Human-aware information-theoretic control of robotic swarms

Rafal David Krzysiak
krzysiakgoral@gmail.com

Follow this and additional works at: https://huskiecommons.lib.niu.edu/allgraduate-thesesdissertations

Part of the Robotics Commons

Recommended Citation
https://huskiecommons.lib.niu.edu/allgraduate-thesesdissertations/7273

This Dissertation/Thesis is brought to you for free and open access by the Graduate Research & Artistry at Huskie Commons. It has been accepted for inclusion in Graduate Research Theses & Dissertations by an authorized administrator of Huskie Commons. For more information, please contact jschumacher@niu.edu.
ABSTRACT

HUMAN-AWARE INFORMATION-THEORETIC CONTROL OF ROBOTIC SWARMS

Rafal Krzysiak, M.S.
Department of Mechanical Engineering
Northern Illinois University, 2021
Sachit Butail, Director

Robotic swarms provide a scalable and robust solution for monitoring large environments. In this context, the inclusion of a human has the potential advantage of incorporating prior knowledge about the target location or dynamics, thus increasing the overall efficiency of the task at hand. This is especially critical when dealing with time-intensive missions such as search and rescue. In this thesis we develop a general information-theoretic framework to control multiple autonomous robots in search and rescue missions that include a human teleoperator. Human prior knowledge is modeled to capture target location and dynamics, and a mutual information based control is formulated to let autonomous robots weight between two strategies: independent search or staying in proximity of a reference robot representing human input. The control actions optimize a weighted sum of normalized mutual information calculated using particle filtered estimates of the target and the reference robot. We implement the framework to simulate two widely different scenarios designed after search and rescue missions from literature and incorporate varying levels of accuracy in human prior knowledge. Our results indicate that within the simulated environments, mission performance depends on how robots weight between the two strategies, with the
amount of the optimal control effort shared between strategies affected by prior knowledge and number of robots. The proposed information-theoretic abstraction of human robot interaction can be implemented on a wide variety of scenarios and the results highlight the role of human prior knowledge towards eliciting effective robotic assistance in time-intensive missions.

To help better understand how humans solve the search and rescue problem and the effect of prior knowledge about environment and target location, we next conduct a human-subjects study where participants teleoperate a ground robot as they search for a missing target with varying levels of prior knowledge about target location and environmental map. Our preliminary experimental results indicate that prior knowledge affects not only the time to find the missing target but also the teleoperation behavior with significant differences in robot speed and control actions made by a participant. The experimental data set and the preliminary results set the stage for a data-driven dynamic model of a human teleoperator in search operations.
HUMAN-AWARE INFORMATION-THEORETIC CONTROL
OF ROBOTIC SWARMS

BY
RAFAL KRZYSIAK
© 2021 Rafal Krzysiak

A THESIS SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE
MASTER OF SCIENCE

DEPARTMENT OF MECHANICAL ENGINEERING

Thesis Director:
Sachit Butail
I would like to thank Dr. Sachit Butail for his guidance and support. He was always ready to help in all aspects of the research, through in person or virtual meetings. The knowledge that he has shared with me throughout my journey has been vital to the development of my research as well as myself in becoming a researcher.

In addition, I would like to thank the other thesis committee members, Dr. Ji-Chul Ryu and Dr. Brian Coller, for their collaboration and their constructive comments.

I would also like to thank my fellow lab members in EAScL: Zachary Taylor and Arunim Bhattacharya. They provided constructive feedback in ways to improve my robotic teleoperated testbed for the human subject experiments. The work also done by Zachary Taylor, regarding the live and offline tracking system, was instrumental and vital to allow not only myself but future robotic research in the NIU Omron Robotics Lab to accurately track robots in the whole lab with ease.

I would like to give thanks to the NIU lab manager Mike Reynolds for the help and tools given to mount the webcams to the NIU Omron Lab, while we were building the tracking system. In addition, I would like to thank the National Science Foundation for their support and an assistantship from the Great Journeys Assistantship program.

Finally, I would like to express tremendous gratitude to my parents, and brother who have always supported me throughout my journey with non-stop encouragement and love.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LIST OF TABLES</strong></td>
<td></td>
<td>v</td>
</tr>
<tr>
<td><strong>LIST OF FIGURES</strong></td>
<td></td>
<td>vi</td>
</tr>
<tr>
<td><strong>LIST OF SYMBOLS</strong></td>
<td></td>
<td>xi</td>
</tr>
<tr>
<td><strong>Chapter</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1</td>
<td>Literature review</td>
<td>3</td>
</tr>
<tr>
<td>1.2</td>
<td>Contribution</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>BACKGROUND</td>
<td>6</td>
</tr>
<tr>
<td>2.1</td>
<td>Particle filtering</td>
<td>6</td>
</tr>
<tr>
<td>2.2</td>
<td>Mutual information based control</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>INFORMATION BASED CONTROL OF MULTI-ROBOT SYSTEMS</td>
<td>11</td>
</tr>
<tr>
<td>3.1</td>
<td>A generic SAR setup for a human robot team</td>
<td>11</td>
</tr>
<tr>
<td>3.2</td>
<td>Information based search with prior knowledge</td>
<td>12</td>
</tr>
<tr>
<td>3.3</td>
<td>Scenario I: indoor environment</td>
<td>16</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Setup</td>
<td>17</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Simulations</td>
<td>20</td>
</tr>
<tr>
<td>3.3.3</td>
<td>Results</td>
<td>24</td>
</tr>
<tr>
<td>3.4</td>
<td>Scenario II: moving target in a park</td>
<td>29</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Setup</td>
<td>29</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Simulations</td>
<td>32</td>
</tr>
<tr>
<td>Chapter</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>--------------------------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>3.4.3 Results</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>4 EXPERIMENTAL STUDY</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>4.1 Introduction</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>4.2 Methods</td>
<td>41</td>
<td></td>
</tr>
<tr>
<td>4.2.1 Experimental setup</td>
<td>41</td>
<td></td>
</tr>
<tr>
<td>4.2.1.1 Teleoperation</td>
<td>43</td>
<td></td>
</tr>
<tr>
<td>4.2.1.2 Tracking</td>
<td>47</td>
<td></td>
</tr>
<tr>
<td>4.2.1.3 Autonomous robot</td>
<td>49</td>
<td></td>
</tr>
<tr>
<td>4.2.2 Experimental conditions</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>4.2.3 Experimental procedure</td>
<td>52</td>
<td></td>
</tr>
<tr>
<td>4.2.4 Data analysis</td>
<td>56</td>
<td></td>
</tr>
<tr>
<td>4.3 Results</td>
<td>62</td>
<td></td>
</tr>
<tr>
<td>4.4 Discussion</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>5 CONCLUSION</td>
<td>74</td>
<td></td>
</tr>
<tr>
<td>REFERENCES</td>
<td>76</td>
<td></td>
</tr>
</tbody>
</table>
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Scenario I simulation parameters</td>
</tr>
<tr>
<td>3.2</td>
<td>Scenario II simulation parameters</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Sampling showing un-normalized mutual information values for comparison between the reference robot and the target, from the perspective of the autonomous robot.</td>
</tr>
<tr>
<td>3.2</td>
<td>Setup for SAR in Scenario I showing the reference robot starting position (blue square) and autonomous robots’ starting positions (red circles). Three different target positions are shown with black diamonds; human prior knowledge was always in the form of Gaussian distribution of possible target locations (blue dots) centered on target location 1.</td>
</tr>
<tr>
<td>3.3</td>
<td>A robot $i$ senses the bearing $\Phi_k^{i,\theta}$ of a target $\theta$.</td>
</tr>
<tr>
<td>3.4</td>
<td>RSME values and standard deviation of the distance between true and estimated position of the target from a single robot in Scenario I as a function of number of particles in a particle filter.</td>
</tr>
<tr>
<td>3.5</td>
<td>Average time to find the target for each of the target locations (levels of accuracy of human prior knowledge) and values of the relative weighting parameter $\alpha$. The number of robots always includes a reference robot and the dotted line marks the standard deviation for search with a single reference robot. Error envelopes denote $\pm$ standard deviation.</td>
</tr>
<tr>
<td>3.6</td>
<td>Sample trajectories in blue of the reference robot with starting positions (blue square) and for three different target locations (black diamonds).</td>
</tr>
<tr>
<td>3.7</td>
<td>Average distance of the autonomous robots to the reference robot for different target locations and a range of $\alpha$ values and number of robots. The number of robots in the legend denotes the total number of robots in the mission including the reference robot. Error envelopes denote $\pm$ standard deviation.</td>
</tr>
<tr>
<td>3.8</td>
<td>Sample trajectories of the reference robot (solid blue) with a single autonomous robot (dot-dash red) with its starting position (red circle) for target locations 1.</td>
</tr>
<tr>
<td>Figure</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>3.9</td>
<td>28</td>
</tr>
<tr>
<td>3.10</td>
<td>30</td>
</tr>
<tr>
<td>3.11</td>
<td>31</td>
</tr>
<tr>
<td>3.12</td>
<td>34</td>
</tr>
<tr>
<td>3.13</td>
<td>34</td>
</tr>
<tr>
<td>3.14</td>
<td>36</td>
</tr>
<tr>
<td>3.15</td>
<td>37</td>
</tr>
<tr>
<td>3.16</td>
<td>38</td>
</tr>
<tr>
<td>4.1</td>
<td>42</td>
</tr>
<tr>
<td>4.2</td>
<td>43</td>
</tr>
<tr>
<td>4.3</td>
<td>45</td>
</tr>
<tr>
<td>4.4</td>
<td>46</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>4.5</td>
<td>Stitched mosaic image comprised of all 10 camera images captured in the search environment.</td>
</tr>
<tr>
<td>4.6</td>
<td>Custom extrinsic calibration board, comprised of two unique ArUco markers arranged along the same orientation and separated by distance L.</td>
</tr>
<tr>
<td>4.7</td>
<td>Target with Aruco fiducial in front, that the autonomous robot searched for during the experiment in the search environment.</td>
</tr>
<tr>
<td>4.8</td>
<td>The search environment was divided into 4 sections labeled A, B, C, and D where the target may be located.</td>
</tr>
<tr>
<td>4.9</td>
<td>Screen capture of the NASA TLX questionnaire that the participants were required to complete after every run with the robot.</td>
</tr>
<tr>
<td>4.10</td>
<td>Searching environment with experimental setup. The red 'X' is the robot location for conditions 2-4, the blue 'O' is the randomized robot location for condition 1 only and the filled in green squares are the boxes that obstruct the lab for condition 1 only as well.</td>
</tr>
<tr>
<td>4.11</td>
<td>Maps given to participant for conditions 1-4. Where (a) is for (No Map, No Target), (b) is for (No Map, Yes Target), (c) is for (Yes Map, No Target), and (d) is for (Yes Map, Yes Target).</td>
</tr>
<tr>
<td>4.12</td>
<td>Sample X trajectory vs time data of the tracker as well as the interpolated+EKF data. Tracker output (red), and the interpolated+EKF processed data (blue) is shown.</td>
</tr>
<tr>
<td>4.13</td>
<td>Sample Y trajectory vs time data of the tracker as well as the interpolated+EKF data. Tracker output (red), and the interpolated+EKF processed data (blue) is shown.</td>
</tr>
<tr>
<td>4.14</td>
<td>Sample trajectory where the participant was given incorrect information on target location for condition 4 (Yes Map, Yes Target). The blue square represents the robot starting point, the red square represents the true target location, the blue 'X' represents the robot end position, and the cyan 'X' represents the point at which the participant realized the target wasn’t where it was claimed to be and rotated 180.</td>
</tr>
</tbody>
</table>
4.15 Sample total user input maneuver percentage during the condition. Where the rows correspond to conditions, starting with the top row as condition 1 and the bottom row is condition 4, and the columns correspond to the participant. Maneuvers are divided up into 3 categories: Forward, Backward and Rotate. ................................................................. 62

4.16 Sample input speeds for the teleoperated robot. .................. 63

4.17 Absolute value of the sample input turn rate for the teleoperated robot. . . 63

4.18 Sample interpolated then EKF processed trajectories from the tracking system. The top row represents condition 1 and the bottom row represents condition 4. Each column corresponds to a participant. Where the blue square represents the starting point of the robot, the red square represents the target location, and the blue ‘X’ represents the robot end location. .... .. 64

4.19 Sample interpolated and EKF processed speeds of the teleoperated robot. . 65

4.20 Sample interpolated and EKF processed turn rate of the teleoperated robot. 66

4.21 Time to find for each condition. C1 data is scaled to represent time to find in a smaller environment and C4 represents the time to find prior to a 180 degree turn in the search path. ................................................................. 66

4.22 Input speed to robot platform over time with C4 representing the input speed prior to a 180 degree turn in the search path. .................. 67

4.23 Input turn rate to robot platform over time with C4 representing the input turn rate prior to a 180 degree turn in the search path. ............... 68

4.24 Speed of robot platform over time with C4 representing the speed of the robot prior to a 180 degree turn in the search path. .................. 69

4.25 Turn rate of robot platform over time with C4 representing the input speed prior to a 180 degree turn in the search path. .................. 69

4.26 Trajectories of autonomous robot with mutual information based control searching for a target in the search environment. Where the black circle represents the robot, the black line represents the orientation of the robot and magenta square represents the true target location. The blue dots represent the robot’s particle estimate of itself and the red dots represent the target estimate. The blue line represents the robot position from overhead tracker, and the cyan line represents the particle filter estimate of robot position. . . 71
LIST OF SYMBOLS

\( N_i \) Number of robots in the neighborhood
\( N_t \) Number of targets that robot \( i \) can see
\( N_p \) Number of particles used for particle filter
\( (x^i_k, y^i_k) \) Location of robot \( i \) at time step \( k \)
\( (x^\theta_k, y^\theta_k) \) Location of target \( \theta \) at time step \( k \)
\( \psi^i_k \) orientation of robot \( i \) in the world
\( \Phi^i_{k}^{\theta} \) angular position of target \( \theta \) w.r.t. robot \( i \)
\( r^i_{k,\theta} \) Range of target \( \theta \) with respect to robot \( i \)
\( X^i_k \) Simulated state of robot \( i \)
\( Z^i_{k,\theta} \) Bearing of target \( \theta \) with respect to robot \( i \)
\( Z^i_{k,\bar{h}} \) Bearing of reference robot \( \bar{h} \) with respect to robot \( i \)
\( v^i_k \) Linear speed of robot \( i \) at time step \( k \)
\( \omega^i_k \) Angular speed of robot \( i \) at time step \( k \)
\( \eta^i_v \) Process noise of robot \( i \)'s linear velocity
\( \eta^i_\omega \) Process noise of robot \( i \)'s angular velocity
\( \eta^i_z \) Process noise of robot \( i \)'s sensor
\( \beta^i_k \) Field of view (FOV) of robot \( i \) at time step \( k \)
\( \rho^i \) Visible range (VR) of robot \( i \)
\( r^i \) Radius of robot \( i \)
\( u_k \) control inputs speed and turn rate for robots
\( K_c \) gain for determine intensity of turn prior to collision
\( \alpha \) relative weighing parameter for varying search strategies
CHAPTER 1
INTRODUCTION

Multi-robot systems are particularly well suited for missions that involve spanning large unstructured environments such as search and rescue (SAR) missions and environmental monitoring [2, 3, 4, 5]. A multi-robot solution typically entails using robots that are inexpensive and controlled in a distributed manner [6, 7, 8, 9, 10]. As opposed to coordinated strategies that may require centralized control [11], distributed control of robotic members scale robustly to larger problems. Controlling multiple robots to solve challenging problems, however, is an open area of research with questions related to navigation, detection, and security that often require high levels of autonomy [12, 13, 14, 15]. In this context, human-robot interaction (HRI), and in particular, human-swarm interaction (HSI) is an upcoming field concerned with solving complex problems where humans can participate directly as a member of the swarm, or indirectly as a remote operator [12, 16, 17]. The inclusion of humans increases the capability of a robotic swarm by offloading the challenges associated with navigation, resource allocation, and security concerns.

In this thesis, we have specifically considered human inclusion into the robotic swarm in the form of teleoperation. Including humans as a teleoperator is attractive for use in unknown and unstructured environments. The applications where humans teleoperate can range from space, medical surgery, hazardous environments and search and rescue applications [18], [19], [20]. These applications can vary in terms of the degree and type of control with the teleoperator ranging from manual control, where the teleoperator fully controls all aspects of the system remotely, shared control where the system has some degree of autonomy but
the teleoperator can assist, to supervisory control where the teleoperator only supervises the system at a high level [21], [22].

The adoption of teleoperated robots during SAR missions in firefighting and on park trails has been motivated by demanding environments, the need for protective clothing that can severely increase physiological strain on the human, and the urgency to search a large area in the smallest possible time which cannot be done on foot [3, 23, 24, 25]. A teleoperated robot not only mitigates the physiological stress, but also reduces the chance of exposure to poisonous toxins, hazardous materials, and extreme temperature fluctuations, while giving the operator freedom and capability to explore large areas [23, 26, 27, 17]. A teleoperator may independently control a single or multiple robots [28], and interpreting their actions or beliefs entails programming additional mediation layers [29].

A significant cognitive benefit of human involvement during SAR missions is that humans are bound to carry some form of prior knowledge with an associated confidence level [30, 26]. This knowledge may be related to the environment such as building layouts [31, 32] or could be related to the target being rescued and based on experience such as typical locations to find a missing person in a firefighting scene [31], general behavior of how a person may move when lost in a park, or platform specific such as how to operate a robot [1, 25, 33, 26]. Despite the benefits of such prior knowledge, how it may be exploited by autonomous robots during a mission is an open question [34]. It is possible for example that the prior knowledge creates an internal belief map that initially drives the human searcher to steer towards a particular location within the search domain [30]. However, how this prior knowledge must be translated into an actual control strategy that can be assigned to robots is an open question [35]. Relatedly, the accuracy of this prior knowledge, which the human may be aware of, could play a significant role in how exactly each robot should assist the human.
1.1 Literature review

Recent work on human robot interaction that explicitly accounts for human participation on an equal basis with the robots include mixed initiative systems, where robotic agents and humans collaborate while switching between independent and human-assigned tasks [36, 37], and policy blending setups where the control strategy involves calculating a blend of human and autonomous input [38, 39].

While mixed initiative strategies demonstrate a clear advantage over exclusive roles in SAR and monitoring missions [36, 37] they are often designed to be task-specific and binary in collaboration decisions (the robots can either follow or lead), which makes it difficult to identify the optimal degree of collaboration as a function of task performance and human expertise unless a large number of experiments are performed. A recently proposed framework based on homotopy classes enables navigation decisions driven by resource sharing [40], however this approach is designed for complementary sharing tasks and may not account for situations where robotic assistance may be better utilized through working together. Policy blending approaches provide a relatively more flexible arrangement of combining human and autonomous robot inputs but its effective use requires connecting the blending strategy to human and robot interpretation of the situation [38, 39]. In this context, a control framework that (a) abstracts the mission in terms of low-level objectives, (b) is responsive to human knowledge of the domain, and (c) enables encoding multiple objectives in a single measure can serve to highlight optimal collaboration strategies in a wide variety of settings.

Information-based control, which relies on an information-theoretic interpretation of the mission goal, is an approach that has been found to be especially useful in abstracting complex scenarios into instantaneously measurable objectives [41, 42, 43]. This is because the control objective is encoded in the form of mutual information gain, as opposed to for
example minimizing the distance to an object in an unknown location or covering a region [41, 42]. Information-theoretic approaches, however, control the robots as they maximize mutual information with respect to a single entity—the target—and therefore need to be expanded for inclusion of humans in the loop.

With respect to teleoperation itself, it has been shown that teleoperators perform differently depending on skill levels and knowledge of the environment. As an example, experiments have been conducted to evaluate the operators teleoperation performance within a mixed reality environment [44], or by using a shared impedance control [45] to avoid obstacles in an environment or using a velocity obstacle with a haptic device for teleoperators [22]. These have shown that teleoperators with less experience than others, exhibit difficulty and are less efficient in navigation and control compared to teleoperators with experience on the system [44]. A quantification of teleoperator knowledge based on their actions can provide the basis for new human-aware swarm control strategies in the field. To the best of our knowledge, the role of teleoperators prior knowledge on the robot’s movement and control has not been explored.

1.2 Contribution

The contributions of this thesis are as follows:

1. We present a framework that builds on particle filtering and mutual information-based control to realistically capture human robot interaction in a search and rescue setting: the framework allows modeling human prior knowledge about the target position and the autonomous robot to preferentially assist the human by staying close and thus reducing target uncertainty in their proximity.
2. We utilize this framework to analyze the dependence of mission performance on the degree of robotic assistance provided to the human and the accuracy of human prior knowledge in two realistic scenarios simulated with parameters informed from literature.

3. We perform an experimental study to analyze the effect of human prior knowledge search task efficiency and teleoperation.

Material from this thesis has been used in the following peer-reviewed publication:
CHAPTER 2
BACKGROUND

In this chapter we provide an overview of the particle filtering methods and mutual information based control that we utilize in formulating an HSI control strategy for search and rescue.

2.1 Particle filtering

Particle filters belong to the class of sequential Monte Carlo methods that are used to perform Bayesian estimation [46]. Unlike Kalman filters which expect linearized dynamics and measurement models of the underlying process, and assume additive Gaussian disturbance and noise, particle filters make no such assumptions and can be used directly with nonlinear motion models and non-Gaussian noise distributions [46].

In a particle filter, a continuous probability density function (pdf), \( p(X) \), of a random variable \( X \in \mathbb{R}^n \), is approximated by weights corresponding to \( N_p \) points in the sample space. These points are called particles. Accordingly, the probability of a point \( \tilde{X} \) is

\[
p(\tilde{X}) = \sum_{q=1}^{N_p} w_q \delta(X_q - \tilde{X}),
\]

where \( w_q \) is the weight of the particle \( q \), with \( q = 1, 2, \ldots, N_p \) as the particle index, and \( \delta(\cdot) \) is the Dirac delta function. The Dirac delta function \( \delta(X_q - \tilde{X}) \) acts as a kernel in that it sets the probability of values that are exactly the same as as one of the particles \( \tilde{X} = X_q \) as equal to the weight \( w_q \); at the same time, the probability of a value that is not
part of the particle representation \((\tilde{X} \neq X_q)\) is zero. Therefore, the only way to make this representation of a probability distribution more accurate is to use more particles.

A particle filter estimates a dynamic quantity \(X_k\) as it evolves according to a known dynamics and is sensed with a known measurement model. This is equivalent to maximizing the posterior pdf conditioned on all measurements up until the current time step \(k\), \(p(X_k|Z_{1:k})\), by recursively iterating through predict and update steps; here \(Z_{1:k}\) denotes all measurements up to \(k\). The predict step evolves the state of each particle through a motion model such as (3.1). The update step involves applying the Bayesian rule to calculate the probability of a state conditioned on a measurement obtained through (3.2). In a particle filter, this is accomplished by resampling the particles according to weights calculated as [46]

\[
w_{k,q} = p(Z_k|X_{k,q}),
\]

where the likelihood function \(p(Z_k|X_{k,q})\) captures the measurement model; for example if the measurement model (3.2) has Gaussian noise \(\eta\) with standard deviation \(\sigma_\eta\), then \(p(Z_k|X_{k,q}) = \mathcal{N}(Z_k, h(X_{k,q}, \theta_{k,q}), \sigma_\eta)\), where \(\sigma_\eta\) is the standard deviation of Gaussian noise \(\eta\).

Being amenable to nonlinear representations, a particle filtering framework is adequately suited to model human prior knowledge in the form of a pdf, nonlinear sensing with limited visual range, and measurement sharing between autonomous robots by combining likelihood functions.

### 2.2 Mutual information based control

Search and rescue operations entail finding a missing target. This amounts to identifying the control action out of a range of possible actions that actively maximize the posterior
pdf of the target location \( p(\theta_k|Z^\theta_{1:k}) \) [41], where the superscript \( \theta \) on \( Z \) denotes the target measurement.

Information theory provides an intuitive framework to actively search for a missing target. In an information theoretic sense, maximizing the posterior pdf is equivalent to minimizing the uncertainty in the target estimate. Within this framework information entropy is used to quantify the uncertainty in an estimate. With respect to target location \( \theta_k \), information entropy of the pdf \( p(\theta_k) \) is defined as \( H(\theta_k) = -\int_\Theta p(\theta_k) \log_2 p(\theta_k) d\theta_k \), and is measured in bits [47]. Therefore, maximizing the posterior pdf is equivalent to minimizing the entropy of target estimate conditioned on current measurement \( H(\theta_k|Z^\theta_k) \).

Minimizing the uncertainty of current target estimate, however, does not enable an active control strategy, which must act future possibilities. Therefore, we consider mutual information between the current target prediction denoted by \( \theta^-_k \) and possible target measurements \( Z^-_k \in \mathcal{Z} \), within a support \( \mathcal{Z} \), defined as [41]

\[
I(\theta^-_k; Z^-_k) = H(\theta^-_k) - H(\theta^-_k|Z^-_k)
\]

\[
= H(Z^-_k) - H(Z^-_k|\theta^-_k),
\]

where \( H(\theta^-_k|Z^-_k) \) for example is the entropy of \( \theta^-_k \) conditioned on knowing \( Z^-_k \). Based on this expression, we note that minimizing uncertainty \( H(\theta^-_k|Z^-_k) \) is equivalent to maximizing the mutual information \( I(\theta^-_k; Z^-_k) \).

Calculating mutual information is possible in a particle-filtering framework by computing the two quantities on the right hand side of equation (2.3b) as [41]
\[ H(\mathbf{Z}_k^\theta) = - \int_{\mathbf{Z}} p(\mathbf{Z}_k^\theta) \log_2 p(\mathbf{Z}_k^\theta) d\mathbf{Z}_k^\theta \]

\[ = - \int_{\mathbf{Z}} \left( \int_{\Theta} p(\mathbf{Z}_k^\theta | \theta_k)^p(\theta_k) d\theta_k \right) \log_2 \left( \int_{\Theta} p(\mathbf{Z}_k^\theta | \theta_k)^p(\theta_k) d\theta_k \right) d\mathbf{Z}_k^\theta \]

\[ \approx - \sum_{l=1}^M \left\{ \sum_{q=1}^{N_p} \left( p(\mathbf{Z}_{k,l}^\theta | \theta_{k,q}) \cdot \log_2 p(\mathbf{Z}_{k,l}^\theta | \theta_{k,q}) \right) \right\} \]

and as

\[ H(\mathbf{Z}_k^\theta | \theta_k) \]

\[ = - \int_{\mathbf{Z}} \int_{\Theta} p(\mathbf{Z}_k^\theta, \theta_k) \log_2 p(\mathbf{Z}_k^\theta, \theta_k) d\mathbf{Z}_k^\theta d\theta_k \]

\[ = - \int_{\mathbf{Z}} \int_{\Theta} p(\mathbf{Z}_k^\theta | \theta_k) p(\theta_k) \log_2 p(\mathbf{Z}_k^\theta | \theta_k) d\mathbf{Z}_k^\theta d\theta_k \]

\[ \approx - \sum_{l=1}^M \left\{ \sum_{q=1}^{N_p} \left( p(\mathbf{Z}_{k,l}^\theta | \theta_{k,q}) \cdot \log_2 p(\mathbf{Z}_{k,l}^\theta | \theta_{k,q}) \right) \right\} \]

where \( w_{k,q} \) is the weight of the \( q \)-th particle after resampling, and the support for measurement space \( \mathbf{Z} \) is made of \( M \) distinct values. Note that the likelihood function \( p(\mathbf{Z}_k^\theta | \theta_k) \) represents the probability of a measurement conditioned on a possible move by the target and the robot and therefore contributes to the effect of possible control input on the uncertainty of the target estimate. The control input is determined by solving the following optimization problem at every time step for a robot \([41]\)

\[ u_k = \max_{u \in U} I(\theta_k; \mathbf{Z}_k^\theta), \]
where $U$ represents the range of control inputs available to the robot. These range of inputs for example can be in the form of speeds and turn rates that the robot may apply at the next time step, or over a sequence of time steps [42].

Implementing this control on a target search scenario gives rise to several intuitive search strategies [41]: in the case of a bearings-only measurement model, multiple robots expectedly move towards the target in trying to minimize the uncertainty by reducing the distance between themselves and the target, and in the case of range-only measurements, multiple robots encircle the target to gain a wide range of perspectives.
CHAPTER 3
INFORMATION BASED CONTROL OF MULTI-ROBOT SYSTEMS

In this chapter we present the information based control of human-swarm robotic systems where the human teleoperates a single robot. We use the control strategy to investigate the role of weighting between two strategies—full assistance and independent search—on task performance across two widely different scenarios simulated to mimic real-world search situations.

3.1 A generic SAR setup for a human robot team

Consider a human robot team consisting of $N$ robots and a single target. All but one robots are autonomous. The human controls a single robot hereby called the reference robot. The state of a robot at a discrete time step $k$ consists of two dimensional position $(x_k, y_k)$ and orientation $\psi_k$ and is denoted by $X_k \in \mathbb{R}^3$. The target state is denoted by two-dimensional position $\theta_k$ within a two dimensional domain $\Theta \subset \mathbb{R}^2$. The kinematics of a robot $i$ can be represented by a nonlinear motion model $f^i$ as

$$X_{k+1}^i = f^i(X_k^i, u_k^i, \omega^i), \quad (3.1)$$

where $u_k^i \in \mathbb{R}_{\geq 1}$ is the control input, and $\omega^i$ is the disturbance, which captures our confidence in the motion model.
The target location, if it falls within the sensor range, can be sensed by both the autonomous and reference robots based on a known measurement model \( h^i \) that relates robot and target position to measurement \( Z^i_k \in \mathbb{R}^{\geq 1} \) by the robot as

\[
Z^i_k = h^i(X^i_k, \theta_k) + \eta^i, \tag{3.2}
\]

where \( \eta^i \) is additive noise. All robots are assumed to be able to communicate with each other throughout the search domain, although a reference robot may only be able to communicate its location.

Target kinematics are modeled using a constant velocity motion model as

\[
\theta_k = \theta_{k-1} + v \Delta t + \vartheta, \tag{3.3}
\]

where \( v \in \mathbb{R}^2 \) is the velocity, \( \vartheta \in \mathbb{R}^2 \) is the disturbance, and \( \Delta t \) is the length of the time step.

The human may possess prior knowledge about target location or dynamics and is expected to search accordingly. The goal of the human-robot team is to find the missing target in minimum possible time. This is equivalent to minimizing the uncertainty in target location estimate.

### 3.2 Information based search with prior knowledge

We seek to develop a robot control strategy which can assist a human with prior knowledge about target location or dynamics in a search and rescue mission. The human is expected to initiate their search according to this prior knowledge possibly favoring certain parts of the search domain; in contrast, all autonomous robots are expected to search through the domain without any location preference. To model robotic assistance in an information-
theoretic sense, we seek to formulate a control strategy that weights between an independent exploratory search of the environment and improving human’s search efficiency by reducing the uncertainty of target location within the neighborhood of the reference robot.

We assume that human prior knowledge can exist either in the form of a particular location within the search domain (for example, a location where witnesses say the victim was last seen prior to a fire) or in the form of dynamics (for example, a lost person will walk randomly from the position last seen with a speed of 1 m/s). We model human prior knowledge about location as initial target distribution \( p(\theta_0^h) \),

where the superscript \( h \), whenever present, is used to denote variables related to reference robot.

The form of \( p(\theta_0^h) \) depends on the type of knowledge, which may be about a particular location within the domain, in which case it can be modeled as a Gaussian distribution, or about a certain region within the search domain, in which case it can be modeled as a uniform distribution within specific bounds contained inside the search domain. In contrast, the initial target distribution \( p(\theta_0^i) \) of autonomous robots, who do not have such prior knowledge is uniform through the search domain. Human prior knowledge about target dynamics is modeled by assigning values to parameters within the constant velocity motion model in (3.3) such as velocity \( v \) or disturbance \( \vartheta \). Target search by the human is accomplished using the mutual information based control (2.6) so that the reference robot naturally explores a preferred target location.

For the human robot collaboration setup, we note that the human is not able to directly communicate with the autonomous robot beyond broadcasting its location. This for example represents a human controlled robot in a first-person view sense as in [12, 48]. Although it may be possible for a human to both teleoperate and communicate their beliefs to other robots, such a setup will likely put additional burden on the human as they continuously indicate a level of confidence they have regarding a particular location. Robotic assistance
here entails a search strategy that reduces target location uncertainty near the reference robot.

In particular, an autonomous robot can reduce local uncertainty near the reference robot, which likely has a further limited sensor range, by being in the proximity of the reference robot itself. In terms of a mutual information based control strategy similar to (2.6) this can be achieved by taking actions that reduce the uncertainty of the reference robot location. A distance-dependent measurement model will automatically enable actions to reduce uncertainty near the reference robot with reduction in distance. An example of a distance-dependent measurement is that from a bearings-only sensor. Accordingly, the goal of the information theoretic control strategy for an autonomous robot that only assists will be to equivalently minimize the location uncertainty of the reference robot conditioned on possible distance-dependent measurements, $H(X_k^{h,-}|Z_k^{i,h,-})$. This in turn amounts to maximizing the mutual information $I(X_k^{h,-};Z_k^{i,h,-})$, where $X_k^{h}$ denotes the reference robot’s two-dimensional position, and $Z_k^{i,h}$ denotes a distance-dependent measurement by the autonomous robot.

The relative weighting between the two types of strategies can be captured with a parameter $0 \leq \alpha \leq 1$; $\alpha = 1$ for example should enable searching for a target only thus deploying a ‘explore’ only strategy by the robot and $\alpha = 0$ should keep the robot close to the reference robot thus increasing the effective field of view of the human in a ‘exploit’ only strategy with the human. Accordingly, the combined mutual information function is formulated as

$$u_k = \max_{u \in U} \left[ \alpha \hat{I}(\theta_k^-;Z_k^{i,\theta,-}) + (1 - \alpha)\hat{I}(X_k^{h,-};Z_k^{i,h,-}) \right],$$

(3.4)

where, for example

$$\hat{I}(\theta_k^-;Z_k^{i,\theta,-}) = \frac{I(\theta_k^-;Z_k^{i,\theta,-})}{H(Z_k^{i,\theta,-})}$$
represents the normalized value of the mutual information of that target searched in the entire domain. Normalization of mutual information with respect to measurement uncertainty allows for meaningful comparison of strategies from two widely different sources of measurement: the target, who likely has random dynamics and whose location is unknown until it falls within a limited sensing range, and the human, who may have relatively more deterministic dynamics and whose location is known throughout the domain. Figure 3.1 below, shows what the mutual information of the target as well as the reference robot looks like before normalization.

![Mutual Information Graph](image)

**Figure 3.1:** Sampling showing un-normalized mutual information values for comparison between the reference robot and the target, from the perspective of the autonomous robot.

The algorithm below summarizes the mutual information based SAR implemented using particle filter in this paper.

We note that even though the control (3.4) is designed for an autonomous robot to weight between two strategies, the formulation can be extended to a multi-strategy setup (such as for example following multiple members of a human team based on their level of expertise and prior knowledge) by attaching separate weights to each strategy. We analyze
### Algorithm for HSI-SAR with mutual information

**Input:** Motion model, $f^i$, and measurement model, $h^i$ for all robots, target dynamics, relative weighting $\alpha$

**Initialize:** target estimates of human $p(\theta_0)$, and robot $p(\theta_i)$, search start location

**Output:** Mutual information based control at a time step $k$, $u_k$

For each time step $k = 1, 2, \ldots$

For each autonomous robot $i$

1. Update target estimate, $\theta_k$, and reference robot estimate, $X^h_k$, by resampling particle weights according to (2.2) and sensor model $h^i$
2. Calculate optimal control $u_k$ according to (3.4) using $f^i$ and likelihood functions $p(Z_k|\theta_k)$ and $p(Z_k|X^h_k)$ in equations (2.5) and (2.4)
3. Apply control $u_k$ to move robot according to model $f^i$ and predict target and reference robot locations according to models (3.3) and $f^i$.

For the reference robot

1. Update target estimate, $\theta_k$, by resampling particle weights according to (2.2) and sensor model $h^i$
2. Calculate optimal control $u_k$ according to (2.6) using $f^i$ and likelihood function $p(Z_k|\theta_k)$ in equations (2.5) and (2.4)
3. Apply control $u_k$ to move the reference robot according to model $f^i$ and predict target location according to the model (3.3)

---

a two-strategy scenario in this work to allow an in-depth analysis of the dependence of task performance on relative weighting.

### 3.3 Scenario I: indoor environment

We utilized the mutual information based control to first evaluate the dependence of time to find a missing stationary target within an indoor environment on the relative weighting parameter $\alpha$ and the accuracy of human prior knowledge. Human prior knowledge was assumed to exist in the form of a preferred location for initial search within the domain.

We simulated differentially driven ground robots with sensor ranges and motion parameters...
informed by actual experiments from the literature [31, 42, 49, 27], which focus on search and rescue within indoor environments and firefighting operations. Specifically, we adopted the size of the domain from [31] and robot dynamics based on [42, 27]. The range of speeds and turn rates for the robot were based on the design from [31, 49, 42].

### 3.3.1 Setup

The indoor environment consisted of a $30 \times 30$ m obstacle free region with the target located in one of three different locations as shown in Fig. 3.2.

![Figure 3.2: Setup for SAR in Scenario I showing the reference robot starting position (blue square) and autonomous robots’ starting positions (red circles). Three different target positions are shown with black diamonds; human prior knowledge was always in the form of Gaussian distribution of possible target locations (blue dots) centered on target location 1.](image-url)
These locations were picked so that they are at same distance of 15 m from the search start location at the corner of the region and the only difference in time to find would be due to searching in the wrong part of the region. Human prior $p(\theta^0_p)$ was modeled as a Gaussian distribution centered at one of the locations with standard deviation $\sigma_\theta$.

Each autonomous robot maintained local estimates of location of the target and that of the reference robot by running an onboard sampling importance resampling (SIR) particle filter [46]; the reference robot maintained an estimate of target location only. All robots were modeled as differentially driven ground robots with the motion model

$$
x^i_k = x^i_{k-1} + v^i_k \cos \psi^i_{k-1} \Delta t + \omega^i_v \cos \psi^i_{k-1} \Delta t
$$

$$
y^i_k = y^i_{k-1} + v^i_k \sin \psi^i_{k-1} \Delta t + \omega^i_v \sin \psi^i_{k-1} \Delta t
$$

$$
\psi^i_k = \psi^i_{k-1} + \Omega^i_k \Delta t + \omega^i_{\Omega} \Delta t,
$$

(3.5)

where $\Omega^i_k \in \Psi$ denotes the turn rate and $v^i_k \in V$ denotes the speed, with $\Psi$ and $V$ denoting the range of possible turn rates and speeds; $\omega^i_v$ and $\omega^i_{\Omega}$ denote zero-mean Gaussian disturbances with standard deviations $\sigma_v$ and $\sigma_\Omega$, in the robot velocity and turn rate respectively, and $\Delta t = 1$ second is the length of a time step.

All robots also had the same limited-range bearings-only measurement model for sensing the target, with the reference robot having a limited field of view of 70° based on field of view as seen in pictures of similar experiments [31], and the autonomous robot having a full 360° field of view. The likelihood function representing the sensing of the target by all robots is

$$
p(Z^i_{k,\theta} | X^i_k) = \begin{cases} 
\mathcal{N} \left( Z^i_{k,\theta} ; h^i(X^i_k, \theta_k), \sigma^i_{\Theta} \right) & \text{if } r^i_{k,\theta} \in \rho^i \\
\mathbb{1}(\Theta) \setminus \rho^i & \text{if } r^i_{k,\theta} \notin \rho^i,
\end{cases}
$$

(3.6)
where $Z_{k}^{i,\theta}$ denotes bearing measurement of the target, $\mathcal{N}(Z_{k}^{i,\theta}; h^{i}(X_{k}^{i}, \theta_{k}), \sigma_{\eta}^{i})$ denotes a normal distribution with mean $h^{i}(X_{k}^{i}, \theta_{k})$ and standard deviation $\sigma_{\eta}^{i}$ sampled at $Z_{k}^{i,\theta_{k}}$, $r_{k}^{i,\theta}$ is the distance between the target and the robot, and $\mathbb{U}(\Theta) \setminus \rho^{i}$ denotes a uniform distribution sampled within the two-dimensional search domain that excludes a circular sensor range with radius $\rho^{i}$. The bearings-only measurement model $h^{i}$ (fig. 3.3)

![Figure 3.3: A robot $i$ senses the bearing $\Phi_{k}^{i,\theta}$ of a target $\theta$.](image)

relates the robot and target state to target measurement as

$$Z_{k}^{i,\theta} = \Phi_{k}^{i,\theta} - \psi_{k}^{i} + \eta^{i},$$

(3.7)

where $\psi_{k}^{i,\theta}$ is the robot orientation, and $\Phi_{k}^{i,\theta}$ is the angular position of the target with respect to robot position, and $\eta^{i}$ is the observation noise of the sensor.

A bearings-only sensor also serves as an effective distance-dependent measurement so that mutual information based control (3.4) may be computed at all times. We also note that the assumption of being able to sense the location of the telerobot throughout the domain is not a significant limitation in RF based communication in small domains [42]. The likelihood
function representing the tracking of a reference robot by an autonomous robot is also set to bearings-only sensing as

\[ p(Z_{k}^{i,h} | X_k^i) = \mathcal{N} \left( Z_{k}^{i,h} ; h^i(X_k^i, X_k^h), \sigma^i_\eta \right). \]  

(3.8)

where \( Z_{k}^{i,h} \) is the bearing measurement of the reference robot.

This setup assumes that estimates of the target and human robot location can be updated by sharing synchronous measurements and measurement models between autonomous robots within a communication range, \( \kappa^i \), which spans the entire domain in this Scenario. Accordingly the combined likelihood function for an autonomous robot is

\[ p(Z_{k}^i, \theta, Z_{k}^{i,h} | X_k^i) = \prod_{j \in \{ \kappa^i \}} \left[ p(Z_{k}^i, \theta | X_k^i) \cdot p(Z_{k}^{j,h} | X_k^j) \right]. \]  

(3.9)

where \( \{ \kappa^i \} \) is the set of all robots, including \( i \), that are within \( \kappa^i \) distance of \( i \).

Collision with other robots and the boundary of the environment is handled so that upon a collision, the robot is able to sense the side on which the collision takes place with respect to its heading. A collision on the left causes the robot to turn right and vice versa. Accordingly, upon a collision, the robots change their instantaneous turn rate to \( \Omega^i_k = -K_c \gamma^i_k \), where the gain \( K_c = 1.5 \) determines the intensity of the turn, and \( \gamma^i_k \) is the angle that the point of collision makes with the robot heading.

### 3.3.2 Simulations

Simulations were run to investigate the dependence of mission performance quantified by time to find the target on (a) the relative weighting parameter \( \alpha = \{0, 0.25, 0.5, 0.75, 1\} \), (b) the accuracy of human prior knowledge, and (c) the number of autonomous robots deployed.
in the search (0–2). We considered three levels of accuracy of human prior knowledge ranging from high accuracy so that the true target location aligned with the center of the Gaussian distribution which models the human prior knowledge about target location, to low accuracy where the target location is far from the center of the human prior (Fig. 3.2). In all cases the distance of the target from the start location was the same. The target was considered found when it fell within the sensor range $\rho^i\theta$ of any robot (autonomous or reference). Imperfect sensing was implemented by setting a probability of detection, $p_d^h = 0.77$ for the reference robot [50] and $p_d^a = 0.66$ for the autonomous robot, assuming that it is equipped with an advanced convolutional neural network based target detection [51]. Table I lists the parameters used to simulate this scenario.

Twenty simulations were run according to the the algorithm in Section 3.2 for each of the conditions and average time to search recorded in each case. The number of particles in the particle filter was determined on the basis of minimizing the variance of target estimate across ten runs in the same domain. Specifically, we ran a particle filter with the number of particles ranging between 100 and 1600 to search for a stationary target (fig. 3.4). The number of particles $N_p$ were selected as the value after which reduction in variance of position estimate was not significant [52].
Figure 3.4: RSME values and standard deviation of the distance between true and estimated position of the target from a single robot in Scenario I as a function of number of particles in a particle filter.

Simulations were initialized with each robot placed at random location near the (0, 0) location and oriented at an angle of between 0 and 45 degrees with the reference robot was always oriented at 20 degrees. Simulations that took more than 2000 seconds were stopped with the time to search recorded as the maximum value.
Table 3.1: Scenario I simulation parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Psi$</td>
<td>range of turn rates</td>
<td>${-0.25, 0, 0.25}$ rad/s</td>
</tr>
<tr>
<td>$V$</td>
<td>range of speeds</td>
<td>${0, .415, .833}$ m/s</td>
</tr>
<tr>
<td>$\kappa^i$</td>
<td>communication range between robots</td>
<td>30 m</td>
</tr>
<tr>
<td>$\rho^i$</td>
<td>sensor range for all robots</td>
<td>3 m</td>
</tr>
<tr>
<td>$L$</td>
<td>size of domain</td>
<td>$30 \times 30$ m</td>
</tr>
<tr>
<td>$v$</td>
<td>target velocity</td>
<td>0 m/s</td>
</tr>
<tr>
<td>$\sigma_\theta$</td>
<td>standard dev. of target motion disturbance</td>
<td>0.05 m/s</td>
</tr>
<tr>
<td>$\sigma_\bar{\theta}$</td>
<td>standard dev. of human prior</td>
<td>4.5 m</td>
</tr>
<tr>
<td>$\sigma_v$</td>
<td>standard dev. in robot speed disturbance</td>
<td>0.1 m/s</td>
</tr>
<tr>
<td>$\sigma_\Omega$</td>
<td>standard dev. in robot turn rate disturbance</td>
<td>0.01 rad/s</td>
</tr>
<tr>
<td>$\sigma_\eta$</td>
<td>standard dev. in target sensing noise</td>
<td>0.1 rad</td>
</tr>
<tr>
<td>$N_p$</td>
<td>number of particles in particle filter</td>
<td>600</td>
</tr>
<tr>
<td>$p^h_d$</td>
<td>detection probability for reference robot</td>
<td>0.77</td>
</tr>
<tr>
<td>$p^a_d$</td>
<td>detection probability for autonomous robot</td>
<td>0.66</td>
</tr>
</tbody>
</table>
3.3.3 Results

Figure 3.5: Average time to find the target for each of the target locations (levels of accuracy of human prior knowledge) and values of the relative weighting parameter $\alpha$. The number of robots always includes a reference robot and the dotted line marks the standard deviation for search with a single reference robot. Error envelopes denote $\pm$ standard deviation.

Figure 3.5 shows mission performance in terms of the time taken to find the target in terms of the relative weighting parameter $\alpha$ for three different levels of accuracy of human prior knowledge. We immediately note that the time to find the target for a single reference robot increases as the human prior knowledge becomes more inaccurate. With target location 3, the reference robot is never able to find the target within the limited time. Furthermore, if the human prior knowledge is accurate there appears to be no significant advantage afforded by the presence of autonomous robots. In all situations where human prior knowledge was inaccurate, the presence of an autonomous robot significantly improves performance.

As expected, when the human prior knowledge becomes inaccurate, we begin to see a dependence both on the relative weighting $\alpha$ as well as the number of robots. First, we note that dependence on $\alpha$ is amplified as prior knowledge becomes inaccurate. Specifically, with respect to relative weighting, when the human is accompanied with one robot and the prior...
knowledge is inaccurate by about 10 m, the time to find the target does not appear to change significantly, but does so when the prior knowledge is inaccurate by 20 m, where we see a clear U shape in the curve attaining a minimum at $\alpha = 0.25$.

With respect to the number of robots, the time to search lowers in both scenarios where human knowledge was inaccurate. In the case of target location 2, the time to find at $\alpha = 0.25$ drops to less than two-thirds when two robots are assisting than when only one robot is assisting the human; in the case of target location 3, the time to find drops less than half at $\alpha = 0.25$ before rising up again at $\alpha = 1$.

Figure 3.6: Sample trajectories of the reference robot searching for the target at three different locations. These trajectories exemplify how a human teleoperator may initially search for the target according to their prior knowledge even if it is inaccurate.

Figures 3.6 shows sample trajectories of the reference robot searching for the target at three different locations. These trajectories exemplify how a human teleoperator may initially search for the target according to their prior knowledge even if it is inaccurate.
Figure 3.7: Average distance of the autonomous robots to the reference robot for different target locations and a range of $\alpha$ values and number of robots. The number of robots in the legend denotes the total number of robots in the mission including the reference robot. Error envelopes denote ± standard deviation.

Figure 3.7 confirms the role of the relative weighting parameter $\alpha$ in terms of proximity to the reference robot. Specifically, on average autonomous robots tend to stay further from the reference robot with increase in $\alpha$ with for example the distance between a single autonomous robot and the reference robot increasing from $2.8 \pm 0.5$ m at $\alpha = 0$ to $7.5 \pm 2.3$ at $\alpha = 1$. This trend is maintained when the number of autonomous robots increase to two where they maintain approximately the same distance from the reference robot as $\alpha$ is increased. This reduction in distance to the reference robot as the weighting parameter decreases is an evidence of belief induced swarming where robots tend to come close to each other to reduce uncertainty.
Figure 3.8: Sample trajectories of the reference robot (solid blue) with a single autonomous robot (dot-dash red) with its starting position (red circle) for target locations 1.
Figures 3.8 and 3.9 show sample trajectories of two-robot and three-robot setup respectively, with one reference robot. Looking at Figure 3.8, We see that as the $\alpha$ parameter is increased from 0 to 1, the autonomous robot has an increased tendency search far from the reference robot. When the human prior knowledge is inaccurate (second and third rows from top), the benefit of partially weighting the control strategy with respect to the reference robot is evident as $\alpha$ is increased with the autonomous robot finding the target on its own (red cross appears near the target).
3.4 Scenario II: moving target in a park

In this scenario we evaluated the effect of accuracy of human prior knowledge and relative weighting parameter on the time to find a missing person in a large park; human prior knowledge is assumed to exist in the form of a probability of area (POA) in the search zone. The POA is a high-probability circular region centered at the position last seen (PLS) [53]. We simulated fixed-wing UAVs with sensor ranges and motion parameters informed by experiments from the literature [25, 49, 54], which focus on search and rescue in outdoor park environments. Specifically we utilized the same park environment as in [1], dynamics were based on the use of fixed-wing airplanes in [25], and the range of speeds and bank angles were based on [25, 55].

3.4.1 Setup

The park consisted of a $10 \times 8$ km region with the target last seen at a location marked ‘X’ (Fig. 3.10). Based on search and rescue strategies in the literature [53, 25], human prior knowledge, $p(\theta_0)$, was modeled in the form of a trail favoring distribution within a circular POA region centered on the PLS [53]. We note that lost person dynamics are complex and likely depend on a number of factors including age, motivation, energy levels, and topography [56, 33, 57, 53]. Here, we adopt a simple representation and assume that all these aspects can be captured in the form of disturbance $\vartheta$ of the target executing a random walk based on (3.3) that favors walking along a trail. In particular, the disturbance $\vartheta \in \mathcal{R}^2$ is sampled as from a two-dimensional Gaussian distribution $\mathcal{N}(0, \sigma_{tr})$, where $\sigma_{tr} = \begin{bmatrix} \cos \zeta & -\sin \zeta \\ \sin \zeta & \cos \zeta \end{bmatrix} \begin{bmatrix} \sigma_v \\ \sigma_v \sqrt{1 - \exp (-d_{tr})^2} \end{bmatrix}$, with $\zeta$ representing the local orientation of the trail,
\(\sigma_v\), the speed of the random walker, and \(d_{tr}\) the distance from the trail in meters; the variable \(d_{tr}\) is used to vary the eccentricity of the covariance ellipse of the random variable so that a walker on the trail will tend to move along the trail, however once off the trail, the tendency to move along the trail will decrease exponentially. The angle \(\zeta\) is computed by fitting a line to ten closest pixels on the trail.

Human prior knowledge distribution was created by weighting a uniform distribution with a blurred image of the map of the trails (in white) and rivers (in black). With a constant time between when the target was seen and the initiation of the search, this amounts to varying sizes of the \(POA\) (Fig. 3.10).

Figure 3.10: Setup for SAR in scenario II with search area map adopted from [1]. Both reference and autonomous fixed-wing UAVs start from position (red X). Trails are shown in dark brown and rivers in blue. Search areas corresponding to three different sizes of \(POA\) are shown as circles with the target location shown as a black diamond; human prior knowledge distribution of target location estimate is shown for one of the \(POAs\).

All UAVs were modeled as fixed-wing Dubins airplanes adapted to fly at zero pitch so that (fig. 3.11) [58]
Figure 3.11: A UAV $i$ senses the bearing $\Phi_{k}^{i,\theta}$ of a target $\theta$.

\[
x_{k}^{i} = x_{k-1}^{i} + v_{k}^{i} \cos \psi_{k-1}^{i} \Delta t + \omega_{v}^{i} \cos \psi_{k-1}^{i} \Delta t
\]
\[
y_{k}^{i} = y_{k-1}^{i} + v_{k}^{i} \sin \psi_{k-1}^{i} \Delta t + \omega_{v}^{i} \sin \psi_{k-1}^{i} \Delta t
\]
\[
\psi_{k}^{i} = \psi_{k-1}^{i} + \frac{g}{v_{k}^{i}} \tan \phi_{k}^{i} \Delta t,
\]

where $x_{k}^{i}$, $y_{k}^{i}$, and $\psi_{k}^{i}$ denote the position and orientation of the UAV, $v_{k}^{i}$ denotes the speed, $g$ is acceleration due to gravity, and $\phi_{k}^{i}$ is the commanded bank angle of the UAV at time step $k$ that is constrained to be less than $\frac{\pi}{4}$ radians, the max allowable bank angle of the aircraft [55], and $\Delta t = 1$ second is the length of a time step. As in scenario I all robots also have the same bearings-only measurement model, with the reference UAV having a limited field of view of $70^\circ$, and the autonomous UAV having a full $360^\circ$ field of view. All UAVs were set to localize themselves at every time step with GPS-like accuracy so that their estimate of their own position was updated to a Gaussian distribution centered on their true position with a standard deviation of 10 m.

Differently from scenario I, the UAVs do not avoid collision among themselves, as they can be made to fly at different heights. The UAVs are however steered to stay within the
search boundary based on the bearing $\gamma^i_k$ of the nearest point on the boundary within 400 m. Specifically, the instantaneous bank angle is set to $\phi^i_k = -K_c \gamma^i_k$, where $K_c = 1$ is the gain. Additionally, because RF range is limited for such large domains, we limit the communication range $\kappa_i = 500$ m [59].

### 3.4.2 Simulations

We investigated the dependence of mission performance quantified by time to find the target on (a) the weighting parameter $\alpha = \{0, 0.25, 0.5, 0.75, 1\}$, (b) the accuracy of human prior knowledge in terms of the three different POAs, and (c) the number of autonomous robots deployed in the search (0–7). The accuracy of human prior knowledge was varied according to the amount of disturbance in the target dynamics which in turn affected the size of the POA. In particular, we model the real target to move at a speed of 2 m/s from the PLS, which corresponds to the distance of 0.72 km ($\sqrt{3600 \times 36}$) [60] moved by a random walker in thirty-six hours. We then set the values of $\sigma_\theta = \{1, 2, 3\}$ m/s. The corresponding POAs were set to uniform distributions within circular regions with radii 0.36 km, 0.72 km and 1.08 km centered on the PLS. The position of the target was arbitrarily set at 0.64 km away from the PLS towards the West. Accordingly, the POA = 0.36 km corresponded to searching inaccurately within too small a region, the POA = 0.72 km corresponded to an accurate and efficient search, and POA = 1.08 km, where the UAVs will search a region larger than needed, corresponds to an inefficient search.

Twenty simulations were run according to the algorithm described in Section 3.2 for each of the conditions and average time to find the target recorded in each case. Simulations were run as follows: first the initial distribution of human prior was created by sampling from a uniform distribution within the POA. Next, the search was initialized with each
robot placed at random location near the PLS and oriented towards East. The reference robot was constrained to stay within the POA by utilizing a boundary avoidance control similar to the one used to keep all robots within the search boundary. As in scenario I, the target was considered found when it came within the sensor ranges with probabilities of detection, $p_d h = 0.77$ for the reference robot [50] and $p_i d = 0.66$ for the autonomous robot [51]. Simulations were run for one hour (3600 seconds) at which point they were stopped and the maximum time to find recorded.

Table 3.2: Scenario II simulation parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\phi_k^i$</td>
<td>range of commanded bank angles</td>
<td>${-\pi/4, 0, \pi/4}$ rad</td>
</tr>
<tr>
<td>$v_k^i$</td>
<td>range of speeds</td>
<td>${20, 30}$ m/s</td>
</tr>
<tr>
<td>$\kappa^i$</td>
<td>communication range for all robots</td>
<td>0.5 km</td>
</tr>
<tr>
<td>$\rho^i$</td>
<td>sensor range for all robots</td>
<td>80 m</td>
</tr>
<tr>
<td>$L$</td>
<td>size of domain</td>
<td>$10 \times 8$ km</td>
</tr>
<tr>
<td>$v$</td>
<td>target velocity</td>
<td>0 m/s</td>
</tr>
<tr>
<td>$\sigma_\phi$</td>
<td>standard dev. of target motion disturbance</td>
<td>${1, 2, 3}$ m/s</td>
</tr>
<tr>
<td>$\sigma_v$</td>
<td>standard dev. of robot speed disturbance</td>
<td>10 m/s</td>
</tr>
<tr>
<td>$\sigma_\eta$</td>
<td>standard dev. of target sensing noise</td>
<td>0.1 rad</td>
</tr>
<tr>
<td>$N_p$</td>
<td>number of particles in particle filter</td>
<td>1500</td>
</tr>
<tr>
<td>$p_{d h}^i$</td>
<td>detection probability for reference robot</td>
<td>0.77</td>
</tr>
<tr>
<td>$p_{d i}^i$</td>
<td>detection probability for autonomous robot</td>
<td>0.66</td>
</tr>
</tbody>
</table>


### 3.4.3 Results

![Graph showing average time to find the target for different sizes of POA as a function of the relative weighting parameter α.](image1)

**Figure 3.12:** Average time to find the target for the different sizes of POA as a function of the relative weighting parameter α. The dotted line marks the standard deviation for search with a single reference UAV. Error envelopes denote ± standard deviation.

![Graph showing average distance of autonomous UAVs to reference UAV for different sizes of POA as a function of the relative weighting parameter α.](image2)

**Figure 3.13:** Average distance of the autonomous UAVs to the reference UAV for different sizes of the POA as a function of the relative weighting parameter α. Error envelopes denote ± standard deviation.

Differently from scenario I, here the reference UAV explored a predetermined POA around the PLS. Therefore, we expected that unless the POA for reference UAV contained the target the searching with reference UAV only would never be successful. On the other hand, if the
POA was larger than needed, searching with reference UAV only could succeed but would be inefficient. Therefore, the lowest time to find the target with a reference UAV only would occur when the \((POA = 0.72 \text{ km})\) is just large enough to contain the target. We find this trend in our results (dashed lines in Fig. 3.12).

When autonomous UAVs were included in the search, we find a dependence on the relative weighting parameter \(\alpha\) when the search region is inaccurate and the number of UAVs are more than equal to four. In particular, when \(POA = 0.36 \text{ km}\), the lowest time to find was found with \(\alpha = 0.25\). In contrast when the search region was larger than needed, the only dependence on \(\alpha\) is noted when four robots are used and they remain close to the reference robot.

An increase in the number of UAVs amplifies the dependence on \(\alpha\) when \(POA = 0.36 \text{ km}\), highlighting the role of a combined strategy in situations where human prior knowledge may not be accurate and the best utilization of robotic teams may involve a strategy that is aware of the accuracy in the prior knowledge. Interestingly, we find a dependence on \(\alpha\) with eight UAVs even when the search region is accurate at \(POA = 0.72 \text{ km}\) with the lowest time to find attained when \(\alpha = 0.25\). This suggests that a large team is needed to effectively increase the sensor range of the human even resulting in higher performance even if the prior knowledge is accurate.
Figure 3.14: Sample trajectories of the teleoperated robot (blue solid lines) with a single autonomous robot (red dashed lines) for various POA sizes. Columns show change in $\alpha$, rows show change in POA size.
Figure 3.15: Sample trajectories of the teleoperated robot (blue solid lines) with three autonomous robot (red dashed lines) for various POA sizes. Columns show change in $\alpha$, rows show change in POA size.
Figure 3.16: Sample trajectories of the teleoperated robot (blue solid lines) with seven autonomous robot (red dashed lines) for various POA sizes. Columns show change in $\alpha$, rows show change in POA size.

Figures 3.14 and 3.16 show sample trajectories of two-robot, four-robot and eight-robot setup respectively, with one reference robot. Looking at Figure 3.16, we see that as the $\alpha$ parameter is increased from 0 to 1 (left column to the right), the autonomous robots just as in scenario I, also have an increased tendency search far from the reference robot. When the POA size increases (middle and bottom rows), the benefit of partially weighting the control strategy with respect to the reference robot is evident as $\alpha$ is increased with the autonomous robots, since the reference robot is not constrained within a small POA. At $\alpha$ values just over the 0, the swarm is more likely to locate the target, as oppose to the swarm having a $POA = 0.36$.

A limitation of this work is with respect to the reference robot control which is used to represent human input. It is possible that human prior knowledge is subject to target location and knowledge of the environment and possibly also changes with time. A realistic
model the role of a human in SAR missions therefore requires a data-driven approach. Such a representation of human knowledge and action would greatly benefit building human-aware swarm control approaches. In the next chapter we describe an experimental study to understand the effect of different types of prior knowledge on human teleoperation during search and rescue.
CHAPTER 4
EXPERIMENTAL STUDY

In this chapter we present the preliminary results of an experimental study conducted to investigate the dependence of prior knowledge on human teleoperating actions and robot dynamics in a search mission.

4.1 Introduction

Humans possess and act on prior knowledge when they are taking part in a time-intensive mission such as search and rescue [61]. Such prior knowledge can be about target location or dynamics or the immediate environment [62]. Because humans are also able to adapt to various situations it is likely that they can modify their knowledge about a search situation as they are monitoring an environment.

To provide efficient robotic assistance in search and rescue scenarios, a robot should be able to detect and act on such knowledge. However, communicating prior knowledge poses additional burden on the human during a time-intensive mission and therefore a method to infer prior knowledge directly from robot operator’s actions is a critical need. This study will conduct experiments to (a) test multiple hypotheses regarding the effect of prior knowledge about target location and environment on human operator actions and (b) provide data for modeling and detection of such prior knowledge from human actions. To provide context with the rest of the thesis work, we focus specifically on scenarios where humans teleoperate a robot to search for a missing target. We tested the following hypotheses:
H1) The time to find a missing target depends on the knowledge of environment and target location

H2) The existence of prior knowledge about environment and target location can be inferred from the movement of the teleoperated robot

Prior knowledge here involved target location and environment. To build prior knowledge of the environment, participants are provided overhead map of the environment and the experimental conditions are sequenced to allow building of an internal map of the lab. Knowledge of target location is created by informing the participants about high probability regions where the target may be location. Since these in turn would depend on trust levels on the experimenter themselves, we pose survey questions inquiring the level of trust in target location during the experiment. Finally, in order to put these results in the context of applications where autonomous and human-controlled robots must be distinguished in the field by other robots, we compare the movement of a teleoperated robot to an autonomous robot which searches for a missing target using a mutual-information based control strategy implemented on a particle filtered estimates of the target location.

4.2 Methods

4.2.1 Experimental setup

The experimental setup consists of a differentially driven robot (iRobot Create 2) with a webcam (Logitech, C920) attached using a serial connection to a micro computer (Raspberry Pi 4) shown in Figure 4.1. The indoor search environment is a large 9 meter wide and 18 meter long robotics laboratory room situated within the Engineering academic building.
The room consists of six large laboratory benches, several chairs and tables that serve as obstacles. Ten webcams (Logitech C920) mounted on the ceiling 4 meter high are used to track the position and orientation of the robots with a custom tracking system (programmed in Python and OpenCV), implemented on a dedicated computer (Ubuntu Linux 18.04 operating system, 16 GB Memory, 3.4 GHz processor). The robots were tracked in real time as they are navigated through the search environment. The robots are teleoperated using a desktop computer (Ubuntu Linux 16.04 operating system, 16 GB Memory, 3.4 GHz processor) and a wide display monitor (2560 x 1080 pixels resolution) located in a smaller (4 m x 6 m) control room adjacent but isolated from the search environment. A WiFi router (ASUS, Nighthawk) and receiver (TP-Link, Archer T3U Plus) on the robots is used to communicate between the desktop computer and the robots.

Figure 4.1: The iRobot Create 2 robotic research platform. A fiducial marker is placed on top of the robot to allow the robot to be tracked at all times.
4.2.1.1 Teleoperation

The robots are teleoperated from an external “master computer” in the control room. Communication between the robot and the master computer is enabled through Wi-Fi using socketing. Socketing, which uses Transmission Control Protocol (TCP) for data transmission, is a client-server framework used to allow communication between two different computers, and was implemented using the open source library [63]. In this case, we setup two pipelines for direct communication between the robot and the master computer. A flow chart of communication architecture is shown in Figure 4.2.

![Diagram](image)

Figure 4.2: Sample flowchart depicting the communication lines between the robot and the computer.

One pipeline is solely for the communication of the keyboard control data from the master computer to the robot through the on board micro computer. The controls forward, back, left, and right are performed using arrow keys. These are converted to wheel speeds as
follows: the computer determines which key was pressed on the keyboard then a timer is started, marking the beginning of the key press.

\[ u_t = \begin{cases} k_s t_{kp} & \text{if forward key pressed} \\ -k_s t_{kp} & \text{if backward key pressed} \\ -k_s t_{kp} & \text{if left key pressed} \\ k_s t_{kp} & \text{if right key pressed} \end{cases} \quad u_r = \begin{cases} k_s t_{kp} & \text{if forward key pressed} \\ -k_s t_{kp} & \text{if backward key pressed} \\ k_s t_{kp} & \text{if left key pressed} \\ -k_s t_{kp} & \text{if right key pressed} \end{cases} \tag{4.1} \]

where \( t_{kp} \geq 0 \) is the duration of time in seconds that a key has been pressed, \( k_s = 100 \) is a gain which determines how fast the robot will reach max wheel speed and \((u_t, u_r)\) are the left and right input wheel speeds in mm/s. The movement of the robot was restricted to one key press at a time. Robot wheel speeds are limited to \([-500, 500]\) mm/s. The computer encodes the left and right wheel command pair as ‘UTF-8’ encoded string to store each byte with 8 bits then sends the speeds to the robot where the robot decodes the received message also using ‘UTF-8’.

The other pipeline is dedicated for the web camera stream captured by the raspberry pi 4 micro-computer using OpenCV. Using open source libraries in python, called pickle [64], when an image is captured by the robot, it converts the image into a string of bytes then is sent back to the master computer. The master computer then re-formats the string of bytes back to the original image format, using the same python library. A custom python script was programmed to present a first person view from the robot along with instructions on how to teleoperate it, and indicate when the target was found. Figure 4.3 shows a screen capture of what a participant would see from the on board camera as they navigate the robot in the search environment.
Figure 4.3: Screen capture of what the participant would see during the experiment

The search target was a stuffed toy with a random numeric code pinned on it for confirmation of search as shown in Figure 4.4
Figure 4.4: Target with confirmation code pinned.
4.2.1.2 Tracking

![Figure 4.5: Stitched mosaic image comprised of all 10 camera images captured in the search environment.](image)

Robots are tracked within the environment with ten overhead cameras. A robot is tracked by placing a unique fiducial pattern called the ArUco marker. These markers can be tracked by a dedicated software library for OpenCV [65] to allow tracking them throughout the environment. However, prior to tracking ArUco markers throughout the environment, intrinsic and extrinsic calibration are needed in order to obtain robot position and orientation within a common inertial reference frame.

Intrinsic calibration is performed using the MATLAB to run the Camera Calibration toolbox [66]. In particular, fifty images of a 101 cm x 81 cm checkerboard pattern with square size of 11.4 x 11.4 cm were taken for each camera and imported into the toolbox. The toolbox extracts the corners in the checkerboard pattern, and outputs the focal length, the
principal point, and distortion coefficients of the camera. Extrinsic calibration is performed next to determine a global reference point. The extrinsic calibration is performed with the help of a custom calibration board that consists of two ArUco markers arranged on a plane and separated by a distance of 2 m (Fig. 4.6)

![Figure 4.6: Custom extrinsic calibration board, comprised of two unique ArUco markers arranged along the same orientation and separated by distance L.](image)

A third ArUco marker is placed on the floor below one of the cameras and serves to indicate the origin of the global reference frame. Once the global reference frame is set, starting with the camera closest to the origin of the world frame, we walk through the environment with the calibration board, making sure that it is seen by two cameras, one marker by each camera so that relative camera position and orientation can be calculated. The output of the extrinsic calibration is a transformation matrix for each camera with respect to the global frame which is typically placed in the upper right corner of the lab (Fig. 4.5).

The ArUco marker tracking system has the capability of tracking multiple markers in the environment and saving all information in a csv file. The information saved is as follows: ArUco ID, camera ID, time in seconds, three-dimensional position in meters, and three-dimensional pose in radians.
4.2.1.3 **Autonomous robot**

The autonomous robot was programmed with the same mutual information based control law described in Chapter 3, section 3.3.1. The major changes that were implemented was that the robot was programmed to obtain real-time updates of its position throughout the search environment from the tracking system. The communication between the tracking computer and the robot is done through Wi-Fi using socketing. The socketing was setup in a way that the autonomous robot initially sends the ID of the ArUco marker that it wants position information about, and receives the information approximately once every 1/10th of a second.

To enable real-time control and sensing onboard a microcomputer the entire code was rewritten from MATLAB to C++ and compiled using CMake [67]. The transition into C++ allowed the autonomous robot to operate much faster resulting in 5 orders of magnitude performance improvement thus enabling real-time operation.

The communication between the raspberry pi 4 micro-computer and the robot is enabled through serial connection implemented using an open source github library [68].

To allow automatic recognition upon having the target within the camera range, an ArUco marker was placed next to the missing target (Fig. 4.7)
Two collision handling approaches are implemented. First, the bump sensors that already exist on the iRobot create platform allowed the robot to maneuver away from objects upon mild collisions. However, collisions with thin objects such as the leg of a table caused the robot to easily get stuck. To mitigate this issue, a second collision handling approach involved creating a virtual boundary that corresponds to the map of the environment including obstacles boundaries. This map was available to the robot at all times.

4.2.2 Experimental conditions

The following experimental conditions were considered:
C1: (No Map, No Target), where the participant was provided neither a map of the environment, nor a target location. To further ensure that the participant did not associate any environmental features, several 0.3 x 0.75 x 1 m sized boxes were placed at different locations within the environment.

C2: (No Map, Yes Target), where the participant was given a sparse map (only the outer boundaries) of the environment, as well as the robot position and orientation and target location in the form of a “high probability” verbal cue.

C3: (Yes Map, No Target), where the participant was given a fully detailed map of the environment with the location and orientation of the robot, but not the target location.

C4: (Yes Map, Yes Target), where the participant was given a fully detailed map of the environment with the location and orientation of the robot, and the target location in the form of a “high probability” verbal cue.

C5: (Autonomous robot, Yes Map, No Target), an autonomous robot operating using mutual-information based control was given the detailed map of the environment but not the target location.
4.2.3 Experimental procedure

Figure 4.8: The search environment was divided into 4 sections labeled A, B, C, and D where the target may be located.

Participants for this study ($N = 24, 4$ Females) were recruited through flyers posted throughout the campus and through email announcements. The experiment was approved by the Institutional Review Board at Northern Illinois University under protocol HS21-0372. Participant exclusion criteria included being below 18 years of age, not having visited the robotics engineering laboratory in the past year, and performed classroom lab prior to the past year for more than six hours (typical time spent doing course labs). The last two exclusion criteria ensured that participants were not familiar with the laboratory layout and could therefore not assume prior knowledge of the map.

Before beginning the experiment, prior to the participants arrival, the participant is assigned a random ID to record all information anonymously. Upon arrival participants were asked to go through the consent form and sign them if they wish to proceed. Next, participants were trained to operate the robot while in the control room. We explained how
the robot will be controlled using the keyboard. We also allocated a time of 2-5 minutes for the participant to control the robot from within the control room maneuvering it to move past objects and also leave the room into the hallway if they so desire. The participant was free to end the training at any time after two minutes when they felt comfortable. The goal of the training phase was to ensure that any changes in robot navigation observed during the experiment were only because of knowledge of the unknown environment or target location.

Prior to the experiment the participant was informed that they will have to find a missing target located in a different room and that once they do they must enter a randomly generated 4-digit numeric code that they see on the target. If the codes matched, the target was considered found. They were also given instructions on what keys to press when they find the target. Once the target was found a NASA TLX questionnaire [69] was prompted on the screen posing questions related to workload related to the task (Fig. 4.9). After the experiment, participants were further requested to fill a post-experiment questionnaire regarding their levels of trust in the target location, prior teleoperating and video gaming experience, and degree of delay they experienced in operating the robot.

![Figure 4.9: Screen capture of the NASA TLX questionnaire that the participants were required to complete after every run with the robot.](image-url)
For condition 1, the target was placed in location B or C (See Fig. 4.8) with the robot placed in the other location on the opposite end of the environment. This placement was different from the remaining conditions to purposefully avoid learning the environment by providing a different starting location. To further reduce any chances of the participant learning the environment, we placed five large 1 x 0.5 m large cardboard boxes throughout the lab shown in Figure 4.10.

Figure 4.10: Searching environment with experimental setup. The red 'X' is the robot location for conditions 2-4, the blue 'O' is the the randomized robot location for condition 1 only and the filled in green squares are the boxes that obstruct the lab for condition 1 only as well.

For condition 2, the robot was placed in the center of the environment as shown by ‘X’ in Figure 4.11(b). All boxes placed to create an illusion of a different environment for condition 1 were put away for the remaining conditions.
Figure 4.11: Maps given to participant for conditions 1-4. Where (a) is for (No Map, No Target), (b) is for (No Map, Yes Target), (c) is for (Yes Map, No Target), and (d) is for (Yes Map, Yes Target)

The participant was also informed about the true target location noting that the target is located in that region with “high probability” (Fig. 4.8).

For condition 3, the participant was given a more detailed map of the search environment. For condition 4, the participant was again asked to search for the target with a detailed map and the target location. Target location was again noted with “high probability” in a given section of the map. However, unlike condition 2, approximately half of the trials were randomly selected to have the target located in a location at the opposite end of the room.
than where it was specified by the experimenter. This amounted to studying the effect of low confidence in prior knowledge of target location.

4.2.4 Data analysis

Tracking can be performed at a high sampling rate of up to 10 frames per second. For experiments, where we sought to simultaneously record an overhead video, this sampling rate dropped to approximately 2 frames per second. The tracking error computed by placing markers on the floor at known positions and orientation and comparing the tracker output with the same. The motion model of the EKF was modeled after a differentially driven robot as follows

\[
\begin{align*}
    x_k^i &= x_{k-1}^i + v_k^i \cos \psi_{k-1}^i \Delta t + \omega_v^i \cos \psi_{k-1}^i \Delta t \\
    y_k^i &= y_{k-1}^i + v_k^i \sin \psi_{k-1}^i \Delta t + \omega_v^i \sin \psi_{k-1}^i \Delta t \\
    \psi_k^i &= \psi_{k-1}^i + \Omega_k^i \Delta t + \omega_\Omega^i \Delta t,
\end{align*}
\]  

(4.2)

where \( \Omega_k^i \) denotes the turn rate and \( v_k^i \) denotes the speed that are available from the onboard measurements from the robot and subsampled to match the tracking data. \( \omega_v^i \) and \( \omega_\Omega^i \) denote zero-mean Gaussian disturbances with standard deviations \( \sigma_v \) and \( \sigma_\Omega \), in the robot velocity and turn rate respectively, and \( \Delta \) is the length of a time step = 0.5 seconds.

The measurement model H of the tracking system was a 3x3 identity matrix recording the measurements collected by the tracking system (4.3).
\[ H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \]  

(4.3)

Figure 4.12: Sample X trajectory vs time data of the tracker as well as the interpolated+EKF data. Tracker output (red), and the interpolated+EKF processed data (blue) is shown.

Figure 4.13: Sample Y trajectory vs time data of the tracker as well as the interpolated+EKF data. Tracker output (red), and the interpolated+EKF processed data (blue) is shown.
The noise covariance matrix $R$

$$R = \begin{bmatrix}
\sigma_x^2 & 0 & 0 \\
0 & \sigma_y^2 & 0 \\
0 & 0 & \sigma_\Psi^2
\end{bmatrix} \quad (4.4)$$

where $\sigma_x = 0.0418$ m, $\sigma_y = 0.0418$ m, $\sigma_\Psi = 0.0277$ radians are the errors of the tracking system measurements of $x$ location, $y$ location, and the orientation respectively. The noise matrix $R$, was determined based on the errors reported by the tracking system as part of two experiments described earlier.

The process noise covariance matrix $Q_k$ is

$$Q_k = \begin{bmatrix}
\cos \Psi_k \Delta t & 0 \\
\sin \Psi_k \Delta t & 0 \\
0 & \Delta t
\end{bmatrix} \begin{bmatrix}
\sigma_v^2 & 0 \\
0 & \sigma_\Omega^2
\end{bmatrix} \begin{bmatrix}
\cos \Psi_k \Delta t & 0 \\
\sin \Psi_k \Delta t & 0 \\
0 & \Delta t
\end{bmatrix} \quad (4.5)$$

Where $\Psi_k$ is the orientation of robot $i$ at time step $k$, $\sigma_v^2 = 1$ m/s and $\sigma_\Omega^2 = .1$ rad/s. In addition to trajectory data from the tracking system, the following data was collected: user control input at 10 Hz, video images from the overhead cameras at 2 Hz, and video images from the robot’s perspective at 10 Hz.

Despite low errors, the tracking output exhibited noise in when the robot went close to a table and occluded the marker, or if the robot was seen by multiple cameras at their periphery causing distortion errors; successful transformations through ten cameras also caused a cascading error which was most pronounced in the height, but not along the ground plane. To mitigate some of these problems the tracking output was filtered using an extended kalman filter (EKF) after interpolating the data so that tracking output was placed 0.5 seconds apart in time.
Prior to analysis, Condition 1’s time was scaled down because the initial starting location for condition 1 was not in the center of the environment as in condition 2-4. Furthermore, the robot, teleoperated by the participant, began at the exact opposite end of the environment, with obstacles. Accordingly, the time to find was scaled by the ratio of time to find the target by moving across the entire environment with obstacles over the time to locate the target from the center of the environment. We ran the robot 5 times for each robot starting point to determine the scaling factor to be 2.27.

For condition 4, we split the trial into two temporal sections based on when the participant searched for the target in the “high probability” region and when they realized that it was not there and searched within the rest of the environment. The time at which the participant realized that the robot was not to be found in the “high probability” region was marked on the basis of a consistent 180 degree turn along the length of the environment. Specifically, we took the positional data in the x-direction over time then smoothed it with a window size of 10 time steps. Once smoothed, we took the derivative of the smoothed positional data and located the time step that produced a 0 slope. The zero slope represents the location the participant maneuvered to change direction 4.14.
Figure 4.14: Sample trajectory where the participant was given incorrect information on target location for condition 4 (Yes Map, Yes Target). The blue square represents the robot starting point, the red square represents the true target location, the blue 'X' represents the robot end position, and the cyan 'X' represents the point at which the participant realized the target wasn’t where it was claimed to be and rotated 180.

To determine the dependence of teleoperation input on prior knowledge we calculated input speed and turn rate as commanded by the participant were calculated directly from the wheel speeds $u_{l,k}, u_{r,k}$, measured onboard the robot at 10 Hz. The wheel speeds were then converted to input speed and turn rate as

$$
v_k = \frac{u_{l,k} + u_{r,k}}{2},
$$

$$
\omega_k = \frac{u_{r,k} - u_{l,k}}{d}
$$

where $(u_{l,k}, u_{r,k})$ are the individual input wheel speeds at time step $k$, $d=0.3084$ m is the distance between the wheels of the robot, and $(v_k, \omega_k)$ are the linear and angular speeds of the robot at time step $k$.

Robot speed and turn rate were calculated filtered trajectory data $\hat{x}_k, \hat{y}_k$, and $\hat{\psi}_k$ as
\[ \hat{v}_k = \frac{1}{\Delta t} \sqrt{(\hat{x}_k - \hat{x}_{k-1})^2 + (\hat{y}_k - \hat{y}_{k-1})^2}, \]
\[ \hat{\omega}_k = \frac{\Delta \hat{\psi}_k}{\Delta t}, \]  

(4.7)

where \((\hat{v}_k, \hat{\omega}_k)\) are the computed robot linear and angular speeds calculated as the difference between successive time steps with a \(\Delta t=0.5s\). Similarly, \(\Delta \hat{\psi}_k\) is the change in robot orientation over time steps \(k\) and \(k-1\) where at time step \(k\) the robot turn rate \(\hat{\omega}_k\) is calculated as the difference between successive orientations calculated as \(\hat{\psi}_k = \tan^{-1}\left(\frac{\hat{y}_k - \hat{y}_{k-1}}{\hat{x}_k - \hat{x}_{k-1}}\right)\).

Statistical analyses were performed using two-way repeated measures ANOVA with prior knowledge about map or target location as independent variable and observables including robot speed, turn rate, input speed and turn rate, and time to find the target as dependent variable. The time to find was scaled for condition 1 to account for navigation in a different environment and the trajectory data for condition 4 was truncated to keep only the section of the trial when the participant followed the initial guess for target location. Significance was noted for \(p\) values less than 0.05. Posthoc comparisons were performed with Bonferroni correction. All analyses were performed in MATLAB.
4.3 Results

Figure 4.15: Sample total user input maneuver percentage during the condition. Where the rows correspond to conditions, starting with the top row as condition 1 and the bottom row is condition 4, and the columns correspond to the participant. Maneuvers are divided up into 3 categories: Forward, Backward and Rotate.

Looking at robot control, all participants seem to have favored moving forward compared to turning. The backwards maneuver was rarely ever used during the experiment by nay of the participants. A distinct rise is observed in turning movement in condition 4 when participants found that the target was not located in the position of “high probability” (last two columns in Fig. 4.15).
Looking at the input speed for conditions 2 and 4 of Figure 4.16 (rows 2 and 4), we see that there is a consistent delay in the action of moving forward or backwards, it is also longer.
in condition 2 compared to condition 4. However, looking at user input turn rates of Figure 4.17, we see that the participant is rotating either left or right to align themselves with they believe the target is located. Furthermore, the instances where the robot start-and-stopped are distinctly more in the first condition compared to second, and third compared to fourth when the participants did not have knowledge about the target location.

Figure 4.18: Sample interpolated then EKF processed trajectories from the tracking system. The top row represents condition 1 and the bottom row represents condition 4. Each column corresponds to a participant. Where the blue square represents the starting point of the robot, the red square represents the target location, and the blue ‘X’ represents the robot end location.
Looking at Figure 4.18, more specifically the top row (condition 1), all participants spent much of the time exploring the environment, looking for the target. However, looking at condition 2 (row 2 from top), when participants knew target location with apparent confidence, albeit without knowledge of local environment features, the trajectories immediately are directed towards the true target location.

Figure 4.19: Sample interpolated and EKF processed speeds of the teleoperated robot.
Similar trends are observed with respect to robot speed and turn rate as calculated from tracking data (Figures 4.19, 4.20).

Figure 4.20: Sample interpolated and EKF processed turn rate of the teleoperated robot.

Figure 4.21: Time to find for each condition. C1 data is scaled to represent time to find in a smaller environment and C4 represents the time to find prior to a 180 degree turn in the search path.
Figure 4.21 shows the average time to find the target across all conditions. One way repeated measures ANOVA with time to find the target as the dependent factor and level of prior knowledge as the independent factor revealed a significant effect \( F(3, 69) = 14.022, p < 0.001 \). Post-hoc pair-wise comparisons revealed that the participants in (No Map, No Target) took longer time to find the target (scaled) than when they knew the location of the target (No Map, Yes Target; Yes Map, Yes Target), but not when only the environmental map location was known (Yes Map, No Target). However, comparison between (No Map, Yes Target) and (Yes Map, Yes Target) reveals that knowledge of environment helps in reducing time to find. Finally, knowing the map only (Yes Map, No Target) had a significant effect on time to find than when target location was also known (Yes Map, Yes Target).

![Input speed to robot platform over time with C4 representing the input speed prior to a 180 degree turn in the search path.](image)

Figure 4.22: Input speed to robot platform over time with C4 representing the input speed prior to a 180 degree turn in the search path.

Figure 4.22 compares input speed across the experimental conditions. One way repeated measures ANOVA with time to find the target as the dependent factor and level of prior knowledge as the independent factor revealed a significant effect \( F(3, 69) = 11.547, p < 0.001 \). Post-hoc pair-wise comparisons revealed that the participants in (No Map, No Tar-
get), traveled slower than when they knew the target location (No Map, Yes Target; Yes Map, Yes Target), but not when only the environmental map location was known (Yes Map, No Target). Finally, knowing the map location and the target location (Yes Map, Yes Target) shows a significant effect on input speed when compared to unknown target location (Yes Map, No Target).

Figure 4.23: Input turn rate to robot platform over time with C4 representing the input turn rate prior to a 180 degree turn in the search path.

Figure 4.23 average input control rate across conditions. The plot shows how different the turn rates that the participants use to control the robots with. The lowest median turn rates correspond with the condition where the target location information was given. One way repeated measures ANOVA with time to find the target as the dependent factor and level of prior knowledge as the independent factor revealed a significant effect ($F(3, 69) = 11.462, p < 0.001$). Post-hoc pair-wise comparisons revealed that the participants in (No Map, No Target), rotated more than when the target location was known (No Map, Yes Target; Yes Map, Yes Target). Finally, knowing the map environment (Yes Map, No Target) effects the input turn rate when compared to not knowing the map environment (No Map, Yes Target).
Figure 4.24: Speed of robot platform over time with C4 representing the speed of the robot prior to a 180 degree turn in the search path.

Figure 4.25: Turn rate of robot platform over time with C4 representing the input speed prior to a 180 degree turn in the search path.

Figures 4.24 and 4.25 compare robot speed and turn rates across conditions. One way repeated measures ANOVA with the robot speed and turn rate as the dependent factor and level of prior knowledge as the independent factor revealed a significant effect ($F(3, 69) =$
12.472, \( p < 0.001 \) for tracker measured speed but no significant effect (\( F(3, 69) = 2.4651, p = 0.06953 \)) for tracker measured turn rate. Post-hoc pair-wise comparisons revealed that the participants in (Yes Map, Yes Target), traveled faster than in any other condition (No Map, No Target; No Map, Yest Target; Yes Map, No Target).

Figure 4.26 shows the autonomous robot navigating throughout the environment over time, searching for the missing target. Two trial runs of the autonomous robot were conducted, searching for the missing target in the same location, while the robot began in the middle of the environment at the same location where participants started in conditions 2-4. One trial run resulted with the autonomous robot finding the target in 66.7 seconds, while the second trial run resulted with 170 seconds for the autonomous robot to find the target in the environment.

### 4.4 Discussion

Speed data of individual trials show that participants frequently start and stop the robot, although this behavior was more common in trials where target location was not known. A similar trend is apparent when looking at the turn rates, where the participants turn many more times when they had no knowledge of target compared to when they had some knowledge of where the target was located. This behavior in commanded inputs is replicated in trajectory data and can be viewed as exploratory. When target location was known, participant commanded inputs demonstrate a significant waiting time at the beginning of the trial where the participants did not command the the robot to move forward. It is likely that the participants rotated in place to gain their bearings prior to searching for the target.
Figure 4.26: Trajectories of autonomous robot with mutual information based control searching for a target in the search environment. Where the black circle represents the robot, the black line represents the orientation of the robot and magenta square represents the true target location. The blue dots represent the robot’s particle estimate of itself and the red dots represent the target estimate. The blue line represents the robot position from overhead tracker, and the cyan line represents the particle filter estimate of robot position.
The lowest times to find correspond to the conditions where the participant was given target information. When target location was unknown, knowledge of the map did not significantly reduce the time to find. This is expected since situations where target location is not known will likely trigger exploratory behavior and knowledge of the environment can help avoid unnecessary obstacles, however, unless the environment is laid out such that knowledge of the map allows for more efficient search, it may not manifest in quicker search. At the same time, when the target location was known, knowledge of the map helped reduced the time to find (prior to performing a 180 degree turn in condition 4). This demonstrates a distinct improvement in how participants were able to efficiently find the target by accounting for obstacles.

The commanded input speed during search varied with prior knowledge. When the participants had no prior knowledge (environment or target location) they moved with the slowest speed, which increased more when the target location was known compared to when the environment was known. This is accompanied by analogous change in commanded turn rates indicating exploratory behavior when the target location was not known.

Similar trends are observed when comparing robot speed from trajectory data but not when turn rates are compared. This can be attributed to the robot not having ideal motors, where one wheel may rotate slower or faster than the other wheel and thus the commanded turn rates do not match the actual turn rate as observed by the tracker. It is also possible that because the turn rate from the tracker is observed at a much lower sampling rate of 2 Hz compared to 10 Hz from onboard the robot. Finally, although a majority of the experiments have been completed, the approved experimental study allows for six more participants. Therefore, it is possible that some of these results may alter as the sample size is increased.
Comparing the limited trials of the autonomous robot searching performance to the condition 3 (Yes Map, No Target) of the preliminary results of the human-subject experiment in terms of time, reveals that the autonomous robot’s time to search for the target within the environment is roughly the same as the participants. However, more experiments for the autonomous robot are needed to establish a full data set to get a more accurate representation of search time.
CHAPTER 5
CONCLUSION

In this thesis, a weighted information-theoretic control was formulated to evaluate the effectiveness of multiple robots as they vary their degree of assistance to human. A particle filtering framework allowed for a realistic representation and Bayesian update of human prior knowledge as well as limited-range sensor models while searching for a missing target. Simulations in two widely different scenarios with parameters informed from experiments in the literature revealed a dependence of mission performance on the weighting parameter as well as the accuracy of human prior knowledge. Specifically, when the human prior knowledge is inaccurate, both in terms of location or the size of the region to search, robotic assistance is effective if autonomous robots weigh between complete independence and human assistance.

To gain a better understanding of how humans perform search using teleoperation, a ground robot teleoperation platform was developed for a human-subjects experiment. Preliminary results from the human-subjects experiment reveal that both knowledge of the environment and target location affects the searching efficiency of the teleoperator. During conditions when the participants were not given target location information, participants spent most of the time exploring the environment searching for the target. While in conditions where target location information was given, participant control inputs revealed higher speeds and lower turn rates, likely due to improved confidence in the search route.

There are several directions where this work can be continued in the future for formulating optimal human-aware search strategies. In the simulations we assumed initially a static value of the relative weighting parameter, however, it is possible that a time-dependent $\alpha$ may constitute an optimal search strategy that is also responsive to human awareness of
the mission. Implementing such a strategy will require online estimation of human behavior and intent, a growing area of research within human-robot interaction [70], [71]. Another direction that can be pursued is with respect to tuning the interaction rules between autonomous robots towards a certain objective such as maintaining a particular formation [72] or by adaptively sharing information with select neighbors as has been shown to improve performance in particle swarm optimization [73].

With respect to the experiments, increasing the complexity of the search environment can highlight further differences in search behavior. Finally, using the autonomous robot test bed, a more realistic model of prior knowledge of the target location could be determined by varying the distributions until a match is obtained between the autonomous robot behavior to human search behavior.
REFERENCES


