Submitted on May 4, 2018 for EEC 793: Autonomous Intelligent Robotics
Volume 1, Number 1,

Rabbit: A Robot for Child-Robot Interaction

Humberto De las Casas and Holly Warner

Abstract Human-robot interaction is a broad field with many disciplines. Among them is child-robot interaction. This project aims to create a robotic system for entertaining children by playing games, specifically tag and hide and seek. The solution will include a dialogue agent to make the robot friendly and interactive as well as accessible to young children. An escape algorithm is created for the tag game. A multi-armed bandit learning algorithm is developed for selecting hiding locations. The complete system is evaluated in simulation in Gazebo.

1 Introduction

Robots are an item of curiosity to the general public in today's world, and no one expresses curiosity better than children. Human-robot interaction (HRI) has long been studied, including applications best suited to children. Nowadays there is an important branch of research of HRI called child-robot interaction (CRI) [2]. Robots in environments with children have shown great achievements in areas such as entertainment, care, therapy, education, and social assistance. Furthermore, children have shown not only to be able to socialize better with robots, but also to improve their social skills with other children through the use of such robots [14, 4].

In this project we develop a game playing robot based on the TurtleBot platform that is capable of hide and seek and tag. This appears to be an unpublished application for the TurtleBot, though not new to robots in general [15, 16]. Completion of this project will include methods from the fields of mobility and intelligence. The robot will require a means of mapping its environment, localizing itself globally, and navigating its field. Also, the robot will have artificial intelligence capable of making its own decisions based on the game that the child wants to play. These features include an algorithm specifying how a hiding place is determined and an escape algorithm. Furthermore, the robot will be able to process natural language to make child-robot communication feasible.

Keywords: robotics, human-robot interaction, children

The remainder of this report is organized as follows. First, the simulated hardware used for this project is described. Second, the software that was prepared is discussed. Within this section the methods used for mobility, such as localization, mapping, and navigation are identified. In the next section the primary components defining the robot's intelligence are described. Test results are then presented. This is followed by some discussion, including a summary of the originally proposed goals of this work along with an evaluation of the progress made toward meeting those goals. The authors' contribution statements conclude the report.

2 Simulated Hardware

The robot identified for implementing this project was the TurtleBot 2 from Clearpath Robotics Inc [3]. It has a depth sensor which includes a rgb camera, bumper/switch sensor, and odometry for the drive wheels. An accurate model of this robot is available in Gazebo, a simulator frequently used for robotics. Typically with few and minor changes, systems developed for the TurtleBot within Gazebo can be directly ported to physical hardware.

3 Software

The software created for this project can be divided into a hierarchy of layers. First, the overarching state switching algorithm is described. Then the individual components used for mobility are identified. Details regarding the language processing and game algorithms will be addressed in Section 4.

3.1 Levels of control The software architecture provides two levels of control. Only one of them can be activated at a time. They will run based on natural language processing. The diagram (see Figure 1) shows the control process. Before the beginning of any of the levels of control, the robot has to perform the following required tasks:

1. Initialize the system: This includes turning the robot on, launching the initial files, and starting the child-robot interaction. In this task the robot greets the child and introduces itself.

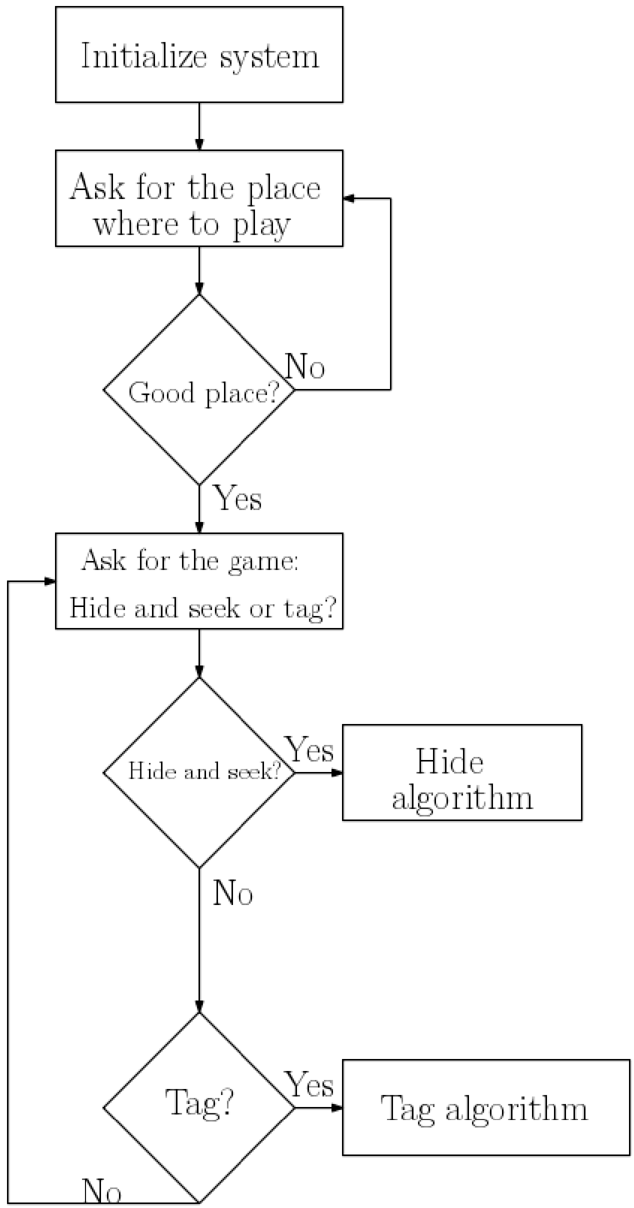
2. Decision process: The robot asks the following questions:

- Where do you want to play?
- What do you want to play?

The robot is able to process the answer from the child and make decisions. If the child wants to play in a place that is not safe, the robot will not allow it. Similarly, if the child wants to play in a place where the games can not be performed due to a lack of space or places to hide, the robot will not allow it. If the robot rejects a requested place to play, it will automatically explain the reasons to the child and ask again for a place to play.

3. Global localization and navigation: Based on the answer from the child, the robot will navigate from its initial position to the room where they will play.

Figure 1 Flow chart of the robot actions.



Once the robot is ready to start the game, it will ask the second question: “What do you want to play?" The robot will wait for the answer, and it will process the human language in the same manner as it did before in order to use it as an input to the system. If the child answers “tag," the level of control **Tag algorithm** will be performed. However, if the child answers “hide and seek," the robot will perform the level of control **Hide algorithm**.

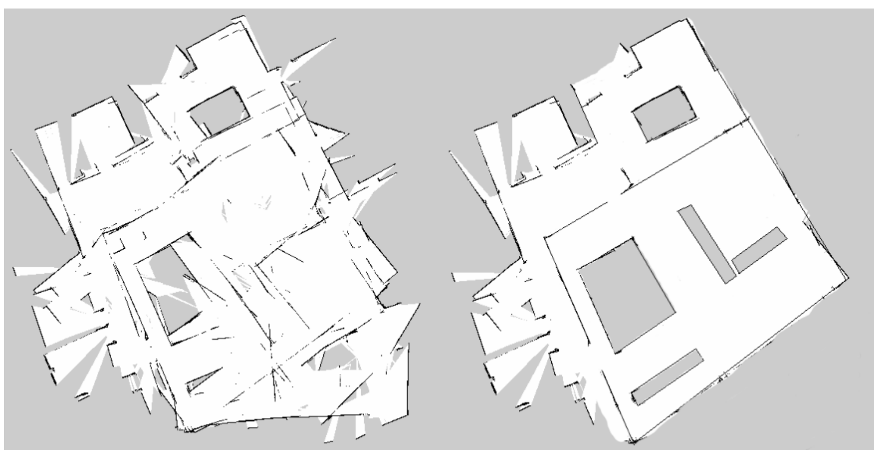
3.1.1 Tag algorithm During this level of control, the robot has to stay away from the child. The robot will perform navigation while sensing the environment to prevent a collision with obstacles. In order to avoid being cornered by the child, the robot will randomly choose different obstacles, and it will run around them. Once the child reaches the robot and says the word **tag**, the game will be finished.

3.1.2 Hide and Seek - Hide algorithm During this level of control, based on the obstacles' locations, the robot will hide in a place that will not be visible by the child. The robot will stay there until the child finds the robot or the waiting time is met.

3.2 Mobility

3.2.1 Mapping The developed algorithms were written in a general way so that any relatively large space where many obstacles exist or can be setup. For creating an occupancy grid map we used simultaneous localization and mapping (SLAM). The software package *gmapping* was specifically selected for this purpose due to its ease of integration with the TurtleBot and Gazebo as opposed to *Google Cartographer*, which was anticipated to be more difficult [13, 8]. Once the map was obtained, some editing was performed in order to clean up the open spaces and redefine the obstacles.

Figure 2 Flow chart of the robot actions.



3.2.2 Localization Because of its accuracy and ability to run in real-time a Monte-Carlo localization algorithm was implemented within this project [6]. This particle filter method has been previously developed in the ROS package *amcl* and was able to be applied directly to the project.

3.2.3 Path Planning Path planning is completed by the *navigation* ROS package. This package implements planning and navigation on two levels, global and local. Globally, Dijkstra's algorithm is used to generate a path [5]. Locally, for obstacle avoidance, a dynamic window approach is implemented [7].

4 Intelligence

The robot's decision making capabilities fall into three categories. First, the overall driver of this system is based on language processing. Next, the learning process for hide and seek is structured as a multi-armed bandit problem. Lastly, the robot's vision system is capable of detecting the child, a feature that is used in a supporting role.

4.1 Language Processing The package titled *pocketsphinx* can be used to obtain a string from speech input [9]. This package was chosen because it processes based on a local dictionary rather than requiring cloud-based resources like one alternative package based on *Google Cloud Speech*. A custom dictionary containing only the commands needed for this project, such as "tag," "hide and seek," "yes," "no," and the names of various hiding locations was developed. This limited vocabulary ensured more accurate speech recognition.

To process the incoming strings from *pocketsphinx*, a dialog agent was developed. Various queries are presented to the child, for example "Where do you want to play?", which then required that the child respond in confirmation. Due to the limited needs of this system, this was able to be implemented by programming a series of cases. The robot's need for text to speech was met by using the *sound_play* package.

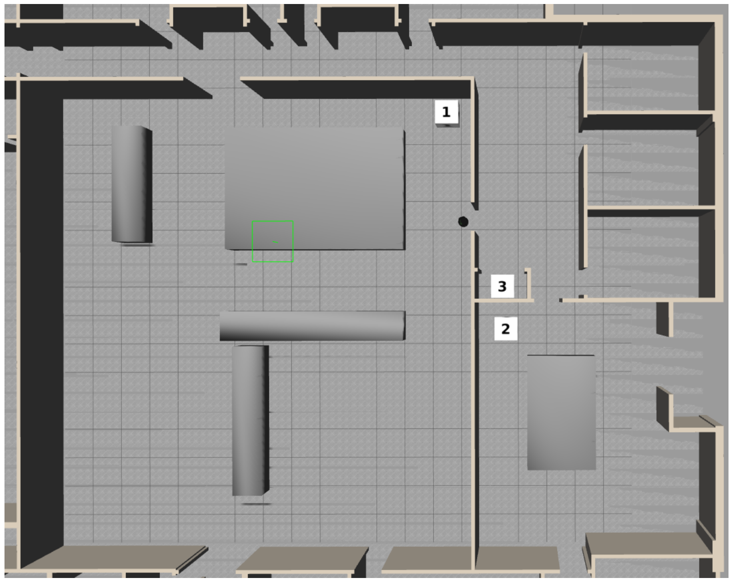
4.2 Multi-Armed Bandit The decision between hiding places can be framed as a Multi-Armed Bandit problem [11]. This problem type can be summarized as a decision process where all options have an unknown reward associated with them. The reward function for a given hiding location was specified based on a combination of hiding space features. These included the distance from the child's initial position, being out of the child's line of sight, and time passed before being found. These were scaled to generate a function as follows

$$reward = d + cd + td = (1 + c + t)d, \tag{1}$$

where d is the distance from the child's initial position in meters, $c \in [0,1]$ represents the coverage of a given hiding place, and $t \in [0,1]$ is the time passed divided by the total time allowed for searching, in this case 30 seconds. The coverage c was evaluated in several steps. First, the ground plane was removed from the depth camera's image by eliminating the pixels below a certain row. This assumes that all of the playing area is perfectly level. Next, assuming that the robot is facing the child's direction from its hiding space, any pixels that were nearer in distance than the known distance to the child's initial position were considered covered and counted. This value was divided by the total number of pixels, excluding the ground plane.

Because a reward such as this is not easily categorized by a probability distribution and especially since it depends on the child's learning, this problem can be further classified as an Adversarial Multi-Armed Bandit problem [1]. As repeated hiding location selections are made, information regarding the result is gathered and affects future decisions. The Adversarial Multi-Armed Bandit Problem is often solved using an algorithm titled EXP3, which stands for "Exponential weight algorithm for Exploration and Exploitation." This algorithm computes a probability distribution based on all previous rewards and all yet unknown possible choices. An action (hiding location) is chosen based on this distribution. The reward associated with this action is collected and assimilated into the knowledge base. Then the process repeats for a new hiding location. This component of the system was coded in Python. In summary, this component determines and publishes a set of goal coordinates for the robot to hide at and after the robot is found evaluates the reward information gathered to improve future hiding decisions.

Figure 3 Gazebo world used for testing Rabbit



4.3 Human Detection Several components of the hiding algorithm require knowledge of the child's initial location. This can be determined under the assumption that the child does not move while counting. Because the recognition required identification of an avatar rather than a human, the package titled *find_object_2d* was identified as suitable for the required level of accuracy instead of using a package built purely for detecting people such as *cob_people_detection*. The package *find_object_2d* executes the ORB (Oriented FAST and Rotated BRIEF) algorithm in OpenCV to identify an object in two dimensions [12]. An average of the identified points is then taken, and the distance to that point is measured by the depth camera, giving a three-dimensional object location. The three-dimensional location can then be transformed to the map coordinates for use by the robot, a feature included in some sample code of the package. To use this package effectively, several images of the child in varied lighting conditions are required.

5 Results

5.1 Test Conditions The system was tested in simulation in Gazebo [10]. The world developed for this purpose was a section of one of the default worlds, Willow Garage. Three primary spaces were identified within this world and are numbered in Figure 3. Several features were built into these rooms. First, obstacles were added to provide interest and a level of difficulty to the game of tag. Second, both good and bad hiding places can be selected, making the resultant learning clearly identifiable. Third, there are several different rooms, providing the opportunity for evaluation of the dialog agent in allowing the child to select where he or she wants to play.

Because the testing was done in simulation, a means of representing the child needed to be created. This was done by a combination of methods, in the end making the authors able to function as the child. First, a package that included a controllable human-shaped avatar was identified, *bwi_gazebo_entities*. By spawning this model within Gazebo and adding a package for teleoperation of this Gazebo object *teleop_twist_keyboard*, a physical representation of a human, though not child-sized, was generated in Gazebo. To address the need for dialogue, an external microphone was able to be used. Limiting the view to only that of the child's vision, a camera mounted on the avatar's head was used as the only vision source during the game. Lastly, because the avatar is unable to directly interact with the TurtleBot in simulation, a ROS topic was generated that when published from the command line would signal the end of the game.

Initializing the system required that global locations be hand-selected for both the tag and hide and seek games. For tag these were the corners of the room of interest. For hide and seek five hiding place candidates were identified per room. Also, the initial navigation goals to travel to the room selected for play to begin were hard coded.

5.2 Evaluation Rabbit was assessed both as individual components and as a whole. Evidence provided by each test shows that the system acts as expected.

5.2.1 Dialog The child-robot dialog performs extremely well. The default voice was changed to a more friendly one. The robot gives the time required for the child to answer. Through the dialog the robot switches between the different tasks as expected.

5.2.2 Map The generated map was good enough. However, it can be seen that it does not match the developed gazebo world perfectly. Sometimes the robot gets lost. Nonetheless, the particle filter algorithm for the global localization solves the problem.

5.2.3 Tag The tag game performs as expected. Since the algorithm requires the robot to be close to a specific obstacle, the robot almost never gets lost during this level of control.

5.2.4 Hide and Seek Though there was not enough time to perform the number of trials required to thoroughly evaluate the multi-armed bandit algorithm, several benchmark tests predict that it should be effective. First, the multi-armed bandit algorithm was tested with a static reward function. Under these conditions, one hundred trials could be completed within a couple seconds. The results showed a clear preference for the action with the highest reward and selected the action with the lowest reward less frequently.

The dynamic reward function that was then developed was also tested independently to observe if it gave consistent results. It was found to accurately represent the features of a hiding space. However, two cases under which the reward function was not consistent were also identified. First, if the robot happens to hide at a location where the child is within view and closer to the child than the initial position, the child will be perceived as part of the robot's coverage. Second, the reward function depends on the robot hiding in an orientation where the camera faces the child. If the global localization is not accurate, the robot will not reach the correct orientation to make the reward function the most accurate.

The third test was for the complete hide and seek algorithm, which performed as expected. After the hiding algorithm was started, the robot did not select a hiding location until it had identified the child. Next, the robot chose a hiding location according to its previous knowledge, a probability distribution, and navigated to that location, orienting itself toward the child. When the robot received the signal that it had been found or when 30 seconds had passed since the robot reached the hiding place, the reward function was evaluated. The overall reward history was then updated to maintain what had been learned for the next game. These three tests suggest that the completion of numerous trials in the simulated world should lead to a clear learning trend.

5.2.5 Overall System The language-based state switching system was tested under various combinations of selections made by voice command. It consistently initiated play in the correct location. Also, the game selection options began as soon as the robot reached the desired room. Games were started as directed. Upon completing one round of the desired game, the robot returned to its introductory state, such that all new selections can be made.

6 Conclusion

6.1 Summary of Goals At the beginning of this project four goals were identified for the successful completion of the proposal:

1. Prepare a mobile robot that speaks selected queries relevant to the project and reacts to simple, vocal responses from a human (Tag and hide and seek).
2. Compose an algorithm that effectively avoids a human (Tag).
3. Formulate an algorithm that searches for a human (Hide and seek).
4. Develop an algorithm that selects hiding places for the robot and progressively improves these hiding places through learning (Hide and seek).

With the exception of implementing the system on physical hardware, goals 1, 2, and 4 were completed. Goal 3 had to be delayed due to time constraints. Implementing goal 3 in the future, however, will not be overly difficult because some key components have already been completed, such as a means of identifying the child's location with the vision system.

6.2 Discussion This project provided the opportunity for us to both develop practical skills and extend our knowledge beyond the topics covered by EEC 793: Autonomous Intelligent Robotics. The extensive variety of topics covered by the project provided us with much experience in combining many packages and writing code in a modular form. We improved our abilities to write components that could communicate across the ROS platform and test components separately. Extensions to the course topics included Gazebo and learning methods. Throughout the project we learned how to use Gazebo more effectively, including the basics of editing world and object files. While the course covered various learning techniques, no assignments were completed on these topics. Therefore, this project allowed practical experience with one such technique. Additionally, our current personal research topics involve human-robot interaction. Since this project has been focused on child-robot interaction, it has helped strengthen our previous knowledge with new approaches and has provided a lot of motivation for new research directions.

7 Statement of Contributions

7.1 Humberto De las Casas Humberto developed the child-robot dialog and the speech processing. It includes the decision making of the robot based on order of importance. The decision process gives the highest priority to safety, next to availability, and finally to what the child wants to do. Based on the feedback from the dialog, the algorithm performed switching between states until starting the levels of control. He was also responsible for building the environment in Gazebo. Later, he performed the mapping of the environment. Since the resulting map was not totally accurate, he worked together with his teammate to edit it in order to make it easier for the robot to process. Additionally, he performed the tag algorithm for the tag game.

7.2 Holly Warner Holly developed the hide and seek game, including the learning algorithm, navigation commands, and reward function. Associated with this, she prepared the code for detecting the child, which required modification and integration of a published package. She was also responsible for generating a controllable human avatar in gazebo and resolving its conflicts with the simulated TurtleBot's published and subscribed topics. Smaller tasks assigned to Holly included assisting with editing the map image and learning to use the *actionlib* stack for navigation commands across the project.

References

[1] Auer, P., N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "Gambling in a rigged casino: The adversarial multi-armed bandit problem," in *Foundations of Computer Science, 1995. Proceedings., 36th Annual Symposium on*, pp. 322–331. IEEE, 1995. 5

[2] Child-Robot Interaction, "Research, methodology and best practices," `https://childrobotinteraction.org/`. Accessed March. 2018. 1

[3] *TurtleBot: Open source personal robotics platform*, Clearpath Robotics Inc., URL `http://bit.ly/1L2FIzG`. 2

[4] David Feil-Seifer and Maja J MatariÂt'c, "Human robot interaction," `http://robotics.usc.edu/publications/media/uploads/pubs/585.pdf`. Accessed March. 2018. 1

[5] Dijkstra, E. W., "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1 (1959), pp. 269–271. 4

[6] Fox, D., W. Burgard, F. Dellaert, and S. Thrun, "Monte carlo localization: Efficient position estimation for mobile robots," *AAAI/IAAI*, vol. 1999 (1999), pp. 343–349. 4

[7] Fox, D., W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4 (1997), pp. 23–33. 4

[8] Hess, W., D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2d lidar slam," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1271–1278, 2016. 4

[9] Huggins-Daines, D., M. Kumar, A. Chan, A. W. Black, M. Ravishankar, and A. I. Rudnicky, "Pocketsphinx: A free, real-time continuous speech recognition system for hand-held devices," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 1, pp. 185–188. IEEE, 2006. 5

[10] Koenig, N., and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, pp. 2149–2154. IEEE, 2004. 6

[11] Robbins, H., "Some aspects of the sequential design of experiments," pp. 169–177 in *Herbert Robbins Selected Papers*, Springer, 1985. 5

[12] Rublee, E., V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *Computer Vision (ICCV), 2011 IEEE international conference on*, pp. 2564–2571. IEEE, 2011. 6

[13] Stachniss, C., and G. Grisetti, "Gmapping project at openslam. org," 2007. 4

[14] Tanaka, F., A. Cicourel, and J. R. Movellan, "Socialization between toddlers and robots at an early childhood education center," in *National Academy of Sciences of the United States of America*. 1

[15] Trafton, J. G., A. C. Schultz, D. Perznowski, M. D. Bugajska, W. Adams, N. L. Cassimatis, and D. P. Brock, "Children and robots learning to play hide and seek," in *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, pp. 242–249. ACM, 2006. 1

[16] Volkhardt, M., S. Mueller, C. Schroeter, and H.-M. Gross, "Playing hide and seek with a mobile companion robot," in *2011 11th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 40–46. IEEE, 2011. 1

Acknowledgments

We would like to thank Dr. Shiqi Zhang for his guidance during several insightful conversations throughout this semester and for providing the ROS package used for generating the human avatar in our simulations.

De las Casas
Department of Mechanical Engineering
Cleveland State University
2121 Euclid Ave
Cleveland OH 44115
USA
h.delascasas@pucp.pe

Warner
Department of Mechanical Engineering
Cleveland State University
2121 Euclid Ave
Cleveland OH 44115
USA
h.warner@vikes.csuohio.edu