do not intend to verify humans’ exact cognitive processes. Instead, we aim to apply our inspiration from cognitive science toward building better human-robot teams. To this end, our hypotheses are:

H1. *Representing supervisor behavior as cognitive keep-out sets allows intervention signals to be distilled into an actionable rule which will decrease supervisory false positives and cognitive strain, thereby increasing team performance and trust.*

H2. *Fitting danger-avoidance behavior to a supervisor’s beliefs is preferable to generic conservative behavior.*

In our experiment, we gather supervisor intervention data, fit our model to the data, and then run a human-robot teaming task that assesses performance.

3.1 Procedure

Our experiment applies the idealized supervisor theory and learning algorithm to supervising simulated robots. The robots moved according to the Dubins car model:

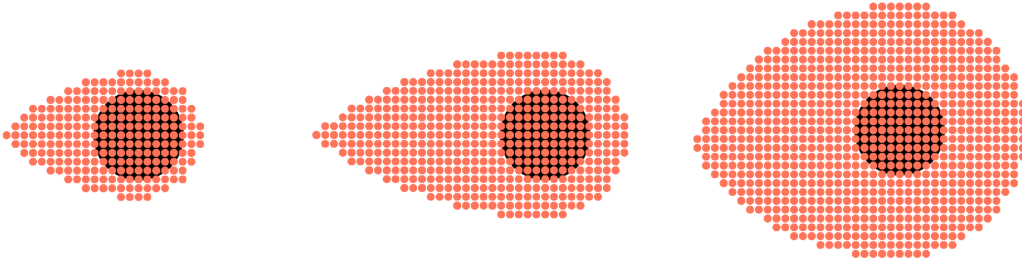


Figure 3.1: Safe sets tested in our experiment (illustrated by their complementary reachable set): (left) Standard safe set (calculated from true dynamics and obstacle size), (middle) example Learned safe set (calculated from fitted supervisory perception of dynamics and obstacle size), (right) Conservative safe set (calculated from true dynamics and inflated obstacle size)

$$\begin{aligned}
 \dot{x} &= 3 \cos(\theta) \\
 \dot{y} &= 3 \sin(\theta) \\
 \dot{\theta} &= u
 \end{aligned} \tag{3.1}$$

$$u \in \mathcal{U} = [-\omega_{max}, \omega_{max}], \omega_{max} = 1$$

The experiment is divided into three phases. In Phase I, the subject is given an opportunity to familiarize themselves with the robotic system’s dynamics. The user is allowed to directly apply the full range of controls through the computer keyboard for one minute. After ensuring the user has some experience from which to build an internal dynamics model, we then assess their emergent conception of safety. In Phase II, supervisory data is extracted from the subject by showing them scenes where the robot is driving towards an obstacle, and the supervisor decides where to intervene to avoid a crash. This intervention data is then fed into our algorithm (described in Section 2.3) that extracts the best fitting safe set. Our estimator used a library of candidate dynamics functions parameterized by values of ω_{max} between 0 and 3, as shown in Fig. 2.2. In this experiment, we enforced conservativeness by excluding subjects whose Learned sets were not supersets of the Standard safe set, rather than enforcing a prior directly on ω_{max} . The Learned safe set is assessed in Phase III against two fixed safe sets (see Fig. 3.1) pre-calculated from the true dynamic equations.

These safe sets were calculated using Hamilton-Jacobi reachability as described in Section 2.1 using the Level Set Toolbox [22] for MATLAB. During this final phase, the subject sequentially supervises homogeneous teams of robots, each team avoiding obstacles based on one of the three assessed safe

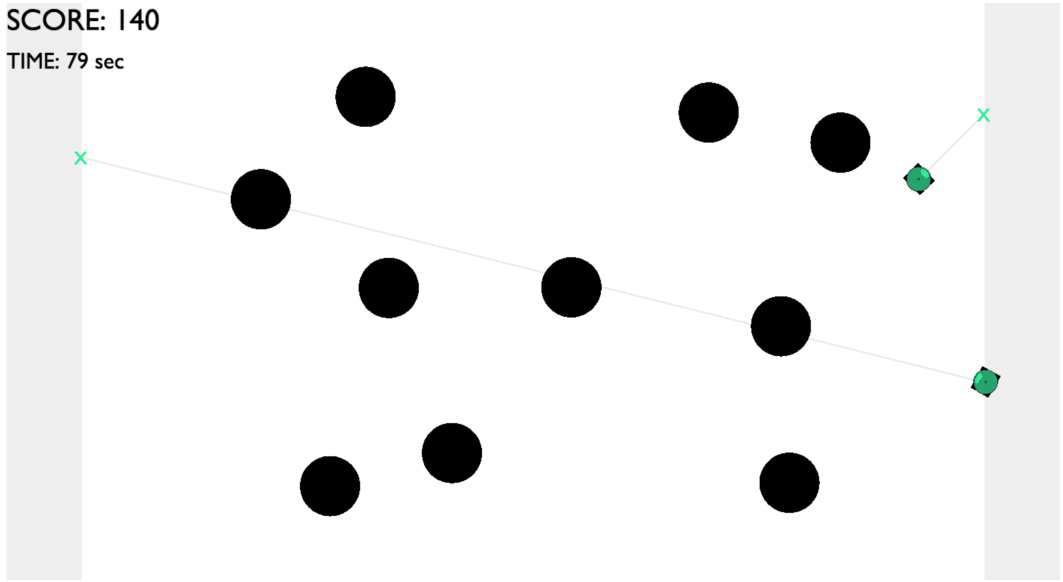


Figure 3.2: Screenshot of the task from Phase III of the experiment. Robotic vehicles make trips back and forth across the screen, detecting and avoiding each obstacle with 80% probability. The human supervisor must remove an obstacle in the event that it is undetected, but must infer this information from the robots' motion.

sets. Ten randomly placed obstacles are strewn about the screen impeding the robots' autonomous trips back and forth across the screen (see Fig. 3.2). Although robots will detect and avoid an obstacles in 80% of their interactions with it, there is a 20% chance that the robot will not detect an obstacle as it approaches. The subject is charged with catching these random failures and removing an obstacle before the robot crashes. Crashing is disincentivized by decrementing an on-screen "score" counter. Removing an obstacle costs only half of what a crash costs the player. This system encourages saving the robot but not guessing wildly. Moreover, simply clearing out all obstacles is not a viable strategy because every obstacle removed generates a new obstacle elsewhere. This score mechanism was also used to make the participant invested in team success by awarding points every time a robot completes a trip across the screen.

3.2 Independent Variables

To assess our hypotheses, we manipulate the safe set used between team supervision trials. We exposed the human subject to three teams, each driving using one of three safe sets. The Learned set is derived from Phase II supervisor intervention observations as described in Section 2, using $\alpha = \mu$. The two baseline kernels are calculated using Hamilton-Jacobi-Isaacs reachability on the true dynamic equations. The Standard set is calculated using the true obstacle size. The Conservative set adds a buffer that doubles the effective size of the obstacle, inducing trajectories that give obstacles a wide berth.

3.3 Dependent Measures

3.3.1 Objective Measures

The team was tasked with making trips across the screen to reach randomized goals. The robots' task was to travel across the screen, safely dodging obstacles along the way, while the human was tasked with supervising as a failsafe to remove an obstacle if the robots should fail to observe and avoid it.

Team performance was quantified using three objective metrics: number of trips completed, number of supervisory interventions, and the number of obstacle collisions. These metrics were presented to the subject as an aggregated, arcade-style score. To incentivize participants to only intervene when necessary, obstacle-removal interventions reduced the score, but only by half as much as an obstacle collision.

The number of interventions taken by the supervisor can also serve as a proxy measurement to quantify the amount of cognitive strain they experience while working with the robotic team. Of particular note are the number of interventions that were not actually required, as the supervisor incorrectly judged that a robot had not detected an obstacle. These false positives needlessly drain supervisor attention and indicate a lack of trust in the system. We aim to increase the human's trust in the system, which we quantify by a decrease in these false positives.

3.3.2 Subjective Measures

After each round of pairwise comparison (completing the task with two different robotic teams), we presented the subject with a questionnaire to gauge how the choice of safe set impacted their experience. These questionnaires

contained statements about each team that subjects would respond to using a 7-point Likert scale (1 - Strongly Disagree, 7 - Strongly Agree). These statements were designed to measure Trust, Perceived Performance, Interpretability, Confidence, Team Fluency, and overall Preference between the teams in the comparison.

3.4 Subject Allocation

The subject population consisted of 6 male, 5 female, and 1 non-binary participants between the ages of 18-29. We used a within-subjects design where each subject was asked to complete all three possible pairwise comparisons of our three treatments (the safe sets used). We used a balanced Latin Square design for the order of comparisons, with no treatment being first in a pair twice. Furthermore, we generated six randomized versions of the task so that subjects were presented with a different version of the task for each trial across the three pairwise comparisons. To avoid coupling the treatment results to a particular version of the task, each treatment was paired with each task version an equal number of times across our subject population.

Chapter 4

Analysis and Discussion

4.1 H1: False Positive Reduction over Standard

Our first hypothesis is that a Learned safe set that reflects the supervisor’s intervention behavior would decrease the number of false positives compared to the Standard safe set. To test this, we performed a one-way repeated measures ANOVA on the number of supervisory false positives from Phase III of the experiment with safe set as the manipulated factor. A false positive was any supervisor intervention where the removed obstacle was actually detected by all nearby robots, which would have avoided it successfully. The robot team’s safe set had a significant effect on the number of supervisory false positives ($F(2, 20) = 8.72, p < 0.01$). An all-pairs post-hoc Tukey method found that the Learned safe set significantly decreased ($p = 0.0328 < 0.05$) false positives over the Standard safe set, but there was no significant difference between the Learned safe set and the Conservative safe set (which also significantly decreased false positives over the Standard safe set, with $p < 0.01$). These results support our main hypothesis that *representing supervisor behavior as cognitive keep-out sets allows intervention signals to be distilled into an actionable rule which will decrease supervisory false positives*.

The second half of that hypothesis, that *decreasing supervisory false positives will increase trust and team performance* was not shown conclusively from our data. We performed a one-way, repeated measures ANOVA on the pairwise comparison surveys between the teams using the Learned and the Standard safe sets. Measures of trust showed no significant improvement

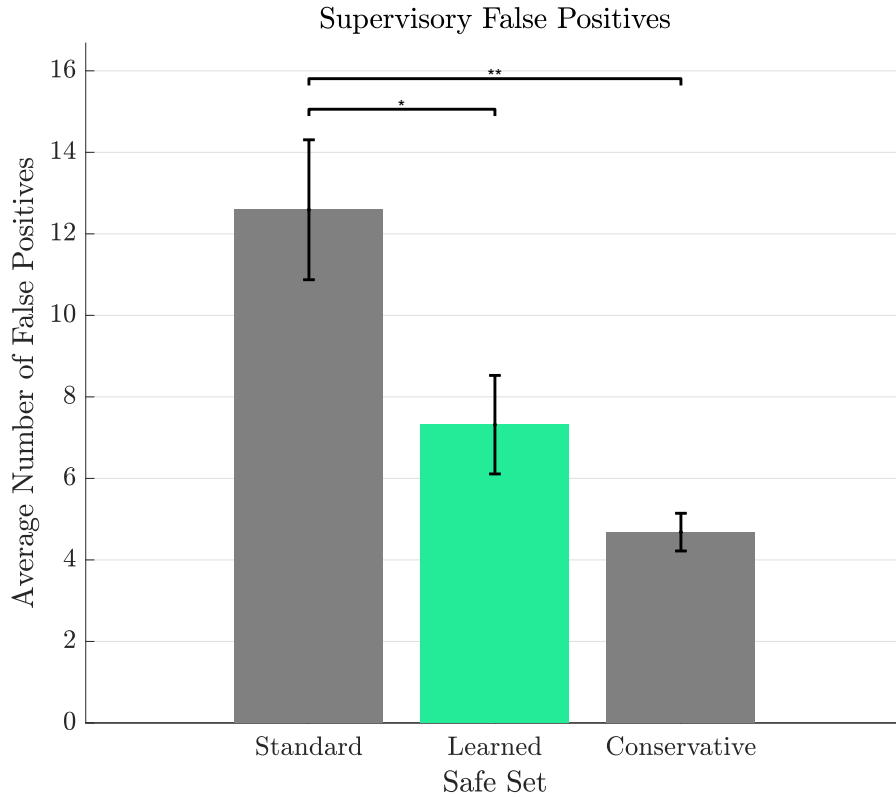


Figure 4.1: Average number of false positives per trial plotted against the three safe set types. There were significant differences between Standard and Learned ($p < .05$) and between Standard and Conservative ($p < .01$). There was no significant difference between Learned and Conservative.

($F(1, 9) = 1.86, p = 0.21$).

4.2 H2: Preference over Conservative

For 9 of 11 participants, the Learned safe set had shorter avoidance arcs than the Conservative set. We hypothesized that this greater efficiency would make the tailored conservativeness of the Learned set preferable to the baseline Conservative safe set. However, a t-test showed that the survey responses for preference were statistically indistinguishable ($p = 0.8$) from a neutral score: an inconclusive result for Hypothesis 2. We believe that this result stems from users judging preference more on intelligibility, the ease of avoiding false pos-

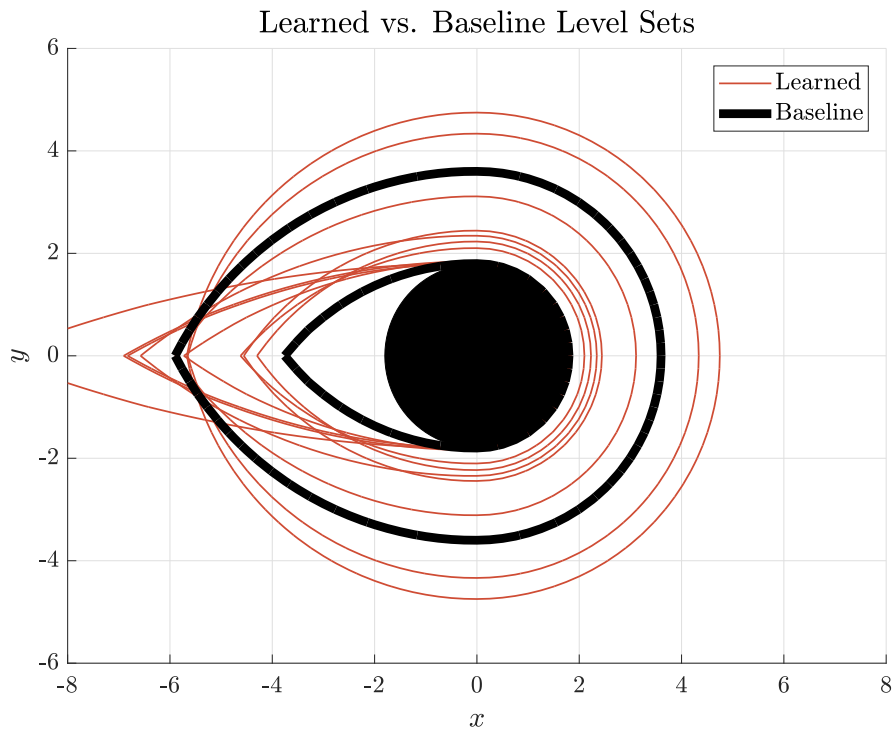


Figure 4.2: Regressed safe sets (viewed on the $\theta = 0$ slice) from supervisor intervention data overlaid on baselines. Three users’ safe sets clustered to arcing like the Standard safe set. Three others clustered to arcing like the Conservative safe set. The final five safe sets exhibit a distinct behavior that reflects supervisors’ preference for gradual, pre-emptive arcs.

itives, than on efficiency, the shortness of paths. As discussed in Section 4.1, both the Learned and Conservative safe sets led to significant false positive reductions over the Standard set.

This indistinguishability is further compounded since a preference for intelligibility seems to be expressed by some subjects in their Phase II intervention data, resulting in their Learned safe sets having similar arcs as the Conservative safe set (see Fig. 4.2). Future work could investigate this efficiency-intelligibility trade-off further by using a conservative baseline that is distinguishably more conservative than user safe sets and by making efficiency more central to the team task.

	Interventions in Safe Set	Predicted F.P. vs Std.	Average F.P.	Observed F.P. vs Std.
Standard	397 / 440	100%	12.54	100%
Learned	220 / 440	55.4%	7.31	58.3%
Conservative	115 / 440	29%	4.68	37.3%

Table 4.1: Predicted and observed false positives. Left: Predicted false positives from Phase II data. Right: Observed false positives in Phase III.

4.3 Model Validity

The statistically significant decreases in false positives observed in Phase III agree with the decreases predicted by the supervisor model based on intervention data from Phase II. Our model posits that interventions occur at states noisily distributed about a safe set boundary. Therefore, it predicts that the empirical distribution of Phase II intervention states contained within a proposed safe set (see Fig. 4.3) will mirror the proportion of false positive interventions observed in Phase III: if states are deemed safe by the controller, they will not be avoided, even when the noisy supervisor would judge them to be unsafe. Since the Learned safe set controller intervenes at the $\hat{\mu}^*$ level set (see Section 2.3), exactly half the intervention states will be contained within the Learned safe set in expectation. The model’s predictions are compared against observed false positives in Table 4.1.

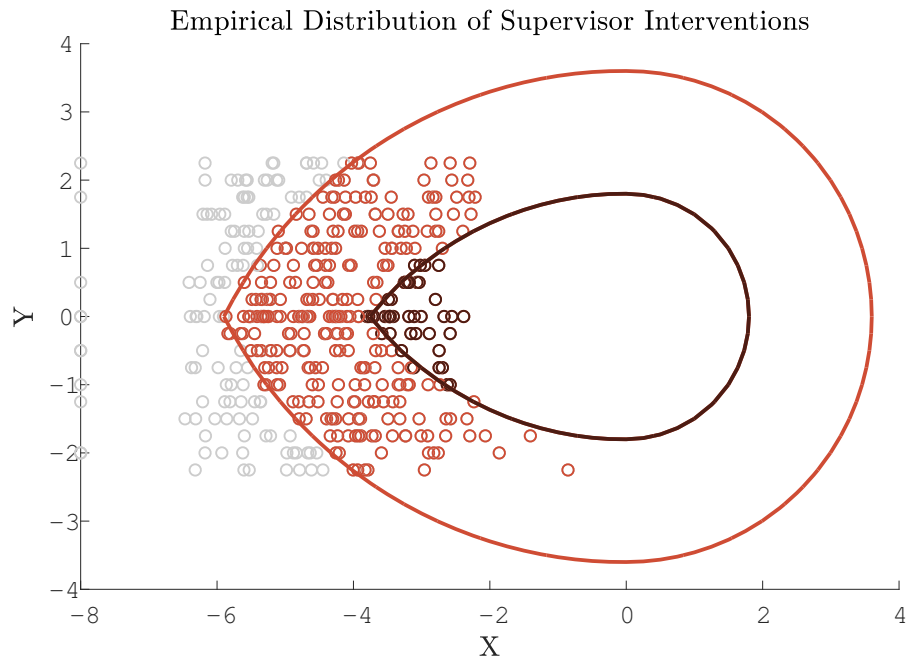


Figure 4.3: Empirical distribution of intervention states observed during data collection (Phase II of the experiment). The interventions within the Conservative reachable set are colored in red, leaving 115 interventions in the corresponding safe set. Similarly, the interventions within the Standard reachable set are colored darker, leaving 397 interventions in the corresponding safe set. Intervention states not contained within a reachable set would have generated a false positive during the human-robot teaming task.

Chapter 5

Conclusion

Automation with human supervisors relies on leveraging the human supervisor’s cognitive resources for success. Respecting these resources is essential for creating well performing human-robot teams. It is especially important to avoid overtaxing the human as automated teams continue to scale up, and a single human worker both accomplishes more and bears more cognitive load than ever. To alleviate this burden, we can decrease the number of issues that command the supervisor’s attention by reducing false positives. By modeling which system states command supervisory attention, we can program autonomous systems to avoid those states when they do not require attention. To capture this information, we combine the concept of mental simulation from cognitive science with formal safety analysis from reachability theory to propose the noisy idealized supervisor model. We employ the noisy idealized supervisor as the generative model in a learning algorithm to predict supervisor safety judgements, and we present a safety controller for robotic agents that respects the supervisor’s perception of safety. This safety controller is guaranteed to reduce false positives for idealized supervisors. Furthermore, for actual supervisors, our human-robot teaming user study demonstrated a significant reduction in false positives when using our approach compared to the standard baseline.

Our results show that it is possible to reduce false positives, and thus cognitive load, by aligning robot behavior with humans’ expectations. Our approach is applicable whenever reachability theory can tractably analyze a dynamical system that will be subject to human safety judgements. Future work will explore the impact of this framework on application domains from air traffic management to self-driving vehicles.

Acknowledgment

Most of the content of this report is adapted from the paper “Modeling Supervisor Safe Sets for Improving Collaboration in Human-Robot Teams” [23]. That work is the result of a collaborative effort with co-lead author David McPherson, who built the experimental platform in addition to his theoretical contributions, Joseph Menke, who contributed key insights in analyzing algorithm performance, Allen Yang, who helped frame the project as part of a broader context, and, of course, our research advisor Shankar Sastry, who provided guidance throughout our research.

This work is supported by the Office of Naval Research under the Embedded Humans MURI (N00014-13-1-0341) as well as a Philippine-California Advanced Research Institutes (PCARI) grant.

Bibliography

- [1] S.-L. Hwang, W. Barfield, T.-C. Chang, and G. Salvendy, “Integration of humans and computers in the operation and control of flexible manufacturing systems,” *The International Journal of Production Research*, vol. 22, no. 5, pp. 841–856, 1984.
- [2] C. J. Tomlin, “Towards automated conflict resolution in air traffic control,” *IFAC Proceedings Volumes*, vol. 32, no. 2, pp. 6564–6569, 1999.
- [3] A. D. Dragan and S. S. Srinivasa, “A policy-blending formalism for shared control,” *The International Journal of Robotics Research*, vol. 32, no. 7, pp. 790–805, 2013.
- [4] S. Javdani, H. Admoni, S. Pellegrinelli, S. S. Srinivasa, and J. A. Bagnell, “Shared autonomy via hindsight optimization for teleoperation and teaming,” *arXiv preprint arXiv:1706.00155*, 2017.
- [5] A. Bajcsy, D. P. Losey, M. K. OMalley, and A. D. Dragan, “Learning robot objectives from physical human interaction,” in *Conference on Robot Learning*, pp. 217–226, 2017.
- [6] A. Jain, S. Sharma, T. Joachims, and A. Saxena, “Learning preferences for manipulation tasks from online coactive feedback,” *The International Journal of Robotics Research*, vol. 34, no. 10, pp. 1296–1313, 2015.
- [7] P. Abbeel and A. Y. Ng, “Apprenticeship learning via inverse reinforcement learning,” in *Proceedings of the twenty-first international conference on Machine learning*, p. 1, ACM, 2004.
- [8] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, “Maximum entropy inverse reinforcement learning,” in *AAAI*, vol. 8, pp. 1433–1438, Chicago, IL, USA, 2008.

- [9] R. E. Kalman, “When is a linear control system optimal?,” *Journal of Basic Engineering*, vol. 86, no. 1, pp. 51–60, 1964.
- [10] P. W. Battaglia, J. B. Hamrick, and J. B. Tenenbaum, “Simulation as an engine of physical scene understanding,” *Proceedings of the National Academy of Sciences*, vol. 110, no. 45, pp. 18327–18332, 2013.
- [11] K. A. Smith and E. Vul, “Sources of uncertainty in intuitive physics,” *Topics in cognitive science*, vol. 5, no. 1, pp. 185–199, 2013.
- [12] R. Shadmehr and F. A. Mussa-Ivaldi, “Adaptive representation of dynamics during learning of a motor task,” *Journal of Neuroscience*, vol. 14, no. 5, pp. 3208–3224, 1994.
- [13] R. M. Robinson, D. R. Scobee, S. A. Burden, and S. S. Sastry, “Dynamic inverse models in human-cyber-physical systems,” in *Micro-and Nanotechnology Sensors, Systems, and Applications VIII*, vol. 9836, p. 98361X, International Society for Optics and Photonics, 2016.
- [14] J. F. Fisac, M. Chen, C. J. Tomlin, and S. S. Sastry, “Reach-avoid problems with time-varying dynamics, targets and constraints,” in *Proceedings of the 18th international conference on hybrid systems: computation and control*, pp. 11–20, ACM, 2015.
- [15] M. Chen, J. F. Fisac, S. Sastry, and C. J. Tomlin, “Safe sequential path planning of multi-vehicle systems via double-obstacle hamilton-jacobi-isaacs variational inequality,” in *Control Conference (ECC), 2015 European*, pp. 3304–3309, IEEE, 2015.
- [16] S. L. Herbert, M. Chen, S. Han, S. Bansal, J. F. Fisac, and C. J. Tomlin, “Fastrack: a modular framework for fast and guaranteed safe motion planning,” *arXiv preprint arXiv:1703.07373*, 2017.
- [17] G. M. Hoffmann and C. J. Tomlin, “Decentralized cooperative collision avoidance for acceleration constrained vehicles,” in *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pp. 4357–4363, IEEE, 2008.
- [18] J. H. Gillula, G. M. Hoffmann, H. Huang, M. P. Vitus, and C. J. Tomlin, “Applications of hybrid reachability analysis to robotic aerial vehicles,” *The International Journal of Robotics Research*, vol. 30, no. 3, pp. 335–354, 2011.

- [19] A. K. Akametalu and C. J. Tomlin, “Temporal-difference learning for on-line reachability analysis,” in *Control Conference (ECC), 2015 European*, pp. 2508–2513, IEEE, 2015.
- [20] E. A. Coddington and N. Levinson, *Theory of ordinary differential equations*. Tata McGraw-Hill Education, 1955.
- [21] J. F. Fisac, A. K. Akametalu, M. N. Zeilinger, S. Kaynama, J. H. Gillula, and C. J. Tomlin, “A general safety framework for learning-based control in uncertain robotic systems,” *CoRR*, vol. abs/1705.01292, 2017.
- [22] I. M. Mitchell, “A toolbox of level set methods,” *Dept. Comput. Sci., Univ. British Columbia, Vancouver, BC, Canada*, <http://www.cs.ubc.ca/~mitchell/ToolboxLS/toolboxLS.pdf>, *Tech. Rep. TR-2004-09*, 2004.
- [23] D. L. McPherson*, D. R. Scobee*, J. Menke, A. Y. Yang, and S. S. Sastry, “Modeling supervisor safe sets for improving collaboration in human-robot teams,” *arXiv preprint arXiv:1805.03328*, 2018.