2014-07-16

Using a Model of Temporal Latency to Improve Supervisory Control of Human-Robot Teams

Kyle Lee Blatter
Brigham Young University - Provo

Using a Model of Temporal Latency to Improve Supervisory Control of Human-Robot Teams

Kyle L. Blatter

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Michael A. Goodrich, Chair
Charles D. Knutson
Dan R. Olsen, Jr.

Department of Computer Science

Brigham Young University

July 2014

ABSTRACT

Using a Model of Temporal Latency to Improve Supervisory Control of
Human-Robot Teams

Kyle L. Blatter
Department of Computer Science, BYU
Master of Science

When humans and remote robots work together on a team, the robots always interact with a human supervisor, even if the interaction is limited to occasional reports. Distracting a human with robotic interactions doesn't pose a problem so long as the inclusion of robots increases the team's overall effectiveness. Unfortunately, increasing the supervisor's cognitive load may decrease the team's sustainable performance to the point where robotic agents are more a liability than an asset. Present approaches resolve this problem with adaptive autonomy, where a robot changes its level of autonomy based on the supervisor's cognitive load. This thesis proposes to augment adaptive autonomy by modeling temporal latency and using this model to optimally select the temporal interval between when a supervisor is informed of a pending change and when the robot makes the change. This enables robotic team members to time their actions in response to the supervisor's cognitive load. The hypothesis is confirmed in a user-study where 26 participants interacted with a simulated search-and-rescue scenario.

Keywords: Human-Robot Interaction, Temporal Latency, Human-Robot Teams, Artificial Intelligence

ACKNOWLEDGMENTS

Elise for her unwaivering support while I worked on this thesis and Hattie for giving me all the more reason to finish.

Mike Goodrich for his long-suffering aid. This research would not have been possible without his experience, knowledge, advice, and support.

Chuck Knutson for showing early interest in me and putting me on a path which lead to graduate school.

Dennis Eggett of the BYU statistics consulting center for providing evidence that I actually accomplished something.

The Army Research Lab and Brigham Young University for funding this research.

Table of Contents

List of Figures **viii**

List of Tables **ix**

1 Introduction **1**

1.1 The Problem 1

1.2 Applications 2

1.3 Related Work 2

1.4 Foundational Work 3

2 Model Description **7**

2.1 Model Overview 8

2.1.1 Actions 9

2.1.2 States 9

2.1.3 Observations 10

2.1.4 Consequences 12

2.2 Model Details 13

2.2.1 Robot Impact on Configuration 13

2.2.2 Estimating Likelihood of Agent Type 14

2.2.3 Estimating Configuration 15

2.2.4 Estimating Operational Tempo 15

2.2.5 Decision Network 16

2.3 Model Stability 26

3 Validation **29**

3.1 Simulator and Scenario 31

3.1.1 Scenario Description 32

3.1.2 Metrics 33

3.1.3 Anonymization 33

3.1.4 General Information 34

3.2 Preliminary Testing 34

3.2.1 Metrics for Comparing Regressions 34

3.2.2 Regression Characteristics 35

3.2.3 Additional Insights 36

3.3 Human-Factors Study 37

3.3.1 Experimental Procedures 38

3.3.2 Application of Model 39

3.3.3 Pilot Study 40

3.3.4 Main Study 41

4 Results **43**

4.1 Analysis Methodology 43

4.2 NASA TLX Comparisons 44

4.3 ANOVA 45

4.4 Participant Workload 46

4.4.1 Low Intensity 46

4.4.2 High Intensity 47

4.5 Participants' Effectiveness 48

4.5.1 Low Intensity 48

4.5.2 High Intensity 49

4.6 AI Involvement 49

4.7 Miscellaneous Results 50

4.7.1 Not Autonomous Robot Teams in Low and High Intensities 50
4.7.2 Map Explored 51
4.7.3 Response Time Distributions 52
4.7.4 Subjective Results 53

5 Summary and Future Work **54**
5.1 Summary 54
5.2 Future Work 55
5.2.1 The Model 55
5.2.2 Asymmetrical Classification Noise 58
5.2.3 Simulator Fidelity 58
5.2.4 New Applications 59
5.2.5 Larger Scale Study 59

A Model Stability **60**
A.1 Environment 1 60
A.2 Environment 2 62
A.3 Environment 3 64
A.4 Analysis 66

B Response-Time Regression Development **67**
B.1 Metrics for Comparing Regressions 69
B.2 Attempted Models 70
B.2.1 Linear regression over the raw data 70
B.2.2 Higher order orthogonal polynomial 70
B.2.3 Second order crossed polynomial 71
B.2.4 Second order crossed polynomial with AIC 72

C User Study Materials **73**

C.1 Participant Involvement Outline 73

D Experimental Data **76**

D.1 TLX Data 77

D.2 Civilian Data 78

D.3 Robot Performance Data 79

References **81**

List of Figures

2.1 The decision network used in estimating the utility of mission-level consequences. A decision network [40] is a generalization of a Bayesian network, where the circular nodes are Bayes nodes, the square nodes produce a range of inputs to the network, and the diamond node is the resulting utility. The input from the square nodes that maximizes utility is then used. 17

3.1 A rescue vehicle encountering a street blockage. The top-left corner holds a minimap, which depicts the environment as it has been discovered so far. The top-right corner is a camera feed from one of the aerial robots. 31

3.2 The percentage of events occurring during each minute for the preliminary phase of testing. 37

4.1 Plots of the effect of adaptive autonomy and temporal latency on four randomly-chosen user's response time distributions for high-intensity scenarios. 52

B.1 The sampled histogram of response times for all users contrasted with its corresponding exGaussian distribution. This graph was produced using the participants' response times during the preliminary phase of testing. 68

D.1 The contents of Table D.1 depicted in a bar chart. 78

List of Tables

2.1 Description of the domain, distribution, and parameters of various nodes in Figure 2.1. 18

2.2 The default and experimental values for each parameter in the model stability tests. 27

2.3 The effect on robot performance in different environments with randomly selected parameters. These were taken from Tables A.3, A.6 and A.9. 28

3.1 Types of autonomy with hypothesized cognitive load and performance. No Autonomy indicates that a robot notifies its supervisor of every individual it encounters. Adaptive Autonomy indicates that the robot decides whether to involve its supervisor in a situation or to handle the situation autonomously. Adaptive Autonomy and Temporal Latency indicates that the robot also considers delaying a notification. . . 30

3.2 Abbreviations for types of autonomy. If any of these abbreviations are described as 'high' or 'low', that refers to aerial robots operating with that type of autonomy in a high or low intensity scenario. Occasionally, the phrase "special autonomy" is used to refer to robots using either AA or AA+TL. 31

3.3 Second order crossed polynomial. This is the regression implemented for the final phase of testing in order to predict users' response times. Consult Section B.2 for a comparison of various attempted regressions. 36

3.4 The 2×3 experimental design. 38

3.5 The values of constants described in the model as applied to the scenarios in the user study. 39

3.6 The sample mean victims rescued ($\bar{x}$) and the sample standard deviation of victims rescued (s). 40

3.7 The AI involvement ratio for various types of autonomy in the pilot study. 41

4.1 The correlation between NASA TLX and AIIR. 45
4.2 The ANOVA values for the TLX scores. 45
4.3 The ANOVA values for victims rescued. 45
4.4 The NASA TLX composite scores for low-intensity scenarios for varying types of autonomy. Lower values indicate lower cognitive load. Table 3.2 for abbreviations. 46
4.5 The p-values for pairwise comparisons of TLX scores for low-intensity scenarios. . 46
4.6 The NASA TLX composite scores for high-intensity scenarios for various types of autonomy. Lower values indicate less cognitive load. Table 3.2 for abbreviations. . 47
4.7 The p-values for pairwise comparisons of TLX scores for high-intensity scenarios. 47
4.8 The p-values for pairwise comparisons of TLX scores across intensity levels. . . . 47
4.9 The mean victims rescued for low-intensity scenarios for various types of autonomy. Higher values indicate greater success. Table 3.2 for abbreviations. 48
4.10 The p-values for pairwise comparisons of victims rescued in low-intensity scenarios. 48
4.11 The mean victims rescued for high-intensity scenarios for various types of autonomy. Higher values indicate greater success. Table 3.2 for abbreviations. 49
4.12 The p-values for pairwise comparisons of victims rescued in high-intensity scenarios. 49
4.13 The AIIR values for various types of autonomy. 50
4.14 The NASA TLX composite scores for high-intensity and low-intensity scenarios non-autonomous robots. 51
4.15 The p-value for NASA TLX composite scores for high-intensity and low-intensity scenarios non-autonomous robots. 51
4.16 The correlation between AIIR and percent of map explored. 51

5.1 Summary of results. 54

A.1 The configuration of environment 1 used to calculate the data in Tables A.2 and A.3. 60

A.2 The effect on robot performance in environment 2 with independently adjusted parameters. 61
A.3 The effect on robot performance in environment 1 with randomly selected parameters. 61
A.4 The configuration of environment 2 used to calculate the data in Tables A.5 and A.6. 63
A.5 The effect on robot performance in environment 2 with independently adjusted parameters. 63
A.6 The effect on robot performance in environment 2 with randomly selected parameters. 64
A.7 The configuration of environment 3 used to calculate the data in Tables A.8 and A.9. 65
A.8 The effect on robot performance in environment 3 with independently adjusted parameters. 65
A.9 The effect on robot performance in environment 3 with randomly selected parameters. 66

B.1 Summary of each attempted regression. The regression used in the user study is highlighted in yellow. 70
B.2 Results of linear regression over the raw data. 70
B.3 Second order orthogonal polynomial results. 71
B.4 Third order orthogonal polynomial results. 71
B.5 Second order crossed polynomial results. 71
B.6 Second order crossed polynomial with AIC results. 72

D.1 Complete list of means and variances of TLX scores. Note that TLX scores range from -10 to 10, with lower values indicating lower cognitive load, lower workload, and more perceived success. Table 3.2 for abbreviations. 77
D.2 Complete list of means and variances for rescues and attacks. Table 3.2 for abbreviations. 78
D.3 The ANOVA p-values for neutrals rescued and hostile attacks. 79

D.4 Complete list of means and variances of human classifications, robot classifications, the AI Involvement Ratio (AIIR), and the percent of the map explored. Table 3.2 for abbreviations. 79
D.5 The ANOVA p-value for robot rescues, human rescues, AIIR and Map Explored. . 79
D.6 The p-values for pairwise t-test comparisons of robot classification count. 80
D.7 The p-values for pairwise t-test comparisons of human classification count. 80

Chapter 1

Introduction

1.1 The Problem

When humans and robots share a common goal, a member of the human team is often assigned to coordinate the efforts of the two groups. In virtually every case, the supervisor[1] must multitask in some regard, whether managing the efforts of multiple robots or carrying out another primary duty assigned to him or her. In either case, the supervisor's ability to make decisions drops quickly as interruptions pour in [28]. Distracting the supervisor not only makes him or her less effective, but the robots under his or her supervision are rendered less efficient in supporting the human's efforts [28].

This thesis presents and validates a model that enables robots to adapt their interactions with the supervisor based on the supervisor's inferred workload. These adaptations fall into one of two categories: adapting the *timing of interactions*, and adapting the *autonomy of interactions*.

Adapting the *timing of interactions* can be done by modeling the future effectiveness of interactions, thus enabling a robotic agent to consider postponing a notification if the supervisor is particularly busy at the moment. This enables the system to make intelligent choices regarding when to notify. *Autonomy of interaction* is defined as the level of autonomy (LOA), defined below, a robotic agent takes on when interacting with its supervisor [18].

[1] To avoid ambiguity, the human teammate tasked with directly observing and coordinating the robots is referred to as the supervisor. This is based on Jean Scholtz's roles that humans take on when interacting with robots [41], which are further described in Section 1.4 of this chapter.

This chapter describes this research's foundational work in Sections 1.3 and 1.4. The model used to make decisions about temporal latency is described in Chapter 2, and the validation the model in is described in Chapter 3. Analysis of the user studies is handled in Chapter 4, and conclusions are drawn in Chapter 5.

1.2 Applications

This research was initially funded by the Army Research Lab in order to coordinate the reconnaissance efforts of "spotter drones" with a supervisor in a nearby armored vehicle. Because every member of a tank's crew already has a primary duty (e.g., commander, driver, gunner, loader), such drones need to be sensitive to a supervisor's workload as he or she carries out his or her primary responsibilities.

In addition to military applications, this research generalizes well to civilian applications such as search and rescue, construction, healthcare, space exploration, and agriculture. In general, the proposed model is intended to enhance the effectiveness of many human-robot teams where the human takes a supervisory role by enabling robots to better coordinate with their human counterparts.

1.3 Related Work

Research into human-robot teams is extremely active and numerous approaches have been developed to facilitate human-robot cooperation. One common approach is adaptive autonomy, where robotic agents adapt their level of autonomy [43] based on the state of environment or supervisor [3]. As the state of the environment evolves, it may trigger changes in autonomy. Several aspects of environmental state have been successfully applied to the problem of adaptive autonomy[2] [12, 13, 21, 32].

[2]The term adaptive autonomy is used throughout this document to refer to systems where robots adapt their level of autonomy in response to their supervisor or the environment.

Temporal features are inherent in environmental state, but most approaches have not explicitly addressed this component in making decisions about adapting autonomy. To understand how temporal elements should influence teaming, a useful analogy is a low-level employee with an important question for his or her boss. Though the employee could attempt to resolve the question alone, the boss is in a position to give a more accurate and detailed response. Interaction with the boss is therefore highly preferable. However, when the employee stops by the boss's office to ask the question, he or she notices the boss is extremely busy.

Current approaches would have the employee choose between asking the question anyway or having the employee increase his or her autonomy by attempting to resolve the question alone. Should the boss be too busy for any type of distraction, the temporal component of the proposed approach has the employee doing what he or she can without the boss's input and attempting to contact the boss at a time when the boss is less likely to be busy. Such a system enables the employee/boss system to be constantly productive.

1.4 Foundational Work

In his seminal paper, "Human and Computer Control of Undersea Teleoperators", Sheridan describes 10 levels of autonomy which an agent may evince when interacting with a human [43]. These levels have become the foundation for most approaches that manage levels of autonomy. For further discussion regarding common applications of these levels of autonomy, consult *Human-Robot Interaction: A Survey* [18].

1. Computer offers no assistance; human does it all
2. Computer offers a complete set of action alternatives
3. Computer narrows the selection down to a few choices
4. Computer suggests a single action
5. Computer executes that action if human approves
6. Computer allows the human limited time to veto before automatic execution

7. Computer executes automatically then necessarily informs the human
8. Computer informs human after automatic execution only if human asks
9. Computer informs human after automatic execution only if it decides to
10. Computer decides everything and acts autonomously, ignoring the human

An interesting study into these levels and their consequences can be found in "The Effects of Level Automation and Adaptive Automation on Human Performance, Situation Awareness, and Workload in a Dynamic Control Task" [28]. This paper presents an important dichotomy: a highly autonomous robot distracts its operator less, while a less autonomous robot tends to be more effective due to the operator's increased influence. This research attempts to find the balancing point between autonomy and effectiveness by extending the definition of autonomy (and adaptations of the level of autonomy) to include a time element.

Though level of autonomy may help define the robot's role in interacting with a human, it only implies the human's role in responding to a robot. Scholtz establishes five roles that a human may take on when interacting with a robot [41]:

- Supervisor. The supervisor takes a high-level role, establishing the long-term goals of the human-robot team.
- Operator. The operator's task is to make sure the robot's behavior is conducive to achieving the team's goals. This can take two forms: modifying the robot's software (through training or programming) and taking direct control when the robot behaves unacceptably.
- Mechanic. The mechanic ensures that the robot's hardware is sufficient to produce the behaviors laid out by the operator.
- Peer. A robot's teammates may give the robot advice or commands which further the team's progress toward its goals, though the peer has no power to set the team's goals or the robot's behavior.
- Bystander. The bystander does not have any direct control over the robot's actions, though he or she may interact with the robot through the environment.

This taxonomy offers context for the types of interactions in which this model will be useful. Specifically, this thesis presumes a supervisory role for the human in a human-robot team.

In her 1988 paper, "Design and Evaluation for Situation Awareness Enhancement", Mica Endsley posits that a too-high level of automation disrupts a supervisor's perception and comprehension of a situation, thus hindering the supervisor's ability to project their goals on the environment [11]. More succinctly, the supervisor's situational awareness suffers. This research indicates that a robot or agent must be autonomous enough to not overburden their supervisor, but not so autonomous that the supervisor is left out-of-the-loop of an emerging situation.

The need to avoid over-automating a system as described in Endsley's paper [11] begs the question, "What should be automated?" Parasuraman et al. provide a framework to determine whether a system merits automation, and how much automation is appropriate to facilitate the supervisor's goals [36]. Specifically, Parasuraman proposes four general activities that may benefit from automation: 1—information acquisition, 2—information analysis, 3—decision and action selection, and 4—action implementation. Each activity an agent undertakes falls under one of these categories and demands a different level of autonomy. For example, an aerial robot might acquire information fully autonomously. It may perform an analysis on the data before determining whether its supervisor should be involved. Again, it may be autonomous in determining whether a percept should be forwarded to a supervisor, but the supervisor may take direct control once he or she reviews the forwarded information.

Finally, Missy Cummings presents a perspective regarding what robotic systems ought to be automated in "Automation Architecture for Single Operator, Multiple UAV[3] Command and Control" [10]. She distinguishes between autonomy for control, navigation and mission management. For example, a UAV might be fully autonomous when it comes to flight controls, but require active supervision when it comes to path planning. This thesis relies heavily on such a taxonomy in order to define the problem of adaptive autonomy, and to what systems it should be applied.

[3] Unmanned Aerial Vehicle

Numerous papers have applied adaptive autonomy to various problems in human-robot interaction. An example offered by Jacob Crandall and Michael Goodrich describes the design of a system the authors implemented which supports adaptive autonomy [8]. This provides a great case study in what factors are important and relevant in the design and construction of systems that use adaptive autonomy.

Benjamin Hardin presents an interesting extension of adaptive autonomy where rather than having the human or the agent unilaterally select a level of autonomy, the two work together to establish an appropriate level of autonomy given the environmental state (mixed-initiative autonomy) [23].

David Kaber applies the traditional concepts of level of autonomy, especially Sheridan's levels [43], to an experiment while measuring the human's situational awareness [29]. This provides a great example of how concepts such as level of autonomy and situational awareness can be brought into the real-world from theory. An interesting result in this paper was confirmed in a later paper by the same author [28] that humans experienced lower-workload but were less-effective at higher levels of autonomy.

Chen and Joyner describe a study where a tank's gunner is tasked with controlling an unmanned land vehicle while being simultaneously distracted by his or her normal duties [7]. The goal of this paper is closely aligned to that of this thesis, primarily differing in approach. Where participants in the study failed to multitask effectively due to being overwhelmed, this thesis presents a potential resolution to the problem by having robots wait to notify when the supervisor appears to be busy.

Chapter 2

Model Description

This chapter details a model for understanding how temporal latency and LOA shifts impact performance of a human-robot team. When a robot detects an event for which the supervisor's input is desired, it takes the current state of the environment (as perceived by the robot) and observations about the supervisor to produce a level of autonomy and a decision about timing.

Any model of human-robot interaction must include two components for the robot: a formalization for what the robot will and will not do in response to cues from the environment, and a set of delimiting assumptions about human behavior. The formalization must be sufficiently detailed to be implementable as an algorithm, and the assumptions about human behavior must be sufficient to improve human-robot interaction. Given these requirements, Section 2.1 of this Chapter describes how a robotic agent perceives and interacts with its environment. Section 2.2 covers specific details and touches on the implications of this model. Section 2.3 concludes this chapter with the results of an experiment regarding the model's stability across a variety of parameters.

This model's principle value added is a method to predict the optimal time at which to involve the robot's supervisor in the robot's task. Though the timing aspect of this model was designed to generalize well across many applications, this description presumes a search and rescue scenario. A human-robot team explores a cluttered urban environment in search of victims of a disaster. Some of the individuals in the environment are hostile to the rescue efforts, and will attack rescuers and robots alike. The remote robots are primarily engaged in exploring the city and classifying individuals as victim, neutral and hostile, while the human navigates the city in a rescue vehicle in an attempt to recover victims. Thus, the model presumes a few key points about its application, namely that the remote robots explore the world and classify objects and that the supervisor is distracted by other responsibilities. Classifying some of these objects entails risk to the robot and the risk is proportional to the proximity to the object.

Citations accompany all assumptions regarding robot capabilities to ensure that this model is based on the state of the art.

2.1 Model Overview

From the robot's perspective, the temporal latency problem can be framed as an optimization problem subject to the constraints and uncertainties of working with a human. To help formalize the optimization problem, consider a standard causal model for optimal decision problems. In this standard model, states of the environment and actions of the robot combine to produce consequences in the world. The ordering of preferences among consequences and the strength of these preferences are encoded as utilities.

Given a set of observations about the world, the robot wants to choose the action that maximizes the expected utility averaged over the likely states of the world [40]. Formally, this is given by

$$a^* = \arg\max_{a \in A} \sum_{s \in S} u(a,s) p_{S|O}(s|o), \tag{2.1}$$

where A is the set of actions available to the robot, S is the set of states that affect the consequences (C) produced by a given action, $u(a,s)$ is the utility of taking action a in state s, and $p_{S|O}(s|o)$ is the posterior probability distribution over states given an observation about the world.

The rest of this section presents a model of states, actions, utilities, and the posterior probabilities, which are necessary to adequately model the temporal latency problem.

2.1.1 Actions

To keep the scope of this thesis reasonable, we restrict attention to two levels of autonomy[1] for the robot [43]: the robot involves human by allowing him or her to select from a list of actions (level 2) and the robot notifies human after taking an autonomously chosen action (level 7). If the situation calls for human involvement, the robot awaits human-input while assuming a safe distance from the situation from which to gather information. Otherwise the robot resolves the situation autonomously, only notifying its supervisor after the fact.

For simplicity, the formal model is less concerned with *what* signal the robot sends to the human and more concerned with *when* the signal should be sent. Thus, the model is concerned with the amount of time between when a robot makes a decision regarding human involvement and when the human is given a generic notification; this is denoted by $w(t+\tau)$, where w denotes the notification (possibly a generic warning signal) t is the current time, and τ is some time interval into the future when the human will be given the signal. Notation from here forward replaces $t+\tau$ in the argument of w to the more simple $w(\tau)$.

Thus, the actions available to the robot at time t is encoded as $a(t) = w(\tau)$.

2.1.2 States

As mentioned in the introduction to this chapter, a robot encounters and classifies three types of individuals found in the environment: victims, neutrals and hostiles. The relevance of these individuals depends on the relative distances between the robot and these individuals as well as on the rate at which these distances are changing. The robot may signal the human supervisor, but the efficacy of this signal depends on the operational tempo[2] of the situation in which the supervisor acts.

[1] Though this model restricts action to two levels of autonomy, the levels of autonomy used and number of levels may be changed to meet the needs of an application.

[2] The term "operational tempo" is used to mean the rate at which new tasks are presented to the supervisor. Operational tempo isn't directly measured, but is inferred from the supervisor's workload, as described in Figure 2.1.

This model focuses on three key dimensions of state: *agent type* (victim, neutral, hostile), *configuration* (relative distances and velocities of the model and humans in the world), and the *operational tempo* of the environment. Agent type is relevant since the robots' primary task is classification. Configuration affects the amount of time available to the robot to resolve a situation. Operational tempo affects the supervisor's ability to focus on a robot's message. Thus, $s_t = (s_t^{\text{type}}, s_t^{\text{config}}, s_t^{\text{optempo}})$ where s^{type} indicates agent type, s^{config} indicates configuration, and s^{optempo} represents operational tempo.

The operational tempo state is general enough to be a stand in for a host of other tasks that the supervisor must do. With this generalized approach, the search and rescue becomes a socially relevant *type* for several other tasks.

2.1.3 Observations

The states in the previous section were selected to be approximately independent of each other, meaning that the types of agents encountered by the robot are independent of the operational tempo and the robot's configuration. Independence is clearly a simplification, but it is a modeling assumption that dramatically simplifies the representation of what happens in the environment. Furthermore, it enhances generalization of the model in that regardless of the definition of classification, workload and timing, the three remain approximately independent.

From a robot's perspective, it is impossible to perfectly know the state of the world. Rather, each state must be inferred from an observation[3]. It follows that associated with each dimension of state is an observation. For operational tempo, it is necessary to have an online, real-time measurement of human workload. (A discussion of workload measurement is presented in a Section 2.2.4.) For the sake of this formalism, it is assumed that such access exists. The workload measurement is a noisy approximation for the hidden operational tempo aspect of state, so the

[3]Note that any state can be inferred from observations using Bayes rule. This enhances the general applicability of this model.

model treats the relationship between operational tempo and workload as a conditional probability

$$P_{S^{\text{optempo}}|O^{\text{workload}}}(s^{\text{optempo}}|o^{\text{workload}}). \tag{2.2}$$

For agent type, the model assumes that the robot has a classifier that returns a best estimate of the type of agent encountered by the robot. The classifier is imperfect, so the relationship between agent type and classifier output is modeled as a conditional probability

$$P_{S^{\text{type}}|O^{\text{classifier}}}(s^{\text{type}}|o^{\text{classifier}}). \tag{2.3}$$

Finally, for configuration, the model assumes that the robot has the ability to (a) localize itself in some global reference frame and (b) track all agents in an agent-centered coordinate reference frame provided that the agents are within some tolerance range of the robot. Both robot localization and agent tracking is noisy, so the model treats the relationship between true configuration and observations as a conditional probability.

$$P_{S^{\text{config}}|O^{\text{environ}}}(s^{\text{config}}|o^{\text{environ}}). \tag{2.4}$$

This model combines Bayes rule with the independence assumption to take observations from the environment in order to infer the true state of the environment.

$$\begin{aligned} p_{S|O}(x|o) &= p_{S^{\text{optempo}},S^{\text{type}},S^{\text{config}}|O^{\text{workload}},O^{\text{classifier}},O^{\text{environ}}}(s^{\text{optempo}},s^{\text{type}},s^{\text{config}}|o^{\text{workload}},o^{\text{classifier}},o^{\text{environ}}) \\ &= p_{S^{\text{optempo}}|O^{\text{workload}}}(s^{\text{optempo}}|o^{\text{workload}})\,p_{S^{\text{type}}|O^{\text{classifier}}}(s^{\text{type}}|o^{\text{classifier}})\,p_{S^{\text{config}}|O^{\text{environ}}}(s^{\text{config}}|o^{\text{environ}}). \end{aligned}$$

This enables estimation of the posterior distribution of the total state vector given the total observation vector piecewise, that is, by estimating the posterior distribution for operational tempo, type, and configuration independently. The total distribution can then be constructed by taking the product.

2.1.4 Consequences

Considering the uncertainty of the environment's true state, it follows that the consequences of the robot's actions are also uncertain. This holds for any application of remote robots. Bayes rule is used in two additional ways to estimate the consequences of the robot's actions: a *Bayes filter* and a *Bayesian network*.

A *Bayes filter* is used to model how the environment changes as a result of the robot's actions [45]. Given a reasonable set of independence assumptions, which are approximately satisfied by the environment, the equations for the Bayes filter are

$$\overline{bel}(s_t) = \int p(s_t|s_{t-1},a_t)bel(s_{t-1})ds_{t-1} \tag{2.5}$$

$$bel(s_t) \propto p(o_t|s_t)\overline{bel}(s_t) \tag{2.6}$$

Applying a Bayes filter to the configuration and operational tempo elements of the state vector yields the following equations:

$$\overline{bel}_{S_t^{\text{optempo}}}(s_t^{\text{optempo}}) = \int p_{S_t^{\text{optempo}}|S_{t-1}^{\text{optempo}}}(s_t^{\text{optempo}}|s_{t-1}^{\text{optempo}},w_\tau)bel_{S_t^{\text{optempo}}}(s_{t-1}^{\text{optempo}})ds_{t-1}^{\text{optempo}} \tag{2.7}$$

$$bel_{S_t^{\text{optempo}}}(s_t^{\text{optempo}}) \propto p_{O_t^{\text{workload}}|S_t^{\text{optempo}}}(o_t^{\text{workload}}|s_t^{\text{optempo}})\overline{bel}_{S_t^{\text{optempo}}}(s_t^{\text{optempo}}) \tag{2.8}$$

$$\overline{bel}_{S_t^{\text{config}}}(s_t^{\text{config}}) = \int p_{S_t^{\text{config}}|S_{t-1}^{\text{config}}}(s_t^{\text{config}}|s_{t-1}^{\text{config}},\Delta_t)bel_{S_t^{\text{config}}}(s_{t-1}^{\text{config}})ds_{t-1}^{\text{config}} \tag{2.9}$$

$$bel_{S_t^{\text{config}}}(s_t^{\text{config}}) \propto p_{O_t^{\text{environ}}|S_t^{\text{config}}}(o_t^{\text{environ}}|s_t^{\text{config}})\overline{bel}_{S_t^{\text{config}}}(s_t^{\text{config}}) \tag{2.10}$$

Note that a Bayes filter is not applied to s^{type}, as the distribution for that random variable is assumed to remain constant. In other words, an individual's type doesn't change, and hence has no temporal component. A more sophisticated approach which adjusts the distribution of agent types as more agents have been met is left for future work.

Models for $p_{S_t^{\text{optempo}}|S_{t-1}^{\text{optempo}}}(s_t^{\text{optempo}}|s_{t-1}^{\text{optempo}}, w(\tau))$ and $p_{S_t^{\text{config}}|S_{t-1}^{\text{config}}}(s_t^{\text{config}}|s_{t-1}^{\text{config}}, \Delta_t)$ are discussed in detail in Section 2.2.

This thesis also uses a *Bayesian network* to estimate how the robot's decisions impact mission-level objectives. While the Bayes filter handles changes over time in configuration and changes in operational tempo, the Bayesian network handles how robot behavior impacts the mission. The utilities the robot uses to choose from among its available actions are calculated by estimating the mission-level consequences identified by the Bayesian network, which are the mission-level consequences produced by a given state and action combination.

Section 2.2 presents the Bayesian network for three mission-level consequences:

- When is a supervisor most likely to respond to a message?
- Will the robot be able to resolve an event satisfactorily without human intervention?
- Does the supervisor have enough time to respond to a notification before an event resolves itself?

2.2 Model Details

This section presents model details for the Bayes filter and Bayesian Network. Details specific to the user study can be found in Chapter 3.

2.2.1 Robot Impact on Configuration

The way robot movements impact world configuration is perhaps the most simple thing to be modeled, largely because so much of the literature deals with estimating the position of objects' true positions in space given noisy observations [22, 30, 31, 44]. The simulator used to validate this model includes very simple behaviors for how victims, neutrals and hostiles move in the environment. Agents behave differentially depending on their type – with some agents moving toward the robot and others moving away. In terms of configuration, the important part is how the classification task imposes time constraints or pressures on the human-robot team. For the search and rescue problem described in this thesis, relative speeds and distances impose time constraints. Thus, changes in configuration consist of changes in the relative positions and velocities of the robot and the agents. Uncertainty arises because the effect agent type has on the world's configuration is not modeled – a limitation of the independence assumption made above. As will be shown, this limitation does not negate the potential benefit of the model.

In practice, the locations and velocities of agents in the world are derived from real sensors that are fed into a motion tracking algorithm based on some variation of the Kalman filter [30, 31, 44]. The robot would then use some form of simultaneous localization and mapping to localize within a reference frame, and then track positions of agents with respect to this reference frame. The simulator used in the user study employs simple approximations of these algorithms.

2.2.2 Estimating Likelihood of Agent Type

The type of an agent is estimated using a classifier on the robot. This classifier is assumed to be imperfect but the error rates of the classifier are known. This enables the creation of a model of $p_{X|S^{\text{type}}}(x|s^{\text{type}})$, the likelihood that the classifier output is correct given a known agent type. The types of agent and the classes of agent are both drawn from the same set, {victim, neutral, hostile}. This yields the following likelihood model.

$$p_{X|S^{\text{type}}}(x|s^{\text{type}} = \text{neutral}) = \begin{cases} \beta & \text{if } x \in \{\text{neutral, hostile}\} \\ 1-2\beta & \text{if } x = \text{victim} \end{cases} \tag{2.11}$$

$$p_{X|S^{\text{type}}}(x|s^{\text{type}} = \text{hostile}) = \begin{cases} \beta & \text{if } x \in \{\text{neutral, hostile}\} \\ 1-2\beta & \text{if } x = \text{victim} \end{cases} \tag{2.12}$$

$$p_{X|S^{\text{type}}}(x|s^{\text{type}} = \text{victim}) = \begin{cases} \alpha & \text{if } x \in \{\text{neutral, hostile}\} \\ 1-2\alpha & \text{if } x = \text{victim} \end{cases} \tag{2.13}$$

Suppose the likelihood of error is less for a victim than for either a neutral or hostile, it would follow that

$$\alpha \leq \beta.$$

This assumption enables generalizing some observations from the user study to other problems where there is an asymmetry between classification categories.

2.2.3 Estimating Configuration

The robot needs to estimate its configuration from observations in the world. This thesis assumes that the robot has a 360° field of view, even if the robot only broadcasts a camera image directly in front of it. This thesis also assumes that the robot is able to accurately estimate the relative positions and relative velocities of agents in the world [16, 42, 48, 49]. There is a slight amount of error, which is modeled as a zero-mean Gaussian, yielding

$$p_{C_t|S_t^{\text{config}}}(c_t|s_t^{\text{config}}) \sim \mathcal{N}(s_t^{\text{config}}, \Sigma). \tag{2.14}$$

The configuration includes x and y locations in the world as well as v_x and v_y velocities, so noise is modeled as

$$\Sigma = \begin{bmatrix} \sigma_{p_x}^2 & 0 & 0 & 0 \\ 0 & \sigma_{p_y}^2 & 0 & 0 \\ 0 & 0 & \sigma_{v_x}^2 & 0 \\ 0 & 0 & 0 & \sigma_{v_y}^2 \end{bmatrix}.$$

meaning that position error is the same for both directions and velocity error is also the same for both directions.

2.2.4 Estimating Operational Tempo

This section provides a framework for estimating operational tempo, with implementation details for the user-study simulator filled out in Chapter 3 and Appendix B.

This thesis uses passive-polling[4] to infer operational tempo, where the observed environment at the time of a measurement provides the the supervisor's workload.

Workload is a vector of independent variables representing tasks the supervisor has to address. There is no rule regarding variable selection for the workload vector, only that a variable should be measurable and represent something which has an effect on workload. For example, the simulator and user study described in Chapter 3 populate the workload vector with three elements: discovered-but-unrescued victims, individuals currently approaching the rescue vehicle, and outstanding robot notifications. Other applications may measure other variables for this vector.

Each variable's true state is then inferred by applying a Bayes filter as described in equations 2.7 and 2.8 to each element of the workload vector to populate an operational tempo vector. The elements of the operational tempo vector represent the estimate of the true state. Section 2.2.5 describes how these values are used to predict the supervisor's response time.

2.2.5 Decision Network

Recall that the model of temporal latency considers the following three mission-level consequences:

- When is a supervisor most likely to respond to a message?
- Will the robot be able to resolve an event satisfactorily without human intervention?
- Does the supervisor have enough time to respond to a notification before an event resolves itself?

Figure 2.1 illustrates the components of the decision network (a form of Bayesian network) that models the temporal latency problem. In Bayesian networks, directed influence-lines connect nodes that represent random variables [40]. When an influence line connects node X to node Y, it indicates that node X influences node Y and that node Y's distribution is dependent on that of node X. Node Y then represents a conditional distribution over the distributions of the input nodes. In the discrete case, where all input nodes have discrete distributions, the conditional distribution can be

[4]Passive-polling is defined here to mean that the supervisor's workload is silently measured in the background. By contrast, active-polling measures the workload by either asking the supervisor directly or creating events for the supervisor to deal with and measuring his or her response.

represented by a table with one axis enumerating the cross product of the support of every input distribution and the other axis the support of the dependent node's distribution. In the case where at least one input node has a continuous random variable, however, the input nodes are used to determine the parameters of the dependent node's distribution. Nodes not connected by an influence line are said to be conditionally independent of each other.

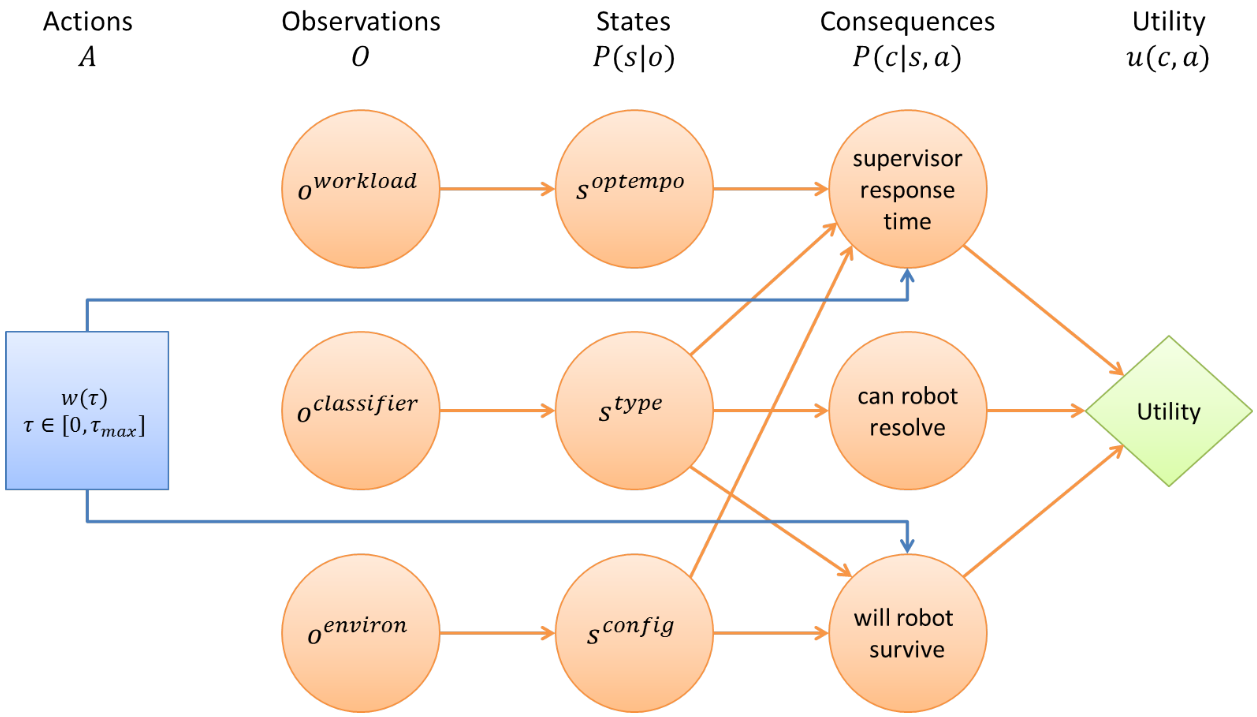

Figure 2.1: The decision network used in estimating the utility of mission-level consequences. A decision network [40] is a generalization of a Bayesian network, where the circular nodes are Bayes nodes, the square nodes produce a range of inputs to the network, and the diamond node is the resulting utility. The input from the square nodes that maximizes utility is then used.

Figure 2.1 then illustrates the conditional independence assumptions between states and consequences discussed in the remainder of this section, and it also shows the relationship between observations, states, actions and consequences, which were modeled using the Bayes filters discussed previously. Notice how the actions available to the robot only affect consequences. Actions are combined with the state of the world to estimate consequences for those actions, and it follows that the current state be independent of any future actions a robot may consider. For each consequence, the previous section implicitly identified (a) a variable that encodes that consequence and (b) a probability distribution that models consequences as a function of this variable. The variables and the domains of these variables are laid out in Table 2.1; the remainder of this section summarizes the functions of each node.

Node	Domain	Distribution/Function	Parameters
O^{workload}	$\mathbb{R}^n$	Vector of n measurements	N/A
$O^{\text{classifier}}$	$\{\text{Victim}, \text{Neutral}, \text{Hostile}\}$	Equations 2.11, 2.12, 2.13	$\alpha(x = \text{victim})$ $\beta(x \in \{\text{neutral}, \text{hostile}\})$
O^{environ}	$\mathbb{R}^6 \times O_{\text{objects}}$	Gaussian for each $x \in O_{\text{objects}}$	O_{objects}, set of observable objects $\mathbb{R}^6$ (position and velocity in $\mathbb{R}^3$) σ_p, σ_v
S^{optempo}	$\mathbb{R}^n$	Vector of n state elements	N/A
S^{type}	$\{\text{Victim}, \text{Neutral}, \text{Hostile}\}$	Non-Uniform Discrete	$P(\text{victim})$ $P(\text{neutral})$ $P(\text{hostile})$
S^{config}	$\mathbb{R}^6 \times O_{\text{objects}}$	Uniform(statespace)	O_{objects}, set of observable objects $\mathbb{R}^6$ (position and velocity in $\mathbb{R}^3$) σ_p, σ_v
supervisor response time	$[0, \infty)$	ExGaussian(μ, σ^2, λ)	μ, Mean of the Gaussian component σ^2, variance of the Gaussian component λ, rate of exponential component
can robot resolve	$(0, 1)$	Uniform	$P(\text{Resolve})$
will robot survive	$(0, 1)$	Uniform	$P(\text{Survive})$

Table 2.1: Description of the domain, distribution, and parameters of various nodes in Figure 2.1.

Can the Robot Resolve

This node answers the question, "Can the robot resolve a situation satisfactorily without involving its supervisor?" In general, whether or not the robot can resolve an event satisfactorily depends on its ability to select an appropriate action for the true state of the environment. Given the search and rescue scenario this chapter models, and assuming that a robot's classification is approximately instantaneous[5], the robot may immediately begin taking an action if it is working autonomously.

[5]In real applications, this depends heavily on the type of classification algorithm employed. Classifiers which use static images can make near-instantaneous classifications, while classifiers which employ recorded video may take substantially longer to capture and analyze the video.

Thus, the expected utility of this chosen action is dependent entirely on the accuracy of the robot's classification.

The accuracy of a classification is already calculated for node s^{type}, which in turn depends on α, β, and the prior probability of encountering each agent type ($P(\text{Victim})$, $P(\text{Neutral})$, $P(\text{Hostile})$).

For example, in a world containing predominantly victims, a robot with an extremely accurate classifier (high α and β) would be very unlikely to misclassify a victim and thus would gain little from distracting its supervisor. By contrast, a robot in a world with a roughly even distribution of agent types and a comparatively inaccurate classifier would run a high risk of misclassifying an individual and would stand to gain a great deal from involving its supervisor in the classification.

This model assumes that the accuracy of the agent type classifier depends only on the agent's type and not on the configuration of the world. This may not be a good assumption given the limitations of robot hardware (see Future Work, Section 5.2.1). Nevertheless, it is assumed that a robot will not attempt a classification until it is close enough to an individual to classify him or her with minimal error.

Figure 2.1 includes edges to the "can robot resolve" consequence node from only the "agent type" node (s^{type}). According to the semantics of Bayesian networks, these edges represent conditional independence assumptions regarding classification, namely that a robot's ability to resolve a situation only directly relies on agent type.

Supervisor Response Time

Whether or not the human is able to successfully respond to an event depends on the configuration of the world, the operational tempo of the situation, and the types of agents involved. This depends on the amount of time available to make a decision as well as the complexity of the situation. This thesis is concerned not only with whether or not the supervisor will respond, but also the time at which the response will be completed. This demands modeling the elapsed time between when the supervisor receives the signal and when the supervisor has successfully responded.

A function approximator is used to estimate the response time of a user[6]. Support data is used to train a regression that maps from the operational tempo vector, the estimated agent type, and the configuration of the environment to the user's response time. Hence, Figure 2.1 includes arrows from elements of state to the *supervisor response time node*.

For example, to implement this model for the user study, support data was generated in an initial phase of testing (see Section 3.2) where users interacted with a simulator. Whenever a robot notified the user of encountering an individual, the operational tempo vector, the type of agent the robot was interacting with, and the state of the environment at the time of notification was logged, along with the amount of time it took the supervisor to respond to the message. The regression, in this case a quadratic regression, was then performed on that data to predict user response times.

Recall from Chapter 1 that an important part of keeping the supervisor productive is to avoid overwhelming him or her [28]. As shown in Figure 2.1, supervisor response time T_{sv} is an important consequence generated by the influence diagram. This consequence impacts both whether the human will respond in time as well as the increase in the human's level of distraction. We use the distribution of response times to compute the utility of a given level of distraction as described in the remainder of this section.

Response time is often defined as the amount of time between when a signal is given and a human initiates a response [50]. Response time is modeled as an ex-Gaussian distribution because it allows modeling when the task is completed, which is a function of the amount of time required before a response begins to be generated and when the response is completed. This distribution is a "convolution of a Gaussian and an exponential distribution that has been shown to fit empirical [reaction time] distributions" [47].

Thus, each supervisor has a response time distribution unique to him or her. This distribution, which is used to compute utility later in this chapter, is estimated by having the supervisor interact with an implementation of this model for some time and sampling the supervisor's response times.

[6]This was originally modeled by directly predicting the parameters of the response time distribution based on the operational tempo, configuration of the world and estimated agent type. This proved extremely difficult to calibrate effectively, especially across different users, so this approach ended up being rejected in favor of a function approximator. See "Future Work", Section 5.2, for information regarding the original approach.

The sample mean (m), sample variance (s^2) and the sample skewness (γ_1) of the supervisor's responses are used to estimate the parameters of the user's true response distribution as follows [34]

$$\bar{\mu} = m - s(\frac{\gamma_1}{2})^{1/3} \tag{2.15}$$
$$\overline{\sigma^2} = s^2[1 - (\frac{\gamma_1}{2})^{2/3}] \tag{2.16}$$
$$\bar{v} = s(\frac{\gamma_1}{2})^{1/3} \tag{2.17}$$

where $\bar{v} = 1/\bar{\lambda}$, the inverse of the exponential rate parameter. Ratcliff suggests that at least 100 samples be gathered before these parameters can be accurately estimated [38].

The utility of a given level of distraction (LOD) can then be estimated by using the cumulative distribution function (CDF) of the supervisor's response time distribution. The CDF of a distribution represents the probability that a randomly sampled value from that distribution is less than or equal to an input value, x. Thus, CDFs map from the support of a distribution to a probability between zero and one. As x increases, it follows that the probability of getting a value less than or equal to x must stay the same or increase, making CDFs monotonically non-decreasing in nature.

Estimating a supervisor's level of distraction is performed by taking a response time and passing it into the supervisor's response time distribution CDF as x. The CDF returns a value between zero and one indicating the percentage of response times which are faster than the one passed in. To illustrate, when a supervisor has a typical response time to a notification and the CDF of the supervisor's personal distribution returns a value of 0.5, half of that supervisor's response times are faster while half are slower. In such a case, the supervisor is normally distracted and responding at a normal pace. If the CDF returns 0.9, the supervisor is clearly distracted, responding faster 90% of the time. In such a case, the supervisor is working much slower than normal and the supervisor must be reaching a point of overload where additional requests would substantially hinder the supervisor's productivity.

The CDF for an ex-Gaussian distribution is described by the following equations.

$$u = \bar{\lambda}(x-\bar{\mu}) \tag{2.18}$$

$$v = \bar{\lambda}\bar{\sigma} \tag{2.19}$$

$$\text{ExGaussianCDF}(x;\mu,\sigma,\lambda) = \Phi(u,0,v)-\exp(-u+v^2/2+\log(\Phi(u,v^2,v))) \tag{2.20}$$

where $\Phi(x,\mu,\sigma)$ is the CDF of a Gaussian. The support of an ex-Gaussian is $[0,\infty)$ meaning that the operator's response times must be greater than zero, but may be arbitrarily long. When a supervisor's response time is estimated, determining how distracted the supervisor must be is then a matter of passing the estimated response time (T_{sv}) into the ex-Gaussian CDF for that supervisor's response time distribution.

$$\text{LOD} = \text{ExGaussianCDF}(T_{\text{sv}};\mu_{\text{supervisor}},\sigma_{\text{supervisor}},\lambda_{\text{supervisor}}) \tag{2.21}$$

Thus, unusually high response times for a supervisor correspond to high levels of distraction and unusually low response times for a supervisor correspond to low levels of distraction and high utilities.

For completeness, note that this model assumes that the human can classify an agent's type perfectly if the human spends enough time on the task, something that the robot cannot do. Once the robot knows the type, this model assumes that it can resolve the situation (see the previous section).

Will Robot Survive

Properly modeling whether the robot will survive an encounter until a given time might be thought of as the amount of time available to resolve a situation. This depends heavily on the rules of engagement. An ideal model would be general enough to be useful across multiple problems with varying rules of engagement, but that is beyond the scope of this thesis. Effort has been expended for this project investigating methods using time-to-contact, but these approaches quickly became intractable, requiring a distribution derived from the combination of several random variables.

Instead, this thesis adopts a model that isn't based on the physics or rules of engagement but that affords sufficient flexibility to model a reasonable range of availability times. We assume the primary limiting factor to the available time to resolve an event is the risk to the robot of being disabled by a hostile. The distribution of time available is then the probability that the robot will survive until a given time.

To model this, it is assumed that a hostile's attack has a probability of success ($P(attack)$) based on distance to the robot (d) and that there is some sort of known 'reset' duration between hostile attacks (t_{reset}). In the context of the search and rescue scenario laid out at the beginning of this chapter, the hostiles attempt to down aerial robots by throwing rocks. Their accuracy is based on their distance to the aerial robot[7] and their reset time set to the expected reset value for the situation.

The probability of a robot surviving until time T is given by Equation (2.22).

$$P(\text{survive}|\tau,T) = (1-P(\text{attack}|d))^{((T+\tau)/t_{\text{reset}})} * P(s^{\text{type}} = \text{hostile}) + P(s^{\text{type}} \neq \text{hostile}) \quad (2.22)$$

where $P(\text{attack}|d)$ is the conditional probability of a successful attack given the distance between the robot and the individual, and $P(s^{\text{type}} = \text{hostile})$ is the robot's estimated probability that an observed individual is a hostile. The equation first estimates the probability of a robot surviving a hostile attack $(1-P(\text{attack}|d))$, and then determines the number of attacks that are expected to occur by time T with $((T+\tau)/t_{\text{reset}})$. The probability of surviving that many attacks (presuming the individual is hostile) is then calculated by raising the probability of surviving one attack to the power of the number of attacks. This value is weighted by the probability the individual is hostile. It is assumed that if the individual is not hostile, then the robot has a probability of one of surviving the encounter. The conditional probability the robot will survive if the individual is hostile, weighted by the probability the individual is hostile, is added to the probability that the individual is not hostile to determine the overall probability of surviving until time T.

[7]This is done with a simple linear accuracy function in the user study, where a fixed-size target can be hit with an accuracy of one for a distance of zero, and an accuracy of zero for some maximum throw distance. Other accuracy predictors exist and may be used in place of this simplistic approach.

Expected Utilities

The decision network gives a distribution over the mission-level consequences (response time, can the robot resolve, will the robot survive). Since each of these consequences depends on the choices made by the robot (via their impact on state), the distributions over mission-level consequences are indexed by the robot's choices.

This thesis adopts the following mission-level standards of performance. If the robot can resolve the problem alone (indicated by $P(\text{resolve} = 1)$), then signaling the supervisor has zero utility (corresponding to a finite signal time τ). If the robot cannot resolve the problem alone (indicated by $P(\text{resolve} = 0)$), then not signaling the supervisor has zero utility (corresponding to an infinite signal time τ). In practice, the robot's ability to resolve is a probability tied to its sensor error (α, β) and the distribution over agent type (s^{type}) as described above.

As described earlier in this section, the supervisor response time is also used to make an estimate of the supervisor's level of distraction (in addition to estimating his or her ability to respond in time). This is done by using the CDF of the supervisor's response time distribution and passing in the supervisor's estimated response time for a given event. The value returned was called the level of distraction (LOD). If a supervisor is distracted, his or her response times will be high and the level of distraction will return a number very close to one. By contrast, a very focused supervisor will have fast response times and a level of distraction close to zero.

Recall from the previous sections that the ranges of the consequences are defined as follows:

- Probability of robot resolving: $P_{\text{resolve}} \in (0, 1)$.
- Supervisor response time: $T_{\text{sv}} \in [0, \infty)$.
- Probability of surviving: $P_{\text{survive}} \in (0, 1)$.

Given these ranges and the mission standards described above, we can identify the following components of utility. Note that $a = (w(\tau))$ in the equations.

$$U(P_{\text{survive}}, a) = P_{\text{survive}} \tag{2.23}$$

$$U(P_{\text{resolve}}, a) = 1 - P_{\text{resolve}} \tag{2.24}$$

$$U(T_{\text{sv}}, \mu_{\text{sv}}, \sigma^2_{\text{sv}}, \lambda_{\text{sv}}, a) = 1 - \text{ExGaussianCDF}(T_{\text{sv}}; \mu_{\text{sv}}, \sigma^2_{\text{sv}}, \lambda_{\text{sv}}) \tag{2.25}$$

$$U(a) = k_{\text{delay}}(\tau + T_{\text{sv}}) \tag{2.26}$$

As described earlier in this chapter, ExGaussianCDF is the CDF of the ExGaussian response time distribution, while μ_{sv}, σ^2_{sv} and λ_{sv} are respectively the mean, variance and exponential rate of the supervisor's response time distribution. Note that the supervisor's distribution parameters are not calculated with the Bayes net, but rather with the method laid out earlier in this section.

In Equation (2.26) above, k_{delay} is a designer-specified slope of the utility and τ_{max} is maximum value of τ, that is, $\tau \in [0, \tau_{\text{max}}]$.

The overall utility is the product of these pieces:

$$U(s,a) = U(P_{\text{survive}}, a)U(P_{\text{resolve}}, a)U(T_{\text{sv}}, \mu_{\text{sv}}, \sigma^2_{\text{sv}}, \lambda_{\text{sv}}, a)U(w(\tau)). \tag{2.27}$$

The optimal notification time (a^*) for the robot is then computed using Equation (2.1). This notification time has a utility associated with it (u_{a^*}). To select a level of autonomy (Level 2 or Level 7 in the search and rescue scenario), a utility threshold, θ is used. If the utility at the notification time with maximum utility exceeds this threshold, Level 2 is selected; otherwise, Level 7 is selected.

$$\text{LOA} = \begin{cases} \text{Level 2} & \text{if } u_{a^*} > \theta, \\ \text{Level 7} & \text{otherwise} \end{cases} \tag{2.28}$$

If the robot selects Level 2, the human is notified τ seconds after the robot becomes aware of the event. Otherwise, the robot acts autonomously and notifies the human regarding its action.

2.3 Model Stability

This model might be rendered useless if it only works when its parameters are within a small tolerance of an ideal value. For that reason, a simulation of robot-interaction is performed while varying the following parameters from the Table 2.1.

- τ_{max}
- τ_{step}
- $P(\text{Victim})$
- $P(\text{Neutral})$
- $P(\text{Hostile})$
- $\eta_{\text{classifier}}$. The simulator assumes equal likelihood of classifying for each class. Thus $\alpha = \beta$. As a shorthand, these values are referred to as the classification noise, $\alpha = \beta = \eta_{\text{classifier}}$.

Note that the autonomy threshold wasn't varied, meaning the same threshold of $\theta = 0.5$ was applied to every test. The same is true for $k_{\text{delay}} = 1$, indicating no utility slope for all tests, that is, utilities in the future are considered to be as valuable as present utilities.

These parameters were chosen since they were the parameters that aren't tuned to a specific human. When these parameters are selected, the model returns the delay at which maximal utility is awarded, that level of utility, and the level of autonomy of the robot. The level of autonomy takes one of two values as described at the beginning of this chapter: the robot involves the human by allowing him or her to select from a list of actions (level 2) and robot notifies the human after taking an autonomously chosen action (level 7). For clarity, autonomy level 2 will be referred to as *limited* while autonomy level 7 will be referred to as *full* in reporting results of this test.

Because this model only works in the context of an environment, the simulation is run under three different environments. For each environment, each parameter is modified independently of the others to isolate any values which cause instability. Several trials are then run where all variables are given random (though plausible) values in order to see if there are any surprising results. The complete results of this analysis can be found in Appendix A, and a summary of interesting results is included in the remainder of this section.

The default value for each parameter is listed in Table 2.2.

Trial	$\tau_{\max}$	τ_{step}	$P(\text{Victim})$	$P(\text{Neutral})$	$P(\text{Hostile})$	$1-\eta_{\text{classifier}}$
default	10	1	0.333	0.333	0.333	0.667
high	15	5	0.667	0.667	0.667	0.833
low	5	0.5	0.167	0.167	0.167	0.5

Table 2.2: The default and experimental values for each parameter in the model stability tests.

Each of the tables reporting results (found in Appendix A and below) displays delay (τ), utility (u_a^*), and autonomy (LOA) given the variables. Delay is the amount of time the robot waits before sending a notification, if a notification is to be sent. In other words, delay is the point of time in the present or future when the robot would reap maximal utility for turning control over to the supervisor. Utility is the benefit of allowing the supervisor to take control after waiting for the delay. Autonomy is decided based on the value of the utility. If $\mathrm{u}_\mathrm{a}^* > 0.5$, control is turned over to the supervisor, otherwise the robot handles the event itself.

The goal for this test is to show robustness in the model, that is, no capricious changes in level of autonomy with varied parameters. The tests showed the model to be robust to small perturbations in its parameters. Tables A.2, A.5 and A.8 all had very stable utilities when only a single parameter was changed, indicating stability across similar sets of parameters. When numerous variables were adjusted, such as in Tables A.3, A.6 and A.9, the model was shown to adapt to untested parameters well. The model's responses weren't wild, and when LOA did change, the change could be shown to be the result of one or two meaningful parameters. Table 2.3 shows a few examples of randomly chosen parameters and the resulting utility and level of autonomy.

Example 1 in Table 2.3 had relatively high utility for its environment. This came from the extremely inaccurate classifier. Since the robot cannot be sure of the individual's classification, it was forced to defer to its supervisor. Example 2 had uncharacteristically low utility compared to the other trials, despite the utility being above the threshold for human involvement. This was due

Environment	$\tau_{\max}$	τ_{step}	$P(\text{Victim})$	$P(\text{Neutral})$	$P(\text{Hostile})$	$1-\eta_{\text{classifier}}$	Delay	Utility	Autonomy
1	9	1	0.493	0.268	0.239	0.440	4	0.5622	Limited
2	9	2	0.045	0.112	0.843	0.773	0	0.5386	Limited
3	8	4	0.119	0.019	0.862	0.957	0	0.3517	Full

Table 2.3: The effect on robot performance in different environments with randomly selected parameters. These were taken from Tables A.3, A.6 and A.9.

to the high probability of the individual being hostile, thus increasing the urgency to resolve the situation. Finally, Example 3 had the lowest utility of the trials in its environment. This was due to a combination of having a very accurate classifier and the high probability of the individual being hostile. The accurate classifier decreased the robot's need to involve its supervisor, while the high proportion of hostiles made the situation more urgent to resolve.

These results, along with the full results listed in Appendix A, confirm the robustness of the model, and offer evidence of its applicability to a diverse set of environments.

Chapter 3

Validation

This thesis makes three claims about the model that require validation:

1. A supervisor's response times given an environment can be predicted with sufficient accuracy to validate the model described in this thesis.
2. The supervisor's cognitive load will be reduced when members of the robotic team modify their behavior in response to the operational tempo. Robots will modify their behavior by either adapting Level of Autonomy (LOA) alone or adapting *both* LOA and request-timing.
3. The human-robot team will be more effective when robots adapt LOA and request-timing when compared to adapting *just* LOA (due to the supervisor's reduced cognitive load).

The above claims are hierarchical in that subsequent claims cannot be addressed until previous claims are satisfied. Since these claims regard human behavior, a human-factors study was conducted.

Claim one was addressed in a preliminary phase of testing. The validity of this project hung on having an accurate response time estimator, and as such the initial phase gathered data to calibrate different real-time metrics for estimating response time. There have been many approaches to this problem, however, the work of Scott Hudson and James Fogarty is of special note as they attempt to infer "interruptibility" based on an electronic device's sensors such as camera, microphone, and accelerometer [15] [14] [26] [13]. Interruptability and operational tempo (upon which response time is based) are both estimates of the same element of state. While interruptability assumes that a distracted individual is less interruptable, high response times indicate that an individual is distracted. Following Fogarty and Hudson's approach, response time was estimated by correlating several environmental variables with the user's performance at the task. After gathering hundreds of samples, a polynomial regression was applied that generalized well and had an acceptably high R^2 value of .572, indicating a reasonably good fit for the data. The methodology and results for this phase of testing are described in greater detail in Section 3.2.

Once the first claim was validated and response times could be estimated with reasonable accuracy, claims two and three were validated in a second phase of human-factors testing. Users interacted with the same simulator from the first phase, with the principal difference being that the robots were able to estimate operational tempo and act in response to it. To show the model's effectiveness, this phase of testing was divided into six individual trials with three levels of robot sensitivity to the operational tempo, each tested on a low-intensity and high-intensity scenario. Table 3.1 summarizes users' hypothesized performance.

Autonomy Type	Cognitive load	Performance
No Autonomy	High	Low
Adaptive Autonomy	Medium	Medium
Adaptive Autonomy and Temporal Latency	Low	High

Table 3.1: Types of autonomy with hypothesized cognitive load and performance. No Autonomy indicates that a robot notifies its supervisor of every individual it encounters. Adaptive Autonomy indicates that the robot decides whether to involve its supervisor in a situation or to handle the situation autonomously. Adaptive Autonomy and Temporal Latency indicates that the robot also considers delaying a notification.

The null hypothesis in this experiment is that the objective performance of the human-robot team and the supervisor's cognitive load will remain constant regardless of whether the model is implemented. Ideally, however, using adaptive autonomy will yield a modest improvement over no autonomy, while adaptive autonomy coupled with temporal latency will produce the greatest improvement. The data gathered during this phase of testing confirms the second and third claims laid out at the beginning of this chapter along with the hypotheses in Table 3.1. Section 3.3 contains an in-depth description of the study's methodology and Chapter 4 presents an analysis of the results.

Abbreviations For brevity in discussing the various types of autonomy, the abbreviations laid out in Table 3.2 are used throughout the remainder of this document, especially when reporting results.

Autonomy Type	Abbreviation
No Autonomy	NA
Adaptive Autonomy	AA
Adaptive Autonomy and Temporal Latency	AA+TL

Table 3.2: Abbreviations for types of autonomy. If any of these abbreviations are described as ‘high’ or ‘low’, that refers to aerial robots operating with that type of autonomy in a high or low intensity scenario. Occasionally, the phrase “special autonomy” is used to refer to robots using either AA or AA+TL.

3.1 Simulator and Scenario

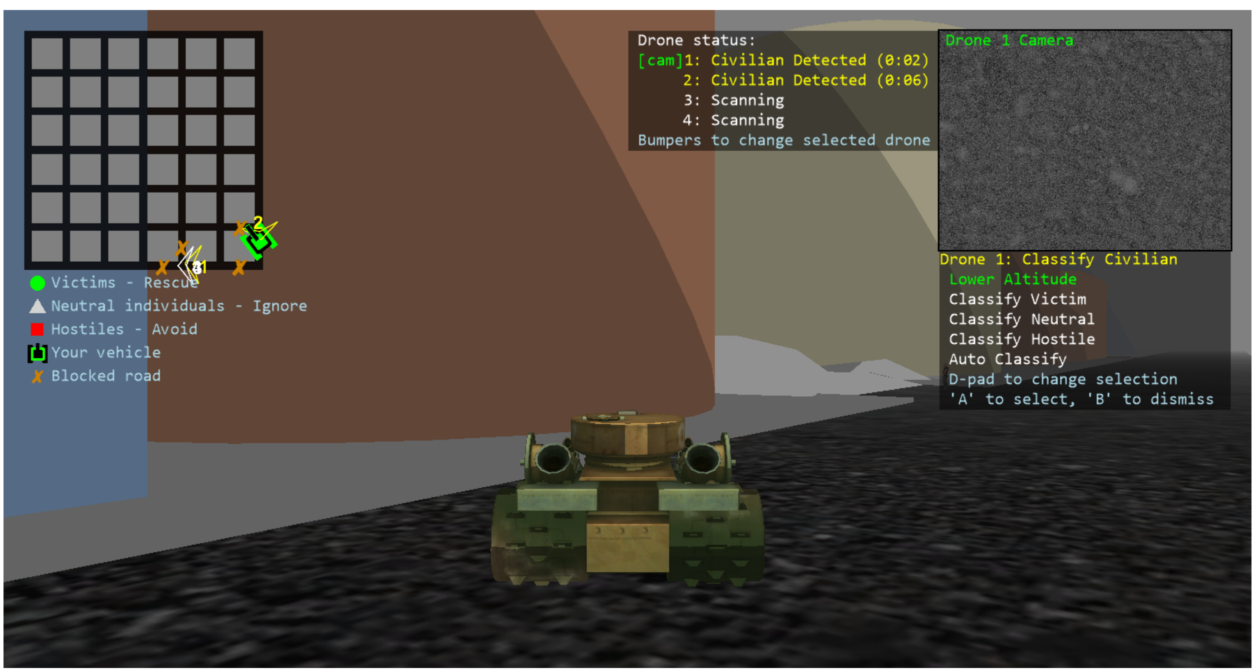

Figure 3.1: A rescue vehicle encountering a street blockage. The top-left corner holds a minimap, which depicts the environment as it has been discovered so far. The top-right corner is a camera feed from one of the aerial robots.

A simulator capable of presenting a search and rescue scenario was developed for this thesis to facilitate testing. The user assumed the role of the supervisor stationed in a rescue vehicle with a team of aerial robots at the user's disposal.

3.1.1 Scenario Description

The simulator presents the user with an earthquake-stricken city in which the user drives a rescue vehicle while managing one or more aerial robots. Though the user has access to a map of the city's streets, several streets are rendered impassable by rubble. Such blockages are unknown until discovered by the user or an aerial robot. The user is tasked with finding victims in need of medical attention. Complicating this task is the presence of other classes of individuals in the city: neutrals and hostiles. Neutrals don't need immediate assistance, while hostiles actively thwart rescue efforts by attacking the supervisor's vehicle or the robots. Such an attack, if successful, disables the attacked vehicle for a reset time of 15 seconds. Depending on the type of autonomy being tested, the aerial robots perform an analysis of the situation when they encounter an individual and may request classification assistance from the supervisor.

The supervisor interacts with a robot's request via a text interface that enables him or her to respond with instructions, defer the request to later, or ignore the request and have the robot use its own judgment to resolve the situation. When responding to a request, the supervisor may view a portion of what the robot sees through its camera. The supervisor is presented with several possible actions to resolve the situation. For example, when a robot is approached by an individual, it will prompt the user to classify the individual as a victim, neutral, or hostile, while also allowing the user to lower the aerial robot's altitude (to get a better look at the individual) or dismiss the request.

The simulator was designed to create a non-homogenous distribution of operational tempo across a scenario's duration. For example, the primary task of navigating the rescue vehicle through the city and finding victims to rescue was designed to induce a low – but constant – cognitive load in the supervisor. Encounters with hostile individuals, on the other hand, induce high cognitive load in the user that only lasts for a matter of seconds. This complicates the problem of estimating the operational tempo at a specific moment, but enables a richer mix of levels of cognitive load a user might encounter in a real-world application of this research.

3.1.2 Metrics

Since the simulator emulates search and rescue, the most important metrics are the number of victims discovered within a time-limit and the user's subject responses to the NASA-TLX subjective workload after each mission. This yields estimates of the simulation's success with regard to the hypotheses listed in Table 3.1 with the number of victims corresponding to performance and the TLX scores corresponding to cognitive load.

Other metrics are gathered as well:

- Percent of map explored
- Hostile attacks
- Neutrals rescued
- AI classifications
- User classifications
- AI involvement ratio (calculated by dividing AI classifications by total classifications)
- Mean of the response time distribution's Gaussian component
- Variance of the response time distribution's Gaussian component
- Lambda of response time distribution's exponential component

Other metrics were calculated by analyzing data stored in each scenario's log files, but the above represent the most useful measures in analyzing users' performance.

3.1.3 Anonymization

The simulator supports anonymization by default. Before a user is allowed to run the simulator, the user must enter a username of their choosing. The username is encrypted using the SHA-1 algorithm, and all of the user's results, log files, and personal data are stored in a folder named after their encrypted user name.

3.1.4 General Information

The simulator was developed in C♯ using the XNA framework produced by Microsoft. It supports square cities of arbitrary dimension. Any number of aerial robots, victims, hostiles and neutrals may be inserted into the city. Street blockages are randomly generated using an algorithm that guarantees that every intersection in the city is accessible from every other intersection, thus precluding the existence of an inaccessible section of a city.

Aerial robots navigate by randomly selecting a new direction every time they reach an intersection. Unvisited roads are weighted more heavily than visited roads. Since the aerial robots fly, they ignore blockages as they navigate the map.

The user interacts with the simulator via an Xbox 360 controller. This controller was chosen as it simplifies the interface for users while still allowing for a rich set of controls.

3.2 Preliminary Testing

This phase of testing was concerned with generating support data in order to create a regressor for estimating a user's response time given the configuration of the environment. Volunteers were asked to interact with the simulator, doing 15 minute long scenarios where they were tasked with responding to robot requests while seeking out victims. Unlike the full study, the robots weren't allowed any measure of autonomy and they always alerted the supervisor whenever they encountered a person.

For a complete treatment of how data was generated and utilized to create a regressor, refer to Appendix B. This section reports on the regression that was used in the full user-test, while providing enough background information to understand the results.

3.2.1 Metrics for Comparing Regressions

Two metrics were optimized as we explored different models for regression:

- R^2 is the goodness of fit for a regressor. This value ranges from 0 to 1 and is to be maximized.

- Standard error is the standard deviation of the regression's residuals. This value is greater than 0, and it is to be minimized. The standard error on 10-fold cross-validation measures the degree to which the regressor overfits the support data. This metric is abbreviated as *10X Standard Error* in reporting the results.

In order to compare the regressions reported in this appendix, two metrics are used. The first one is the standard error, which represents the standard deviation of the regressor's residuals. This is easy to calculate since most regression packages report the mean squared error, which is the variance of the residuals. Taking the square root of the mean squared error then yields standard error. The standard error was chosen over mean squared error simply because standard error offers an intuitive idea of the spread of the residuals and the accuracy of the regression.

It is difficult to know exactly how low standard error needs to be in order to make a usable system. Instead, the analysis of this metric is characterized as an optimization problem where the goal is to get it as close to ideal as possible. Standard error should be minimized, as smaller standard error corresponds to more accurate regressions.

R^2 is used as a measure of goodness of fit. R^2 ranges from 0 to 1, where higher values indicate a better fit for a regressor over a set of data points. Acceptable values of R^2 vary significantly between applications, making it difficult to determine how high R^2 needs to be. As with standard error, R^2 is treated as a value to optimize (albeit with maximization in this case).

Finally, a regression's ability to generalize is measured by using 10-fold cross-validation. The standard error increasing substantially in cross-validation over the non-cross-validated regression indicates that the regression overfits the data. This metric will be shortened to *10x Standard Error* when reporting data.

3.2.2 Regression Characteristics

Among the regression functions tested, the most simple regressor with a high R^2 value was a second-order polynomial regression, which produced the results summarized in Table 3.3. Its high value for R^2 indicates (a) a strong correlation between the environment and users' response times and (b) that this regression represents the relationship well. Its low error on 10-fold cross-validation compared to its not-cross-validated error indicates that the regression generalizes well to unseen users and environmental configurations.

R^2	Standard Error	10X Standard Error
0.572	5.130	5.595

Table 3.3: Second order crossed polynomial. This is the regression implemented for the final phase of testing in order to predict users' response times. Consult Section B.2 for a comparison of various attempted regressions.

3.2.3 Additional Insights

In addition to generating support data, which proved useful in predicting response times, this phase of testing acted as a pilot study and yielded insight into the logistics of a full study.

One insight was that the fifteen minute simulation time was far too long for most users, the principal reason being the relatively small maps they interacted with – most users had fully explored the map within ten minutes. This meant that anybody who could be discovered was discovered in that amount of time, and the remaining five minutes or so were spent driving around the map picking up victims without the additional cognitive load induced by robot interactions. After examining the evolution of operational tempo across time, five minutes seemed to be sufficient time to induce cognitive load as 53% of all events happened during that time. Figure 3.2 shows the distribution of events across a 15 minute simulation.

Furthermore, the number of people per block needed to be increased. This phase of testing had a population density of one individual per block. This proved to be too slow-paced to induce significant workload. Further experiments into population densities yielded a density of 1.33 individuals per block. This value seems to consistently induce cognitive load without overwhelming the supervisor.

Participants indicated that some UI changes were necessary. In response to that, icons for victims, neutrals and hostiles were changed to different shapes on the map to make them more

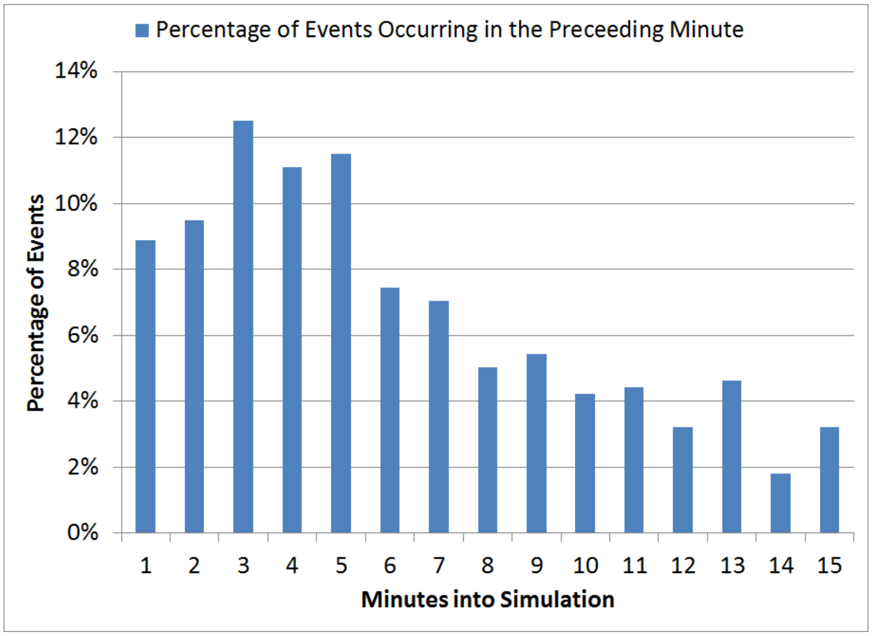

Figure 3.2: The percentage of events occurring during each minute for the preliminary phase of testing.

visually distinct. Keys and widget-descriptions (see Figure 3.1) were added to the screen so users would know at a glance the function of various parts of the interface.

3.3 Human-Factors Study

This section describes the design of the human-factors study that was carried out, especially in the context of why each component of the study was necessary for validating the model. For an analysis of this study's results, consult Chapter 4.

This study was carried out under BYU's Institutional Review Board's approval of a larger project entitled *Cyber Teams: Virtual and Haptic Interactions between Multiple Operators and Multiple Robots*. Volunteers were compensated $12 for one hour of their participation. Advertisement was carried out via social media, invitations from instructors to their classes, and public postings.

3.3.1 Experimental Procedures

For this study, participants were asked to interact with the simulator for a series of six scenarios. These scenarios followed a 2×3 experimental structure where the three types of autonomy (see Table 3.1) were tested for low and high intensity scenarios. Thus, the six scenarios were as depicted in Table 3.4.

	NA	AA	AA+TL
Low Intensity	Low NA	Low AA	Low AA+TL
High Intensity	High NA	High AA	High AA+TL

Table 3.4: The 2×3 experimental design.

A low-intensity scenario is characterized by a just-high-enough pace to keep the user engaged, but not so high that the user is overwhelmed or frustrated. This shares some characteristics with the psychological state known as flow [9], where users are challenged, but are capable of overcoming their challenges.

High-intensity scenarios, on the other hand, are intended to induce significant cognitive load. Though this can be accomplished easily by assigning a user a needlessly large robot team, having too many aerial robots may make it so that the model will be ineffective for the user. Thus, the ideal high-intensity scenario has enough robots that the robots overwhelm the user when they notify the user of every event, while adaptive autonomy puts a user in a workload state similar to the low-intensity scenario.

In preliminary testing, users carried out scenarios with robot teams ranging in size from one to six robots. Based on the users' debriefing, two robots seemed to be a good fit for a low intensity scenario, while four robots seemed ideal for high intensity.

In order to ensure users were all at approximately the same skill-level while using the simulator, every user went through 15 minutes of training and practice. The first five minutes consisted of an automated tutorial where users were introduced to the controls and UI elements one at a time, and given a chance to practice using each. Once the tutorial was finished, users were given practice interacting with the simulator without the model (i.e. robots notify the user for every event) for a low-intensity and a high-intensity scenario. These practices were for learning purposes only and were not included as part of the experiment or its results.

Ideally, the users would have learned how to interact with the simulator, thus mitigating any learning bias going forward. Nevertheless, after training, the experiment's scenarios were presented in random order to the users to mitigate the effect learning might have on the experimental outcomes.

3.3.2 Application of Model

To implement the model described in Chapter 3.5, the constants in Table 3.5 were used on every scenario.

Constant	Value
τ_{max}	15
τ_{step}	1
k_{delay}	0.99
θ	0.5
α	0.667
β	0.667
$P(\text{Victim})$	0.333
$P(\text{Neutral})$	0.333
$P(\text{Hostile})$	0.333

Table 3.5: The values of constants described in the model as applied to the scenarios in the user study.

This research requires a personal response time distribution for each user, something which was not available for users before they participated in the study. To bootstrap the process of estimating a user's distribution, the user's response time parameters from the training scenarios were used. This was a reasonable approach as the parameter values appeared to converge quickly, with only small differences between the training values and the full simulation values.

3.3.3 Pilot Study

A pilot study was carried out to test the experimental procedures, ensure the model's correct implementation, and generate some example data to estimate the number of participants required for statistical significance on the full study. Three volunteers participated in the pilot study. The pilot study and the larger-scale user study followed the same experimental procedures, except where otherwise noted in this section.

Results

Autonomy Type (Table 3.2 for abbreviations)	Victims Rescued $\bar{x}$	s
NA	12.833	2.317
AA	7.833	2.858
AA+TL	9.167	2.483

Table 3.6: The sample mean victims rescued ($\bar{x}$) and the sample standard deviation of victims rescued (s).

Table 3.6 summarizes the results from the pilot study. The sample means don't signify much owing to the small sample size. The standard deviation, on the other hand, offered a coarse estimate of the population's standard deviation, useful in estimating a required sample size. For the sake of a statistical power analysis[1], a standard deviation of 2.5 in the number of victims rescued was assumed. Using $\alpha = .05, \beta = .2$ as parameters, 19 participants would be required to resolve a difference of at least 2 in number of victims rescued. This was within the tolerance of the expected results, so a user study of 20 participants was scheduled.

The pilot study indicated that the average user finishes the experiment in approximately 55 minutes. This was under the one hour time limit imposed by this study's IRB approval, so there was no need to modify the length of the experiment. There was never any confusion about what each participant ought to be doing, and there was always time for questions should the participant have any uncertainty regarding his or her task.

[1]Special thanks to Dennis Eggett in the BYU Statistical Consulting Center for helping in this.

The pilot study indicated that a more uniform training experience needed to be offered to each participant. To this end, experiment proctors followed an outline while administering the experiment to participants (included in Appendix C). Participants were also given an automated tutorial for learning how to interact with the simulator.

Perhaps the most salient insight offered by this pilot study, however, was uncovering a severe bug that would have skewed all of the results in the user-study. This was made clear through use of a metric referred to as the AI Involvement Ratio (AIIR). The AIIR is the number classifications done automatically by the aerial robots divided by all classifications performed during a scenario. If AIIR = 1, then all classifications are being done by robots while AIIR = 0 indicates all classifications being performed by the supervisor.

Autonomy Type (Table 3.2 for abbreviations)	AI Involvement Ratio
NA	.039
AA	.081
AA+TL	.055

Table 3.7: The AI involvement ratio for various types of autonomy in the pilot study.

Table 3.7 shows the average AIIR for participants in the pilot study. Notice how it remains relatively low and constant for all simulations regardless of type of autonomy. This is an indication of a bug in the model, as both AA and AA+TL should have AIIR values substantially greater than NA, probably greater than .5. As it turns out, the low values for AIIR were the result of an error in the workload formula, causing the model to significantly underestimate a user's response time and therefore level of distraction.

3.3.4 Main Study

The full results and analysis methodology for this study can be found in Chapter 4. In short, the study was a success. It confirmed the hypotheses for high-intensity scenarios, showing that temporal latency provided a substantial improvement over standard adaptive autonomy and no autonomy. Interestingly, low intensity situations failed to find a significant difference, due to the lessened need for robot intervention when the supervisor is not required to work at high workloads (the supervisor has spare capacity).

Interest was higher than expected, for the study, and 26 volunteers ended up participating. This was an unexpected boon when it became apparent that more than the original 19 participants were required for statistically significant results.

Chapter 4

Results

This chapter summarizes the important and interesting results of the user study. For a comprehensive list of results, consult Appendix D. For a description of the study's design and methodology, consult Section 3.3. Finally, though this chapter analyzes the study's results, conclusions of the study are found in Chapter 5.

4.1 Analysis Methodology

The two primary goals of this study were to collect evidence that sensitivity to temporal latency (1) eases the supervisor's workload while (2) enhancing the team's effectiveness. Both of these should be true when compared to both the control approach (the robots are not autonomous) and the predominant approach (the robots adapt their autonomy based only on operational tempo).

The NASA TLX results gathered after every scenario were used as an estimate of cognitive load. To ensure the validity of using NASA TLX, its results are correlated with the AI involvement ratio[1]. This offered a sanity check that NASA TLX is useful as a workload metric. As for the team's effectiveness, the primary goal participants were asked to accomplish was rescuing as many victims as possible. As such, the primary measure for team-success is the number of victims rescued during the scenario.

[1] The AI involvement ratio was used here because assuming a constant amount of classifications across each scenario, the more involved the AI is the less work the human has to do. It follows that a high AI involvement ratio corresponds to a lower workload for a user and a low AI involvement ratio corresponds to a higher workload.

The first step in validating the success of this experiment is establishing that teams using AA aerial robots are more effective than teams using NA aerial robots. This is important, because any well-implemented adaptive autonomy system ought to be able to beat robots with no special autonomy model. Failing to establish this indicates a flaw in the experiment's methodology.

The next comparison necessary for the success of this experiment is between teams with non-autonomous aerial robots and teams with robots incorporating adaptive autonomy and temporal latency. Establishing a difference here demonstrates that the implementation of the model described in the beginning of this thesis is better than non-autonomous aerial robots.

The final comparison is between teams using just adaptive autonomy and teams using both adaptive autonomy and temporal latency. If teams incorporating temporal latency are shown to be more effective than the prevailing approach of adaptive autonomy, then this thesis is validated.

The experiment uses a within subjects, repeated measures design. An ANOVA is used to identify when statistically significant comparisons exist, while the one-tailed, two-sample T-test shows which comparisons are significant. Results are statistically significant if $p \leq \alpha$. For the purpose of this analysis, the significance level is $\alpha = .05$. Significant p-values are bolded in the results tables in this chapter.

4.2 NASA TLX Comparisons

Owing to its subjective nature and its critical role in analysis of the results, the validity of NASA TLX needs to be confirmed by showing its correlation to another workload metric. This is done before any other analysis, as NASA TLX is used extensively throughout the remainder of this chapter. The AI Involvement Ratio (AIIR) is used in this case, see Section 4.6 for a detailed explanation of AIIR. The correlation coefficient, R, represents the correlation between two variables and can range between -1 and 1. The magnitude of R represents the strength of correlation, while the sign of R represents the direction of correlation. Table 4.1 summarizes the correlation between these metrics, overall and for high and low intensity scenarios.

Scenario Intensity	R
All	-0.919
Low	-0.960
High	-0.996

Table 4.1: The correlation between NASA TLX and AIIR.

In every case, $R < -.9$. This indicates closely and negatively correlated variables. This stands to reason, as a higher workload corresponds to a higher TLX score and a lower level of AI involvement.

4.3 ANOVA

Because of inflation (the T-test's tendency to increase the probability of a type-I error when repeated), an ANOVA is used to ensure statistically significant relationships occur within the data. Once confirmed, T-tests can be used to test binary relationships in the data for significance. This section uses a two factor ANOVA to compare high and low intensity scenarios for the three levels of treatment received. This is done separately for both workload and victims rescued.

Source	SS	df	MS	F	P
Rows (Scenario Intensity)	797.95	1	797.95	26.54	$<$ **0.0001**
Columns (Autonomy)	866.68	2	433.34	14.42	$<$ **0.0001**
r x c	939.81	2	469.91	15.63	$<$ **0.0001**
Error	4509.24	150	30.06		
Total	7113.68	155			

Table 4.2: The ANOVA values for the TLX scores.

Source	SS	df	MS	F	P
Rows (Scenario Intensity)	53.27	1	53.27	5.64	**0.0188**
Columns (Autonomy)	220.07	2	110.04	11.64	$<$ **0.0001**
r x c	72.01	2	36.01	3.81	**0.0243**
Error	1417.56	150	9.45		
Total	1762.91	155			

Table 4.3: The ANOVA values for victims rescued.

As Tables 4.2 and 4.3 confirm, there were significant differences between every tested group.

4.4 Participant Workload

The general results for level of workload was that robots with special autonomy tend to significantly decrease user workload.

4.4.1 Low Intensity

Autonomy Type	Composite TLX	
	$\bar{x}$	s
NA	-0.617	6.334
AA	-4.080	5.392
AA+TL	-4.327	4.571

Table 4.4: The NASA TLX composite scores for low-intensity scenarios for varying types of autonomy. Lower values indicate lower cognitive load. Table 3.2 for abbreviations.

Comparison	p-value
NA $>$ AA	**0.019**
NA $>$ AA $+$ TL	**0.010**
AA $>$ AA $+$ TL	0.430

Table 4.5: The p-values for pairwise comparisons of TLX scores for low-intensity scenarios.

Table 4.4 demonstrates that the means follow what is expected, with participants experiencing less cognitive load when robots had special autonomy. Indeed, the p-values for most comparisons support that. However, the comparison between AA and AA+TL fails to find a significant difference and thus cannot reject the hypothesis that $AA < AA + TL$. Nevertheless, robots with special autonomy clearly reduce participants' cognitive load compared to no autonomy.

4.4.2 High Intensity

Autonomy Type	Composite TLX	
	$\bar{x}$	s
NA	5.133	5.834
AA	-3.668	7.293
AA+TL	-1.583	6.858

Table 4.6: The NASA TLX composite scores for high-intensity scenarios for various types of autonomy. Lower values indicate less cognitive load. Table 3.2 for abbreviations.

Comparison	p-value
NA $>$ AA	$<$ **0.001**
NA $>$ AA $+$ TL	$<$ **0.001**
AA $>$ AA $+$ TL	0.853

Table 4.7: The p-values for pairwise comparisons of TLX scores for high-intensity scenarios.

Table 4.6 shows that greater autonomy tends to induce less workload in participants. The same conclusion can be drawn from low-intensity scenarios. There was no significant difference between adaptive autonomy and adaptive autonomy with temporal latency (Table 4.7).

Comparison	p-value
low NA $\neq$ high NA	**0.001**
low AA $\neq$ high AA	0.818
low AA $+$ TL $\neq$ high AA $+$ TL	0.097

Table 4.8: The p-values for pairwise comparisons of TLX scores across intensity levels.

An interesting result is that while cognitive load was much higher for NA teams under the high intensity condition than under the low intensity condition, having robots with special autonomy seems to mitigate the effect. As Table 4.8 shows, there was no significant difference in workload for teams containing robots with some form of adaptive autonomy (though there was a marginally significant difference for AA+TL teams). In other words, regardless of the intensity of the scenario, some special autonomy in the robots kept their supervisors at a consistent workload.

4.5 Participants' Effectiveness

4.5.1 Low Intensity

Autonomy Type	Victims Rescued	
	$\bar{x}$	s
NA	7.544	2.751
AA	10.194	3.486
AA+TL	7.784	3.040

Table 4.9: The mean victims rescued for low-intensity scenarios for various types of autonomy. Higher values indicate greater success. Table 3.2 for abbreviations.

Comparison	p-value
NA $<$ AA	**0.002**
NA $<$ AA $+$ TL	0.383
AA $<$ AA $+$ TL	**0.995**

Table 4.10: The p-values for pairwise comparisons of victims rescued in low-intensity scenarios.

Tables 4.9 and 4.10 show some extremely interesting results. In low-intensity scenarios, teams with adaptive autonomy and temporal latency were no different in effectiveness from non-autonomous teams. Both were far less effective than teams with adaptive autonomy.

This seems to arise from the nature of low-intensity scenarios. In such a scenario, the supervisor is rarely over-burdened and can usually respond quickly to messages. Having a robot wait for any period of time before sending off a message unnecessarily lengthens the response time. This makes the team far less effective than it might have been.

4.5.2 High Intensity

Autonomy Type	Victims Rescued	
	$\bar{x}$	s
NA	6.182	2.991
AA	8.914	3.088
AA+TL	10.495	3.633

Table 4.11: The mean victims rescued for high-intensity scenarios for various types of autonomy. Higher values indicate greater success. Table 3.2 for abbreviations.

Comparison	p-value
NA $<$ AA	**0.001**
NA $<$ AA $+$ TL	$<$ **0.001**
AA $<$ AA $+$ TL	**0.049**

Table 4.12: The p-values for pairwise comparisons of victims rescued in high-intensity scenarios.

The increasing number of victims rescued with AA and then AA+TL under high intensity (see Table 4.11) hints that implementing AA and then AA+TL correlates with greater success. Table 4.12 confirms this. This is exciting because greater productivity with temporal latency is the crux of this thesis. When supervisors are about to become overwhelmed, the robots manage to effectively delay messages to a point in time when the supervisor can effectively manage them.

4.6 AI Involvement

A useful metric in understanding the activity of the AI was the AI Involvement Ratio (AIIR). This represents the number of user classifications performed by the robots divided by the total number of classifications. Thus, if AIIR $= 0$, the human did all of the classification, and if AIIR $= 1$ the robots did all of the classification.

Autonomy Type	Average AIIR
NA All	0.034
AA All	0.635
AA+TL All	0.521

Table 4.13: The AIIR values for various types of autonomy.

Table 4.13 summarizes the AI involvement for the different types of autonomy. Notice that AIIR $\neq 0$ for non-autonomous teams. Though the aerial robots never initiate a classification on their own, users may dismiss robot requests, forcing robots to perform classifications. Apparently this happened about 3.4% of the time.

Next, notice that teams with adaptive autonomy had the highest level of AIIR, while teams with adaptive autonomy and temporal latency were about 10% less likely to have the aerial robots intervene. AA+TL robots consider auto-classifying just as often as AA robots. The difference is that in those cases, AA+TL have the additional option to defer sending the request until later, which they apparently did about $63.5\% - 52.1\% = 11.4\%$ of the time. This actually decreases TLX workload slightly, from -2.995 to -3.874. However, these values are not significantly different, as the p-value for $AA \neq AA + TL$ is .602. This is an extremely interesting result; it indicates that even despite the lower AI involvement when using adaptive autonomy with temporal latency, there is no significant difference in cognitive load.

4.7 Miscellaneous Results

This section contains interesting, albeit not critical statistics to supporting this study's hypotheses.

4.7.1 Not Autonomous Robot Teams in Low and High Intensities

It is interesting to compare results for teams in low and high intensity scenarios with non autonomous robots. Table 4.14 illustrates the comparison.

The significant difference in workload confirms our subjective classification of low and high intensity scenarios, as described in the Experimental Procedures section (Section 3.3.1). When the

Scenario Intensity	Composite TLX	
	$\bar{x}$	s
Low	-0.617	6.334
High	5.133	5.834

Table 4.14: The NASA TLX composite scores for high-intensity and low-intensity scenarios non-autonomous robots.

sole responsibility for classification fell to the supervisor, participants reported significantly higher cognitive load in high-intensity situations.

Scenario Intensity	Composite TLX p-value
High $>$ Low	$<$ **0.001**

Table 4.15: The p-value for NASA TLX composite scores for high-intensity and low-intensity scenarios non-autonomous robots.

4.7.2 Map Explored

Surprisingly, lower participant workloads correlated with higher percentages of the map explored. The reason for this is that robots spend a significant amount of time waiting for their supervisor to respond. While waiting, the robots aren't exploring. Thus, when the robots take up more of the classification workload (which lowers the supervisor's workload), more of the map is explored. Table 4.16 summarizes the correlation between map explored and AIIR. The table reports a very strong positive correlation between robot involvement and percentage of the map explored. Note that the table reports R, which measures correlation as described above in Section 4.2.

Scenario Intensity	R
All	0.735
Low	0.999
High	0.989

Table 4.16: The correlation between AIIR and percent of map explored.

4.7.3 Response Time Distributions

When the robots are correctly calibrated, they shift a supervisor's response time distribution closer to zero. The reason for this is that the supervisor isn't presented with notifications when the supervisor is likely to take a long time responding to the message. Thus, the supervisor's responses tend to be faster.

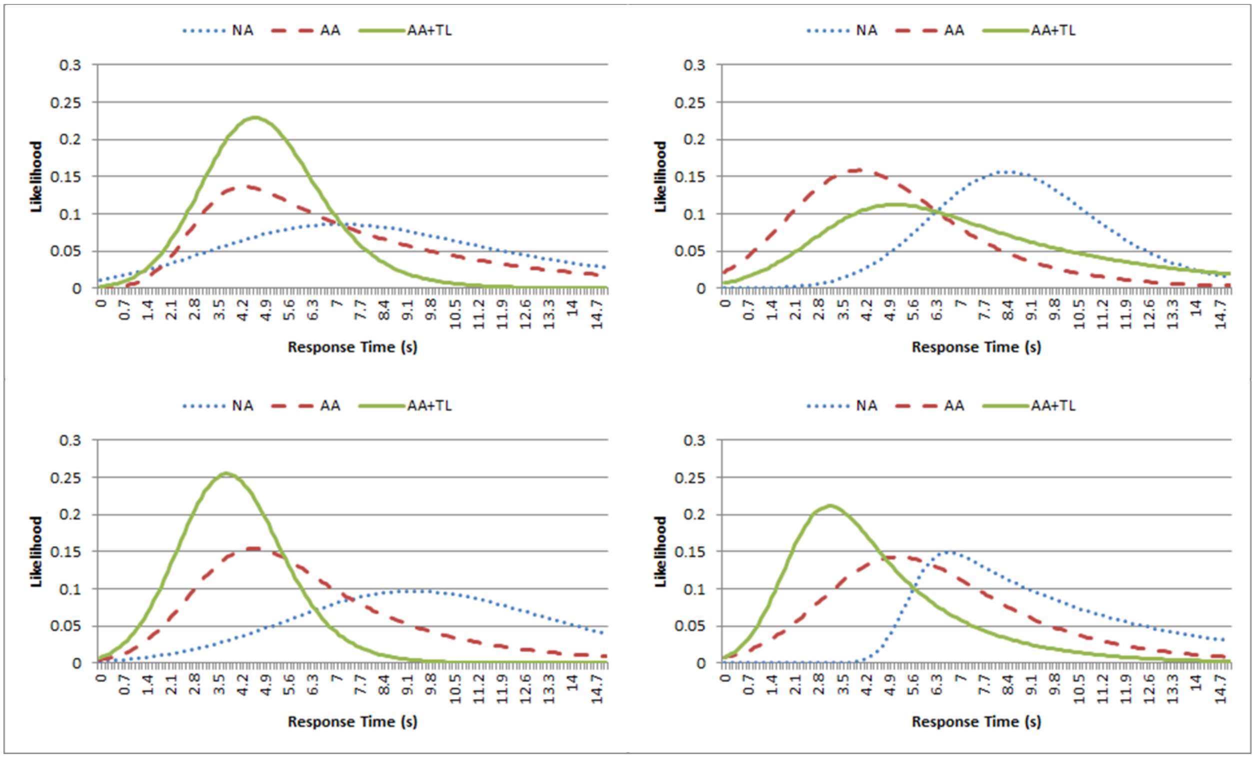

Figure 4.1: Plots of the effect of adaptive autonomy and temporal latency on four randomly-chosen user's response time distributions for high-intensity scenarios.

Due to the fact that exponential-Gaussian distributions have a variable balance between Gaussian and exponential components, it is difficult to come up with a single number to fully describe their shape. Instead, visual evidence of this effect is depicted in Figure 4.1. This suggests that when the robots use some form of adaptive autonomy, the users' response times are faster and users reported a greater feeling of productivity.

4.7.4 Subjective Results

Participants were given a debriefing questionnaire regarding their experience. Following are the questions given participants and a subjective report of the responses. Since participants didn't know which scenarios had which level of autonomy, participants didn't

What improvements would you make to the study?

Most of the responses didn't have any suggestions, and the majority of the rest dealt with the robots' path planning through the city. Most participants who responded to this question suggested the robots explore more intelligently as far as picking a direction and not following another drone. One participant suggested that the supervisor be given the ability to manually plan paths for the aerial robots. One participant suggested improving the visual fidelity of the simulator.

What was your greatest frustration?

Most participants responded that their greatest frustration was when all of the robots would notify the user of an event at the same time, thus overwhelming them. A few participants, possibly the same who reported difficult with robot path planning, expressed frustration when the robots would explore the same area of the map and not fan out better.

Were you more frustrated with more or fewer robots?

Most participants (19 of the 26) expressed being more frustrated with more robots. Five of the 26 participants expressed more frustration with fewer robots, while two declined to answer.

Were you more effective with more or fewer robots?

This question was split fairly evenly, with 12 participants answering in favor of more robots, 11 in favor of fewer, and three declining to answer. This is interesting in light of how frustrated more robots seemed to make most participants.

Chapter 5

Summary and Future Work

5.1 Summary

At the beginning of the Validation chapter (Chapter 3), the thesis made three claims that needed to be addressed through the two phases of the user study. Table 5.1 summarizes the results, while the following paragraphs describe them in greater detail.

Claim	Low-Intensity Result	High-Intensity Result
1. Predict Response Time	✓	✓
2. Reduced Cognitive Load	✓	✓
3. Increased Effectiveness	✗	✓

Table 5.1: Summary of results.

A supervisor's response times given an environment can be predicted with sufficient accuracy to validate the model described in this thesis. This was accomplished by creating a regressor with an acceptable value for R^2 while minimizing the value for the standard error.

The supervisor's cognitive load will be reduced when members of the robotic team modify their behavior in response to the operational tempo. Robots will modify their behavior by either adapting Level of Autonomy (LOA) alone or adapting both LOA and request-timing. This was verified by not having any statistical difference between teams whether they used just adaptive autonomy or adaptive autonomy with temporal latency. In both cases, participants experienced significantly less cognitive load than they did when the robots were not autonomous at all. This effect was seen in both high-intensity and low-intensity scenarios.

The human-robot team will be more effective when robots adapt LOA and request-timing when compared to adapting just LOA (due to the supervisor's reduced cognitive load). This was rejected in low-intensity scenarios hence the ✗ in Table 5.1. Nevertheless, this effect was strongly verified in high-intensity scenarios. This is likely a result of supervisors having spare 'capacity' owing to the lower intensity of the scenarios. The results support this: supervisors in high-intensity scenarios consistently performed more classifications than they did in low-intensity scenarios.

In short, all three claims were validated in high-intensity scenarios, while low-intensity scenarios didn't necessitate the full temporal latency model for users to be successful.

5.2 Future Work

In order to keep the scale of this project reasonable, a form of triage was performed on which avenues of research to pursue. Following are a list of areas that may have yielded interesting research or improved performance for this thesis, but for some reason or another were excluded from the final project.

5.2.1 The Model

Much of the work on the model was uninformed in that the model was developed before many of its concepts could be tested. Even after the simulator was functional, few components of the model were individually tested and validated. It's possible that testing components individually may expose weaknesses in the model.

Furthermore, several modeling assumptions were made without being fully tested.

World Configuration Affects Robot Sensors

Having the world's configuration affect the robot's classification accuracy might take the form of the robot being less accurate in its classifications based on distance, for example. Modeling this would bring the work closer to a real-world application.

Handling Response Time Distribution

An exGaussian distribution, used to model response times, is composed of two parts: a Gaussian and an exponential distribution. The exponential component represents when an individual becomes aware of a stimulus while the Gaussian component represents when the individual actually finishes his or her response to the stimulus.

In a preliminary form, the model made use of these representations explicitly by ascribing constant values to the mean and variance of the Gaussian component, since it is assumed that users would follow an unchanging Gaussian distribution for formulating a response to a stimulus. The operational tempo would then be used to modify the exponential rate, thereby modeling when the user becomes aware of an event as a function of how distracted the user is thought to be. This proved difficult to work with as the exGaussian parameters had to be customized for each user, and early tests indicated that a substantial number of samples would be required for the parameters to converge.

For that reason, this approach was nixed in favor of using a regressor, which appeared to produce better results on the same data. Nevertheless, this approach has some very nice theoretical properties, and merits further investigation.

Velocity As Robot Action

A preliminary form of the model had another dimension of robot action – velocity relative to an encountered individual. As the model in its current state dictates, the robot maintains a constant distance relative to the encountered individual until the human responds or the robot acts autonomously. Keeping the constant distance is easy with an aerial robot as it can guarantee a minimum distance just by setting its altitude. Using a unmanned ground vehicle, however, necessitates constant vigilance and navigation to successfully avoid an approaching person. Modeling relative velocity, enables such actions by creating a richer set of responses for the robot.

Hidden Markov Model of Operational Tempo

Operational tempo and workload are both modeled as a fraction between 0 and 1 in this study. We had considered using a Hidden Markov Model [HMM] with discrete states for operational tempo (such as low, medium, high) in order to simplify working with the concept of operational tempo, but lack of time made it difficult to explore this option in any depth. Using a HMM may simplify the modeling problem and produce more intuitive results.

Regression

The environmental variables used in the regressor are neither independent nor linear, and using a quadratic regression likely didn't fully address the problem. Early experiments also involved neural networks, though they tended to overfit the data.

Developing a more sophisticated regression technique would be a priority going forward. Neural networks can be made more robust to overfitting by carefully controlling the number of nodes, encouraging sparsity, and using lower layers composed of autoencoders (or restricted Boltzman machines) to reduce dimensionality [25]. This would require significant work on its own, but may enable far more accurate predictions.

Another area to explore regarding the prediction model is fine-tuning the model to individuals. This was addressed in a limited fashion by including a user's response-time distribution's parameters in the predictive model, but this only yielded small improvements in accuracy over ignoring it. Part of the reason for this is users only interacted with the simulation for a total of one hour. Again, it was better to incorporate the user's parameters into the predictive model than to ignore them, but more support data would enable far more accurate fine-tuning to a user.

One final improvement might involve gathering a richer set of environment variables. One aspect in particular is that the model ignores a user's recent response times in predicting the next response time. It does account for the user's overall response-time distribution, but it ignores specific instances of recent response times which may prove useful in predicting the user's future responses.

Forecasting Model

The current forecasting model is an extension of the prediction model, where τ is used as another parameter to put into the regressor. This may be improved by separating the predictive model and forecasting model. This would be accomplished by training the predictive model without the τ variable, so that it only predicts response times given the current state of the environment.

A new environmental forecasting model would need to be created to predict the state of the environment at time $t + \tau$ given its current state and possibly previous states. The forecasted environmental state could then be passed into the predictive model to predict response time after a delay of τ seconds.

5.2.2 Asymmetrical Classification Noise

A simplifying assumption of the simulator was that the robot's classifier would be equally noisy regardless of what type of individual it was classifying. In real applications, classifying individuals is an asymmetric task, which the model allows for. This should be tested in future applications.

5.2.3 Simulator Fidelity

The simulator used to validate this thesis' claims could be improved in many ways. For example, it uses a rough abstraction of a search-and-rescue application. A higher-fidelity simulation would make real-world applications of this research more obvious.

Specific improvements may include:

- Higher graphical fidelity.
- More realistic interface to what real search and rescue teams use.
- More realistic mixture and placement of victims, neutrals and hostiles in the world.
- More realistic control scheme, for example, a steering wheel with a simulated dashboard for controlling and tracking robots while driving.

5.2.4 New Applications

Considering the heavy search and rescue slant of this implementation of the model, other applications thereof may not be obvious. Another application of it may be required to demonstrate its generality.

5.2.5 Larger Scale Study

The user study ran as part of this thesis only has 5 minute long simulations. A larger scale study should include more participants longer simulations and more training for participants before they begin with the experiment.

Appendix A

Model Stability

Note that θ wasn't varied, and the same autonomy threshold of .5 was applied to every test. The same is true for $k_{\text{delay}} = 1$, indicating no utility slope for all tests, that is, utilities in the future are considered to be as valuable as present utilities.

A.1 Environment 1

Variable	Value
Encountered Classification	Hostile
Number of Robots	4
Total Blocks	36
Game Time	8.23
Vehicle Speed	0
Vehicle Turn Rate	0
Vehicle Looking Off	0
Map Explored	0.061
Individuals on Map	0
Following Vehicle	0
Robot Requests	1
Vehicle Disabled	0
Robots Disabled	0

Table A.1: The configuration of environment 1 used to calculate the data in Tables A.2 and A.3.

This environment takes place early on in a simulation when a hostile individual has been classified (with a chance of misclassification, $\eta_{\text{classifier}}$). This environment, for the most part, was

Adjusted Variable	New Value	Delay	Utility	Autonomy
All Default	N/A	4	0.2786	Full
τ_{max} high	15	4	0.2786	Full
τ_{max} low	5	4	0.2786	Full
τ_{step} high	5	5	0.2771	Full
τ_{step} low	0.5	4	0.2786	Full
$P(\text{Victim})$ high	0.667	4	0.4179	Full
$P(\text{Victim})$ low	0.167	3	0.2419	Full
$P(\text{Neutral})$ high	0.667	4	0.4179	Full
$P(\text{Neutral})$ low	0.167	3	0.2419	Full
$P(\text{Hostile})$ high	0.667	3	0.1991	Full
$P(\text{Hostile})$ low	0.167	4	0.4137	Full
$1-\eta_{\text{classifier}}$ high	0.833	4	0.2229	Full
$1-\eta_{\text{classifier}}$ low	0.5	4	0.3715	Full

Table A.2: The effect on robot performance in environment 2 with independently adjusted parameters.

Trial	τ_{max}	τ_{step}	$P(\text{Victim})$	$P(\text{Neutral})$	$P(\text{Hostile})$	$1-\eta_{\text{classifier}}$	Delay	Utility	Autonomy
1	5	4	0.374	0.353	0.276	0.600	4	0.3510	Full
2	8	1	0.032	0.109	0.858	0.705	3	0.1921	Full
3	5	1	0.212	0.061	0.727	0.863	3	0.1913	Full
4	14	3	0.095	0.450	0.455	0.807	3	0.2119	Full
5	10	2	0.221	0.461	0.318	0.460	4	0.4195	Full
6	8	2	0.441	0.136	0.595	0.971	4	0.1884	Full
7	9	1	0.493	0.268	0.239	0.440	4	0.5622	Limited
8	15	5	0.177	0.316	0.507	0.577	5	0.2505	Full
9	9	4	0.310	0.233	0.457	0.785	4	0.2159	Full
10	11	3	0.033	0.117	0.849	0.767	3	0.1903	Full

Table A.3: The effect on robot performance in environment 1 with randomly selected parameters.

not conducive to involving the supervisor. This is due to the environment's configuration (Table A.1). In general, the regressor tends to predict long response times early on in a simulation due to the supervisor's initial job of exploring the map and avoiding any hostiles which may have started the simulation near the supervisor. Robots also discover individuals very quickly early on, which can easily overwhelm the supervisor.

The one exception was when the classifier was particularly noisy (random test 7 highlighted in Table A.3). The classifier noise, coupled with the comparative dearth of hostiles (making it safer for the robot to risk a long response time) yielded the only observed combination of parameters where control might be yielded to the supervisor.

Another interesting trend can be seen in the delay columns in Tables A.2 and A.3. For the most part, when the number of hostiles increased, the maximum utility was yielded after a shorter delay. The reason for this is that the increased chance an individual is hostile increases the urgency for the robot to resolve the situation, lest the hostile successfully attack the robot before the delay expires.

A.2 Environment 2

This environment was characterized by being late in the simulation, with few robots and a disabled vehicle. A robot has just classified an individual as neutral (with classifier error). Since the rescue vehicle is disabled, the supervisor has little to do and his or her estimated response time drops substantially. This increases a robot's utility for involving the supervisor.

Notice how in Tables A.5 and A.6 that the delay is 0 for all variable combinations. The reason for this is that a disabled vehicle has perhaps the least distracted supervisor possible. After repairs to the rescue vehicle finish, the supervisor will resume navigating and presumably be distracted by his or her other duties. For that reason, the utility to involve the supervisor decreased monotonically as τ increased.

Variable	Value
Encountered Classification	Neutral
Number of Robots	2
Total Blocks	36
Game Time	265.03
Vehicle Speed	0
Vehicle Turn Rate	0
Vehicle Looking Off	1
Map Explored	0.837
Individuals on Map	31
Following Vehicle	0
Robot Requests	1
Vehicle Disabled	1
Robots Disabled	0

Table A.4: The configuration of environment 2 used to calculate the data in Tables A.5 and A.6.

Adjusted Variable	New Value	Delay	Utility	Autonomy
All Default	N/A	0	0.7864	Limited
τ_{max} high	15	0	0.7864	Limited
τ_{max} low	5	0	0.7864	Limited
τ_{step} high	5	0	0.7864	Limited
τ_{step} low	0.5	0	0.7864	Limited
$P(\text{Victim})$ high	0.667	0	0.7931	Limited
$P(\text{Victim})$ low	0.167	0	0.7830	Limited
$P(\text{Neutral})$ high	0.667	0	0.7931	Limited
$P(\text{Neutral})$ low	0.167	0	0.7830	Limited
$P(\text{Hostile})$ high	0.667	0	0.7731	Limited
$P(\text{Hostile})$ low	0.167	0	0.7931	Limited
$1-\eta_{\text{classifier}}$ high	0.833	0	0.6291	Limited
$1-\eta_{\text{classifier}}$ low	0.5	0	0.8738	Limited

Table A.5: The effect on robot performance in environment 2 with independently adjusted parameters.

Trial	τ_{max}	τ_{step}	$P(\text{Victim})$	$P(\text{Neutral})$	$P(\text{Hostile})$	$1-\eta_{\text{classifier}}$	Delay	Utility	Autonomy
1	13	1	0.280	0.477	0.243	0.790	0	0.7413	Limited
2	10	5	0.107	0.323	0.570	0.352	0	0.8894	Limited
3	7	3	0.271	0.098	0.631	0.531	0	0.6599	Limited
4	5	2	0.457	0.153	0.390	0.677	0	0.7196	Limited
5	8	4	0.243	0.199	0.558	0.386	0	0.8544	Limited
6	9	2	0.045	0.112	0.843	0.773	0	0.5386	Limited
7	5	1	0.181	0.240	0.578	0.404	0	0.8067	Limited
8	10	3	0.456	0.319	0.225	0.838	0	0.6979	Limited
9	6	2	0.184	0.358	0.458	0.546	0	0.7818	Limited
10	15	4	0.084	0.052	0.864	0.678	0	0.5438	Limited

Table A.6: The effect on robot performance in environment 2 with randomly selected parameters.

Interestingly, in trials 6 and 10 in Table A.6 (highlighted in yellow), the utility nearly dropped below .5. In both cases, the classifier was comparatively accurate and there were a large number of hostiles. Both worked to bring down the utility to the point where the robot nearly acted autonomously.

A.3 Environment 3

This environment is characterized by several outstanding robot requests and a moving vehicle which is looking off in a different direction. A robot has just classified an individual as a victim (with classification error).

This environment had some extremely interesting attributes. First, estimates for the utility of delaying notification immediately jumped up at $\tau = .5$ and slowly decayed from there. This meant that the delay was very sensitive to τ_{step}, and the utility suffered if τ_{step} was too large.

Table A.8 shows relatively stable predictions. Table A.9, on the other hand, had very erratic predictions which were extremely sensitive to τ_{step}, $P(\text{Hostile})$, and $1-\eta_{\text{classifier}}$. This is especially evident in trials 7, 3, 5, and 10 where high classifier accuracy and hostile prevalence combined to force the utility of notifying well below .5. By contrast, when those values were relatively small, as in trials 4, 2, and 9, the utility increased substantially.

Variable	Value
Encountered Classification	Victim
Number of Robots	4
Total Blocks	36
Game Time	93.34
Vehicle Speed	1
Vehicle Turn Rate	0
Vehicle Looking Off	1
Map Explored	0.427
Individuals on Map	12
Following Vehicle	3
Robot Requests	2
Vehicle Disabled	0
Robots Disabled	0

Table A.7: The configuration of environment 3 used to calculate the data in Tables A.8 and A.9.

Adjusted Variable	New Value	Delay	Utility	Autonomy
All Default	N/A	1	0.5428	Limited
τ_{max} high	15	1	0.5428	Limited
τ_{max} low	5	1	0.5428	Limited
τ_{step} high	5	0	0.5401	Limited
τ_{step} low	0.5	0.5	0.5430	Limited
$P(\text{Victim})$ high	0.667	1	0.5483	Limited
$P(\text{Victim})$ low	0.167	1	0.5400	Limited
$P(\text{Neutral})$ high	0.667	1	0.5483	Limited
$P(\text{Neutral})$ low	0.167	1	0.5400	Limited
$P(\text{Hostile})$ high	0.667	0	0.5300	Limited
$P(\text{Hostile})$ low	0.167	0	0.5483	Limited
$1-\eta_{\text{classifier}}$ high	0.833	1	0.5373	Limited
$1-\eta_{\text{classifier}}$ low	0.5	1	0.5483	Limited

Table A.8: The effect on robot performance in environment 3 with independently adjusted parameters.

Trial	τ_{max}	τ_{step}	$P(\text{Victim})$	$P(\text{Neutral})$	$P(\text{Hostile})$	$1-\eta_{classifier}$	Delay	Utility	Autonomy
1	5	4	0.350	0.320	0.330	0.670	0	0.5399	Limited
2	14	2	0.461	0.402	0.138	0.546	2	0.9149	Limited
3	15	1	0.341	0.121	0.538	0.815	1	0.3922	Full
4	15	3	0.470	0.192	0.338	0.499	0	0.7142	Limited
5	5	1	0.127	0.393	0.480	0.756	1	0.4214	Full
6	5	5	0.169	0.299	0.532	0.500	0	0.5137	Limited
7	8	4	0.119	0.019	0.862	0.957	0	0.3517	Full
8	11	2	0.307	0.369	0.323	0.601	2	0.6112	Limited
9	12	1	0.347	0.433	0.221	0.366	0	0.9793	Limited
10	15	2	0.161	0.455	0.384	0.773	0	0.4438	Full

Table A.9: The effect on robot performance in environment 3 with randomly selected parameters.

A.4 Analysis

The above data indicate that the model can select an appropriate level of autonomy across a wide range of environments with different input parameters. Tables A.2, A.5 and A.8 all had very stable utilities when only a single parameter was changed, indicating consistency across similar sets of parameters. However, when all parameters were randomly chosen, the model reported different utilities for what were very different environments. Still, there was a logic behind the autonomy and response time selections: the robot would act autonomously when the supervisor was unlikely to respond in time or the robot was likely to address the situation successfully. When the supervisor was particularly busy, the robot would delay a response instead of acting autonomously. Though time constraints prohibited exhaustively tested differing parameters in a human-factors study, we are confident the model would still perform adequately.

Appendix B

Response-Time Regression Development

This appendix describes the process of developing a regression in order to predict a user's response time given the environment's configuration.

Data was collected from the preliminary phase of user-testing. Sixteen trials were run for a total of four hours worth of simulation. This produced approximately 500 data points correlating the state of the environment with the user's response time to a robot's request. Each data point consisted of the following attributes:

- Number of robots
- Total blocks in city
- Time into the simulation
- Vehicle speed
- Vehicle turn rate
- Absolute vehicle turn rate (irrespective of direction)
- Is vehicle looking away
- Percent of map explored
- Individuals discovered
- Individuals following vehicle
- Outstanding robot requests
- Is vehicle disabled
- How many robots are disabled
- User's response time distribution mean

- User's response time distribution variance
- User's response time distribution lambda
- Time for a user to respond to a request sent when the environment was as described in the above values (response time)

Because the model demands being able to forecast a user's response time if a message is sent τ seconds into the future, an additional dimension for each data point is added, τ. This is done by assigning all data points directly read from log files $\tau = 0$. This makes sense, because those data points all represent the response time given the current ($\tau = 0$) environmental state. The next logged response is determined, and the duration into the future it occurred is calculated. That becomes the value for τ for a new data point consisting of the environment at $\tau = 0$, the new value for τ and the response time for the next logged response. This allowed the creation of a substantially larger pool of 1200 data points for varying levels of τ.

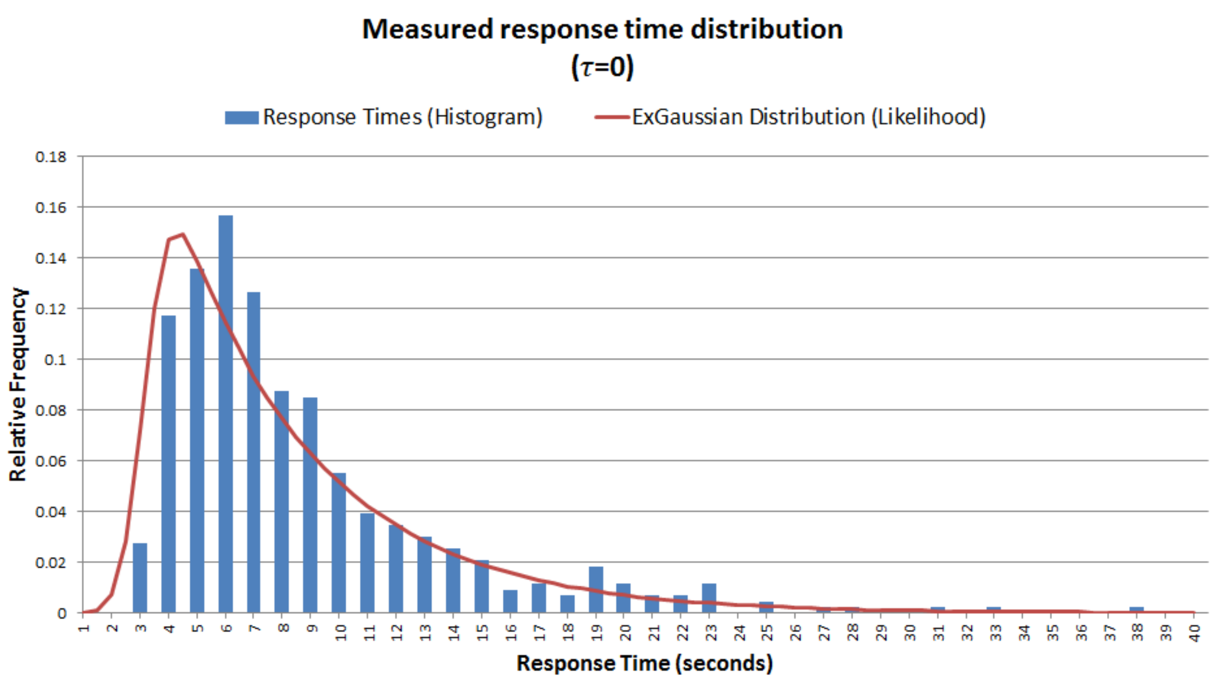

Figure B.1: The sampled histogram of response times for all users contrasted with its corresponding exGaussian distribution. This graph was produced using the participants' response times during the preliminary phase of testing.

In order to predict response time, this thesis uses a regression on the support data gathered during this phase of testing. In order to establish a regression that worked sufficiently well, numerous iterations of improving the regression algorithm were carried out.

B.1 Metrics for Comparing Regressions

In order to compare the regressions reported in this appendix, two metrics are used. The first one is the standard error, which represents the standard deviation of the regressor's residuals. This is easy to calculate since most regression packages report the mean squared error, which is the variance of the residuals. Taking the square root of the mean squared error then yields standard error. The standard error was chosen over mean squared error simply because standard error offers an intuitive idea of the spread of the residuals and the accuracy of the regression.

It is difficult to know exactly how low standard error needs to be in order to make a usable system. Instead, the analysis of this metric is characterized as an optimization problem where the goal is to get it as close to ideal as possible. Standard error should be minimized, as smaller standard error corresponds to more accurate regressions.

R^2 is used as a measure of goodness of fit. R^2 ranges from 0 to 1, where higher values indicate a better fit for a regressor over a set of data points. Acceptable values of R^2 vary significantly between applications, though a minimum in regression involving environmental state seems to be $R^2 > .4$. As with standard error, R^2 is treated as a value to optimize (albeit with maximization in this case).

Finally, a regression's ability to generalize is measured by using 10-fold cross-validation. The standard error increasing substantially in cross-validation over the non-cross-validated regression indicates that the regression overfits the data. This metric will be shortened to *10x Standard Error* when reporting data.

B.2 Attempted Models

This section reports on various regression models investigated while attempting to fit the generated support data. Table B.1 summarizes the results of each iteration, with more specific descriptions in the following subsections.

Model Name	R^2	Standard Error	10X Standard Error
Linear regression over raw data	0.346	6.997	7.588
Second order orthogonal polynomial	0.370	6.889	7.203
Third order orthogonal polynomial	0.375	6.893	7.315
Second order crossed polynomial	0.548	5.590	24.413
Second order crossed polynomial + AIC	0.572	5.130	5.595

Table B.1: Summary of each attempted regression. The regression used in the user study is highlighted in yellow.

B.2.1 Linear regression over the raw data

This first, naïve approach is a linear regression over the unaltered data set described above in order to predict response times.

R^2	Standard Error	10X Standard Error
0.346	6.997	7.588

Table B.2: Results of linear regression over the raw data.

Table B.2 summarizes the results of this regression. This approach's R^2 value was far too low to be useful. Nevertheless, this approach provides a baseline for the next regressions to beat.

B.2.2 Higher order orthogonal polynomial

The relatively poor values in this approach stem from the fact that several of the variables in each datapoint have a non-linear relationship with the user's response time. A second order polynomial over the dataset is used to address this issue. This is done by incorporating, in addition to the attributes in the original data set, each attribute multiplied by itself. For example, if the original dataset contained three attributes, $\{X,Y,Z\}$, the new data set would contain six, $\{X,Y,Z,X^2,Y^2,Z^2\}$. The word orthogonal is used to describe this approach because each attribute is treated as independent of other attributes.

R^2	Standard Error	10X Standard Error
0.37	6.889	7.203

Table B.3: Second order orthogonal polynomial results.

Table B.3 shows the modest improvement over the original approach. Since this approach did yield an improvement, we proceeded by incorporating the third order orthogonal polynomials as well.

R^2	Standard Error	10X Standard Error
0.375	6.893	7.315

Table B.4: Third order orthogonal polynomial results.

Table B.4 shows virtually no improvement in the third order polynomial over the second order polynomial, while overfitting more than the second order polynomial. This makes sense, as a higher-order polynomial would have more inflection points, allowing it to more closely match each data point while missing the overall trend. For that reason, second-order polynomials are used in future tests.

B.2.3 Second order crossed polynomial

This approach takes the cross product of the set of attributes with itself in order to produce a new set of attributes to union with the original set of attributes. For example, if the original dataset contained three attributes, $\{X,Y,Z\}$, the new data set would contain nine, $\{X,Y,Z,X^2,XY,XZ,Y^2,YZ,Z^2\}$.

R^2	Standard Error	10X Standard Error
0.548	5.590	24.413

Table B.5: Second order crossed polynomial results.

Though this produced a substantial improvement over previous attempts in R^2 and standard error, it overfits and fails to generalize. This disappointing result comes from the fact that explicitly modeling every second-order relationship between variables allows for chance-correlations between variables to dominate true-correlations.

B.2.4 Second order crossed polynomial with AIC

This approach uses the cross product of the set of attributes, as in the previous attempt. To simplify the model, and decrease the influence of chance-correlations, we apply a model selection method known as Akaike information criterion (AIC) [1][6]. AIC scores regression models by rewarding goodness of fit and penalizing number of attributes. It then simplifies the model by removing attributes which don't contribute to the model's goodness of fit. Since more attributes almost always correlates with a better fit, AIC eventually reaches an equilibrium where all remaining attributes are important enough to the goodness of fit that none of them can be removed. This effectively decreases the dimensionality of a dataset, simplifying it without sacrificing the regressor's ability to predict.

When passing in the second order crossed polynomial dataset with 256 attributes, AIC whittled the number of attributes down to 49. This has the desirable effect of mitigating overfitting by applying Occam's Razor [4]: if two models have similar ability to predict on a dataset, the simpler model tends to generalize more gracefully.

R^2	Standard Error	10X Standard Error
0.572	5.130	5.595

Table B.6: Second order crossed polynomial with AIC results.

Indeed, Table B.6 shows the dramatic improvement offered by this method. Not only did this produce strong improvements in R^2 and standard error, but it generalized extremely well. Considering the substantial improvement this model offers, we opted to use it in the full user study.

Appendix C

User Study Materials

C.1 Participant Involvement Outline

This is the outline proctors used while administering the experiment to participants. Capitalized text in italics indicates actions, bold text indicates timing of when to present a block of text, while normal text indicates talking points to cover.

Before starting

- Thank for participation.
- Mention that their test will be used for my thesis and a paper.
- Go over informed consent.
- *HAVE THE PARTICIPANT FILL OUT THE QUESTIONNAIRE*
- No cell phones.
- If the participant has questions, don't stop. Just keep interacting with the simulator and ask me.
- If you're curious about what we're testing, I'll be happy to describe it afterward. For now just focus on your tasks.
- If the participant needs to take a break or go to the bathroom please do so during the break in simulations, if possible. Otherwise, we may have to repeat a simulation.
- The simulator is timed, I need you for up to the full hour you signed up for. If you have another conflict before then, we can reschedule.

- Describe the scenario. City underwent a disaster. Participant is on a search and rescue team and must uncover survivors. Some streets blocked, though all intersections are accessible—no blocked off parts of the city. Robots help you discover blockages and victims. Three types of individuals: victims, neutrals, hostiles. go over each individual's role.
- *PARTICIPANT USES TUTORIAL*

After tutorial

- Ask about volume levels.
- Warn that even though you can lower altitude on the robots, it puts the robots at risk of being hit by a rock.
- Autoclassification is noisy. Only use if you're overwhelmed.
- The robots may occasionally classify without notifying participant.
- the robots may occasionally delay classification.
- Go over reference sheet with participant.
- TLX survey at the end of the scenario.
- Inform participant that they will now have two practice runs with the simulator. Have the participant focus on speedy responses to the robots.
- *PARTICIPANT DOES TWO PRACTICE RUNS, ONE LOW INTENSITY, ONE HIGH INTENSITY*

After practice runs

- Tell participant we're starting into the actual experiment.
- Tell participant his or her primary objective is to rescue victims. Classifying civilians for the robots is only supposed to aid in that main task.
- Mention any known bugs and how the participant should respond to them.
- *SELECT A NEW RANDOM SCENARIO ORDERING, START HAVING THE PARTICIPANT WORK THROUGH THE SCENARIOS*

After each scenario

- Ask for general thoughts about the mission.
- How did that mission compare to the previous missions?
- After all missions: *HAVE THE PARTICIPANT FILL OUT THE DEBRIEFING. THEN FILL OUT PAYMENT VOUCHER.*

Appendix D

Experimental Data

This appendix contains the raw data collected during the user study. Note that these tables use the same abbreviations listed in Table 3.2. Additionally, the word 'Low' refers to low-intensity scenarios while 'High' refers to high-intensity scenarios.

The following metrics are examined in the tables in this section:

- TLX Mental (Table D.1)
- TLX Temporal (Table D.1)
- TLX Performance (Table D.1)
- TLX Effort (Table D.1)
- TLX Frustration (Table D.1)
- TLX Composite (Table D.1)
- Victims Rescued (Table D.2)
- Neutrals Rescued (Table D.2)
- Hostile Attacks (Table D.2)
- Robot Classifications (Table D.4)
- Human Classifications (Table D.4)
- AIIR (Table D.4)

D.1 TLX Data

Scenario	Mental		Temporal		Performance		Effort		Frustration		Composite	
	$\bar{x}$	s	$\bar{x}$	s	$\bar{x}$	s	$\bar{x}$	s	$\bar{x}$	s	$\bar{x}$	s
Low NA	0.324	6.701	2.113	6.700	−1.200	7.446	−1.015	7.086	−1.312	7.341	−0.617	6.334
High NA	5.140	5.691	4.276	6.543	1.774	8.125	3.764	6.631	2.631	6.676	5.133	5.834
Low AA	−3.409	5.776	−1.350	6.220	−3.033	6.528	−2.656	6.382	−2.113	8.121	−4.080	5.392
High AA	−2.427	7.764	−1.768	7.949	−3.709	5.756	−3.093	6.446	−2.099	7.843	−3.668	7.293
Low AA+TL	−1.478	5.476	0.738	6.417	−3.996	5.526	−3.994	5.273	−2.382	6.519	−4.327	4.571
High AA+TL	−1.474	7.181	−1.235	7.858	−2.522	6.536	−1.808	6.356	0.710	6.600	−1.583	6.858

Table D.1: Complete list of means and variances of TLX scores. Note that TLX scores range from -10 to 10, with lower values indicating lower cognitive load, lower workload, and more perceived success. Table 3.2 for abbreviations.

The values in the High NA row were significant outliers in the data in this table. This is made obvious in Figure D.1. Apparently when the robots messaged their supervisor for every event, it induced significant load on the supervisor. Because of High NA, running an ANOVA on each category yields significant differences in the data.

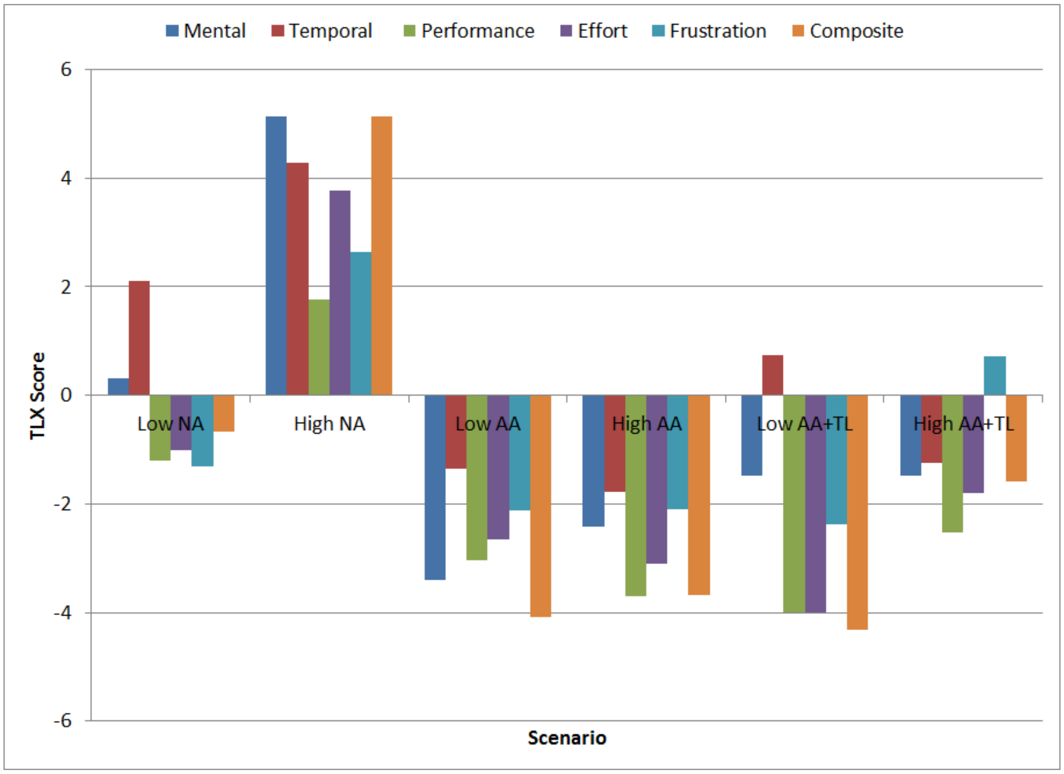

Figure D.1: The contents of Table D.1 depicted in a bar chart.

D.2 Civilian Data

Scenario	Victims		Neutrals		Hostile	
	$\bar{x}$	s	$\bar{x}$	s	$\bar{x}$	s
Low NA	7.544	2.751	2.261	1.514	0.870	2.341
High NA	6.182	2.991	2.261	1.322	0.870	1.890
Low AA	10.194	3.486	2.130	1.486	0.522	1.504
High AA	8.914	3.088	1.913	1.443	0.739	2.750
Low AA+TL	7.784	3.040	2.000	1.314	0.739	2.544
High AA+TL	10.495	3.633	1.913	1.379	0.696	1.636

Table D.2: Complete list of means and variances for rescues and attacks. Table 3.2 for abbreviations.

Chapter 4 already did an extensive analysis on the number of victims rescued. As for neutrals rescued and hostile attacks, Table D.3 reports the ANOVA for both of those stats. As it depicts, there was no significant difference between scenarios for either neutrals rescued or hostile attacks.

	Neutrals Rescued	Hostile Attacks
ANOVA p-value	0.888	0.993

Table D.3: The ANOVA p-values for neutrals rescued and hostile attacks.

D.3 Robot Performance Data

Scenario	Robot		Human		AIIR		Map Exp	
	$\bar{x}$	s	$\bar{x}$	s	$\bar{x}$	s	$\bar{x}$	s
Low NA	0.304	0.635	14.217	4.056	0.019	0.039	0.614	0.146
High NA	1.217	3.044	24.000	5.970	0.049	0.123	0.771	0.122
Low AA	12.217	7.198	6.130	3.507	0.594	0.284	0.694	0.114
High AA	21.000	9.601	8.826	6.300	0.677	0.241	0.873	0.090
Low AA+TL	8.478	6.508	8.217	3.503	0.460	0.267	0.673	0.117
High AA+TL	17.565	11.373	10.826	6.638	0.581	0.280	0.837	0.096

Table D.4: Complete list of means and variances of human classifications, robot classifications, the AI Involvement Ratio (AIIR), and the percent of the map explored. Table 3.2 for abbreviations.

Though these measures are commented on elsewhere, their ANOVA values are listed in Table D.5. Unsurprisingly, there are significant difference within each category.

	Robot	Human	AIIR	Map Explored
ANOVA p-value	$< .001$	$< .001$	$< .001$	$< .001$

Table D.5: The ANOVA p-value for robot rescues, human rescues, AIIR and Map Explored.

AIIR and map explored are both covered in Chapter 4. Below is a brief treatment of human classifications and robot classifications.

Comparison	p-value
LowNA $>$ LowAA	$<$ **0.001**
LowNA $>$ LowAA $+$ TL	$<$ **0.001**
LowAA $>$ LowAA $+$ TL	0.055
HighNA $>$ HighAA	$<$ **0.001**
HighNA $>$ HighAA $+$ TL	$<$ **0.001**
HighAA $>$ HighAA $+$ TL	0.245

Table D.6: The p-values for pairwise t-test comparisons of robot classification count.

Comparison	p-value
LowNA $>$ LowAA	$<$ **0.001**
LowNA $>$ LowAA $+$ TL	$<$ **0.001**
LowAA $>$ LowAA $+$ TL	**0.037**
HighNA $>$ HighAA	$<$ **0.001**
HighNA $>$ HighAA $+$ TL	$<$ **0.001**
HighAA $>$ HighAA $+$ TL	0.271

Table D.7: The p-values for pairwise t-test comparisons of human classification count.

Tables D.6 and D.7 contain the p-values for the pairwise comparisons of human and robot classifications. Interestingly, they report similar results. There is a significant difference for most comparisons, but with no significant difference between AA and AA+TL. The likely reason for this is that in both cases the robots carry a significant amount of the supervisor's workload.

The sole exception is highlighted in yellow in Table D.6, where the t-test just barely failed to find significant difference between LowAA $>$ LowAA $+$ TL.

References

[1] Hirotugu Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(2):716–723, 1974.

[2] Lisanne Bainbridge. Ironies of Automation. *Control*, 19(6), 1983.

[3] K. Suzanne Barber, Anuj Goel, and Cheryl E. Martin. Dynamic adaptive autonomy in multi-agent systems. *Journal of Experimental & Theoretical Artificial Intelligence*, 12(2):129–147, 2000.

[4] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Occam's razor. *Information Processing Letters*, 24(6):377–380, 1987.

[5] Erwin R. Boer, Okihiko Nakayama, Tohru Futami, and Tomokazu Nakamura. Development of a steering entropy method for evaluating driver workload. *Internation Congress and Exposition*, 1999.

[6] Kenneth P. Burnham and David R. Anderson. *Model Selection and Multimodel Inference: A Practical Information-Thoretic Approach*, volume 2. Springer-Verlag, 2002.

[7] Jessie Y. C. Chen and Carla T. Joyner. Concurrent Performance of Gunner's and Robotics Operator's Tasks in a Multitasking Environment. *Military Psychology*, 21(1):98–113, January 2009. ISSN 0899-5605.

[8] Jacob W. Crandall and Michael A. Goodrich. Experiments in adjustable autonomy. In *Systems, Man, and Cybernetics, 2001 IEEE International Conference on (Tuscon, AZ, USA)*, volume 3, pages 1624–1629. IEEE, October 2001.

[9] Mihaly Csikszentmihalyi. *Flow: The Psychology of Optimal Experience*. Harper & Row, 1990.

[10] Mary L. Cummings, Sylvain Bruni, S. Mercier, and Mitchell P. J. Automation architecture for single operator, multiple uav command and control. *The International C2 Journal*, 1(2):1–24, 2007.

[11] Mica R. Endsley. Design and evaluation for situation awareness enhancement. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting (Anaheim, CA, USA)*, volume 32, pages 97–101. SAGE Publications, October 1988.

[12] Marco Javier Flores, José María Armingol, and Arturo de la Escalera. Driver drowsiness warning system using visual information for both diurnal and nocturnal illumination conditions. *EURASIP Journal on Advances in Signal Processing*, 2010:3, 2010.

[13] James Fogarty, Scott E. Hudson, Christopher G. Atkeson, Daniel Avrahami, Jodi Forlizzi, Sara Kiesler, Johnny C. Lee, and Jie Yang. Predicting human interruptibility with sensors. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 12(1):119–146, 2005.

[14] James Fogarty, Scott E. Hudson, and Jennifer Lai. Examining the robustness of sensor-based statistical models of human interruptibility. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Vienna, Austria)*, pages 207–214. ACM, April 2004.

[15] James Fogarty, Andrew J. Ko, Htet Htet Aung, Elspeth Golden, Karen P. Tang, and Scott E. Hudson. Examining task engagement in sensor-based statistical models of human interruptibility. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Portland, OR, USA)*, pages 331–340. ACM, April 2005.

[16] Puneet Goel, Stergios I. Roumeliotis, and G. Sukhatme. Robust localization using relative and absolute position estimates. In *Intelligent Robots and Systems, 1999. IROS'99. Proceedings. 1999 IEEE/RSJ International Conference on*, volume 2, pages 1134–1140. IEEE, 1999.

[17] Michael A. Goodrich and Erwin R. Boer. Designing human-centered automation: Tradeoffs in collision avoidance system design. *Transportation*, 1(1):40–54, 2000.

[18] Michael A. Goodrich and Alan C. Schultz. Human-robot interaction: A survey. *Foundations and Trends in Human-Computer Interaction*, 1(3):203–275, 2007. ISSN 1551-3955.

[19] Michael A. Goodrich, Dan R. Olsen, Jacob W. Crandall, and Thomas J. Palmer. Experiments in adjustable autonomy. In *Proceedings of IJCAI Workshop on Autonomy, Delegation and Control: Interacting with Intelligent Agents (Seattle, WA, USA)*, pages 1624–1629, August 2001.

[20] Michael A. Goodrich, Erwin R. Boer, Jacob W. Crandall, Robert W. Ricks, and Morgan L. Quigley. Behavioral entropy in human-robot interaction. In *Proceedings of PERMIS 2004 (Gaithersburg, MD, USA)*, August 2004.

[21] Michael A. Goodrich, Timothy W. McLain, Jeffrey D. Anderson, Jisang Sun, and Jacob W. Crandall. Managing autonomy in robot teams: observations from four experiments. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (Washington, DC, USA)*, pages 25–32. ACM, March 2007.

[22] Neil J. Gordon, David J. Salmond, and Adrian F. M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEEE Proceedings F (Radar and Signal Processing)*, volume 140, pages 107–113. IET, April 1993.

[23] Benjamin Hardin and Michael A. Goodrich. On using mixed-initiative control: A perspective for managing large-scale robotic teams. In *Proceedings of the 4th ACM/IEEE International Conference on Human Robot Interaction (San Diego, CA, USA)*, pages 165–172. ACM, March 2009.

[24] S. G. Hart and L. E. Staveland. Development of NASA-TLX (task load index): Results of empirical and theoretical research. *Human Mental Workload*, pages 139–183, 1988.

[25] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[26] Scott Hudson, James Fogarty, Christopher Atkeson, Daniel Avrahami, Jodi Forlizzi, Sara Kiesler, Johnny Lee, and Jie Yang. Predicting human interruptibility with sensors: A Wizard of Oz feasibility study. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Fort Lauderdale, FL, USA)*, pages 257–264. ACM, April 2003.

[27] Matthew Johnson, Jeffrey M. Bradshaw, Paul J. Feltovich, and Robert R. Hoffman. Beyond cooperative robotics: The central role of interdependence in coactive design. *IEEE Intelligent Systems*, (June):81–88, 2011.

[28] David B. Kaber and Mica R. Endsley. *The effects of level of automation and adaptive automation on human performance, situation awareness and workload in a dynamic control task*, volume 5. Taylor & Francis, January 2004. ISBN 1463922021000.

[29] David B. Kaber, Emrah Onal, and Mica R. Endsley. Design of automation for telerobots and the effect on performance, operator situation awareness, and subjective workload. *Human Factors and Ergonomics in Manufacturing*, 10(4):409–430, 2000.

[30] Rudolph E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

[31] Rudolph E. Kalman and Richard S. Bucy. New results in linear filtering and prediction theory. *Journal of Basic Engineering*, 83(1):95–108, 1961.

[32] Bennie Lewis, Bulent Tastan, and Gita Sukthankar. Improving multi-robot teleoperation by inferring operator distraction. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1 (Toronto, Canada)*, pages 1505–1506. International Foundation for Autonomous Agents and Multiagent Systems, May 2010.

[33] Robert Duncan Luce. *Response Times: Their Role in Inferring Elementary Mental Organization*, volume 8. Oxford University Press, 1986.

[34] Jake Olivier and Melissa M Norberg. Positively skewed data: Revisiting the box-cox power transformation. *International Journal of Psychological Research*, 3(1):68–77, 2010.

[35] Evan M. Palmer, Todd S. Horowitz, Antonio Torralba, and Jeremy M. Wolfe. What are the shapes of response time distributions in visual search? *Journal of Experimental Psychology: Human Perception and Performance*, 37(1):58, 2011.

[36] Raja Parasuraman, Thomas B. Sheridan, and Christopher D. Wickens. A model for types and levels of human interaction with automation. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 30(3):286–297, 2000.

[37] Karl Pearson. The problem of the random walk. *Nature*, 72(1865):294, 1905.

[38] Roger Ratcliff. Group reaction time distributions and an analysis of distribution statistics. *Psychological Bulletin*, 86(3):446, 1979.

[39] Doug Rohrer and John T. Wixted. An analysis of latency and interresponse time in free recall. *Memory and Cognition*, 1994.

[40] Stuart Russell and Peter Norvig. *Artificial intelligence: A modern approach*. Prentice Hall, 1995.

[41] Jean Scholtz. Theory and Evaluation of Human Robot Interactions. *Human-Computer Interaction*, 00(C):1–10, 2003.

[42] Dirk Schulz, Wolfram Burgard, Dieter Fox, and Armin B. Cremers. Tracking multiple moving targets with a mobile robot using particle filters and statistical data association. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on (Seoul, South Korea)*, volume 2, pages 1665–1670. IEEE, May 2001.

[43] Thomas B. Sheridan and William L. Verplank. Human and computer control of undersea teleoperators. *Technical Report, MIT Man-Machine Systems Laboratory*, 1978.

[44] Peter Swerling. *A proposed stagewise differential correction procedure for satellite tracking and prediciton*. Rand Corporation, 1958.

[45] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*, volume 45. MIT Press, 2005.

[46] G. Weiss. Aspects and applications of the random walk (random materials & processes). 2005.

[47] Robert Whelan. Effective analysis of reaction time data. *The Psychology Record*, 58:475–482, 2008.

[48] William J. Wilson. *Visual servo control of robots using kalman filter estimates of robot pose relative to work-pieces*, volume 7. World Scientific Series on Robotics and Automation, 1993.

[49] Yasushi Yagi, Shinjiro Kawato, and Saburo Tsuji. Real-time omnidirectional image sensor (copis) for vision-guided navigation. *Robotics and Automation, IEEE Transactions on*, 10(1): 11–22, 1994.

[50] Dror Yashpe. *Influence of Human Reaction Time in Human-Robot Collaborative Target Recognition Systems*. PhD thesis, Ben-Gurion University of the Negev, 2009.