CONTROL ALGORITHMS FOR *SPACE TUG* RENDEZVOUS

Final Report

16.622

Spring 2003

Authors: Gergana Bounova - Timothée de Mierry

Advisor: Olivier L. de Weck

Tuesday, May 13^{th}, 2003

Abstract

This project sought to study the efficiency of three possible two-dimensional search strategies. In the context of a broader undergoing *Space Tug* project sponsored by the Defense Advanced Research Projects Agency (DARPA), there were a number of modeling challenges that needed to be met in order to validate the experimental results. Probably the most crucial problem was to reduce the complex space problem involving at least four degrees of freedom to a simple two-dimensional model. The goal of the project was to run two-dimensional searches for an inert target using LEGO robots. The data collected using the robots, as well as results from simulations, showed strong trends in the relationship between time and energy expanded during the search sequence. The project provided a starting point for the rendezvous control system to be implemented in the *Space Tug* by showing which of the three strategies - random, semi-autonomous and autonomous - is most efficient in the two-dimensional case. It was found that the semi-autonomous algorithm is the most energetically efficient approach, but the most time-consuming. This finding disproved the hypothesis which stated that the semi-autonomous strategy is the most efficient in terms for both time and energy. Instead, the conclusion is that an autonomous algorithm is more suitable for space applications. The results also suggest that, depending on knowledge of the search space and the mission requirements, a hybrid approach might be more efficient. With knowledge of orbital dynamics, the meaning of these results can be extended to the space problem.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Background and Motivation

During the past decade there has been an increasing need to service vehicles in space. Satellites and entire missions have been lost due to misalignments, wrong placements in orbit and software errors. These problems have triggered efforts to design a new type of vehicle called the *Space Tug*, or the Orbital Express. The *Space Tug* is a satellite that carries out rendezvous and docking with a target satellite. Its functions are to capture targets, change their position and orbital elements by a preset amount and release them at destination safely. The *Space Tug* has to be capable of orbital debris removal, satellite rescue missions and tactical operations.

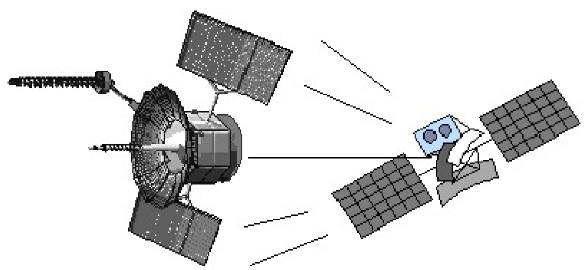


Figure 1: The *Space Tug* moving a broken satellite

The *Space Tug* project is an ongoing MIT/DARPA research project that aims to develop a satellite - the Tug - to carry out rendezvous and docking with a target satellite. The mission of the *Space Tug* will be to capture its target, change its position and orbital elements by a predefined amount, and release it without damage to the target satellite or itself. The capabilities of the tug must include the following: rescue satellites from unusable orbits, orbital debris removal, tactical operations, and other emergent uses. The control system of the tug used to find and dock with the target is a major aspect of this project. The search for the target satellite is a complex procedure. Although approximate coordinates for its location would be provided, the tug would still have to search a finite space to find it, since current tracking does not give precision below a magnitude of the order of one hundred meters. As a result, the *Space Tug* has to have intelligent identification and sensing strategies implemented in its control system to approach the target.

A major technology risk in the *Space Tug* project is the target identification and docking. Showing that such a process is feasible would reduce this risk and would provide a possible solution to the problem. In addition, given that the control system of the *Space Tug* has thus far been modeled as a black box, the results of this research project could give clues as to what the architecture of the control system of the vehicle should be. Furthermore, successful search and rendezvous strategies could be used in other aeronautical applications, such as autonomous and formation flight.

Search strategies and algorithms for robots have been studied extensively in the past years. As a result, the design part of this project was influenced by previous work done on search algorithms, notably by the MIT Department of Electrical Engineering and Computer Science. The basic strategies were enhanced to fit the purpose of the *Space Tug*. Time and energy consumption, as well as the successful implementations of three searching strategies, are the focus of the research project. It provides a first look at the most effective rendezvous strategy that could potentially be used in the *Space Tug* project. Spanning the space of possible search algorithms, three strategies were tested: (1) random sensor-less search, (2) semi-autonomous with a human in the loop search and (3) fully autonomous search with sensors. Ultimately, the goal of this project was to show that the semi-autonomous with a human in the loop search will satisfy this important requirement.

1.2 Hypothesis

The use of a semi-autonomous search system with a human in the loop is the algorithm that is the most effective for rendezvous and docking strategies in terms of time and energy consumption.

2 Objectives and Success Criteria

The project is divided into two distinct parts that achieve two different but closely related goals. The preliminary objective is to develop, implement, and test three different strategies for two-dimensional, non-cooperative target search and precise docking. This goal is the first step in achieving the primary objective.

The primary objective of the experiment is to evaluate a cost function in order to compare these three strategies based on the trade-off costs between time and energy. The algorithms are evaluated using predefined criteria, developed into the cost function, to examine the performance of each.

Success for this project is a clear definition of whether or not the semi-autonomous search system is the most effective algorithm for rendezvous and docking strategies in terms of time and energy consumption. The three different search strategies span the full space of search algorithms in an effort to provide a valid assessment of the hypothesis.

3 Previous Work

3.1 Literature Review

The article entitled "On-board software for the Mars Pathfinder Microrover", written by Morrison and Nguyen [1] describes the software used to control the motion of the rover on Mars. The constraints, in terms of communication and energy, on the control system of the Mars rover are similar to what the tug will face in space. It is thus important that these constraints be taken into account when modeling the searching procedure. Due to electrical and processing power limitations, the control system of the rover is unable to communicate and move at the same time. In addition, due to communication restrictions, mainly the time it takes to transmit information from Mars to Earth and back, the control system of the rover uses waypoint navigation and autonomous collision avoidance algorithms. In the absence of any obstacles, the rover proceeds directly forward to the waypoint, including stops for proximity scanning - for hazard detection. During proximity scanning processes, the rover uses its on-board optical sensors to generate an approximate map of the terrain map in front of the vehicle. Based on height differences in the map, the navigation system analyzes the possible locations of obstacles. Finally, an alternate working mode of the control system is the "rock finding" option, which uses the terrain map to detect a rock. The navigation system corrects the rover heading, centering it between the rock edges.

The Mars Pathfinder rover uses a collision-avoidance control system. This technique is the opposite of what has to be developed for the *Space Tug*. While the rover uses the terrain map to trace a route around objects, the tug will have to trace a route to its target. Although this paper is not very useful in describing the actual search algorithms, it provides a background for the type of software architecture that is usually used in space vehicles. The same power and communication constraints apply to the tug, and therefore its on-board software needs to make use of the same techniques for telemetry, which are necessary for the semi-autonomous search. Furthermore, the proximity scanning process implemented in the Mars Pathfinder is similar to what needs to be developed for the fully autonomous version of the tug. Although this paper is general and does not provide more detailed information about the core of the software, it describes a basis for the architecture for the control system that will be used on the tug.

The second reference is a paper by Gelenbe [2] entitled "Autonomous search for information in an unknown environment", which describes different search strategies from a "computer science" point of view. The author models the autonomous process in which an agent, a robot or a software algorithm searching for information in a computer database, searches the space around its current location for information it wants. The search area is divided in a set of locations (x, y), defined in a Cartesian space. Associated with each location is a probability q(x, y) representing the likelihood of finding the information wanted at this location. Assuming the environment is static, the space can thus be described as a probability space. The agent, which in the context of this project is the tug, always moves in the direction where the probability q(x, y) is the greatest. Once the agent moves to the new location - from (x0, y0) to (xnew, ynew) -, the probability q(xnew, ynew) of finding information at the new point is updated depending on what was found. The algorithm thus continuously updates the probability space, until the agent finds the right information - the target for the tug. The paper further develops a more advanced model which is more applicable to information search in a computer system than to robotic search.

The above search algorithm, referred to as the "Greedy Algorithm" by Gelenbe [2], is relevant to the random search strategy that needs to be implemented in the tug simulator. The algorithm that will be used in the project will most likely incorporate some or all components and rules of the Greedy Algorithm. While the results of the experiments are not significant for the project, the modeling process used by Gelenbe in his experiment is useful in developing the model of the search space for the tug. The mathematical tools used in the Greedy algorithm can be applied to the tug's random search strategy, since the underlying probabilistic decision-making processes are similar (e.g. the agent goes to the location with the greatest probability in the space). Furthermore, the same principles can be used in the fully autonomous search. The main difference is that the agent, which has sensors with a given range R, can now check for information in a space of radius R around the location (x, y). In this process, a greater number of probabilities can be updated to recalculate the space. In addition, the agent is also able to build a map of the environment revealing the exact location of the information with greater precision. Gelenbe [2] falls short of developing a smarter algorithm and explaining how the sensors would affect the efficiency of the search.

The third most relevant article to the project is by Hillenbrand and Hirzinger [3], and is entitled "Probabilistic search for object segmentation and recognition". Object recognition is viewed as a two part process. Firstly, a sequence of hypothesis about the object - its location, geometric shape, possible movement - is generated, using exterior sensors. The second part of the process evaluates these hypotheses based on the object model. This paper describes a new technique for object recognition in a specific scene in a probabilistic framework. It also introduces a new statistical criterion - the truncated object probability - to produce optimal hypotheses about the object to be evaluated for its match to the data collected by sensors. The author further develops a mathematical model to fit the search sequence in the experiment.

The depth in which this article goes is most likely beyond the scope of the *Space Tug* project. However, some of the concepts developed are useful for the autonomous search strategy to be implemented in the tug. Based on the data from its sensors, the tug should be able to recognize the target in a largely unknown scene.

The object recognition technique developed by Hillenbrand and Hirzinger [3] is too advanced to use in an environment with one target. However, if implemented in the control system of the autonomous tug, it will be an expandable algorithm that can, for example, be slightly altered to recognize multiple targets in motion.

The second reference related to the Mars Pathfinder mission is on crater and rock hazard modeling for Mars landing [4]. It describes measures for safe landing on the rough and hazardous terrain of Mars. The investigation examines simple models of crater size-frequency distribution, rock size-frequency distribution and scaling relationships to determine the hazard probabilities and choose landing terrains. The approach to hazard modeling and navigation generated a useful idea. A search of satellite sizes and geometries was performed in order to scale the search space for the experimental setup [4]. For example, a 100-meter radius in space with a 2-meter long target translates to a 5-meter experimental radius with a 10-centimeter robot target. This determines the grid size to search space diameter ratio which impacts the choice of search strategies. Thus, the Pathfinder rock-frequency distribution model helped focus the search space domain while retaining the validity of the experiment.

The four analyzed references show that the field of search algorithms is heavily explored. At the same time, the lack of an exact match to the problem at hand demonstrates that the *Space Tug* application has not been modeled previously but that one can gain by utilizing technologies from other domains.

3.2 Applicable Theory

The application of this experiment depends on the extendability of the obtained results. This demands proper reduction of the 3-dimensional space problem to the 2-dimensional grid search experiment. The main difference between reality and the model experiment is the number of degrees of freedom. All results obtained in this *Space Tug* experiment are valid only for target search in two dimensions. In order to make a conclusion, it is necessary to either find a reasonable reduction of the space problem to 2D or extend the meaning of the experimental results to three dimensions. In this case applicable knowledge of orbital dynamics makes the first approach feasible and more suitable.

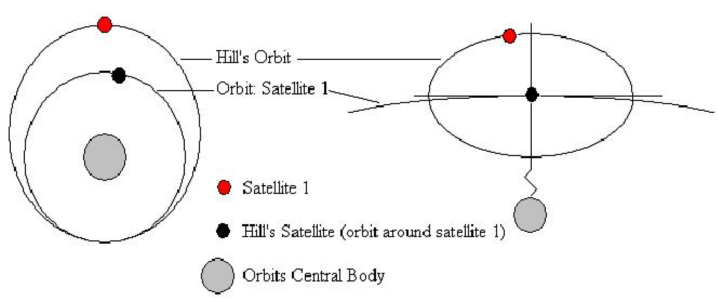


Figure 2: The Hill's Orbit

In particular, assuming the tug spacecraft is within a short distance of the target satellite, their relative dynamics can be described using Hill's frames. This means that with some approximation it can be assumed that the target is in circular orbit around the tug (or vice versa).

Figure 2 illustrates the two spacecraft relative positions. The left drawing in the figure depicts a service vehicle and its target satellite orbiting the Earth assuming that their orbits are in the same plane but with different eccentricities. The right drawing looks at the same three-body problem but from the point of view of the service vehicle (in black). In its reference frame, the target appears to be in elliptic (circular) orbit around it while performing its motion around the Earth.

This shows that the Hill's relative frame allows the reduction of the 3D model to two dimensions. To adapt the experiment to this model, the target has to be designed such that it will move in a circle at the edge of a circular search space, thus making the strategies implemented in 2D valid for the space problem.

4 Experimental Approach

4.1 Experimental Overview

The experiment's main objective is to simulate the *Space Tug*'s rendezvous with its target in a simplified two-dimensional environment. The space in which the real *Space Tug* contains several degrees of freedom and is too complicated to reproduce in two dimensions. As a result, important modeling assumptions were made. The simulation makes use of the relative positions of the *Space Tug* and the target. The satellites are in the same orbital plane relative to Earth and their orbits have the same eccentricity. Therefore, the target is fixed at a point in space, relative to the search space's reference frame.

The experiment makes use of floor space for the search area, whose dimensions represent the appropriate ratio of search area to tug/target sizes. This ratio was calculated using the real sizes of these vehicles and the space around the target created by position uncertainties. The experimental set up is shown in Figure 3.

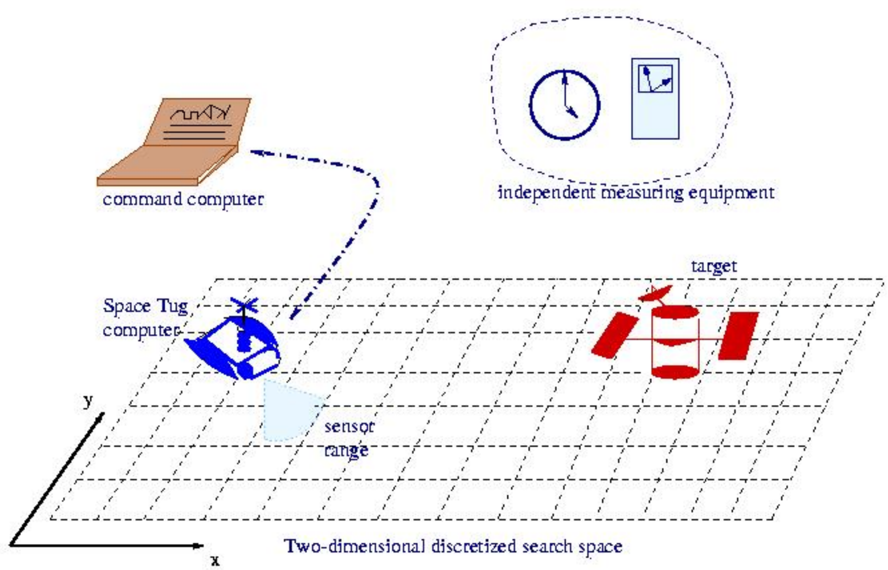


Figure 3: Test-bed environment for Tug/Target rendezvous simulation

As shown above, the *Space Tug* computer has to search through the space for the target, using its sensors. It is understood that the sensors ranges are much smaller than the size of the search space. Furthermore, in the case of the human-in-the-loop search strategy, a computer is used to transmit commands to the robot using a serial communication system, in accordance with the sensor data that the *Space Tug* computer sends to the human. For the other two search strategies, the computer is used only to download the control system that moves the robot and the decision making software that tells it where to go. The *Space Tug* is made with LEGO Mindstorms, using an on-board computer, and the target is a 10×20-centimeter box. While the target is non-cooperative and inert, the *Space Tug* carries, as mentioned previously, a collection of on-board sensors, including long and short-range infrared distance sensors and touch sensors. The first two collect data about the position of the target, while the last one stops the vehicle from going out of the search space.

The independent measuring equipment shown in Figure 3 is used to record the time it takes for the *Space Tug* to find its target and the energy consumed during the process. Using this data, the cost function between time and energy can be evaluated, and the effectiveness of each strategy can be compared in order to assess the hypothesis of the experiment.

4.2 Design of search space

The size of the search space is an important aspect of the experiment setup. It has to match the relative sizes of the satellites in space. Furthermore, it is necessary to model the space correctly, so that the results from this experiment can be validated for the space environment. To calculate the size of the test search space, some information such as global position system (GPS) accuracy, satellite sizes and sensor ranges has to be collected. In the US Army Corps of Engineers manual [6], GPS accuracy is reported as approximately 100 meters. Once the target satellite has been located, the space is which the *Space Tug* has to search is thus a sphere of radius 100 meters, centered at the expected location of the target. a database of satellite sizes was searched and compared to the 100-meter search radius [7]. It turns out that the average satellite geometry is 2x2x2 meters which gives a 50:1 length scale with respect to the search space radius. The space transformation process is shown in Figure 4.

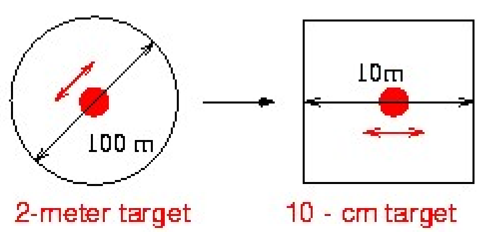


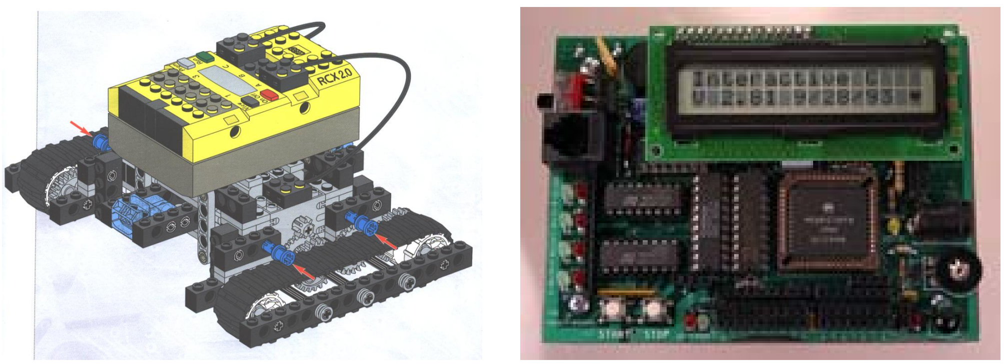
Figure 4: Search space transformation

As can be seen in Figure 4, the actual spherical space that the real orbital servicer will have to search is three-dimensional. However, since the target is not stationary, four variables are needed to define its position, a length, two angles and time. The space transformation involves going from four dimensions to only two. As a result, the main modeling assumption that has to be made is that the *Space Tug* is capable of insertion in the same orbit as the target. The space is then modeled as a two-dimensional problem such that the tug and the target are in the same orbital plane with respect to Earth. From Figure 4, it can be seen that, for a target of size 0.1 meters, the size ratio is maintained if the search space is of radius 5 meters. Since the space is designed to be a square, the sides of the search space will be 10 meters.

4.3 Overview of Hardware

4.3.1 *Space Tug* Robot

The robot simulating the orbital servicer is made of LEGO Mindstorms parts. The on-board computer is a hand-held, battery-powered microcontroller board, called the Handy Board and developed by MIT. Shown in Figure 5(b), the Handy Board is based on the Motorola 68HC11 microprocessor and includes 32K of battery-backed static Random Access Memory (RAM), outputs for four DC motors, inputs for a wide range of sensors and a 16x2 character Liquid Crystal Display (LCD) screen [8].



(a) **(b)**

Figure 5: *Space Tug* robot (a) and on-board computer (b)

Originally, the experiment was designed to use the LEGO RCX 2.0 on-board computer, shown in Figure 5(a). However, due to memory constraints on this computer, the on-board controller had to be changed to the Handy Board, in order to be able to load the search map necessary for the random and autonomous searches. The functionalities of both computers are very similar, except in terms of memory and communication. As mentioned earlier, the Handy Board has more memory than the RCX and thus is able to handle the search algorithms. While the RCX communicates with the command computer through an infrared interface, the Handy Board uses a standard serial port. Hence, the communication problem in the semi-autonomous search tests was solved by tethering the robot to the command computer, which also cut down on communication lags.

The Handy Board runs Interactive C. The latter is a cross-platform, multi-tasking version of the C programming language, which is perfectly adapted to make full use of the controller's resources. A more detailed description of the software can be found in the *Overview of Software* section.

4.3.2 On-Board Sensors

The *Space Tug* simulator carries an array of sensors in order to carry out its search for the target. Added to the standard touch sensors, the tug possesses two infrared distance sensors with overlapping ranges, as can be seen in Table 1.

Table 1: Operating range for *Space Tug* sensors

Sensor	Upper bound [meters]	Lower bound [meters]	Resolution [meters]
Long Range Infrared Distance	0.20	1.50	0.01
Short Range Infrared Distance	0.04	0.30	0.005

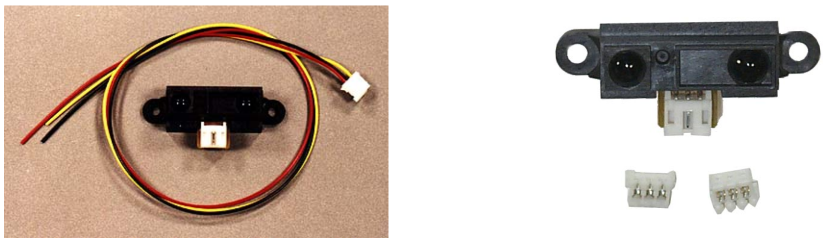


Figure 6: Sharp Infrared Distance sensor

The two infrared distance sensors are built by Sharp and are shown in Figure 6. Even though they are essentially the same sensors, their overlapping ranges provide an appropriate field of view, so that the target can be detected. The long-range sensor is used in the semi-autonomous and autonomous searches, but not in the random search. On the other hand, the short-range sensor is used in all three searches to detect whether the target is found or not. Once a value high enough - corresponding to approximately 10 centimeters - is returned by the short-range sensor, the search is stopped and the target is said to be found. The Sharp sensors are available off the shelf and fairly reliable. They use an infrared beam, coupled with an optical triangle measuring method which reduces the influence of the reading on the colors of the reflected objects and their reflectivity [9].

In order to use these sensors with the Handy Board, the controller had to be physically modified. The Sharp sensors do not work with the Handy Board unless the pull-up resistors connected to the analog sensor ports are taken out of the loop. Hence, the Handy Board needs to be rewired. The leads to the pull-up resistors were cut on analog ports 5, 6 and 7, and a wire was soldered to close the loop behind these three ports. The latter modification allows for the use of the analog ports 8 through 16. A disadvantage caused by these hardware modifications is that it makes the readings from the sensors less accurate. Furthermore, if the leads to the pull-up resistors are not cut completely, current spikes in the analog port can cause flawed sensor readings that can be misinterpreted by the robot's control system software.

The third type of sensor used on the tug robot is the touch sensor. There are two touch sensors used on a dynamic bumper built on the vehicle. These sensors detect any pressure applied on the bumper arms. When no pressure is applied the sensor reading is the passive reading on the analog port (255). Once the robot bumps into an object, the current through the analog port changes and and the sensor reading decreases from the rest value of 255.

4.4 Overview of Software

The software for this project is developed in two steps. First, the random and autonomous strategies are coded as *Matlab* simulations. The semi-autonomous search cannot be simulated due to the involvement of a human operator. The purpose of this simulation at this time is to validate the soundness and logic of the algorithms. Second, the algorithms are converted to the C language to be uploaded on the Handy Board computer. The compiler used is Interactive C, developed by Newton Labs, and is specially adapted to load programs on the Handy Board. Interactive C compiles the C code, customized with special sensor and motor functions to use the computer's resources, and loads it onto the board. The Handy Board contains firmware, called the PCode, which then serves as an interpreter for the compiled C code.

The three search strategies are random sensor-less search, semi-autonomous with a human decision maker search and fully autonomous with sensors search. The algorithms were designed to be general enough, so that they span the space of all different strategies that could be used to find the target.

4.4.1 Random Search Strategy

The random search algorithm is inspired by the "Greedy Algorithm" described by Gelenbe [2] in his paper on autonomous search for information. It is a probabilistic search where the agent - the tug in the experiment - is able to learn as it moves in the space. Each displacement in the space provides information to the robot. In other words, when the robot moves to a point and does not find the target, it then knows that the target is not located at that point. Its knowledge about the search space has increased. The search space is transformed into a grid that contains a certain number of locations, as shown in Figure 7.

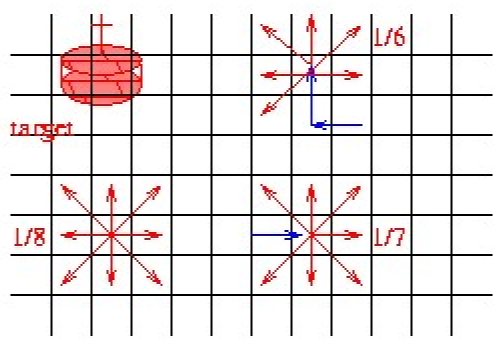


Figure 7: Grid for random search

The distance between each point has to be dependent upon the size of the *Space Tug* and the size of the target. An appropriate separation between two points given that the tug and target sizes are approximately 20 centimeters would be of the order of two times the size of the objects, or 40 centimeters. The *Space Tug* at its starting location has eight possibilities for its next move. As can be seen in Figure 7, the probability of going to any of the eight next locations is 1/8. Once the tug has moved, the probability associated with the location that the vehicle just left is set to zero. As a result, the tug has now only seven possibilities for its next move. The *Space Tug* computer thus learns about the space as it moves from point to point. The search ends when the target is found, which is detected by the short-range infrared distance sensor - due to its relatively short range, this infrared sensor will only detect the target once the *Space Tug* is close enough to it.

4.4.2 Semi-autonomous Search Strategy

The basic concept for the semi-autonomous search is that the decision-maker is a human controller. Using the on-board sensors, the human operator moves the *Space Tug* to find the target. Figure 8 shows a simplified flowchart for the procedure to be followed during the semi-autonomous search with human-in-the-loop. At any time, the tug can perform a 360-degrees sweep of the surroundings using the long-range infrared distance sensor. If the sensor does not report any presence of an object, then the human operator has to make a decision about where to move next. The operator sends a command to the tug on-board computer, which then moves the vehicle to the next desired location. All sensor data at each step is sent back to the human operator, in order to decide the next move in the search.

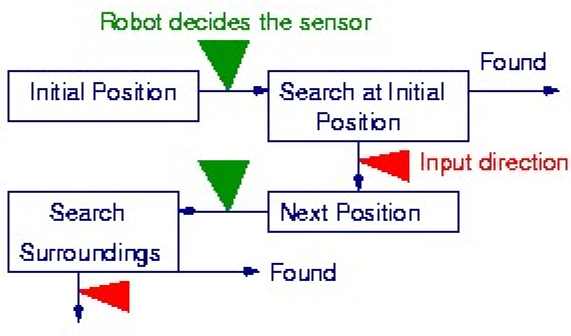


Figure 8: Flow chart for semi-autonomous search

Another decision is made based on the new data, and so on until the target is found. In order to transmit information, the tug is tethered and waits for the new command through its serial port. As a result, there is a lag between the command transmission and the tug's move. Although the transmission could be time-consuming, it is a good simulation of what happens with space transmission. For instance, as Morrison and Nguyen [1] describe, the Mars Pathfinder also uses waypoint navigation and delayed transmission to communicate with the Earth operator.

An important consideration for this strategy is human bias. A human operator has to have no prior knowledge of the initial conditions on the search space, so that decisions will not be influenced by that knowledge. Therefore, the operator cannot see the experiment, but will only read sensor data on the computer. Another part of human bias is the employment of a consistent strategy by a single person. A variety of people should be invited to conduct the semi-autonomous search in order to ensure unbiased data. These logistics require prior organization, communication with people external to the project and maybe an additional expense.

4.4.3 Autonomous Search Strategy

The fully autonomous search makes use of the long-range ultrasonic sensor to find the target in the test space. The autonomous strategy is based on a probabilistic model, in which the algorithm develops a probability density function to describe the search area. Since the *Space Tug* initially has no information about the location of the target, the probability density has to be uniform across the space.

As can be seen in Figure 9, the symmetry and the uniformity of the distribution places the center of

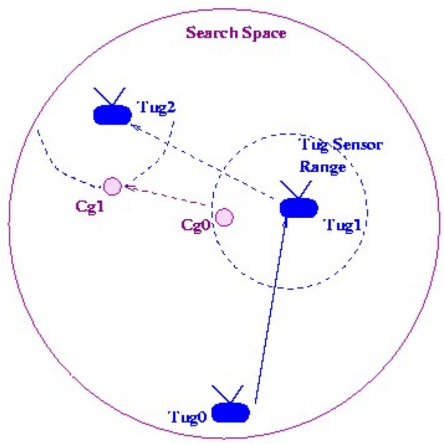


Figure 9: Autonomous search strategy

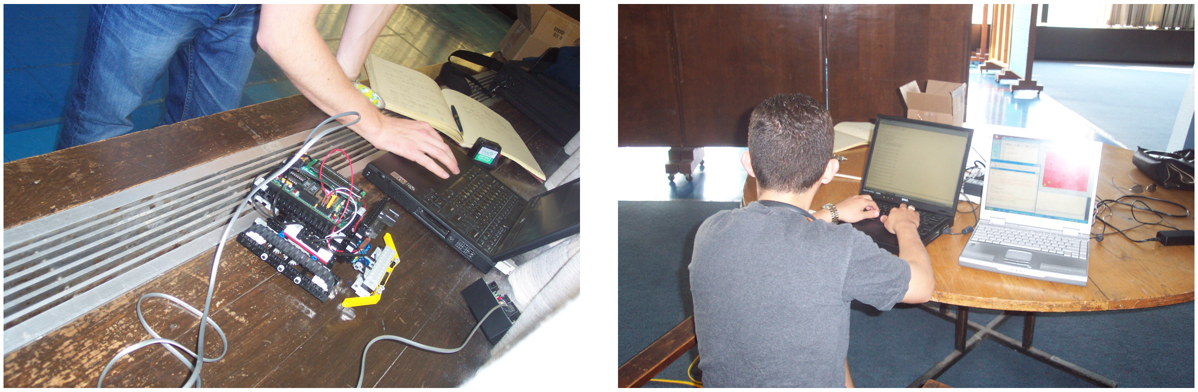
mass - labeled "Cg0" - in the middle of the two-dimensional search area. At the start of the search sequence, the tug travels toward the center of mass of the probability density distribution to its first waypoint. This location has to be a point in the search space close enough to the center of mass so that the latter is in range of the tug's long-range infrared distance sensor. Once at its new location, the vehicle performs a 360-degree sweep of the surroundings in an effort to locate the target. During this process, the tug learns about the search space. If the target is not found, the density of the swept area is set to zero. The probability density is then redistributed uniformly across the remaining space and the new center of mass is located - labeled Cg1 in Figure 9. The process just described is then repeated. Once the target is found in range of the long-range infrared distance sensor, the *Space Tug* vehicle moves straight towards it for rendezvous.

4.5 Testing Method and Error Calibration

4.5.1 Testing Procedure

All of the experimental tests were run in a student residence on the MIT campus. A large dance floor in this residence was determined to be the most appropriate surface for the robot. It allowed for reduced friction and easy setup of the search space. The search space was constructed using a standard garden hose to surround it. This allowed for the robot to recognize the limits with the touch sensors on the bumper without seeing the hose with the infrared sensors, thus avoiding any confusion between the limits of the search space and the target itself. Because the testing area is used by other students and groups, the search space had to be dismantled after each testing period.

Before each test, the appropriate control system was loaded onto the Handy Board, as shown in Figure 10(a). Three of these were available, corresponding to the random, semi-autonomous and autonomous search algorithms. The search was then run with the appropriate target relative position. Once the target was found, the *Space Tug* on-board computer would display the time elapsed during the search and the number of steps taken to find the target. These measurements were recorded for each run. Figure 10(b) shows a human operator during one of the semi-autonomous search tests. As can be seen, the operator could not see the search space, which is crucial to this strategy, in order to eliminate bias errors and make the



(a) (b)

Figure 10: Loading the code onto the *Space Tug* robot (a) and human operator during semi-autonomous search (b)

search realistic.

4.5.2 Measurement Systems

The goal of the experiment is to evaluate a cost function that relates time and energy consumption during the search strategies. As such, the relevant quantities that need to be measured are the time elapsed during the search and the energy consumed from the tug's batteries. The time data is taken using the Handy Board computer's internal clock. The procedure for measuring time is directly embedded in the software, in an effort to be as precise as possible.

As for the energy measurement, a minor change was made from the procedure introduced in the design phase. Looking at different battery energy measurement devices, it was realized that the level of precision they provide is unnecessary for the purpose of this experiment. For comparison, the number of steps taken is a simpler and easier to simulate measure. It was estimated that the difference in energy expenditure among strategies is negligible as it is solely due the use of sensors. The power needed by the sensors was estimated as insignificant compared to the power taken by the motors. As a result, the movements of the robot in the search space were modeled as based on a standard-sized step, which was constant across all searches.

4.5.3 Sources of Error

Sources of error are associated either with measurements taken or with logical error in the coding of the search strategies. Software or logic errors in the implementation of the search strategy are systematic errors that would be hard to detect. However, a thorough and detailed debugging and testing stage for each software component eliminated these errors. Furthermore, cross-checking of the code between the experimenters reduced the chances of implementing a logical error in the search strategy. Efforts to eliminate these systematic errors are particularly important for the implementation of the random and the autonomous search strategies.

Errors associated with energy and time measurements are easier to ascertain. The energy measurement is done using the number of steps taken by the robot. Hence, it is calculated using the tug's control system and the measuring procedure is embedded in the software. As a result, the number of steps returned is exact.

On the other hand, there exists some error to the way the number of steps relates to the actual amount of energy depleted during the search. However, the size of the error is small and thus negligible for the purpose of this project, as stated in section 4.5.2. Time measurement is also accurate, since it uses the *Space Tug* computer's internal clock. The latter device measures time in seconds and is precise to approximately one millisecond.

An important source of error arises from the semi-autonomous search strategy. The human controller can be subject to decision-making bias in choosing the Tug's next waypoint during the search. It is of importance that the human operator has no knowledge of either the location of the target or the type of search being run. Such information about the situation introduces a bias in the human's interpretation of the data and decision-making process. In order to eliminate this possible error, it is necessary to use an outside person to control the tug. The authors have extensive knowledge of the search strategies and the situation and therefore cannot be bias-free human controllers. To reduce this effect, searches were run with five different human operators. The results of the experiment need to be independent from the human operator. It is therefore important to eliminate the human factors effect from the tests that are run for the semi-autonomous search. For each test run, a different decision maker was used. The best human operator has minimal knowledge of the search except the rules.

4.6 Test Matrix

4.6.1 Variables and trials

The independent and dependant variables chosen in the design phase were kept the same for the experiment. The three types of strategies (random, autonomous and semi-autonomous) were tested against three relative target positions as described in the *Background and Motivation* section. Five trials were run for each of the 9 (3×3) tests. Table 2 shows the relative target positions tested versus strategies. *Target at x%* means that the target is placed at a distance roughly x% of the search space diameter. For example, 100% means that upon start the target is located at the opposite corner of a rectangular grid space.

4.6.2 Modifications from design phase

Two major modifications to the experiment design were made due to space and time constraints. First, the originally planned number of 10 trials was reduced to 5 because testing was delayed due to hardware difficulties. Nevertheless, even from 5 trials, the data gathered is conclusive. A greater number of runs can be included in a future more detailed experiment where it is expected that these results will be confirmed.

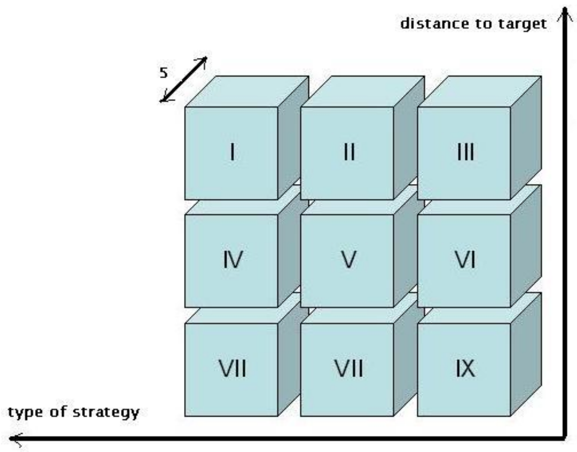
The second major modification was reduction in grid size. Instead of the planned 50×50 grid, the search space was 25×25. One reason for this change that the 50×50 grid caused extremely time-consuming trials. Thus, the space available, the time limits and the reliability of the hardware induced the reduction of the grid size. Comparison between simulations with both grid sizes and the experimental results suggest that the qualitative results and conclusions drawn are valid for the larger scale model.

5 Data Analysis

5.1 Raw Data Analysis

The experimental data presented in the next section was collected and filtered on site. As a result, all the data points are valid measurements. During testing, some runs were not recorded due to an error during

Table 2: Test matrix: Target position vs. strategy and number of trials



Pos-Strategy	Random	Autonomous	Semi-autonomous
Target at 100%	I	II	III
Target at 50%	IV	V	VI
Target at 10%	VII	VIII	IX

the search. For instance, due to the relatively low reliability of the hardware - both the Handy Board and the sensors - inaccurate sensor readings would, on a few occasions, stop the robot even though the target was not in sight. These bad runs were not recorded and the test trial was repeated. As a result, the data points shown and analyzed in the next sections are only the runs during which the robot found the target successfully and without any sensor errors.

5.1.1 Experimental data tables

Experimental data is shown below. Table 3 shows the time and energy for the five trials of the random search runs and Table 4 shows the data points for the semi-autonomous search. In the latter, each trial represents a different human operator. Finally, the results of the deterministic run for the autonomous search is shown in Table 5.

Table 3: Random search data

	10% Relative Distance		50% Relative Distance		100% Relative Distance	
Trial	Time [seconds]	Energy [steps]	Time [seconds]	Energy [steps]	Time [seconds]	Energy [steps]
1	429.36	134	101.56	35	485.56	156
2	12.06	3	47.19	13	782.17	267
3	58.11	19	275.10	96	143.37	52
4	25.98	7	229.31	75	373.70	119
5	558.08	165	981.20	315	308.31	97

Table 4: Semi-autonomous search data

	10% Relative Distance		50% Relative Distance		100% Relative Distance	
Trial	Time [seconds]	Energy [steps]	Time [seconds]	Energy [steps]	Time [seconds]	Energy [steps]
1	352.36	81	245.84	30	775.52	158
2	70.55	23	211.54	28	1066.1	168
3	49.86	7	53.88	16	776.79	139
4	245.65	75	150.25	46	256.17	43
5	51.13	7	366.01	45	323.99	48

Table 5: Autonomous search data

	10% Relative Distance		50% Relative Distance		100% Relative Distance	
Trial	Time [seconds]	Energy [steps]	Time [seconds]	Energy [steps]	Time [seconds]	Energy [steps]
1	254.90	67	33.05	9	354.35	93

5.1.2 Discussion of results and data validation

It is difficult to see any trends in the data in table form. The data was processed and graphed in order to observe what the relationship between time and energy was for each search algorithm. Figure 11 shows all the experimental data points in the time-energy space. It is already possible to observe some linear trends, especially with the measurements from the random search runs.

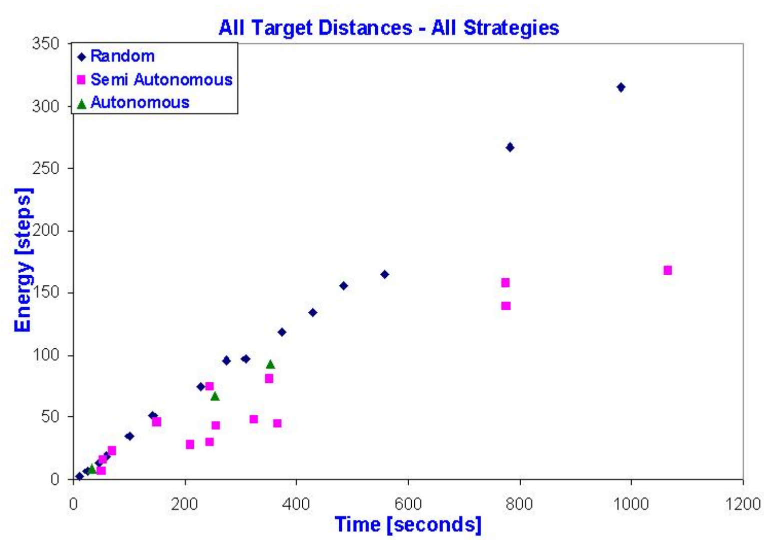


Figure 11: All searches: all test runs

An important aspect of the results is to look at the distance factor in the data. Figures 12, 13(a) and (b) show the results for all strategies for the 100%, 50% and 10% relative distance test cases, respectively. In Figure 12, the general trend of the trade-off between time and energy for each strategy can be seen well.

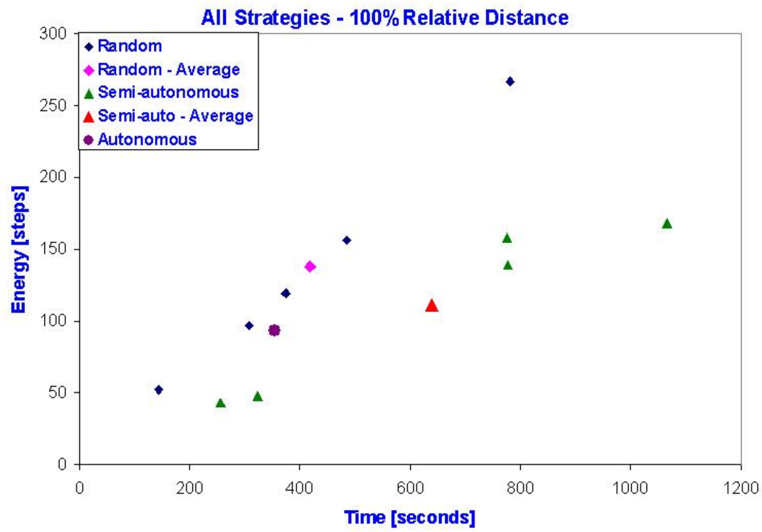
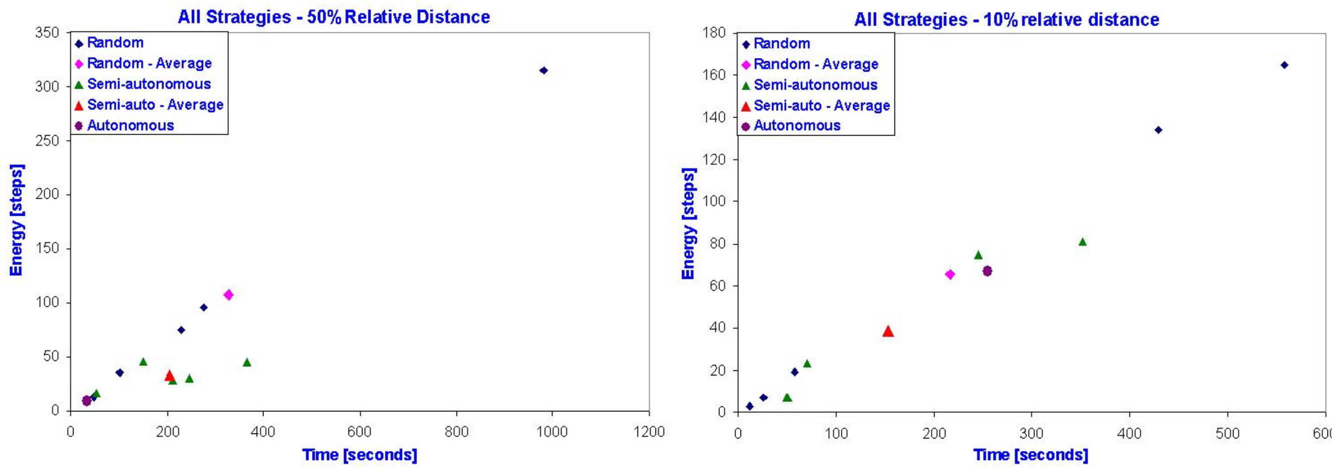


Figure 12: All searches: target at 100%

On average, the random search algorithm is the strategy that expands the most energy, while the semi-autonomous search algorithm is the slowest strategy. It can also be observed that the autonomous search is more efficient in terms of both time and energy than the other two strategies. The same phenomenon can be observed in Figure 13(a), where the autonomous search algorithm is the most efficient of all three strategies for the 50% relative distance test case. The fact that the efficiency difference between the autonomous and semi-autonomous searches for this test case is so flagrant can be explained by the way the autonomous search works. The autonomous version of the *Space Tug* robot uses a probabilistic search to find its target. Since the search space is initially overlayed with a uniform probability distribution, the center of mass of the distribution is right where the target has to be in the 50% relative distance case. The autonomous search always goes to the center of mass, and hence finds the target extremely quickly in this test case.

The experimental results for the 10% relative distance test case show a slightly different trend. Unlike the other two cases, as can be seen in Figure 13(b), the semi-autonomous search is the most efficient strategy, with regards to both time and energy. The autonomous search in this case does not perform as well. Again, as mentioned earlier, the reason for this phenomenon is the way the autonomous search was coded. The robot goes very close to the center of the search space as its first move. Hence, after that first move, the *Space Tug* finds itself further away from the target than it originally was.

Overall, it can be seen that the random search is not efficient at all compared to the other two strategies. Moreover, the autonomous search seems to be most efficient in terms of time and as efficient as the semi-autonomous strategy in terms of energy needed. Looking at only experimental data, the hypothesis is disproved, as the semi-autonomous search algorithm is not the most efficient strategy. The following section is the analysis of the simulation data collected from the *MatLab* simulation which allows the validation of the experimental data.



(a) **(b)**

Figure 13: All searches: target at 50% (a) and target at 10% (b)

5.2 Simulation Analysis

The simulation code for this experiment was developed with a twofold purpose. First, it was used as a way to create the logic for each algorithm and test case. This logic form was written in *MatLab* (because of the easy graphical interface) and then translated to Interactive C to make it readable by the Handy Board computer. Second, the *MatLab* code was continuously used to validate the experimental results by comparing the theoretical expected data with the actual numbers. Finally, key trends in the simulation were used to help draw important conclusions from the actual data.

From the three strategies only the random and the autonomous were simulated due to the complexity of modeling human behavior which makes the semi-autonomous hard to model. For comparison, the first two strategies were tested in simulation under the experimental conditions: a 25×25 grid with the three relative target positions for each algorithm. For example, Figure 14 shows the random data points plot versus the experimental data points.

A hundred simulation points were plotted to demonstrate the expected trend and to remove a possible bias due to the randomness. The experimental data does not match exactly the model, but it is scattered around the simulation trend. A better match can be seen at 50% and 10% target distance plotted on Figure 15(a) and (b). For small number of steps and short times, the experimental points correlate with the simulation trend. In general, the experimental points always fall below the simulation scatter. This means that for a given number of steps, the experiment took longer than expected. The reason for that is that the simulation probably does not model well enough all the time delays caused by the hardware.

The autonomous results demonstrate a better match between theory and practice because the autonomous algorithm is deterministic. This is illustrated in Figure 16 where the simulation data points match the experimental very well, apart from one data point. At 10% target distance the experiment took longer and larger number of steps than the model because the robot *missed* the target on its way to the center (center of probability density).

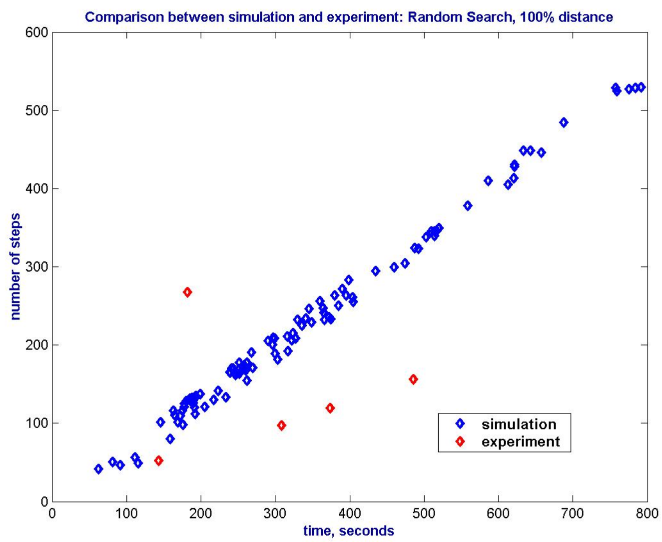
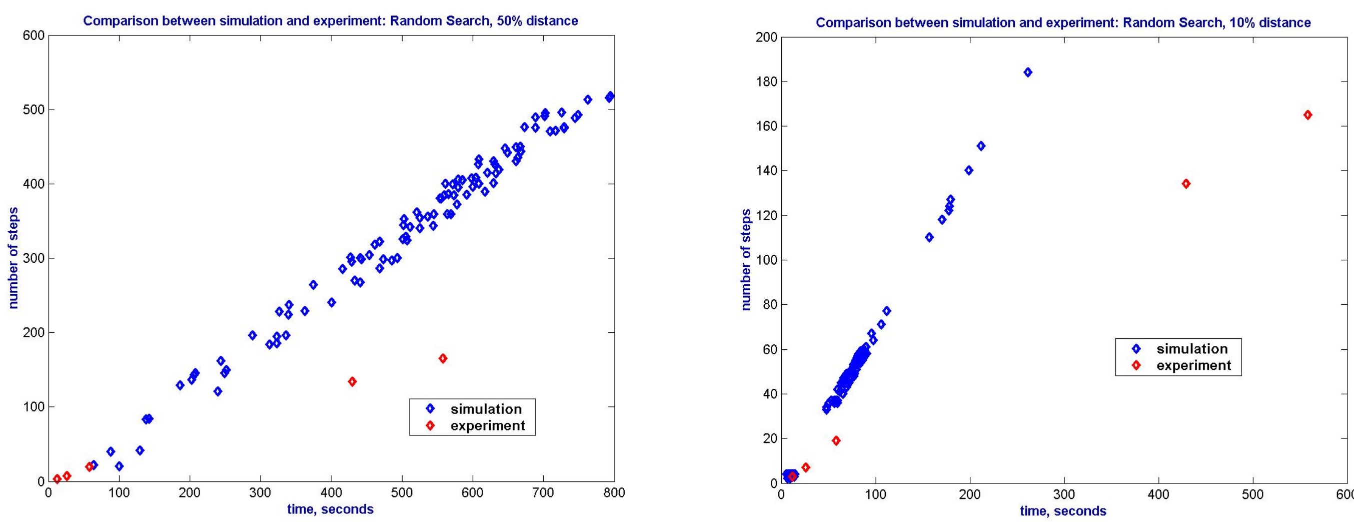


Figure 14: Random search: target at 100%



(a) **(b)**

Figure 15: Random search: target at 50% (a) and target at 10% (b)

To emphasize, the simulation results were not merely used for comparison and validation but also to identify key trends in the data that would help the assessment of the hypothesis. One possible generalization of the experimental design is to randomize the target location. Comparing 100 trials with a randomly

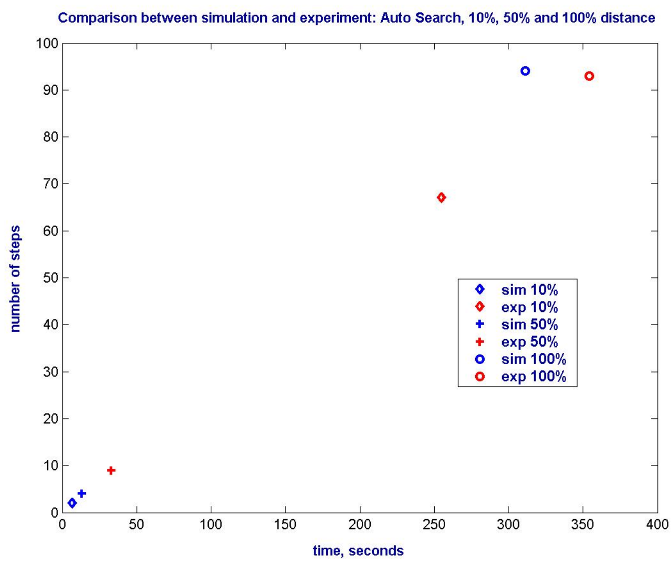


Figure 16: Autonomous search: simulation versus experiment

generated target location across the board produced the results in Figure 17. According to this simulation, the autonomous algorithm is less time and energy-consuming for all data points.

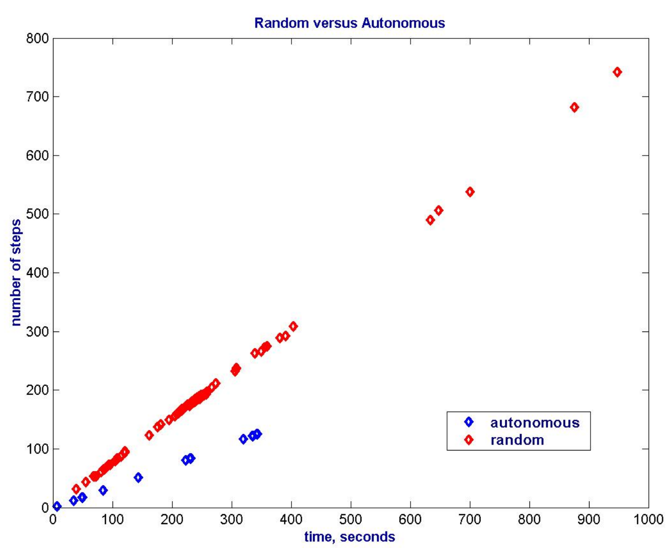


Figure 17: Simulation trends: random vs. autonomous

The two questions arising from this result are whether the experimental data matches this behavior and also, where do the semi-autonomous data points fall in this pattern. A complete discussion and hypothesis assessment is done in the next section.

5.3 Comparison of Simulation and Experimental Data

As suggested by the simulation analysis, the experimental data confirmed the overall better performance of the autonomous algorithm. This makes sense even without the results because of the design of each strategy. The autonomous strategy uses a larger sensor range, so it covers the search space much faster. The autonomous robot also goes to more *likely* areas of the grid, as opposed to the random which can get stuck in low-probability space and waste more energy.

The key result concerns the semi-autonomous strategy. The experimental data on Figure 18 not only confirms the simulations trends (Figure 17). Moreover, the semi-autonomous scatter appears below the trend-lines of the other two algorithms. The human operator strategy turns out to be the most energy efficient, but also the most time-consuming. One lesson from the trials is that human operators make good decisions but take too long to decide. Since fuel and propulsion design is often a larger constraint than time for space applications, the semi-autonomous search seems like a good strategy. On the other hand, time can be more important on a smaller scale like for eclipses in LEO and phasing with the target satellite. The communication delay also makes the autonomous algorithm look better. Finally, these arguments suggest that a hybrid approach might the most efficient.

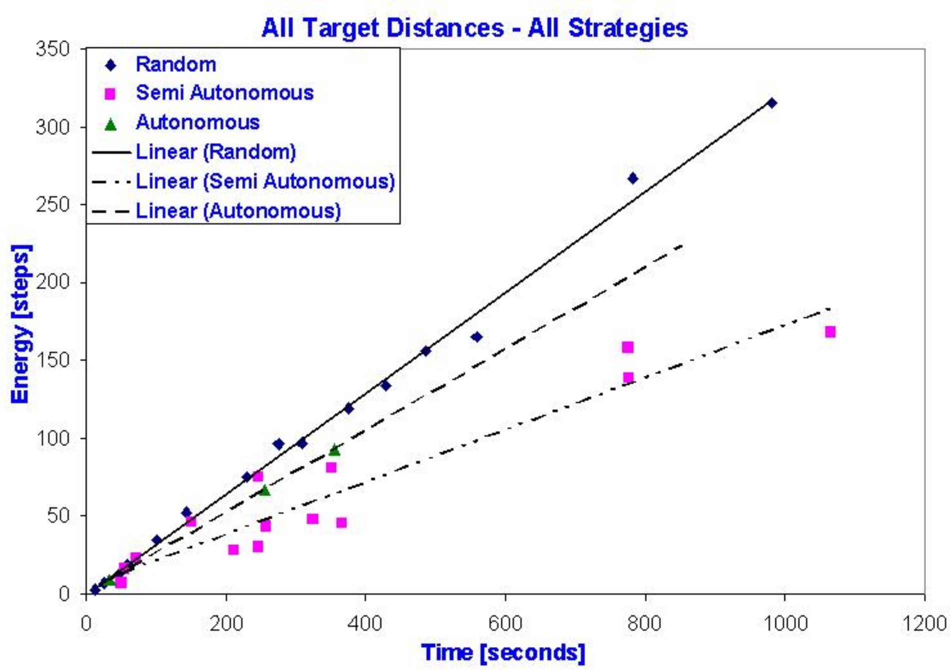


Figure 18: All data with trendlines

A combined autonomous with human operator approach will depend on the background, motivation and knowledge of the mission. For example, a smaller search space or more information about the target coordinates would favor an autonomous approach. On the other hand, a larger search with more uncertainty might be accomplished better with a hybrid approach.

In conclusion, the hypothesis is disproved since it was established that the autonomous search performs best in terms of both energy and time. A cost function with equal weight of time and energy is a linear $y = x$ function which divides the *Energy vs. Time* plane in two. The algorithm whose trend-line approaches best this line (falls in the middle of all strategies) is the best suitable for an equally weighted cost function. With that assumption in mind, the autonomous strategy was established as the most robust performer. Clearly, a different cost function caused by different customer or mission requirements might incite a different conclusion.

5.4 Experiment Validation and Future Work

The primary goal for designing this two-dimensional target search experiment was to solve a subset of the general *Space Tug* problem. The results from this project support one of the key themes in the servicing vehicle concept - autonomy. Depending on the mission, the tug can have different degrees of autonomy which corresponds to the hybrid search concept. A higher-fidelity experiment can be designed with a greater level of detail, mission and customer requirements to assess the same hypothesis for a larger design trade space.

In summary, this 16.62x project successfully models an important aspect of the general *Space Tug* problem by assessing uncertainty and autonomy with a simple scheme. Future work in this area might involve modifications in both the model and the experiment design. There are a number of possible ways to improve the experiment as designed. For example, obtaining more data might give more insight into important trends and possibly point towards better versions of the strategies used. Also, randomizing the target location (as done in simulation) will remove some of the bias in the algorithms relative performance. Furthermore, for higher precision of the data, metrology on the robot can be implemented to close the control loop and thus approach better the situation in space.

The experiment design can also be modified in a variety of ways. For instance, the Hill's frame scheme can be implemented by designing a target which moves on the edge of a circular search space. Thus the robot will have to find, track and phase with the target satellite, which is much closer to real scenario in orbit. Another potential arising from the semi-autonomous data is to develop a separate human factors experiment which would model decision-making and human behavior in comparison with automated logic. Together with all the above, a higher-fidelity simulation will be needed to precede the spacecraft software and testing programs development for the real *Space Tug*. This would involve not only modeling the orbital dynamics, but also all hardware effects.

6 Summary and Conclusion

In view of the results presented above, we conclude that this 16.62x experiment was successful. The goal of implementing the designed experiment to assess the hypothesis was achieved. The theory based in simulation was confirmed by the tests. As expected, the autonomous strategy outperforms the random, which is the most time and energy inefficient overall. Moreover, it was found that the semi-autonomous algorithm is the most energetically efficient approach, but the most time-consuming. This finding disproved our hypothesis which stated that the semi-autonomous strategy is the most efficient in terms for both time and energy. Instead, we conclude that an autonomous algorithm is more suitable for space applications. The results also suggest that, depending on knowledge of the search space and the mission requirements, a hybrid approach might be more efficient.

The successful hypothesis assessment together with our conclusions about autonomy make this experiment an important asset for the general *Space Tug* project. The analysis of the results demonstrates a lot of potential for a new phase of modeling and experimentation.

Acknowledgements

First and foremost, the authors would like to thank Professor Olivier de Weck, project advisor, for his support and insights on the project. The multiple suggestions and advices that he has given to this team have helped guiding it toward an interesting experiment with great potential.

Secondly, the authors thank the human operators - Victoria Davis, Carlos Pinedo, Devjit Chakravarti, Danny Craig and Jennifer Shih - who took time out of their busy schedules to help us collect the necessary data. Furthermore, the experiment would not have been possible without a few friends and classmates who were there to support and help this team through its worst times. Their assistance is more than greatly appreciated.

Finally, this project would not have seen the light without the support of the 16.62x staff and faculty. The many suggestions made by the technical staff during the oral presentation and team meetings have provided the necessary information to make this project report complete. Moreover, the feedback from the faculty, Professors Edward Greitzer, John Deyst, Earll Murman and Jennifer Pixley, has been of great importance in the shaping of this experiment. The authors would also like to acknowledge Danny Craig and Greg Mark for their comments and support throughout the project.

References

References

[1] Morrison, J., Nguyen, T. *On-board Software for the Mars Pathfinder Microrover* Proceedings of the Second IAA International Conference on Low-Cost Planetary Missions, John Hopkins University Applied Physics Laboratory, Laurel, Maryland, April 1996
http://mars.jpl.nasa.gov/MPF/roverctrlnav/publications.html, NASA public site, Date accessed: 10/5/2002.

[2] Gelenbe, E. *Autonomous Search for Information in an Unknown Environment*, M. Klusch, O.M. Shehory, and G. Weiss (Eds): CIA'99, LNAI 1652, pp. $47 - 60, 1999$
http://link.springer.de/link/service/series/0558/bibs/1652/16520047.htm, *Lecture Notes in Computer Science* Journal site, Date accessed: 10/4/2002.

[3] Hillenbrand, U., Hirzinger, G., *Probabilistic Search for Object Segmentation and Recognition*, Proceedings European Conference on Computer Vision 2002, *Lecture Notes in Computer Science* vol. 2352, pp. 791-806, 2002. http://cogprints.ecs.soton.ac.uk/archive/00002393/, Cogprints site, Date accessed: 10/6/2002.

[4] Bernard, D., Golombek, M. *Crater and Rock Hazard Modeling for Mars Landing*, AIAA Space 2001 Conference and Exposition, Albuquerque, NM Aug. $28 - 30, 2001$ $AIAA - 2001 - 4697$

[5] Diop, Julie Claire, *Robotic Vision*, Premium Content Upstream Article, October 2002
https://www.technologyreview.com/articles/upstream1002.asp, Technology Review site, Date Accessed: 10/5/2002

[6] "Chapter 5 - GPS Absolute Positioning Determination Concepts, Errors, and Accuracies" Engineering and Design - NAVSTAR Global Positioning System Surveying, US Army Corps of Engineers Engineering Manuals, August 1, 1996, EM 1110-1-1003. http://www.usace.army.mil/usace-docs/eng-manuals/em1110-1-1003/, US Army Corps of Engineers site, Date accessed: 09/30/2002.

[7] Galabova,K., "Satellite Database", Space Tug Project Progress Presentation, summer 2002, source: \\aero-astro\PublicShare\Kalina's Database\SatDatabase.ppt, MIT AeroAstro public domain

[8] The Handyboard Page. http://www.handyboard.com, Date accessed: 03/02/2003.

[9] Sharp SMA Optoelectronics Distance Measuring sensors.
http://www.sharpsma.com/sma/products/opto/OSD/distance_measuring_sensors.htm, Date accessed: 03/06/2003.

A Parts List

Table 6: Detailed Parts List

Part	Manufacturer	Reference Number
Mindstorms Robotics Invention System 2.0	Lego	3804
Handy Board micro controller	MIT	N/A
Long-Range Infrared Distance Sensor	Sharp	GP2Y0A02YK
Short-Range Infrared Distance Sensor	Sharp	GP2D120
Touch Sensor Multiplexor	HiTechnic	MX1075
Garden Hose	Home Depot	N/A