Multi-Robot Human Communication

Maciej Musialek

April 15, 2016

Contents

Chapter 1

Introduction

Communication has been a subject of research for a long time now. With the development of computers and robotics a new area was opened regarding communication between humans and robots. With the addition of job automation allowing robots to perform tasks on their own and take instructions, the communication started playing an even bigger part in job distribution. Humans handle the job distribution and robots would perform tasks carried onto them.

This project will also focus on turn-taking communication for energy limited robots with an addition of trying to give some decision power to robots, like proposing potential moves rather than list all of them. It focuses on creating a robot platform with a user interface that will allow testing how single human interaction can improve multi robot collaboration. This will be done by comparing different levels of freedom to robots from no control to complete control over the environment and their choices. This was largely motivated by a paper done by Elizabeth I. Sklar and M. Q. Azhar [1] in their paper on Argumentation-Based Dialogue Games for Shared Control in Human-Robot Systems.

The aim of their project was a proposition of human-robot dialogue that would give an outline to how such communication could be implemented. With this a proposition of such communication was given with different layers of communication taken into account that describe the communication. In further research using their system, the researchers found that humans trust robots more with their decisions when such dialogue is implemented.

This leads to the main motivation of the report, which is to develop a real-time version of the dialogue system where robots interact with a human team to give even a better picture of how this dialogue can affect decisions made by humans. Such an environment could shed more light in argumentation research where in certain scenarios trusting the robot more in the environment could be of key to maximizing results. Such situations could involve search-and-rescue missions where quick correct decisions need to be made in order to maximize the number of victims saved and such argumentation from robots could pose an ideal scenario for when humans give in to the pressure and machines can still do the calculations.

One last motivating factor for the project is that robotics is a growing field of technology that is getting more accurate every year. Robots are beginning to recognize objects, with recognition that could soon approach human perception and later on go beyond it. In such a situation, having strong research done in human robot communication could pave the way for quick implementations of systems where situations might require a more focused approach. This can range anywhere from search and rescue to self-aware cleaning robots at homes that propose which rooms are the dirtiest by a quick glance of each of them.

This leads to a conclusion that a framework needs to be developed for real-time simulation of robot environments with dialogue plugins being inserted to suit the researchers needs. For this project, a stack of software will be created ranging from back end Robot Operating System simulation software to a Graphical User Interface that tells the user where each robot is and allow commands to be given for specific robots. Such an environment is going to focus on allowing maximum freedom to the researchers to ensure that their dialogue research is only limited by what kind of dialogue system they would like to set up. The next sections of the report will focus on giving a concrete description of how this system was developed and how researchers can use it to help their work.

Chapter 2

Background

Before setting off to outline what has been finished in the project and how certain choices were made in comparison to others, the report will be discussing the current situation of the robotics community and the human-robot collaboration in general for how the robots communicate with each other.

With this in mind, it is important to outline what ideas are currently looked at and how they influence the field of robotics and more importantly, what they bring to the project. In the following sections, an outline of two main areas of research will be evaluated with for human-robot collaboration and research that was focused around the ROS environment.

2.1 Related work in Human-Robot collaboration

Currently research is greatly focused on human-robot collaboration for humans and robots in the same environment, working simultaneously on a task together. This in turn makes the current problem more unique. The problem we consider in this paper works on the assumption that humans and robots are in separate environments, and with that the robot can potentially have a lot of control over the environment. The following subsections give a more precise background in research of human-robot collaboration.

Same space collaboration

In their papers Liu et al.[2] and Govindarajan et al.[3] discuss collaboration in same space.

In the first paper[2] the collaboration is outlined by robot working together with the human. The setup of the experiment was to give each human-robot team a group of tasks that either required one agent to complete or a joint task for both of them. The aim was to finish a list of tasks as quickly as possible. The measurement parameters were mainly the fluency of collaboration outlined by

Hoffman's metrics. Their results found many conclusions; one of which was that humans preferred working with robots that could infer plans based on human movement. This gives an implication to the project that ultimately guides the project to try and develop a robot that could propose movements as to giving the human full power over the environment and the process. In that way it can be considered good practice to let an idle robot move based on its own path planner to the closest room if it was neglected. This of course, is subject to human preference but allowing such functionality could be considered.

In Govindarajan et al's[3] paper the main focus of research was search and rescue actions where robots infer different paths from where the human is headed thus maximizing the search space of a room. To develop an algorithm that would handle the robots, the researchers used the ROS platform. In their results they have found that using complementary homotopy classes for intelligent path finding increased the performance of robots helping them make more intelligent decisions. While not particularly useful for robots that search different rooms, using homotopy classes for finding out which robot should potentially reach specific rooms could be useful when making plans that revolve around finding maximum number of rooms helping robots achieve more efficient paths. Additionally another area of research would be with large amount of robots. In that research the robots would use homotopy classes to make decisions based on first movements of robots to try and predict which would be the next rooms that the current robot would choose. With that, the robots would move away from that area that will potentially be visited anyway to start exploring other areas.

Spatial recognition

Goto et al.'s[4] paper work on maximizing fluency in human-robot collaboration and identifying challenges posed by robots when it comes to recognizing parts that would be required for assembly of parts. To achieve this they focused on a finite state machine approach for robots to help with table assembly. This can be considered a more limited approach in terms of robot intelligence since robots are limited in the amount of states they will be in. In their results they managed to see limitations with the robot both recognizing the human action and with the scalability of the software due to most of it being hand made beforehand. The paper mainly focused on being able to achieve the task in comparison to achieving the task efficiently or effectively. With that in mind it poses considerations that need to be taken when looking at robot design one of the most important being a challenge being the robot understanding what it sees and how limited they can be when it comes to object recognition.

Modalities of human preference

While in their paper Fiore et al.[5] did not use the robot operating system, they embarked on a task that would prove that robots are capable of completing collaborative tasks based on human supervision. In this sense, their paper is

very related to research outlined in this paper. Fiore et al. Propose a complex system that is built on sophisticated software for intention inference, path planning, task execution and communication. In their results they have proved that the system is capable of: handling joint goals and actions, handling users preferences, handling agent beliefs and monitoring human actions. With that in mind, a multitude of research has been proposed in creating human-robot collaboration and proving that the tasks are in fact possible complete.

Conclusion

It seems that a lot of current research has put a great focus on proving that human robot collaboration is indeed possible with a huge variation of approaches between each paper. While quite a new area, it is important to note that currently, there doesn't seem to be a greater standard in how research is carried out with researchers assembling software based on their research preference. It does not dispute the fact that every research paper proposes new considerations in human-robot collaboration and that possible improvements are drawn based on their approaches.

This leads to a conclusion that Human-Robot collaboration is a very undeveloped field and will start presenting more exciting opportunities in the future for researchers to look at. Currently it seems that a great deal of effort is in ensuring that fluency between agents is maximized making the experience faster and more enjoyable for the participant in the study and potential agents for developed systems in the future.

2.2   Research in ROS environment

In comparison to previous research that focused on proving various possibilities of robotics systems, research that focuses on using ROS environment is mostly aimed towards developing various frameworks to ease the use of the environment to achieve certain results. As such, it can be compared to human-robot collaboration research as focusing more on solving problems than proving. Following sections outline various works that used the ROS environment to achieve their goals.

Frameworks

In their two papers Fok et al.[6] and Liang S Ng et al.[7] focus on creating two separate networks that provide an interface for grabbing robots and control for complex whole body robots with multiple parts. In both of the papers, ROS was being used which stands to justify just how powerful ROS can be in manipulating various environments. In fact, the paper on creating a hardware and software problem for intelligent applications, is very similar to the approach this report is focused on. The simulation platform is in fact exact with the only difference being controlling single robots forwards and backwards rather than using path planning for the problem. From these findings, given time, the report

can be further studied to include complex body robots that can be controlled over the cloud with hand held devices. This only signifies the availability of technology for implementing complex robot systems using ROS.

Domain specific research

Deusdado et al.'s[8] research focused on creating an aerial-ground teams in ROS for systematic soil and biota in estaurine sampling. Mario Vieira et al.'s research further focused on creating applications for monitoring human daily activity and risk situations in robot-assisted living. Both of these papers can be considered extremely centered around the areas that they focus on and show the variety of scenarios that ROS can be used for. Both teams achieved great success in their plan to centralize use of ROS for their respective goals and showed how useful they can be in further research. The first paper gives us an idea of how ROS can be used to implement robot teams, giving us ideas about how to space out cloud platforms to further enrich the environment while the second, allows us to see how to use robot sensors to view activity of humans.

This knowledge, can in fact be extrapolated to the project in future works to give a richer environment. One where each robot is directly responsive to actions of the other through the use of sensors rather than shared goal reaching creating a more human like collaboration not only between the robot and the human but throughout the whole team. It is important to note that while right now the robots focus is to reach desired destinations and share data to produce better maps. An alternative of proposed research above shows us that using a more intuitive approach can be just as useful.

Conclusion on ROS research

It is very easy to see that ROS research and implementations revolves around creating better frameworks and centralized domain problem solving. It is a very young field that so far didn't set standards towards what would be the best approach so that researchers could start tackling these issues and start improving on standard solutions. This in turn, emphasizes that the field is going to expand creating more elaborate solutions and frameworks for specific domain problems. At the time this report was written, most papers presented here are papers that came out in 2016th to ensure that the image of current affairs is as accurate as possible. With this information, as presented above ideas can be drawn about possible improvements to the implementation that could in turn create an environment where the collaboration is as fluent as possible with robots taking same type of initiative that normal humans would.

2.3 ROS challenges posed by its creators and community

Currently ROS is an overwhelmingly expanding software with new additions being added. ROS started in 2010 and already had 9 releases with the 10th coming out in May 2016. This sort of expansion rate pushes developers to relearn the structure and standards set by ROS every few months, in turn making the frameworks obsolete every 8 months or so IF they used code that ROS creators deemed unnecessary in future releases or made it deprecated through certain decisions. This challenge is actually an obstacle for ROS creators themselves as in some examples, the Wiki pages can reference functions that can be considered deprecated in future releases. This in turn makes developers turn to the ROS community which is limited to the small group of robotics specialists using ROS that are willing to help on answers.ros.org.

The community itself is compromised of a fairly small amount of user group that cannot be compared to regular software developer websites like StackOverflow or BigResource where commercial development and advice seeking can start to compare itself to a competition of its own. This produces problems when a regular developer that didn't have much experience with the environment, thus making it harder to develop in this network.

It is also extremely important to note that at this point in time there is a huge deficit in the amount of available resources the developer can reference to when it comes their struggles. The few books that do exist to help users develop their knowledge are few and their knowledge as previously stated can become obsolete extremely fast due to ROS's quick expansion rate. When this report was created, there were also no current frameworks available for faster safer development or third party libraries to help a developer create a bigger understanding which means that the wheel of development for ROS is mostly reinvented every time a new idea is presented for development. This leads to a conclusion that the outlined specification will be extremely challenging posing a lot of stalls for implementation of the software.

2.4 Conclusion

It is clear that ROS is an amazing platform offering its users a great diversity in use. The research in human-robot collaboration and ROS is starting to expand the use of ROS in human-robot collaboration is starting to slowly become a standard. This in turn means, that selecting ROS as a developer platform for robot simulation is definitely a right fit for this project.

With the previous section, it is clear that there will be challenges in this project that might in fact not get fulfilled simply due to lack of resources that will be able to acquire and through that, a careful consideration will be taken to ensure that the goals are realistic and not reaching outside the scope of the possibilities.

Chapter 3

Specification for the report

Many projects are constantly being conducted in order to give insight into how Human-Robot interaction should be conducted and its effects on the efficiency of the collaboration. This requires that each report needs to give a solid specification of how the environment is set. This in turn provides basis on which future research can judge the results and compare them against others fairly. The project will be split into three parts each of which is defined below:

The Robot environment: The robot environment will be the center of emulating the environment. It will be responsible for housing the Robot Operating System server with maps, odometry for robots and mapping the robots inside a map so that they will be able to move around and accept commands from outside sources i.e. the GUI and transfer these commands into robot movement and task allocation. Below is the outlined specification for the environment.

- An environment that allows simulation of a robot on a 2-D plane with the map received from project supervisor

- Accepting commands that set a goal for specific robots for them to reach goals

- Each robot having their distinct levels of energy based on how much they moved

- An ability for the robot to realize which rooms are closest to it

- Robot not being able to tell if the treasure is useful but asking the user whether or not to retrieve it.

- Robot is able to take pictures(Which costs energy) to the user so that the user can decide treasure retrieval

- Allow spawning more robots in the environment.

Graphical user interface: The Graphical User Interface is a crucial aspect to how the task distribution will be managed. Whether it is click and go or hot

keyed commands is a question that will be crucial here. The human factor needs to be taken into account as basis for testing but having a user interface that doesn't allow comfortable work will only result in slowing down the human factor and making the results biased towards the robots working by themselves. Below is the outlined specification for the GUI.

- The interface should be able to contact the server with current location of the robots

- Provide results at the end of the simulation

- Should suggest the most efficient way for the robot to reach its goal.

Modes for testing collaboration Testing the environment that was developed to the specification is the main focus of this project. The tests of this project will focus on how the human factor influences robot collaboration in different settings as well as see how robots will behave by themselves in different contexts be it going to random rooms or rooms in closest proximity. It is also important to note that contextual situations are what will make the human factor more important. These situations could involve: Treasures only available as spread resources or resource grouping where robots with closest proximity algorithms would only be able to retrieve the same kinds of treasures which might not be an objective here. Below are the outlined tests for the environment to see how human factor will influence robots and how numbers of robots will factor into the equation.

- First Test - Testing how robots move on their own in the environment and how efficient they are.

- Second Test - Adding human interaction so that the person can decide on where the robots will proceed.

- Third Test - Test how a number of robots influences the usefulness of adding the human factor into the equation.

Chapter 4

Design

4.1 Protocol for communication

This section will ideally outline how the Server Java class works with consid-
erations taken into commands that the environment will embrace to ensure
communication. Based on that, figures will be created to ensure that the mes-
sage about message passing is clear. Then a more thorough explanation will
be given about how communication is handled in ROS and how this project is
taking advantage of it... This will include: Publishers and Subscribers in ROS
as well as talking about how C++ servlet will have to be created and separated
from ROS codebase to ensure that each side can be adequately changed without
affecting the other. Also quickly discuss that such implementation will not be
possible in Java as advertising and publishing is not as possible however a quick
workaround using Observer design patterns can be implemented and with that,
a similar approach can be taken.

4.2 ROS Environment

Discuss the ROS environment with how there are two options for the simulation
design(Gazebo or Rviz) and Navigation stack that could potentially reduce the
need to hard-code solutions. Discuss then that navigation stack is a better
option considering flexibility when the environment is eventually to be physical
and as such flexibility on the ROS environment is key. Also discuss that Gazebo
poses limitations that require to be switched on which could potentially be
resource expensive as Rviz can be switched off and will only be there for testing
stages when setting up the navigation stack.

4.3   GUI for User and Hider

Present with diagrams the idea behind the GUI and then discuss possible implementation environments that could be there to use it. These can in fact be done in various programming languages. The multitude of environments can be brought down to: C++, Java, Python, JavaScript all of which propose something to the table. C++ is great for memory management and performance management, if the server was also in C++, then this would probably be the language of choice. Java offers built in GUI and it's communication protocol agrees with the server so disconnections on socket dumping is actually very possible and a very sought after feature. Discuss that the GUI however is limiting unless using external libraries such as JavaFX that implements web languages such as JavaScript inside. This could be a great idea but adding extra libraries to learn more about the project is beyond the scope of the project for the time frame of the project. When it comes to JavaScript, there is an AJAX functionality built in but this is a RESTful approach that the server does not at the time of the project implement, hence API calls are currently not supported and additional platforms for socket implementation would have to be created. Also, to run JavaScript in such a manner would require to use node.js which would increase the project stack. We can lastly consider Python that has great support for 2D game implementation as can be seen in the book of Python for beginners where a library for creating sprites dynamically is greatly explained. Python also has great support for running off the go on UNIX based systems, this however is not of such importance as only ROS needs to be on a Ubuntu Linux machine in comparison to the GUI that should ideally run on any machine. Most machines have Java but not many have support Python considering that majority of systems run on Windows. With that conclusion a Java implementation seems to be the best decision.

4.4   Testing design

This generally will talk about various algorithms and where they should be stored inside the ROS network when the implementation time comes for Robots to start moving on their own. Discuss where this feature should be stored taking into consideration how the ROS stack operates and whether it would be more comfortable to do this in a ROS class that is similar to ROS

Chapter 5

Implementation

The fully implemented product has been derived by the guidelines received from Elizabeth Sklar, King's College London who guided the project and provided a Server package that in turn is now the main communicator of information of the finished product. This product can be broken down into several components these being(For a better explanation please refer to figure 1):

- Simulation ROS stack - A package composed of: Server Control, Robot Control and Robot Simulation.

- Server - A package composed of communication protocol classes.

- Hider - Composed of classes responsible for server connection and Hiding the treasures.

- User - Composed of classes responsible for UI, server connection and Remote Robot Control.

In following sections of this report, a more broken down structure to provide insight into how each of these is achieved.

5.1   Simulation ROS Stack

This chapter will outline ROS and how it works with information for the user that will ensure they can use ROS and reducing the time needed for reproducing results.

5.2   Server

Here I will focus on outlining the structure for the server. I will try to make it quite brief as that server package was not my implementation but I will go into some detail about possibilities it provides and the protocol it follows.

5.3   Hider

Currently Hider is a very static class without any GUI. I will try to develop some Hide treasure GUI for last calls. Later on I will explain its function, how the classes are broken down and reasons for choices.

5.4   User

Finally with all the required functionality up I will explain a bit more about how the user environment works. This will be done through various figures of classes, telling how the communication works, explaining the use of threading and design patterns to ensure that the environment is called accordingly and reasons for choosing Swing Java over JavaFX etc. etc. etc.

Chapter 6

TESTING - Modes Specification and analysis (Finish this after next meeting with what is had)

Modes specification and analysis will focus on altering the code in such a way that will allow provision of different results. I will provide algorithms for three different modes that allow an unbiased picture that will show how human interaction and robot collaboration will influence the results. This part will definitely allow for some research work.

6.1   System against Requirements testing

Here I will weigh what the system managed to achieve in comparison to the requirements set out in the specification and design chapters.

6.2   ROS Testing

This will be testing on the ROS environment to fish out any bugs in the code to ensure that software is of quality.(Loop missing time lookups)

6.3   Server Testing

Testing server connections to fish out any errors for future fixes with regards to how connections are handled. (%%shutdown messages could be improved on). What happens when we suddenly close the server.

6.4 GUI Testing

Testing the GUI, resizing etc. and testing different situations(Do the dialogs pop up correctly, Can user interrupt traveling robots etc.) What happens when the GUI closes by itself.

6.5 Use Case Testing

Testing the full implementation when switched on the application correctly and seeing if at any point the program crashes. Do these about 5-10 times in a row and see how many times there is a problem with software(if there is any). With these results elaborate on what could be causing issues if such present and include it in future works.

6.6 Known problems

GridLayer

Talk about how the environment doesn't let robots update the costmap but are rather reliant on the static map which means that at current stage they are rather reliant on the environment being unchanging. They can also overlap in their travels.

Environment mostly simulated

A lot of the work to create the game is simulated like energy levels and taking pictures. Elaborate on how this might pose difficulties in the future.

ROS package stack

Talk about how libraries and executables are very rarely properly working together and how inclusion of external classes can lead to errors. An example of this can be integrating GridLayer using robotPosition class to set new obstacles.

System currently made to run in an uninterrupted environment

As of right now closing either of the applications could cause an error forcing the user not to notice any error until sufficient time has passed to realize there is a fault in the system. **Will try to fix this to ping everyone occasionally or make a note that Server could notify all other nodes of one node disabling to halt all actions and ask for a restart.**

A very strict Server

This actually ensures that no impostors attack the server however more frequent pings could be specified for certain applications where if the application

is switched off, the user will not have to wait until the server cleans out the old GUI before a new one can be switched on.

Chapter 7

Evaluation (Finish along different modes for testing)

7.1 Project Progress Evaluation at the time of report

This section will largely focus on evaluating the progress made during the report. It will discuss the original goals against what was achieved and what was missing in the project. What things could have been done better? What things have been done well? Why have some things not worked out quite the way they should(Think robots overlapping each other or not constantly upgrading the map or robots not proposing what rooms to go to). Also discuss the help that was reached out for and critique the ROS community as they are difficult at times. Discuss where the project could be applied? Also make a note how compared to real life this map was really small and usually robots will have much bigger environments to transverse through. This means that if time is of essence, supporting multiple robots is absolutely critical and being scalable is another issue to look at which the project did focus on in code design.

7.2 How the project compares with other research

Following that, discuss how this project compares to other papers previously stated in the background and as such. In what way was this project something completely different(Think humans not in the same environment as robots). How was it similar?(Think about the collaboration papers focusing on fluency). For more ask at tomorrows meeting what can be discussed here. I could also discuss the inverse property of the project against projects like Amazon Drone Delivery.

7.3 Future possibilities for expanding the project

Should I write something here?

Minimizing dependencies

Complete care was given in the project to made the code as scalable as possible through the use of non discriminating functions(Functions that apply to all robots) through Object Oriented Programming etc. But some functions in ROS do require the user to create separate file for each robot and as such, more elaborate ways could be thought of to allow a more Object Oriented approach to be taken. With that, the scalability of the project could increase to the point that it would turn more into a non-deterministic framework that allows plug-ins to be taken in.

Automatic area clustering of robots

Based on research done in Sensor networks, energy limitations are a huge concern there and multitude of research was carried out on maximizing the lifetime of networks. As such, motivation can be given to displacing the robots throughout the map to maximize their lifetime and guide the user better around the game. This could be done through the use of clustering algorithms for robots to negotiate on which part of the map they will focus on. In this project this will take on a bigger challenge as in some scenarios where environments are hostile robots could be lost along their path and as such two robots could agree to search regions of the map.

Start using physical robots for the Game

The project stopped at using simulations to run the game. In this way it is not much different than using simple game software to simulate such an environment. Creating a framework using ROS actually allows the project to transfer from simulation to real life is actually fairly straightforward with the foundations already built up. With the robot controller, all functions are sent to specific robots odometry and since odometry will always be present the next focuses in the project could be to start implementing real life robots with maps uploaded onto them.

In such scenarios a lot of issues would be overcome with robots creating real time data and submitting them to the map as well as avoiding obstacles that could be other robots. In such a case study users would have access to robots but not see the maze themselves. It would add excitement to the game.

Expand the GUI to JavaFX

Currently, the GUI is built using Java Swing which in itself is a very formal language that is usually used to build business like components with little support for drawing extra objects or on panes. There are packages such as AWT but

their use is widely discouraged by the community and could stall the project more.

Instead a proposition of using JavaFX is in place. This package is slowly used more frequently to add flexibility to otherwise difficult to handle Java GUI design. In the end the projects audience is what will determine the projects quality so ensuring a solid GUI could be the way to move forward.

Add Phone support

While a minor feature requiring to only boot up another GUI, gaming on the phone is becoming more and more of a standard. With that in mind giving users the freedom to not carry around a laptop to events where the game might be played, using hotspot routers for the game and allowing apps to play the game on top of computers could be in fact a very useful feature. There are numerous possibilities for implementing such a feature but there are also technologies that allow writing applications that export to all phone platforms. Discussing them however falls outside of the scope of the project.

Flexibility through Multi ROS Support

If the environment stayed as it is, creating VM's that run multiple ROS environments and handle them accordingly could be an approach to create a gaming environment that multiple users could enjoy online whenever they would choose to wish so. Such environment would work by using name assigning and would require adding another Server node that would handle multiple instances of the current server class that would populate itself over the machines ports with sockets.

This would create a scalable game environment that potentially an unlimited number of users could use. This could open opportunities for competitions in the game where based on a scoring system of choice users could compete in robotics challenges that use actual real robot operating system as opposed to simulations that wouldn't otherwise give them an experience of using a ŕealŕobot in its environment.

Deep sea exploration

ROS currently lacks the support for 3D path planning. This is not to say that 3D control, that is actually available but in the future, plans might extend to ros being able to support 3D path planning and then another world of opportunity will open up in Deep sea exploration. The current project allows for communication between ROS and whatever GUI is presented so its not difficult to see how inputting the right packages for such implementations could be extremely useful. Already a lot of research has been going into node deployments in deep sea and how to use these to tackle communication however, map building of deep sea ocean and understanding more about what's beneath us is an area of research that could benefit from the framework suggested in this project where

robots are given coordinates through a server allowing for a customizable GUI on the side of developers for such projects.

Search-And-Rescue Scouting

This is actually an extension to the paper by Govindarajan[3] where robots would assist a person to minimize time spent in a single room during search and rescue missions. This project could be an extension to that but much rather use robots for pre-made environments that suffered some kind of trauma to weight risks of assisting certain people and search for survivors in places that have the best chance of making it out. Building maps of risky environments could give rescuers chances of creating robust plans that would maximize the number of people saved in such harsh and difficult times.

Chapter 8

Conclusion (Finish this by end of April for submission and rewriting code.)

Bibliography

[1] Elizabeth I. Sklar and M. Q. Azhar, Argumentation-Based Dialogue Games for Shared Control in Human-Robot Systems, *Journal of Human-Robot Interaction, Vol. 1, No. 1, 2012, Pages 78-95. DOI 10.5898/JHRI.1.1.Tanaka.*

[2] Chang Liu et al. Goal Inference Improves Objective and Perceived Performance in Human-Robot Collaboration, *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016), J. Thangarajah, K. Tuyls, C. Jonker, S. Marsella (eds.), May 913, 2016, Singapore..*

[3] Vijay Govindarajan et al. Human-Robot Collaborative Topological Exploration for Search and Rescue Applications, *Distributed Autonomous Robotic Systems Volume 112 of the series Springer Tracts in Advanced Robotics pp 17-32 15 Jan 2016.*

[4] Hiraki Goto et al. Human-Robot Collaborative Assembly by On-line Human Action Recognition Based on an FSM Task Mode, *Proc. HRI-2013 Workshop on collaborative Manipulation: New challenges for robotics and HRI, Mar. 3 2013.*

[5] Michelangelo Fiore, Aurelie Clodic, Rachid Alami.On Planning and Task achievement Modalities for Human-Robot Collaboration. *International Symposium on Experimental Robotics (ISER 2014), Jun 2014, Marrakech, Morocco. 15p. ¡hal-01149109¿ 13 November 2015.*

[6] Chien-Liang Fok et al. Integration and Usage of a ROS-Based Whole Body Control Software Framework *Robot Operating System (ROS)Volume 625 of the series Studies in Computational Intelligence pp 535-563 10 February 2016.*

[7] Liang S Ng et al. Open source hardware and software platform for robotics and artificial intelligence applications *IOP Conf. Series: Materials Science and Engineering 114 (2016) 012142 doi:10.1088/1757-899X/114/1/012142.*

[8] Pedro Deusdado et al. An Aerial-Ground Robotic Team for Systematic Soil and Biota Sampling in Estuarine Mudflats *Robot 2015: Second Iberian Robotics Conference Volume 418 of the series Advances in Intelligent Systems and Computing pp 15-26.*