

Efficient Collaboration with Trust-Seeking Robots

Anqi Xu

Doctor of Philosophy

School of Computer Science

McGill University

Montreal, Quebec, Canada

October 2nd, 2016

A thesis submitted to McGill University in partial fulfillment of the requirements
of the degree of Ph.D.

© Anqi Xu 2016

ACKNOWLEDGEMENTS

I would like to thank foremost my supervisor, Professor Gregory Dudek, for his constant guidance and support throughout my academic studies. His dedications to exploring unconventional ideas, striving for scientific excellence, and gentle yet firm persistence proved to be perpetual sources of motivation.

Additionally, I would like to thank Professors Joelle Pineau and Frank Ferrie for their invaluable advice and insights as members of my Ph.D. progress committee. I am also grateful to Professor Pineau for hosting an information session on academic research opportunities during my undergraduate studies years ago. Her suggestion to seek out Professor Dudek's mentorship during this event ultimately precipitated into my academic accomplishments as well as the contributions in this thesis.

I am proud to be part of the Mobile Robotics Lab (MRL) at McGill University, and I am incredibly grateful to all my fellow MRL friends and colleagues, both past and present. In particular, I would like to thank Florian Shkurti for being a close and ever-dependable friend over the years. I would also like to thank Professor David Meger, who has been a highly knowledgeable and respected role model for me. Furthermore, I would like to acknowledge the entire MRL family, including Professor Ioannis Rekleitis, Isabelle Lacroix, Yogesh Girdhar, Malika Meghjani, Juan Camilo Gamboa Higuera, Sandeep Manjanna, Jimmy Li, David Cortes, Nikolaos Pateromichelakis, Travis Manderson, Arnold Kalmbach, Qiwen Zhang, Andrew Holliday, Emma Akouri, Alexander Chatron-Michaud, Victor Barbaros, Chelsea

Ma, and many more. I am profoundly thankful to each and every one of my fellow MRL comrades, both for helping me countless times over the years, as well as fostering a stimulating, welcoming, and unique lab environment.

I am grateful to all of the volunteered participants in my many user studies and field trials, whether affiliated with the Centre for Intelligent Machines, the School of Computer Science, or McGill University at large. I would also like to thank the Canadian Space Agency, and in particular Tom Lamarche, Éric Martin, and Erick Dupuis, for assisting in my robot field experimentations and providing support for the SL-Commander planetary rover vehicle.

All user studies conducted in this thesis were sanctioned by McGill University's Research Ethics Board (#183-1112). I am a proud member of the NSERC Canadian Field Robotics Network (NCFRN), and I am thankful for the network's generous financial support toward my research.

Finally, I wish to acknowledge Professor Kim Binsted from the University of Hawaii at Manoa for giving a research presentation in 2010 that helped me realize originally the importance and diverse utilities of **trust** in remotely operated systems and human-robot teams.

This thesis is dedicated to my parents, Jing Xu and Manna Lu, for their everlasting and unconditional support, faith, and love.

ABSTRACT

We are interested in asymmetric human-robot teams adhering to a “supervisor-worker” relationship, where the human supervisor occasionally takes over control to aid an autonomous robot agent in its given task. Our research aims to increase and maintain efficient collaboration by improving the robot’s task performance, decreasing the human’s workload, and sustaining high levels of satisfaction. We address this problem through the lens of **trust**, which is pervasive among such human-robot teams, and is inherently linked to all of the above efficiency facets.

Our contributions revolve around a novel “trust-seeking robot framework” that augments an arbitrary robot agent with the abilities to sense and react to the human’s changing trust state. This framework includes a fluid interaction paradigm that enables non-expert users to train and help robot agents adapt to changing task conditions. We also elaborate on multiple large-scale user studies that investigated factors from the interaction experience which influence the rapidly-changing dynamics of trust. Building upon these empirical insights, we propose a personalized, probabilistic model for inferring the human’s moment-to-moment trust state. This trust inference engine extends the two dominant modeling approaches used in the literature, attains greater prediction accuracy compared to several existing techniques, and features the unique ability to update trust beliefs in real time. We further introduce the first-ever realization of robot agents that react in direct response to the human’s trust losses and actively work to restore efficient teamwork. Finally, we demonstrate the diverse efficiency gains of these trust-seeking robots through both extensive controlled experiments as well as challenging real-world field deployments with aerial drones, wheeled robots, and autonomous cars.

RÉSUMÉ

Nous sommes intéressés par les équipes homme-robot asymétriques qui adhèrent à une relation « superviseur-travailleur » où le superviseur humain prend de temps en temps le contrôle du robot pour aider l'agent robotique autonome dans sa tâche. Notre recherche vise à augmenter et à maintenir des collaborations efficaces par l'amélioration de la performance du robot, la réduction de la charge de travail du superviseur, et le maintien d'un niveau de satisfaction élevé. Nous abordons ce problème en utilisant la perspective de la **confiance**, qui est répandue parmi les équipes homme-robot, et est liée de façon inhérente à toutes les facettes précédemment citées de l'efficacité.

Nos contributions sont centrées autour d'un nouveau « système de l'agent robotique fiable » qui améliore un agent arbitraire en lui conférant des capacités de détecter et de réagir à l'état changeant de la confiance de l'humain. Ce système comprend un paradigme d'interaction fluide qui permet aux utilisateurs non-experts d'aider les agents à adapter à leurs conditions de tâches changeantes. Nous présentons également plusieurs études contrôlées à grande échelle pour enquêter sur les facteurs provenant de l'expérience d'interaction qui influencent la dynamique à court terme de la confiance. Équipés de ces connaissances empiriques, nous proposons un modèle probabiliste personnalisé pour inférer l'état de la confiance de l'humain à chaque instant au cours des interactions. Ce modèle de la confiance étend les deux approches de modélisation dominantes utilisées dans la littérature, et atteint des prédictions de l'état de la confiance plus précises par rapport à plusieurs techniques existantes. Ce modèle offre aussi la particularité unique de mettre à jour ces estimations de la confiance en temps réel. En outre, nous introduisons la première

réalisation d'agents robotiques qui réagissent directement aux pertes de la confiance provenant de l'humain, et qui travaillent activement à rétablir une collaboration efficace. Enfin, nous démontrons les divers avantages d'efficacité de ces agents robotiques fiables à travers plusieurs études contrôlées, et nous avons également effectué des déploiements dans des environnements difficiles avec des drones aériens, des robots à roues, et des voitures autonomes.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	ii
ABSTRACT	iv
RÉSUMÉ	v
LIST OF FIGURES	xi
LIST OF TABLES	xv
1 Introduction	1
1.1 The Roles of Trust	3
1.2 Interaction Context: Supervisor-Worker Teams	5
1.2.1 Collaborative Visual Navigation Tasks	8
1.3 Research Methodology: Toward a Framework for Trust-Seeking Robots	9
1.4 Contributions	11
1.5 Statement of Originality	12
1.6 Outline	13
2 Background	15
2.1 Characterization of Asymmetric Human-Robot Teams	15
2.2 Trust	19
2.2.1 Fundamentals of Human Trust	19
2.2.2 Trust Assessment Toward Robots	21
2.2.3 Computational Models of Human-Robot Trust	24
2.3 Trust Elicitation in Human-Robot Teams	25
2.3.1 Interaction Design	25
2.3.2 Interactive Feedback and Learning	27
2.4 Foundations of Trust-Seeking Robots	29
2.4.1 Characteristics of Supervisor-Worker Teams	29
2.4.2 Representation of Human-Robot Trust State	30
2.4.3 Development of Trust-Seeking Behaviors	31
2.5 Autonomous Visual Navigation	31
2.6 Robot Platforms	32
2.6.1 Unicorn Fixed-Wing Aerial Vehicle	33
2.6.2 SightedTurtleSim Aerial Vehicle Simulator	34
2.6.3 Husky Terrestrial Platform	35

2.6.4	SL-Commander All-Terrain Vehicle	36
3	Adaptation from Participation	38
3.1	Problem Formalism	39
3.1.1	Related Problems	41
3.2	Adaptive Parameter EXploration (APEX)	42
3.2.1	Applicability and Prerequisites	42
3.2.2	Algorithm	43
3.2.3	Particle Types	46
3.3	Aerial Coverage User Study with Simulated UAV	47
3.3.1	Participants	48
3.3.2	Infrastructure	48
3.3.3	Procedure	51
3.3.4	Selection of APEX’s Hyper-Parameters	53
3.3.5	Results and Discussion	54
3.4	Campus Patrol Field Evaluation with Husky Wheeled Robot	56
3.4.1	Participants	58
3.4.2	Infrastructure	58
3.4.3	Procedure	60
3.4.4	Results and Discussion	61
3.5	Interactive Driving Field Demonstration with SL-Commander Vehicle	64
3.5.1	Infrastructure	65
3.5.2	Results and Discussion	67
3.6	Summary	70
4	Understanding Human-Robot Trust	72
4.1	Methodology	73
4.1.1	Event-Centric View	73
4.1.2	Trust Factors	74
4.1.3	User Assessment	75
4.2	Experimental Study on Event-Induced Trust Changes	76
4.2.1	Participants	78
4.2.2	Infrastructure	79
4.2.3	Procedure	81
4.2.4	Results and Discussion	82
4.2.5	Predictive Model for Event-Time Trust Changes	87
4.3	Observational Study on Real-Time Trust Dynamics	90
4.3.1	Participants	91
4.3.2	Infrastructure	91
4.3.3	Procedure	95
4.3.4	Results and Discussion	97
4.4	Summary	98

5	Real-Time Human-Robot Trust Modeling	100
5.1	Methodology	101
5.2	Online Probabilistic Trust Inference Model (OPTIMo)	102
5.2.1	Data Preprocessing	103
5.2.2	Local Trust Relationships	104
5.2.3	Inference, Personalization, and Prediction	107
5.2.4	Histogram Inference Engine	109
5.3	Experimental Validation	110
5.3.1	Procedure	111
5.3.2	Characteristics of a Trained OPTIMo Instance	111
5.3.3	Effects of Non-Personalized Parameters	114
5.4	Comparisons Among Temporal Trust Models	119
5.4.1	Procedure	119
5.4.2	Results and Discussion	120
5.5	Summary	122
6	Trust-Induced Behavior Alterations	123
6.1	Trust-Aware Conservative Control (TACtiC)	124
6.1.1	Conservative Control Alterations	125
6.1.2	Trust Triggers	126
6.1.3	TACtiC Algorithm	129
6.2	TACtiC User Study with Simulated UAV	130
6.2.1	Participants	131
6.2.2	Infrastructure	132
6.2.3	Procedure	136
6.2.4	Results and Discussion	138
6.3	TACtiC Field Trial Extension with SL-Commander Vehicle	144
6.3.1	Participants	145
6.3.2	Infrastructure	145
6.3.3	Procedure	147
6.3.4	Results and Discussion	148
6.4	Summary	152
7	Conclusions	154
7.1	Summary of Contributions	154
7.1.1	Interaction Design Guidelines for Supervisor-Worker Teams	156
7.2	Future Directions	157
7.3	Closing Words	160
A	A Vision-Based Boundary Tracking Framework	162
A.1	Boundary Tracking Pipeline	163
A.1.1	Segmentation Stage	164

A.1.2	Boundary Selection Stage	166
A.1.3	Control Mapping Stage	166
A.1.4	Summary of Configuration Parameters	168
A.2	Aerial Coastline Tracking Field Trial	168
A.2.1	Infrastructure	169
A.2.2	Procedure	170
A.2.3	Empirical Analyses	171
A.2.4	Autonomous Flight Results	174
A.3	Summary	175
	REFERENCES	176

LIST OF FIGURES

<u>Figure</u>		<u>page</u>
1–1	Mobile robots in diverse domains that adhere to supervisor-worker relationships	2
1–2	Block diagram of a supervisor-worker human-robot team.	5
1–3	Block diagram of a supervisor-worker team enhanced with trust-seeking agent capabilities.	9
1–4	Mapping of thesis chapters to elements within a supervisor-worker team enhanced with a trust-seeking agent.	13
2–1	Goodrich and Shultz’s classification of levels of autonomy with emphasis on human interaction (replicated from [38]).	16
2–2	Unicorn: a fixed-wing UAV with an integrated GPS autopilot and a gimbal-mounted camera.	33
2–3	SightedTurtleSim: an in-house simulator for aerial robots with downwards-facing cameras.	34
2–4	Husky: an outdoors wheeled robot platform equipped with GPS, IMU, laser range-finder, and a camera.	35
2–5	SL-Commander: an electric vehicle customized for planetary research purposes, with a programmatic drive-by-wire interface and diverse sensors.	37
3–1	Display interface for the Adaptive Parameter EXploration (APEX) aerial coverage user study.	49
3–2	Target terrain boundaries for the APEX aerial coverage user study.	50
3–3	Flowchart for the APEX aerial coverage user study.	51
3–4	Rates of change for discrete and continuous parameters of the robot agent as functions of APEX’s hyper-parameters.	53
3–5	Mean agent rankings from the APEX aerial coverage user study.	55
3–6	A participant supervising alongside the Husky wheeled robot during the APEX campus patrol field trial.	57

3–7	Target terrain boundaries for the APEX campus patrol field trial.	60
3–8	Flowchart for the APEX campus patrol field trial.	61
3–9	Mean agent rankings from the APEX campus patrol field trial.	62
3–10	Individual efficiency metric scores for different agent configurations from the APEX campus patrol field trial.	64
3–11	A user assisting an adaptive agent onboard the SL-Commander during the APEX interactive driving field demonstrations.	65
3–12	The primary test course for the APEX interactive driving field demonstrations.	66
3–13	Snapshots from a session through the primary test course during the APEX interactive driving field demonstrations.	68
3–14	Snapshots of our robot agent adapting to follow wheel tracks during the APEX interactive driving field demonstrations.	69
3–15	Snapshots of our robot agent adapting to a camera pose change during the APEX interactive driving field demonstrations.	69
4–1	Post-session questionnaire interface for the event-induced trust changes study.	76
4–2	Display interface for the event-induced trust changes study.	80
4–3	Flowchart for the event-induced trust changes study.	81
4–4	Prior trust assessments for the event-induced trust changes study. . . .	83
4–5	Effects of event scenarios for the event-induced trust changes study. . . .	84
4–6	Learning curves for the event-time trust changes model $\Delta\mathbb{T}$	89
4–7	Display interface for the real-time trust dynamics study.	92
4–8	Gamepad interface for the real-time trust dynamics study.	93
4–9	Post-session trust feedback answer format for the real-time trust dynamics study.	94
4–10	Flowchart for the real-time trust dynamics study.	96
5–1	Dynamic Bayesian Network structure of the Online Probabilistic Trust Inference Model (OPTIMo).	102
5–2	Window-based preprocessing of interaction factors for OPTIMo. . . .	103

5–3	Sample Conditional Probability Distributions relating latent trust changes to the likelihood of user-reported “trust gained” critiques.	107
5–4	Visualizations of a trained OPTIMo instance.	112
5–5	Effects of window duration W on OPTIMo’s performance.	115
5–6	Effects of histogram bin size B on OPTIMo’s performance.	117
5–7	Effects of trust propagation variability σ_t on OPTIMo’s performance.	117
5–8	Effects of trust feedback uncertainty σ_t on OPTIMo’s performance.	118
6–1	Sample trust sequences and associated trust shift δt_k signals.	127
6–2	Visual interface for the Trust-Aware Conservative Control (TACtiC) user study.	133
6–3	Interim trust feedback questionnaire for the TACtiC user study.	134
6–4	Complete systems diagram for the trust-seeking robot framework.	135
6–5	Flowchart for the TACtiC user study.	136
6–6	Sample interaction experience with a mildly conservative trust-aware agent from the TACtiC user study.	139
6–7	Per-metric efficiency comparisons from the TACtiC user study.	141
6–8	Mean agent rankings across users for the aggregate objective and subjective efficiency criteria from the TACtiC user study.	144
6–9	Passenger layout for the TACtiC field trial extension.	146
6–10	Flowchart for the TACtiC field trial extension.	147
6–11	Per-metric efficiency comparisons for the TACtiC field trial extension.	150
6–12	Mean agent rankings across users for the aggregate objective and subjective efficiency criteria from the TACtiC field trial extension.	151
7–1	Evolution of display interfaces for our interactive boundary tracking robot agent throughout user study and field trial designs.	156
A–1	Sample frames from different robot camera placements.	163
A–2	Block diagram of the boundary tracking pipeline.	164
A–3	Camera placements on the underbelly of the Unicorn UAV.	169
A–4	Analog frames received from the Unicorn UAV’s camera contain significant transmission noise.	170

A–5	Flight trajectory for the coastline boundary tracking field trial.	171
A–6	Inter-human steering discrepancy among 5 volunteers' labels across 3 flight sessions during the aerial coastline tracking field trial.	. .	172
A–7	The boundary tracking agent's steering error compared against inter-human steering discrepancy, as functions of the minimum water ratio and maximum steering change parameters.	172
A–8	The boundary tracking agent's steering error compared against inter-human discrepancies under two parameter configurations.	. .	173

LIST OF TABLES

<u>Table</u>	<u>page</u>
2–1 Classifications for the basis of trust toward different trustee types. . .	20
3–1 Kemeny-Young aggregate agent rankings resulting from the APEX campus patrol field trial.	63
4–1 Summary of major factors relating to event-induced trust changes resulting from stepwise regression.	86
5–1 Personalized parameter values from an OPTIMo instance trained on a single study participant’s dataset.	113
5–2 Comparison of OPTIMo against three temporal trust models.	121
6–1 Performance of trained trust models during the TACtiC user study. . .	140

Chapter 1

Introduction

This thesis is motivated by the desire to establish and maintain efficient collaboration between autonomous mobile robots and their human users. There are a wide variety of robot platforms with autonomous capabilities in common use today, including aerial drones, sea gliders, and household cleaning robots. Manned vehicles are also being equipped with increasingly sophisticated autonomous capabilities, such as adaptive cruise control and auto-steering for cars, as well as automated takeoff, landing, and navigation features for aircraft. These mobile robots all require the collaboration of one or more humans, either remotely or on site, that are responsible for deploying, monitoring, and possibly guiding these robots to complete their tasks, as seen in Figure 1–1.

Research on mobile robots and vehicle automation have been primarily focused on enhancing their capabilities and task performance. Such efforts are essential toward maximizing the efficiency of these human-robot teams. Nevertheless, an equally vital yet often overlooked related aspect is the *interaction* between robots and their human operators. Rich and frequent interactions allow teams to synergize effectively, by coupling the robot’s comprehensive planning and rapid execution capabilities together with the human’s innate problem-solving and decision-making



(a) *Unicorn* Aerial Drone [90]



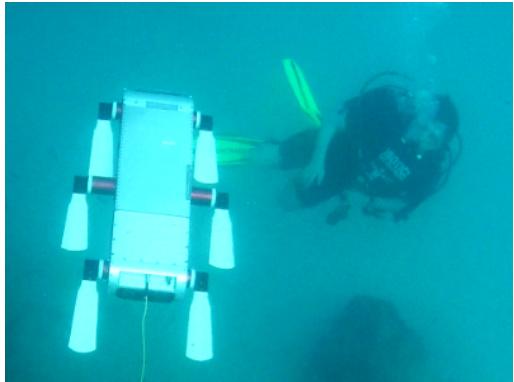
(b) *AR.Drone* Quadrotor [70]



(c) *SL-Commander* Smart Car [10]



(d) *Husky* Wheeled Platform [76]



(e) *Aqua* Swimming Robot [28]



(f) *Heron* Autonomous Vessel [75]

Figure 1–1: Our research aims to establish efficient interactions and teamwork between humans and autonomous mobile robots operating in diverse domains.

skills. Such team synergy is especially important for coping with dynamic or unexpected changes to the task objectives and the environmental conditions. Consequently, our research in this thesis will aim to simultaneously improve task performance as well as enforcing high level of user satisfaction, toward maintaining efficient team collaborations.

1.1 The Roles of Trust

We postulate that *trust* — one’s belief in another’s competence and reliability — is the cornerstone of *all* long-lasting collaboration, both among human teammates, as well as between humans and robots. This view is grounded on the vital roles that trust occupies within the workplace, for instance between a lead engineer and a junior assistant. Strong degrees of trust allow the lead engineer to confidently delegate tasks to the assistant while undertaking other duties in parallel. On the other hand, if the lead distrusts the assistant, then he might be inclined to double-check or repeat the delegated task. In extreme cases, the senior engineer may even cease to delegate tasks altogether and choose to work alone.

These scenarios also occur in human-robot teams that are characterized by similar asymmetric relationships. Imagine an example of a driver is experimenting with the automated lane-keeping feature of her new car. Despite being initially anxious when engaging the autopilot, she slowly builds confidence as the vehicle continues to drive smoothly on the road on its own. Suddenly, another vehicle approaches the left side aggressively, and the self-driving car reacts by abruptly swerving to its right to make way. Although both vehicles continue to operate safely, the autonomous car’s driver, however, is taken back by the sudden jerk motion and thus decides to switch off the autopilot until the aggressive motorist speeds away. This scenario demonstrates that as trust accumulates, the user increases the degree of dependency and task delegation toward the autonomous robot, while in contrast, trust losses can cause the human to disregard the robot in favor of manual execution. Moreover, on an even broader scope, this example illustrates the *essential and pervasive natures of trust* as seen in our everyday interactions with automated systems onboard vehicles and other embodiments.

It is important to acknowledge that trust influences human-robot teams in more ways than one, similar to its influences at the workplace. The previous scenarios

focused on the impacts of trust on the *performance* of an individual’s work. A distinct way that trust affects relationships is by putting into question one’s intentions, ethics, or integrity [42]. These *intention-centric* elements naturally carry over to human-robot teams as well: for instance, the driver in our autonomous car scenario may be concerned that the vehicle’s controller might not *always* make his well-being the highest priority. These concerns are often driven by survival instinct, and some would further argue that these fears are justified based upon age-old philosophical paradoxes such as the trolley dilemma [61]. In particular, researchers have found that while users approved of autonomous cars with utilitarian or “Vulcan”¹ mindsets, they would personally prefer to ride in vehicles that protected passengers at all costs [8]. The complexities of these issues speak to the richness of the notion of trust as seen in human-robot teams.

On the other hand, **this thesis focuses purely on “performance-centric” facets of human-robot trust.** Despite having fewer profound quandaries, we believe that these aspects are more amenable to be modeled by rational and mathematical principles. Consequently, all of our empirical studies included explicit instructions assuring that our robot systems were programmed solely to complete tasks and duties in a subservient and non-adversarial manner.

Returning to the workplace scenario, a conscientious assistant would be able to infer a sense of distrust by observing the lead engineer’s actions, or more specifically, a reduction or lack in task delegation. Consequently, this assistant would then try to change behaviors and improve performance, in order to seek to regain

¹ The fictitious Vulcan race in the Star Trek franchise is known for having logical and utilitarian mindsets, with a famous quote from the popular character Spock: “Logic clearly dictates that the needs of the many outweigh the needs of the few.” Similarly, an utilitarian autonomous vehicle will risk the life of its passengers in collision situations that would otherwise harm a greater number of pedestrians.

the lead's trust (and avoid being re-assigned or demoted). We thus observe that not only is trust useful for *characterizing* the nature of collaboration, the need to seek trust creates additional types of interactions for maintaining healthy teamwork.

Our research capitalizes on the diverse roles of trust to enforce efficient collaboration for asymmetric human-robot teams. We use trust to gauge the quality of existing interactions between human operators and autonomous robots, and also imbue robots with adaptive behaviors for eliciting trust and preserving ongoing teamwork. This “trust-seeking robot methodology” is driven by the core belief that *if humans can remedy their mistakes to regain others' trust, then robot agents should be able to capitalize on this capacity as well!*

1.2 Interaction Context: Supervisor-Worker Teams

This thesis targets two classes of asymmetric human-robot teams, namely remotely operated autonomous robots and manned vehicle with autonomous capabilities. In both classes, the human and robot exhibit a *supervisor-worker* relationship

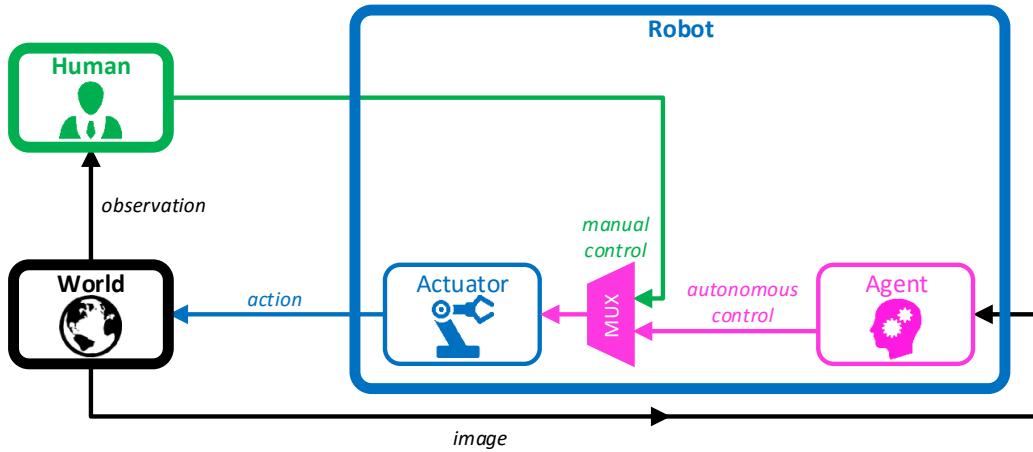


Figure 1–2: In a supervisor-worker team, the autonomous *agent* onboard the mobile robot senses the state of the world (e.g. using a camera) and generates control commands to the vehicle's low-level *actuator*. Meanwhile, a *human* supervises the robot's task performance by observing its behaviors, and can also issue intervening control signals to the robot's actuator. A command multiplexer (MUX) ensures that the supervisor's interventions always supersede the agent's commands.

and interact following a supervisory control scheme [84], as illustrated in Figure 1–2. under typical operations, the *autonomous agent* onboard the mobile robot assumes responsibility for controlling the vehicle’s actuators, toward accomplishing a given task. All this while, the human “supervisor” passively monitors the robot’s actions to ensure that adequate task progression is made. In certain situations, the human may also choose to *intervene* to assist or correct the robot agent, by taking over manual control of the vehicle’s actuators for a period of time. *We assume that both the human and the robot agent are always working collaboratively toward a set of common task goals.*

The “agent” block depicts a generalized encapsulation of one or more autonomous robot systems that ultimately send commands to the vehicle’s low-level actuation controller. These elements can incorporate sensor-based perception units, mapping, localization, and planning modules, machine learning policies, and/or feedback control laws. Examples of such agents include a car autopilot that incorporates data from laser range-finder, camera, and other sensors to generate throttle, brake, and steering commands [92], or an aircraft navigation unit that stipulates desired vehicle velocity and pose to guide it along a set of Global Positioning System (GPS) waypoints. These complex systems often have many configurable parameters, and *our research assumes that each robot agent’s parameters can be programmatically updated to alter its behaviors dynamically*, as a means for giving it new trust-seeking capabilities.

The “actuator” block in Figure 1–2 is an abstraction for low-level controllers that regulate the robot’s motors and actuators. We distinguish these components from the “agent” block since vehicular controllers are typically provided by the robot platform’s vendor and thus are often treated as black-box systems with *predictable responses* and which *do not need end-user tuning*. Furthermore, expressed

in terms of the human’s trust toward the robot, *we assume that the supervisor always has full confidence in the actuator block’s correctness of operations.*

Our research focuses on human interventions that are specifically in the form of control-level commands to the mobile robot’s actuators, as opposed to targets at higher levels of abstraction such as waypoints or task goals. Thus, interactions conforming to our definition of a supervisor-worker relationship can be classified as a subclass of supervisory control that involves “continuous closed loop systems with fast dynamics” [63]. Although limiting the communication modality to such low-level control may appear overly restrictive and mundane, often humans are already familiarized with this form of interaction, as can be seen when the supervisor takes over the steering wheel from a self-driving car agent. More importantly, the availability of a continuous stream of input from the human will turn out to be a vital precursor for the robot agent to sense and thus seek the human’s trust *in real time*, as will be discussed in Chapter 5.

In this interaction context, several types of situations can cause humans to intervene. For instance, the robot agent may sometimes fail to perform its duties to the human’s level of satisfaction. In other circumstances, the supervisor may realize that the agent has reached a limitation in its programming and thus require additional help to overcome challenging task conditions. As an example, an aerial drone may struggle to navigate directly toward its GPS waypoint under strong headwind, and so the human may decide to manually steer the vehicle in a snake-like trajectory to make incremental progress toward the target. Another cause for intervention arises when the supervisor wishes to change the agent’s current task objective by demonstrating actions toward a different task target.

The multiplexer (MUX) block in the diagram is used to ensure that *the supervisor’s intervening commands will always supersede those of the autonomous agent.* As a corollary, we also assume that the *robot will always execute the supervisor’s*

commands without questioning their validity. Consequently, our research focuses solely on exploring the directed trust link from the human supervisor to the robot agent, while studies of reciprocal trust and related concerns such as the safety of the supervisor’s commands [81] are beyond the scope of this thesis.

Aside from interventions, the human may also wish to convey assessments about the agent’s performance throughout operations. Concrete instances include giving *critiques* such as praise or criticism to a recent action or event, and providing *feedback* by verbalizing their current degree of trust in the agent. These communication modalities are seen in workplace settings as well, and they can help the team establish a common understanding and synergize toward common task goals.

1.2.1 Collaborative Visual Navigation Tasks

Our investigations in this thesis are grounded on the task domain of collaborative visual navigation. Despite this focus, we studied human-robot trust from a platform-independent and application-agnostic manner, to accommodate supervisor-worker teams in other task domains as well. With that said, all of our robot agents are built upon a generalized vision-based boundary tracking framework (see Appendix A). To demonstrate the generalized nature of our solutions, we evaluated these agents onboard multiple aerial and terrestrial platforms, and targeted both terrain coverage as well as trajectory patrol tasks.

Visual navigation problems are natural research fits for human collaboration and interaction, since we humans innately excel at many vision-based tasks. Also, since cameras are among one of the most prevalent sensors in robots, using this modality enables our trust-seeking robot agents to be deployed onboard a range of different robot platforms. Furthermore, the necessary sophistication and complexity in autonomous visual navigation solutions [60, 71] can often lead to uncertainties about its internal processes, and therefore naturally warrants the need for the human supervisor to build trust in a robot agent.

1.3 Research Methodology: Toward a Framework for Trust-Seeking Robots

There are multiple ways to imbue autonomous agents with trust-seeking capabilities in supervisor-worker teams. One strategy is to design and constrain the types of interactions between the human supervisor and the robot agent as means to encourage team-building and trust. For instance, adding visualizations of the agent’s perceived view of the world, internal state, and output commands might help the human better understand the agent’s reasoning process and take notice of its capabilities and limitations. By baking such types of *transparencies* [56] into the interface and interaction scheme at design time, the supervisor at deployment time would be able to better identify which task instances are appropriate to delegate to the robot and which conditions require manual assistance.

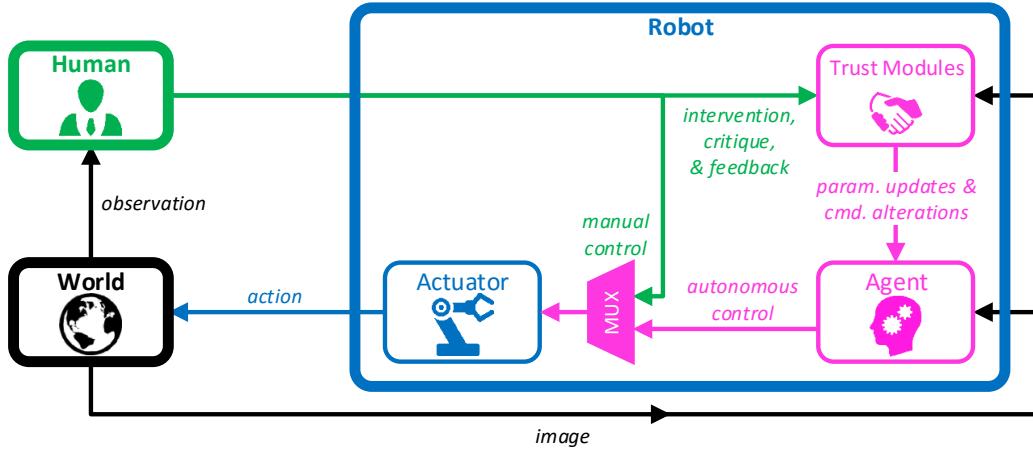


Figure 1–3: Our trust-seeking robot framework enhances a plain supervisor-worker team with *trust modules* serving diverse purposes. These modules update the agent’s parameters to adapt to occasional human interventions, infer the supervisor’s level of trust in the robot based on reactions and feedback, and help the robot agent regain lost trust by altering its behaviors.

As shown in Figure 1–3, another family of approaches entails augmenting the robot agent’s existing programming with new modules to generate behaviors dynamically that incite trust or mend trust losses. As a precursor to such trust-induced behaviors, a robot must first be able to sense or infer the human’s degree of trust

on a moment-to-moment basis. The agent can then alter its behaviors *momentarily* when the supervisor's inferred trust signal drops, for instance to engage the human for assistance during challenging task conditions. The agent can also adapt its actions in a *committal* manner by learning from occasional human interventions to improve task performance as well as build a trusting bond.

These online adaptation strategies make the robot agent react to the human's trust changes *directly and explicitly*. In contrast, the previously discussed design principles such as interface transparency aim to enhance usability, situational awareness, or other factors, which in turn are *indirectly* correlated to greater trust [80].

This thesis will investigate all of the approaches above toward developing trust-seeking robots, including designing and analyzing trust-building interaction schemes, modeling and inferring the human's dynamic trust state, and developing trust-induced robot behaviors. These will involve research efforts in the realms of cognitive Human-Robot Interaction and Machine Learning. We will address several key theoretical problems and technical challenges, including:

1. How to adapt an *arbitrary* existing robot agent to behave similarly to the human's *occasional* intervening control commands, especially when *task conditions and goals change over time?*
2. Which *factors from the interaction experience* influence the supervisor's *trust dynamics*, and what are the *relative importances* of the influences among different factors?
3. How to *infer changes* to the human's trust state *online* so that the robot agent can then react to these changes promptly?
4. How to alter a mobile robot's actions in a *platform-agnostic* way toward *regaining lost trust?*

1.4 Contributions

This thesis will present solutions to each of the problems listed above in turn, namely:

1. **Adaptation from Participation (AfP):** a paradigm that encourages *fluid interactions* and also calls for the robot agent to *adapt to occasional intervening commands* from the supervisor, toward coping to *changing task conditions* as well as encouraging *the building of human-robot trust* indirectly;
2. **Trust factor analyses** of *typical interaction experiences* for supervisor-worker human-robot teams, with a particular emphasis on characterizing trust dynamics at *diverse time scales*;
3. **Online Probabilistic Trust Inference Model (OPTIMo):** a *real-time, personalized* trust model that captures dynamics at *arbitrary time scales* and also attains *superior prediction accuracies* over existing methods;
4. **Trust-Aware Conservative Controls (TACtiC):** a control alteration strategy for the *general class of locomotive-centric robots* that realizes the *first-ever instantiations* of robot agents that *react in direct response to salient trust losses* from their human supervisors.

We tailored the expositions on these interaction paradigms, empirical findings, and software systems to appeal to various types of readers. For instance, we elaborated on the *iterative designs* of several user studies and robot field trials for collecting *extensive empirical evidence* to support our trust analyses. We also demonstrated the application of *Bayesian probability principles* to capture and unify influences from multiple sources of interaction signals into *timely trust estimates* for the supervisor. Furthermore, we provided *end-to-end technical details* for all of our trust-seeking robot agent implementations.

These developments have also led to the following auxiliary contributions:

5. **Boundary tracking agent:** an autonomous visual navigation controller design, with instances deployed on *aerial, terrestrial, and marine robots* (including all of the platforms depicted in Figure 1–1) to guide along *arbitrary boundaries* of visually homogeneous terrains;
6. **Adaptive Parameter EXploration (APEX):** an *online, anytime algorithm* for realizing the AfP paradigm with an *arbitrary* parametrized robot agent;
7. **SightedTurtleSim:** an open-source robot simulator [96] that is integrated with the Robot Operating System (ROS) software ecosystem; this framework simulates *planar-locomotive* robots such as aerial or underwater vehicles, and synthesizes frames from their bird’s-eye view cameras using *real satellite imagery*;
8. **Four large-scale interaction studies** that investigated human-robot trust and quantitatively evaluated our trust-seeking robot agents, involving a total of 112 participants *across 7 Canadian institutions* with diverse backgrounds;

1.5 Statement of Originality

Parts of my research in this thesis have been published in peer-reviewed international conference venues [98–104, 106]. Aside from my own research and development efforts under the guidance of my supervisor Professor Gregory Dudek, several colleagues have also made key contributions as well. Arnold Kalmbach assisted in the developments and deployments of the APEX aerial coverage user study and campus patrol field trials, which will be presented in Chapter 3. Professor David Meger provided a ROS-based software interface for controlling the SL-Commander vehicle, and both he and Qiwen Zhang assisted in the deployment of the APEX interactive driving field demonstration, as will be discussed in Chapter 3. Professor Joelle Pineau provided invaluable guidance in the research that evolved into OPTIMO’s Dynamic Bayesian Network formulation of a real-time human-robot trust model, which will be elaborated on in Chapter 5.

1.6 Outline

Chapter 2 elaborates on background material, including fundamental traits of human trust, research on human-automation trust and human-robot trust, surveys of trust elicitation and learning methods, and overviews of the robot platforms employed in our evaluations. This chapter also enumerates and justifies the key assumptions used to form the foundations of our research.

All of the robot agents in this thesis are built upon a general-purpose boundary tracking controller that we developed. Implementation details on this vision-based robot navigation system are elaborated in Appendix A, in order to focus the main thesis content on trust-related topics. The curious reader should also note that this appendix describes and reflects upon an initial set of robot field evaluations, which historically and personally motivated many of our human-robot trust research thrusts in this thesis.

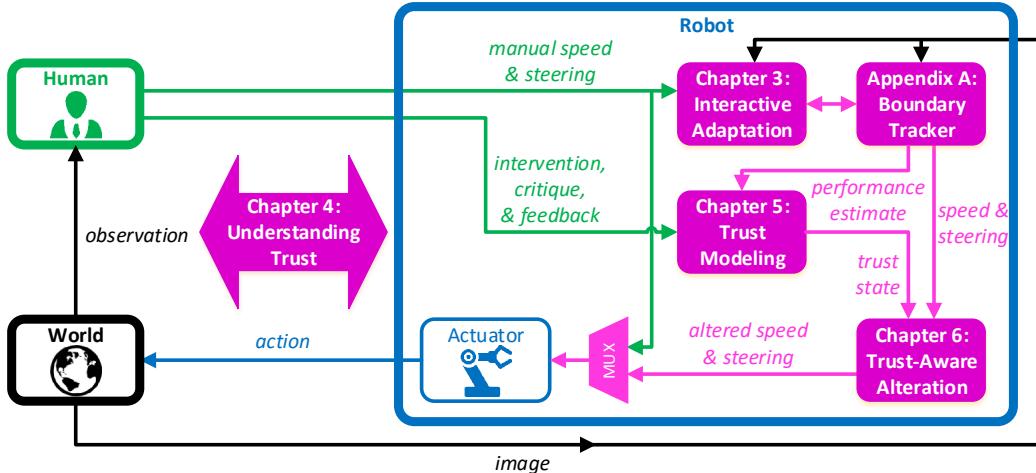


Figure 1–4: Our main research contributions in this thesis entail multiple elements within a supervisor-worker team enhanced with a trust-seeking agent. These module also map to individual thesis chapters (magenta-shaded blocks).

As depicted in Figure 1–4, each of our four primary thesis contributions are discussed in turn in separate chapters. Chapter 3 introduces the Adaptation from Participation interaction paradigm and describes a corresponding module that adapts

the boundary tracking agent’s behaviors dynamically. This scheme allows the agent to imitate intermittent steering signal from the supervisor as an implicit means to seek trust. In contrast, Chapter 4 sows the seeds in a three-stage *explicit* trust-seeking research thrust that starts with a set of interaction studies of typical interaction experiences and trust evolutions for supervisor-worker teams. Chapter 5 uses key insights from these interaction studies to synthesize a personalized and real-time trust inference engine. Armed with this engine, Chapter 6 describes a strategy for our robot agent to react to the supervisor’s inferred trust state, toward remedying situations where the human lost trust and maintaining healthy collaboration. Finally, Chapter 7 discusses consequences of our contributions and highlights follow-up research topics.

Chapter 2

Background

The first half of this chapter begins by elaborating on the characteristics of our supervisor-worker team structure and discussing related interaction contexts within the fields of Human-Robot Interaction and Human-Automation Interaction. We next present an overview of the vast literature on the human notion of trust and also highlight existing trust research in human-robot interaction literature. In particular, this exposition on trust will highlight dominant attributes that are pertinent specifically to supervisor-worker human-robot teams. Next, we survey existing approaches for developing robot systems that seek to establish trusting relationships with their human collaborators. Following these discussions, we summarize the various fundamental assumptions regarding trust and supervisor-worker teams employed by our research.

The second half of the chapter focuses on more practical topics. In particular, we survey related research on autonomous robot systems for vision-based navigation tasks similar to the concrete implementations of our trust-seeking agents. We also elaborate on the diverse mobile robot platforms used to empirically evaluate our research efforts toward realizing efficient trust-seeking robots.

2.1 Characterization of Asymmetric Human-Robot Teams

Among the oldest modes of interaction between human operators and robots (or more generally, automation software) is “teleoperation” or “manual control” [84]. In these systems, the human uses a controller to send signals to the robot’s actuators

while data from its sensors are visualized via a display. In contrast, modern human-automation configurations include a software “agent” that interacts with available sensors and actuators *independent of the human’s control signal*.

One way to characterize types of interaction with these agents is by specifying their “level of autonomy” (LOA), which reflects how much *independent responsibilities* are given to the agent. For instance, Sheridan and Verplank’s LOA scale [85] contrasts agents that recommend actions to the operator to execute, from those that can execute actions on their own and inform the operator, to those that act under full autonomy while ignoring the human. Goodrich and Shultz proposed another LOA scale with an explicit emphasis on human interaction [38]. As shown in Figure 2–1, their classification distinguishes teleoperated systems with little or no mediation by the agent, from relationships where the human supervises and occasionally intervenes in the autonomous agent’s operations, and from teams where the human and robot act independently and jointly toward a common set of task goals.



Figure 2–1: Goodrich and Shultz’s classification of levels of autonomy with emphasis on human interaction (replicated from [38]).

The supervisor-worker relationship that our research focuses on falls under the classification of “supervisory control”. Nevertheless, this term has been defined in very broad terms, for instance referring to “one or more human operators [that] are intermittently programming and continually receiving information from a computer that itself closes an autonomous control loop [...]” [84]. Due to its broadness, researchers have applied this term to refer to many types of interaction schemes.

Endsley and Kaber proposed a distinction between “shared control” from “supervisory control” [30]: in the former case, both the human and computer agent bear responsibility of generating and implementing actions toward accomplishing a given task. On the other hand, in the latter context these duties are delegated to the agent, while the operator occasionally intervenes by selecting alternative actions. Yanco and Drury also described a supervisory role as one in which the human “needs to monitor the behavior of a robot, but does not need to directly control it” [109].

Another limitation of the broad definition of supervisory control is that it does not prescribe the level of abstraction for the commands issued by the human to the autonomous agent. To illustrate this, consider distinct schemes in which a human remotely supervises an aerial drone: for instance, the operator could stipulate a *mission-level* end-goal such as reaching a final destination or carry out aerial coverage of a designated region [105]. Alternatively, the human could specify a set of *task-level* actions such as GPS waypoints for the robot agent to follow [20]. In a third instance, the supervisor could enact *control-level* interventions by teleoperating the vehicle [99].

In this thesis, we define the “supervisor-worker team” as a supervisory control context where human interventions take on the form of control-level commands. This interaction context can also be (less succinctly) characterized as “continuous closed loop systems with fast dynamics” [63]. We specifically targeted this level of interaction as it can be related to rapid fluctuations in the human’s trust state, as we shall elaborate in Chapter 5. Therefore, this type of continuous interaction signal provides a vital opportunity for estimating trust *in real time*, which subsequently enables robot agents to promptly react to changes in their supervisors’ trust state.

The interactions within a supervisor-worker team are also related to the concept of “sliding autonomy”, or equivalently, “shared autonomy”. Sliding autonomy

refers to the ability of an autonomous agent to share control over the robot’s actuators with another individual such as a human or robot partner. Brookshire [9] showed that a human-robot team using sliding autonomy can achieve better task performance compared to either a purely teleoperated system or a fully autonomous robot. Dias *et al.* [25] similarly found that enabling sliding autonomy within a peer-to-peer human-robot team led to an improvement in task completion times and reduced number of mistakes. Dragan and Srinivasa [27] established a unifying formalism of *policy blending* for sliding autonomy systems, where action policies from a human expert and a planning algorithm are combined to produce optimal behaviors. Using policy blending, the authors developed a robot manipulator planner that generates trajectories by estimating the *intent* of human-demonstrated motions.

The systems cited above shared “control-level” autonomy over the robot’s actuators just as in supervisor-worker teams. Other sliding autonomy instantiations have generalized this binary control authority state to multiple levels of autonomy [20, 23]. A closely related concept is that of “mixed-initiative control”, which also encourages the sharing of autonomy, but places further emphasis on whether the human or the robot agent initiates control [11]. In the above systems, the human and/or the agent are given the ability to toggle between different levels of command abstractions dynamically, for instance switching from control-level interventions to task-level commands when tackling simple task conditions. Generalized sliding autonomy instances represent rich and powerful interaction schemes with the capacity to optimize the moment-to-moment control allocation strategy. Nevertheless, *relating commands at higher levels of abstraction to the supervisor’s dynamic trust state is beyond the scope of our research.*

Two other attributes of human-robot interactions that are important to our trust research entail the human-robot *spatial relationship* (i.e. co-located or remote) and *task criticality* (i.e. whether failures affect the human’s physical safety) [109]. As

Chapter 6 will discuss, we observed notably different trust responses and behaviors from supervisors between a user study with a remotely operated aerial vehicle, versus a field trial with an autonomous car transporting the human within.

2.2 Trust

Trust is a highly rich human phenomenon that brings forth myriad interpretations, constructs, and cognitive as well as emotional responses [62]. These elements of trust have been analyzed from many perspectives, such as societal and political views [42], workplace and management aspects [17], as well as for personal relationships [52]. Others have also studied characteristics of an individual's trust within online commerce setting [111], trust toward general information technologies [16, 55], as well as trust toward robots and software automation systems [41, 57]. Given this vast diversity of disciplines studying trust, different proposed theories often have unique aspects that are pertinent to their specific contexts, and may even be at odds with alternative formulations. In this section, we provide a brief overview of the main attributes of trust that are commonly applicable across diverse human contexts, before focusing on human-robot relationships.

2.2.1 Fundamentals of Human Trust

One of the most important points of distinction lies in the two separate roles of trust, as either a utility mapping or an end-state of the human decision process:

- The *degree of trust* is a quantifiable assessment toward another individual,
- whereas the *act of trust* reflects the decision and behavior of relying on an individual's abilities or services.

Both notions are instantiated for a given point in time, and we use the term “trust state” to denote the degree of trust at a particular time instant. Also, both the degree of trust and the act of trust can be related to a specific goal, be it a desire, need, task, or objective [31]. The specificity of trust *toward concrete goals* is of particular importance when reasoning about trust in robots, since typically these

systems are built to accomplish a specific set of tasks. For instance, the notion of placing trust in a vacuum cleaning robot usually implies the task of sweeping the floor. Furthermore, one would generally not share the same degree or act of trust for a different goal. Continuing along our example, we would not likely trust the same vacuuming robot to open doors and greet guests.

The degree of trust measures the amount of a truster’s assessment of a trustee’s abilities. This measure can be affected in two ways: through *direct experiences* or via *delegation*. In the former case, the truster builds confidence by interacting with the trustee and evaluating the quality of this direct engagement. In the latter case, trust delegation [13] refers to the use of assessments from a third-party “witness” to assess a trustee indirectly. When adopting the witness’s recommendations toward the trustee, the truster must further account for his personal assessments of the *witness’ reliability*.

Both types of trust updates can be seen as discrete transactions over one’s historical experiences. Consequently, these updates can be accumulated over the interaction history to characterize the truster’s *prior degree of trust* for a given event. These trust states can be further influenced by personality factors that are not driven by utility-theoretic means, such as society-induced predisposition, or personal faith [62].

Table 2–1: Classifications for the basis of trust toward different trustee types.

	Human Employee [17]	Information Technology [16]	Artificial Agent [31]	Automation System [57]
<i>performance-centric</i>	results capabilities	competence predictability	ability	performance
<i>intention-centric</i>	intent integrity	positive intentions ethics	willingness	process purpose

Another prominent topic studied in social sciences is the characterization for the *basis of trust*. Table 2–1 highlights several proposed categorizations of these bases, while a more exhaustive survey can be found in [57]. When dealing with

a non-human, software-based trustee, applicable bases can be categorized into two broad dimensions: typically, one's trust toward a robot agent or automated tool is based on constructs related to task *performance*, such as their accuracy and consistency in carrying out an assigned task. These concepts are in stark contrast to factors associated with the trustee's *intentions*, which pertain to the software system's general integrity and benevolence toward performing an assigned task.

The trust state is a personal attribute of the truster, and therefore only she will have mental access to the *actual trust state*. Although the truster can introspect and express this cognitive state, these assessments are subject to potential biases and noise resulting from the human decision process and caused by answer formats in questionnaires [15]. Nevertheless, the trustee can maintain an *inferred trust state* by either accepting the reported trust feedback or by combining these with observations of the truster's actions. In Chapter 5, we will demonstrate a concrete instantiation of this trust inference process based on diverse interaction factors that accounts for several forms of noise and biases in their values.

Due to personality-based factors, trusters may form differing opinions about the trustee after experiencing the same sequence of interactions. By eliminating these truster-dependent variations, what remains is arguably an objective assessment of the *trustworthiness* based purely on the trustee's demonstrated performance and interaction experiences. Some of the proposed models for human-robot trust (e.g. [34, 40, 56]) adhere to this causal attribution approach [50].

2.2.2 Trust Assessment Toward Robots

Studies of trust in Human-Robot Interaction (HRI) historically evolved out of a multi-decade literature investigating interactions with automation systems and information technology. Nevertheless, engaging a physically-embodied robot agent is significantly richer than dealing with an online commerce agent or a piece of automation [109]. Many of these robots operate in dynamic and noisy environments

that impose significant difficulties on localizing their positions and assessing their task performance. Such noise, including poor visibility, unexpected obstacles, and intermittent communications, are typically not present in the structured environments of software systems, such as a grammar checker or an email spam filter.

Another crucial distinction is that robots have the potential of inflicting physical damage to tangible objects and the real-world environment. Consequently, human operators tend to behave differently given that they may be blamed for these physical damages, or worse yet if there is a high likelihood that they may be injured.

A common motivation for studying trust toward automation and robots is ultimately to devise techniques for detecting, preventing, and mitigating instances of distrust (i.e. lack of trust) or mistrust (i.e. excessive trust) toward these systems. In extreme cases, such mis-calibrations of the degree of trust in automation have led to fatal accidents, for example in train derailments where the automated alert systems were disabled due to prior false alerts [69].

Many research groups have quantified how a human operator’s trust toward a robot can be affected by a wide range of factors, including task performance and errors [21, 29, 36], the nature of these failures [7, 26, 79], the human’s self-confidence [56], the operator’s mental load [29], and the control allocation strategy [34, 36]. Hancock *et al.* [41] carried out a meta-analysis of empirical results from these and other HRI trust studies to establish quantitative estimates of various factors influencing trust across different interaction domains.

The vast majority of human-robot trust research, including those above, are concerned with “performance-centric” attributes of trust. Arkin *et al.* [4] ventured into the seldom-explored realm of “intention-centric” trust bases by proposing a framework for robot agents to act deceptively via false communication. The authors suggested potential utilities in military contexts as well as health-care robotics (e.g.

deceiving an Alzheimer’s patient to administer treatment), although current work has been limited to theoretical contributions only.

Our investigations into human-robot trust in this thesis are closely related to the work by Desai *et al.* [23, 108], who carried out a multitude of investigations on trust within an urban search-and-rescue setting. The authors quantified the effects on trust from many interaction factors, including the level of autonomy, the reliability of the robot agent, and the situational awareness afforded by the user interface. These efforts culminated in a set of trust-sensible design guidelines for robot agents and interfaces, as well as a comprehensive Human and Autonomous Remote Robot Teleoperation (HARRT) model that relates myriad interaction factors to a trust assessment scale [64]. The authors acknowledged that this regression-based model has limited capability in predicting the human’s trust state dynamically, and have also expressed desires to expand efforts toward real-time trust modeling and trust-calibrating agent behavior modifications. Our research has realized both of these capacities, as will be demonstrated by the online trust inference engine in Chapter 5 and the trust-induced control alteration strategy in Chapter 6.

Finally, several studies (e.g. [24, 41, 56]) have highlighted the influences on the degree of a human’s trust in a robot originating from diverse factors, such as:

- the human’s **demographic attributes**: e.g. age, gender, occupation;
- the human’s **attitudes and experiences**: e.g. propensity to trust robots, prior experience with robots and with task setting;
- the human’s **perception of robot attributes**: e.g. adaptability, benevolence;
- the robot’s **task performance**: e.g. amount of algorithmic failures, frequency of task errors, types of errors;
- attributes of the **interaction setting**: e.g. communication quality, task complexity.

2.2.3 Computational Models of Human-Robot Trust

Diverse representations have been proposed to quantify the degree of trust in a robot agent or a software automation system. In their survey on human-automation trust [63], Moray and Inagaki identified five classes of models:

1. **regression** using trust factors at a fixed time instant;
2. **time series** modeling of trust dynamics;
3. **qualitative** relationships among trust factors;
4. **rule-based** representation of the decision-making process;
5. **neural network** modeling attempts based purely on evidence maximization.

Proposed model representations include binary [40] and continuous [34, 56] measures that characterize the robot’s trustworthiness *caused* by its performance, as well as ordinal scales [47, 64, 68, 82, 107] used to elicit *evidence* of a person’s actual amount of trust. Each representation has its own merits and drawbacks, and there is no “true” model since trust is intrinsically a non-observable construct.

Uncertainty plays a major role in characterizing trust towards robots, as automation researchers have agreed that trust enables collaboration with complex and hard-to-predict agents [40]. Otherwise, lingering uncertainties about the robot’s behaviors and intentions would require immense effort in rational reasoning based on extensive direct experiences alone [13].

A common theme among several trust models is the distinction between mistrust and uncertainty, i.e. knowing with full certainty that an agent will perform poorly is different from inferring that the agent may sometimes misbehave. Depending on the application and context, trust can be modeled against mistrust [51], against uncertainty [31, 40], or against both [49].

Many of the studies above *described* the human’s degree of trust toward the robot through correlations with past interaction experiences and subjective assessments, although few are capable of *predicting* a human’s trust state during operations. Lee and Moray presented a temporal model for relating trust assessments to task performance in a human-automation context, using an Auto-Regressive and Moving Average Value regression approach (ARMAV) [56]. Desai and Yanco [23] conducted a series of robotic search and rescue experiments during which users were asked to report at regular intervals whether their trust state has increased, decreased, or remained unchanged. These signals were quantified as $\{+1, -1, 0\}$ and integrated over time to obtain the Area Under Trust Curve (AUTC) measure. Our research has produced two temporal trust models (see Section 4.2.5 and Chapter 5) using similar approaches. Notably, Section 5.4 will present a quantitative performance comparison of all these models.

2.3 Trust Elicitation in Human-Robot Teams

There are two existing classes of techniques for improving the degree of trust and encourage the act of trust in human-robot teams. One class consists of configuring the interaction scheme with the robot agent at *design time*, following established principles, to facilitate teamwork and mitigate trust-impeding pitfalls. There is also a large corpus of research on robot agents that can learn from their human collaborator as well as provide feedback at *deployment time*, so as to actively improve task performance and elicit greater trust from the human.

2.3.1 Interaction Design

Goodrich and Olsen [37] proposed seven general principles for designing efficient interactions and interfaces for remotely-operated robots. Concrete recommendations include switching between control schemes seamlessly, using natural cues in input and display elements, allowing the human to manipulate interface and

control components naturally, externalizing memory by fusing and displaying historical information, and guiding human attention using visual and audio interface cues. Despite being common sense, all of these recommendations are vital toward enhancing usability and mitigating frustrations.

Desai *et al.* [23, 108] derived a set of observations and guidelines toward improving system performance based on an extensive series of interaction studies with teleoperated search-and-rescue tasks. General themes among recommendations include coping with limited situational awareness, providing selective feedback, reducing task difficulty, factoring long-term interaction effects, enforcing stable and high agent reliability during initial interactions, and accommodating the target audience explicitly during interface design. These guidelines are applicable across a wide range of human-robot teams, and all contribute to enhanced team efficiency and greater trust.

A common finding among multiple studies showed that trust in robots and automation can be improved by providing the human user with *transparent* explanations of the agent’s decision-making process [56]. Dzindolet *et al.* empirically substantiated this principle for a visual target detection task, and showed that users trusted an automated visual aid more after being explained about its limitations (i.e. sometimes it would falsely recognize tree shadows as humans due to similar shapes) [29]. Within a peer-based collaborative search domain, Sanders *et al.* found that users expressed greater trust toward robot agents that communicated using rich modalities (e.g. graphics over audio over text) and that provided constant and verbose feedback [80]. Despite these promising findings, the authors admitted that the results were marginally significant and expressed the need for larger-sized experimentation.

2.3.2 Interactive Feedback and Learning

Continuing with the theme of transparency in the agent’s decision process, Wang *et al.* [94] proposed a method to automatically generate explanations of a robot’s task-level actions by semantically translating the state and output of its probabilistic planner. After carrying out an online study with simulated military search tasks, the authors found that users expressed greater trust when the robot agent provided text descriptions of the rationales and associated confidence levels to explain its actions. This work demonstrated great potential in the use of dynamic feedback from the robot agent to improve transparency and trust, although it would benefit from further investigations, such as real-world deployment results and long-term side-effects of repeated communications. In Chapter 6, we propose a control alteration strategy for the robot agent that similarly conveys hesitation toward seeking to build trust. Also, contrary to operating with discrete semantic task choices (e.g. “search the restaurant down the street”), our technique generalizes to continuous control-level actions.

The topic of Learning from Demonstration (LfD) [2] addresses the transfer of task-domain knowledge from (typically human) experts to robot learners. LfD is related to several other learning problems with equivalent or similar formulations, including Robot Programming by Demonstration (PbD) [6], imitation learning [1, 73], Inverse Reinforcement Learning [65], and Apprenticeship Learning [1]. These learning agents naturally incite greater trust from their human teachers upon successfully learning and imitating their demonstrated actions.

Nicolescu and Matarić presented a LfD technique where a robot learns to complete a given task by observing changes in the world state caused by a demonstrator [66]. This indirect learning approach has the added benefit of allowing a robot

student to learn from either a human or robot teacher. Abbeel and Ng presented a solution to imitation learning using the framework of Markov Decision Process without Rewards (MDP\textbackslash R), and demonstrated the ability for a robot to acquire complex behaviors such as highway driving by observing humans [1]. Chernova developed a similar LfD framework in which a robot, initially with no autonomous capabilities, learned new behaviors by incorporating demonstrated state-action pairs into its policy [14]. A unique aspect of this work was that the robot could request a demonstration when the action recommended by its policy had an insufficient self-confidence value. Ross proposed the Data Aggregation (DAgger) framework to solve imitation learning tasks in an efficient manner, by selectively adding human demonstrations only in situations where the agent’s current policy failed to perform adequately [77]. Knox developed the framework of “Training an Agent Manually via Evaluative Reinforcement”, or TAMER, that incorporated positive and negative critique values from a human observer into a reinforcement learning formulation to interactively shape a robot agent’s behaviors over time [53].

Similar to our Adaptation from Participation (AfP) paradigm in Chapter 3, Dogged Learning is a related interaction scheme that combined concepts from LfD and sliding autonomy [39]. This technique instantiated a robot agent using an online LfD formulation, and then arbitrated between commands produced from this agent, from the human demonstrator, and optionally from a reactive controller via a common measure of confidence. The authors demonstrated empirical results for a degenerate variant of this novel arbitration process, where human interventions were always assigned as full confidence.

2.4 Foundations of Trust-Seeking Robots

We now highlight the core assumptions regarding trust and supervisor-worker teams made in our research. These assumptions help to define and delineate trust-related concepts both for our problem formalisms as well as for participants in our interaction studies.

2.4.1 Characteristics of Supervisor-Worker Teams

Our supervisor-worker team definition in Section 1.2 assumes the following:

1. *Goal-oriented purpose*: both the robot agent and the human supervisor are working toward a single and shared set of task goals;
2. *Agent/actuator separation*: the robot’s high-level sensing, planning, and control modules are decoupled with its low-level actuator interface, and their intermediary channel consist of state-change commands specified with respect to either the robot’s local frame or a fixed world frame;
3. *Parametrized agent*: the autonomous agent onboard the robot has various (discrete and continuous) configuration parameters, which can be altered during operations to change its behaviors;
4. *Human control authority*: all intervening commands from the human are executed by the robot’s actuator block and thus override the control signals generated by its autonomous agent;
5. *Asymmetric relationship*: all software onboard the robot accepts the human’s feedback and carries out the human’s commands without questioning their legitimacy or safety;
6. *Supervisor attentiveness*: the human supervisor is assumed to be actively engaged in the team and continuously attentive to the robot agent’s actions and its task progress.

2.4.2 Representation of Human-Robot Trust State

Our research adheres to a “performance-centric” definition of trust and assumes that the robot agent always maintains good intentions and is never deceptive. This trust definition is shared by numerous other systems and studies in the human-automation and human-robot interaction literature (e.g. [23, 64, 107]). Also, in a supervisor-worker team, the human and the robot agent are working together toward a common set of task goals. Therefore, this setup limits the utility of deception due to the lack of *different task objectives or utility functions*. Furthermore, by focusing solely on performance-centric bases, we can factor out the vast number of intention-centric trust factors and concentrate the scope of our empirical studies on trust dynamics in Chapter 4.

Another attribute of trust follows from our assumptions that the robot agent is always motivated and also always yields to the human’s commands without question. Consequently, we assume that these robot agents always have absolute and full trust in their supervisor’s capabilities. This assumption enables our research to focus solely on the unidirectional trust characterization and quantification from the human supervisor to the robot agent, without worrying about reciprocal aspects.

We chose to represent the human’s trust state t as a continuous value over an closed interval, i.e. $t \in [0, 1]$. This continuous representation captures the *magnitude of change* in the trust state, which is useful for designing trust-induced agent behaviors that are sensitive only to *salient* trust changes. The topic of salient trust-induced behavior alterations will be discussed further in Chapter 6. Another benefit of this representation is the ability to apply continuous-state regression and inference techniques for estimating the supervisor’s trust state. Our probabilistic trust model in Chapter 5 places a belief over this interval scale, which allows it to distinguish between distrust (i.e. low mean value) versus uncertainty (i.e. large variance). Furthermore, having a bounded trust space is helpful in practice when

prompting users to report their trust state during a questionnaire, and facilitates comparisons between trust feedback from different users. Our methodology for trust elicitation will be discussed in Chapter 4.

2.4.3 Development of Trust-Seeking Behaviors

Our realizations of trust-seeking robot agents incorporate many established trust-eliciting design principles [23, 37]. Given their importance and relevance, we consider these design-time elements as standard and mandatory features that should be implemented on all robot agents. In contrast, the main thrust of this thesis revolves around deployment-time approaches for the robot agent to elicit trust by changing its behaviors dynamically.

An important point of distinction among such behavior changes is that some are merely *temporarily altered* in reaction to certain interaction events, while other changes are *adapted in a committal and permanent manner* akin to the accumulation of knowledge. Chapter 3 will propose a computational method for a robot agent to learn from its human supervisor’s occasional intervening commands and adapt its behaviors in a committal way. This formulation assumes that *by successfully imitating the supervisor’s actions, the agent will indirectly gain the human’s trust over time*. Separately, Chapter 6 will present a trust-induced reaction strategy that momentarily alters the robot’s behaviors, to mitigate impacts of recent trust loss as well as signal the supervisor for help.

2.5 Autonomous Visual Navigation

Our research on human-robot trust is carried out through the use of an autonomous agent for tracking visual terrain boundaries. This visual navigation agent draws inspiration from the literature on automated road-following robot controllers. Pomerleau [71] investigated the use of Artificial Neural Networks along with an appearance-based approach toward autonomous visual driving. This work contrasts with approaches using supervised and unsupervised probabilistic models proposed

by Crisman and Thorpe [19], which was also deployed on Carnegie Mellon’s vehicles to detect and track roads. Ma *et al.* [60] proposed a system for tracking curve dynamics in noisy images of roads using an Extended Kalman Filter. Aufrère *et al.* [5] presented a probabilistic model for detecting and tracking lanes on paved roadways within noisy camera frames and demonstrated its implementation within a real-time road recognition system.

As both research and commercial motor vehicles are being equipped with ever more intelligent autopilots [12, 58, 92], there has been an emergence of recent research interests on the problem of predicting driver behaviors and incorporating them into the vehicle’s control and planning systems. Shia *et al.* developed a semi-autonomous controller that can infer the driver’s attentiveness and distraction levels, predict their actions, and intervene when the vehicle is deemed unsafe [86]. Jain *et al.* presented an Advanced Driver Assistance System (ADAS) that fused various information from the driver’s perceived state and the car’s surroundings and warned users of potential imminent danger [46]. The authors conducted extensive assessments on freeway and city driving datasets and demonstrated that their system could accurately anticipate maneuvers up to 3.5 seconds before they occurred. Sadigh *et al.* formalized a dynamical system for a human-driven car that jointly modeled the behaviors of both the autonomous and human agents [78]. This model was also incorporated within a planner that accounted for the actions of the robot affecting those of the human and vice-versa, which was validated using a simulated setup.

2.6 Robot Platforms

Our research places great emphasis on deploying and evaluating the proposed trust-seeking robot agents in challenging real-world conditions and onboard diverse types of mobile robot platforms. This section discusses the main vehicles used in our quantitative interaction studies and field trials.

To facilitate re-use of the algorithms and systems developed in this thesis, we employed the open-sourced Robot Operating System (ROS) [72] to establish a common abstraction layer. ROS marshals content and requests in standard language-agnostic data formats between our software modules, as well as with sensor and low-level control drivers for our different robot platforms. This software ecosystem also boasts an abundance of community-supported tools for logging, diagnosing, and post-processing all of the inter-process communicated content.

2.6.1 Unicorn Fixed-Wing Aerial Vehicle

The Unicorn is a rigid-body fixed-wing plane manufactured by Lockheed Martin Procerus Technologies [90]. As seen in Figure 2–2, this Unmanned Aerial Vehicle (UAV) has a 1 m wingspan built using expanded polypropylene foam, which efficiently absorbs impact upon landing. An electric motor powered by a pair of 3-cell lithium polymer batteries drives this vehicle at average ground speeds of 14 m/s and for durations up to 30 minutes.



Figure 2–2: The Unicorn is a commercial fixed-wing Unmanned Aerial Vehicle with an on-board autopilot microprocessor and gimbal-mounted camera.

The plane’s heading and flight dynamics are regulated by an embedded autopilot unit. This vehicle is equipped with numerous sensors, including a 3-axis accelerometer, a pressure sensor, a magnetometer, and a GPS unit. Communication between the autopilot and the ground control software is achieved via radio frequency. An on-board camera transmits live analog video stream at 30 frames per second via a separate radio frequency. This camera is attached to a gimbal,

which can be actuated via software through the ground control application. The Unicorn can operate in several modes ranging from purely manual control to fully autonomous waypoint-based navigation.

2.6.2 SightedTurtleSim Aerial Vehicle Simulator



Figure 2–3: SightedTurtleSim is an in-house framework for simulating planar-controlled aerial robots with downwards-facing cameras. The environment map can be set to any high-resolution satellite footage or other imagery. Each robot (turtle icons) is shown along with its camera swath (as colored dashed rectangles).

SightedTurtleSim is an open-source ROS framework for simulating holonomic aerial robots and synthesizing frames from their downward-facing cameras [96]. We created this tool to enable our research in this thesis by side-stepping robot deployment challenges such as limited battery and operating range, varied wind and lighting conditions, prohibitive deployment sites, and persistent safety concerns. SightedTurtleSim simulates aerial vehicles as idealized point robots with either fixed linear velocities or throttle and brake input commands to a second-order plant. For simplicity however, angular velocities are confined to constant-altitude

planar motions only. These robots can further be teleported to arbitrary positions and orientations within a bounded environment.

As shown in Figure 2–3, the graphical front-end depicts a map of the environment, icons for all spawned robots, as well as each of their camera swaths. The user can load any image file as the map, and we typically load high-resolution satellite images to produce realistic scenes from the simulated robots’ cameras. All functionalities above can be procedurally triggered via standard ROS messages and service calls, or alternatively enacted through the graphical user interface. Each robot’s locomotive and camera settings are specified at spawn time.

2.6.3 Husky Terrestrial Platform



Figure 2–4: The Husky is a commercially-available wheeled robot for outdoor research. This robot is equipped with assorted sensors, including a GPS receiver, an IMU, a laser range-finder, and a camera with adjustable pitch and roll.

The Husky Unmanned Ground Vehicle (UGV) [76], illustrated in Figure 2–4, is a wheeled platform commercially available from Clearpath Robotics, designed for terrestrial robotics research in outdoor environments. Powered by lead-acid batteries, this vehicle has a typical operating duration of 3 hours, while achieving a maximum speed of 1.0 m/s. Its onboard computer processes and advertises telemetry and low-level control channels through standard ROS interfaces.

This vehicle also serves as a flexible base platform that can house a wide variety of sensors and actuators, including cameras, LIDARs, GPS, and manipulators. Our visual navigation robot agents in this thesis processed frames from a network-connected front-facing camera with adjustable pose and chest-level positioning. This sensor is capable of streaming compression video at 30 frames per second under SVGA resolution (800×600 pixels).

2.6.4 SL-Commander All-Terrain Vehicle

The SL-Commander, as seen in Figure 2–5, is an electric All-Terrain Vehicle (ATV) jointly developed by MacDonald Dettwiler and Associates (MDA), Bombardier Recreational Products (BRP), and Quanser Consulting [67], for the Canadian Space Agency’s planetary research efforts. Its base platform is an electric side-by-side ATV produced by BRP [10], with the capability to traverse terrain ranging from paved roads to undulating back-country trails. The SL-Commander utilizes Ackerman steering and a four-wheel drive with selectable locking differentials, independent suspension, rugged construction and a high-torque propulsion system based around a 48V electric AC induction motor. This vehicle has been designed for a comfortable ride on rough terrain and has many passenger safety features, including a steel roll cage, front and rear ventilated disc brakes, and both on-board and remote kill switches.

A drive-by-wire system onboard the vehicle replicates many of the operations typically conducted by a human driver during manual control, including actuation of the throttle pedal and the steering wheel. Telemetry information, such as wheel odometry, speed, and battery charge, are interfaced via a Controller Area Network (CAN) Bus with an onboard Drive Computer. The Drive Computer is responsible for managing real-time low-level tasks such as steering control, and is also connected to another Smart Computer, which exposes a high-level control interface via the Joint Architecture for Unmanned Systems (JAUS) SAE AS-4 standard [45]. In



Figure 2–5: The SL-Commander ATV is a custom research platform featuring a programmable drive-by-wire control interface and houses many sensors including cameras, a laser range-finder, a GPS unit, and an IMU.

the autonomous control mode, both the steering wheel and throttle pedal are locked from human input, although the driver can nevertheless enforce safe operations at all times via permanent control over the brake pedal and the kill switches.

The SL-Commander is equipped with an impressive slew of sensors, including an industrial-grade laser range-finder, high-precision differential GPS and IMU devices, a high-definition camera with 18x optical zoom mounted on a pan-tilt unit, and two stereo cameras, and individual wheel cameras.

Along with my colleagues Professor David Meger and Qiwen Zhang, we developed wrapper modules for the SL-Commander’s sensor and control interfaces to integrate within ROS. These efforts facilitated the integration of our autonomous controllers with many community-contributed tools. Time-stamped data frames from the vehicle’s cameras and laser range-finder were published using standard ROS message formats and accompanied by meta-data on sensor characteristics. Throttle and turn rate commands were processed by a custom ROS node that interfaced with the Drive Computer’s API. We further modified an existing ROS module to read and control the state of the frontal camera’s pan-tilt unit.

Chapter 3

Adaptation from Participation

Our definition of a supervisor-worker human-robot team calls for the human to help out the autonomous robot via occasional control interventions. Such teams have the potential to tackle challenging tasks, since they combine the heightened dexterity and comprehensive planning capabilities of autonomous agents along with keen instincts and creative problem-solving skills that are innate to humans.

Nevertheless, incorporating a shared control scheme alone does not address some common issues in robot field deployments. For instance, configuring the autonomous agent’s parameters to optimize performance for a given task instance requires laborious data collection and expert knowledge of the agent’s internal workings. Moreover, it can often be mentally straining for the operator to fine-tune parameter settings during deployment, while juggling between task supervision and safety monitoring duties. Yet another class of concerns not addressed by shared control is the need to handle various types of *dynamic events* that affect the moment-to-moment task performance. Examples include pre-determined as well as reactionary task switches, changes in the supervisor’s task intent or preference, varying environmental conditions, and unexpected external perturbations to the robot’s sensors and actuators.

To address all of the challenges above, we propose to augment shared control by enabling the agent to adapt its behaviors interactively, into a paradigm that

we coin as “Adaptation from Participation” (AfP). AfP encourages the human supervisor to focus solely on participating in the team, and without having to worry about configuring or adjusting the agent’s parameters directly. More importantly, the robot agent learns from the human’s intermittent control signal and adapts its parameter settings and behaviors in response, so as to improve task performance on its own and reduce repeated interventions.

This interaction paradigm allows the robot agent to seek for greater trust indirectly by imitating the human’s actions to adopt a common task intent and preference. Behavior adaptation also helps to reduce the likelihood of repeating misbehaviors, thus preventing redundant interventions and the tendency for the supervisor to degenerate into pure teleoperation. We hypothesize that AfP improves the overall efficiency of supervisor-worker human-robot teams, by striving to achieve superior task performance, reduced active human workload, and greater user satisfaction.

This chapter begins by formulating the computational problem of realizing interactive adaptation for parametrized robot agents. We next propose the Adaptive Parameter EXploration (APEX) algorithm to implement AfP in an anytime and parallelized manner. We then assess the potential efficiency gains of the AfP interaction paradigm by empirically evaluating APEX-enabled boundary tracking agents for three separate application domains using distinct robot platforms.

3.1 Problem Formalism

Recalling our supervisor-worker team layout in Figure 1–3, the robot agent can be viewed as a function \mathbb{A} that maps its sensory inputs x , parameter settings θ , and internal state s , into action commands y_r for the robot’s actuator block:¹

¹ The variables $\bar{x}, \bar{y}, \bar{\theta}, \bar{s}$ are generally multi-dimensional vectors, although we omit the overline notations here for simplicity.

$$(y_r, s) \leftarrow \mathbb{A}(x, \theta, s) \quad (3.1)$$

Independent of the agent's operations, the human supervisor can choose to assume manual control at any moment by issuing intervening commands y_h . Importantly, both the human's commands y_h and the robot's commands y_r consist of low-level control signals to the actuator block. Also, the agent's commands y_r are carried out only when the human is not intervening currently, i.e. $y_h \neq \emptyset$.

During periods of human intervention, AfP calls for the agent's parameter settings θ to be continuously adjusted to adapt its behaviors and imitate the supervisor's control signals. Imagine that at each time instant there exists (at least) one *optimal parameter setting* θ^* that will allow the robot agent to exhibit exemplary behaviors, as assessed by the supervisor. Importantly, optimal settings for the agent are altered whenever dynamic events occur, such as changes in task targets, environment conditions, or the supervisor's intent. Although in general these events and optimal values are not observable by the agent, we make the key assumptions that *the supervisor is constantly attentive, will intervene whenever events cause the agent to behave in sub-optimal ways, and will specifically steer the robot to fix these misbehaviors*. Consequently, the agent should adapt its current parameter values θ based on the history of previously seen commands $\{y_h\}$ throughout periods of intervention, and also assume that once the supervisor stops intervening, then θ has adapted to within a tolerable margin of θ^* .

The objective of AfP is to optimize the human-robot team's efficiency, as quantified by complementary aspects of task performance, active human workload, and user satisfaction. These individual measures cannot be combined in general, even though in some domains such as manufacturing, performance and workload can be uniformly quantified in terms of financial gains and costs. Nevertheless, we will

present in Section 3.3.3 an evaluation scheme for the AfP problem that employs non-parametric statistics to assess the aggregate contributions from these multiple facets of team efficiency.

3.1.1 Related Problems

The computational problem of Adaptation from Participation is closely related to the research topic of Learning from Demonstration (LfD). Most notably, both problems involve optimizing the robot agent’s policy parameters based on observed human actions.

Nevertheless, AfP differs from LfD in several key aspects that highlight this problem’s uniqueness and novelty. For instance, conventional LfD realizations take an *episodic* approach to learning, whereas AfP is designed specifically to encourage *online and fluid* interactions. Also, conceptually, in LfD human teachers are asked to focus on *teaching* new tasks to the agent and *correcting* its misbehaviors (e.g. [14]). In contrast, in AfP the human supervisor is responsible for *participating* in the team and focus on successfully carrying out the team’s tasks, without needing to worry about the robot agent’s internal learning process. Furthermore, research objectives of LfD and AfP are complementary, since LfD seeks to learn previously unseen tasks with minimal or *no prior domain knowledge*. On the other hand, AfP aims to improve the moment-to-moment task performance for a robot agent that has a certain degree of *prior task competence*. Finally, perhaps the most important distinction is that LfD assumes a *fixed task goal* and thus aims to learn a stationary optimal policy in a (statistically) unvarying environment. In contrast, we devised the AfP paradigm specifically to *cope with varying task objectives* caused by contextual changes, unexpected environmental disturbances, and evolutions in the supervisor’s intent and preferences.

3.2 Adaptive Parameter EXploration (APEX)

We developed an *anytime* algorithm [112], named as *Adaptive Parameter EXploration* (APEX), that realizes the AfP paradigm and adapts an arbitrary robot agent’s parameters during periods of human interventions. APEX maintains multiple concurrent hypotheses i , called *particles*, of potentially viable agent settings θ_i . During the time interval between consecutive sensor updates, each particle’s parameter value θ_i is refined so as to produce agent commands that closely imitate the supervisor’s intervening commands within a fixed-duration recent window. Since this optimization process is susceptible to overfitting to short-term control signal noise, APEX selects a *winning particle* based on its historical imitation consistency and smooths the refined parameter values into the agent’s settings. This incremental adaptation approach sidesteps the need to predict occurrences of dynamic interaction events that affect the agent’s moment-to-moment optimal settings.

3.2.1 Applicability and Prerequisites

The APEX algorithm assumes that the supervisor is attentive throughout operations and will take over control whenever the agent is misbehaving. Consequently, interventions from the supervisor imply that the robot’s current performance is perceived as sub-optimal and thus requires re-tuning. Chapters 4 and 5 will expand our investigations to account for intervention causes other than sub-optimal observed performance, such as preemptive assistance due to worsening task conditions, and personal propensity for manual control.

We originally designed the APEX add-on module to adapt parameters of arbitrary robot agents in a *black-box* manner, i.e. without altering their internal logic and without given an analytical form for \mathbb{A} . Nevertheless, we assume that each APEX particle can make simulate the command outputs of the agent given arbitrary inputs, state, and parameter values. This ability allows particles to refine parameter values of black-box agents using numerical gradient-based optimization in

the duration between consecutive sensory inputs. *The success of such an iterative approach also depends on the agent’s pipeline having significantly faster execution time than the sensor update interval.*

For practicality, we also considered simple robot agents that have a known *linear inverse mapping*, i.e. $\theta = \mathbb{A}^{-1}(x, s) \cdot y$. Unfortunately, most robot agents are sufficiently complex such that obtaining \mathbb{A}^{-1} is either impractical or ill-posed due to the pipeline’s non-linear or non-bijective nature. Nevertheless, autonomous systems with an accessible \mathbb{A}^{-1} , referred to as *white-box agents*, can rely on linear least squares to efficiently optimize their parameter values given a set of (non-degenerate) input-output exemplar pairs.

APEX’s generic design allows it to refine both continuous and discrete parameter types. When deploying APEX, one must provide judicious parameter ranges to ensure that the agent’s command outputs do not saturate or have near-zero gradients for nearby parameter values. In practice however, we observed that APEX operates well with even loosely-specified parameter ranges, since its particles are designed to search and refine hypotheses within the parameter space.

3.2.2 Algorithm

APEX particles iteratively refine their parameter hypothesis on separate execution threads during the time interval between consecutive sensory inputs. Each particle i is assigned a *long-term cost* \mathcal{C}_i , which is an accumulator term that keeps track of the consistency of the performance of its searched parameter results over time. The main APEX procedure manages these optimization threads and operates before and after executing the agent’s pipeline, as illustrated by Algorithm 1. Whenever a sensory update becomes available, APEX pauses all particles’ optimization threads, updates \mathcal{C}_i based on the quality of each particle’s latest hypothesis, and smooths the lowest-cost winning particle’s value into the agent’s settings. After executing the agent’s pipeline \mathbb{A} , if the human supervisor is still intervening, then

Algorithm 1 Adaptive Parameter EXploration (APEX)

Inputs: initial agent parameters θ and state s

- 1: initialize hypotheses and long-term costs $\theta_i \leftarrow \theta, \mathcal{C}_i \leftarrow 0 \forall i$
 - 2: **loop**
 - 3: wait for new sensor data x
 - 4: **if** particle optimization threads are actively refining θ_i **then**
 - 5: **for all** particles i **do**
 - 6: pause optimization thread i
 - 7: update long-term cost \mathcal{C}_i
 - 8: choose winning particle $i^* \leftarrow \text{argmax}_i (\{\mathcal{C}_i\})$
 - 9: update agent's settings θ using θ_{i^*}
 - 10: store prior state $s' \leftarrow s$
 - 11: execute agent's pipeline $y_r, s \leftarrow \mathbb{A}(x, \theta, s')$
 - 12: **if** $y_h \neq \emptyset$ **then**
 - 13: store latest data exemplar $\{x, s', y_h\}$
 - 14: resume all particle optimization threads
 - 15: **else**
 - 16: $\mathcal{C}_i \leftarrow 0 \forall i$
-

APEX resumes the optimization processes for all particles after incorporating the latest input-state-output exemplar tuple; otherwise, during periods of autonomous control, all particles are disabled, and their long-term costs are also reset.

The optimization thread for each particle i continuously refines its parameter hypothesis θ_i using the W most recent training exemplars $\{x_w, s'_w, y_{h,w}\}_{w=1..W}$, each consisting of a sensor input instance, the robot agent's prior state, and the desired command from the supervisor to imitate. A mean squared cost is used as the optimization objective:

$$\text{cost}(\theta_i) = \frac{1}{W} \sum_{w=1}^W \|y_{h,w} - \mathbb{A}(x_w, \theta_i, s'_w)\|^2 \quad (3.2)$$

Optimization is carried out using iterative gradient-based search for black-box agents and linear least squares for white-box agents that have an inverse mapping \mathbb{A}^{-1} .

As an anytime algorithm [112], APEX updates the agent's settings as soon as new sensory data becomes available, by integrating the refined parameter values from a winning particle i^* . Nevertheless, choosing the winning particle based on

$\text{cost}(\theta_i)$ alone would be short-sighted, as these costs are measured against the W most recent training exemplars only. Instead, these near-term similarity measures are folded into the particles' long-term costs \mathcal{C}_i , which are then used to determine the winning particle based on its historical success at consistently imitating the supervisor's commands in the past:

$$\mathcal{C}_i \leftarrow \gamma \mathcal{C}_i + \text{cost}(\theta_i) \quad (3.3)$$

The discount factor $\gamma \in [0, 1]$ dictates the relative importance of previously accumulated performance versus the latest short-term cost. This hyper-parameter captures the degree of temporal consistency required for a given application context and reduces the likelihood of oscillations between multiple winning particles, which can lead to jittery and sub-optimal agent behaviors.

Once the winning particle i^* is determined, its parameter hypothesis is smoothly integrated into the agent's settings. Whereas discrete parameter values are copied directly, continuous parameters θ^c are smoothed using a learning rate $\alpha \in (0, 1]$:

$$\theta^c \leftarrow \theta^c + \alpha (\theta_{i^*}^c - \theta^c) \quad (3.4)$$

The hyper-parameter α also enforces temporal consistency by attenuating short-term noise in the intervening commands y_h caused by factors such as imprecise human input signals.

Returning to the formulation for the short-term cost, the time window duration W serves multiple purposes in determining the success of APEX in practice. For instance, W places emphasis on only the most recent human commands that presumably address the latest dynamic event during operations. Separately, when deploying the agent on an embedded system with limited computational power, the window duration W can be tuned to improve the quality of gradient-based search.

In particular, changing W balances between the quality (i.e. cost) of each a gradient update to θ versus the number of parameter updates that can be sampled within a fixed time interval. Since W is innately related to the interaction context and task domain, we empirically tuned this hyper-parameter by deploying our boundary tracking agent in test runs and comparing APEX’s online adapted parameters against post hoc batch-optimized parameter values.

3.2.3 Particle Types

Inspired by Monte Carlo sampling techniques such as particle filtering [91], APEX uses multiple particle types to explore the parameter space effectively:

- *Local search particles* use gradient descent line search to iteratively find numerical solutions to locally optimal parameter values for black-box agents;
- *Random restart search particles* are identical to local search particles except that they reset to random initial values each time after executing the agent’s pipeline; they also inherit the latest winning particle’s long-term cost \mathcal{C}_{i^*} ;
- *Inverse optimal search particles* use the linear inverse mapping \mathbb{A}^{-1} of white-box agents within a least squares formulation to directly solve for optimal parameter values;
- A *persistence particle* is used for both black-box and white-box agent instantiations to preserve the previous winning particle’s state and ensure that successive parameter updates are never worse than the existing settings.

The above search strategies are designed to adapt continuous parameters while adhering to a rate of adaptation determined by the learning rate hyper-parameter α . In contrast, categorical parameters are refined by instantiating multiple particles for different combinations of discrete configuration settings. This process allows APEX to determine current discrete configuration settings based on the particles’ long-term costs while enforcing a given rate of adaptation using the discount factor hyper-parameter γ .

3.3 Aerial Coverage User Study with Simulated UAV

We conducted three sets of empirical assessments to validate the hypothesized gains in team efficiency contributed by the Adaptation from Participation paradigm. During this process, we deployed the APEX algorithm to adapt parameters of a vision-based boundary tracking system (see Appendix A for implementation details), and instantiated three interactive adaptive agents onboard distinct robot platforms. This section presents a user study involving APEX-enabled interactive agents on flying robots for performing aerial coverage tasks. This study was carried out under controlled conditions and using a simulated aerial vehicle in order to assess AfP’s contributions independently from practical deployment concerns such as limited operating range and extraneous disturbances. As complement to this controlled evaluation, the next two sections elaborate on field trials that assess APEX-enabled interactive agents in real-world settings, deployed on the Husky wheeled robot and the SL-Commander vehicle respectively.

In this user study, we compared APEX-enabled robot agents against other common human-robot team configurations, namely:

- APEX BB: a black-box APEX-enabled agent that employed 6 *local search particles*, one for each combination of discrete parameter values, as well as 2 *random restart search particles* and a *persistence particle*;
- APEX WB: a white-box APEX-enabled boundary tracking agent equipped with 6 *inverse optimal search particles* and a *persistence particle*;
- CONST: a non-adaptive boundary tracking controller with refined parameter settings that were hand-tuned to ensure competent tracking performance for multiple types of terrain boundaries;
- MANUAL: a baseline configuration where the robot was controlled solely using teleoperation, reflecting situations where the supervisor completely distrusted the agent due to persistent poor performance.

In each study session, the participant collaborated with one of the agent configurations above to control the simulated aerial drone. The objective in each session involved steering the vehicle over a designated sequence of terrain boundaries and covering as much of this flight course as possible within a fixed time limit.

3.3.1 Participants

We recruited 15 individuals (1 female) from the Mobile Robotics Lab at McGill University to engage in this study. Participants were all actively involved in robotics research, and comprised of 5 undergraduate students, 8 graduate students, 1 post-doctorate fellow, and 1 professor. We specifically targeted roboticists since they already work with autonomous robot agents on a regular basis, and they are also likely to be among early adopters of mainstream robotic technologies.

3.3.2 Infrastructure

To enforce repeatable study conditions, we integrated the boundary tracking framework to control the SightedTurtleSim holonomic aerial drone simulator (see Section 2.6.2). Although this robot exhibits idealized vehicular dynamics, frames from its downward-facing cameras were synthesized from satellite footage to provide realistic visual stimuli.

We used the APEX algorithm to tweak configuration settings for the planar-view boundary tracking pipeline. Discrete parameters of this agent comprised of the boundary type $T_b \in \{Edge, Strip\}$ that differentiated between coastline-style contours from road-style boundaries, as well as the appearance type $T_a \in \{Hue, Grayscale, HueValueHybrid\}$, which designated the pixel representation used for segmenting out the terrain of interest in each frame. After detecting the target boundary line in each image, its intersection with the frame borders determined a heading direction to steer the robot along the target boundary. This heading direction is then fed into a Proportional-Derivative (PD) controller, whose control gains K_p, K_d were also regulated by APEX.



Figure 3–1: The APEX aerial coverage user study features a two-pane display interface. The left view shows frames from the simulated drone’s downward-facing camera, overlaid with the detected boundary (blue line) and the agent’s/supervisor’s angular rate steering commands (blue/green arrows). The right map view depicts covered/missed portions of the flight course (green/cyan regions), the robot’s current position (yellow square near the lower-left corner), and out-of-bound areas (red-tinted regions).

The display interface for this study is shown in Figure 3–1, and integrates a live camera feed from the aerial robot, a mini-map of the designated coverage course, as well as information about the current session’s goals. The map view is helpful for visualizing overall coverage progress, although the zoomed-in camera view provided close-up visual details to help users steer along terrain boundaries reliably. This camera view also incorporates overlays reflecting the state of the boundary tracking process, such as the detected boundary line in the current camera frame, as well as steering commands from both the autonomous agent, y_r , and from the human supervisor, y_h .

We designed the boundary tracking agent to continue processing camera frames even during periods of manual intervention, and to visualize its generated steering

arrows on screen at all times. This constant feedback aids the supervisor in deciding if the agent is capable of tracking the target boundary on its own, or if further assistance is needed when the tracker is behaving poorly.

User are provided with a standard dual-joystick gamepad (Sony DualShock®3) to interact with the boundary tracking agent. The supervisor could engage intervening control at any time by holding down a shoulder button and then manually steer the vehicle by moving the left analog stick horizontally.

Each interaction session is structured in a game-like manner, and incorporates both a coverage score as well as a time limit. These gamification elements are designed to motivate participants to remain attentive and enforce adequate tracking performance [95]. Also, we empirically tuned task conditions such as the camera's field of view and the robot's forward speed and maximum turn rate to provide challenging experiences and motivate users to delegate the tracking task to the agent.

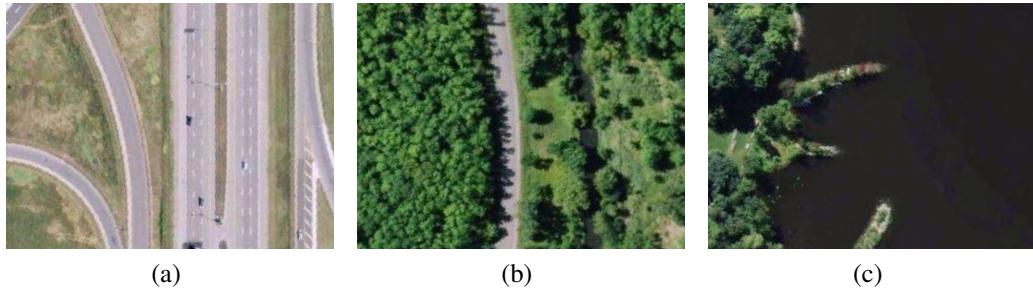


Figure 3–2: The flight course for the APEX aerial coverage user study incorporated terrain boundaries with varying degrees of tracking difficulty, including a straight *Highway* (a), a narrow *Forest Path* (b), and a curvy *Coastline* (c).

Furthermore, the designated flight course incorporated multiple boundary targets with varying degrees of tracking difficulty. As illustrated in Figure 3–2, these terrains comprised of a straight *Highway* with many competing visual boundaries, a narrow *Forest Path* with significant tree cover, and a curvy *Coastline* segment. This final boundary target is particularly challenging to track for both humans and

agents since it requires aggressive steering to keep up with the coastline's constantly changing landscape.

This study is executed in a fully automated manner to enforce uniform user experience and remove experimenter bias. At the start of each session, the simulated robot is displaced to its starting position, and the user can then initiate operations by pressing a button on the gamepad. During subsequent interactions, the on-screen interface warns the user whenever the robot deviates from the designated path. If the participant does not react to these warnings and steer the vehicle back on track promptly, then the session is automatically reset to its starting state.

3.3.3 Procedure

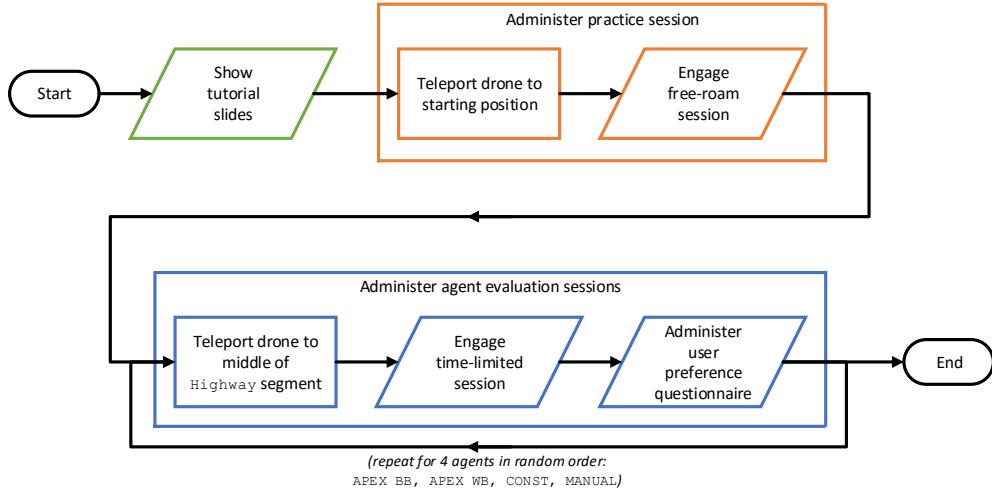


Figure 3–3: Flowchart for the APEX aerial coverage user study.

The flowchart for this study is shown in Figure 3–3. Each study run begins with a brief presentation that explains elements of the interface, the human-robot interaction context, and the adaptive boundary tracking agents being evaluated. This overview is followed by a non-timed practice session to familiarize the user with the control interface and task conditions. Subsequently, the agent evaluation phase of the study consists of 4 aerial coverage sessions, each with a time limit of 3 minutes and featuring one of the aforementioned human-robot team configurations. The order of these configurations is randomly determined for each study instance.

This study compares the overall efficiency of the adaptive agents against baseline human-robot team configurations. In particular, we compute several evaluation metrics over the interaction experience at the session-wide level, as well as for individual terrain boundary segments. The overall task completion is reflected by the *coverage score*, which represents the amount of the flight course visited by the robot’s camera swath. The moment-to-moment boundary tracking accuracy is obtained by analyzing the traveled path and computing the *mean distance to ground truth* statistic. Furthermore, the autonomous agent’s reliability is measured by the *agent failure ratio*, i.e. the fraction of frames rejected due to detection pipeline failures. All three performance metrics above are important for determining the level of overall team efficiency. Another analogous metric is the *supervisor intervention ratio*, which ascertains the fraction of frames under manual control, and reflects the amount of active human workload incurred. Finally, *user preference* ratings were solicited after each session, via a 5-point Likert scale [59], to gather subjective assessments toward each robot agent.

Each metric highlighted above quantifies a distinct aspect of team efficiency and are all essential in forming a thorough evaluation of the AfP interaction paradigm. Nevertheless, it can be challenging to determine an overall efficiency ordering of the human-robot team configurations using these diverse metrics. For instance, if we wanted to compute a linear aggregate score, we would need to perform extensive empirical analyses to determine relative weightings for each metric. Separately, we anecdotally observed a wide range of user behaviors that were consistent with varied tolerances and emphases on the separate aspects of team efficiency. For instance, some users frequently intervened to ensure that the robot was always flying above the target boundaries, while others tolerated minor deviations made by the agent as long as part of the terrain boundary remained in view.

We addressed these concerns by adopting a non-parametric statistical approach for establishing the aggregate agent ordering. Specifically, we compute mean rankings of the four robot agent configurations by averaging across per-metric orderings. We also use the Friedman test [35] to identify the presence of statistically significant preferences within these mean rankings. These findings are then refined using the post hoc Nemenyi test [22] to determine the identity of the preferred agent configurations. Furthermore, to corroborate these ranking results, we also compute aggregate orderings using the Kemeny-Young voting method [110], which is useful for resolving potential cyclic preferences by ranking agent configurations based on the frequency of pairwise ordering comparisons.

3.3.4 Selection of APEX’s Hyper-Parameters

APEX regulates the rates of change for the agent’s parameter settings using its learning rate α and discount factor γ hyper-parameters. We assessed the effects of these settings prior to the study by having two expert users complete coverage sessions with the APEX BB agent multiple times while using a $\{5 \times 5\}$ grid sampling for the hyper-parameter values. The resulting efficiency metric scores were aggregated statistically, and revealed $\alpha = 0.4$ and $\gamma = 0.4$ as most suitable configuration for our fast-paced boundary tracking tasks.

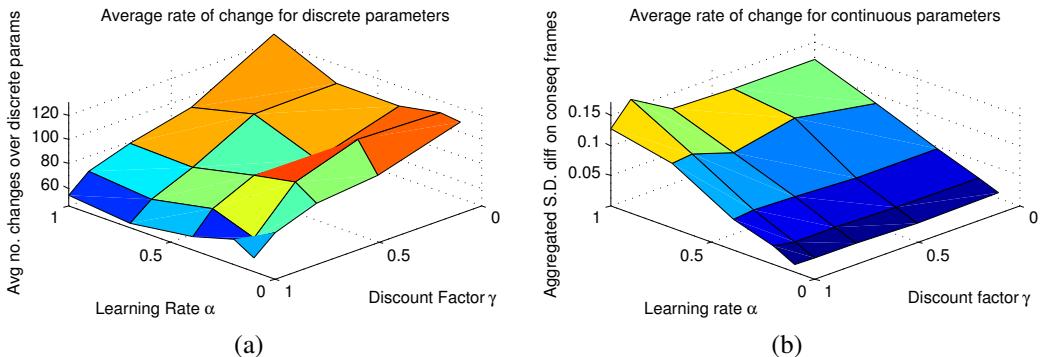


Figure 3–4: Average rates of change for the agent’s parameters are computed by aggregating over 2 expert users’ aerial coverage datasets. Variability among discrete parameters is correlated with APEX’s discount factor γ (a), while changes in the agent’s continuous parameters are affected by APEX’s learning rate α (b).

Figure 3–4 displays the average rates of change for the boundary tracking agent’s parameters as functions of α and γ . We observe that these hyper-parameters independently control the rate of adaptation for the agent’s continuous and discrete settings. In particular, the association between γ and discrete agent settings is due to the instantiation of individual APEX particles for all combinations of distinct parameter values. Since γ regulates the importance of each particle’s historical search performance, smaller discount factor values will result in myopic selections with more frequent parameter changes.

3.3.5 Results and Discussion

Figure 3–5 depicts mean agent rankings for the session-wide scale as well as separately for each terrain boundary segment. These visualizations are accompanied with statistical analysis resulting from Friedman and post hoc Nemenyi tests, both suggesting the presence of significant different agent preferences among users. We corroborated these efficiency orderings for the different human-robot team configurations by computing aggregate rankings using the Kemeny-Young method, which found identical orderings.

Session-wide rankings revealed the APEX WB adaptive agent to be the most efficient overall, followed by CONST and APEX BB. All three agents were ranked higher than the MANUAL teleoperated configuration, although no statistically significant differences were found between the adaptive and non-adaptive agents.

Evaluations at the granularity of individual terrain boundaries showed that the non-adaptive expert-tuned boundary tracker, CONST, outranked both adaptive agents during the *Highway* and *Forest Path* segments. This preference can be attributed to the high quality of the expert-tuned settings of the CONST agent, which resulted in excellent tracking performance for relatively stable boundaries. Nevertheless, post hoc analyses in Figures 3–5b and 3–5c did not reveal any significant

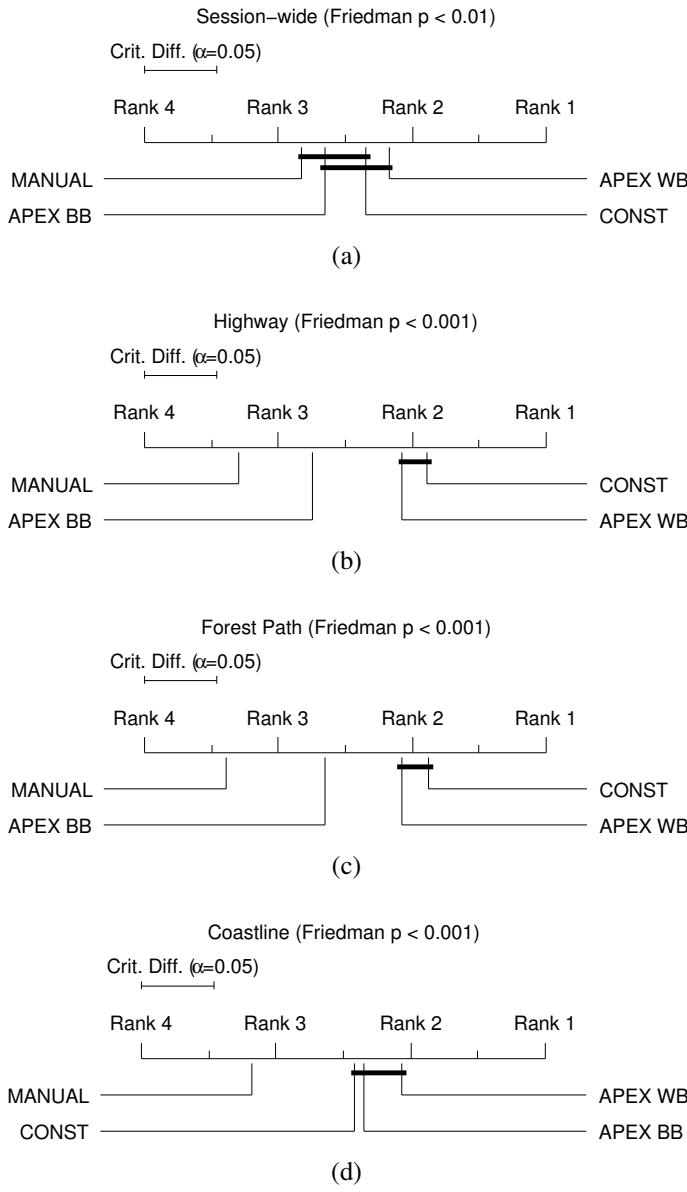


Figure 3–5: Critical difference plots from the APEX aerial coverage study revealed statistically significant differences among the agents’ per-metric rankings. At the session-wide scale (a) and for the coastline segment (d), users preferred both the adaptive and expert-tuned agents, although post hoc Nemenyi test found the *lack of any critical differences among agents* (i.e. the presences of thick horizontal lines straddling APEX WB, APEX BB, and CONST). In contrast, both APEX WB and CONST configurations were ranked significantly greater for the highway (b) and forest path (c) segments. All agent rankings matched the aggregate orderings resulting from the Kemeny-Young voting scheme.

differences between CONST and APEX WB during these terrain segments, thus suggesting that the adaptive agent also attained comparable levels of team efficiency.

During the *Coastline* segment, the non-adaptive CONST agent was outranked by both adaptive agents, APEX WB and APEX BB. By analyzing the raw interaction experiences, we observed that CONST was not able to track the constantly changing shape of the coastline boundary despite repeated intervening assistance

from participants. This finding demonstrated the utility of the Adaptation from Participation paradigm, as both APEX agents were able to adjust their steering styles dynamically after seeing the supervisor aggressively track the boundary target.

All of the aggregate rankings showed consistent preferences of the white-box variant APEX WB over the black-box agent APEX BB. This result can be naturally explained by the fact that APEX BB used gradient-based optimization to numerically approximate the analytical least squares solutions found by APEX WB.

Finally, although a few participants were able to track the target terrains accurately using MANUAL teleoperation, overall this baseline configuration was statistically the least preferred setting among this study's population.

In summary, this study showed that APEX-enabled agents contributed to high degrees of team efficiency, and in particular attained levels to collaboration comparable to those from an expert-tuned non-adaptive robot controller. Whereas the expert-tuned agent required time-consuming manual parameter tweaking by a knowledgeable robot designer, the Adaptation from Participation paradigm helped interactive agents attain similar degrees of task competency based only on *occasional* intervening assistance from *non-expert operators*. Furthermore, the AfP paradigm was shown to be especially advantageous in situations where task conditions varied wildly, which naturally warranted the need for dynamic behavior adaptation.

3.4 Campus Patrol Field Evaluation with Husky Wheeled Robot

Our second set of evaluations sought to re-substantiate the efficiency gains of APEX-enabled agents, by moving from a controlled environment in the previous study to a real-world deployment setting. As seen in Figure 3–6, participants in this field study were asked to assist diverse types of agents and steer the Husky wheeled robot to complete terrain patrol tasks on McGill University's downtown campus. In addition to coping with real-world constraints such as finite communication range and dynamic obstacles like pedestrians, these participants also had to continuously



Figure 3–6: A participant (left) collaborating with the interactive boundary tracking agent to steer the Husky robot during the campus patrol field trial while being monitored by a study conductor (right). Inset: internal state of the interactive agent (not shown to the participant) depicts the detected terrain boundary (blue line) and turn rate steering signals from the agent and the operator (blue and green arrows).

monitor the physical well-being of the robot as well as its surroundings while completing their assigned patrol duties.

These in-field deployments compared the overall efficiency rankings between the APEX BB adaptive agent, the expert-tuned static boundary tracker CONST, and the fully teleoperated MANUAL baseline. The white-box agent APEX WB was omitted since the frontal-view boundary tracking agent did not offer a linear inverse mapping \mathbb{A}^{-1} (see Appendix A for details).

In each trial session, the participant interacted with one of the agent configurations above to control the Husky robot. The session goal entailed steering the vehicle *at a specified distance* alongside a sequence of terrain boundaries while adhering to transition points corresponding to salient visual landmarks such as lamp posts. Participants were instructed to complete this fixed-length patrol course accurately yet also as quickly as possible.

3.4.1 Participants

We recruited 7 individuals to collaborate with adaptive and baseline robot agents during this campus patrol field trial. All participants had previously completed the user study and thus were accustomed to the supervisor-worker interaction scheme. Nevertheless, none of the operators had any prior experience controlling the Husky or similar types of wheeled robots.

3.4.2 Infrastructure

We deployed the frontal-view variant of the boundary tracking agent (see Appendix A for details) to steer the Husky robot based on visual input from its tilted front-facing camera. Contrary to the planar-view variant, the frontal-view control mapping module accounts for the camera’s non-planar pose by projecting image-plane boundary information onto the vehicle’s ground plane. This is realized through a feature-based control law: we compute the intersection χ between the image-plane boundary line and the bottom of the camera frame, as well as the slope ϕ of the line, and prescribe the following parametric function for producing normalized turn rate commands $y_r \in [-1, 1]$:

$$y_r = M_1\chi + M_2\phi + M_3$$

The scaling factors M_1, M_2, M_3 are used to linearly approximate the camera-to-ground-plane projection transform. These parameters also encode the nominal lateral distance to following alongside a given boundary target. In addition to this more complex control mapping, the planar-view agent further pre-filters out horizon content in each scene by cropping out the top H_0 percentage of every camera frame prior to detecting the ground-plane terrain boundary.

For this terrestrial robot controller, we used APEX to adapt the horizon cut-off parameter H_0 , discrete appearance T_a and boundary type T_b specifications for the visual detection stage, as well as the various control mapping parameters M_1 ,

M_2 , and M_3 . We specified loosely-estimated ranges for all continuous parameters, which were obtained during preliminary field testing. During these testing runs, we also manually refined the learning rate α and discount factor γ of the APEX algorithm, starting from the user study’s settings. The final hyper-parameter values were set to reflect the slower pace of the Husky robot: $\alpha = 0.2$, $\gamma = 0.7$.

The gamepad control interface in this field trial was directly transplanted from the aerial coverage study. As previously, the user assumed intervening control by holding down a shoulder button and moving the left analog stick horizontally to change Husky robot’s steering direction, which always traveled at a fixed speed.

Users were asked to supervise operations by walking alongside the vehicle and were not provided with a visualization of the boundary tracking process. Although this third-person perspective offered greater situational awareness of the robot’s surroundings, the agent’s state was also less transparent. Consequently, we anecdotally observed that operators were more sensitive to momentary agent misbehaviors and intervened more readily overall. Furthermore, without visual feedback of the agent’s steering commands, users were forced to adopt a trial-and-error approach for disengaging control to assess whether the boundary tracking agent has adapted sufficiently.

Most of the automated infrastructure from the aerial coverage user study were adapted for this field trial, including data logging and randomized agent ordering generation. Nevertheless, a human “study conductor” walked alongside each participant during the trial sessions and used a separate gamepad to move the Husky vehicle to its starting location as well as to trigger the start and end of each interaction session. As a last-resort safety measure, the conductor’s gamepad could also be used to teleoperate the Husky while overriding all other control signals.



(a) Footpath (follow at 1.5 ft) (b) Grass-side (follow at 1.0 ft) (c) Curb (follow at 0.5 ft)

Figure 3–7: The test course for the APEX campus patrol field trial consisted of moving alongside a sequence of terrain boundaries with visually-defined transition points. Boundary targets had varied visual appearances and included a footpath segment (a), a grass-side sidewalk section (b), and a road-side curb (c).

As seen in Figure 3–7, the test course for this field trial involved patrolling alongside three visually distinct terrains, comprising a *Footpath*, a *Grass-side* sidewalk, and the *Curb* of a long and curved stretch of road. Participants were instructed to maintain specific distances laterally between the Husky robot and each boundary target in order to facilitate our quantitative evaluations. Such a task requirement is essential to several application domains, including street cleaning and agricultural robotics.

3.4.3 Procedure

The flowchart for this field trial is shown in Figure 3–8. At the beginning of each trial run, the participant is briefed on the patrol tasks and the test course. Next, a free-roam practice session re-familiarizes the user with the gamepad-based control scheme. During each of the following test sessions, both the participant and the study conductor walk alongside the Husky robot to complete the designated patrol duties as quickly as possible. The ordering for the robot agent configurations was determined at random at the beginning of each trial run.

These field trial sessions again assessed a multitude of efficiency metrics. Since the Husky robot lacked accurate outdoor localization capabilities, the *mean distance to ground truth* measure was modified to represent average angular distances between the agent’s heading commands and ground truth values. These

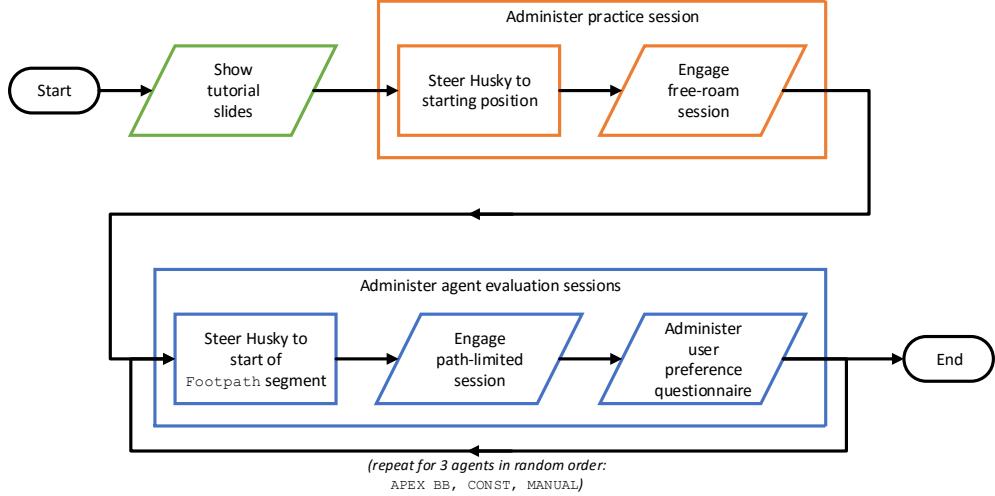


Figure 3–8: Flowchart for the APEX campus patrol field trial.

ground truth headings were generated post hoc by the boundary tracking agent while using terrain-specific optimized settings. Separately, overall task completion was quantified as the *total durations* traveled in each session and across each patrol segment. Further evaluation metrics were identical to those in the user study, including the *agent failure ratio*, the *supervisor intervention ratio*, and the *user preference* ratings. Aggregate agent orderings were again determined based on mean rankings and the associated statistical tests, which were further corroborated using the Kemeny-Young voting scheme.

3.4.4 Results and Discussion

Both the mean agent orderings in Figure 3–9 and the Kemeny-Young aggregate rankings in Table 3–1 revealed that the adaptive APEX BB agent consistently outranked both the expert-tuned CONST boundary tracker and the MANUAL teleoperated configuration. Nevertheless, Friedman and post hoc Nemenyi analyses only found minor statistical differences among agent preferences, and only pertaining to the *Footpath* and *Grass-side* task segments.

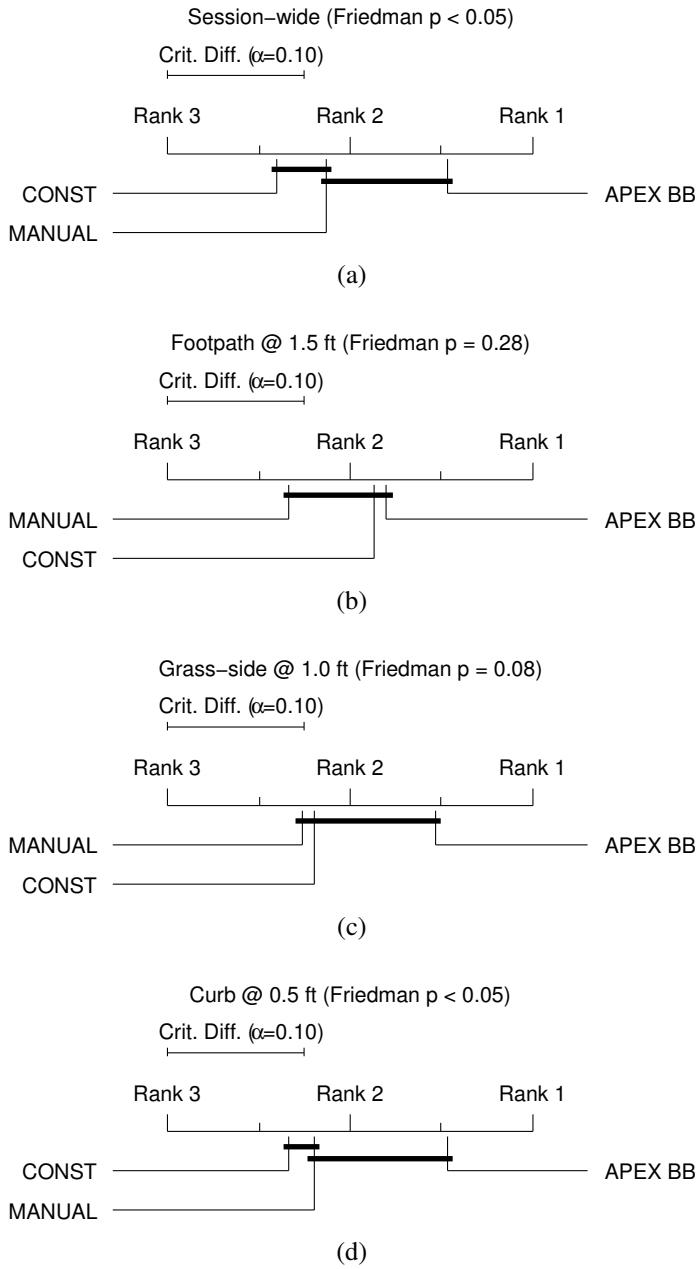


Figure 3–9: Critical difference plots from the campus patrol field trial indicated that the APEX BB adaptive agent consistently contributed to the greatest overall team efficiency in comparison to both the non-adaptive CONST agent and the baseline MANUAL teleoperated configuration. Post hoc Nemenyi analysis revealed the *lack of significant differences* (i.e. the presence of thick horizontal lines straddling multiple agents) among these orderings.

The individual session-wide metric scores in Figure 3–10 revealed that APEX BB attained the highest placement across nearly all metrics and participants. Notably, the statically-configured agent CONST achieved worse task completion and accuracy than APEX BB in all the trial runs. This result can be attributed to the difficulty in manually conceptualizing the effects of the different mapping parameters

Table 3–1: Kemeny-Young aggregate efficiency rankings from the campus patrol field trial revealed identical efficiency orderings, favoring the APEX BB adaptive agent, over the expert-tuned CONST non-adaptive agent, and trailing with the MANUAL baseline configuration.

Segment	Worst	Second	Best
Session-wide	MANUAL	CONST	APEX BB
Footpath (@ 1.5 ft)	MANUAL	CONST	APEX BB
Grass-side (@ 1.0 ft)	MANUAL	CONST	APEX BB
Curb (@ 0.5 ft)	MANUAL	CONST	APEX BB

M_1, M_2, M_3 , as well as to the inherent need to adapt tracking behaviors to cater to the varied distance requirements for each patrol segment.

These individual metric scores also reflected alternating preferences between the non-adaptive CONST agent and the baseline MANUAL teleoperated setup. A similar discrepancy between individual participants can be seen from their varied subjective assessments, where some users were content to collaborate and help out an autonomous agent, while others focused on attaining superior task completion and accuracy using manual control alone. The conflicting merits of the CONST and MANUAL configurations can further be seen in the minor discrepancies between mean agent orderings and the Kemeny-Young aggregate rankings, although, as shown in Figure 3–9, these two agents attained very similar mean ranks.

In summary, this field evaluation demonstrated the various efficiency gains of APEX-enabled interactive robot agents. Notably, our Adaptation from Participation paradigm enabled human-robot teams to achieve superior task performance in comparison to a manually-tuned interactive robot agent. This achievement can be attributed to both the presence of dynamic task requirements as well as the inherent difficulty in conceptualizing inter-dependent parameter effects in complex robot agents. Additionally, APEX-enabled human-robot teams attained competing levels of task performance to plain teleoperated runs, yet required noticeably less amount of active human workload. Although interpretations of the field trial findings were

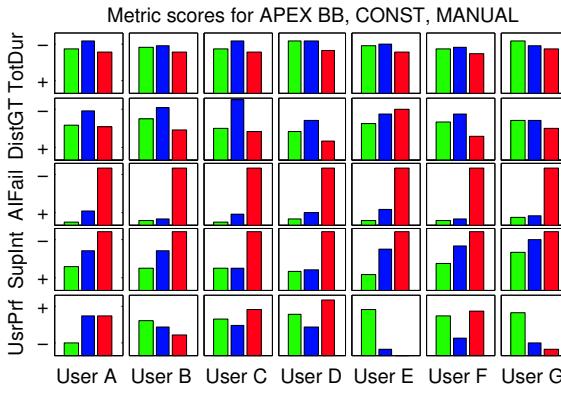


Figure 3–10: Session-wide scores from the APEX campus patrol field trial assessed separate aspects of efficiency, including total elapsed duration (TotDur), mean distance to ground truth (DistGT), agent failure ratio (AIFail), supervisor intervention ratio (SupInt), and user preference (UsrPrf). Smaller values are preferred on all metrics except for UsrPrf . Each subplot compared results between the adaptive agent APEX BB (green bar), the statically-optimized agent CONST (blue bar), and the MANUAL teleoperated configuration (red bar).

complicated by limitations of field deployments, such as variabilities in environmental conditions and restrictive scheduling, all of these results were consistent with the statistically significant findings from the user study.

3.5 Interactive Driving Field Demonstration with SL-Commander Vehicle

We conducted a third set of field assessments to investigate the feasibility of the AfP paradigm for the emerging domain of autonomous driving. Such advanced computer-assisted capabilities promise innumerable benefits to society and are being adopted by mainstream vehicle manufacturers as well as government bodies [12, 93]. A common goal shared by many is for autonomous driving solutions to operate robustly within diverse outdoor environments, which poses many challenges. For instance, outdoor paths can vary highly in their definition (e.g. gravel, grass, metal, dirt trails). Also, navigation solutions based on the Global Positioning System (GPS) are prone to failures when operating in previously unseen or changing paths. Furthermore, reactive way-finding using local sensors, such as cameras, is susceptible to restrictions such as having constrained field of views, which limit their autonomous navigation performance.



Figure 3–11: The participant, sitting in the SL-Commander’s passenger seat, collaborated with our adaptive boundary tracking agent during the APEX interactive driving field demonstrations. A human driver was also present and provided redundant safety duties by maintaining access to the vehicle’s brake pedal and emergency stop buttons.

In contrast, an appealing solution is to ask a competent human driver to occasionally help the autonomous agent in navigating through dynamic and challenging situations. Toward this end, we deployed the adaptive boundary tracking agent onboard the SL-Commander vehicle, as seen in Figure 3–11, and invited 4 roboticists to collaborate with this interactive adaptive automobile and steer along several test tracks. These preliminary deployments aimed to explore the benefits and challenges of adaptive human-agent interactions within an *in situ* context, with particular focuses on adapting to each operator’s driving preferences and coping with dynamic environmental changes.

3.5.1 Infrastructure

We carried over the interactive boundary tracking agent from the Husky field study and connected it to the SL-Commander’s drive-by-wire interface. This seamless integration demonstrated the flexibility of our general-purpose boundary tracking controller at accommodating different types of camera and actuator configurations, thanks to the online behavior adaptation capabilities afforded by APEX.

As seen in Figure 3–11, during these experiments a human sitting in the driver seat assumed the role of safety manager, and assumed permanent control over the

vehicle's brakes and kill switch. The passenger-side user interacted with the boundary tracking agent using a gamepad and received visual feedback on a laptop screen. During autonomous mode, the APEX-enabled agent exerted control over the vehicle's steering rate while traveling at a fixed speed. Similar to the Husky's control mapping, the user could engage intervening mode at any time by holding down a shoulder button, and then steer by moving the left analog stick horizontally. Also, the passenger could change the vehicle's speed by pressing shoulder buttons to increase or decrease the desired velocity by a fixed increment. Furthermore, the supervisor could alter the camera's pan-tilt position dynamically by moving the right analog stick, as well as reset to its default pose by clicking into the analog stick.

The ability to dynamically change the camera's pose was useful for quickly re-positioning the limited viewport when the terrain boundary changed shape. Unfortunately, these pose changes significantly affected the boundary tracking agent's control law, since its feature gains reflected an approximation of the camera's perspective transformation and thus needed to be updated. Tilting the camera vertically also altered the amount and location of the horizon in view, which negatively impacted the terrain segmentation process. Thankfully, the AfP paradigm enabled the human-robot team to address these concerns by allowing the agent to learn from brief demonstrations of proper steering after each pose change.

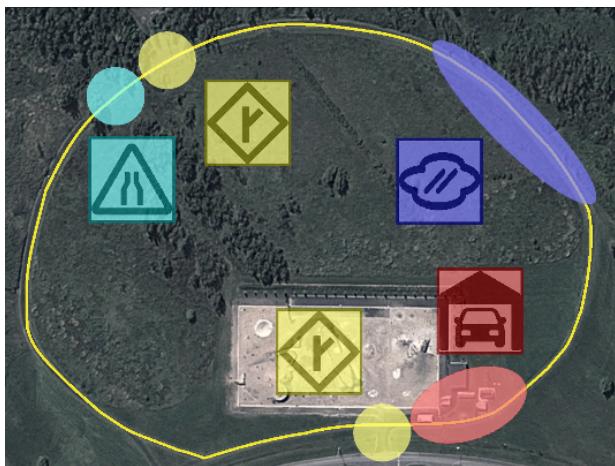


Figure 3–12: The primary test course for the APEX interactive driving field demonstrations featured a 1 km gravel course (yellow) at the Canadian Space Agency site, which was surrounded by numerous hurdles including narrow passages (cyan), road intersections (yellow), watery ditch (blue), and nearby parked cars and structures (red).

The primary task scenario in these field demonstrations consisted of circumnavigating a 1 km closed-loop gravel pathway at the Canadian Space Agency, as seen in Figure 3–12. This course featured several challenging scenario segments, including a forested patch, narrow land bridges passing over pipes, and water-filled ditches. In addition to this primary test course, several participants also directed the vehicle to navigate less well-defined paths, such as along the tire tracks of another vehicle that had previously passed through a muddy field. These experiments were performed within a controlled area located at the Canadian Space Agency, as part of the 2014 NSERC Canadian Field Robotics Network (NCFRN) field trials.

3.5.2 Results and Discussion

All 4 participants assisted our boundary tracking agent in safely navigating through the 1 km primary test course without requiring any interference from the safety driver. Analyzing the recorded experiences revealed a common pattern across all sessions: starting from cautious speeds as low as 2 km/h during initial task training, participants quickly increased the target velocity up to 20 km/h as the agent soon latched onto the gravel road and drove smoothly without incident. Aggregating across all test sessions, the overall average speed of the vehicle was 8.98 km/h, while a breakdown between periods of user intervention and autonomous control revealed mean values of 8.76 km/h versus 9.35 km/h. Both the large initial speed increase and the faster average velocity during autonomous control are indications that users developed confidence in our APEX-enabled agent.

Rather than having to steer manually through the entire test course, the average frequency of interventions during these test sessions was 40%, thus demonstrating a notable decrease in active human workload. Nevertheless, there was an increase in intervention frequency compared to recorded experiences from both the aerial coverage user study and campus patrol field trial. We attribute this discrepancy to the *in situ* context, and specifically to the participants' reported precautionary

concerns for their physical well-being. These concerns caused users to develop heightened sensitivity to even slightly perturbing motions made by the robot agent.

Further investigations into intervention patterns revealed an interesting side-effect: despite consistent tracking performance from our agent, some participants *proactively* assumed control in anticipation to upcoming terrain hazards, such as when moving into the segment bordering a watery trench. This phenomenon is beneficial in several ways, since it shows that AfP allows users to preserve their sense of control at all times, and also since such proactive interventions provide extra training exemplars for the robot agent to further refine its behaviors.

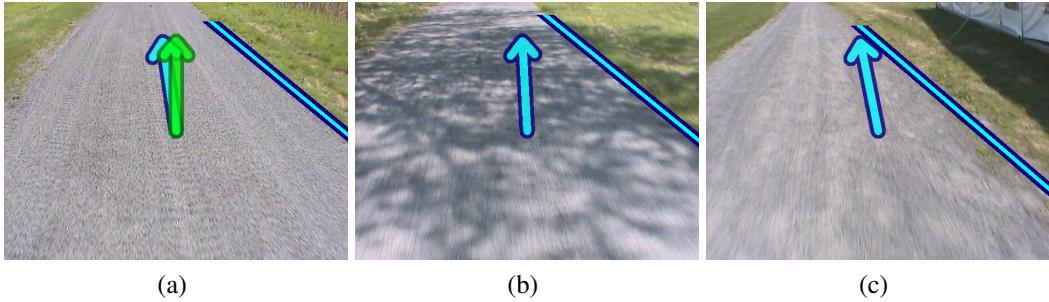


Figure 3–13: Snapshots from a session through the primary test course during the APEX interactive driving field demonstrations: (a) the human operator began by training the boundary tracking agent at 2 km/h to follow a previously-unseen gravel road (with the manual steering command shown as a green arrow); (b) the passenger subsequently relinquished control and ramped up speed to 20 km/h after witnessing robust autonomous tracking performance through diverse task conditions (with autonomous steering command shown as the blue arrow); (c) the run concluded near a large tent whose shadow represented a distracting secondary boundary on screen, although the agent remained unfaltering in tracking the gravel path.

Figure 3–13 illustrates a typical run around the main test course, and demonstrates our adaptive agent’s robust tracking performance throughout variations in lighting and terrain. Several users also took the liberty to deviate away from the gravel road during repeat sessions. Our APEX-enabled agent catered to these task intent changes by adapting to track along distinct visual boundaries around the test site, including a set of muddy wheel tracks as seen in Figure 3–14.

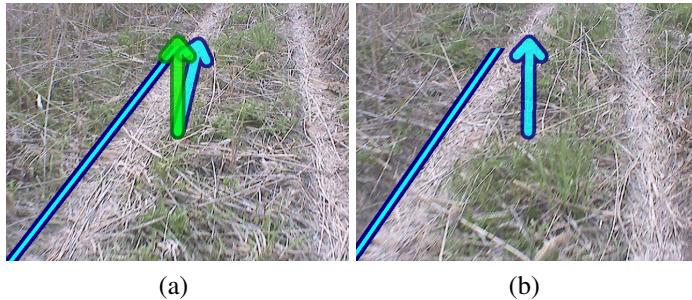


Figure 3–14: During a session of the APEX interactive driving field demonstrations, the user steered away from the gravel road and onto fresh wheel tracks on a muddy field: (a) following a brief period of manual steering, (b) our robot agent adapted to follow the left wheel track on its own.

A key characteristic of our robot agent is the ability to quickly adapt to changing task scenarios. Evidence of such adaptations can be seen in its resilience to track changes in the gravel road despite the presence of ample tree shadows, as illustrated by Figure 3–13b. Also, Figure 3–15 shows that our autonomous controller was also able to adapt its perception and control processes when the user made impromptu changes to the vehicle’s camera positioning. Importantly, these adaptations required only brief periods of manual steering to re-train the agent’s parameters.

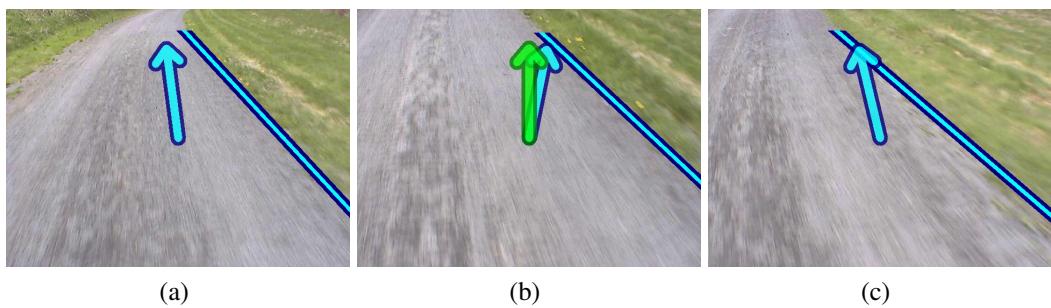


Figure 3–15: These snapshots depict interactive adaptation capabilities following a user-induced change to the camera’s positioning during APEX the interactive driving field demonstrations. (a) After initial training, the agent autonomously tracked the desired gravel pathway; (b) the agent’s perception and control gains were mismatched as the user panned the onboard camera to the right and downwards, inciting intervention; (c) our agent swiftly adapted its behaviors based on this brief intervention period, causing the operator to promptly relinquish control.

In summary, preliminary evaluations with our adaptive visual navigation agent onboard the SL-Commander vehicle showed great success in its ability to adapt to novel task objectives, naturally varying environmental conditions, and impromptu changes in the user’s task intent. These interaction experiences further corroborated our previous empirical findings by demonstrating the potentials of AfP at reducing workload, building user satisfaction, and most importantly, improving the robot agent’s performance over time through occasional intervening assistance.

3.6 Summary

This chapter introduced *Adaptation from Participation* (AfP) as a human-robot interaction paradigm that extends shared control via behavior adaptation capabilities. We also formulated AfP as a computational problem similar to Learning from Demonstration yet with a distinct focus on changing task goals and dynamic conditions. At its core, AfP offers robot operators freedom from the choice of manually specifying and adjusting system parameters, while having the agent still cater to their personal preferences through behavior adaptation.

Additionally, we presented the *Adaptive Parameter EXploration* (APEX) algorithm as an anytime concurrent realization of the AfP paradigm. We developed three end-to-end instantiations of APEX-enabled agents on distinct aerial and terrestrial robot platforms, and assessed the resulting efficiency of such adaptive human-robot collaboration within both controlled and real-world application scenarios. These multi-domain investigations revealed that APEX-enabled agents attained superior task accuracy and reduced active human workload in teleoperated contexts. Our results also showed that AfP allowed enthusiast users to help robot agents to achieve high levels of task performance rivaling manually-optimized systems, without requiring tedious empirical analyses or expert knowledge of the agent’s internal workings.

One of our motivations for developing the AfP paradigm was to establish robust and adaptive autonomous agents as the baseline for our investigations into human-robot trust. Additionally, these adaptive agents indirectly sought to gain their human collaborator’s trust through behavior imitation. Nevertheless, despite their demonstrated efficiency gains, these APEX-enabled agents lacked the capacity to address the possibility of excessive interventions from the human supervisor due to repeated distrust. These agents further lack the ability to distinguish interventions that are induced by distrust from those arising due to other causes, such as a change in the user’s task intent, or a natural tendency to favor manual control (especially as seen in *in-situ* interaction contexts). These limitations substantiate the need to quantify the human collaborator’s trust state during interactions explicitly, so that the robot agent can address losses of trust via a direct approach. The remainder of this thesis will address each of these topics in turn.

Chapter 4

Understanding Human-Robot Trust

Chapter 3 presented the Adaptation from Participation paradigm as a means for the robot agent to *implicitly* seek to improve the human supervisor’s trust. For the remainder of this thesis, we consider how the supervisor’s moment-to-moment trust state can be capitalized by the agent *directly* to maintain an efficient and trusting collaboration. This approach can be broken down into three constituents:

1. characterizing factors that affect the human’s moment-to-moment trust state;
2. modeling the temporal dynamics of the supervisor’s trust state;
3. building reactive agent behaviors induced by this trust signal to mitigate trust loss and potential teamwork breakdown.

This chapter focuses on the first component, and reports on two interaction studies investigating the dynamics and evolution of trust within supervisor-worker human-robot teams. The first experiment on “event-induced trust changes” analyzed changes in the supervisor’s trust state in response to different *manipulated* events. The second observational study on “real-time trust dynamics” expanded the scope to cover experiences over prolonged and typical interaction sessions, and *without manipulating the agent’s behaviors*.

Descriptive analyses on these datasets revealed correlations with each human supervisor’s evolving trust state to a key set of factors arising from the interaction experience. These factors are valuable toward building computational models that predict changes to the individual’s trust state. In fact, the first study included such

a preliminary modeling effort, although modeling real-time trust dynamics will be the primary topic discussed in Chapter 5. In addition to relating the trust state to diverse factors, both studies also produced practical insights that will help establish a natural structure to our temporal trust modeling efforts. Finally, the interaction datasets themselves represent a rich training corpus of experiences from many human supervisors with diverse backgrounds, which we will use to build data-driven and personalized trust dynamics models.

4.1 Methodology

One of the main goals of the two studies presented in this chapter is to quantify the relative importance of different interaction factors that can affect the human’s trust state. We addressed this objective by first enumerating important factors identified in relevant literature, and then evaluating their effects on the supervisor’s trust dynamics. This section describes our event-centric view for assessing trust factors. We also elaborate on the final set of chosen trust factors, and further explain the methods and instruments used to elicit trust responses from study participants.

4.1.1 Event-Centric View

Similar to the formulation for Adaptation from Participation (AfP), we partition a period of human-robot interaction based on a sequence of discrete events, each of which corresponding to a notable change in the state of the robot and/or the environment. Sample events for visual navigation tasks include a sustained period of boundary misdetection failures by our agent, or strong force (such as gust) that pushes the robot sideways and causes it to lose track of the target boundary. By measuring the change in trust state in response to different event types, we can quantify fine-grained trust dynamics at small time scales. This event-centric perspective differentiates our investigations from the majority of existing studies, which have characterized impacts on the *post-experience* trust state aggregated from longer-term interactions (e.g. [24, 34, 107]). This view also complements our AfP paradigm:

whereas optimal behaviors for the robot agent are affected by dynamic events that partition the interaction experience into disjoint periods, the human’s trust state is impacted in an cumulative manner during each of these periods.

4.1.2 Trust Factors

Prior to carrying out our experiments, we considered the large corpus of factors identified by the literature to influence trust in human-robot and human-automation teams (e.g. [4, 23, 41, 79]). We refined this list by eliminating factors that were incorporated into the design of our agent and interface (including situational awareness and interactive adaptability), as well as factors whose values were expected to be stationary across all study sessions and participants (such as the type of robot vehicle and the operator-robot proximity). Given our research end-goal of inferring the human’s trust dynamics *during interactions*, we also excluded entries that could not be obtained by the robot agent, either via sensory observations or by querying the human (e.g. task complexity, shared mental models, etc. [41]). Furthermore, since our agents are assumed to be always well-intentioned and never adversarial, we discarded all *intention-centric* trust factors.

The remaining entries include several groups of factors that can be extracted directly from the interaction experience:

- the task accuracy and completion;
- the rate at which the robot agent fails to produce sensible commands (i.e. “algorithmic failures”), possibly due to noisy or challenging task conditions;
- the frequency of interventions from the supervisor;

Several other trust factors can be obtained by querying the human supervisor at different times during the study:

- a pre-experiment survey: user demographics, general attitudes, and prior experience with robots and remote control tasks (following [23]);

- post-session questionnaires: assessments of the robot’s and user’s task performances, as well as the robot’s perceived robustness and adaptability;
- a debriefing questionnaire: experiment-wide task load assessments (via Raw TLX [43]), and post hoc updates on trust propensity toward robots.

Our study design highlights the important notion that the supervisor’s trust state is dependent on factors at *multiple time scales*. In particular, we expect perceptions obtained during the survey and debriefing questionnaires to be constant throughout the study, in contrast to per-session user assessments. Also, experience-based factors can be measured both on an event-centered granular time scale, as well as aggregated over a cumulative period of interaction.

4.1.3 User Assessment

Figure 4–1 depicts the general visual layout for questionnaires administered during the studies. Most of the questions used a continuous interval answer format to elicit either unipolar (e.g. Likert scale [59]) or bipolar responses, while few exceptions asked for discrete-choice responses. We gathered responses using the Visual Analogue Scale (VAS), i.e. a continuous and bounded scale without tick marks, which boasts superior metric properties over N-point discrete scales [74]. We specifically chose this continuous scale so that we could apply regression-based techniques to model these user responses.

In addition to gathering assessment-based trust factors, users were also asked to introspect and quantify their trust states both before and after each interaction event. Each trust assessment query employed a single-question format, i.e. “*What is your degree of trust in the robot agent’s performance right now?*”

Our single-question trust query contrasts with multi-item trust scales proposed in the literature [47, 64]. In general, such single-item queries potentially suffer from reduced expressiveness and diminished psychometric reliability. Nevertheless, existing multi-item trust scales degenerate to closely resemble our answer format,

Session 1/4 Post-Session Questionnaire

TASK: Follow road and turn left at first intersection

Instructions

A) Task

1. **Out of view:** did the target terrain boundary(ies) leave the camera's viewport?

yes no

2. **Robot's Performance:** please rate the robot's task performance during autonomous control.

poor excellent

3. **Your Performance:** please rate your own task performance during manual control.

poor excellent

I did not intervene

B) Robot Performance

C) Robot Competence

D) Trust

Submit

Figure 4–1: The post-session questionnaire for the event-induced trust changes study comprised of assessment queries using a continuous Visual Analogue Scale (VAS) format, as well as few discrete-choice questions.

since many of their questions were not applicable to our interaction context (e.g. concerns about the robot’s integrity or its physically harmful actions), while we administered other queries as separate factors (e.g. perceptions of the agent’s reliability and predictability). Also, we chose this single-question trust format to facilitate collecting repeated trust assessments at event-time scales and mitigate the likelihood of discouraging and annoying participants by inundating them with lengthy questionnaires after every minute or so during the study. This practicality consideration has been emphasized by similar studies as well [23, 79].

4.2 Experimental Study on Event-Induced Trust Changes

The first study asked participants to supervise and assist our boundary tracking agent (see Appendix A for details) in controlling a simulated aerial vehicle along rural roadways. These task scenarios were carried out via the SightedTurtleSim drone simulation framework using static satellite footage of various farmlands. At

a certain point during each session, the robot agent’s behavior would be altered programmatically to elicit reactions and changes in the user’s trust state. In addition to these recorded interaction experiences, each session began and ended with short questionnaires about the robot agent’s perceived performance. Analyses on the resulting datasets quantified key relationships between trust responses, different types of interaction events, as well as a number of other factors. We further describe a preliminary modeling attempt for predicting event-time changes to the human’s trust state based on the recorded dataset.

Interaction sessions in this study all had short durations (< 60 seconds each), and also shared a common task structure: follow a straight road for about 30 seconds until an intersection, make a specific turn, then continue tracking the new path. All participants were told before the study that the boundary tracking agent was capable of following along the sides of roads proficiently, although it lacked the ability to decide between multiple target boundaries at intersections.

This study focused on events corresponding to different types of robot failures, i.e. periods of decreased reliability in the agent’s road tracking performance. We realized these low-reliability states by altering the terrain segmentation parameters of the agent to make it harder to distinguish between the visual appearance of the road versus its surrounding farmland. By toggling between reliability states at different times during each session, this study investigated the effects of the following event scenarios:

- **Baseline:** the boundary tracking agent is left in its high-reliability state throughout the entire session;
- **PoorStart:** the agent starts in the low-reliability state for 10 seconds and is then switched into the high-reliability setting before the road intersection;
- **RobotFault:** the agent is momentarily toggled into the low-reliability state for a 10-second period in the middle of tracking the first road segment;

- Limitation: the agent is switched into the low-reliability state at the road intersection and then switched back into the high-reliability state after 10 seconds.

Prior studies with similar human-robot team configurations have shown that the robot’s reliability affected trust differently depending on its timing, notably with poor initial performance having the most negative impact on the resulting trust state [23]. This study extended investigations by relating low reliability to distinct causes, such as poor initial tuning of the agent’s settings (`PoorStart`), inexplicable algorithmic failure (`RobotFault`), or an expected lack of programming to decide between multiple concurrent task objectives (`Limitation`). We hypothesized that most users would behave in a *rational* manner, and will attribute blame following misbehaviors differently based on the cause of each failure.

Several research groups have investigated similar types of links between trust responses and the cause of automation failures in the human-automation realm [26, 48, 79]. Nevertheless, these prior studies found differences in trust responses to false-positive and false-negative behavior of *classifier-style* automation tools, which are not relevant to our *control-oriented* robot agent.

4.2.1 Participants

We recruited 30 participants (6 females) from the School of Computer Science at McGill University to participate in this human-robot interaction study. The study population had a predominantly young adult age range ($\mu = 27$, $\sigma = 7$ years), and included 11 undergraduate students, 13 graduate students, 2 professors, and 4 university personnel.

These participants reported vastly diverse degrees of prior knowledge, experiences, and attitudes. These diversities were reflected in their questionnaire responses on driving skills ($\mu = 64\%$, $\sigma = 36\%$ degree of agreement with “competent driver”), propensity toward car automation ($\mu = 48\%$, $\sigma = 25\%$ “comfort

using cruise control”), Radio-Controlled (RC) vehicles ($\mu = 45\%$, $\sigma = 33\%$ “competence”), and visual-feedback teleoperation ($\mu = 32\%$, $\sigma = 35\%$ “competence”). Additionally, only 47% of users self-reported to being robotics researchers. Nevertheless, none of the users had any interactions with our boundary tracking system before this study.

4.2.2 Infrastructure

The interaction experiences in this study adhered to a “Wizard **with** Oz” style, i.e. with a *human-centric focus with real technology* [88]. The simulated aerial robot was controlled by our boundary tracking agent alone, as opposed to being covertly steered by a human “Oz”. From the participant’s perspective, these interactions closely resembled real-world settings, thus ensuring that the resulting trust assessments and recorded experiences reflect natural and realistic interactions.

The graphical user interface shown in Figure 4–2 was based on the camera view used during the evaluations of APEX-enabled adaptive agents in Chapter 3. In addition to depicting the steering direction arrows and the detected boundary line, this interface iteration also displayed the detected boundary curve, as a cyan-colored contour. We added these extra visualization elements to help participants better understand the agent’s internal state, by perceiving when its visual segmentation algorithm either inaccurately recognized the target boundary or failed altogether.

At any time during the sessions, the participant could intervene and manually steer the aerial vehicle by holding down a mouse button while displacing the cursor over the camera view in the desired direction. This on-screen control interface differed from the use of a gamepad in our previous studies and was specifically chosen to accommodate this study’s broader audience.

To enforce consistent task conditions, the study scenarios involved tracking rural roadways with very similar appearances, as seen in Figure 4–2. Nevertheless, road segments and turn directions were varied across sessions so as to introduce

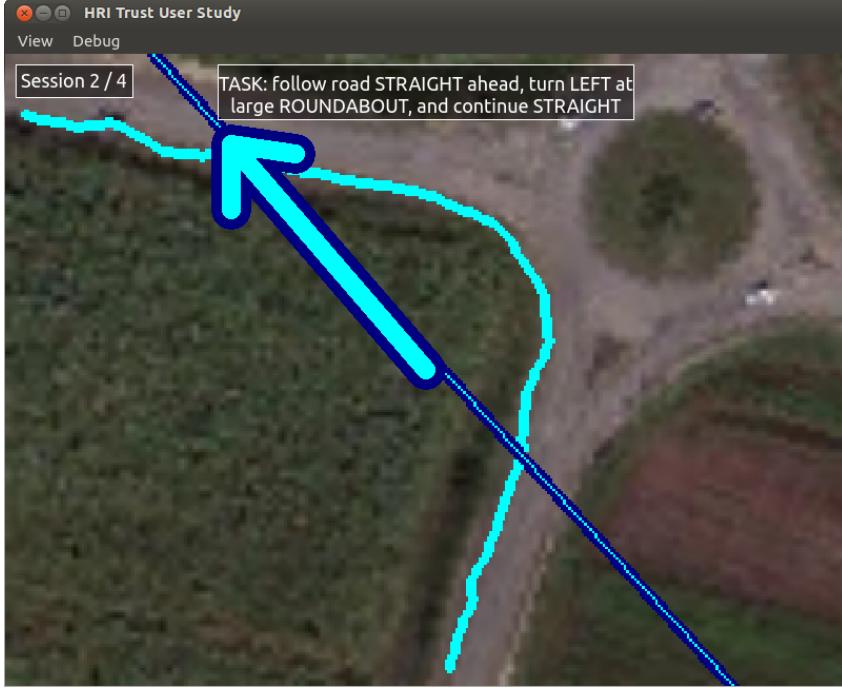


Figure 4–2: Display interface for the event-induced trust changes study depicted camera frames from a simulated aerial robot, overlaid with its boundary tracking agent’s detected boundary contour (cyan curve), line fit (blue line), and heading command (blue arrow). Also, the user’s command was shown as a green arrow during interventions. Further overlays denoted the current session ID and task goal.

slight diversity in experiences and to mitigate potential memory effects. Both the vehicle’s forward speed and altitude were kept constant throughout the sessions, and their values were hand-tuned and validated by several test users. These settings were tuned to ensure that the visual tracking tasks were gently paced so as to accommodate the study’s broad audience.

The infrastructure of this study was designed to operate in a fully automated manner, without the need of a human study conductor. This approach enforced consistent interaction experiences across participants and mitigated possible biases introduced by a human experimenter. Additionally, this automated infrastructure handled all event triggers, data logging, and synchronizations between the Sighted-TurtleSim simulator, the boundary tracking agent, and the visual interface between study sessions.

4.2.3 Procedure

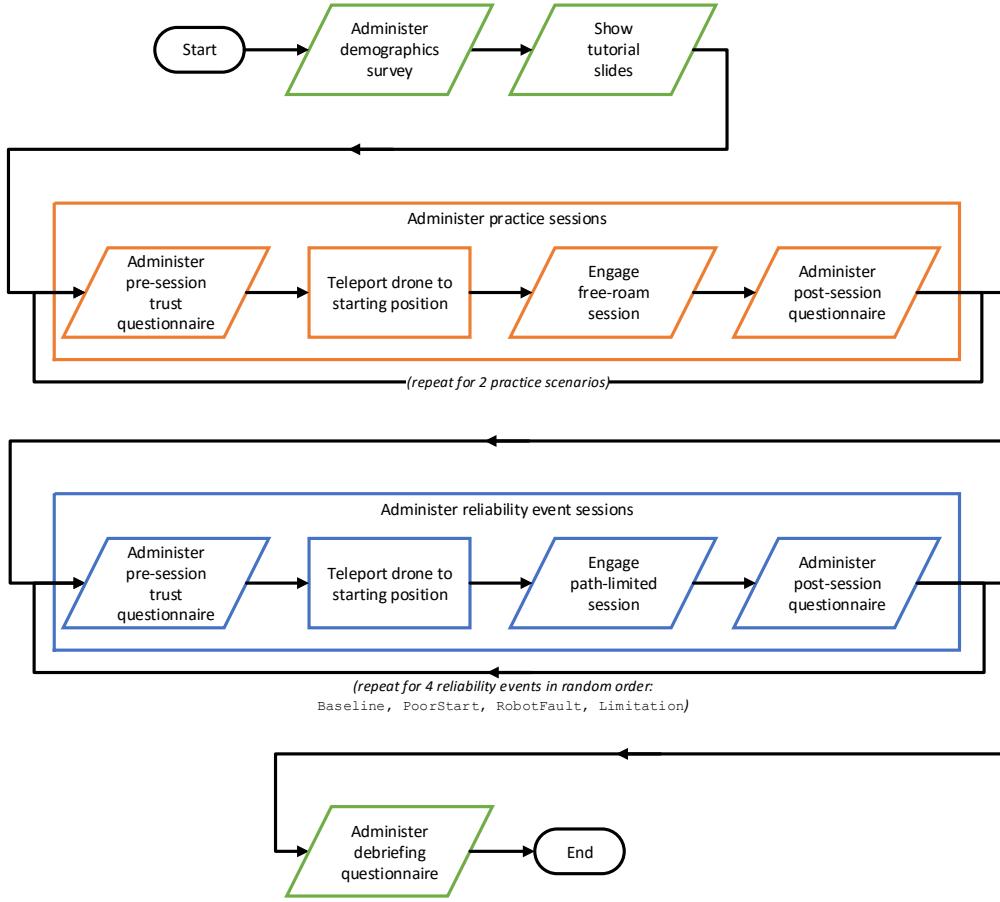


Figure 4–3: Flowchart for the event-induced trust changes study.

As depicted by the flowchart in Figure 4–3, this study was separated into multiple phases that began with a demographics survey questionnaire. A set of tutorial slides then explained the supervision task and interaction context, and also described the robot’s capabilities at tracking visual boundaries along with its lack of ability to change tracking targets. The tutorial further emphasized that the robot agent was programmed purely to complete its tasks in a motivated and non-adversarial manner, and therefore users should base their trust assessments solely on the robot’s performance, as opposed to questioning its intentions.

Next, participants interacted with boundary tracking robots during 6 short sessions, including 2 practice instances and 4 distinct event scenarios in a randomized

and counterbalanced order. Every session began by asking the user to indicate their prior trust assessment toward the yet-unseen robot agent, which was followed by the actual interaction phase, and then ending with a post-session questionnaire. The first practice session helped the user get acquainted with the interface and the robot agent during free roam. The agent was programmatically toggled into its low-reliability state on several occasions during this initial session, both to demonstrate failure events and to prompt the participant to engage in interventions. The second practice session consisted of a variant of the Baseline scenario and acquainted participants with the road-following task objective by providing a demonstration of a typical interaction experience.

Following the 6 sessions, the study concluded with a debriefing questionnaire, which assessed the cumulative interaction experience and collected free-form feedback. These study runs lasted 27 minutes on average.

4.2.4 Results and Discussion

This section presents statistical analyses on several key aspects of the recorded study dataset. We investigated order effects resulting from the crossover session design, as well as the impact of event scenarios on the amount of change in users' trust assessments. We also studied the relative significance of various factors previously identified in the literature on the influence of real-time human-robot trust.

Session Order Effects and Properties of Pre-Session Trust

It is important for the participants to evaluate each session *independently* in order to characterize the causality of different events on their trust responses. Toward this end, the tutorial and every pre-session trust questionnaire repeatedly emphasized that the experiences from each interaction session should be independently assessed and that each session may encompass robot agents with different reliability levels, differing task objectives, as well as distinct environments.

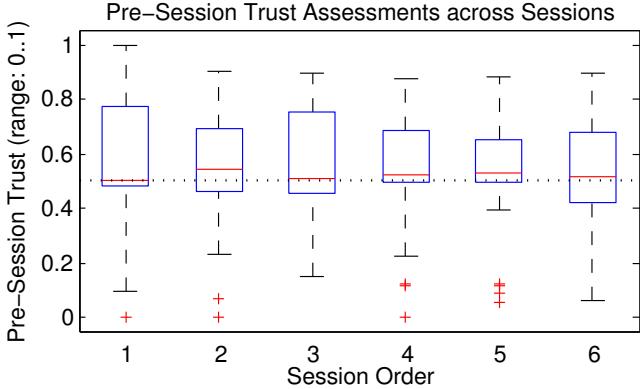


Figure 4-4: Pre-session trust assessments for the event-induced trust changes study were consistent across sessions, and their means showed slight positive trust bias.

Figure 4-4 shows the prior trust assessments reported at the start of each session. A repeated measures analysis of variance (rmANOVA) revealed no significant effects between these prior trust states to the session ordering ($F(5, 145) = 0.30, p = 0.91$), although there was a strongly significant effect from the different users ($F(29, 145) = 24.18, p \ll 0.001$). We thus conclude that although users naturally had different trust propensities, their prior trust states were not significantly biased by the session ordering.

These results also suggest that participants exhibited slight positive trust bias, assuming that a response of 0.5 corresponded to a neutral state. A one-way two-tailed Student's t -test revealed that mean prior trust assessments across users and all sessions (including practice) were significantly different from the null hypothesis ($p < 0.05$). This positivity bias corroborated similar findings in previous human-robot studies [24, 29].

Effects of Event Scenarios

Figure 4-5 shows changes between the pre- and post-session trust assessments in response to the 4 event scenarios in this study. Repeated measures ANOVA revealed significant effects on the mean amount of trust changes due to events ($F(3, 87) = 16.61, p \ll 0.001$) and due to users ($F(29, 87) = 2.35, p < 0.01$). Post hoc pairwise comparisons using the Tukey's range test [44] at the $\alpha = 0.05$ level showed *non-significant differences* in trust changes only between Baseline &

Limitation, PoorStart & RobotFault, and PoorStart & Limitation.

The large number of event pairings with significant trust responses suggest that users attentively discriminated the robot's behaviors during different task sessions.

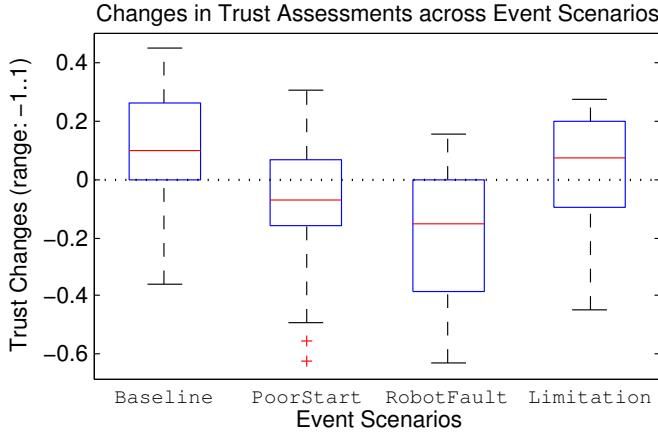


Figure 4-5: Trust variations in response to event scenarios in the event-induced trust changes study were consistent with rational reactions based on causal attribution theory. The slight trust increase in the Limitation scenario suggested that users deliberately did not blame the robot agent for failing to carry out a task change given its known programming limitations.

Looking at average user responses, the gains in trust in reaction to the high-reliability Baseline setting were expected, and similarly so were trust losses due to failures during the PoorStart and RobotFault scenarios. Although the magnitude in trust lost between PoorStart and RobotFault were not significantly different, Figure 4-5 suggests that users reacted more leniently when the agent in PoorStart started with poor performance but then soon showed improvements, as opposed to when the initially-reliable robot agent in RobotFault *inexplicably* failed to track the roadway.

In contrast, the dominant response of *increases in trust* for the Limitation scenario may appear surprising, since the robot agent was programmatically switched into the low-reliability mode at the road intersection. Nevertheless, we believe that users *deliberately* did not penalize such failures because they were aware that the agent lacked the capabilities to carry out changes in the task goal. Therefore,

the robot’s momentary drop in reliability following the intersection was likely interpreted in analogy to a switch from the Baseline to the PoorStart conditions. These similarities in trust responses were consistent with our post hoc pairwise comparative findings.

The keen reader may have noted the apparent dichotomy between these results and similar human-robot interaction studies [23] that found trust to be most detrimentally affected when low-reliability events occurred during initial interactions between foreign operators and robot agents. This prior empirical finding captured the critical essence of “making a good first impression”. Nevertheless, these results focused solely on low-reliability events with *non-apparent* causes, while our experiment should be viewed as a complementary extension that assessed the impacts of different *types* of low-reliability events.

In summary, our findings indicated that different event causes had significant effects on the magnitude of change in the user’s trust state. Following our hypothesis, behaviors from most users were consistent with deliberate and rational considerations with the robot agent’s limitations in mind.

Impacts of Trust Factors

We performed a backward stepwise regression analysis to isolate the most significant relationships among various experience-based and assessment-based factors with event-induced trust changes. This regression process started from a full linear model and was carried out using the Sum Squared Error (SSE) criterion, which iteratively removed the most insignificant factors (when $p > 0.1$) and re-introduced relevant factors (when $p < 0.05$). We disallowed interactions and high-order terms to preclude spurious associations between factors at different time scales. Experience-based factors reflecting the immediate post-event reactions were computed over a 10-second window. This duration setting was chosen to match the length of the pre-determined lapse into the low-reliability mode during the

study scenarios. The placement of event windows were sampled at random for the Baseline scenario, due to the lack of any failure events.

Table 4–1: Among major factors relating to event-induced trust changes resulting from stepwise regression, experiment-wide and post-session personality factors exhibited stronger effects compared to metrics derived from the actual experience.

Categories (Init. FC)	Final FC	DF	Σ MS	Min. p	Avg. p	Max. p
Survey & debriefing (24)	13	15	4.25	$< 1e^{-10}$	< 0.01	0.02
Post-session (4)	2	2	1.27	$< 1e^{-16}$	< 0.01	0.02
Session-wide experience (12)	4	4	0.30	$< 1e^{-2}$	0.02	0.04
Post-event 10 sec window (12)	3	3	0.21	$< 1e^{-3}$	0.27	0.79
Residual		84	0.01			

FC: factor count

DF: degrees of freedom

Σ MS: combined mean sum of squares

Table 4–1 provides a summary of the stepwise regression results. The regression process distilled 52 trust factors at multiple time scales into a final form with 22 major factors. Notably, both the session order ($p = 0.60$) and event scenarios ($p = 0.20$) were removed during the iterative regression process. Remaining factors at the experiment-level scale included demographic entries (e.g. age, occupation), prior expertise and attitudes (e.g. driving and robot control experience, willingness to use a self-driving car), as well as post-experiment assessments (e.g. measures of mental task load). Significant session-level factors entailed post-session assessments of the robot’s performance, as well as session-wide measures of the agent’s internal failure rate and the supervisor’s intervention frequency. At the event-time scale, the externally-quantified task accuracy and the agent’s mistakes were the most influential factors. The final model demonstrated excellent data fit, with a Root Mean Squared Error of $RMSE = 0.11$ and $R^2 = 0.83$ goodness-of-fit.

We believe that the categorical event scenario factor was dropped during the iterative regression process in favor of the more expressive continuous-valued measures for characterizing interaction experience. More importantly, these experience-based factors were dwarfed in significance compared to user assessments, particularly at the experiment-level time scale, as reflected by the aggregated Mean Sum of squares (MS) and average p -value statistics in Table 4–1. *Therefore, these results*

suggest that short-term trust dynamics were influenced by the supervisor’s personality traits (e.g. expertise, beliefs, tendencies, perceptions) much more so than the actual interaction experiences, which is somewhat surprising and counter-intuitive.

User Feedback

One participant noted during the debriefing questionnaire that he experienced difficulty in quantifying trust changes using the VAS format, specifically in that the range between “absolute lack of trust” versus “full certain trust” was excessively broad compared to most of his short-term trust changes. A few users also indicated that the mouse-based intervention interface was “somewhat non-intuitive and hard to get used to”. Furthermore, several participants who had little prior experiences with visual teleoperated systems and video games felt that the displayed boundary curves were not useful, and instead overwhelmed and distracted their camera view.

We subsequently contacted all study participants to query about these issues, and found that the concerns were isolated only to a small minority of the study population. Nevertheless, we addressed this feedback in the design of our subsequent user study, which will be discussed in Section 4.3.

4.2.5 Predictive Model for Event-Time Trust Changes

Following the analyses above, we made a preliminary effort to build a model for predicting trust changes at the event-time scale. This effort required a critical change in methodology from previous investigations, which quantified the relationships between trust and all types of related factors from the collected dataset. In contrast, our *event-time trust changes model* $\Delta\mathbb{T}$ aimed at predicting moment-to-moment trust changes during interactions with *potentially new* human supervisors while having minimal or no prior knowledge about these users. Although previous results revealed the dominance of personality-based factors over experience-based metrics on event-induced trust changes, such information may not be available to the robot agent. These personality-based factors also tended to be stationary during

short to medium-length interactions, so they would not be able to predict event-time dynamics in the trust state. Therefore, we excluded all factors obtained during the survey and debriefing questionnaires for this predictive modeling effort.

Parametric Model Form

This event-time trust changes model was constructed using the same stepwise regression approach from previous analyses. We quantified the per-session trust change $\Delta \mathbb{T}^{session} \in [-1..1]$ as a weighted linear sum (with weights ω) of several experience-based metrics, namely the tracker failure rate, the user intervention rate, and the distance to ground truth trajectory. These metrics were summarized at both the post-event W -second scope ($\mathbb{E}_i^{post\text{-}event}$) and at the session-wide level ($\mathbb{E}_j^{session}$). In addition, post-session assessments ($\mathbb{A}_k^{session}$) of the robot agent's perceived performance, adaptability, and robustness were also included in the linear model.

$$\begin{aligned}\Delta \mathbb{T}^{session}(W, Q) &= \frac{1}{Q} [\text{round}(Q \cdot \mathbb{T}^{post\text{-}session}) - \text{round}(Q \cdot \mathbb{T}^{pre\text{-}session})] \\ &= \sum_{i,j,k} (\omega_0 + \omega_i \mathbb{E}_i^{post\text{-}event}(W) + \omega_j \mathbb{E}_j^{session} + \omega_k \mathbb{A}_k^{session})\end{aligned}\tag{4.1}$$

This model has two hyper-parameters: the post-event window duration W , and the trust quantization level Q . The window duration W characterized the model's temporal sensitivity to post-event experiences. W can be tuned to match the required sensitivity for a given interaction context, be it our continuous-time control tasks or turn-based episodic tasks. The trust response quantization level Q addressed the separate concern of potential sources of bias in questionnaire responses [15]. Some of these biases were addressed by our study design, such as using the VAS answer format to achieve desirable metric properties [74]. The Q hyper-parameter worked in conjunction by discretizing trust state values to attenuate the variability in exact pixel placements for each questionnaire response.

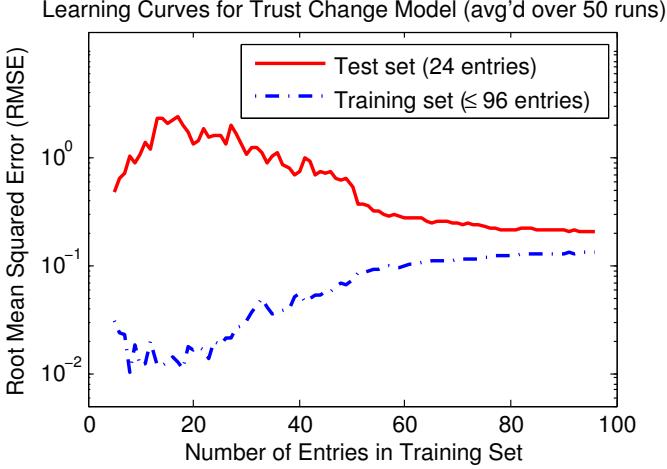


Figure 4–6: Learning curves for the event-time trust changes model $\Delta\mathbb{T}$ (with $W = 2$, $Q = 31$) compared prediction errors of *variable-sized* training sets (up to 80% of the dataset) and of a held-out test set, averaged across 50 independent runs. The asymptotic gap between the two error curves suggest the presence of high variance in this learned model, thus indicating that larger-sized datasets could further improve overall prediction accuracy.

Evaluation of Model Accuracy

We constructed a test set containing a randomly chosen 20% of the study population and carried out hyper-parameter model fitting using 6-fold cross-validation on the remaining dataset. Regression instances were trained for a combination of 11 window duration values ($W \in [0.5, 20]$ sec) and 13 quantization levels ($Q \in [3, 501]$). Among these, the instance with the smallest $RMSE^{xval}$ was designated as the final model form, and had parameter values of $W = 2$ seconds and $Q = 31$ levels. The short 2-second time window was sensible given the fast pace of our visual navigation tasks, whereas the fine-grained quantization level reflected the need to capture minute trust changes at the event-time scale.

The learning curves for the final model are shown in Figure 4–6. Note that the horizontal axis depicts the size of the training set, rather than the learning iteration, and thus the *increase* in training-set error expectedly demonstrated reduced overfitting as the training dataset grew. Prediction accuracies for trust changes trained

on 80% of the study dataset were as follows: $\overline{RMSE}^{train}(96) = 0.13 (\sigma = 0.01)$, $\overline{RMSE}^{test}(24) = 0.19 (\sigma = 0.05)$. The asymptotic convergence in the training-set and test-set errors indicated that this regression model required a reasonably-sized training set to allow for sufficient generalization. More vitally, the asymptotic gap between the two error curves suggested that our predictive model exhibited high variance and possible over-fitting behaviors. Therefore, we could expect to improve prediction accuracy by further expanding the size of our training dataset.

Finally, by interpreting \overline{RMSE}^{test} as the standard deviation in expected prediction errors of event-time trust changes for previously unseen human supervisors, we expect in 95% of cases to see prediction errors within ± 0.37 (recalling that trust change values lied within $[-1..1]$). We thus conclude that this preliminary trust model exhibited only moderate levels of predictive power. This limitation may be attributed to the variability in behavior and responses among different users, which were shown in our previous analyses to have dominant effects on these users' changing trust states. Consequently, our subsequent trust modeling efforts in Chapter 5 will prioritize model personalization as a critical feature.

4.3 Observational Study on Real-Time Trust Dynamics

The previous study assessed event-time impacts on the supervisor's trust state by manipulating the interaction experience. To complement its findings, we also conducted an observational study to collect and evaluate interaction data during extended operations while avoiding the use of experimental conditioning. Participants were asked to provide an additional form of live assessment by critiquing the robot agent's performance, in order to assist in quantifying real-time trust evolutions. This study targeted a remarkably diverse yet domain-specific population of roboticists from multiple institutions belonging to a nation-wide robotics network.

Beyond the primary purpose of building a sizable corpus of interaction datasets toward our real-time trust modeling efforts, this study also compared the relative

importances of diverse experience-based trust factors. In particular, by contrasting the effects on trust between the robot agent’s own performance estimates and various forms of feedback from the supervisor, our investigations aimed to extend and complement findings from the previous event-centric experimental study, which highlighted the dominant influences of personality-based trust factors.

4.3.1 Participants

We recruited 21 participants (1 female) during the NSERC Canadian Field Robotics Network (NCFRN) April 2014 trials to take part in this observational study. The study population consisted wholly of robotics researchers, comprising predominantly of graduate students (86%), as well as few post-doctorates and professors. The average age of these participants was 27 ($\sigma = 4$).

In contrast to the broad diversity of the audience during the event-induced trust changes experimental study, this observational study specifically targeted users who were knowledgeable about the internal logic of autonomous robot systems. Participants had on average 4 years of experience ($\sigma = 3$) programming mobile robots and thus were likely to have more realistic expectations and reactions to such systems. Additionally, compared to the previous study’s population, users exhibited higher propensity toward car automation ($\mu = 71\%$, $\sigma = 20\%$ degree of agreement with “comfort using cruise control”), and also had more operational experience with gamepads ($\mu = 76\%$, $\sigma = 20\%$ “competence”), radio-controlled vehicles ($\mu = 63\%$, $\sigma = 18\%$ “competence”), and visually teleoperated systems ($\mu = 60\%$, $\sigma = 19\%$ “competence”). Despite their expertise toward robotics, participants had geographically diverse backgrounds and were affiliated with 7 universities in 4 separate time zones across Canada.

4.3.2 Infrastructure

In this observational study, our APEX-enabled adaptive boundary tracking agent steered a simulated drone via the SightedTurtleSim framework. Using such

controlled environments ensured that the starting conditions of each session could be exactly specified. The simulated framework also provided accurate live performance metrics in terms of distances to the specified boundary trajectories.

Nevertheless, contrary to our previous experimental study as well as to similar human-robot trust studies [23], neither the robot agent nor the scenes (generated from satellite footage) were manipulated during task scenarios. This observational approach ensured that the recorded experiences would remain as faithful as possible to real-world conditions, given its main purpose serving as training and evaluation datasets for our subsequent real-time trust modeling efforts.

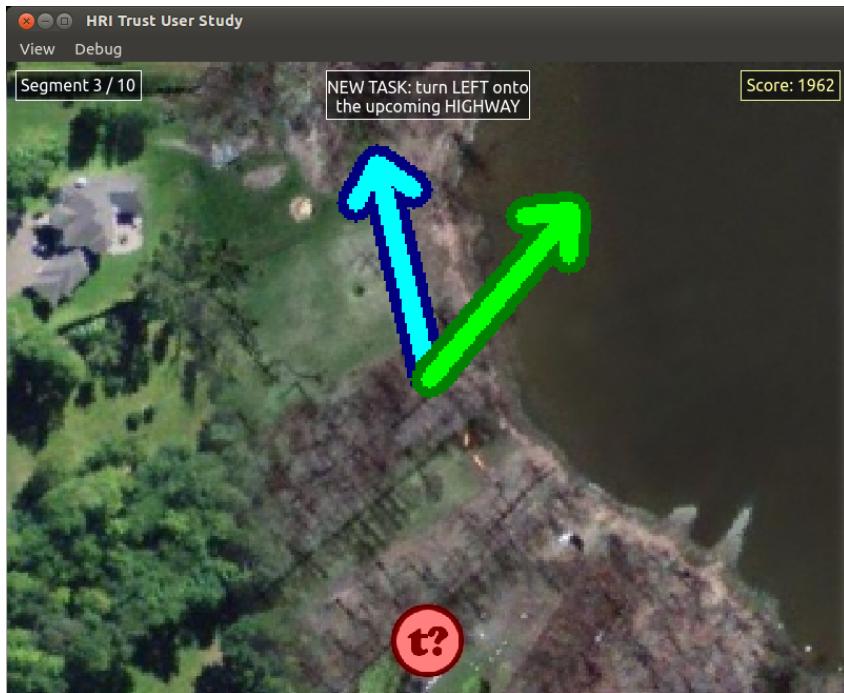


Figure 4–7: The display interface for the real-time trust dynamics study mostly resembled the camera view from the previous event-centric trust changes study. This view depicted the detected boundary (blue line), steering commands from the robot agent and human (as blue/green arrows), as well as text overlays denoting the session ID, the current task goal, and the tracking performance score.

We developed a fully automated study infrastructure that included robust monitoring capabilities and provided visual warnings whenever the robot deviated away

from the specified boundary task. If the participant failed to recover promptly, the interaction session would then reset to a previous checkpoint state.

The display interface was incrementally modified from our previous study and again presented an augmented camera view, as shown in Figure 4–7. Following user feedback from the event-induced trust changes study, the boundary curve was no longer shown on screen to reduce visual clutter. Additionally, the input interface was modified, following previous study feedback, from a mouse-based scheme to a gamepad-based modality. We employed the same gamepad control scheme used during the evaluations of APEX-enabled adaptive robot agents.

We emphasized the importance of the usability of this control interface, especially since participants were asked to provide dynamic critiques on the robot’s performance. Toward this end, we decorated the gamepad with adhesive labels, as shown in Figure 4–8. Furthermore, instead of holding down an explicit button to engage intervention, the supervisor could now steer manually simply by pushing the analog stick in the desired direction, and cede control back to the robot agent by releasing the stick.

The purpose of this observational study was to record many instances of typical human-robot interaction experiences. Every time the boundary tracking agent processed a camera frame, we logged whether it had failed to detect any boundaries (i.e. “agent failures” reflecting task performance $p \in \{0, 1\}$), and the human’s



Figure 4–8: Gamepad interface for the real-time trust dynamics study included additional labels annotating the analog stick for manual interventions use (steer), as well as buttons for issuing trust change critiques ($t+$, $t=$, $t-$).

intervention state $i \in \{0, 1\}$ at that time. We also noted frames in proximity to a change-over between boundary targets, corresponding to external intervention causes $e \in \{0, 1\}$.

We separated multiple study scenarios into minute-length interaction sessions. During pauses between sessions, the user was asked to report on their current trust state $f \in [0, 1]$. This feedback served as vital ground truth data, for training as well as for evaluating our modeling efforts on predicting real-time trust dynamics. In light of previous study feedback that reported difficulty in quantifying trust states temporally, we modified the VAS answer scale by adding labeled anchors, thus turning it into a continuous Likert-like scale. Although the resulting format, shown in Figure 4–9, has potentially distorted metric properties of the original VAS scale, this format was designed to improve the practicality in repeated user responses as well as to mitigate concerns of end-aversion bias [18].

1. Degree of Trust: What is your degree of trust in the robot right now?
Think carefully about your interaction experiences before answering.

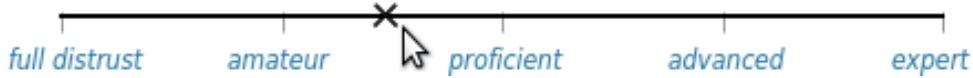


Figure 4–9: The post-session trust feedback in the real-time trust dynamics study extended the VAS format with mid-anchor points to facilitate repeated responses.

Despite improvements to the trust feedback format, we sought to minimize such queries in order to reduce disruptions from the primary task and mitigate added mental strain. Inspired by existing human-robot studies [23, 53], we asked users to report changes in their trust state *during interactions* using gamepad buttons that indicated whether trust has been gained, lost, or remained unchanged, i.e. $c = \{+1, -1, 0\}$. Additionally, the interface encouraged reporting these trust critiques c at 5 – 10 second intervals, both by using visual feedback (with a “t?” icon in Figure 4–7) as well as vibrating the gamepad.

Our study design incorporated gamification elements [95] to incentivize user participation and attention. A numerical score was shown during interaction sessions to reflect the boundary tracking task performance. This score was computed from the accumulated area covered around the specified boundary in each scenario, although the score accrued at a reduced rate of 10% during interventions. Furthermore, each time the session state was reset due to prolonged and excessive deviation from the target boundary, a fixed penalty was deducted from the score. This measure was designed to encourage the user to be attentive and intervene when the robot agent performed poorly, yet also penalize those users who teleoperated the robot even when it was capable of tracking the specified boundary targets. At the end of the study, users could optionally submit their final score to a public leaderboard as a competitive incentive for subsequent study participation. Importantly, this score information was not used in our subsequent real-time trust modeling efforts, as its purpose was solely to motivate user engagement and attentiveness.

This study featured two extensive task scenarios that were divided into minute-length sessions. The first scenario originated from the APEX aerial coverage study and comprised of three terrain boundaries of varying tracking difficulty: a straight highway, a narrow forest path, and a curvy coastline. To complement these tasks, we also devised a flight trajectory around a fjord, featuring watery inlets with visually-distracting side-veins, as well as blurry contours involving icy channels, barren plains, and snow-covered hillsides. These boundary targets were chosen specifically to be more visually challenging than the previous scenario.

4.3.3 Procedure

The flowchart for this study is shown in Figure 4–10. Following a demographics survey, a short slideshow described the study’s purpose and boundary tracking task, and also explained elements of its visual and control interfaces. As in all of

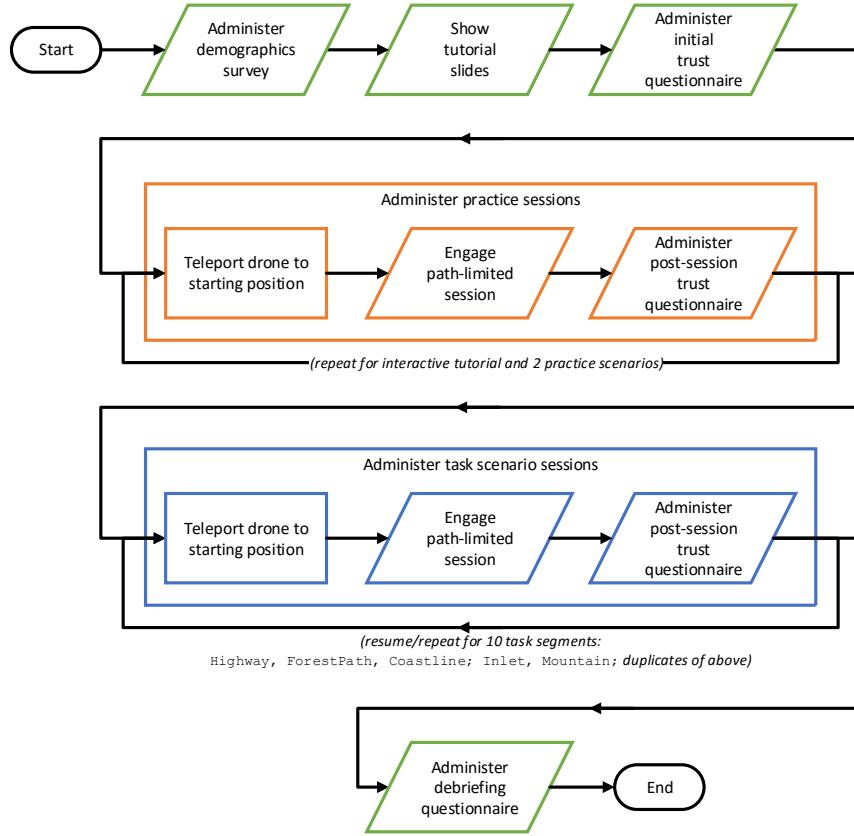


Figure 4–10: Flowchart for the real-time trust dynamics study.

our studies, the tutorial verbosely discussed performance-centric versus intention-centric facets of trust assessment, and asked users to assume away intention-centric bases. The user then worked through an interactive tutorial and 2 practice sessions to familiarize with the tracking tasks, input interface, and trust feedback queries.

After these practice sessions, the study explicitly announced the start of the main interaction sessions. A total of 10 recorded sessions ensued, which comprised of a 3-session highway-forest-path-coastline scenario followed by a 2-session inlet-mountain-inlet fjord scenario, and then followed by repeats of both scenarios again. All participants experienced these task sessions in the same order, both since the practice sessions were designed to mitigate initial learning effects, and because the study was primarily aimed at collecting interaction experiences rather than assessing the effects of manipulated events.

4.3.4 Results and Discussion

This section reports on several descriptive analyses of the recorded study dataset, as well as post-study feedback from participants. Both forms of findings along with the interaction dataset itself will have major influences toward shaping the elements, structure, and instantiations of our real-time human-robot trust modeling efforts, which will be elaborated in Chapter 5.

Temporal Traits of User Behavior and Trust

The 2 task scenarios were each instantiated twice during each study run to assess whether users behaved consistently in similar situations. No significant differences in the rate of human interventions i per matching sessions were found (2-tailed paired $t_{104} = 1.49, p = 0.14$), and the numerical sum of trust change critiques c across session pairs also did not reveal any significant differences (2-tailed paired $t_{104} = -0.43, p = 0.67$). In contrast, trust feedback f were found to be noticeably different when users repeated the same scenarios (1-tailed paired $t_{104} = -4.85, p \ll 0.01$).

These results suggest that users reacted consistently to similar events, yet their trust assessments changed over time as they accumulated more interaction experiences. Therefore, our findings substantiate the need to model *temporal dynamics* toward accurately predicting the supervisor's trust state during interactions.

Impacts of Interaction Factors on Trust

A linear regression on trust feedback f was carried out to identify significant covariates from the interaction experience. An analysis of variance showed that both the user identifier ($F_{20,188} = 17.4, p \ll 0.01$) and the ratio of user interventions i per session ($F_{1,207} = 76.2, p \ll 0.01$) were significantly correlated with trust feedback f , whereas the ratio of agent failures per session was related to a lesser degree ($F_{1,207} = 3.02, p = 0.08$).

The strong dependence of trust feedback on each user’s interventions supports the need for a *personalized* trust model. This finding is also consistent with results from our event-induced trust changes study and has been corroborated by other human-robot studies as well [23]. Our real-time trust dynamics model in Chapter 5 will explicitly account for the dominant effects of user interventions over the robot agent’s performance estimates on trust in its model structure.

User Feedback

Participants provided several useful remarks in their debriefing questionnaire that reflected vital insights about the evolution of their trust state in the robot’s task performance. Several users indicated that their “trust changed when the robot did something unpredictable”, which suggests a dependency between the trust state t_k at time k and the *change* in the robot’s recent task performance, i.e. $p_k - p_{k-1}$. Others said that their “trust fluctuated a lot initially” given the lack of prior experiences with the robot. This suggests that it is sensible to assume a *uniform prior* belief the initial degree of trust when interacting with a new autonomous robot system.

During pilot runs of the study, users frequently pressed the “trust gained” and “trust lost” buttons unintentionally when prompted for c . Consequently, the slideshow in the final study form explicitly encouraged participants to press the “trust unchanged” button as a default. Nevertheless, multiple participants reported that they found it “hard to suppress the urge to press ‘trust gained’ or ‘trust lost’” hastily and recalled making multiple accidental misclicks. This *idling bias* will be taking into account in our trust dynamics model, specifically in quantifying the relationship between the latent trust state t_k and trust change critiques c .

4.4 Summary

In this chapter, we collected a large corpus of interaction experiences from a wide range of roboticists and enthusiasts, who collaborated with our robot agents

in a variety of scenarios. Analyses of these datasets revealed the following characteristics of trust for asymmetric supervisor-worker human-robot teams:

1. Trust feedback assessments were correlated with each supervisor's personality and propensity dominantly, while factors emerging from the interaction experience had notably less influence.
2. Among experience-based factors, the supervisor's intervention frequency was more strongly correlated with trust feedback than the robot agent's internal failure assessments.
3. When the agent misbehaves, the human supervisors' reactions and trust assessments were consistent with *rational* responses based on the cause of the misbehavior, (e.g. lack of initial training, versus limitation of agent capabilities, versus inexplicable failure).
4. Although supervisors exhibited consistent reactions to similar events, their trust states evolved differently during repeated scenarios.
5. Supervisors reported difficulty in confidently providing initial trust assessments due to lack of hands-on experience with a new robot agent.

All of these findings along with their raw interaction datasets will directly contribute to the development of a data-driven model of trust dynamics, as well as an associated online computation engine. The two interaction studies also gave us opportunities to improve the quality of the user interface and the automated study infrastructure, both via pre-study testing, as well as post hoc user feedback. Our trust dynamics modeling efforts discussed in the next chapter will ultimately enable autonomous robot agents to be able to sense and react to their human supervisor's trust state toward maintaining effective partnerships.

Chapter 5

Real-Time Human-Robot Trust Modeling

In this chapter, we present OPTIMo: an Online Probabilistic Trust Inference Model for inferring the human’s moment-to-moment trust state during interactions with an autonomous robot agent. This computational model formulates Bayesian beliefs about the supervisor’s latent trust state based on recently observed interaction experiences. A separate model instance is trained on each supervisor’s experiences and assessments, leading to an *interpretable* and *personalized* characterization of that individual’s behaviors and attitudes.

These trust modeling efforts correspond to the second stage in our three-stage plan for realizing trust-seeking robots. An important model design requirement is the ability to predict the supervisor’s trust state at frequent intervals, as trust information “in general varies very rapidly” [13]. This real-time attribute also enables the robot agent to react responsively to the human’s trust changes toward actively maintaining efficient teamwork.

This chapter begins with a discussion on our modeling approach for predicting real-time trust dynamics. We next describe OPTIMo’s core elements, including its model structure, its data preprocessing procedure, its inference and training mechanisms, as well as details on a concrete model implementation. Based on the dataset from our trust dynamics study (see Section 4.3), we also delve into a detailed evaluation of OPTIMo that includes an exposition of a model instance and its trust

inference outcomes, as well as an investigation into the impacts of configurable parameters of this model. We further present a quantitative comparison with respect to several existing trust models and demonstrate that trained OPTIMo instances can predict each human supervisor’s reported trust feedback with *greater accuracy* and *at faster time scales* compared to previously proposed methods.

5.1 Methodology

As highlighted in Section 2.4, our research considers the supervisor’s trust state as a continuous and bounded value, spanning complete distrust and absolute trust. We take a performance-centric trust modeling approach that is predicated on two simple observations of supervisor-worker human-robot teams. Firstly, we note that *the robot agent’s trustworthiness arises due to its task performance*: good performance and progress should lead to greater trust, whereas low reliability would likely induce trust loss. Secondly, we have observed that *when the human intervenes, it often reflects a lapse in trust* due to the robot’s task failures.

These observations adhere to separate approaches for modeling trust, through *causal* reasoning about the agent’s expected trustworthiness [56], and by incorporating user-provided *evidence* into an estimate of the true trust state [23]. By combining both the agent’s own performance estimates and the human’s reactions and feedback into a single computational framework, our work aims to unify both types of existing trust modeling approaches.

Formally, our modeling effort addresses the problem of estimating the human’s trust state $t_k \in [0, 1]$ regarding the robot agent at various time steps $k \in [1, K]$ during their interactions. We address this problem by estimating a probability distribution (or “belief”) over this *latent* trust signal, and by updating the belief over time through correlations with other *observable* factors. Similar to our regression-based modeling efforts in Section 4.2.5, we focus solely on trust factors that are obtainable during deployment. This approach ensures that we can train our model

to predict trust feedback from unseen users by simply observing an initial period of their interactions with the robot and without administering lengthy questionnaires.

The various trust factors under consideration include the well-established link between trust and the robot's *instantaneous task performance* p [23, 56]. The robot agent can estimate task performance by noting the rate of its algorithmic failures. Another important factor is the *intervening state* i , which reflects whether the supervisor is currently intervening or not. This signal has been shown to strongly correlate with the human's trust state, as demonstrated by our user studies (Chapter 4) and in the existing literature [23, 63]. We additionally consider *external factors* e that may cause the operator to intervene irrespective of trust, such as when the user switches the task target by steering the robot along a new terrain boundary. Finally, our modeling approach incorporates several types of assessment-based factors from the human supervisor, including *trust change critiques* c that acknowledge task successes and mistakes, as well as questionnaire-based *trust feedback* f that quantify their current trust state at different moments during the interaction.

5.2 Online Probabilistic Trust Inference Model (OPTIMo)

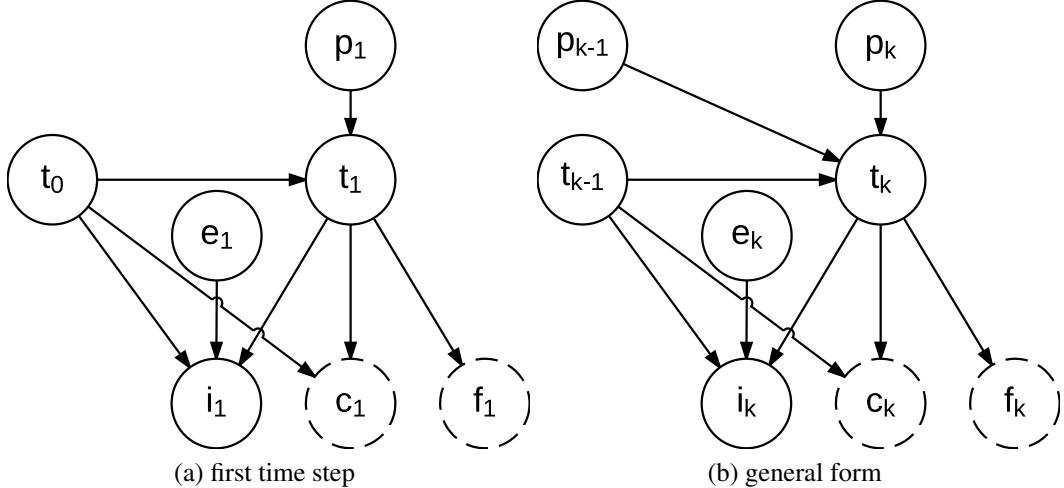


Figure 5–1: Our Online Probabilistic Trust Inference Model (OPTIMo) is represented using a Dynamic Bayesian Network. Dashed circles in this graph structure depict optional interaction factors that are not observed on all time steps k .

OPTIMo's structure is represented as a Dynamic Bayesian Network [54] and is shown in Figure 5–1. This graph structure efficiently encodes *local relationships* between the human's latent trust state t_k and related factors, as well as the temporal dynamics of trust itself. The Bayesian Network formulation also has the advantage of being able to process interaction factors that are observed at different time scales in a mathematically sound manner. Furthermore, by using a probabilistic representation for trust estimates, OPTIMo can capture useful traits such as the *expected* trust state at a given time and the degree of *uncertainty* associated with each estimate.

5.2.1 Data Preprocessing

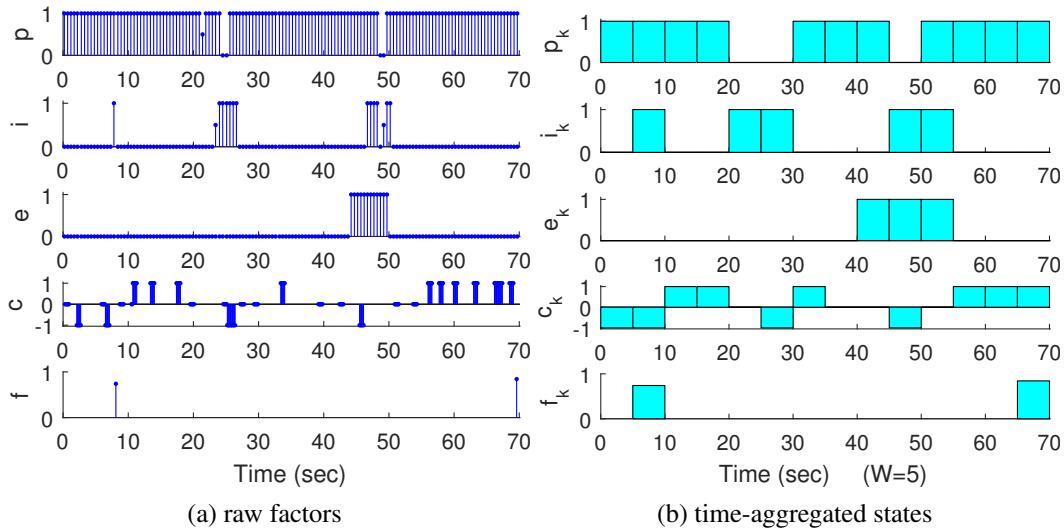


Figure 5–2: Window-based preprocessing of interaction factors for the Online Dynamic Probabilistic Trust Inference Model (OPTIMo).

As illustrated by Figure 5–2, OPTIMo discretizes a continuous period of interaction into a sequence of K non-overlapping time windows, $k = 1: K$, each lasting W seconds. The window-aggregated state of task performance $p_k \in [0, 1]$ is computed as the ratio of frames within the k -th window for which the agent

failed to produce any commands. $i_k \in \{0, 1\}$ is asserted when the supervisor intervenes at some point during the k th time window. Similarly, the external cause state $e_k \in \{0, 1\}$ records the presence of a change in task target during step k .

If the supervisor reports one or more trust change critiques c during a time window k , then these values are summed and transformed into a single critique value $c_k \in \{-1, 0, +1, \emptyset\}$ using the sign function:

$$sgn(\Sigma) := \begin{cases} +1, & \text{if } \Sigma > 0 \\ 0, & \text{if } \Sigma = 0 \\ -1, & \text{if } \Sigma < 0 \end{cases}$$

This mapping identifies the dominant critique value in each time window and is especially useful in cases where the user reports conflicting critiques $\{c\}$ in quick succession due to the volatility in their trust assessments.

Finally, since trust feedback instances f are expected to occur infrequently, we assume that each time step has *at most* one feedback value $f_k \in \{[0, 1], \emptyset\}$.

5.2.2 Local Trust Relationships

Each edge in OPTIMo’s graph structure represents a relationship between the latent trust state t_k to a related factor and is modeled as a conditional probability distribution (CPD). These relationships have all been previously corroborated in the literature [23, 56] as well as by our empirical studies in Chapter 4. For instance, our observational study on real-time trust dynamics found that user interventions i were much more strongly correlated with trust than the robot’s task performance estimates p . Consequently, OPTIMo uses p_k to *propagate* the trust belief t_k to a set of *plausible* next states, whereas human-provided assessment factors i_k, c_k, f_k are then used to *exclude* inconsistent trust hypotheses. The links from p_k to t_k to assessment factors also reflect a natural and causal depiction of the supervisor’s decision process that is driven by trust internally.

We now explore each of these local relationships in turn. For instance, we use a bounded linear Gaussian CPD to model the amount for which trust t_k is expected to change given the robot's recent and current performance estimates p_{k-1}, p_k :

$$\begin{aligned}\mu_{t_k} &:= t_{k-1} + \omega_{tb} + \omega_{tp} p_k + \omega_{td} (p_k - p_{k-1}) \\ \mathcal{P}(t_k, t_{k-1}, p_k, p_{k-1}) &:= \text{Prob}(t_k | t_{k-1}, p_k, p_{k-1}) = \end{aligned}\tag{5.1}$$

$$\frac{\mathcal{N}(t_k; \mu_{t_k}, \sigma_t)}{\Phi(1; \mu_{t_k}, \sigma_t) - \Phi(0; \mu_{t_k}, \sigma_t)}$$

where $\mathcal{N}(x; \mu, \sigma) := \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right)$ and $\Phi(x; \mu, \sigma) := \int_{-\infty}^x \mathcal{N}(y; \mu, \sigma) dy$ denote the Gaussian probability distribution and cumulative distribution, respectively, for the random variable x with mean μ and standard deviation σ . The mean expression μ_{t_k} of this bounded Gaussian CPD represents the expected update to trust t_k from its previous state t_{k-1} as a weighted sum expression. The personalized parameters $\omega_{tb}, \omega_{tp}, \omega_{td}$ reflect the relative impacts on each user's trust updates of the **bias**, the current task **performance**, and the **difference** in the robot's performance. The propagation uncertainty parameter σ_t quantifies the variability in each user's trust update dynamics.

The propagation step expands the range of hypotheses of potential trust states that are consistent with the robot agent's latest performance estimate. To narrow down the possibility of propagated hypotheses, the model checks if each candidate state is consistent with the human's latest reactions (i.e. i_k) and assessments (i.e. c_k, f_k). OPTIMo explains the likelihood of interventions i_k based on several potential causes, such as the current trust state t_k , a recent change in trust $\Delta t_k := t_k - t_{k-1}$, and external intervention causes e_k . These linkages are modeled as a logistic CPD:

$$\mathcal{O}_i(t_k, t_{k-1}, i_k = 1, e_k) := \text{Prob}(i_k = 1 | t_k, t_{k-1}, e_k) = \quad (5.2)$$

$$\mathcal{S}(\omega_{ib} + \omega_{it} t_k + \omega_{id} \Delta t_k + \omega_{ie} e_k)$$

$$\text{Prob}(i_k = 0 | t_k, t_{k-1}, e_k) = 1 - \text{Prob}(i_k = 1 | t_k, t_{k-1}, e_k)$$

where $\mathcal{S}(x) := (1 + \exp(-x))^{-1}$ is the sigmoid distribution for the binary random variable x . The parameters $\omega_{ib}, \omega_{it}, \omega_{id}, \omega_{ie}$ separately account for the likelihood of intervention $i_k = 1$ arising from personal **bias** (i.e. predisposition to micromanage), low **trust**, **drop** in trust, and **external** factors (e.g. switching task targets; wind perturbation).

During time steps when the user reports trust changes $c_k \in \{-1, 0, +1\}$, these are accounted as evidence to ground the latest update to latent trust, Δt_k . The likelihoods of observing a “trust gained” or “trust lost” critique are modeled as sigmoid CPDs, while the “trust unchanged” likelihood is implicitly defined via exclusion:

$$\begin{aligned} \mathcal{O}_c(t_k, t_{k-1}, c_k) &:= \text{Prob}(c_k | t_k, t_{k-1}) \\ \text{Prob}(c_k = +1 | t_k, t_{k-1}) &= \beta_c + (1 - 3\beta_c) \cdot \mathcal{S}(\kappa_c [\Delta t_k - o_c]) \\ \text{Prob}(c_k = -1 | t_k, t_{k-1}) &= \beta_c + (1 - 3\beta_c) \cdot \mathcal{S}(\kappa_c [-\Delta t_k - o_c]) \\ \text{Prob}(c_k = 0 | t_k, t_{k-1}) &= 1 - \text{Prob}(c_k = +1 | t_k, t_{k-1}) \\ &\quad - \text{Prob}(c_k = -1 | t_k, t_{k-1}) \end{aligned} \quad (5.3)$$

These CPDs parametrize the nominal offset o_c in a change to latent trust Δt_k that is likely to cause the user to report a non-zero c_k , along with the variability κ_c in the reporting likelihoods. The uniform error term β_c accounts for the *idling bias* identified in our observational study on trust dynamics, in which users hastily pressed wrong critique buttons when prompted. The effects of these parameters on the resulting sigmoid CPD are depicted in Figure 5–3.

Finally, OPTIMo uses a bounded Gaussian CPD to quantify the degree of uncertainty σ_f between the user's latent trust state t_k and their trust feedback f_k :

$$\mathcal{O}_f(t_k, f_k) := \text{Prob}(f_k | t_k) = \frac{\mathcal{N}(f_k; t_k, \sigma_f)}{\Phi(1; t_k, \sigma_f) - \Phi(0; t_k, \sigma_f)} \quad (5.4)$$

5.2.3 Inference, Personalization, and Prediction

OPTIMo's main purpose is to infer the probability that the human's trust state $t_k \in [0, 1]$ at a given time step k takes on a particular value. Two types of trust inferences are of particular interest: firstly, the *filtered belief* estimates the trust state at the *current* time step k in an online fashion based on a history of past experiences, i.e. $\text{bel}_f(t_k) = \text{Prob}(t_k | p_{1:k}, i_{1:k}, e_{1:k}, c_{1:k}, f_{1:k}, t_0)$. In contrast, given a sequence of recorded experiences, the *smoothed belief* at *any* time step $k \in [0, K]$ can be computed offline by incorporating interaction factors that occurred both prior and after k , $\text{bel}_s(t_k) = \text{Prob}(t_k | p_{1:K}, i_{1:K}, e_{1:K}, c_{1:K}, f_{1:K}, t_0)$.

We derived expressions for both inference processes based on OPTIMo's graph structure through repeated applications of Bayes' rule, simplifications using the Markovian assumption, and variable marginalization [97]. These derivations resulted in succinct recursive definitions for the filtered and smoothed beliefs, which both depend on a joint likelihood expression for successive latent states $\overline{\text{bel}}(t_k, t_{k-1})$:

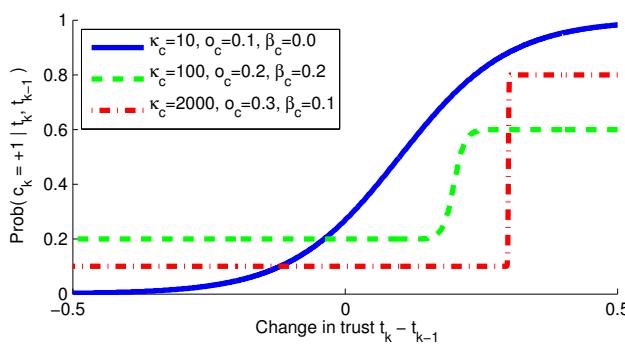


Figure 5–3: The effects of changes in latent trust $\Delta t := t_k - t_{k-1}$ to the likelihood of a “trust gained” critique report $\text{Prob}(c_k = +1)$ is modeled as a sigmoid CPD. Its parameters include the nominal trust change offset o_c , the critique reporting variability κ_c , and a misclick error term β_c .

$$\begin{aligned} \overline{bel}(t_k, t_{k-1}) &:= \mathcal{O}(t_k, t_{k-1}, i_k, e_k, c_k, f_k) \\ &\quad \cdot \mathcal{P}(t_k, t_{k-1}, p_k, p_{k-1}) \cdot bel_f(t_{k-1}) \end{aligned} \quad (5.5)$$

$$bel_f(t_k) = \frac{\int \overline{bel}(t_k, t_{k-1}) dt_{k-1}}{\iint \overline{bel}(t_k, t_{k-1}) dt_{k-1} dt_k} \quad (5.6)$$

$$bel_s(t_{k-1}) = \int \frac{\overline{bel}(t_k, t_{k-1})}{\int \overline{bel}(t_k, t_{k-1}) dt_{k-1}} \cdot bel_s(t_k) dt_k \quad (5.7)$$

where $\mathcal{O}(t_k, t_{k-1}, i_k, e_k, c_k, f_k)$ denotes the product of $\mathcal{O}_i(\cdot)$ with possibly $\mathcal{O}_c(\cdot)$ and/or $\mathcal{O}_f(\cdot)$, depending on whether $c_k \neq \emptyset$ and/or $f_k \neq \emptyset$.

Both the filtering and smoothing algorithms have efficient runtime complexities that are linear in terms of the number of time steps K . The filtered belief $bel_f(t_k)$ in Equation 5.6 is updated recursively from its previous distribution $bel_f(t_{k-1})$; a similar recursive relationship also holds for the smoothed belief $bel_s(t_k)$ in Equation 5.7. Therefore, starting from a given prior trust belief $bel_f(t_0)$, one can compute filtered beliefs $bel_f(t_k)$ forward in time in a single pass, and then sequentially ascertain smoothed beliefs $bel_s(t_k)$ backward in time.

Our OPTIMo implementation uses an uniform prior trust belief, $bel_f(t_0) := Prob(t_0) = 1$, which enforces the assumption that *the human supervisor has uniform prior trust when interacting with a robot agent for the very first time*. This assumption was substantiated by feedback from our trust dynamics study, where participants expressed difficulty in confidently providing initial trust assessments prior to any interactions with the robot.

Most parameters of OPTIMo's CPDs aim to capture the supervisor's behaviors and trust tendencies. These *personalized parameters* Θ include trust propagation settings $\{\omega_{tb}, \omega_{tp}, \omega_{td}\}$, intervention likelihood weights $\{\omega_{ib}, \omega_{it}, \omega_{id}, \omega_{ie}\}$, and parameters for the trust change likelihood $\{o_c, \kappa_c, \beta_c\}$. OPTIMo also has a few parameters that do not capture personalized tendencies, such as the window duration W ,

the trust propagation variability σ_t , and the trust feedback uncertainty σ_f . The window duration W determines this model’s time scale, and should be set accordingly (e.g. to obtain real-time second-scale updates or capture minute-scale dynamics). σ_t reflects the amount of variability in the change in trust over consecutive time steps, and is affected both by the time window duration W and by the pace of the task for the human-robot team. σ_f captures the degree of uncertainty between the latent trust state t_k and a noisy trust feedback f_k obtained via questionnaire.

We tailor personalized OPTIMo instances for each individual user by optimizing the parameters Θ based on a *training set* of their interaction experiences. This is achieved using the hard-assignment Expectation-Maximization (“Hard EM”) algorithm [54]. Hard EM jointly optimizes the observational likelihood of the training data and the most likely sequence of latent trust states, as follows:

$$\Theta^* = \arg \max_{\Theta} \max_{t_{1:K}} \text{Prob}(t_{1:K}, p_{1:K}, i_{1:K}, e_{1:K}, c_{1:K}, f_{1:K} | t_0)$$

In addition to inferring trust beliefs, OPTIMo can also be used to predict probability distributions for the observed factors i_k, c_k, f_k . Expressions for these predictive beliefs are derived in the same manner as filtering and smoothing processes [97]. For instance, when targeting the intervention state, we first omit all observed instances $i_{1:k}$ from consideration, and then predict the likelihood that i_k will take on a particular value as:

$$\begin{aligned} \text{Prob}(i_k | p_{1:k}, e_{1:k}, c_{1:k}, f_{1:k}) &= \\ &\frac{\iint \mathcal{O}_i(t_k, t_{k-1}, i_k, e_k) \cdot \overline{\text{bel}}(t_k, t_{k-1}) dt_k dt_{k-1}}{\iint \overline{\text{bel}}(t_k, t_{k-1}) dt_k dt_{k-1}} \end{aligned} \quad (5.8)$$

5.2.4 Histogram Inference Engine

We implemented OPTIMo using a histogram-based inference method [91], which approximates continuous belief densities as discrete vectors of probability masses. Specifically, the bounded interval $[0, 1]$ for the trust state is discretized into

B equally spaced bins, where the likelihood of t_k taking on a particular value is approximated as the probability mass of the nearest bin center. The precision of this histogram approximation improves when using larger bin sizes B at the cost of additional computations. We also used this histogram approximation to predict trust feedback $f_k \in [0, 1]$, by dividing the range into 100 bins¹ and computing the probability masses of each bin’s center using a form similar to Equation 5.8.

We personalized model instances using EM with multiple restarts to avoid convergence to local optima. In each EM run, randomly-initiated model parameters Θ are iteratively improved using constrained least squares optimization. In each E-step, we calculate the smoothed trust beliefs at all time step $k = 1 : K$, and then compute their expected values $\mathcal{E}(\text{bel}_s(t_k))$. In the following M-step, we update model parameters using an off-the-shelf non-linear optimization solver. This iterative process is terminated when parameters have stabilized within expected tolerances or after a maximum number of iterations have lapsed.

5.3 Experimental Validation

We carried out several empirical assessments to quantify the performance of OPTIMo, using the recorded interaction dataset from the 21 participants of our observational study on trust dynamics (Section 4.3). In this section, we present a visualization of the inference process and outputs of a trained OPTIMo instance, and also highlight several features of this human-robot trust model. We also study the effects on OPTIMo’s performance of its non-personalized model parameters W , B , σ_t , and σ_f . Finally, the next section will present a quantitative comparison of OPTIMo against several existing temporal trust models.

¹ Sensitivity analysis found negligible differences in prediction accuracies beyond 100 bins; we thus chose this bin size to expedite the computation runtime.

5.3.1 Procedure

Since we assume that the human’s trust state t_k is latent and therefore not directly observable, we evaluate OPTIMo based on the accuracies in its predictions for intervention states i_k , trust change critiques c_k , and absolute trust feedback f_k . Given specific values for OPTIMo’s non-personalized parameters, we train and evaluate model instances for each study participant’s dataset individually. This process begins by preprocessing raw interaction experience into W -second windowed aggregates. We then optimize model parameters Θ using a *training set* consisting of experiences from the first 5 study sessions, while assuming uniform prior trust. Next, we separately predict the factors $i_{1:k}$, $c_{1:k}$, and $f_{1:k}$ within the *test set*, comprising of the remaining 5 study sessions, while using the last inferred trust belief from the training set as prior. Finally, we determine Maximum Likelihood Estimates (MLE) from the predictions of intervention, critique, and trust feedback factors, and compare these estimates against user-reported values.

The metrics of this evaluation process include prediction accuracies for intervention states and trust change critiques, $acc_i, acc_c \in [0\%, 100\%]$. MLE predictions for trust feedback are also evaluated using the root-mean-squared-error $RMSE_f$ and the Pearson product-moment correlation coefficient ρ . This latter metric is particularly useful for comparing OPTIMo against trust models that use different (e.g. unbounded) representation scales.

5.3.2 Characteristics of a Trained OPTIMo Instance

We begin by studying the features of a typical OPTIMo instance, which was trained on one study participant’ dataset. This model instance was configured with the following settings: $W = 3\text{ s}$, $B = 300$ bins, $\sigma_t = 0.005$, $\sigma_f = 0.1$. Figure 5–4a depicts time-aggregated factors from the 5 training sessions, as well as the resulting *smoothed* trust beliefs $bel_s(t_k)$ during this interval. Similarly, Figure 5–4b shows the *filtered* trust beliefs $bel_f(t)$ during the subsequent 5 test sessions, as

well as Maximum Likelihood Estimate (MLE) predictions for trust feedback f_k , user interventions i_k , and trust change critiques c_k . The switch from smoothing to filtering allows the model to be evaluated in an online manner as if the inferences and predictions were computed live during the test sessions.

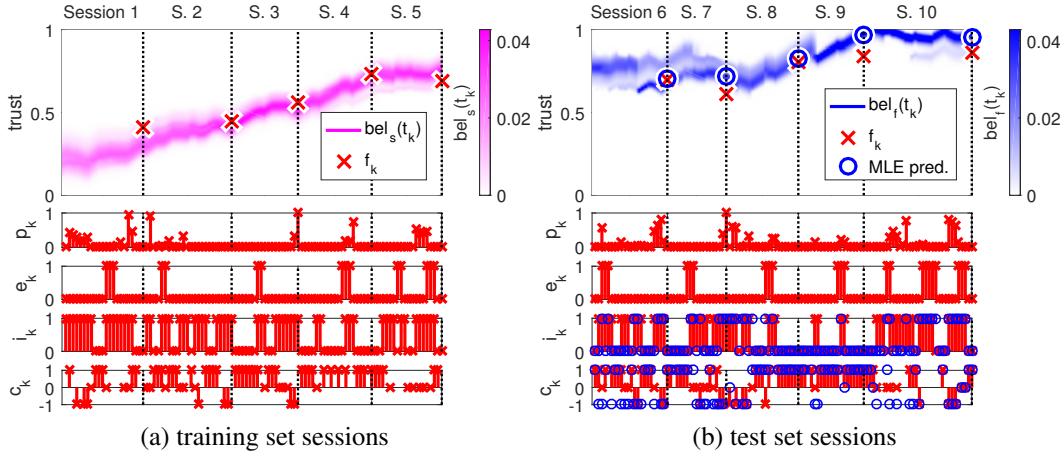


Figure 5–4: Visualizations of a trained OPTIMO instance ($W = 3$ s, $B = 300$ bins, $\sigma_t = 0.005$, $\sigma_f = 0.1$): OPTIMO takes as input time-aggregated factors from the interaction experience, including the agent’s task performance estimate p_k , the supervisor’s intervention state i_k , external intervening factors e_k , trust change critiques c_k , and trust feedback f_k ; model parameters are batch-optimized using EM based on training dataset; outputs of OPTIMO comprise of batch-smoothed trust beliefs $bel_s(t_k)$, online-filtered beliefs $bel_f(t_k)$, and MLE predictions for f_k, i_k, c_k .

The inferred trust beliefs, each shown as a vertical slice in Figure 5–4, reflect a precise characterization of this user’s trust tendencies, as seen from accurate test-set prediction results: $RMSE_f^{test} = 0.09$, $acc_i^{test} = 72\%$, $acc_c^{test} = 70\%$. Also, despite only having 5 trust feedback values each for training and evaluation, test-set predictions of f_k achieved strong correlation $\rho = 0.91$ ($p < 0.01$).

These visualizations further highlight OPTIMO’s unique ability in quantifying *multi-modal beliefs* of a person’s trust state. These competing hypotheses may arise due to momentary contradictions among observed factors p_k, i_k, c_k, f_k . For

instance, consider a situation where the agent repeatedly fails at tracking the boundary task, yet the supervisor inattentively continues to report “trust gained”. Multi-modal distributions may also occur when the user’s reactions are notably different from historical experiences. As result of this multi-modal trait, it is preferable to reason about changes in the human’s trust state by comparing the *expected values* of the inferred beliefs rather than their *modes*, since the latter signal may exhibit sudden discontinuous jumps.

Table 5–1 enumerates the trained parameter values for this personalized OPTIMo instance. We can derive meaningful characterizations of this human supervisor’s *particular trust tendencies* by first substituting these parameters into the CPDs reflecting local trust relationships in Section 5.2.2 and then considering assorted scenarios for the evolution of the trust state.

Table 5–1: Personalized parameter values from an OPTIMo instance trained on a single study participant’s dataset.

Trust Propagation			
$\omega_{tb} = -0.0089$	$\omega_{tp} = 0.0153$	$\omega_{td} = 0.0029$	
Intervention Likelihood			
$\omega_{ib} = 131.2$	$\omega_{it} = -157.1$	$\omega_{id} = -9887$	$\omega_{ie} = 83.84$
Trust Change Likelihood			
$o_c = 0.0003$	$\kappa_c = 1277$	$\beta_c = 1.063 \times 10^{-7}$	
Non-Personalized Parameters			
$W = 300$ s	$B = 300$ bins	$\sigma_t = 0.005$	$\sigma_f = 0.1$

To illustrate this process, let us consider different ways that the intervention likelihood CPD in Equation 5.2 will predict that this user will intervene with near certainty, i.e. $Prob(i_k = 1 | t_k, t_{k-1}, e_k) \approx 1$. For instance, suppose that the trust state is stationary but less than maximal ($t_k = t_{k-1} = 0.8$), and assume that there is currently no external cause for intervention ($e_k = 0$), then we can compute the output of the CPD as:

$$Prob(i_k = 1 | t_k = 0.8, t_{k-1} = 0.8, e_k = 0) = \mathcal{S}(131.2 + (-157.1) \cdot 0.8) = 0.9960$$

Alternatively, this OPTIMo instance will also predict high intervention likelihoods when the user’s trust changes by $\Delta t_k = -0.004$, or when he is intentionally switching between task targets ($e_k = 1, t_k = t_{k-1} = 1$).

By applying similar procedures, we can compare hypothetical values for the current trust state t_k against the mean estimate μ_{t_k} from the trust propagation CPD in Equation 5.1. Consequently, we note that between successive time steps of $W = 3\text{ s}$, this OPTIMo instance will make the following predictions:

- trust will increase by 0.0064 when the agent performs well ($p_k = p_{k-1} = 1$);
- trust will drop by 0.0118 upon an initial agent failure ($p_k = 0, p_{k-1} = 1$);
- trust will continue to drop by 0.0089 ($p_k = p_{k-1} = 0$) until the agent recovers.

Based on the interpretations above, our OPTIMo instance suggests that this supervisor will penalize agent failures with 38% *more* trust loss in comparison to the amount of trust gained during competent operations. We anecdotally observed similar attitudes among many other study participants, which naturally reflect the age-old adage that “it is easy to lose trust, but hard to regain it”.

5.3.3 Effects of Non-Personalized Parameters

Next, we investigated the effects of different non-personalized model parameters on OPTIMo’s predictive performance. These include the window duration W , which dictates the model instance’s time scale and also reflects the interval between successive trust inferences. The number of histogram bins B affects the precision of the discrete approximation to the underlying continuous trust beliefs. The variability in propagated trust states σ_t captures the degree of volatility in successive trust updates. This parameter is affected by both the natures of the task and the interaction context as well as the chosen time scale W for an OPTIMo instance. Finally, σ_f represents the amount of uncertainty that relates trust feedback values f_k to the latent trust state t_k , and accounts for the noise associated with the questionnaire answer format used to elicit f_k .

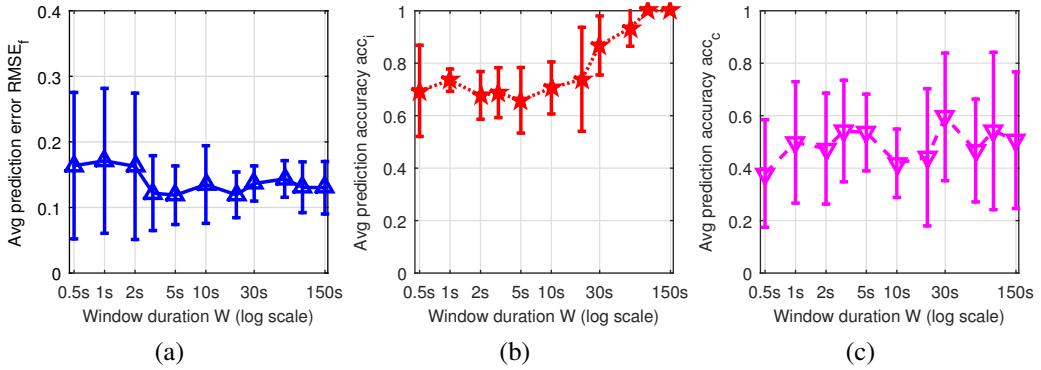


Figure 5–5: Means and standard deviations (as error bars) of OPTIMo’s prediction performances as functions of the window duration parameter W (with $B = 300$ bins, $\sigma_t = 0.005$, $\sigma_f = 0.1$).

Figure 5–5 illustrates the effects of the time window duration W on OPTIMo’s performance. Both the magnitude and variance of prediction errors for trust feedback $RMSE_f$ consistently decreased for wider window periods. This trend can be attributed to the availability of frequent f_k observations at coarser time scales, thus ensuring that trained trust dynamics t_k can more accurately predict f_k . In contrast, prediction accuracies for i_k and c_k dropped slightly as W increased from a 1 s scale to the 10–20 s range. The model’s decreased performance likely resulted from having fewer training exemplars, after the raw per-frame i and per-report c occurrences were collapsed into fewer time-aggregated states. This potential cause also explains the increase in prediction accuracies at the session-wide scale of $W = 150\text{ s}$ as a result of statistical degeneracy.

The opposite effects of W on predictions for f_k , versus those for i_k and c_k , reflects the natural contrast between absolute trust states summarizing *cumulative* experiences, versus the *event-reactive* natures of interventions and trust change critiques. This contrast is commonly seen in human-robot interactions, as reflected by

the observed difference between trust assessments versus user reactions in our observational study on trust dynamics, as well as differences seen in related studies between a cumulative trust scale $Trust_{Muir}$ and a real-time measure $Trust_{AUTC}$ [23]. The fact that our results captured this contrast demonstrates OPTIMo’s versatility.

Furthermore, despite having worse $RMSE_f$ results for sub-second time windows (e.g. $W = 0.5\text{ s}$), competent prediction accuracies for i_k and c_k nevertheless reflect the usefulness of the inferred trust beliefs at these extremely fine time scales. More importantly, all three metrics revealed OPTIMo’s exceptional prediction performances at the $W = 3 - 5\text{ s}$ range. This degree of prediction responsiveness is unseen in existing trust models, which operated at much coarser scales of tens of minutes or longer [23, 56].

We take a brief tangent here to highlight the strict nature of the acc_c metric, which only accredits exact predictions of each tri-valued trust critique state $c_k \in \{-1, 0, +1\}$. All of the acc_c results in this section are notably better than the chance outcome of 0.33, although admittedly OPTIMo has limitations in accurately modeling the complex dynamics of trust critiquing. Specifically, providing critiques at regular intervals imposes added mental strain on top of the human’s task supervision and intervention duties, and therefore is dependent on each individual’s communication propensity and multi-tasking capacity.

Figure 5–6 shows the effects of using histograms with different bin sizes B . Prediction performances were mediocre and unstable among small bin sizes ($B \leq 150$), and were likely due to under-sampling of the continuous trust space. For instance, the trained model in Section 5.3.2 indicated that the user’s latent trust t_k increased nominally by 0.0064 between time steps under typical operations; this would require at least $B > 156$ bins to represent faithfully. Aside from under-sampling errors, our empirical results showed that using a large number of bins

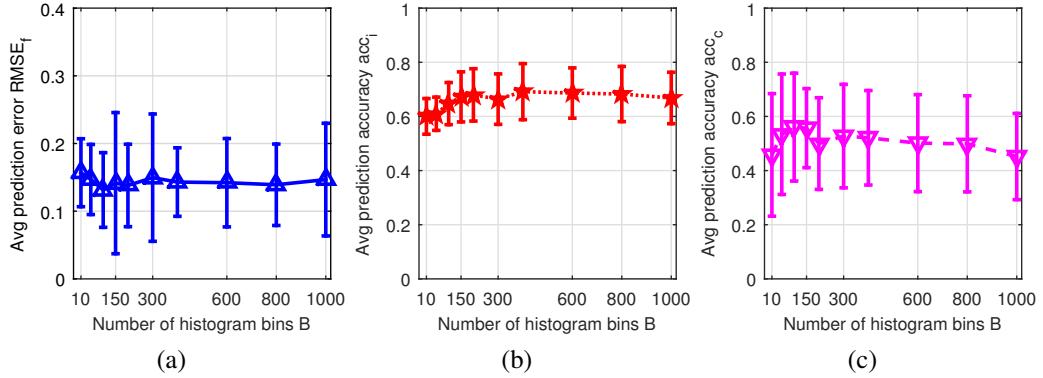


Figure 5–6: Means and standard deviations (as error bars) of OPTIMo’s prediction performances as functions of the number of histogram bins B (with $W = 3\text{ s}$, $\sigma_t = 0.005$, $\sigma_f = 0.1$).

did not lead to greater performance. We therefore conclude that a histogram with $B = 300$ bins can sensibly capture beliefs for a typical person’s trust dynamics.

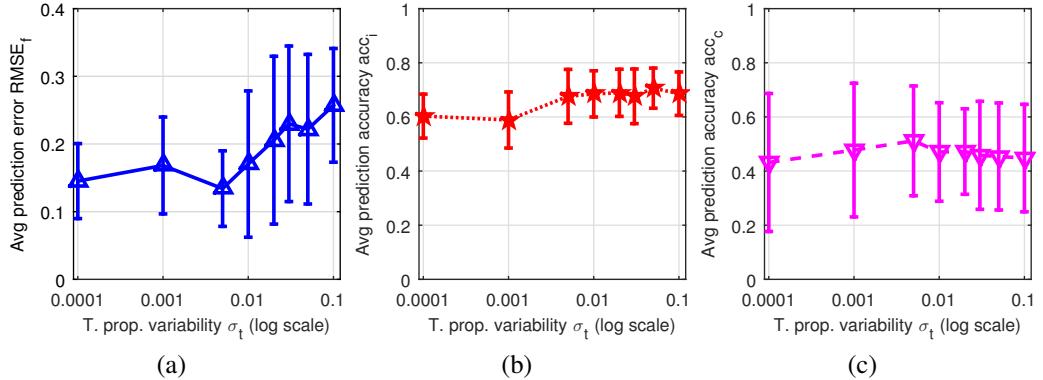


Figure 5–7: Means and standard deviations (as error bars) of OPTIMo’s prediction performances as functions of the trust propagation variability parameter σ_t (with $W = 3\text{ s}$, $B = 300$ bins, $\sigma_f = 0.1$).

It is technically possible to optimize the variability parameter σ_t along with the other trust propagation settings $\{\omega_{tb}, \omega_{tp}, \omega_{td}\}$ based on each supervisor’s recorded experience using EM. Nevertheless, we anecdotally observed that this type of evidence maximization approach tended to result in overly confident values for σ_t and ultimately poor model performance. Therefore, we opted to manually tune this parameter across all users for our boundary tracking task domain.

Figure 5–7 shows that the trust propagation variability parameter σ_t had negligible effects on the predicted accuracies of interventions and critiques acc_i, acc_c . On the other hand, OPTIMo instances with larger σ_t values exhibited worse mean and spread in their trust feedback prediction errors $RMSE_f$. This phenomenon suggested an empirical mismatch of larger sampled σ_t settings against our fast-paced boundary tracking task context, as well as the magnitudes of typical changes in consecutive trust states as illustrated in Section 5.3.2. Thus, all subsequent OPTIMo instances for boundary tracking tasks were configured with $\sigma_t = 0.005$, corresponding to the best demonstrated results.

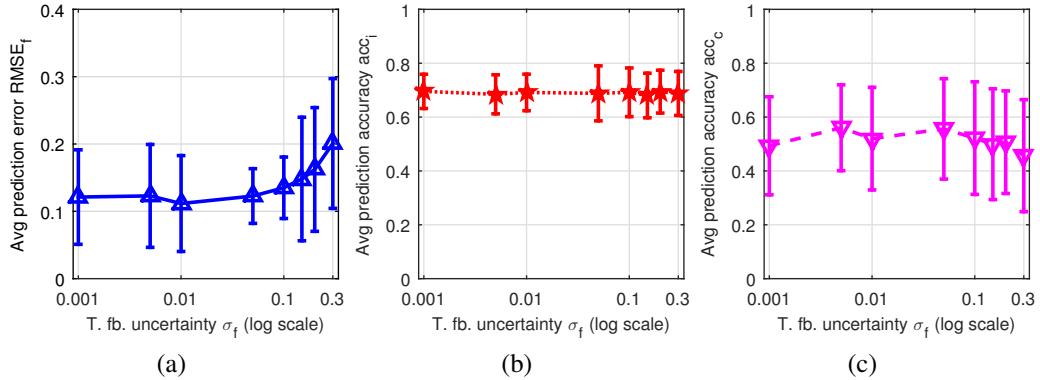


Figure 5–8: Means and standard deviations (as error bars) of OPTIMo’s prediction performances as functions of the trust feedback uncertainty parameter σ_f (with $W = 3\text{ s}$, $B = 300$ bins, $\sigma_t = 0.005$).

In general, it is difficult to estimate the amount of variability of introspected reports on an unmeasurable sentiment such as trust. Since we employed a 5-anchors format to elicit trust feedback $t_k \in [0, 1]$ (see Figure 4–9), a conservative variability estimate can be derived from the half-width between consecutive anchors, i.e. $\sigma_f = 0.1$. Figure 5–8 showed that model instances trained with different σ_f values had minimal overall effects on OPTIMo’s performance metrics, with the minor yet expected exception of worsened trust feedback *prediction errors* $RMSE_f$ for higher trust feedback *uncertainty* values σ_f . Consequently, we decided to maintain the conservative estimate of $\sigma_f = 0.1$ during subsequent model analyses.

5.4 Comparisons Among Temporal Trust Models

In addition to the above assessments, we also contrasted OPTIMo against three other temporal trust models in their ability to predict user-reported trust feedback f_k . These included the Auto-Regressive Moving Average Value (ARMAV) model for quantifying human-automation trust [56], our stepwise regression model for predicting event-induced trust changes $\Delta\mathbb{T}$ (from Section 4.2.5, [103]), and the Area Under Trust Curve (AUTC) metric [23].

5.4.1 Procedure

Model comparisons were carried out using the same dataset collected during our trust dynamics observational study, and adhered to the same 5 training session and 5 test session regime used above. We selected two OPTIMo variants for these evaluations, both configured with $B = 300$ bins, $\sigma_t = 0.005$, and $\sigma_f = 0.1$. These comprised of a fine-grained real-time variant $OPTIMo_{fine}$ with $W = 3$ s, as well as a coarse-level model instance $OPTIMo_{coarse}$ with $W = 150$ s.

The ARMAV model [56] predicts the degree of trustworthiness in an automated system as a linear function of its task performance and internal failure rates at both the latest and most recent time steps. Three variants of this first-order lag model were instantiated, including $ARMAV_{online}$, which solely considered recent agent failure rates p_k, p_{k-1} . $ARMAV_{perf}$ in contrast added an external performance metric, measured as the ratio of frames for which the target boundary was completely out of the robot’s view. This second variant is expected to be more accurate, although it requires a performance factor that is typically not available during online operations and is therefore not used by OPTIMo. Separate instances for both model variants were trained on each user’s interaction experiences, while a third, user-aggregated model form $ARMAV_{aggr}$ was regressed using both agent failure and external performance data from *all* study participants. This aggregated instance most faithfully replicated the original proposed model form.

During analyses of our event-induced trust changes study, we proposed a linear regression model similar to ARMAV for predicting session-level trust changes $\Delta \mathbb{T}$ induced by various interaction events. In addition to associating the trust state to agent failures p_k , $\Delta \mathbb{T}$ also added links to the supervisor’s intervention state i_k , as well as to the instantaneous task accuracy as measured by the robot’s distance to the designated trajectory. Since the latter metric is typically not available during online operations, we computed per-user regression variants that both ignored ($\Delta \mathbb{T}_{online}$) and included ($\Delta \mathbb{T}_{gt}$) the task accuracy factor as ground truth. Furthermore, a user-aggregated variant $\Delta \mathbb{T}_{aggr}$ regressed against interaction experiences from all study participants. To evaluate these models, predicted trust change values for the 5 test sessions were successively summed into absolute trust states, starting from the final trust feedback provided at the end of the training sessions.

During several human-robot studies conducted by Desai and Yanco [23] for assistive robotic search tasks, participants were asked to report their trust changes $c \in \{-1, 0, +1\}$ at regular intervals through gamepad button presses. The cumulative sums of these values, termed as the Area Under Trust Curve (AUTC) metric, were then used to characterize trust states at different times. We computed AUTC values at the end of each test session, and compared Pearson’s ρ correlations between user-reported trust feedback f_k and these unscaled predictions.

5.4.2 Results and Discussion

The results in Table 5–2 quantify the abilities of different models at predicting trust feedback values f_k reported by the participants of our trust dynamics study². Model prediction performances were evaluated using both the Root Mean Squared Error (RMSE) metric as well as Pearson’s ρ correlation statistics.

² OPTIMo’s performance results were updated during extended experiments following the submission of [100].

Table 5–2: Comparison of OPTIMo against three temporal trust models for predicting trust feedback f_k , assessed using Root Mean Squared Error (RMSE) across all study participants and Pearson’s ρ for the subset of users exhibiting significant correlations with their f_k values; the top two models under each evaluation metric are shaded in dark & light green respectively.

	Model Variant	avg (std) <i>RMSE</i>	% sign ρ ($\alpha = 0.10$)	avg sign ρ
<i>OPTIMo_{fine}</i>	real-time ($W = 3$ s)	0.16 (0.06) ¹	33.33%	0.88
<i>OPTIMo_{coarse}</i>	session-wide ($W = 150$ s)	0.12 (0.05) ¹	33.33%	0.90
<i>ARMAV_{online}</i>	personalized	0.32 (0.42)	9.52%	0.86
<i>ARMAV_{perf}</i>	w/ external perf. metric	0.27 (0.21)	9.52%	0.85
<i>ARMAV_{aggr}</i> [56]	user-aggregated	0.14	—	0.64
$\Delta\mathbb{T}_{\text{online}}$	personalized	0.76 (1.68)	9.52%	0.92
$\Delta\mathbb{T}_{\text{gt}}$	w/ ground truth metric	0.36 (0.32)	14.29%	0.93
$\Delta\mathbb{T}_{\text{aggr}}$	user-aggregated	0.19	—	0.73
<i>AUTC</i> [23]	personalized	(unscaled)	38.10%	0.90

At the session-wide time scale, *OPTIMo_{coarse}* produced the *most accurate* trust predictions overall. This model variant notably excelled over competing methods using similar information sources, namely *ARMAV_{online}*, $\Delta\mathbb{T}_{\text{online}}$, and *AUTC*. The real-time variant *OPTIMo_{fine}* also attained excellent prediction performance as well, and was only outperformed by the competing model *ARMAV_{aggr}*, which incorporated oracle-provided task performance data. Furthermore, predictions from both OPTIMo variants were significantly correlated to reported trust feedback values f_k for a *large portion of study participants*.

These results showed that OPTIMo variants were able to infer trust feedback values reported by our 21 study participants with *greater accuracy* and at *much finer time scales* compared to existing methods. We attribute these successes to several key features of OPTIMo, including its ability to update estimates of the latent trust state at an interaction-time scale, the accounting of the influences of multiple interaction factors on trust dynamics, and the use of a multi-modal probabilistic trust representation. In contrast, it would be highly impractical to operate regression-based models such as ARMAV and $\Delta\mathbb{T}$ at second-level time scales, since they

would require a training dataset obtained by constantly pestering users to provide trust feedback every few seconds during their interactions with the robot.

5.5 Summary

This chapter presented the Online Probabilistic Trust Inference Model (OPTIMo): a personalized, performance-centric trust model that is capable of inferring a human supervisor’s degree of trust in an autonomous robot. OPTIMo subsumes the two dominant trust modeling approaches in the literature, namely through causal reasoning of the robot’s trustworthiness given its task performance, and by using evidence from the interaction experience to support beliefs about the human’s latent trust state. Both OPTIMo’s graphical structure and its parametrized relationships were grounded on previous human-robot interaction studies, including insights from our own observational study on real-time trust dynamics.

Our empirical evaluations demonstrated OPTIMo’s success at predicting personalized trust tendencies, and also identified sensible settings for its non-personalized parameters. We further showed that OPTIMo could predict trust assessments with greater accuracies and at much finer time scales compared to existing works. Although these evaluations focused specifically on tasks within the visual navigation domain, OPTIMo’s use of generalized trust factors such as performance and intervention state allows this model to be scaled and instantiated for other task contexts as well. In summary, the high accuracy, personalized nature, and real-time inference attributes of this human-robot trust model are all useful elements contributing to our ultimate research goal of developing responsive trust-seeking robot agents.

Chapter 6

Trust-Induced Behavior Alterations

This chapter addresses the final problem in our three-stage plan toward realizing trust-seeking robots: “how should the robot agent react when the human supervisor’s trust changes?” We focus specifically on situations where the human *loses* trust in the agent, as these perceived offenses are of grave importance since they may cause the supervisor progressively to stop delegating tasks to the agent and resort to pure teleoperation. To prevent such teamwork breakdowns, the agent should react to resolve the cause of each trust loss event. In this way, it can actively work toward maintaining the efficiency of the asymmetric human-robot team.

We approach this problem by identifying different types of potential causes for trust loss and considering how to react in each case. Generally speaking, the human loses trust when the agent acts in a manner that is not to the supervisor’s likings. Such actions can be categorized into two classes:

- *misbehaviors*: the agent consistently takes *undesirable* actions, such as moving in the wrong direction; such occurrences may be due to its control logic failing (e.g. no boundary target detected) or when the agent’s task target differs from the supervisor’s;
- *unstable motions*: the agent makes an *inconsistent* sequence of actions, such as moving in an oscillatory, jittery, or otherwise incomprehensible manner; such behaviors may be caused by the presence of noise in the robot’s sensory inputs (e.g. blurry visual target).

One approach for addressing both types of situations is to elicit the human to intervene and correct the erroneous or jittery motions. Separately, the agent could also alter its own behaviors to lessen the adverse effects of these undesired behaviors.

Toward these ends, we propose the strategy of Trust-Aware Conservative Control (TACtiC), in which the robot agent reacts to trust losses by conservatively limiting changes in its actions. Altering the agent’s behaviors conservatively allows it to convey a sense of hesitation to the supervisor and implicitly prompt for intervening assistance. This strategy also attenuates unstable motions caused by noisy sensory inputs and task conditions. Given the combination of these two effects, we hypothesize that *Trust-Aware Conservative Control will contribute to greater efficiency for supervisor-worker human-robot teams.*

We begin by presenting the algorithm of Trust-Aware Conservative Control for mobile robots. The TACtiC strategy completes our trust-seeking robot framework by making use of the OPTIMo real-time trust model and complementing the interactive adaptation capabilities of APEX. We then discuss evaluations of two end-to-end instantiations of trust-seeking robots within distinct task domains, which are carried out through a large-scale interaction study as well as its extension into a set of robot field trials. These evaluations will assess the potential efficiency gains contributed by the TACtiC strategy, and synonymously, by trust-seeking robots.

6.1 Trust-Aware Conservative Control (TACtiC)

Our Trust-Aware Conservative Control strategy must address two important questions: how can mobile robots display conservative behaviors, and when to engage and disengage these temporary behavior alterations? We address the first subproblem by designing a “conservative state” of operations that alters the speed and steering commands issued by the autonomous robot agent. This conservative state post-processes control signals without modifying the agent’s internal executions, therefore enabling TACtiC to be integrated with arbitrary autonomous agents.

Focusing on the second sub-problem, we devise a trigger condition based on *salient changes* in the supervisor’s trust state, for engaging and disengaging the conservative state. This condition uses a “trust shift” signal to quantify how humans perceive recent trust assessments when making decisions. We compute personalized trust shift thresholds reflecting *salient* amounts of trust losses and gains by analyzing each user’s behavior patterns. The agent can then engage and disengage its conservative state whenever the trust shift signal exceeds these thresholds.

6.1.1 Conservative Control Alterations

Generally speaking, “conservative behavior” involves a disposition to preserve existing conditions and limit change. To help realize conservative behaviors for vehicular control specifically, we analyzed incident reports of aggressive driving by motorists [32], and identified two main categories of factors for fatal crashes:

- *exceeding regulatory limits*: speeding; tailgating; racing; illegal driving on shoulder, ditch, sidewalk, or median;
- *unexpected motions*: erratic driving and lane change; failure to signal, yield, or obey signs; sudden speed change; illegal passing.

Contrasting against these aggressive maneuvers, we implement conservative vehicular control by defining a *conservative state* of operations $\mathcal{C} \in \{\text{on}, \text{off}\}$ during which both the vehicle’s speed is reduced and the agent’s steering command is smoothed. The speed reduction gain Γ facilitates locomotion tasks by enabling longer reaction times for both the robot agent and the human supervisor, and is also useful for conveying hesitation when the agent struggles at its task (e.g. fails to detect the boundary target). Separately, we wish to enforce *predictable* behaviors by attenuating oscillatory control signals that are induced by blurry or otherwise noisy visual inputs. Toward this end, we apply an exponential filter to smooth the agent’s steering command ω_k at each time step k into an altered signal $\bar{\omega}_k$:

$$\bar{\omega}_k = a \cdot \bar{\omega}_{k-1} + (1 - a) \cdot \omega_k \quad \text{where } a = e^{-\Delta K / \tau} \quad (6.1)$$

The time constant τ introduces a temporal delay and signal smoothing, both in proportion to the command interval ΔK . Section 6.2 will investigate the significance of the Γ and τ parameters, which together affect the degree of conservative alteration being applied to the robot’s motions.

6.1.2 Trust Triggers

The conservative state \mathcal{C} is designed specifically to target situations where the human supervisor experiences salient trust losses, and should not be enabled at all times so as not to interfere with regular operations. Therefore, we devised a trigger mechanism for the conservative state that accounts for human biases in the perception of these salient changes. In particular, the “peak-end” rule is a well-studied heuristic stating that humans base their affective evaluations predominantly on past events with extreme outcomes (“peak”), as well as the most recent experience (“end”) [33]. When making decisions, people also tend to recall recent events with greater ease, as explained by the “availability heuristic” [83]. Recent experiences further have greater relevance after switching task targets during extended operations. For instance, our vision-based robot agent’s excellent coastline-tracking ability has little pertinence when it subsequently struggles at following a road.

We have anecdotally observed manifestations of these biases that favor extreme and recent events consistently throughout participant behaviors during our previous studies and field trials. Consequently, we designed TACtiC to identify *salient* changes in the supervisor’s trust state within a *sliding time window* of $K \cdot W$ seconds. The window duration is parametrized as an integer multiple of OPTIMo’s W -second update interval. Given a sequence of trust estimates within a recent window of time, we define the “trust shift” signal δt_k as the difference between the latest expected trust state $\widehat{t}_k \triangleq E[\text{bel}(t_k)]$, and either the smallest or largest recent trust value, depending on the direction of the latest trust change:

$$\delta t_k = \begin{cases} \hat{t}_k - \min \{\hat{t}_{k-K}, \dots, \hat{t}_{k-1}\}, & \hat{t}_k > \hat{t}_{k-1} \\ \hat{t}_k - \max \{\hat{t}_{k-K}, \dots, \hat{t}_{k-1}\}, & \hat{t}_k < \hat{t}_{k-1} \\ \delta t_{k-1}, & \hat{t}_k = \hat{t}_{k-1} \end{cases} \quad (6.2)$$

The duration of the memory window should be chosen appropriately to align with the rate of significant events during operations, which is highly dependent on the task domain. We analyzed typical experiences of fast-paced boundary tracking tasks from our previous observational study on trust dynamics (Section 4.3), and set a window duration of $K \cdot W = 5 \cdot 3$ seconds to match the average interval between intervention periods.

As seen in Figure 6–1, the trust shift signal reflects piecewise-shifted segments of the expected trust states that are distorted by the finite-length window. These two samples illustrate that the memory window is useful for ignoring short-term trust

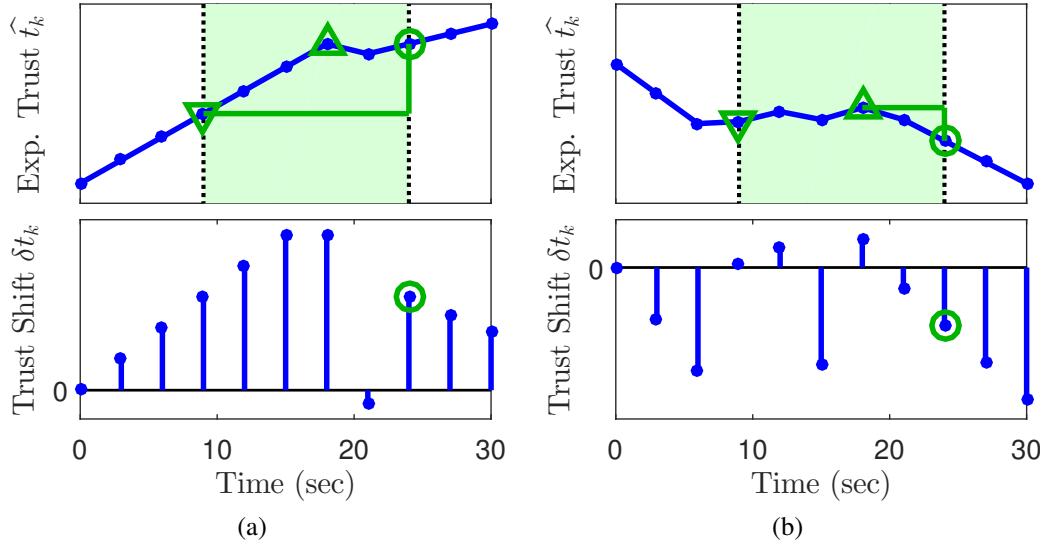


Figure 6–1: Sample sequences of expected trust states are plotted along their associated trust shift δt_k signals (using a memory window of $K \cdot W = 5 \cdot 3$ seconds). Each trust shift (e.g. green circle in bottom plots) is computed as the difference between the latest trust state (green circle in top plots) and the minimum/maximum (downward/upward triangles) within a sliding window (light green area).

perturbations while at the same time properly ignores long-outdated trust values. The trust shift is notably different from the derivative of the expected trust state because δt_k captures *cumulative* changes in recent trust evolutions rather than the *instantaneous* rate of change.

Armed with this trust shift signal, we next quantify the notion of *salient* trust losses and gains by applying threshold parameters Δt^- , Δt^+ on the trust shift signal. In other words, we engage conservative control whenever the trust shift δt_k falls below Δt^- and disengage control alterations when it surpasses Δt^+ .

These trust shift thresholds Δt^- , Δt^+ should be calibrated to each human supervisor’s unique behaviors and attitudes. Since the OPTIMo inference engine requires a training dataset from each supervisor, we conveniently reuse the collected experience to derive saliency thresholds by analyzing statistical properties of user behaviors. Specifically, we assume that the human will intervene each time the agent misbehaves or encounters a noisy task condition. We also assume that, after the agent successfully adapts to these interventions, the supervisor will then issue a trust gain critique ($c_k = +1$) as explicit approval. Consequently, we analyze each user’s behaviors from their training interaction dataset, and identify the largest negative trust shifts during each intervention period, as well as the largest positive values after intervention and before a trust gain critique. The trust loss Δt^- and trust gain Δt^+ thresholds are then computed as the *medians* of these two sets of extreme values.

A benefit of this statistical selection method is that it does not need to identify the cause of each intervention. Also, interventions due to non-trust related causes, such as when switching boundary targets, tend to have minimal impact on these thresholds. Our event-induced trust changes study in Section 4.2 substantiated this trait by showing that such causes did not lead to significant trust loss.

Algorithm 2 Trust-Aware Conservative Control (TACtiC)

Inputs: recent expected trust states $\{\hat{t}_k, \hat{t}_{k-1}, \dots, \hat{t}_{k-K}\}$; conservative state \mathcal{C}_{k-1} ; nominal speed ν_k ; agent's steering command ω_k and previous filtered command $\bar{\omega}_{k-1}$;

Parameters: speed reduction gain Γ ; smoothed steering time constant τ ; salient trust change thresholds Δt^- , Δt^+

```
1: compute trust shift  $\delta t_k$  (see Equation 6.2)
2: update conservative state  $\mathcal{C}_k \leftarrow \mathcal{C}_{k-1}$ 
3: if  $\delta t_k < \Delta t^-$  then
4:    $\mathcal{C}_k \leftarrow on$ 
5: else if  $\delta t_k > \Delta t^+$  then
6:    $\mathcal{C}_k \leftarrow off$ 
7: if  $\mathcal{C}_k = on$  then
8:   reduce speed  $\bar{\nu}_k \leftarrow \Gamma \cdot \nu$ 
9:   computed smoothed steering  $\bar{\omega}_k$  (see Equation 6.1)
10: else
11:    $\bar{\nu}_k \leftarrow \nu$ ,  $\bar{\omega}_k \leftarrow \omega_k$ 
12: return  $\bar{\nu}_k, \bar{\omega}_k, \mathcal{C}_k$ 
```

6.1.3 TACtiC Algorithm

Algorithm 2 summarizes the TACtiC strategy, which post-processes steering commands from the autonomous agent before sending them to the robot's low-level control interface. The simplicity of this strategy reflects the intuitive nature of analogous human reactions to criticisms and trust loss at the workspace, namely by eliciting actionable feedback and adapting their behaviors to amend for deficiencies. Another benefit of prompting the supervisor to help out the robot agent is the capacity to cope with arbitrary misbehaviors without requiring separate detectors for different types of failures or noisy task conditions. Finally, it is important to acknowledge such trust-induced reactions are contingent on the ability to infer the human's evolving trust states in real time, and therefore is only enabled by our recent advances in online human-robot trust modeling with OPTIMo.

6.2 TACtiC User Study with Simulated UAV

We developed two end-to-end implementations of trust-seeking robots, each including an APEX-enabled adaptive boundary tracking agent, an OPTIMo real-time trust inference engine, as well as a TACtiC command alteration module. To assess the potential efficiency gains of these robot agents, we conducted a large-scale interaction study in which 46 users teamed up with our agent to carry out aerial coverage tasks. As an extension to this study, we also conducted field trials with 12 participants to assess the efficiency of their collaboration with a trust-seeking self-driving car. This field trial extension will be discussed in Section 6.3.

The goal of this user study is to compare the relative team efficiency among TACtiC-enabled agents and also against a baseline agent without any trust-induced behaviors. These robot agents were instantiated as follows:

- *Strongly Conservative* (A): APEX-enabled agent exhibiting an exaggerated degree of trust-aware conservative behaviors ($\Gamma = 60\%$, $\tau = 1.0$ sec);
- *Mildly Conservative* (B): APEX-enabled agent exhibiting a mild degree of trust-aware conservative behaviors ($\Gamma = 80\%$, $\tau = 0.5$ sec);
- *Baseline* (C): plain APEX-enabled agent without running the OPTIMo inference engine and the TACtiC strategy.

The conservative motion parameters Γ , τ for the two trust-seeking agents were adjusted empirically to exhibit sensible degrees of conservative motions for fast-paced boundary tracking tasks. We use the “strongly” and “mildly” conservative labels to succinctly refer to these agents, and thus the reader should restrain from over-interpreting these qualifiers.

All three agents were equipped with the APEX algorithm and thus the capacity to interactively adapt their boundary tracking performances. By using the plain APEX agent as the comparative baseline, this study builds upon our previous findings in Chapter 3, which demonstrated that such adaptive agents achieved greater

efficiency as compared to expert-tuned non-adaptive agents as well as plain teleoperation. Interactive adaptation is also helpful in general-purpose human-robot team deployments (such as the scenarios in this study) since task targets do not need to be specified before deployment and can even be switched dynamically.

The TACtIC strategy has two complementary goals: eliciting the supervisor to participate, and mitigating adverse effects of undesirable motions. These goals are aligned with diverse aspects of team efficiency such as improving the agent’s perceived active collaborative efforts and attenuating drops in task performance. To form a thorough evaluation of trust-seeking robots, both this interaction study and its field trial extension seek to validate the following hypotheses:

Hypothesis 1 *The mildly conservative trust-aware agent (B) yields greater **objective** efficiency ranking, quantified by performance and workload, compared to strongly conservative (A) and non-conservative (C) agents.*

Hypothesis 2 *The mildly conservative trust-aware agent (B) yields greater **subjective** efficiency ranking, quantified by perceived collaboration effort and trust, compared to strongly conservative (A) and non-conservative (C) agents.*

6.2.1 Participants

We recruited 46 users (13 females) of varied age ($\mu = 28$, $\sigma = 6$) from the School of Computer Science at McGill University to participate in this study. Users had diversified levels of prior robot experience, and consisted of 8 undergraduate students, 32 graduate students studying robotics, 4 professors involved in robotics research, and 2 robot engineers.

Similar to our previous interaction studies, we targeted users with technical backgrounds and high expected propensity toward robots. These participants exhibited similar degrees of general proficiency and attitudes, such as moderate reported levels of comfort with their driving skills ($\mu = 62\%$, $\sigma = 27\%$ degree of agreement with “competent driver”) and propensity toward car automation ($\mu = 54\%$,

$\sigma = 29\%$ “comfort using cruise control”). Nevertheless, compared to the expert roboticists who took part in our real-time trust dynamics study (see Section 4.3), this study population has markedly less familiarity with robots ($\mu = 3$, $\sigma = 3$ years programming experience), using a gamepad ($\mu = 54\%$, $\sigma = 30\%$ “competence”), radio-controlled vehicles ($\mu = 30\%$, $\sigma = 26\%$ “competence”), and visual-feedback teleoperation ($\mu = 32\%$, $\sigma = 30\%$ “competence”). We specifically sought out a broader audience in order to thoroughly assess the *practicality* of our trust-seeking robot agents in addition to its efficiency gains. This expansive study population also enabled us to evaluate the degree of generalizability of our previously-shown trust prediction accuracies for OPTIMo instances.

6.2.2 Infrastructure

This study aims to evaluate trust-seeking robots within idealized interaction scenarios that are isolated from environmental disturbances, dynamic conditions, and practical robot deployment concerns. We used our trust-seeking agents to control simulated drones through SightedTurtleSim, to enforce repeatable and controlled conditions. Despite relying on idealized vehicular dynamics, we designed realistic task scenarios using scenes synthesized from real-world satellite imagery.

Figure 6–2 shows the visual interface for this interaction study, which is nearly identical to our real-time trust dynamics study. The live camera view from the robot is overlaid with the agent’s steering commands as well as occasional manual steering from the supervisor. These commands are depicted as blue and green arrows respectively. The user is also prompted at regular intervals by a $\pm ?$ icon to provide critiques reflecting changes in their trust state.

The top-center text overlay describes the current task target, which is updated to reflect task switches as the robot travels past pre-determined locations. We also used a synthetic speech engine to read updated instructions out loud to ensure that participants are aware of these task switches.

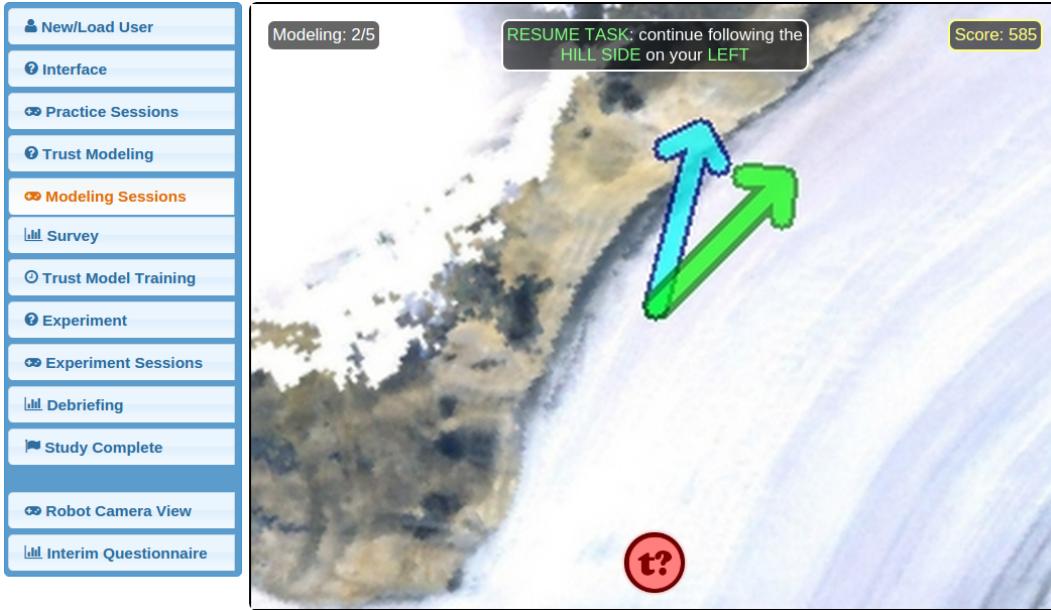


Figure 6–2: Visual interface for the TACtIC user study depicted various stage of the study (left) as well as the aerial drone’s live camera feed (right). The camera view was overlaid with the agent’s steering command (blue arrow) and the human supervisor’s interventions (green arrow). Additional overlays provided information on session progress, current task goal, terrain coverage score, and further prompted for trust change critiques ($t?$).

The top-right score overlay reflects the boundary coverage progress during each session. This score is computed based on hand-annotated ground truth flight trajectories. During interventions, the score is incremented at a one-tenth rate and is colored differently, so as to inform and incentivize the user to return control back to the agent whenever appropriate.

The control interface in this study consists of a dual joystick gamepad with an identical mapping from our previous configurations (see Figure 4–8). The user can intervene at any time by pushing and holding the left analog stick in the desired steering direction, while the autonomous robot agent assumes control by default when the analog stick is not engaged. Trust change critiques can also be freely issued by pressing the corresponding $t+$, $t=$, and $t-$ buttons. In light of feedback following our trust dynamics study concerning erroneous trust critique button

presses, we explicitly instruct participants to press the $t=$ button by default, as well as to be conscientious when issuing positive and negative critiques.

Interaction scenarios in this study were divided into short sessions lasting 1-2 minutes each. These scenarios were revised from previous studies, and included terrains with varying degrees of tracking difficulty. As depicted in Figure 3–2, such terrains comprised of smooth highways, blurry snow-covered hillsides, and curvy coastlines. The robot’s speed and altitude settings were empirically tuned to ensure ample-paced task conditions and a limited field of view. This design aimed to motivate users to delegate the steering task to the autonomous agent whenever possible, and especially when tracking rapidly-changing boundary targets.

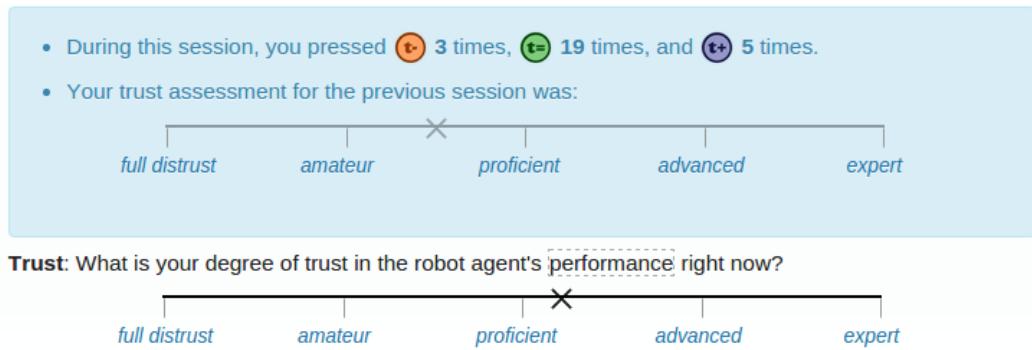


Figure 6–3: Interim trust feedback questionnaire for the TACtIC user study incorporated the user’s last trust assessment from the previous session as well as counts for trust change critiques issued during current session.

During pauses between study sessions, the user is asked to provide feedback about their current trust state. As seen in Figure 6–3, this questionnaire used the same modified Visual Analog Scale (VAS) [74] variant with five anchors from previous studies, which was designed to facilitate repeated trust feedback. We also displayed the trust response from the last study session as well as trust change critique press counts made during the current session. These elements were added to remind the participant of their past behaviors and attitudes and help them provide consistent incremental trust assessments.

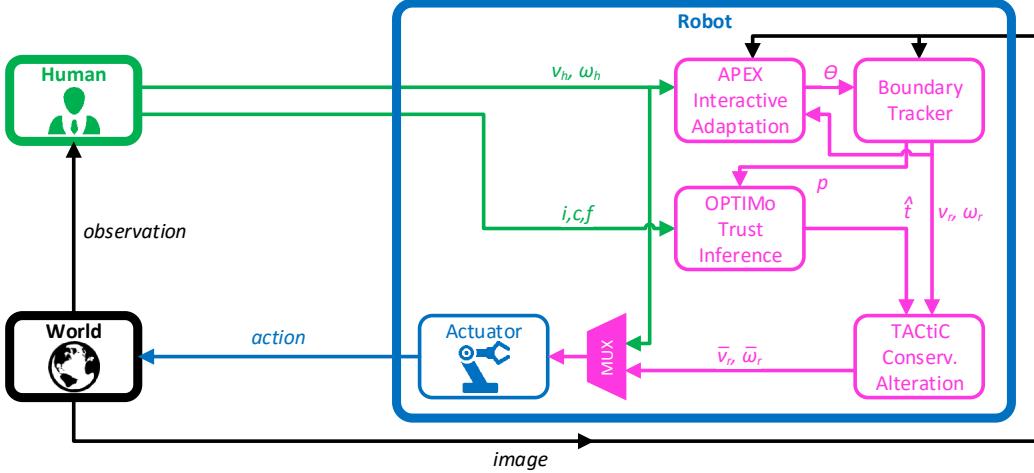


Figure 6–4: The final form of the trust-seeking robot framework entails multiple modules (mauve-colored) running onboard the mobile robot. The boundary tracking base agent processes images from the robot’s camera to produce steering commands ω_r under a nominal velocity $\nu_r \triangleq \nu_h$. The APEX adaptation module continuously optimizes the tracker’s parameters Θ by comparing its outputs ω_r against the human supervisor’s intervening steering ω_h . The OPTIMo engine unifies the agent’s performance estimates with intervention states as well as trust critiques and feedback from the human, to produce moment-to-moment beliefs over the trust state t . The TACtiC unit alters the agent’s commands ν_r, ω_r during periods of salient trust loss; these altered commands $\bar{\nu}_r, \bar{\omega}_r$ are issued to the robot’s actuator interface only when not overruled by the supervisor, i.e. $\omega_h = \emptyset$.

All elements of our trust-seeking agents along with a fully automated study infrastructure were implemented as inter-connected processes built on top of the Robot Operating System (ROS) middleware. The final form of our robot agent is shown in Figure 6–4, and includes the boundary tracking base agent, the APEX interaction parameter adaptation module, the OPTIMo real-time trust inference engine, and the TACtiC trust-induced conservative alteration unit. These blocks were developed using a combination of programming languages including C++, MATLAB, and Python, while the user interface was built using HTML and Javascript. Our development of trust-seeking robot agents amounted to over 53,000 lines of code that were deployed on various mobile and embedded computing architectures.

The adaptive boundary tracking agent was configured to process camera frames at 10 Hz with a downscaled resolution of 160×120 pixels while using optimized APEX learning rates of $\alpha = 0.2, \gamma = 0.7$. The TACtiC algorithm processed trust beliefs inferred by OPTIMo at a $W = 3$ second update interval, and used a memory window of $K \cdot W = 5 \cdot 3$ seconds.

6.2.3 Procedure

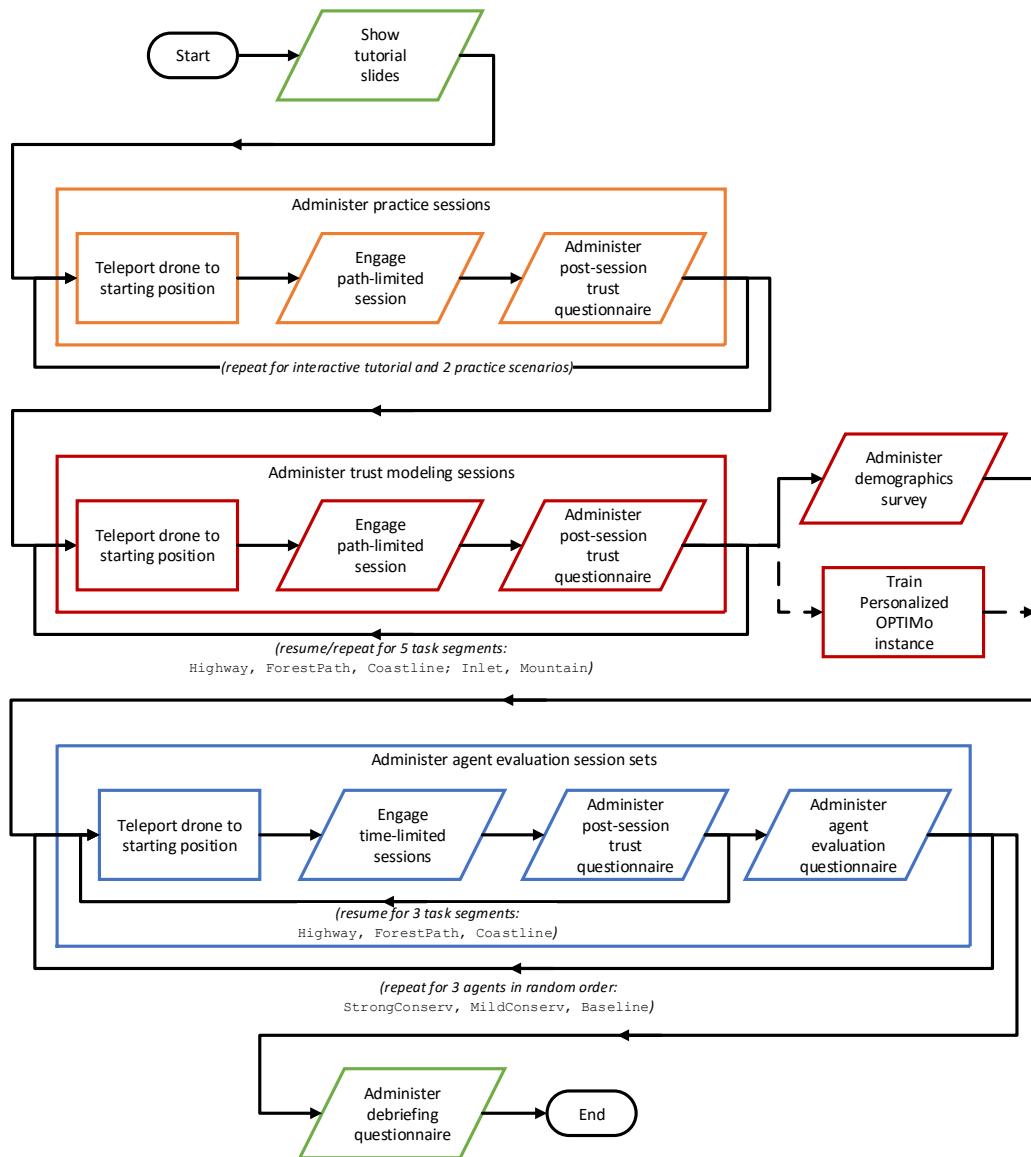


Figure 6–5: Flowchart for the TACtiC user study.

As shown in Figure 6–5, this study is separated into three phases: coaching users, collecting interaction data to build a personalized trust model, and evaluating the three robot agents. The study begins with an initial set of slides explaining the task, user interface, study flow, and our performance-centric trust definition. Users are also told that only some of the robot agents in this study will react to changes in their trust state. Nevertheless, to avoid biasing participants, throughout the study the three agents are labeled anonymously as A, B, and C. Also, the display interface is identical for each agent type, and no explicit feedback is provided when a trust-seeking agent engages or disengages its conservative state.

Following the initial slides, the user is led through an interactive tutorial and two practice sessions to familiarize with the boundary tracking agents. One practice session purposefully features an ambiguous forest-tracking task in which the robot repeatedly veers off due to narrow branching tree-lines. This lets users practice critiquing and intervening the agent, as well as calibrate their trust expectations.

The second study phase consists of 5 trust modeling sessions, each ending with a trust feedback questionnaire. At the end of the 5 sessions, the user is asked to answer a demographics survey, while their personalized OPTIMo instance is trained in the background. To ensure that the user is not forced to wait excessively after completing the survey, we tuned OPTIMo’s EM convergence and restart settings to strike a balance between accuracy and process time. Once OPTIMo’s parameters were trained, the study interface also provided quantitative interpretations of the user’s modeled behaviors (e.g. intervention likelihood, trust propagation) as interesting insights. These interpretations were computed following the procedure previously outlined in Section 5.3.2.

During the final study phase, the user is asked to partner up with each of the three robot agents in turn and collaborate through 3 interaction sessions. These sessions are identical across agents and are slightly modified versions of previous

task scenarios, while the ordering of the agents is randomized following a counterbalanced repeated-subjects design. After each agent interaction block, the user is prompted to assess the agent’s active collaborative efforts using the Visual Analog Scale (VAS) answer format [74]. Also, the study interface explicitly instructs participants to treat each new agent independently after interacting with a previous one. Both the modeling and evaluation phases entail 20 minutes of interactions each, and the entire study has a nominal length of 60 minutes.

We summarize the recorded interaction experiences using several metrics reflecting objective and subjective aspects of team efficiency. Task performance is captured by the area of coverage around each session’s designated trajectory. This coverage metric is similar to the score displayed during the study, although coverage increments are not penalized during intervention periods. Separately, the amount of active user workload is quantified as the frequency of manual interventions during each session. These measures of objective efficiency complement the supervisor’s perceptions toward the agent, which are comprised of the post-session trust feedback values and the post-interaction assessment of the agent’s collaborative efforts.

6.2.4 Results and Discussion

We now analyze the study dataset to seek to validate our hypotheses on the potential efficiency gains of trust-seeking agents. These analyses include a sample illustration of typical interaction experiences, an assessment of the predictive performance for the trained human-robot trust models, a comparison of the different trust-seeking agents under each efficiency metric individually, and finally aggregate agent rankings on their overall objective and subjective team efficiency.

Typical Interaction Experience

Figure 6–6 illustrates the experience from a sample participant’s interactions with the mild trust-seeking robot agent (B). This diagram also vertically illustrates the procedural flow of the TACtiC strategy, from integrating the user’s behavior

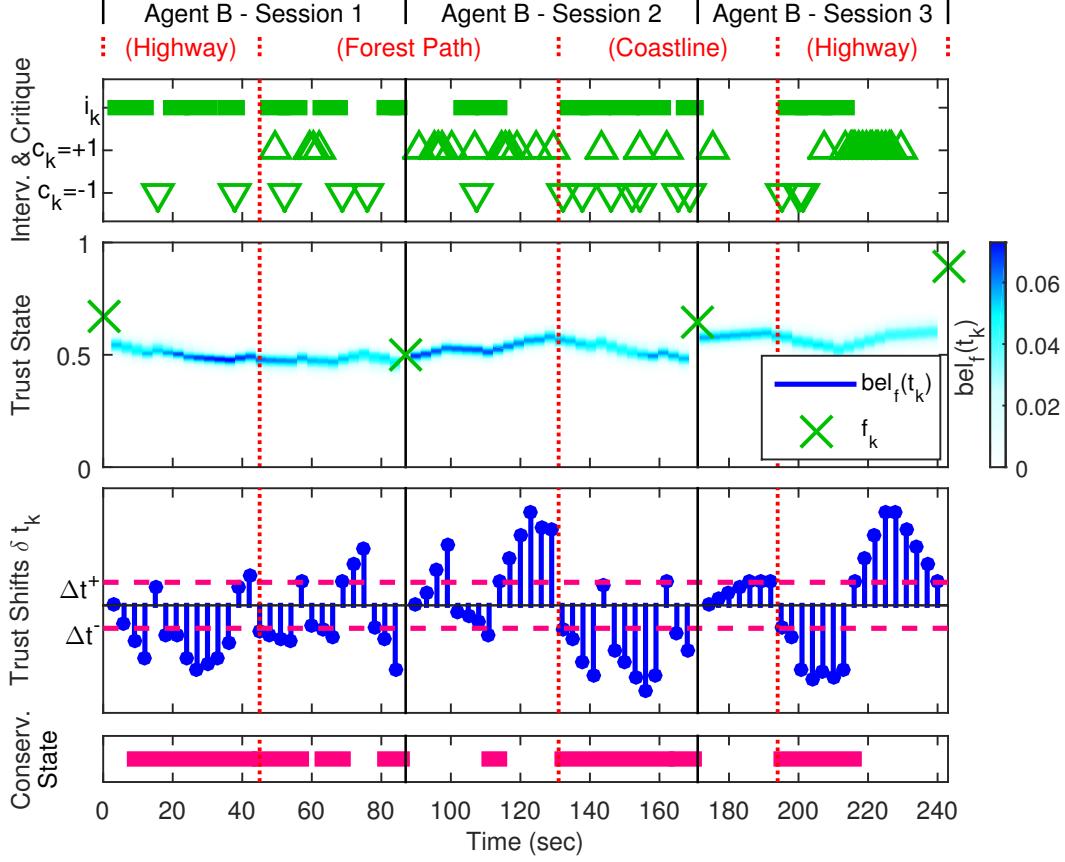


Figure 6–6: Sample interaction dataset with the mildly conservative trust-aware agent (B) from the TACtIC user study: interventions i_k and trust change critiques c_k are integrated into beliefs of the trust state $bel_f(t_k)$; the trust shift signal δt_k is computed from the expected trust state at each time step k ; the agent engages and disengages the conservative state as the user’s trust shift surpasses the trust loss Δt^- and trust gain Δt^+ thresholds (purple dashed lines).

and assessment factors, to updating the trust belief, then computing the trust shift signal, and finally updating the conservative state based on salient trust shifts.

Aside from addressing a few reactions to poor agent tracking behaviors, the majority of interventions and critiques from this supervisor resulted from training the agent to follow each of the three boundary targets sequentially. Since the boundary segments had varying degrees of task difficulty, each segment required different amount of interventions, notably with the curvy coastline (133 – 195 seconds)

requiring the highest frequency of interventions. The evolution of trust beliefs inferred by the OPTIMo engine consistently matched these success and failure events, and the trust-seeking agent further reacted to these extreme events accordingly by engaging its conservative state.

Accuracy of Trained Trust Models

The personalized OPTIMo instances in this study resulted in few training errors, as seen in Table 6–1’s small Root Mean Squared Error $RMSE_f$ for predicting trust feedback, and the high accuracies acc_i, acc_c for predicting intervention states and trust change critiques. These trained models also attained comparable levels of performance during the post-training agent evaluation phase.

Table 6–1: Across-users prediction performance of trained OPTIMo instances during the TACtIC user study; results are shown separately for training-set and test-set performance, which both reflected great modeling performance, and were comparable to previous empirical assessments in Section 5.3.

	Trust Feedback $RMSE_f$	Intervention acc_i	Trust Critique acc_c
Model Training Phase	0.08 ($\sigma = 0.04$)	78% ($\sigma = 9\%$)	63% ($\sigma = 11\%$)
Agent Evaluation Phase	0.14 ($\sigma = 0.07$)	73% ($\sigma = 14\%$)	58% ($\sigma = 14\%$)

The test-set trust prediction results in this section were quite similar to OPTIMo’s predictive performance from our original model assessments in Section 5.3. Notably, the previous evaluations were based on interaction data from *expert roboticists* in our trust dynamics study (Section 4.3), while results in this section were assessed against a *much broader study population*. Therefore, these results suggest that OPTIMo has adequate expressiveness to accurately model trust-related behaviors and attitudes of a wide range of individuals beyond expert roboticists.

Per-Metric Team Efficiency

Figure 6–7 presents baseline comparison results for the strongly conservative and mildly conservative trust-seeking agents (A and B), assessed against the individual efficiency aspects of performance, workload, perceived collaboration, and

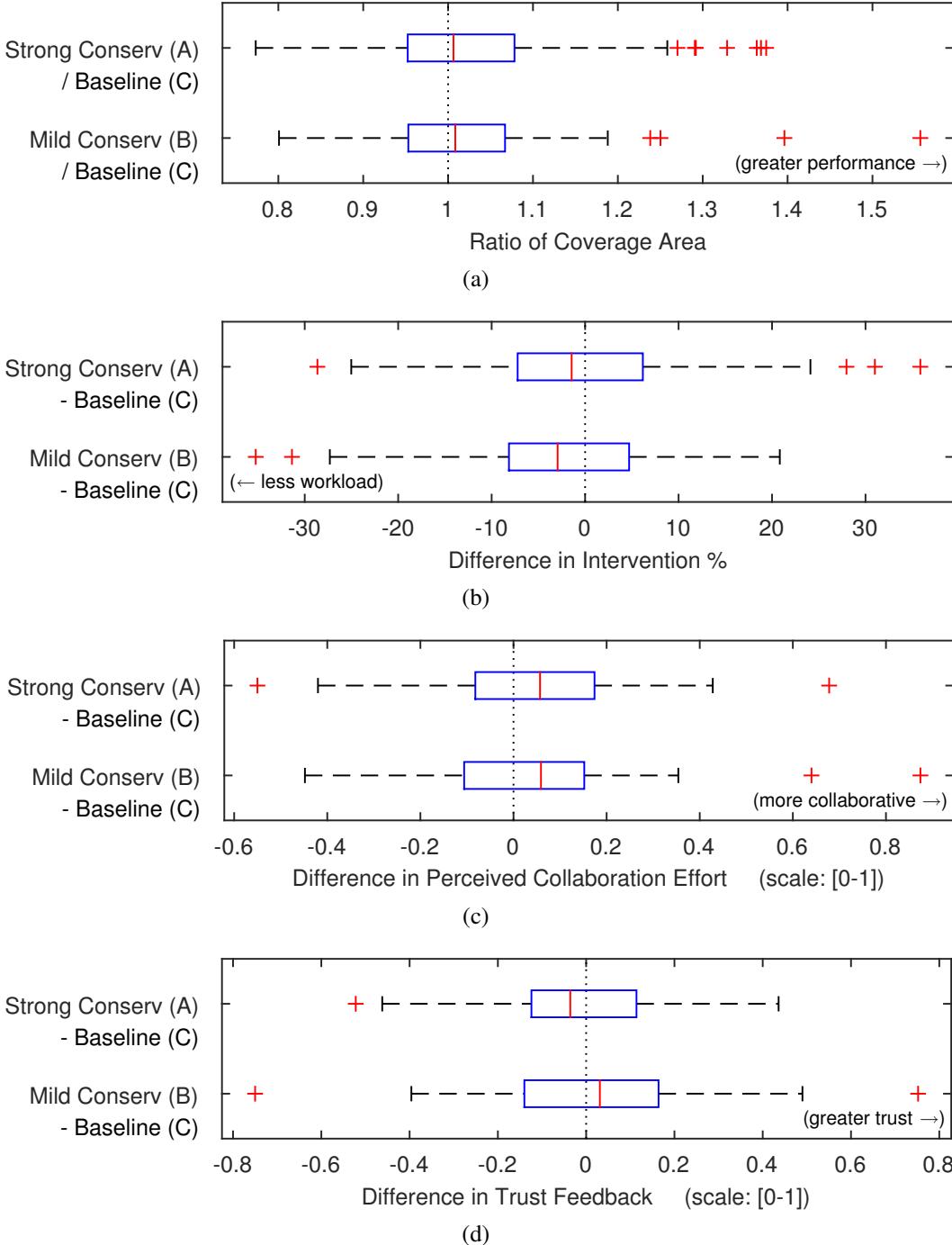


Figure 6–7: Per-metric efficiency comparisons from the TACtIC user study are shown as box plots comparing both trust-seeking agents (A and B) separately against the baseline agent (C). The vertical dashed line in each plot correspond to the null hypothesis representing the lack of difference between the trust-seeking and baseline agents. These plots revealed that both trust-seeking agents attained greater task performance, less active workload, and were perceived as more collaborative. The mildly conservative agent (B) also earned greater trust than the baseline (C), while the strongly conservative agent (A) was the least trusted.

trust. Averaged across study participants, overall both trust-seeking agents attained greater boundary task coverage and required less active workload from the supervisor when compared to the plain APEX-enabled agent (C). Furthermore, the majority of users perceived both trust-seeking agents to be more collaborative. Finally, the mildly conservative agent (B) garnered higher trust assessments than the baseline (C), whereas the strongly conservative agent (A) was the least trusted overall.

The performance and workload results suggest that both TACtiC-enabled agents improved the objective aspects of team efficiency. In particular, we hypothesize that the reduced amounts of cumulative intervention may have resulted because the TACtiC strategy prompted users to intervene swiftly, which made it easier to resolve the robot agent’s misbehaviors.

On the other hand, subjective efficiency metrics showed a discrepancy in preference for the strongly conservative agent (A), which was perceived to be more collaborative yet less trusted than the baseline agent (C). Upon closer analysis of the post-study feedback, we found that many participants acknowledged agent A’s active efforts toward maintaining teamwork, but at the same time were frustrated by its extremely slow speed and delayed reactions. Nevertheless, the mildly conservative trust-seeking agent (B) attained greater perceptions for both its collaborative efforts and trustworthiness, therefore suggesting that its conservative state parameters were more appropriate in this interaction context.

The per-metric comparisons did not reveal any significant differences in magnitudes among the agents. We justify these numerical similarities by noting that the metrics were computed *cumulatively* across extended periods of interactions, whereas the conservative state in trust-seeking agents was by design only meant to be triggered during *few and short occurrences of notable trust loss*. Also, all three agents in this study, including the baseline, were equipped with interactive

adaptation capabilities. These capabilities have been previously shown in Chapter 3 to yield significantly greater efficiency, in comparison to both non-adaptive expert-tuned agents as well as to plain teleoperation.

Aggregate Team Efficiency

In addition to per-metric assessments, we also investigated whether any of the agents had received notably greater preference when considering their aggregate contributions toward the objective and subjective facets of team efficiency. Looking at Figure 6–8, the mildly conservative trust-aware agent (B) was ranked highest in terms of both aggregate efficiency aspects. Both mean agent orderings were corroborated by identical across-users aggregate rankings computed using the Kemeny-Young method [110].

Friedman test [35] on the objective efficiency rankings of performance and workload revealed weakly significant differences among the agents ($\chi^2(2) = 5.609$, $p \leq 0.10$). In contrast, a similar analysis found evidence of strong agent preference among subjective efficiency rankings of perceived collaboration and trust ($\chi^2(2) = 11.783$, $p \leq 0.01$). Post hoc Nemenyi testing [22] at $\alpha = 0.05$ found agent B to be consistently preferred among users’ subjective assessments.

Summarizing the above analyses, we have empirically demonstrated that the Trust-Aware Conservative Control strategy, when suitably tuned for the task domain, contributed to both superior objective and subjective facets of efficiency within asymmetric human-robot teams. In particular, the mildly conservative trust-seeking agent (B) was consistently perceived as more collaborative and trustworthy among robotics researchers and enthusiasts alike. Consequently, these results have substantiated both of our Hypotheses 1 and 2.

Additionally, the corroborations of these hypotheses were contingent on several precursor conditions, including proper user coaching, accurate trust modeling,

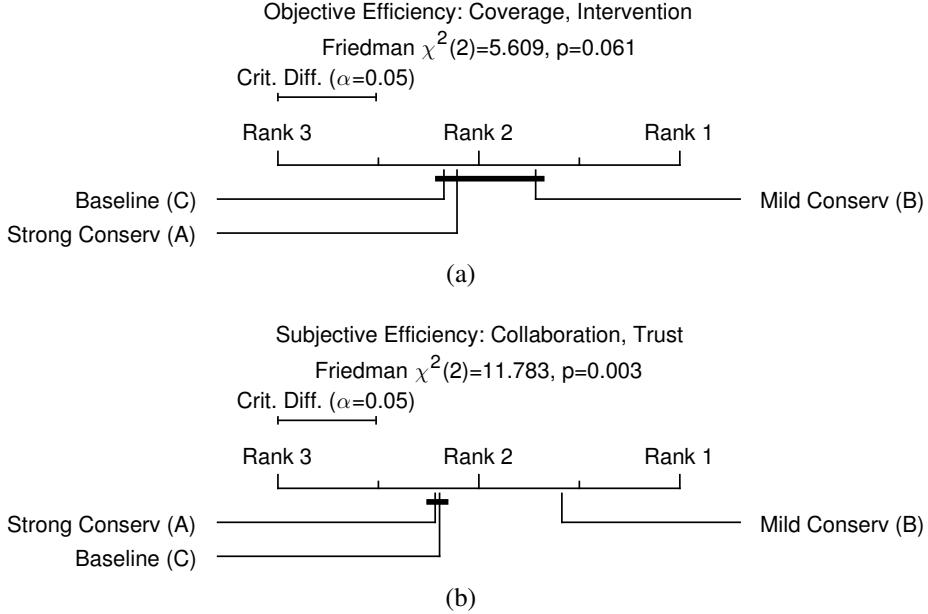


Figure 6–8: Mean agent rankings across users from the TACtiC user study revealed that the mildly conservative trust-aware agent (B) was ranked highest for both objective and subjective efficiency facets. Agent B was also *consistently* preferred among users, as reflected by the presence of a critical difference during post hoc Nemenyi analysis (i.e. the *lack* of a thick horizontal line straddling agent B in (b)).

and suitable selection of trust triggers. These conditions were all met within the study’s tight duration, thus reflecting the rigor in our study design process.

Furthermore, after the study, several users reported delight in perceiving the trust-seeking agents as having great foresight. Many cited specifically that these agents smoothly tracked the curvy coastline without wastefully turning in and out of every little inlet. These anecdotes demonstrated that TACtiC was not only beneficial toward swiftly prompting the supervisor’s interventions, but its command filtering feature also contributed both to improved task performance and superior user satisfaction, even for challenging task scenarios.

6.3 TACtiC Field Trial Extension with SL-Commander Vehicle

The previous user study showed that the TACtiC strategy contributed consistently to team efficiency gains within controlled and idealized scenarios. We extended this interaction study through a set of field trials with the SL-Commander vehicle, in which 12 of our study participants partnered up with boundary tracking

agents to navigate through a challenging test course at the Canadian Space Agency. These field trials aimed to obtain further empirical evidence supporting the efficiency gains of trust-seeking agents, and especially within *real-world conditions*. We specifically sought to re-affirm Hypotheses 1 and 2 by contrasting the mildly conservative trust-aware agent (B) against the plain APEX agent (C).

Several logistic challenges emerged during the field trial planning, including tight test schedules, limited availability of the vehicle, and access restrictions to the test site. In light of these challenges, we designed these field trials as an *extension* of the previous user study, rather than a standalone experiment. This decision enabled us to make incremental adjustments to the infrastructure toward streamlining the development process, and to optimize the length of each trial run by re-using the trained OPTIMo instance from the participant’s interaction study dataset.

6.3.1 Participants

We designed these field trials to accommodate all participants from the previous study. Nevertheless, only 12 users (1 female) were able to meet the tight scheduling and security requirements for accessing the test site at the Canadian Space Agency. All of these participants were actively engaged in robotics research, and comprised of 8 graduate students, 2 professors, and 2 robot engineers.

6.3.2 Infrastructure

Only minor adjustments were made to the boundary tracking agent and the fully-automated infrastructure of our TACtiC user study. We replaced the planar-view aerial boundary tracker with the frontal-view variant suitable for the SL-Commander, while using identical settings for the APEX adaptation algorithm, the OPTIMo inference engine, and the TACtiC control alteration strategy. The SL-Commander’s front-hood camera was hand-tuned to a static pose of 8° right-side pan and 20° downward tilt. This configuration ensured proper visibility of the road-side boundaries within the test course.



Figure 6–9: During the TACtiC field trial extension, the participant sat in the SL-Commander’s passenger seat and interacted with boundary tracking agents using the same gamepad interface from the TACtiC user study, while a tablet below the windshield displayed the agent’s overlaid camera visualizations and post-session trust feedback questionnaire.

As shown in Figure 6–9, the participant sat in the passenger seat while supervising the boundary tracking agents. A familiar visual interface featuring the live camera feed with overlaid steering commands was displayed on a tablet placed below the windshield on the passenger’s side. In order to accommodate participants who preferred to keep their eyes on the road at all times, an audio cue was used to prompt for trust change critiques every 5 seconds.

The interface for issuing interventions and critiques remained unchanged and used an identical gamepad interface. Additionally, users could press two shoulder buttons to increase or decrease the car’s nominal speed by ± 1 km/h. The speed was set to 0 km/h at the beginning of each session, and also had a maximum setting of 20 km/h. During preliminary testing, we found that speed changes consistently correlated with changes in trust, where one would decrease speed in response to abrupt motions and increase speed after appropriate training. Therefore, we configured speed change commands also to produce positive and negative trust critiques.

These field trials employed the same 1 km gravel circuit at the Canadian Space Agency site that was used previously during the APEX interactive driving field demonstrations. Recalling from Figure 3–12, the target gravel road was surrounded by assorted hurdles, including narrow pathways, road intersections, bordering watery ditch, and nearby parked cars and sheds. These obstacles presented constant challenges with varying degrees of difficulties in the boundary tracking task, and also regularly impacted the comfort of each participant.

6.3.3 Procedure

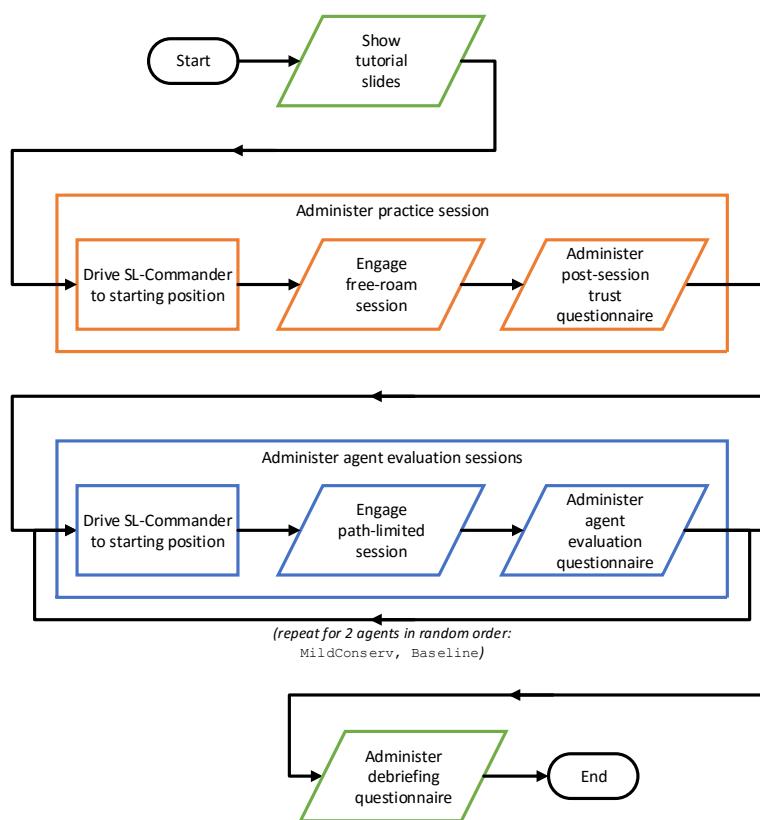


Figure 6–10: Flowchart for the TACtIC field trial extension.

As shown in Figure 6–10, each trial run was comprised of three interaction sessions along the 1 km test course. In the first session, the participant practiced collaborating with a plain APEX-enabled agent to familiarize with the interface and course layout. The following two sessions featured the mild trust-induced

conservative agent (B) and the baseline agent (C) in a randomized order. During these evaluation sessions, participants were instructed to proceed through the course quickly while operating within their comfort levels. We also explicitly encouraged users to be mindful of the robot agent’s performance and to consider delegating the steering task to the agent when appropriate. The simplicity and brevity of these field sessions were predominantly due to the severe restrictions on the deployment schedules, vehicle, and test site.

Interaction experiences collected during trial runs were evaluated using a near-identical set of metrics reflecting objective and subjective aspects of team efficiency. More specifically, the unchanged efficiency metrics included the intervention frequency as well as post-session trust feedback and perception of each agent’s collaborative efforts. As the sole exception, task performance was quantified by the elapsed duration of each session while traveling through the fixed-length course. This metric was chosen for practicality purposes within a field deployment setting.

6.3.4 Results and Discussion

We now present several empirical analyses based on the recorded dataset from the TACtIC field trial. These include an assessment of OPTIMo’s performance at predicting the supervisor’s behaviors and attitudes while collaborating with the SL-Commander robot, per-metric efficiency results, and aggregate efficiency rankings between the mildly conservative agent (B) and the baseline agent (C).

Accuracies of Trust Models

We evaluated the performance of our OPTIMo instances, which were trained previously for each participant of the aerial coverage TACtIC user study, on interaction factors within the different task context of interactive autonomous driving in these field trials. The across-users MLE prediction accuracies were as follows: $RMSE_f = 0.21 (\sigma = 0.18)$, $acc_i = 69\% (\sigma = 5\%)$, $acc_c = 50\% (\sigma = 15\%)$.

These results were generally comparable to the test-set performance values from our user study, albeit with slightly increased error for predicting trust feedback.

We suspect that the discrepancy in trust feedback prediction performance may be related to the context change from remotely operating a simulated drone to steering a vehicle from its inside. Due to the physical and material risks, we believe that participants likely had adopted cautious and sensitive attitudes when interacting with our trust-seeking agent within this custom research vehicle, which was known to be valued at over 1,000,000 dollars US. This theory is consistent with anecdotal evidence that many users during the initial practice session were visibly stressed and operated the vehicle during most of the course at low speeds of 2 – 6 km/h. All participants were receptive to gentle encouragements during the practice session to gradually increase the vehicle speed, although only 4 out of 12 users were comfortable enough to operate the SL-Commander at the 20 km/h maximum velocity during the agent evaluation sessions that followed.

Per-Metric Team Efficiency

Per-metric efficiency results for the field trials are shown in Figure 6–11. The trust-seeking agent (B) contributed on average to faster session completion times and was perceived as both more collaborative and trusted than the baseline agent (C). Nevertheless, the amount of active workload expanded to supervise both agents were comparable. Additionally, none of the per-metric results revealed any significant differences between the two agents’ efficiency attributes.

We attribute the comparable intervention frequencies among robot agents to the frequent occurrence of obstacles and hurdles within the short test circuit, such as narrow pathways as well as nearby trees and buildings. We suspect that the impacts of these obstacles were amplified by the observed cautious attitudes of our participants. Also, the lack of significant changes in the *magnitudes* for the individual efficiency metrics was likely due to both the trust-seeking agent and baseline agent

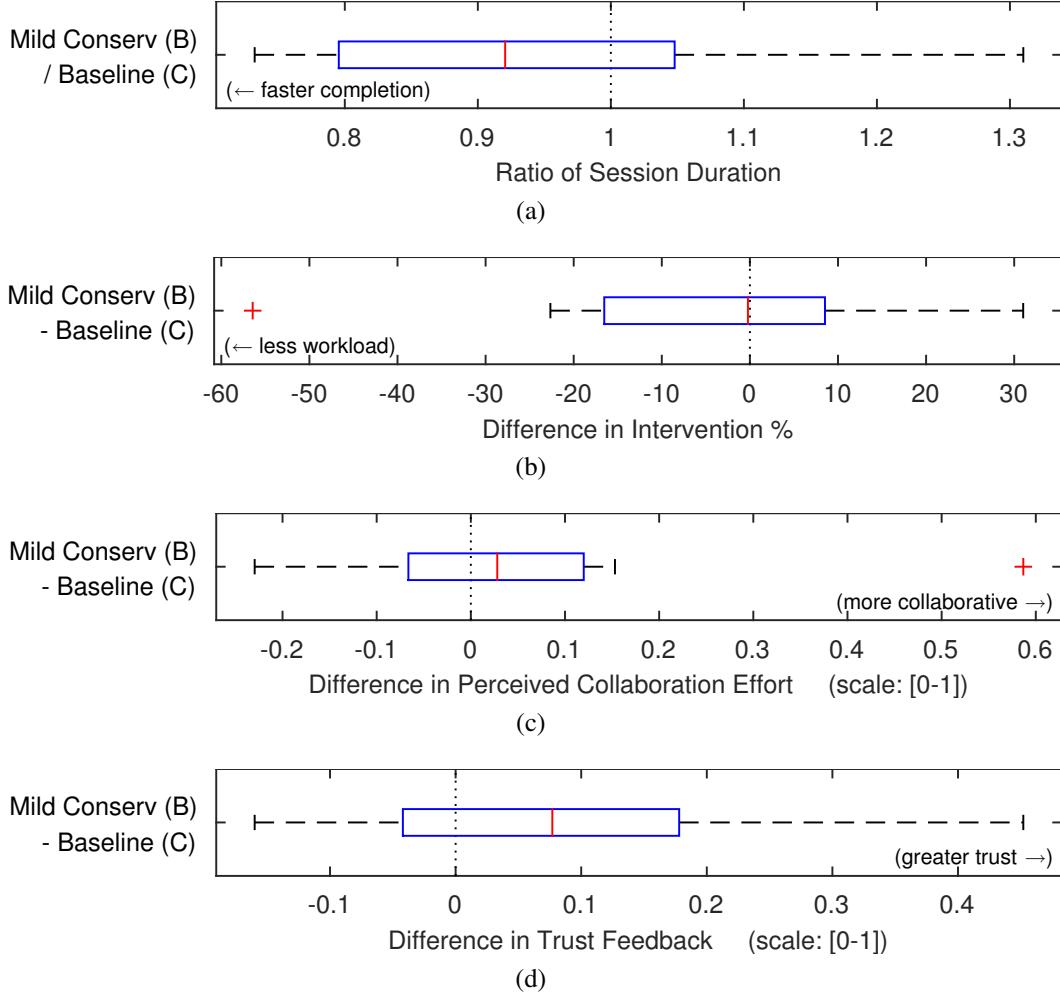


Figure 6–11: Per-metric efficiency comparisons for the TACtiC field trial extension are shown as box plots, while the vertical dashed lines denote the points of indifference between the mildly conservative trust-aware agent (B) and the baseline agent (C). Results suggest that the mildly conservative trust-aware agent achieved faster task completion, and was also perceived as more collaborative and trusting. Despite these efficiency gains, users nevertheless intervened at comparable frequencies when interacting with both agents.

having exhibited identical behaviors during most of the field sessions, with the sole exception of *rare occasions of salient trust loss*. Nevertheless, it is important to acknowledge that these results were qualitatively consistent with the demonstrated efficiency gains of TACtiC-enabled agents during our user study. This substantiates

both the practical efficacy of trust-seeking robots as well as their capacity to generalize to different robot vehicle types, remote or *in-situ* operating contexts, as well as diverse application domains.

Aggregate Team Efficiency

As shown in Figure 6–12, aggregate efficiency rankings demonstrated that the majority of participants preferred the mildly conservative agent over the plain APEX agent, both in terms of objective and subjective team efficiency facets. These preferences were supported by identical agent orderings computed using the Kemeny–Young voting scheme [110]. Nevertheless, neither the objective rankings (Friedman: $\chi^2(1) = 0.333, p = 0.564$) nor the subjective orderings (Friedman: $\chi^2(1) = 1.333, p = 0.248$) were statistically significant.

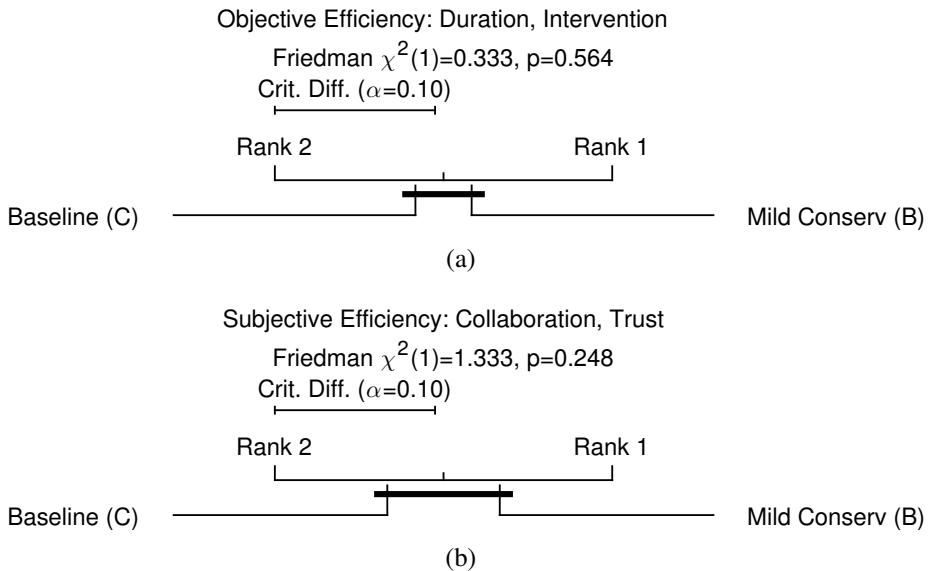


Figure 6–12: Mean agent rankings across users from the TACtiC field trial extension found that the mildly conservative trust-aware agent (B) was ranked highest for both objective and subjective efficiency facets. Nevertheless, post hoc Nemenyi testing revealed the *lack of critical differences among the agent rankings* (i.e. the presence of thick horizontal lines straddling both agents).

These aggregate evaluations are consistent with the per-metric assessments and suggest that there was a notable amount of variability between the experiences among different participants. We also attribute the lack of statistical significance in

these results to the limited number of participants and evaluation sessions. Both of these design factors were affected by the severe constraints imposed during these robot field trials. Despite the challenging field deployment hurdles, our TACtiC-enabled trust-seeking robot agent was still preferred over the baseline agent both in terms of objective and subjective team efficiency attributes. Also, these results are consistent with findings from our user study, which notably demonstrated statistically dominant preference of trust-seeking agents over plain adaptive agents within controlled conditions.

In summary, empirical findings from this field study extension provided additional support for Hypotheses 1 and 2, thus substantiating the efficiency gains contributed by trust-seeking robots. Also, all of the participants expressed content with the agents' performances and the overall experience of collaborating with a smart car. One user commented: "Since I was pressing the trust change buttons constantly during autonomous control, it dawned on me pretty late that I was not actually in control; even then I felt unusually comfortable." Another stated: "Once I got used to teaching the agent quickly, I was more confident in the narrow parts of the course, and overall my trust was increased." These reports suggest that agents that can react to user's actions and attitudes have great utility and promise toward enabling autonomous cars and other mobile robots to achieve efficient and trusting collaboration with humans.

6.4 Summary

In this chapter, we presented the final and completed form of our trust-seeking robot framework. This framework imbued autonomous agents with interactive behavior adaptation capabilities via Adaptive Parameter EXploration (APEX), the ability to infer their human supervisor's trust state in real-time using the Online Probabilistic Trust Inference Model (OPTIMo), and the novel capacity to react to

and resolve trust offenses via Trust-Aware Conservative Control (TACtiC). We described end-to-end instantiations and evaluations of trust-seeking robot agents for distinct tasks of collaborative aerial coverage and interactive autonomous driving. These implementations represent the *first-ever realizations* of autonomous robots that can react *in direct response* to changes in their human collaborator’s trust state.

Results from our large-scale interaction study and field trial extension have shown that the pervasive human notion of trust can be capitalized to maintain efficient supervisor-worker human-robot partnerships. These demonstrated efficiency gains also reflect our successful efforts at accurately modeling each user’s trust tendencies. All of these achievements were further contingent on the massive amount of systems development to realize our evaluations, which included an iteratively-designed and fully-automated study infrastructure, as well as real-world deployments under severely constrained logistics onto the state-of-the-art self-driving SL-Commander vehicle.

While there were some degrees of variability among users’ sensitivity toward robots that inferred and reacted to their trust states, some collaborators synergized exceptionally well with trust-seeking agents. These participants attained, for instance, over 50% improved task performance during the controlled study on aerial coverage, and 30% faster task completion in the follow-up interactive driving field trials. More importantly, *all* of our aggregate efficiency analyses have shown that trust-seeking robot agents were ranked higher than baseline adaptive agents devoid of trust-related capabilities, under both the objective and subjective facets of team efficiency, and within both our large-scale user study as well as its field trial extensions. Therefore, we expect these dominant preferences of trust-seeking agents to accumulate into every greater efficiency boosts in the long run.

Chapter 7

Conclusions

This thesis investigated several aspects of trust within asymmetric collaborative teams comprised of a human supervisor and an autonomous robot agent. Our end-goal of efficient team collaboration extended the dominant research goal in robotics of improving task performance, by also considering equally important factors such as mitigating unnecessary workload and ensuring high satisfaction.

Guided by extensive evidence from supervisor-worker human-robot collaboration experiences, we developed the trust-seeking robot framework that integrated into arbitrary robot agents to grant them the capacities to adapt and react to the human’s evolving trust state. We evaluated multiple instantiations of such trust-seeking robot agents within diverse application domains and demonstrated their contributions empirically toward improving team efficiency as well as actively mitigating teamwork breakdown. Although many robot learning methods, including our APEX module, have been shown to improve task performance, few autonomous systems have demonstrated the ability to cater to each human supervisor’s unique preferences and personalities. Our trust-seeking robot framework fulfilled both objectives simultaneously by imbuing autonomous robot agents with the capacity to adapt to their human collaborator’s actions *and* attitudes.

7.1 Summary of Contributions

The main contributions of this thesis reflect the individual components that make up our trust-seeking robot framework:

- Chapter 3 established the Adaptation from Participation (AfP) paradigm as a *fluid interaction scheme for non-expert supervisors* to collaborate with robot agents toward improving task performance and reducing active workload. These adaptive agents also sought to *gain their supervisor's trust implicitly via behavior imitation*.
- Chapter 4 described extensive data collection efforts for studying supervisor-worker team interactions from a large group of roboticists and enthusiasts with *wide varieties of experiences, attitudes, and geography*. Analyses on these datasets substantiated the *dominant influences* of personality factors that call for *personalized trust modeling*, and also revealed several key insights that shaped the structure of our temporal trust model.
- Chapter 5 presented the Online Probabilistic Trust Inference Model (OPTIMo) and demonstrated its *superior trust prediction performance* over existing formulations. OPTIMo's *unique ability of real-time trust inference* also opened up rich new possibilities for robot agents to *continuously react and adapt to the human's trust evolutions*.
- Chapter 6 proposed Trust-Aware Conservative Control (TACtiC) as a *generalized means* for the large class of locomotion-centric robot vehicles to remedy trust losses from the human supervisor. To the best of our knowledge, these TACtiC agents are the *first-ever instantiations* of autonomous robots that have the capacity to *react in direct response* to each human supervisor's evolving trust state.

Our research also called for a substantial amount of systems development, including general-purpose autonomous boundary tracking agents, field-ready integrations with aerial and terrestrial, and marine robot vehicles, as well as robust and fully automated study infrastructures. These endeavors amounted to over 53,000

lines of code that were deployed on various mobile and embedded computing architectures.

7.1.1 Interaction Design Guidelines for Supervisor-Worker Teams

The display interfaces for facilitating human-robot interactions in our numerous user studies and field trials are shown in Figure 7–1. In addition to these realizations, we have also experimented with various textual, icon, and graphical overlays to provide greater situational awareness to the human supervisor and also selectively guide their attention, adhering to established interface design guidelines [37].

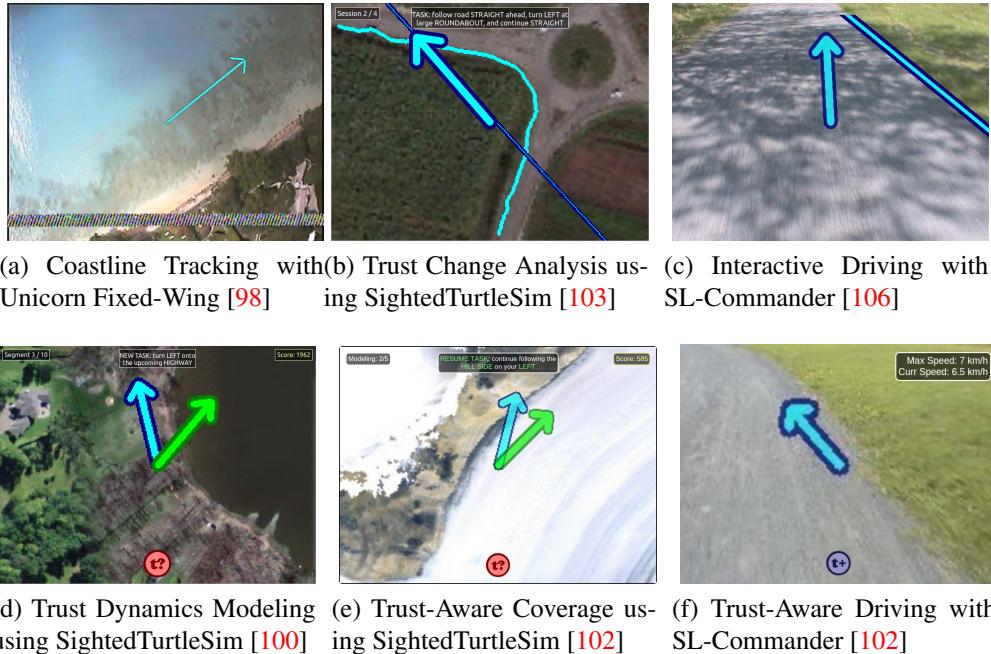


Figure 7–1: Evolution of display interfaces for our interactive boundary tracking robot agent throughout user study and field trial designs.

Along with each design iteration, we have collected anecdotal evidence and participant feedback that highlighted several important usability-related factors. We synthesized these observations into the following interface design guidelines for supervisor-worker human-robot teams:

- **Guideline 1:** *Continue to execute the robot agent and visualize its output commands during periods of human intervention*, so as to help the human

supervisor ascertain when the agent has adapted sufficiently and can resume autonomous control (Section 3.1);

- **Guideline 2:** *Avoid using complex visualizations of the robot agent’s internal state* (e.g. animations of detected boundary curves and line fits for our boundary tracking agent) to prevent confusing and distracting users who do not have expert understanding about the agent’s internal logic (Section 4.2.4);
- **Guideline 3:** *Reduce the number of inputs on peripheral devices by merging functionalities* (e.g. using the same analog stick on a gamepad for both intervention engagement and control) to help reduce mental strain, especially for users having limited experience with such devices (Section 4.3.2);
- **Guideline 4:** *Be wary of accidental and erroneous user inputs, such as reflex reactions to stimuli* (e.g. pressing incorrect trust change button following visual, audio, or vibrational prompt) (Section 4.3.4), and address these issues by educating users to be conscientious *as well as* modeling their effects accordingly (Section 5.2.2).

7.2 Future Directions

Our research in this thesis has contributed to many scientific strides, such as establishing interactive adaptation as a baseline capability of robot agents, identifying the dominant effects of personality factors toward human-robot trust dynamics, unifying trust modeling approaches of causal performance attribution and evidence grounding, and finally exploring agent behaviors that were explicitly induced by the human’s trust losses. Each of these contributions also opened the doors to numerous technical improvements as well as theoretical extensions, which we now highlight.

- **Time-aligned adaptation:** One improvement to APEX involves aligning image frames with past intervening commands to account for the supervisor’s hand-eye coordination delay, similar to related agent learning systems [53].

Preliminary testing showed that this noticeably improved adaptation quality and rate for rapidly-changing boundary targets. This technical extension also leads to several open-ended theoretical problems, such as ascertaining each individual's delay value and coping with the added latency in the agent's adaptation process.

We have also investigated a similar time alignment strategy for OPTIMo. Empirical results showed negligible difference in across-user model performance at interaction-time scales (e.g. $W = 0.5$), while such sub-second time alignment also made negligible changes to the aggregated interaction factors when using wider time windows.

- **Adaptive learning agents:** It would be valuable to combine the multi-task interactive capabilities of the Adaptation from Participation (AfP) paradigm with powerful *learning-from-scratch* capacities of Learning from Demonstration (LfD) techniques. We are keen to enhance robot agents with the ability to commit and recall learned abilities when encountering repeated scenes, while still adapting their behavior to the latest task conditions.
- **Persuasion to stop intervening:** Several users in our studies tended to intervene persistently even when the robot agent demonstrated consistent and excellent tracking performance. Therefore, we see value in *detecting* such cases of unnecessary interventions and *persuading* these supervisors to give the agent a fair chance to prove itself. In addition to the potential use of physiological sensors and electrodes, we are especially interested in non-intrusive solutions based on cognitive human-robot interaction techniques similar to our research in this thesis. In particular, the former problem can be addressed potentially by looking for discrepancies between observed interventions and OPTIMo's predictions, or by using APEX's long-term command cost. We

speculate that the latter problem may warrant occasional use of behavioral economics techniques such as *asymmetric dominance* [3].

- **Localized trust inference:** We have demonstrated OPTIMo’s state-of-the-art trust inference capabilities using a histogram-based engine implementation. While the histogram approximation accurately captures the underlying continuous belief at a given sampling resolution [91], it can nevertheless be wasteful by uniformly sampling the entire trust space at every time step. Since the human’s trust state is often highly localized (e.g. see Figure 5–4), we can exploit this feature to make efficient inferences using distribution-biased Markov-Chain Monte Carlo (MCMC) sampling methods.
- **Trust archetypes:** An exciting application of personalized OPTIMo instances is to group like-minded users based on the similarities in their trained model parameters. Such capabilities can then be used to cross-train or bootstrap trust models for new supervisors by leveraging the corpus of experiences from previous individuals with similar trust-induced reactions and assessments. This extension can also enable class-specific interface designs with different layouts and display elements catering to amateur, enthusiast, and expert users.
- **Unified trust modeling and elicitation:** We have the desire to unify trust modeling and trust-induced behavior alteration into a single computational framework, for instance, using the Influence Diagram extension to Probabilistic Graphical Modeling [54].
- **Lifelong trust adaptation:** We are keen to augment OPTIMo and TACtIC to cater to long-term evolutions in the supervisor’s behaviors and attitudes as she acclimates to the robot’s capabilities and limitations over time. Following the principle of Adaptation from Participation, parameters pertaining to the trust model, the amount of reactive conservative control, and personalized

trust shift thresholds could all benefit by evolving alongside this maturing supervisor-worker relationship.

- **Distrust vs. mistrust:** Two different sub-optimal teamwork states are caused by *distrust* (i.e. lack of trust) that is reflected by excessive interventions, versus *mistrust* (i.e. unjustifiably strong trust) issues such as inattentiveness and amplified expectations of competence and safety. Both causes can impede productivity and collaboration, and have even lead to fatal consequences in extreme cases [69]. The trust-seeking robot framework currently targets distrust, although it would be interesting to apply a similar methodology toward detecting and correcting cases of robot mistrust.
- **Trust in social robots:** Another venue of interest is to extend the trust-seeking robot framework to social robotics in medical care and household contexts. One exciting research problem in these domains is whether robots with anthropomorphic forms may evoke more affective reactions from human supervisors, and if so, then how to account for these factors in trust modeling and elicitation methods.

As these human-robot interactions verge toward collaboration among human peers, we must also begin to account for *intention-centric trust* factors in the agent’s inferences and reactions. This generalization may require a fundamentally different methodology however, since often when a violation arises due to poor intentions or integrity, it can be extremely laborious and challenging to regain trust [17].

7.3 Closing Words

This thesis has contributed a novel methodology and end-to-end realizations of affect-based robot agents that are predicated on the pervasive human sentiment of trust. Affect-based agent adaptation can be seen as the fourth evolutionary stage that

traces back to feedback control (i.e. adapt to odometry and sensor-based error estimates), then to reinforcement learning (i.e. adapt to world experiences), and in turn to Learning from Demonstration (i.e. adapt to the user's actions). All of these forms of adaptation are essential for the robot agent, along with its human supervisor, to cope with myriad types of challenging task conditions in an efficient and robust manner. As advanced robotics of the likes of self-driving cars, fleets of planes and drones, and household robots continue to immerse into the mainstream, *tenacious trust* and *rich interactions* will have ever greater importance toward enhancing and maintaining the efficiency of society's collaboration with these intelligent systems.

Appendix A

A Vision-Based Boundary Tracking Framework

All of our investigations into human-robot trust in this thesis are grounded within the visual navigation task domain. We had originally developed a general-purpose robot controller for tracking visually homogeneous terrain boundaries. This controller then served as the base robot agent for all of the trust-related studies and developments that followed in this thesis.

Boundary tracking tasks constitute an essential component within many aerial, terrestrial, and marine applications, including mapping the contours of biomes, monitoring the evolution of phenomena such as forest fires and oil spills, as well as inspecting pipes and roadways. Our boundary tracker evolved from an initial project for autonomously steering fixed-wing aerial drones alongside coastlines [98]. During the corresponding coastline tracking field trial, we encountered several limitations of the agent’s capabilities and of the nature of its interactions with a human operator. These observations directly fueled our investigations into human-robot trust, all of which were aimed at improving the collaborative efficiency with similar types of robot agents. We now present the final form of this visual-based robot controller, and also elaborate on a field deployment where it was used to steer an aerial drone along coastlines.

A.1 Boundary Tracking Pipeline

Our autonomous agent identifies and tracks salient terrain boundaries within image frames acquired from a mobile robot’s camera. The detected boundary location data are then converted into steering commands for the robot, which operates at a fixed forward velocity.

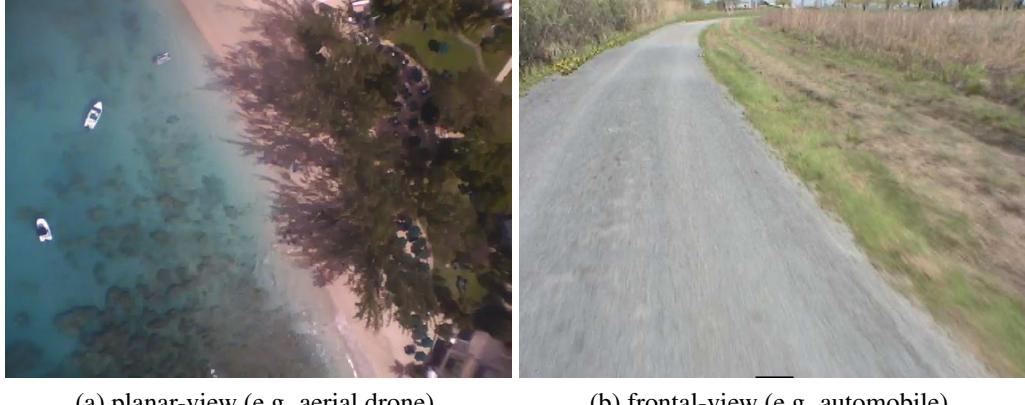


Figure A-1: Sample frames from different robot camera placements.

The visual processing pipeline has two variants with mostly overlapping elements, each designed to accommodate robot platforms with distinct camera placements. As seen in Figure A-1, the “planar-view” variant assumes that the image plane is parallel to the ground plane containing the target terrain boundary, whereas the “frontal-view” variant relaxes the planar constraint and assumes that the camera is pointing forwards, albeit with arbitrary pan and tilt. The planar-view agent is designed to accommodate aerial or marine robots and regulate their heading direction while operating in fixed-altitude or fixed-depth modes. In contrast, the frontal-view variant is intended to cater to wheeled or terrestrial platforms. Depending on the robot platform’s low-level actuation interface, this boundary tracking agent outputs steering commands in the form of either angular velocities, yaw rates, or incremental GPS waypoints.

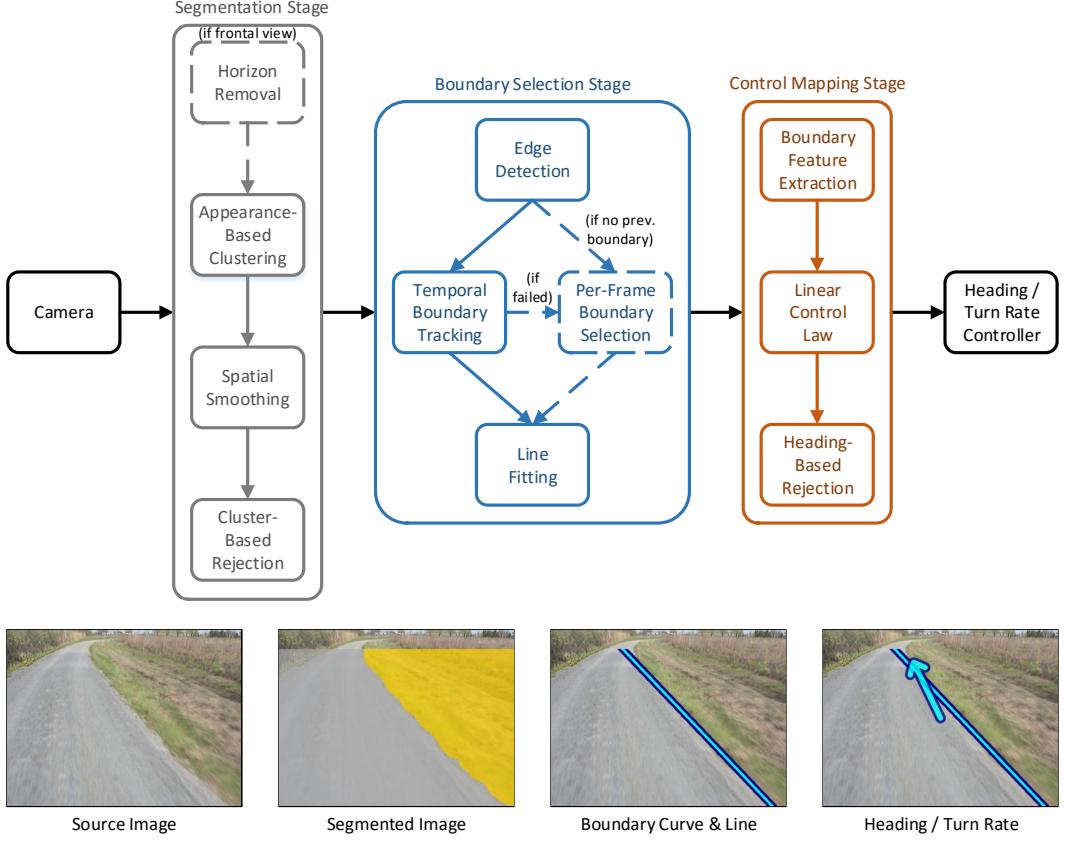


Figure A–2: Our boundary tracking pipeline consist of segmenting terrain patches in each camera frame, selecting a target boundary to track across frames, and mapping boundary data into steering commands; dashed blocks depict conditional steps.

A.1.1 Segmentation Stage

The boundary tracking pipeline has three processing stages, as illustrated in Figure A–2. In the first stage, visually homogeneous image patches are segmented out to highlight a particular target terrain of interest. Prior to segmentation however, a preprocessing step is needed for the frontal-view variant to remove horizon content from the scene and focus on tracking ground-plane terrains. This step is implemented by cropping out a certain portion of rows from the top of each frame, and is dictated by the *horizon cutoff ratio* configuration parameter H_0 .

The segmentation process takes as an input parameter the *appearance type* T_a , which corresponds to a chosen pixel representation that is well-suited at distinguishing the target visual regions from its surroundings for a given visual navigation task. We implemented several T_a options suitable for different application domains, for instance using hue to distinguish water bodies from land, and grayscale intensity to segment out man-made structures such as roadways. We also offered a third, more complex appearance type based on the Hue-Saturation-Value (HSV) colorspace that selectively favors the hue or value channels [89]. This *hue-value hybrid* representation is designed to facilitate recognizing objects in natural scenes.

Given a chosen appearance type T_a , we apply K-means cluster analysis to label each pixel in the image as either the target region or non-target region. Akin to the work of Stauffer *et al.* [87], when processing each frame, cluster centroids for the target and non-target labels are updated for a fixed number of iterations. This termination criterion allows the system to adapt to slowly-varying transitions in illumination and terrain appearances over time.

Since the cluster analysis above focuses solely on the pixel appearance, the segmented image often results in jagged boundaries and patches within patches. To enforce spatial consistency, we use a median filter to smooth out pixelation artifacts near the region contours. We also blend smaller-sized patches into its surrounding regions using a two-pass algorithm, to isolate the most dominant regions in each frame associated with the target terrain of interest.

This region merging step operates differently when tracking contours with an edge-like appearance (such as a coastline) versus boundaries with a strip-like look (such as a roadway). This *boundary type* T_b is specified as an input parameter to the boundary tracking pipeline. If the boundary is identified as a strip type, then the region merging step will consolidate smaller-sized patches until there is at most *two* connected regions for the target and non-target labels respectively. Otherwise,

for an edge-type boundary, the merging process continues until there is at most *a single* patch each for the two labels.

At the end of the segmentation stage, we ensure that the labeled regions for the target terrain occupy a reasonable portion of the frame. This cluster rejection check prematurely terminates the entire pipeline if the total label count for the target region divided by the frame resolution lies beyond designated minimum and maximum *cluster ratio* parameters $[R_{min}^{clstr}, R_{max}^{clstr}]$.

A.1.2 Boundary Selection Stage

The second pipeline stage consists of identifying the target boundary contour being tracked. This process is facilitated by the previous region merging step, which had removed small-sized spurious image patches from consideration. Consequently, an edge-type boundary curve is obtained by simply applying an edge detection operator to the binary-segmented image. In contrast, if both sides of a strip-type boundary are in view, then the target boundary curve is designated either as the longest curve within an initial frame, or the connected set of edgels in subsequent frames that is closest to the previous-detected target curve. The selected boundary is represented both by its edgel-based contour, as well as a linear fit.

A.1.3 Control Mapping Stage

The final pipeline stage is responsible for generating steering commands to drive the robot alongside the target terrain boundary based on the detected boundary within each camera frame. The final tracking behavior depends on several criteria: for instance, one may choose to steer aggressively toward the intersection point between the boundary curve and the image borders, or conservatively navigate to align the boundary curve with the center of the image plane. The degree of smoothness in tracking rapidly-changing boundaries can also be more generally specified in the form of a Proportional-Derivative (PD) controller. Also, the frontal-view controller variant must account for the projective mapping between the image

plane and the ground plane, which are not assumed to be parallel. Furthermore, when using this agent to steer wheeled robots alongside roadside curbs or similar contours, one must designate a nominal lateral distance to maintain between the vehicle and the target boundary.

In prior work, we parametrized robot-specific tracking requirements into separate pipeline iterations, such as steering aerial robots directly toward the boundary-line-to-image-border intersection [98], a PD controller generalizing the previous formulation [104], and a feature-based formulation for steering frontal-view wheeled robot at fixed distances alongside boundaries [104]. Our latest pipeline formulation subsumes all of these previous variations by computing the steering command as a *weighted* linear combination of features derived from the detected boundary data, namely:

- the image-plane direction pointing toward the intersection between the boundary curve and the image borders, closest to the robot’s forward direction;
- a similar direction pointing toward the intersection of the boundary’s line fit and the image borders;
- the orientation of the boundary line fit;
- values of all of the above features in the previous frame, if available;
- the intersection between the boundary line and an imaginary horizontal line going through the image center;
- a unit bias.

We saturate the output of this weighted linear expression at a specified *maximum steering magnitude* for safety. The resulting control signal is then transformed into either yaw rates, angular velocities, or incremental GPS waypoint commands, depending on the particular robot platform’s actuation interface. The *feature weights* in the linear control law implicitly encode various attributes of the tracking process, such as its responsiveness to changes in the boundary’s shape, perspective

compensation for non-planar-aligned cameras, and the nominal lateral spatial distance between the robot and the boundary.

A.1.4 Summary of Configuration Parameters

As described above, this general-purpose visual navigation framework can be used to steer different robot configurations and along various types of terrain boundaries. Nevertheless, the accuracy at which the agent can detect a specific boundary of interest and steer alongside it depends strongly on choosing appropriate values for the various pipeline parameters, namely:

- appearance type: hue, grayscale, or hue-value hybrid
- boundary type: edge or strip
- horizon ratio (planar-view variant only)
- feature weights in the control law

In addition to the previous configuration parameters, this autonomous agent features several secondary system settings, such as the cluster ratio bounds, and the maximum steering magnitude. We can tune these nuisance parameters by carrying out sensitivity analyses when encountering a new robot configuration or a new boundary tracking task. Consequently, these settings have negligible impacts on the agent’s moment-to-moment performance.

A.2 Aerial Coastline Tracking Field Trial

To assess the initial implementation of our autonomous boundary tracking agent, we carried out a field trial that used the Unicorn fixed-wing UAV to track coastlines. During three initial manual flight sessions, we collected aerial footage of a 1 km stretch of tropical shoreline, and used this dataset to assess the agent’s coastline tracking performance in a human-like manner. This dataset was also used to tune the agent’s various configuration parameters. Subsequently, we deployed our autonomous agent to steer the fixed-wing drone along the same 1 km coastline in a fourth flight.

A.2.1 Infrastructure

We integrated the boundary tracking agent to steer the Unicorn UAV through its commercial ground control software. Specifically, the drone’s heading direction was regulated by repeatedly updating a single GPS waypoint, incrementally away from the UAV’s location, in the desired heading specified by our agent. The boundary tracker’s visual input comprised of analog-transmitted frames from the robot’s onboard camera, which were digitized into a resolution of 320×240 pixels at a 10 Hz frame rate.



Figure A–3: The Unicorn UAV’s integrated camera (black unit) is attached to a gimbal and is protected by a transparent plastic hull. We also secured an iPod nano to the UAV’s underbelly to record high-resolution flight footage that are devoid of analog transmission artifacts.

This onboard camera was mounted on a gimbal, as shown in Figure A–3. We built a software controller to regulate the gimbal’s position and align its viewing direction continuously with the gravity vector based on the onboard Inertial Measurement Unit (IMU). This solution ensured that the camera frames were parallel to the ground plane, thus minimizing perspective distortions. We separately mounted an iPod nano on the vehicle’s underbelly to record high-quality and non-transmitted aerial footage, with a 640×480 pixel resolution at 30 Hz.

Although the drone’s camera positioning was constantly regulated to compensate for roll, tracking the coastline boundary using frames received by ground control was nevertheless challenging due to the presence of noise induced by the analog video transmission. Concretely, types of noise included scan line artifacts, color distortions, and frame tearing, as illustrated in Figure A–4.

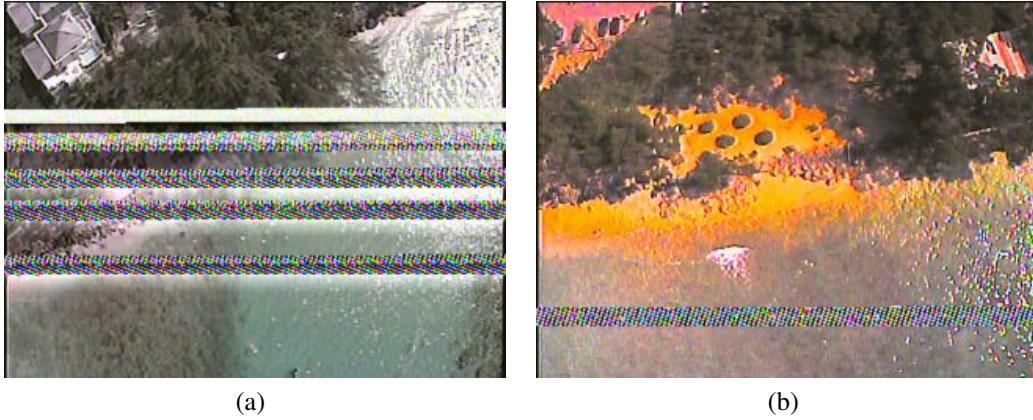


Figure A-4: Analog frames received from the Unicorn UAV’s camera contain significant transmission noise, including scan line artifacts and color distortions.

To cope with these practical issues, the initial implementation of the boundary tracking pipeline looked for potential processing failures at two stages. Firstly, we configured the segmentation stage to isolate blue-hue watery terrain specifically and rejected frames that exceeded a prescribed range $[R_{min}^{water}, R_{max}^{water}]$ of sensible amounts of watery regions in each frame. Also, the control mapping stage produced planar heading commands in the direction of the boundary-line-to-image-border intersection nearest to the front of the aerial vehicle. To ensure that this simplistic controller did not produce jittery motions, we rejected frames whenever the steering command changed abruptly, by imposing an upper bound on the maximum change in magnitude between consecutive steering commands R_{max}^{steer} .

A.2.2 Procedure

The target flight course, as depicted in Figure A-5, consists of a 1 km stretch of shoreline on the tropical island of Barbados. This coastline path features multiple terrain types with differing visual appearances, including sandy beaches, forested patches, seaside buildings, and shallow coral reefs in the water.

After flying the UAV along the course three times using manual GPS waypoints, we extracted nearly 6000 frames from the iPod nano’s recorded videos. Each



Figure A–5: Our fixed-wing UAV (red icon) executed multiple flights over a 1km target stretch of tropical shoreline during our coastline tracking field trial.

frame t was processed through the boundary tracking pipeline, resulting in a corresponding target heading direction $\phi_t \in [0^\circ, 360^\circ]$. We also asked five volunteers to manually label a line in each frame corresponding to the coastline’s position, which was then converted into a ground-truth heading direction $\hat{\phi}_t^i$ per person i .

To quantify our autonomous agent’s coastline tracking performance, we computed the *steering error* $\Delta\phi_t$ of each frame t as the mean angular difference from ground-truth headings. As baseline for comparison, we also calculated the *inter-human steering discrepancy* $\Delta\hat{\phi}_t$ as the mean pairwise heading differences among human-labeled targets:

$$\Delta\phi_t = \frac{1}{5} \sum_{i=1}^5 \left| (\phi_t - \hat{\phi}_t^i + 180^\circ) \bmod 360^\circ - 180^\circ \right| \quad (\text{A.1})$$

$$\Delta\hat{\phi}_t = \frac{1}{(5-1)!} \sum_{i=1}^{5-1} \sum_{j=i+1}^5 \left| (\hat{\phi}_t^i - \hat{\phi}_t^j + 180^\circ) \bmod 360^\circ - 180^\circ \right| \quad (\text{A.2})$$

A.2.3 Empirical Analyses

The inter-human steering discrepancy values shown in Figure A–6 suggest that users strongly disagreed with each other in regards to the coastline’s location and the target heading direction during the middle portion of each of the three manual

flights. This disagreement was attributed to a long patch of shallow coral reef, which blurred the exact water-land boundary. The fact that even humans could not agree upon a consistent steering direction illustrates the inherent difficulty of the general-purpose visual boundary detection task.

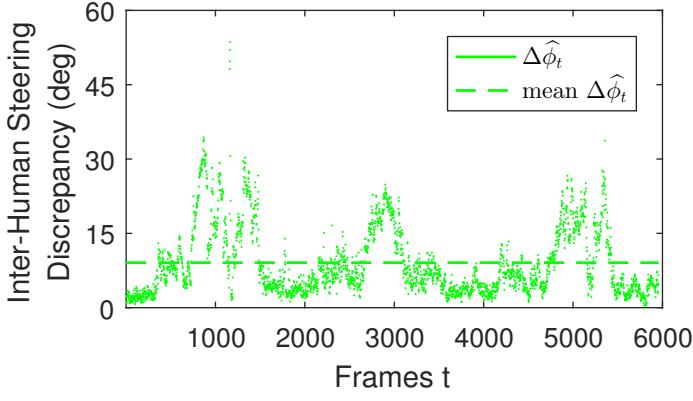


Figure A-6: Inter-human steering discrepancy $\Delta\hat{\phi}_t$ among 5 volunteers across 3 flight sessions (~6000 frames): even humans strongly disagreed on the coastline positioning in some scenes, notably for a patch of shallow reef during the middle of each session.

Using the collected aerial dataset, we separately analyzed the effects of the minimum water ratio R_{min}^{water} and the maximum steering change R_{max}^{steer} parameters. As shown in Figure A-7a, imposing strict requirements for minimum watery regions in each frame did result in reduced steering error, although at the expense

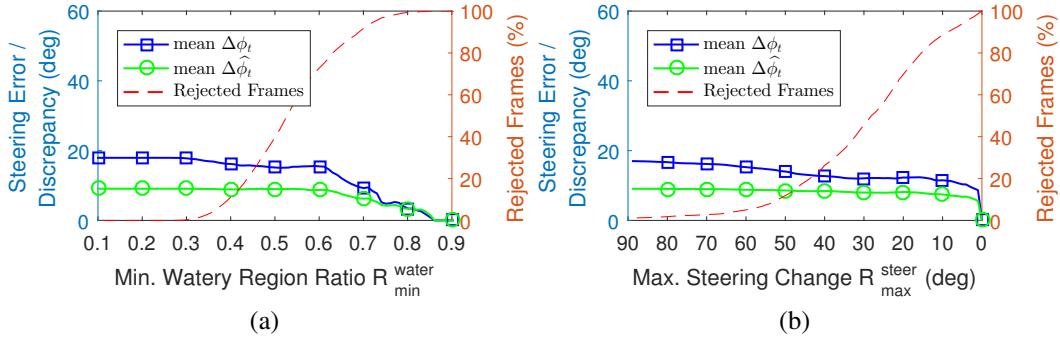


Figure A-7: The boundary tracking agent's steering error $\Delta\hat{\phi}_t$ compared against inter-human steering discrepancy $\Delta\hat{\phi}_t$, as functions of (a) the minimum water-labeled frame ratio R_{min}^{water} and (b) maximum steering change R_{max}^{steer} parameters: (b) reduced steering error without rejecting as many frames as (a).

of a significant portion of rejected frames. On the other hand, Figure A–7b depicts similar attenuations to the steering error by tightening the maximum steering change R_{max}^{steer} , while preserving the majority of frames.

Furthering these analyses, we assessed per-frame boundary tracking performance under two configurations that tuned performance separately based on the minimum water ratio ($R_{min}^{water} = 0.5$, $R_{max}^{water} = 1$, $R_{max}^{steer} = 180^\circ$), versus using the maximum steering change parameter ($R_{min}^{water} = 0$, $R_{max}^{water} = 1$, $R_{max}^{steer} = 45^\circ$). Our empirical testing showed that the incremental-waypoint-based control scheme used to steer the drone was tolerant to several seconds of regulation inactivity before the UAV would fly out of range of the boundary target. Figure A–8a shows that the former setting brought steering error to within 1.5 times of the corresponding inter-human discrepancy data. Unfortunately, a large number of *consecutive* frames were rejected (e.g. $t \in [3500, 3800]$ and $t \in [5550, 5850]$), meaning that the UAV’s heading would not be regulated for periods of up to 9 seconds. In contrast, the latter pipeline configuration resulted in steering errors with a factor of 1.3 to the inter-human discrepancies, and notably with a mean error of below 15° , as seen

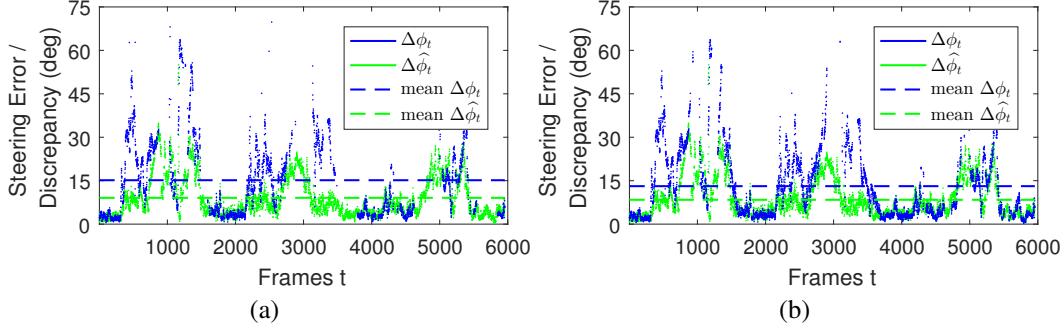


Figure A–8: The boundary tracking agent’s steering error $\Delta\phi_t$ compared against inter-human discrepancies $\hat{\Delta\phi}_t$ under two parameter configurations: (a) tightening the minimum water ratio ($R_{min}^{water} = 0.5$) gave decent performance (less than twice of the mean inter-human discrepancy) but of dropping many consecutive frames; (b) constraining the maximum steering change ($R_{max}^{steer} = 45^\circ$) achieved superior tracking accuracy, with at most 4 seconds of consecutive dropped frames, and only for frames in which humans strongly disagreed with each other.

in Figure A–8b. More importantly, the longest contiguous sequence of dropped frames in this configuration was merely 4 seconds long ($t \in [2900, 3100]$), which meant that the drone was constantly being steered.

A.2.4 Autonomous Flight Results

In light of the previous offline analyses, we configured the boundary tracking agent with lenient rejection checks ($R_{min}^{water} = 0.35$, $R_{max}^{water} = 0.8$, $R_{max}^{steer} = 135^\circ$) and deployed it to control the UAV during a 4th flight session. As seen in Figure A–5, the UAV successfully flew alongside the 1km coastline segment, while traveling at 14m/s ground speed and 150m altitude, and with a 7m/s lateral wind condition. As the vehicle was about to reach its operating range at the end of the flight, however, it came across a large shallow reef, which our boundary tracking agent mistakenly labeled as land. Subsequently, the UAV flew toward the ocean, and we thus had to disengage the autonomous agent and command it to fly back to home base. Nevertheless, this flight session demonstrated that our visual navigation robot agent was capable of tracking shorelines competently, while operating under real-world and noisy flight and imaging conditions.

Based on the results of our empirical analyses and field evaluations, we concluded that it was safer to refine the agent’s tracking performance via the heading-based rejection criterion rather than via the cluster ratio check. Thus, in all subsequent boundary tracking agent instances, we set the cluster rejection check to a loose bound ($R_{min}^{clstr} = 0.015$, $R_{max}^{clstr} = 0.975$) to only reject frames that were dominated by a single segmentation label. Also, we transformed the heading-based criterion R_{max}^{steer} into the maximum steering magnitude threshold, as discussed in Section A.1.3. Whenever deploying the agent on a new robot platform, we tuned this configuration parameter using evidence maximization, similar to the approach carried out during the coastline tracking field trial.

A.3 Summary

We developed a fully autonomous vision-based control framework for steering mobile robots along the boundaries of various terrains or regions of interest. This agent has an interpretable processing pipeline that combines classical image analysis techniques with a feature-based control law. We deployed this autonomous agent onboard a fixed-wing aerial robot, and successfully demonstrated fully autonomous flight along a 1km coastline with varied terrains.

Reflecting upon our coastline tracking field trial, we hypothesize that the reef-land confusion event may have been rectified if the operator was able to override the agent momentarily and manually steer the UAV back toward land. Separately, to prepare for this autonomous aerial tracking session, we had to collect a *tedious* amount of flight footage, apply *expert knowledge* of the agent’s internal workings, and ultimately choose a *static* setting that worked adequately across diverse task conditions. We defined the attributes of a supervisor-worker human-robot team (see Section 1.2) specifically to address these concerns. These issues also motivated our development of an interactive behavior adaptation method for robot agents in Chapter 3 that can learn from occasional interventions issued by non-expert operators.

Furthermore, the operator’s trust in the autonomous agent varied throughout the fourth flight session. Since our analyses had previously revealed that tracking reef-bound coastlines was challenging, we became extremely cautious while the UAV flew above the large reef patch, and gained confidence after seeing it successfully navigated through the segment. This evolution in the degree of trust toward the agent reflects our perception of its moment-to-moment task performance, and can be useful both as an assessment tool and as a trigger for the agent to become weary of potential challenging task conditions. In Chapters 4 and 5 we will quantify and model this trust signal, and we will propose a means for the autonomous agent to capitalize on trust toward maintaining efficient operations in Chapter 6.

REFERENCES

- [1] P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceeding of the International Conference on Machine Learning (ICML'04)*, 2004.
- [2] B. D. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57:469–483, 2009.
- [3] D. Ariely. *Predictably Irrational: The Hidden Forces That Shape Our Decisions*. HarperCollins, 2008.
- [4] R. C. Arkin, P. Ulam, and A. R. Wagner. Moral decision making in autonomous systems: Enforcement, moral emotions, dignity, trust, and deception. *Proceeding of the IEEE*, 100(3):571–589, 2012.
- [5] R. Aufrère, R. Chapuis, and F. Chausse. A model-driven approach for real-time road recognition. *Machine Vision and Applications*, 13:95–107, 2001.
- [6] A. Billard, S. Calinon, R. Dillmann, and S. Schaal. Survey: Robot programming by demonstration. *Handbook of Robotics*, ch. 59, 2008.
- [7] A. M. Bisantz and Y. Seong. Assessment of operator trust in and utilization of automated decision-aids under different framing conditions. *Industrial Ergonomics*, 28(2):85–97, 2001.
- [8] J.F. Bonnefon, A. Shariff, and I. Rahwan. The social dilemma of autonomous vehicles. *Science*, 352(6293):1573–1576.
- [9] J. D. Brookshire. Enhancing multi-robot coordinated teams with sliding autonomy. Master’s thesis, School of Computer Science, Carnegie Mellon University, 2004.
- [10] Bombardier Recreational Products (BRP). Commander — Can-Am. can-am.brp.com/off-road/side-by-side/commander/commander.html. Accessed: 2016-07-28.
- [11] D. J. Bruemmer, J. L. Marble, D. D. Dudenhoeffer, M. Anderson, and M. D. McKay. Mixed-initiative control for remote characterization of hazardous environments. In *Proceedings of the 36th Annual Hawaii International Conference on System Sciences*, pages 1–9, 2003.

- [12] Tesla Canada. Model s autopilot press kit. www.tesla.com/en_CA/presskit/autopilot. Accessed: 2016-07-28.
- [13] M. Carbone, M. Nielsen, and V. Sassone. A formal model for trust in dynamic networks. In *Proceeding of the 1st International Conference on Software Engineering and Formal Methods*, pages 54–61, 2003.
- [14] S. Chernova. *Confidence-Based Robot Policy Learning from Demonstration*. PhD thesis, School of Computer Science, Carnegie Mellon University, 2009.
- [15] B. CK Choi and A. WP Pak. A catalog of biases in questionnaires. *Preventing Chronic Disease*, 2(1), 2005.
- [16] K. Chopra and W. A. Wallace. In *Proceeding of the 36th Hawaii International Conference on System Sciences*, volume 9, 2003.
- [17] S. M.R. Covey. *The Speed of Trust: The One Thing that Changes Everything*. Free Press, 2006.
- [18] J. A. Cowley and H. Youngblood. Subjective response differences between visual analogue, ordinal and hybrid response scales. *Proceeding of the Human Factors and Ergonomics Society Annual Meeting*, 53(25):1883–1887, 2009.
- [19] J. Crisman and C. Thorpe. Color vision for road following. In *Proceeding of SPIE Conference on Mobile Robots*, 1988.
- [20] M. L. Cummings, C. E. Nehme, J. Crandall, and P. Mitchell. *Predicting Operator Capacity for Supervisory Control of Multiple UAVs*, pages 11–37. Springer Berlin Heidelberg, 2007.
- [21] P. de Vries, C. Midden, and D. Bouwhuis. The effects of errors on system trust, self-confidence, and the allocation of control in route planning. *International Journal of Human-Computer Studies*, 58(6):719–735, 2003.
- [22] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *J. of Machine Learning Research*, 7:1–30, 2006.
- [23] M. Desai. *Modeling Trust to Improve Human-Robot Interaction*. PhD thesis, Computer Science Department, University of Massachusetts Lowell, 2012.
- [24] M. Desai, M. Medvedev, M. Vázquez, S. McSheehy, S. Gadea-Omelchenko, C. Bruggeman, A. Steinfeld, and H. Yanco. Effects of changing reliability on trust of robot systems. In *Proceeding of the ACM/IEEE International Conference on Human-Robot Interaction (HRI'12)*, pages 73–80, 2012.
- [25] M. B. Dias, B. Kannan, B. Browning, E. Gil Jones, B. Argall, M. M. Veloso, and A. Stentz. Sliding autonomy for peer-to-peer human-robot teams. In

Proceeding of the 10th International Conference on Intelligent Autonomous Systems, 2008.

- [26] S. R. Dixon and C. D. Wickens. Automation reliability in unmanned aerial vehicle control: A reliance-compliance model of automation dependence in high workload. *Human Factors*, 48(3):474–486, 2006.
- [27] A. Dragan and S. Srinivasa. Formalizing assistive teleoperation. In *Robotics: Science and Systems*, 2012.
- [28] G. Dudek, M. Jenkin, C. Prahacs, A. Hogue, J. Sattar, P. Giguère, A. German, H. Liu, S. Saunderson, A. Ripsman, S. Simhon, L. A. Torres-Mendez, E. Milios, P. Zhang, and I. Rekleitis. A visually guided swimming robot. In *Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'05)*, pages 3604–3609, 2005.
- [29] M. T. Dzindolet, S. A. Peterson, R. A. Pomranky, L. G. Pierce, and H. P. Beck. The role of trust in automation reliance. *International Journal of Human-Computer Studies*, 58(6):697–718, 2003.
- [30] M. R. Endsley and D. B. Kaber. Level of automation effects on performance, situation awareness and workload in a dynamic control task. *Ergonomics*, 42(3):462–492, 1999.
- [31] R. Falcone and C. Castelfranchi. Trust dynamics: How trust is influenced by direct experiences and by trust itself. In *Proceeding of the 3rd International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'04)*, pages 740–747, 2004.
- [32] AAA Foundation for Traffic Safety. Aggressive driving: Research update. Technical report, AAAFoundation.org, 2009.
- [33] B. L. Fredrickson. Extracting meaning from past affective experiences: The importance of peaks, ends, and specific emotions. *Cognition and Emotion*, 14(4):577–606, 2000.
- [34] E. Freedy, E. DeVisser, G. Weltman, and N. Coeyman. Measurement of trust in human-robot collaboration. In *International Symposium on Collaborative Technologies and Systems (CTS'07)*, pages 106–114, 2007.
- [35] M. Friedman. A comparison of alternative tests of significance for the problem of m rankings. *Annals of Mathematical Statistics*, 11(1):86–92, 1940.
- [36] F. Gao, A. Clare, J. Macbeth, and M. L. Cummings. Modeling the impact of operator trust on performance in multiple robot control. In *AAAI Spring Symposium: Trust and Autonomous Systems*, 2013.

- [37] M. Goodrich and D. Olsen. Seven principles of efficient human robot interaction. In *Proceeding of the IEEE International Conference on Systems, Man and Cybernetics*, pages 3942–3948, 2003.
- [38] M. A. Goodrich and A. C. Shultz. Human-robot interaction: A survey. *Foundations and Trends in Human-Computer Interaction*, 1(3):203–275, 2007.
- [39] D. H. Grollman and O. C. Jenkins. Dogged learning for robots. In *Proceeding of the IEEE International Conference on Robotics and Automation (ICRA '07)*, pages 2483–2488, 2007.
- [40] R. J. Hall. Trusting your assistant. In *Proceeding of the 11th Knowledge-Based Software Engineering Conference*, pages 42–51, 1996.
- [41] P.A. Hancock, D.R. Billings, K.E. Schaefer, J.Y.C. Chen, E.J. De Visser, and R. Parasuraman. A meta-analysis of factors affecting trust in human-robot interaction. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 53(5):517–527, 2011.
- [42] R. Hardin. *Trust*. Polity Press, 2006.
- [43] S. G. Hart. NASA-Task Load Index (NASA-TLX); 20 years later. In *Proceeding of the Human Factors and Ergonomics Society Annual Meeting*, volume 50, pages 904–908, 2006.
- [44] Y. Hochberg and A. C. Tamhane. *Multiple Comparison Procedures*. John Wiley & Sons, 1987.
- [45] SAE International. SAE AS-4 JAUS Standard. www.sae.org/standardsdev/jaus.htm. Accessed: 2016-07-28.
- [46] A. Jain, H. S. Koppula, B. Raghavan, and A. Saxena. Car that knows before you do: Anticipating maneuvers via learning temporal driving models. In *Proceedings of the International Conference on Computer Vision (ICCV'15)*, 2015.
- [47] J.Y. Jian, A. M. Bisantz, and C. G. Drury. Foundations for an empirically determined scale of trust in automated systems. *International Journal of Cognitive Ergonomics*, 4(1):53–71, 2000.
- [48] J. D. Johnson. Types of automation failure: The effects on trust and reliance on automation. Master's thesis, School of Psychology, Georgia Institute of Technology, 2004.
- [49] A. Jøsang, S. Marsh, and S. Pope. Exploring different types of trust propagation. In *Proceeding of the 4th International Conference on Trust Management (iTrust'06)*, 2006.

- [50] H. H. Kelly. *Attribution: Perceiving the Causes of Behavior*, chapter Causal Schemata and the Attribution Process. General Learning Press, 1972.
- [51] R. Kerr and R. Cohen. Modeling trust using transactional, numerical units. In *Proceeding of the International Conference on Privacy, Security and Trust (PST'06)*, pages 21:1–21:11, 2006.
- [52] M. Kirshenbaum. *I Love You But I Don't Trust You: The Complete Guide to Restoring Trust in Your Relationship*. Berkley, 2012.
- [53] W. B. Knox. *Learning from Human-Generated Reward*. PhD thesis, Department of Computer Science, The University of Texas at Austin, 2012.
- [54] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.
- [55] H. Lacohée, P. Cofta, A. Phippen, and S. Furnell. *Understanding public perceptions: Trust and engagement in ICT-mediated services*. International Engineering Consortium, 2009.
- [56] J. Lee and N. Moray. Trust, control strategies and allocation of function in human-machine systems. *Ergonomics*, 35(10):1243–1270, 1992.
- [57] J. D. Lee and K. A. See. Trust in automation: Designing for appropriate reliance. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 46:50–80, 2004.
- [58] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J Z. Kolter, D. Langer, O. Pink, V. Pratt, et al. Towards fully autonomous driving: systems and algorithms. In *Proceeding of IEEE Intelligent Vehicles Symposium (IV)*, pages 163–168, 2011.
- [59] R. Likert. A technique for the measurement of attitudes. *Archives of Psychology*, 22:1–55, 1932.
- [60] Y. Ma, J. Kosěcká, and S. S. Sastry. Vision guided navigation for a non-holonomic mobile robot. *IEEE Transactions on Robotics and Automation*, 15(3):521–536, 1999.
- [61] B. Malle, M. Scheutz, T. Arnold, J. Voiklis, and C. Cusimano. Sacrifice one for the good of many? people apply different moral norms to human and robot agents. In *Proceeding of the ACM/IEEE International Conference on Human-Robot Interaction (HRI'15)*, pages 117–124, 2015.
- [62] D. H. McKnight and N. L. Chervany. The meanings of trust. Technical report, University of Minnesota, 1996.

- [63] N. Moray and T. Inagaki. Laboratory studies of trust between humans and machines in automated systems. *Transactions of the Institute of Measurement and Control*, 21(4–5):203–211, 1999.
- [64] B. M. Muir. *Operators' trust in and use of automatic controllers in a supervisory process control task*. PhD thesis, University of Toronto, 1989.
- [65] A. Y. Ng and S. J. Russell. Algorithms for inverse reinforcement learning. In *Proceeding of the International Conference on Machine Learning (ICML'00)*, pages 663–670, 2000.
- [66] M. N. Nicolescu and M. J. Mataric. Experience-based representation construction: Learning from human and robot teachers. In *Proceeding of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'01)*, pages 740–745, 2001.
- [67] M. Nimmelman. SL-Commander - Design Highlights. In *Unmanned Systems Canada*, 2012.
- [68] T. Nomura, T. Suzuki, T. Kanda, and K. Kato. Measurement of negative attitudes toward robots. *Interaction Studies*, 7(3):437–454, 2006.
- [69] R. Parasuraman and V. Riley. Humans and automation: Use, misuse, disuse, abuse. *H Factors*, 1:230–253, 1997.
- [70] Parrot. AR.Drone 2.0 official site. www.parrot.com/usa/products/ardrone-2/. Accessed: 2016-07-28.
- [71] D. A. Pomerleau. Efficient training of artificial neural networks for autonomous navigation. *Neural Computing*, 3(1):88–97, 1991.
- [72] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. B. Foote, J. Leibs, R. Wheeler, and A. Y. Ng. ROS: An open-source robot operating system. In *ICRA Workshop on Open Source Software*, 2009.
- [73] N. D. Ratliff. *Learning to Search: Structured Prediction Techniques for Imitation Learning*. PhD thesis, School of Computer Science, Carnegie Mellon University, 2009.
- [74] U. Reips and F. Funke. Interval-level measurement with visual analogue scales in internet-based research: VAS generator. *Behavior Research Methods*, 40(3):699–704, 2008.
- [75] Clearpath Robotics. Heron unmanned vessel for bathymetry. www.clearpathrobotics.com/heron-bathymetry-unmanned-surface-vessel/. Accessed: 2016-07-28.

- [76] Clearpath Robotics. Husky UGV - Outdoor Field Research Robot by Clearpath. www.clearpathrobotics.com/husky-unmanned-ground-vehicle-robot/. Accessed: 2016-07-28.
- [77] S. Ross. *Interactive Learning for Sequential Decisions and Predictions*. PhD thesis, School of Computer Science, Carnegie Mellon University, 2013.
- [78] D. Sadigh, S. Sastry, S. A. Seshia, and A. D. Dragan. Planning for autonomous cars that leverages effects on human actions. In *Proceedings of the Robotics: Science and Systems Conference (RSS)*, pages 239–248, 2016.
- [79] J. Sanchez. *Factors that affect trust and reliance on an automated aid*. PhD thesis, School of Psychology, Georgia Institute of Technology, 2006.
- [80] T. L. Sanders, T. Wixon, K. E. Schafer, J. Y. C. Chen, and P. A. Hancock. The influence of modality and transparency on trust in human-robot interaction. In *Proceeding of the IEEE International Inter-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA'14)*, pages 156–159, 2014.
- [81] J. Sattar and J. Little. Ensuring safety in human-robot dialog - a cost-directed approach. In *Proceeding of the IEEE International Conference on Robotics and Automation (ICRA'14)*, pages 6660–6666, 2014.
- [82] K. E. Schaefer. *The Perception and Measurement of Human-Robot Trust*. PhD thesis, College of Sciences, University of Central Florida, 2013.
- [83] N. Schwarz, H. Bless, F. Strack, G. Klumpp, and A. Simons. Ease of retrieval as information: Another look at the availability heuristic. *Journal of Personality and Social Psychology*, 61(2):195–202, 1990.
- [84] T. B. Sheridan. *Telerobotics, automation, and human supervisory control*. MIT Press, 1992.
- [85] T.B. Sheridan and W.L. Verplank. *Human and Computer Control of Under-sea Teleoperators*. Man-Machine Systems Laboratory, Massachusetts Institute of Technology, 1978.
- [86] V. A. Shia, Y. Gao, R. Vasudevan, K. D. Campbell, T. Lin, F. Borrelli, and R. Bajcsy. Semiautonomous vehicular control using driver modeling. *IEEE Transactions on Intelligent Transportation Systems*, 15(6):2696–2709, 2014.
- [87] Chris Stauffer and W. EL Grimson. Adaptive background mixture models for real-time tracking. In *Proceeding of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR'99)*, volume 2, pages 252–258, 1999.

- [88] A. Steinfeld, O.C. Jenkins, and B. Scassellati. The Oz of Wizard: Simulating the human for interaction research. In *Proceeding of the ACM/IEEE International Conference on Human-Robot Interaction (HRI'09)*, pages 101–107, 2009.
- [89] S. Sural, G. Qian, and S. Pramanik. Segmentation and histogram generation using the hsv color space for image retrieval. In *Proceeding of the IEEE International Conference on Image Processing (ICIP'02)*, pages 589–592, 2002.
- [90] Lockheed Martin Procerus Technologies. UAV Test Platforms: EPP foam Unicorn wing. www.lockheedmartin.ca/content/dam/lockheed/data/ms2/documents/procerus/UAV_Platform_04_18_2008.pdf. Accessed: 2016-07-28.
- [91] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.
- [92] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, MN Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer, et al. Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics*, 25(8):425–446, 2008.
- [93] U.S. National Highway Traffic Safety Administration (NHTSA). NHTSA Automated Vehicle Operational Guidance Public Meeting. www.nhtsa.gov/nhtsa/symposiums/april2016/index.html. Accessed: 2016-07-28.
- [94] N. Wang, D. V. Pynadath, and S. G. Hill. Trust calibration within a human-robot team: Comparing automatically generated explanations. In *Proceeding of the ACM/IEEE International Conference on Human-Robot Interaction (HRI'16)*, pages 109–116, 2016.
- [95] K. Werbach and D. Hunter. *For The Win: How Game Thinking Can Revolutionize Your Business*. Wharton Digital Press, 2012.
- [96] A. Xu. anqixu/sightedturtlesim: Holonomic robot simulator for agents with 2D XY + height motion capabilities and with downward-facing cameras. github.com/angqixu/sightedturtlesim. Accessed: 2016-07-28.
- [97] A. Xu. 2-Step Temporal Bayesian Networks (2TBN): filtering, smoothing, and beyond. Technical Report TRCIM1030, McGill University, 2014. [www.cim.mcgill.ca/~angqixu/pub/2TBN.TRCIM1030.pdf](http://cim.mcgill.ca/~angqixu/pub/2TBN.TRCIM1030.pdf).
- [98] A. Xu and G. Dudek. A vision-based boundary following framework for aerial vehicles. In *Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'10)*, pages 81–86, Taipei, Taiwan, 2010.

- [99] A. Xu and G. Dudek. Trust-driven interactive visual navigation for autonomous robots. In *Proceeding of the IEEE International Conference on Robotics and Automation (ICRA'12)*, pages 3922–3929, 2012.
- [100] A. Xu and G. Dudek. OPTIMo: Online probabilistic trust inference model for asymmetric human-robot collaborations. In *Proceeding of the ACM/IEEE International Conference on Human-Robot Interaction (HRI'15)*, pages 221–228, 2015.
- [101] A. Xu and G. Dudek. Towards efficient collaborations with trust-seeking adaptive robots. In *Proceeding of the ACM/IEEE International Conference on Human-Robot Interaction Extended Abstracts (HRI Pioneers'15)*, pages 221–222, 2015.
- [102] A. Xu and G. Dudek. Maintaining efficient collaborations with trust-seeking robots. In *the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'16) (to appear)*, 2016.
- [103] A. Xu and G. Dudek. *Robotics Research: The 16th International Symposium ISRR*, chapter “Towards Modeling Real-Time Trust in Asymmetric Human-Robot Collaborations”, pages 113–129. Springer International Publishing, 2016.
- [104] A. Xu, A. Kalmbach, and G. Dudek. Adaptive Parameter EXploration (APEX): Adaptation of robot autonomy from human participation. In *Proceeding of the IEEE International Conference on Robotics and Automation (ICRA '14)*, pages 3315–3322, 2014.
- [105] A. Xu, C. Viriyasuthee, and I. Rekleitis. Efficient complete coverage of a known arbitrary environment with applications to aerial operations. *Autonomous Robots*, 2013.
- [106] A. Xu, Q. Zhang, D. Meger, and G. Dudek. Interactive autonomous driving through adaptation from participation. In *Proceeding of the 10th International Conference on Intelligent Unmanned Systems (ICIUS'14)*, 2014.
- [107] R. E. Yagoda and D. J. Gillan. You want me to trust a ROBOT? the development of a human-robot interaction trust scale. *Social Robotics*, 4(3):235–248, 2012.
- [108] H. Yanco, M. Desai, J. Drury, and A. Steinfeld. *Methods for Developing Trust Models for Intelligent Systems*, pages 219–254. Springer US, 2016.
- [109] H. Yanco and J. Drury. Classifying human-robot interaction: An updated taxonomy. In *Proceeding of the IEEE International Conference on Systems, Man and Cybernetics*, pages 2841–2846, 2004.

- [110] P. Young. Optimal voting rules. *Journal of Economic Perspectives*, 9(1):51–64, 1995.
- [111] B. Yu and M. P. Singh. Distributed reputation management for electronic commerce. *Computational Intelligence*, 18:535–549, 2002.
- [112] S. Zilberstein. Using anytime algorithms in intelligent systems. *AI magazine*, 17(3):73–83, 1996.