

Developing Safe and Efficient Robotic Assistants for Close-Proximity Human-Robot Collaboration

by

Przemyslaw A. Lasota

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

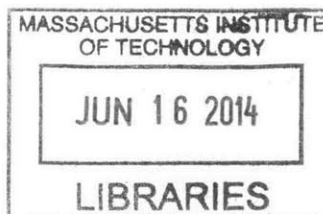
Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2014

ARCHIVES



© Massachusetts Institute of Technology 2014. All rights reserved.

Signature redacted

Author

Department of Aeronautics and Astronautics

May 22, 2014

Signature redacted

Certified by

Julie A. Shah
Assistant Professor of Aeronautics and Astronautics
Thesis Supervisor

Signature redacted

Accepted by

Paulo C. Lozano
Associate Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

Developing Safe and Efficient Robotic Assistants for Close-Proximity Human-Robot Collaboration

by

Przemyslaw A. Lasota

Submitted to the Department of Aeronautics and Astronautics
on May 22, 2014, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

As the field of robotics continues to advance and the capabilities of robots continue to expand, there is a strong incentive to introduce robots into traditionally human-only domains. By harnessing the complementary strengths of humans and robots, the human-robot team has the potential to achieve more together than neither could alone. To allow for the paradigm shift from isolation to collaboration, however, technologies in support of close-proximity human-robot interaction must be developed.

The first contribution of this thesis is the development and evaluation of a real-time safety system designed specifically for close-proximity human-robot interaction. The developed safety system, which uses a continuously updated virtual representation of the workspace for accurate human-robot separation distance calculation, is shown to allow for safe human-robot collaboration at very small distances of separation and with a very low latency. Furthermore, it is shown that this soft real-time system does not require hardware modification, which makes it easy and inexpensive to deploy on current industrial robots.

To understand how humans respond to adaptive robotic assistants, and whether the response leads to efficient and satisfying interaction, a robot control architecture capable of Human-Aware Motion Planning, a type of motion-level adaptation, is implemented. This architecture is then used in a human-subject experiment in which participants perform a collaborative task with the robot in two distinct motion planning modes: human-aware and standard. The fluency of the team in both modes is then compared with the use of quantitative metrics like task execution time, amount of concurrent motion, human idle time, robot idle time, and human-robot separation distance, as well as a subjective evaluation of the robot based on questionnaire responses. It is shown that Human-Aware Motion Planning leads to improvements across all quantitative metrics and to a more satisfied human co-worker.

Thesis Supervisor: Julie A. Shah

Title: Assistant Professor of Aeronautics and Astronautics

Acknowledgments

Personal Acknowledgments

I would like to extend my deepest gratitude to everyone who made the last two years one of the most intellectually enriching, exciting, and interesting periods of my life. First, I would like to thank my advisor, Dr. Julie Shah, for giving me this great opportunity by accepting me into her fantastic research group and providing me with terrific guidance and support. Her invaluable insight and vast technical knowledge was immensely helpful in everything from formulating the topic of my research to tiny details of data analysis. Her cheerful and enthusiastic attitude allowed me to stay motivated and to follow through with my goals. I am also thankful for the opportunities she has given me which allowed me to grow as a researcher. From going across the Charles to present my first conference paper, to spending a week and a half abroad analyzing the final assembly of aircraft in a factory in Brazil, I am truly grateful for each and every opportunity and experience being part of her group has given me. As a young, talented researcher, Julie also served as a source of inspiration and a role model. She is truly the best advisor one could ask for, and I look forward to working under her guidance in the following years.

I would also like to thank and acknowledge my fellow labmates in the Interactive Robotics Group. The cheerful lab atmosphere and diversity of knowledge and skills of the group has helped me immensely over the course of the last two years. From hanging out during IRG game night to in depth discussions about research, I truly enjoyed each experience we shared, and I appreciate being part of such a great group.

I would not be where I am today without the unconditional love and support of my family. I would like to thank my parents, Witold and Monika, for their unending support throughout the years and for providing me with everything I needed to pursue my dreams. I want to thank my older brother, Sebastian, for “paving the way” and inspiring me to strive toward achieving my goals. I would also like to thank my little sister, Paula, for her cheerfulness and support when I needed it the most.

I would also like to thank Christina, whose love and support over the last 6 years

allowed me to persevere through the difficult moments and enjoy and celebrate the happy ones. She believed in me and supported me in the pursuit of my goals and aspirations, even when I showed doubt. I am truly lucky to have her in my life.

Funding

This thesis was supported in part by ABB Corporate Research, NSF Award #6928599, and the Skolkovo Foundation.

Contents

1	Introduction	15
1.1	Motivation	15
1.2	Background	16
1.3	Real-Time Safety System for Human-Robot Interaction	19
1.4	Human-Aware Robot Control Architecture	20
1.5	Analysis of Effects of Human-Aware Motion Planning	21
2	Implementation and Evaluation of a Real-Time Safety System for Human-Robot Interaction	23
2.1	Introduction	23
2.1.1	Classifying Safety	23
2.1.2	Safety Standards for Collaborative Robots	24
2.2	Background	25
2.3	Required Functionality	27
2.4	Implementation	29
2.4.1	Hardware	29
2.4.2	Software	29
2.4.3	Implementation Discussion	31
2.4.4	Latency Improvement	34
2.5	System Latency Evaluation	35
2.6	Conclusions and Future Work	36

3	Design of a Human-Aware Robot Control Architecture	39
3.1	Introduction	39
3.1.1	Human-Aware Motion Planning	39
3.1.2	Required Functionality	41
3.2	System Architecture	42
3.2.1	Safety System for Human-Robot Interaction	42
3.2.2	Human Action Tracking	42
3.2.3	Action Override	44
3.2.4	Robot Control	45
3.2.5	Generation of Human-Aware Robot Trajectories	47
3.3	Improving Robustness and Latency of the Robot Control System . . .	50
3.3.1	Trajectory Downsampling	50
3.3.2	System Latency Improvements	51
3.3.3	Final Robot Control Algorithm	54
3.4	Conclusions and Future Work	55
4	Analyzing the Effects of Human-Aware Motion Planning on Close- Proximity Human-Robot Collaboration	57
4.1	Introduction	57
4.2	Method	58
4.2.1	Task	59
4.2.2	Participants	60
4.2.3	Procedure	60
4.2.4	Dependent Measures	62
4.2.5	Hypotheses	64
4.3	Results	64
4.3.1	Quantitative Team Fluency Metrics	64
4.3.2	Subjective Evaluation	66
4.4	Discussion	67
4.4.1	Differences in Team Fluency	67

4.4.2	Differences in Perceived Safety and Comfort	68
4.4.3	Human and Robot Idle Time	69
4.5	Conclusion	70
5	Conclusion and Future Work	73
5.1	Conclusion	73
5.1.1	Real-Time Safety System for Human-Robot Interaction	73
5.1.2	Human-Aware Robot Control Architecture	74
5.1.3	Analysis of Effects of Human-Aware Motion Planning	75
5.2	Future Work	76

List of Figures

1-1	Total industrial robots sold (in millions of units) up to the end of given years in the range of 2000 to 2011. Data from [31]	16
1-2	Examples of tasks which are done mostly manually: final assembly of airplanes (left) and automobiles (right) [Boeing Company, BMW] . .	17
2-1	Real workspace (top) and corresponding virtual representation (bottom)	31
2-2	Three possible modes of speed reduction as a function of separation distance	33
2-3	Histogram of system latencies. Note the very low number of high latencies on the right side of the graph.	36
2-4	Portion of histogram of system latencies shown in Fig. 2-3 near the average latency	37
3-1	Illustration of Human-Aware Motion Planning. The left panel depicts a shared workspace in which a human and robot are working on placing and sealing screws, respectively. The right panel shows a standard, shortest path motion (blue) and a human-aware motion (green) that the robot could take given the expected human workspace occupancy represented by the red cylinder.	40
3-2	An example of what a set of action volumes for a task might look like. This particular implementation utilizes spheres of equal diameter for each task, depicted in green.	43

3-3	Simplified pseudo-code describing the algorithm of the human-aware system for selecting and performing actions in a human-robot collaborative task	45
3-4	Pseudo-code describing the algorithm of the human-aware motion generation tool	48
3-5	Sample virtual environment generated by the human-aware trajectory maker tool. The left side of the figure depicts the expected human workspace occupancy as two green cylinders. The right side of the figure shows the extension of this region to the safety system’s deceleration region.	49
3-6	Illustration of the effects of robot task prioritization on the safety system task’s latency. Figure (a) depicts the latency of the safety system while the robot control task is running simultaneously before the fix was applied. Figure (b) shows what the latency looks like when the safety system is running alone. Figure (c) depicts what the latency looks like while both tasks are running after the implemented fix was applied.	53
3-7	Extended simplified pseudo-code describing the algorithm of the human-aware robot for selecting and performing actions in a human-robot collaborative task which implements fixes for robustness of trajectory execution and latency improvement	55
4-1	Photograph of the task setup used in the human-subject experiments.	59
4-2	Diagram depicting the experimental procedure. The group on the left represents the “human-aware first” condition, while the group on the right represents the “standard first” condition.	61

4-3 Mean values, with error bars indicating standard error of the mean (SEM), of (a) task execution time, (b) percent of concurrent motion, (c) average separation distance between the human and robot, (d) robot idle time, and (e) human idle time for the standard and human-aware robot executions 65

4-4 Mean values, with error bars indicating standard error of the mean (SEM), of (a) percent of concurrent motion, (b) robot idle time, and (c) average separation distance between the human and robot for groups of participants working with the standard and human-aware robots prior to exposure to the second robot type 66

Chapter 1

Introduction

1.1 Motivation

With the rapid progress of robotics related technologies in the recent years, robots have become increasingly prevalent in a wide variety of domains. In the years 2000-2011, the average increase in sales of industrial robots was 36.39%, with the total number of industrial robots sold nearly doubling from 1.15 million to 2.28 million in that decade. In the same year span, the average increase in sales of personal and private service robots was even more impressive at 74.15%, growing from 112,500 to 13.4 million in the same year span. This boom in the beginning of the 21st century is even more apparent in the fact that when considering the worldwide sales of robots up to the end of 2011, 93.31% of all robots sold since 1961 were sold since the year 2000 [31].

While this huge boom is mostly due to the tremendous increases in sales of service robots, the growth in deployment of industrial robots worldwide is also quite substantial, as can be seen in Fig. 1-1. In many domains where industrial robots are used today, however, robots are often deployed in complete isolation from people. In these domains, the flow of work is designed specifically such that robots interact with people as little as possible, with cages and barriers separating the two types of agents.

While physical separation of people and robots can be an effective strategy for

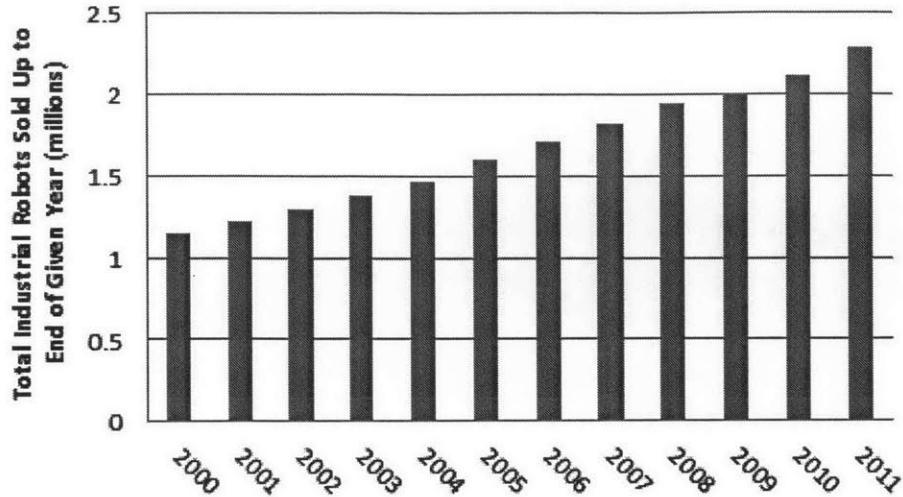


Figure 1-1: Total industrial robots sold (in millions of units) up to the end of given years in the range of 2000 to 2011. Data from [31]

some applications, a lack of human-robot integration prevents robots from being utilized in domains that stand to benefit from robotic assistance. The final assembly of aircraft and automobiles, for example, is still mostly a manual operation with minimal use of automation (SME Input, Boeing Company; SME Input, BMW), as can be seen in Fig. 1-2.

While the capabilities of robotic systems are continuously expanding, many tasks in such domains require a level of judgment, dexterity, and flexible decision making that surpasses current robots' abilities, leading to the tasks becoming human-dominated. While this is the case, there are also many non value-added tasks in these domains that could be performed by robotic assistants. Allowing robots to collaborate with people in shared workspaces and perform such tasks thus has the potential to increase productivity and efficiency, providing a strong incentive for the development of technologies that allow such collaboration.

1.2 Background

A significant amount of research has been done in the recent years in support of this goal across a variety of complementary domains. The first step in creating robots that

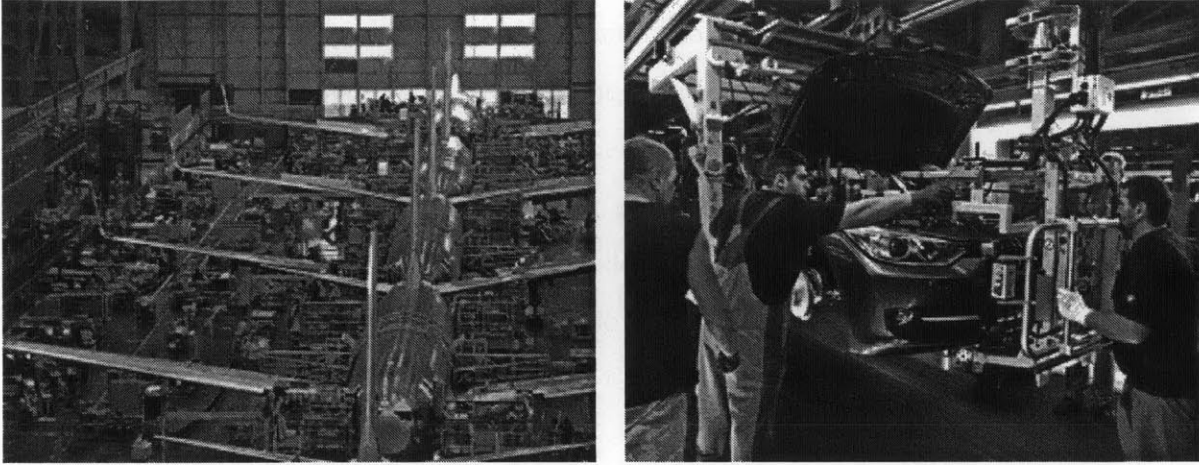


Figure 1-2: Examples of tasks which are done mostly manually: final assembly of airplanes (left) and automobiles (right) [Boeing Company, BMW]

can successfully collaborate with people is allowing the robots to navigate the shared workspace, requiring the development of specialized path planning algorithms and frameworks designed with the human element in mind. Sisbot et. al. developed one such framework, which utilizes parameters like human-robot separation distance, the human's field of vision, and his or her stance to generate safe and socially-acceptable robot paths [33]. An extension of this framework also reasons on task constraints and human kinematics [32]. To further enhance human-robot co-navigation, Chung and Huang investigated the development of a predictive path planning framework based on pedestrian motion models trained via statistical learning methods [14]. Bennewitz et. al. showed that a combination of clustering and Hidden Markov Models could be used to learn and predict human motion patterns, which allows for improved co-navigation [12].

Prediction can also be deployed on the task level, with a large variety of approaches being investigated. Prior work has indicated that observing changes to the entropy rate of a Markov Chain, produced from a task description encoded as a Markov Decision Process, could be utilized to encode the uncertainty of the robot about what action the human will perform next [29]. Lenz et. al. used a high-level architecture for joint human-robot action which could then be used to predict human tasks based on knowledge databases and decision processes [24]. Alami et. al. approached this

problem by encoding discrete sets of human and robot actions, which allowed for the incorporation of task-specific rules and preferences, which could then be utilized to predict likely sequences of human actions [10]. Kwon and Suh showed how Bayesian networks could be used for simultaneous inference on temporal and causal information, allowing to predict what task a robot should take and also when it should do it [23]. Hoffman and Breazeal, on the other hand, used a formulation based on a first-order Markov Process, and showed it could be successfully used for anticipatory action selection [20].

It has been shown that one can also make predictions on human actions without specific task models, but instead by observing motion features of the human worker. Mainprice and Berenson, for example, showed that early stages of human motion can be used to predict what action a human is taking in a reaching task, and that those predictions could be used to select tasks which avoid motion conflict [25]. In the domain of advanced driver-assistance systems, Doshi and Trivedi showed that head motion can be a useful factor in predicting human actions [16].

Conversely, allowing the human to be able to predict robot actions is also important. Legibility of robot trajectories, meaning their successful portrayal of the robot's intent, is often cited as of key importance for fluid human-robot interaction [33, 32, 18, 17]. Dragan et. al. has shown, however, that legible robot trajectories are often not the same as what the person would predict, making legibility and predictability contradictory properties of robot motion [18]. Indeed it has been shown that there needs to be a careful balance between these two properties, as the benefits of legibility suffered when robots moved beyond a "trust region" of expectation [17].

Being able to predict what action a robot is taking, however, is just one of many aspects one must take into account when considering the effects robotic assistants have on humans and the implications of these effects on the quality of the interaction. Arai et. al. showed that several parameters, including separation distance, end effector speed, and advance notice of robot motion, have a significant effect on the mental strain of human operators as measured by skin potential response [11]. It can be seen from the work by Meisner et. al., in an experiment evaluating a robot controller

designed for human-friendly trajectories, that preventing collision between humans and robots in a co-navigation task is not sufficient to maintain human comfort [27]. The effects of robotic assistants on people can also be seen in the work by Unhelkar. et. al., who showed that there are significant differences in team dynamics in human-robot and human-human teams as measured by interaction and idle times as well as subjective evaluations of team fluency, situational awareness, comfort, and safety [35]. In terms of human response to adaptive robots, Hoffman and Breazeal showed that human-robot teams in which participants worked with robots that anticipate their actions performed a task more efficiently and had different perceptions of the robot's contribution to the team's success [20].

While a significant amount of work has been done in a variety of domains in support of human-robot collaboration at close distances of separation, the evaluation of each work was limited to the particular components being developed (e.g. motion planning, action prediction, etc.). This focused evaluation led to a lack of analysis of an end-to-end system that incorporates the various components into a comprehensive system. Additionally, some of the research was evaluated solely in simulation without analyzing how real human co-workers respond to the proposed improvements. The focus of this work, therefore, was the implementation and evaluation, through human-subject experiments, of an adaptive robotic assistant that incorporated a functional real-time safety system along with motion-level adaptation.

1.3 Real-Time Safety System for Human-Robot Interaction

When dealing with close-proximity human-robot interaction, the primary concern that must always be addressed first is safety. While a variety of research in complimentary topics has been performed in support of this goal, no prior safety system architecture was found to be capable of supporting continuous human-robot interaction at low distances of separation in a robust manner. Additionally, research in

safety systems often focuses on developing specialized actuators or even entire new robotic platforms, which does not provide a solution for enabling safe human-robot interaction with currently deployed robots.

Consequently, the first contribution of this work, outlined in Chapter 2, focused on the implementation of a real-time safety system which would allow for close-proximity interaction with currently deployed industrial robots. In the first part of the chapter, two distinct categories of safety are defined, physical and psychological, both of which must be accounted for in a successful safety system implementation. Next, the current state of formal safety standards for human-robot collaboration is discussed, along with an examination of past research in human-robot safety systems. The implementation details, both on the hardware and software sides, are then discussed.

The next sections of the chapter then focus on describing why the chosen implementation scheme, namely the use of a virtual environment constructed according to the position of the human and robot in the workspace, is ideal for safe close-proximity interaction. It is shown that psychological safety can be accounted for by adjusting the tuning parameters of the robot's speed adjustment function, which provides flexibility in defining the manner in which the robot reduces speed. Next, it is shown that physical safety can be maintained as well, even at separation distances as low as 6 cm.

In the final sections of the chapter, work done in improving the latency of the safety system is described, followed by a latency evaluation. It is shown that the implemented safety system is soft real-time, with an average latency of 6.13 ms, and with latencies expected to be below 9.64 ms with 95% probability, below 11.10 ms with 99% probability, and below 14.08 ms with 99.99% probability.

1.4 Human-Aware Robot Control Architecture

While guaranteeing safety in human-robot interaction is necessary, ensuring safety alone does not necessarily lead to efficient human-robot collaboration. Consequently, a robot architecture which monitors the progress of a task and adjusts robot mo-

tions to avoid motion conflicts was developed, as described in Chapter 3. First, Human-Aware Motion Planning, an adaptive motion planning technique which utilizes prediction of human actions and workspace occupancy to plan robot motions which avoid motion conflict, is discussed. Next, the various sub-components of the system architecture are described, including the safety system, human action tracking, action override, and robot control. The approach used for the generation of human-aware robot trajectories is then discussed in detail. Finally, the efforts in improving the robustness and latency of the architecture are described.

1.5 Analysis of Effects of Human-Aware Motion Planning

Once an architecture capable of supporting Human-Aware Motion Planning was developed, the next step of this work was to analyze the effects of this motion planning technique on human-robot collaboration at low distances of separation, which is the focus of Chapter 4. As mentioned in the beginning of this chapter, evaluation of technologies developed for human-robot collaboration is often carried out in simulation and only for specific sub-components instead of for a full end-to-end system. As a result, the developed robot architecture was used in a human-subject experiment in which participants collaborated with an industrial robot to complete a task, and the response of the human participants and effectiveness of the human-robot team was evaluated.

The dependent measures considered in the experiment included the quantitative team fluency metrics of task execution time, amount of concurrent motion, human idle time, robot idle time, and human-robot separation distances, as well as subjective evaluation of participants' satisfaction with the robot as a teammate and their perceived safety and comfort. The results indicate that Human-Aware Motion Planning leads to improvements in all of these categories when compared to a baseline, shortest-path motion planning method. When working with a human-aware robot,

participants completed the task 5.57% faster ($p = 0.038$), with 19.9% more concurrent motion ($p < 0.001$), 2.96% less human idle time ($p = 0.019$), 17.3% less robot idle time ($p < 0.001$), and a 15.1% larger separation distance ($p < 0.001$). In terms of subjective evaluation, when describing the human-aware robot, participants agreed more strongly with “I trusted the robot to do the right thing at the right time” ($p < 0.001$), “The robot and I worked well together” ($p < 0.001$), “I felt safe when working with the robot” ($p < 0.001$), and “I trusted the robot would not harm me” ($p = 0.008$) and disagreed more strongly with “The robot did not understand how I wanted to do the task” ($p = 0.046$), “The robot kept getting in my way” ($p < 0.001$), and “The robot came too close to me for my comfort” ($p < 0.001$). The implications of these results and a discussion of the statistical analysis is also described in Chapter 4.

Chapter 2

Implementation and Evaluation of a Real-Time Safety System for Human-Robot Interaction

2.1 Introduction

In this chapter, we describe the development and evaluation a low-latency, real-time safety system capable of turning a standard industrial robot into a human-safe platform. The system ensures the physical safety and comfort of the user without the need for specialized actuators or any other modification of the robot's hardware, and is capable of supporting precise stopping thresholds that allow for human-robot interaction (HRI) at low distances of separation.

2.1.1 Classifying Safety

As mentioned in the previous chapter, enabling humans and robots to work together in close proximity to each other would not only allow for more efficient human-robot collaboration in fields where humans and robots already coexist, but also for the introduction of robots into many previously human-only domains. However, safety will always be the primary concern in any application of HRI, and as various HRI

technologies are researched, it is of the highest importance that methods guaranteeing human safety during human-robot interaction are developed in parallel.

One can classify safety into two categories: The first, and most obvious, is physical safety. To maintain physical safety, all unwanted human-robot contact must be prevented, and if contact is required by the task at hand or is inevitable for another reason, the forces exerted by the robot on the human must fall below limits that could cause discomfort or injury.

The second – and often overlooked – category is psychological safety. In the context of human-robot interaction, this means ensuring that human-robot interaction does not cause excessive stress and discomfort for extended periods of time. Take, for example, a hypothetical robotic system capable of moving a sharp end effector at very high speeds within centimeters of a human operator’s arm. While the system might be able to prevent unwanted injury via contact, a human working with such a system is likely to be in a state of constant stress and discomfort, which can have very negative long-term health effects [26].

2.1.2 Safety Standards for Collaborative Robots

In light of these categories of safety, it is therefore critical that methods ensuring both physical and psychological safety are developed and designed to meet international standards. Close-proximity interaction between humans and robots is still a fairly new and developing interaction paradigm, and, as such, formal definitions of safety within this context are still under development. Toward the goal of establishing these definitions, the International Organization for Standardization (ISO) developed the ISO 10218 international standard, entitled “Robots and robotic devices – Safety requirements for industrial robots,” which was most recently updated in 2011 [1]. A technical specification (ISO TS 15066), entitled “Robots and robotic devices – Safety requirements for industrial robots – Collaborative operation,” which provides information and guidance on how to achieve the safety standards described in ISO 10218 specifically for collaborative robots, is still under development [5].

While specific safety standards are not yet fully defined, the National Institute of

Standards and Technology (NIST) has provided guidance on the two main areas of focus of the ISO TS 15066: speed and separation monitoring and power and force limiting. In terms of the former, the guidance intends to enable collaborative robots to track people within a workspace and adjust speed according to the distance of separation between the human and robot. In the second area of focus, the aim is to enable robots to moderate applied forces to ensure that they remain below biomechanical limits. Additionally, the project is intended to develop performance measures to test how well a robot conforms to the required standards [8]. This guidance provided by NIST allowed us to pursue the development of an early implementation of the forthcoming standards.

2.2 Background

The task of maintaining safety during human-robot collaboration is multidisciplinary in nature, and thus has been approached in a variety of ways. In terms of the psychological aspect of safety, it has been shown that providing physical safety through collision avoidance is not sufficient to maintain human comfort [27]. Furthermore, it has been shown that several parameters, including separation distance, end effector speed, and advance notice of robot motion, have a significant effect on the mental strain of human operators, even if there is no contact between the human and robot workers. The same research also indicated that having grown accustomed to working with robots does not necessarily diminish these effects [11].

These results illuminate the importance of not only maintaining physical safety, but also ensuring that robot motions are comfortable for the humans interacting with the robot. Prior work that has taken this point into consideration evaluated parameters including the human's field of vision, posture, and kinematics when planning safe and comfortable robot paths [33, 32].

With regard to the physical aspect of safety, work has been done toward minimizing the negative effects of human-robot collision, as well as preventing collision from occurring altogether. Work on collision reaction control strategies has shown

that switching to torque control with gravity compensation upon impact can greatly reduce the force exerted on the human in the event of a collision [19]. New types of actuators with variable impedance have been developed, and show great potential in allowing for intrinsically safer robots by reducing joint stiffness when the robot is moving quickly [34].

In the realm of collision prevention, innovation in 3D sensor fusion and the use of dynamic safety zones appears to be a promising method [30]. The ability to predict human actions also has the potential to prevent collision. On the human motion level, recent work has shown that human actions can be predicted from early stages of movement [25]. On the task level, prior work has indicated that observing changes to the entropy rate of a Markov Chain, produced from a task description encoded as a Markov Decision Process, could be utilized to encode the uncertainty of the robot about what action the human will perform next [29]. In other work, the encoding of discrete sets of human and robot actions allowed for the incorporation of task-specific rules and preferences, which could then be utilized to predict likely sequences of human actions [10].

Research and innovation in these various fields has led to the development of new, inherently human-safe robots, such as the RethinkRobotics Baxter, which features force sensors at each joint and Series Elastic Actuators that minimize the force of impact [4]; or ABB's Dual Arm Concept Robot, which has built-in power and speed limitation, as well as software-based collision detection [2]. Besides the creation of brand-new robot designs, work in the field of robot safety has also led to the development of add-on technologies, such as ABB's SafeMove, which provides programmable, complex safe zones by monitoring robot speed and position [22].

While many of these works have yielded very promising results for maintaining human safety in HRI, a great majority of them focus on technologies that can be applied to new robot designs, rather than to existing robotic platforms. While utilizing new robot designs like the Baxter and ABB's Dual Arm Concept Robot and developing more human-safe robots with the technologies mentioned above can ensure human safety in HRI, purchasing new robots or retrofitting existing robots with new

hardware components can be cost-prohibitive or physically impossible. With an estimated 1.2 to 1.5 million industrial robots already in use worldwide [3], there is great incentive to design a solution that can turn these robots into human-safe platforms without the need for hardware modification.

In the work mentioned above that does not explicitly require new actuators or arrays of internal robot sensors, safety systems are often designed such that the robot completely avoids a large region where the human is located, or uses approximations of human and robot locations that are too coarse or uncertain to allow for the robot and human to interact in close proximity to one another. As a great deal of industrial work is still performed by humans – even in fields where robots have been successfully integrated, such as the automotive and aerospace industries – many industrial applications stand to benefit from the introduction of robotic assistants that aid human workers. However, this assistance will require close-proximity HRI, which makes the development of a safety system capable of operating effectively at small separation distances attractive.

The goal of the work presented in this chapter was, therefore, to build upon prior work in the field in order to overcome the abovementioned drawbacks and create a robot safety system capable of turning current, standard industrial robots into human-safe platforms for close-proximity HRI, without the need for robot hardware modification.

2.3 Required Functionality

In the context of this work, the functionality required by the safety system was guided by its main application: to enable the use of a 6 DOF industrial robot (an ABB IRB-120, see Section 2.4.1 for details) as a collaborative assistant for human subject experiments in the Interactive Robotics Group. In order to achieve this goal, the system was designed to ensure safe interaction by preventing unwanted contact between the robot and experiment participants by following the ISO TS 15066 guidance given by NIST described in Section 2.1.2, specifically the adjustment

of speed based on separation distance between the human and robot.

In terms of the second main area of focus of the ISO TS 15066, power and force limiting, due to a lack of force sensors on the ABB IRB-120, it was decided that the speed limiting would be set such that the robot stops completely before coming in contact with the person. This effectively eliminated the need for force regulation, as the robot would never come in contact with the person and thus would not be exerting any force at all. This decision limited the types of tasks the robot could collaboratively perform with a human to only those which do not require contact between the human and robot, but this compromise was necessary due to a lack of force sensors on the robot.

From a technical standpoint, the safety system has two important requirements. First, the system has to be designed such that it can run as a separate process, independent of any other programs that might be executed on the robot. This allows for the safety system to be deployed on the robot for virtually any experiment involving the ABB IRB-120 robot without interfering with, and conversely getting interference from, robot control software developed for a particular experiment or task. This was achieved through a multithreaded software design described in Section 2.4.2.

The second important technical requirement was that a low latency needed to be achieved for the safety system to work effectively and robustly. Several challenges were met in ensuring low system latency, and the various improvements which addressed these challenges are described in Section 2.4.4 and the analysis of the latency performance of the system is described in Section 2.5. Additional latency challenges caused by the integration of the robot control software used in our experiments and the solutions are also described in the next chapter, in Section 3.3.2.

2.4 Implementation

2.4.1 Hardware

The robot used in the implementation and evaluation of the safety system described in this work is the ABB IRB-120. This is a standard industrial robot with no built-in safety systems for HRI and noncompliant joints, capable of moving at speeds as high as 6.2 m/s [21]. Without an additional safety layer, this type of robot is required to operate within a safety cage – which, when opened, stops the robot immediately. Consequently, it is not capable of safe HRI in its stock form.

A PhaseSpace motion capture system was utilized to sense the position of the human worker within the workspace. This type of active motion capture system provides accurate human localization that is robust to temporary occlusions. While this type of system might not be a viable tracking solution in factory environments, the developed safety system can be utilized with any sensing system capable of providing accurate localization data. As advancements are made in the field of computer vision and new 3D sensing hardware is introduced, the motion capture system may be replaced by a less-intrusive option, if necessary.

The computer platform used to run the safety system software was a standard Windows 7 machine with a Core i7-3610QM 2.3GHz processor.

2.4.2 Software

The software implementation of the safety system consists of several subsystems that exchange information in a coordinated, low-latency fashion: the core program, the motion capture software, the robot software running on the ABB IRB-120's controller, and a virtual workspace.

- **Core Program:** The core program serves as the main logic of the system. It connects to the other sub-components and relays information between them via TCP sockets. The core program also performs some geometrical calculations used by other sub-components and logs data for system analytics.

- **Motion Capture Software:** This sub-component is responsible for capturing the most recent position of the human. The software takes in raw data from the motion capture system, transforms it into the correct coordinate frame, and relays it to the core program.
- **Robot Software:** This portion of the system software resides on the robot's controller. It continuously monitors the robot's configuration and relays this information to the core program. The software is also responsible for adjusting the robot's speed according to separation distance data received from the core program. The code runs as a secondary task on the robot's controller, completely independent of the primary task used to command the robot's motions. Consequently, the safety system can run in the background of any task given to the robot, making it easy to use for virtually any task the robot is programmed to perform with a human co-worker.
- **Virtual Workspace:** This sub-component is responsible for constructing a virtual representation of the workspace shared by the human and robot, based on information received from the motion capture system and robot controller, using OpenRAVE, a robot simulation environment [15]. The current robot position is translated into the virtual workspace based on the known position of the robot's base, its 3D CAD model, and the joint angles received from the robot's controller. In the particular workspace in which the robot and safety system were tested, the only portion of a human worker the robot was able to reach was the right arm and hand. Consequently, the position of the human in the virtual workspace was approximated by two concentric cylinders: one cylinder for the forearm and one larger-diameter cylinder for the hand. The length of the forearm cylinder was adjusted to the particular user's arm length based on information received from the motion capture system. The diameters of the two cylinders were such that the virtual cylinders completely enclosed the user's arm and hand. A sample workspace configuration and corresponding virtual representation are depicted in Fig. 2-1. Once the configuration of the

human and robot is updated in the virtual environment, the separation distance between them is accurately calculated and relayed to the core program, which in turn relays this information to the robot's controller for speed adjustment.

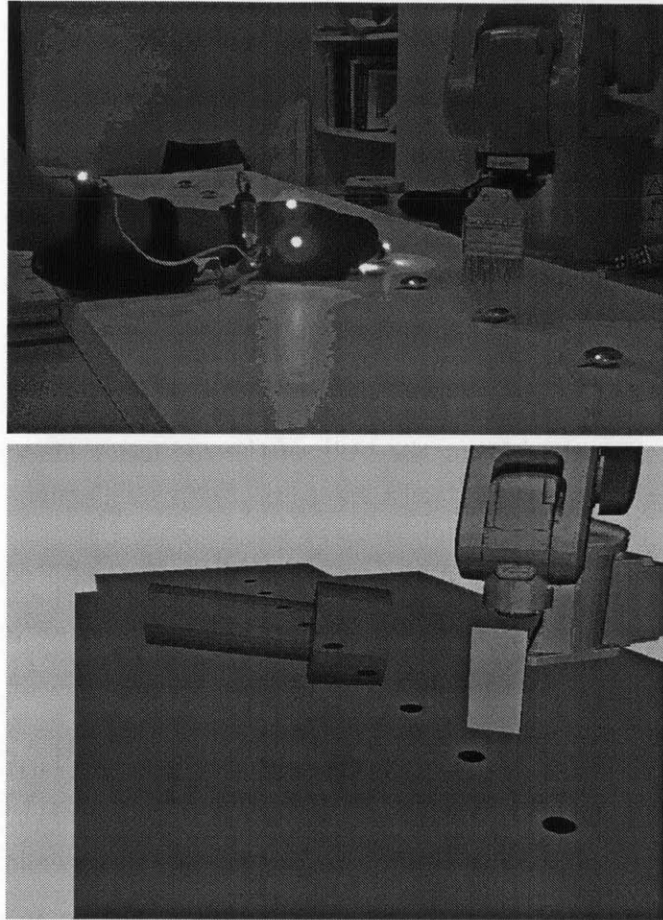


Figure 2-1: Real workspace (top) and corresponding virtual representation (bottom)

2.4.3 Implementation Discussion

The decision to use this type of implementation scheme – a virtual environment constructed according to the position of the human and robot in the workspace – as opposed to, for example, a purely vision-based approach, was made in order to fully leverage the known robot configuration, rather than approximate it via another method. Since the position of the human is also accurately known, the virtual environment implementation scheme allows for very accurate separation distance mea-

surements in real-time, resulting in precise robot speed control. The robot's speed was adjusted according to a function of the form:

$$\alpha(d) = \begin{cases} 1 - \beta(d-d_{stop})^\gamma & \{d \mid d_{stop} \leq d \leq d_{slow}\} \\ 0 & \{d \mid d > d_{slow}\} \\ 1 & \{d \mid d < d_{stop}\} \end{cases}$$

In the above form, α represents the percentage reduction in the robot's speed expressed as a decimal, d is the current separation distance between the human and robot, d_{stop} is the distance at which the robot should be stopped, d_{slow} is the distance at which the robot's deceleration begins, and β and γ are tuning parameters that define the behavior of the speed reduction function; for example, how quickly the speed should drop off and whether the bulk of the reduction should occur near d_{stop} or d_{slow} .

The ability to precisely control the speed of the robot as a continuous function of separation distance allows the safety system to be effective at very low separation distances. At moderate robot speeds and with the proper choice of parameters in the speed reduction function, the safety system is effective with the tested hardware at d_{stop} values as low as 6 cm. This means that the human and robot can safely perform tasks in very close proximity to one another, which is not possible with other safety systems that incorporate coarse, discretized workspace occupancy approximations or discrete safety zones.

An additional benefit of precise robot speed control based on separation distance is that the system can be tuned such that the deceleration of the robot occurs at a rate comfortable for the human. By properly tuning the parameters, we can have the robot ease to a stop gently and smoothly, as opposed to stopping abruptly or using coarse "slow" and "stop" zones that cause sudden changes in robot speed. Fig. 2-2 depicts three possible modes of speed reduction based on the tuning of β and γ in the speed reduction equation, with $d_{slow}=15$ cm and $d_{stop}=6$ cm. The green dashed line in the figure represents the strategy of reducing speed slowly when the slow-down threshold is passed, then quickly reducing speed to zero when the stop

threshold approaches. The red dotted line represents the opposite strategy: rapidly decreasing speed once the reduction threshold is passed, then gradually easing to a stop. The blue solid line represents a balanced approach between these two modes.

Which mode is appropriate is left for the end user to determine, as their choice might depend on the robot's speed, the tool it is holding, or other task-dependent parameters. The freedom to finely tune the deceleration behavior of the robot allows the user to adjust it so that the interaction is comfortable and stress-free, even at small distances of separation.

Another key benefit of this type of implementation scheme is that it does not require any robot hardware modification. While other systems require special actuators or the retrofitting of robots with force and torque sensors, this implementation can be utilized with standard, unmodified industrial robots. This makes our safety system easy and cheap to implement for organizations that already own industrial robots, as they are able to turn their previously dangerous industrial robots into human-safe platforms capable of close-proximity HRI at a very low cost.

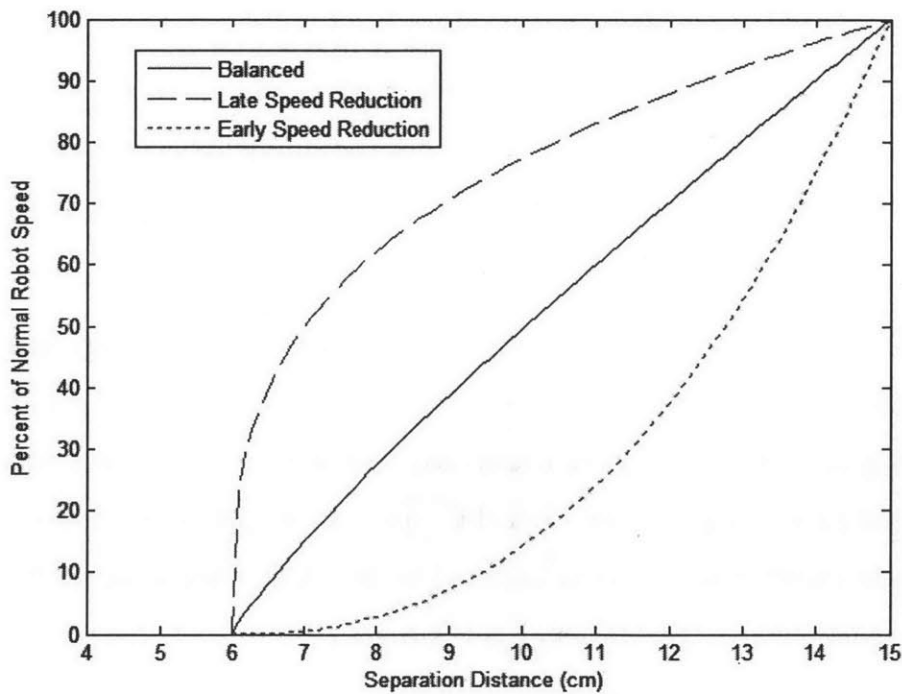


Figure 2-2: Three possible modes of speed reduction as a function of separation distance

2.4.4 Latency Improvement

A key requirement for the robustness of this type of safety system is low latency. This means that the amount of time necessary to perform a complete cycle of the safety system, from sensing to robot speed adjustment, must be very low. Consequently, several measures were taken to improve the latency of the described system.

First, to decrease the amount of time needed to perform minimum distance calculations, the CAD model of the robot was reduced in quality. An increase of 0.75 mm in the maximum deviation of the model reduced the number of polygons in the CAD model quite drastically (~50% reduction in file size). This led to significantly faster separation distance calculation at a very low penalty to model accuracy.

The second key improvement in latency resulted from network optimization. The various subsystems described in 2.4.2 reside on different physical machines, and so they must communicate with each other via an internal LAN. In order to ensure that all pertinent data is delivered successfully and in the correct sequence, Transmission Control Protocol (TCP) sockets were chosen as a mode of connection and transmission. While User Datagram Protocol (UDP) sockets can provide faster communication, this protocol does not guarantee successful delivery or correct sequence and is not supported by the RAPID programming language used to program ABB robots [7].

Due to the design of the system, very small packets of data are continuously sent over the TCP sockets, and each successive transmission must finish before the next one begins. This led to slow transmission speeds due to Nagle's Algorithm, a network optimization algorithm designed to prevent network congestion by chunking small packets together and sending them all at once [28]. Instead of sending a small packet immediately upon generation, Nagle's Algorithm instructs the program to wait for more data to send, reducing bandwidth at a cost to latency. As the continuous transmission of small packets is required by the safety system, Nagle's Algorithm was disabled for all socket connections, leading to significant improvements in latency.

2.5 System Latency Evaluation

While our safety system is capable of robot speed adjustment in real-time, there are no hard guarantees for latency (i.e., it is not "hard real-time"). Consequently, it is desirable to evaluate the latency based on collected performance data to ensure that latency, on average, is at a sufficiently low level to yield consistent performance. To allow for such evaluation, the safety system was utilized during an array of human-subject experiments involving the ABB IRB-120 robot.

Safety system latencies were recorded for a total of 174 experiment runs, resulting in approximately 1.8 million latency measurements over the course of 3 hours. The average latency was 6.13 ms, with a maximum latency of 389.6 ms. While the maximum latency was substantially higher than the average, large deviations from the average happened very rarely. Based on statistical analysis of the collected data, assuming the distribution is normal with a mean of 6.13 ms and a standard deviation of 2.14 ms, latencies are expected to be below 9.64 ms with 95% probability, below 11.10 ms with 99% probability, and below 14.08 ms with 99.99% probability.

In order to visualize these results, a histogram of latencies was constructed. Fig. 2-3 depicts the overall histogram of latencies, indicating a very large peak around the average latency and only sporadic high latency jumps. The source of these jumps is not known, but they may be caused by infrequent network connection drops. Fig. 2-4 shows a zoomed-in view of the histogram near the average latency.

While no hard guarantees on latency are made by the current implementation of the system, based on these results, we can say with high confidence that the average latency is very low and the sporadic jumps in latency do not occur often enough to significantly degrade system performance. In fact, with the robot used in this implementation, there is an inherent delay of 300-500 ms from the time the robot is ordered to reduce speed until it begins to execute the order [7], making the approximately 6 ms latency of the safety system an insignificant contribution to overall system latency. Since the frequency of high latencies is low and the sporadic jumps do not jeopardize the performance of the system, the system is classified as soft

real-time according to the definition of the IEEE Technical Committee on Real-Time Systems [9].

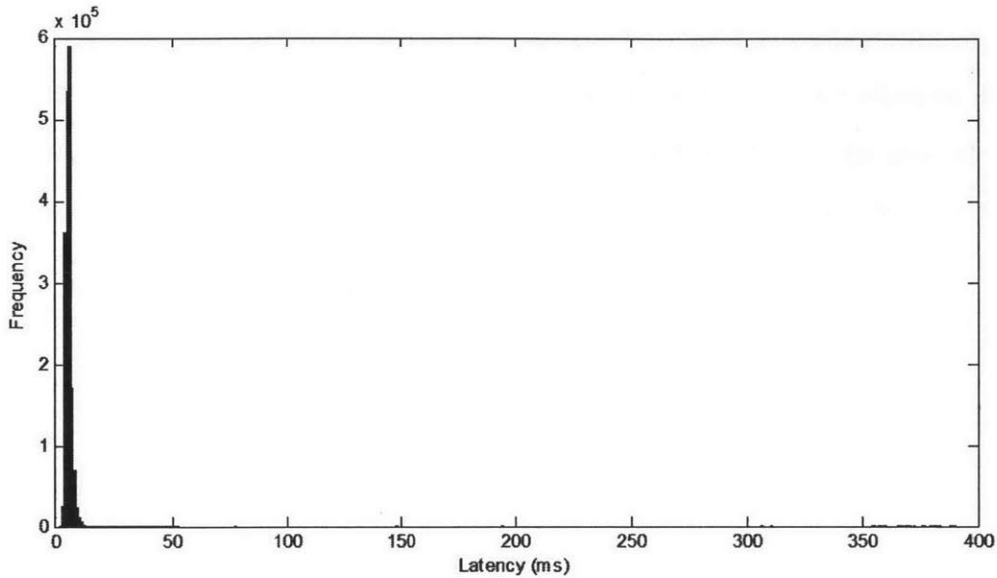


Figure 2-3: Histogram of system latencies. Note the very low number of high latencies on the right side of the graph.

2.6 Conclusions and Future Work

In this chapter, we described a novel implementation of a real-time safety system capable of turning a standard industrial robot into a human-safe platform. We showed that this implementation does not require any robot hardware modifications, such as special actuators or internal force and torque sensors, making our safety system inexpensive and easy to implement in domains where robots are already present. By leveraging known robot joint angles and utilizing accurate human localization, we showed that we can construct a virtual representation of the workspace that allows for the calculation of accurate separation distance data in real-time. We then described how this information can be used to precisely control robot speed, allowing for safe HRI at distances of separation as low as 6 cm, as well as for robot deceleration comfortable for the human worker. Finally, we demonstrated the benefit of deploying various latency improvement strategies, which resulted in system latencies falling

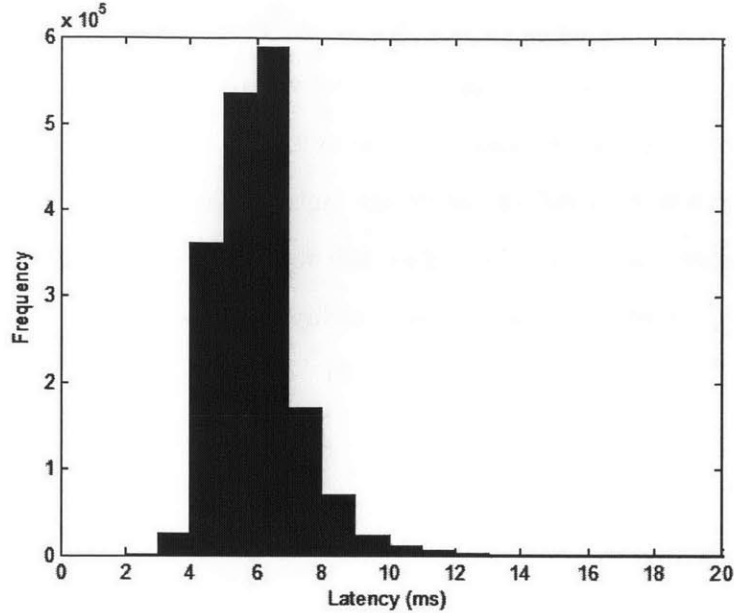


Figure 2-4: Portion of histogram of system latencies shown in Fig. 2-3 near the average latency

below 9.64 ms with 95% probability, below 11.10 ms with 99% probability, and below 14.08 ms with 99.99% probability.

While the latency is low compared to the inherent delay between speed adjustment commands and the execution of those commands by the robot used in this work, the lack of formal guarantees on system latency means that this is not a "hard real-time" system. In the future, we plan to incorporate a middleware solution, such as OROCOS RTT [6], to allow for hard real-time operation. The addition of middleware will also add a level of platform independence, making it easier to apply the system to a variety of robots and 3D sensors.

The inherent delay between speed adjustment commands and the execution of those commands by the robot utilized in this work can be expected with other industrial robots. As such, there is a substantial delay that cannot be removed through latency improvement within the safety system. We must overcome this limitation in order to improve system performance and allow for even closer HRI or higher robot speeds.

Consequently, another future direction of our work is to augment the safety system

through the prediction of future locations of the human and robot. If robot trajectories are known and we can accurately model where a human might move to in the next 300-500 ms using current motion or previously learned motion models, we can attempt to look a few hundred milliseconds "into the future" and adjust robot speed based on this information. Such an approach would help to overcome the limitation imposed by the inherent speed adjustment delays, but would be highly dependent on the accuracy of the future location prediction.

Chapter 3

Design of a Human-Aware Robot Control Architecture

3.1 Introduction

As described in the previous chapter, ensuring human safety during human-robot interaction is of the highest importance. While providing methods which guarantee physical and psychological safety is necessary, these methods alone do not allow for efficient human-robot collaboration. In tasks with significant potential for motion conflicts, simply slowing down and stopping the robot when the two agents approach each other can result in very inefficient teamwork. As a result, the next step of this work was to develop a robot control architecture which would monitor the progress of a task and adjust the robot's motions to actively avoid motion conflict. This chapter describes the requirements of this type of architecture, its development and structure, as well as some insights on improving the latency of the system based on the hardware used.

3.1.1 Human-Aware Motion Planning

In order to generate motions which avoid motion conflict, we implemented an adaptive motion planning technique that we call Human-Aware Motion Planning. In this

motion planning technique, the system attempts to predict the next human action, and then approximate what portion of the shared workspace the human worker will be using in the upcoming moments based on an appropriate motion model. The system then uses this prediction to modify the robot's motions to avoid this portion of the shared workspace in an attempt to eliminate motion conflicts.

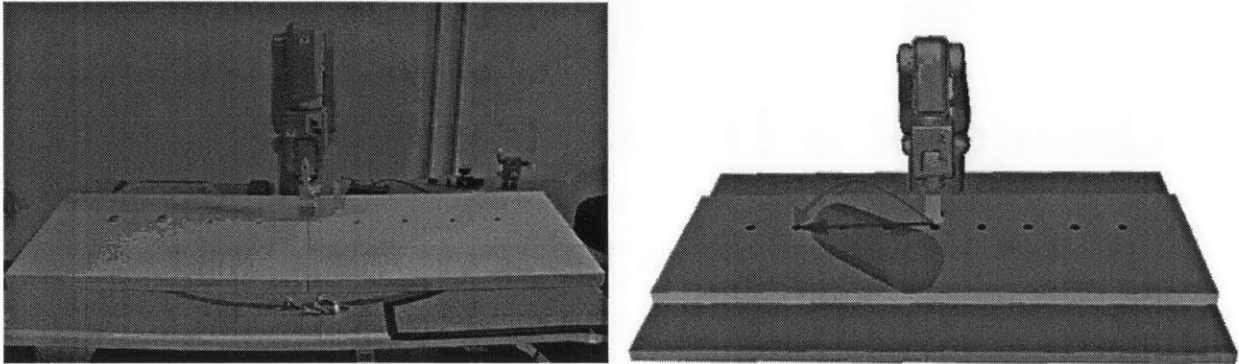


Figure 3-1: Illustration of Human-Aware Motion Planning. The left panel depicts a shared workspace in which a human and robot are working on placing and sealing screws, respectively. The right panel shows a standard, shortest path motion (blue) and a human-aware motion (green) that the robot could take given the expected human workspace occupancy represented by the red cylinder.

To illustrate this technique, consider the shared workspace depicted in Fig. 3-1. The left side of the figure shows a shared workspace in which a human and robot work together by placing screws and applying a sealant, respectively. If we can predict with high accuracy that the human worker will place a screw at the third hole from the left, beside the two screws already placed, we can then approximate what portion of the shared workspace will be used by the human worker in the next moments. A simple and effective motion model for this particular task is to approximate the predicted workspace occupancy of the human with a cylinder that encompasses the worker's arm as he or she completes the placing task, as this is the only part of the human the robot can reach. This cylinder is shown in the virtual representation of the workspace depicted in the right side of Fig. 3-1. Once the robot has a prediction of human workspace occupancy, it can adapt its motion planning by selecting a path to its own goal, in this case the second screw from the left, which avoids where it predicts the human will be. This human-aware motion is shown as the green line in the figure,

while the simple, shortest-path motion the robot would have taken otherwise is shown in blue.

Human-aware motion planning allows for the robot to actively avoid motion conflicts as it completes its own tasks. Since these types of motions are less direct and take more time to complete, it is important that they are used only when necessary. One can see that the human-aware motion planning method described above indeed fulfills this requirement. Using the example from before depicted in Fig 3-1, if the robot's action was still to move to the second screw from the left but the next predicted human action changed to be at the location all the way on the right, the predicted workspace occupancy would shift to that location, and there would be no potential for motion conflict. Consequently, the robot would select the more direct path depicted in blue. In this manner, the human-aware robot will select efficient, shortest-path motions when no predicted motion conflict exists, and only adapt its motions to be less direct if it is necessary.

3.1.2 Required Functionality

In order to provide the desired functionality of producing human-aware robot motions appropriately in real time, the system had several key requirements. First, the robot had to have the capability of tracking human actions to determine what actions have been taken thus far and what actions can be taken in the future. The robot then needed to be able to use this information to predict what action the human will take next and use an appropriate motion model to predict what portion of the shared workspace will be used by the human worker. With this knowledge, the robot then needed to be able to produce risk-aware motions as described in the previous section with an appropriate motion planning algorithm. Finally, in order for the system to function as desired, all of these sub-routines needed to be implemented in a software architecture that supported concurrency among the safety system, human action prediction, and robot action execution, requiring a multithreaded approach. Section 3.2 describes how these sub-components were implemented in detail.

3.2 System Architecture

As mentioned in the previous section, multiple sub-systems contribute to a functional implementation of a human-aware robotic assistant. This section describes these sub-systems in detail and explains how each was implemented.

3.2.1 Safety System for Human-Robot Interaction

The most important component of the system is a robust safety system that ensures the interaction between the human and robot agents is safe. This system must run in the background continuously without any interaction with other systems in order to ensure that it does not get interrupted by an unexpected failure of another sub-system. For this purpose, the safety system described in the previous chapter was utilized. For details on its implementation and evaluation, see Chapter 2. The implemented safety system is a soft real-time system that allows for close-proximity interaction that is both physically and psychologically safe.

3.2.2 Human Action Tracking

In order to aid the system in predicting what action the human worker might take next, it is useful to keep track of what actions have been taken by the human so far. In our implementation, the system continuously tracks the position of the human worker's hand with the use of a PhaseSpace motion capture system. While the same motion capture system is being used to track the position of the human in the safety system implementation, the two tracking programs are implemented independently of one another. The two programs are both clients to the PhaseSpace system's server, but due to their being independent processes, the failure of the action tracking process does not cause a failure of the safety system process, which is one of the key requirements mentioned previously.

In our implementation of the action tracking sub-system, we assume that actions are executed at specific locations in the shared workspace. Using this assumption, we then assign an action volume to each action which represent the volume of space in

which a human worker's hand will fall when he or she is taking that specific action. For simplicity of implementation, the current system utilizes spheres as action volumes. To allow for some flexibility and tuning, the diameter of these spheres is adjustable independently for each action. Figure 3-2 depicts what these action volumes might look like for an example task.

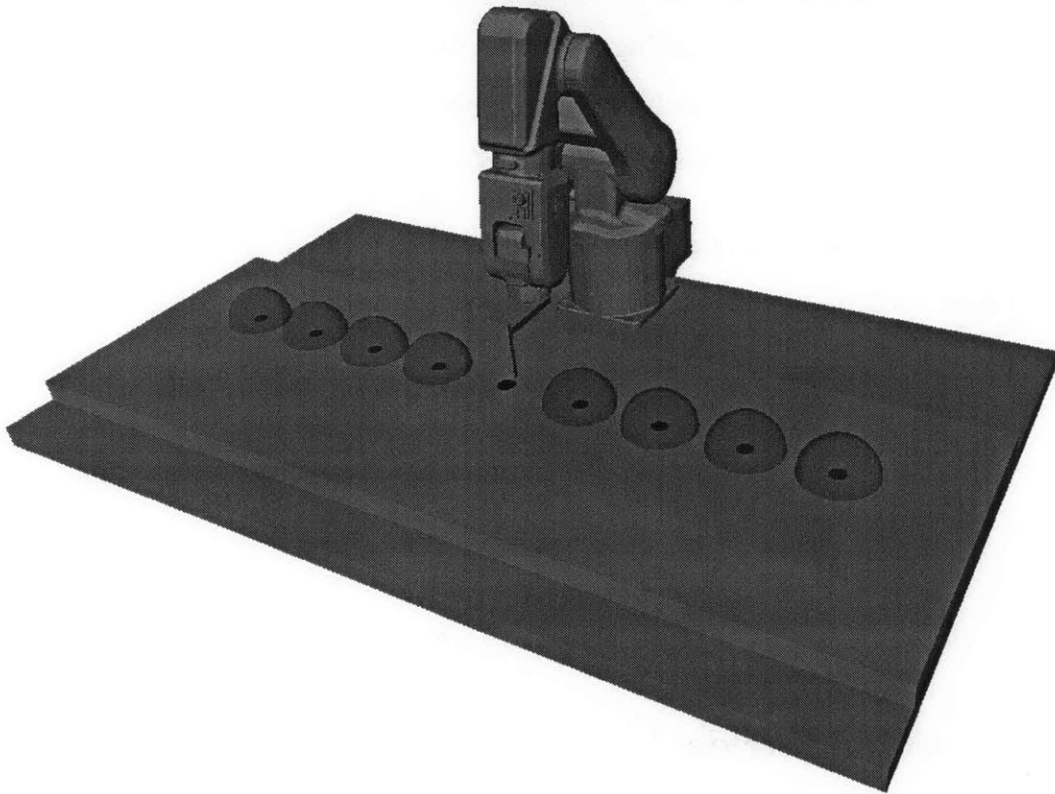


Figure 3-2: An example of what a set of action volumes for a task might look like. This particular implementation utilizes spheres of equal diameter for each task, depicted in green.

The manner in which the system recognizes if an action has been taken is based on penetration of these action volumes. If the user's hand transitions into an action volume, the system begins to track how long it has stayed within this volume. Once the user's hand leaves the action volume, if the time elapsed within the volume surpasses a specified threshold, the system marks that action as performed. If the time falls below the threshold, we assume the user did not perform the action and instead, for example, was simply moving his or her arm next to the task location. Just as with

the action volumes, the time thresholds are adjustable for each action independently. When marking the action as performed, the action detection sub-system accesses a shared data structure which defines which actions have been performed and which ones have not. This is a simple binary array of length equal to the number of actions in the task. For each action, the array takes on a 0 if the action has not been performed yet and a 1 if it has. As other sub-systems also access this data structure, as will be discussed in the following sections, we ensure that no two sub-systems attempt to access it simultaneously by making use of appropriate Java libraries and functionalities (e.g., the *synchronized* keyword).

While this implementation is very simple and works well for tasks such as the one shown in figure 3-2, it is not suitable for all types of tasks or situations. Due to the multithreaded nature of the implementation, however, the action tracking component can be swapped out with a more complex system, with additional sensors or more sophisticated algorithms, without requiring changes in other the sub-systems.

3.2.3 Action Override

While the action tracking sub-system described in the previous section works relatively well in practice, it is prone to some error if the user acts in an unexpected manner. For example, a user might move toward an incorrect task location and start performing the task only to stop and move to the correct location. In this instance, the incorrect task was not performed, but it will be marked as such by the system. In order to allow for correction of such errors during task execution, a separate action override sub-system runs as an independent thread within the system. In the current implementation, the action override is a simple text based interface that allows one to modify the data structure holding information about which tasks have been performed thus far, as described in the previous section.

```

robotControl( $\mathbf{A}$ ,  $\tau$ ,  $\Gamma$ )
1:  $\mathbf{A}_{rem} \leftarrow \mathbf{A}$ 
2: while  $\mathbf{A}_{rem} \neq \emptyset$  do
3:    $a_{robot} \leftarrow null$ 
4:    $a_{human} \leftarrow null$ 
5:   while  $a_{robot} = a_{human}$  do
6:      $a_{human} \leftarrow \text{getNextHumanAction}(\mathbf{A}, \pi_{human}, \tau)$ 
7:      $a_{robot} \leftarrow \text{getNextRobotAction}(\mathbf{A}_{rem}, \pi_{robot}, \tau)$ 
8:   end while
9:    $\gamma \leftarrow \text{getRobotTrajectory}(a_{human}, a_{robot}, \Gamma)$ 
10:  executeTrajectory( $\gamma$ )
11:   $\mathbf{A}_{rem} \leftarrow \mathbf{A}_{rem} \setminus \{a_{robot}\}$ 
12: end while

```

Figure 3-3: Simplified pseudo-code describing the algorithm of the human-aware system for selecting and performing actions in a human-robot collaborative task

3.2.4 Robot Control

The final component of the human-aware robot system is the component which controls what actions the robot performs, when they are performed, and which paths the robot takes when performing them. In our implementation, the robot control sub-system utilizes the algorithm shown in Fig 3-3 while the other sub-components simultaneously run in the background.

In the parameters of this algorithm, $\mathbf{A} = (a_1, a_2, \dots, a_n)$ represents the set of all possible joint actions the human and robot could perform in the given task. A joint action is one in which both agents must complete a portion of the action for the action as a whole to be completed. The second parameter, τ , is a binary vector of length $n = |\mathbf{A}|$ which specifies whether or not the human operator has completed each action (as described in Section 3.2.2). Finally, $\Gamma = (\gamma_{1,1}, \gamma_{1,2}, \dots, \gamma_{n,n})$ represents a set of robot trajectories for all possible combinations of human action a_i and robot action a_j .

Let us now go into more detail about the meaning and implementation of the steps defined by Algorithm 3-3. The first step (line 1) is to create a data structure which keeps track of the remaining robot actions, $\mathbf{A}_{rem} \subseteq \mathbf{A}$. While there are robot actions left to perform (line 2), we check what action the human and robot should perform

next. This is done by the `getNextHumanAction` and `getNextRobotAction` functions, respectively (lines 6 and 7). Both of these functions take as parameters sets of actions available, a decision policy π and, the information about human actions taken thus far stored in τ . The policies π_{human} and π_{robot} can be derived in a wide variety of ways, for example modeling the task as an MDP [29], PDDL [10], or another method. In the simplest case, the sequence of actions for the human and robot can be preset, in which case π_{human} and π_{robot} are to simply execute the next action in the sequence that has not been executed yet.

In our formulation, we assume that in a joint action the human agent must perform his portion of the action first. Consequently, once we select a_{human} and a_{robot} we first check if the two actions are the same (line 5). If they are, it indicates that neither agent has completed their portion of the next joint action, and so the robot has to wait until its partner finishes the first part of the task. Once the human worker completes his part of the joint action, τ will be updated, which will result in a different action being returned by `getNextHumanAction`. When this happens, a_{human} will no longer be the same as a_{robot} , terminating the inner loop and allowing the algorithm to progress to the next steps.

Once the actions for the two agents are calculated and verified to be different, we retrieve the appropriate human-aware trajectory $\gamma_{i,j} \in \Gamma$ using our `getRobotTrajectory` function (line 9). This function can be implemented to either generate an appropriate human-aware motion plan on-line or to utilize a pre-computed database of trajectories. In our implementation, we utilized the latter approach, which is described in more detail in Section 3.2.5. Once we generate a human-aware trajectory, we have the robot execute the action (line 10), after which we remove this action from the set of remaining robot actions, \mathbf{A}_{rem} (line 11).

This process of calculating next human and robot actions and executing robot actions at the appropriate time continues until the robot executes all actions, resulting in $\mathbf{A}_{rem} = \emptyset$.

3.2.5 Generation of Human-Aware Robot Trajectories

[Note: For definitions of variables used in this section, refer to the robot control algorithm and its description in Section 3.2.4]

One of the most important aspects of a human-aware robotic system is its ability to generate robot motion plans that actively evade expected human locations. Once we know what action the human worker will be taking and we select the action the robot should execute, we need to be able to create a motion plan which uses both of these pieces of information to create a human-aware robot trajectory. Towards this goal, we developed a human-aware motion generation tool with the use of the OpenRAVE robotic simulation environment (for more information on OpenRAVE, see [15]).

In this tool, we first define the shared workspace in order to construct its virtual representation, similarly to what was done in creating the safety system described in Chapter 2. Once we define all the objects in our workspace and the location, 3D CAD model, and kinematics of the robot, we define a set of robot transforms $\Omega = (\omega_1, \omega_2, \dots, \omega_n)$ for all actions $a \in \mathbf{A}$. These transforms contain information about both the position and orientation of the robot's end effector when performing each task. Similarly, based on an appropriate motion model, we define the positions of the human while performing each action as the set $\Psi = (\psi_1, \psi_2, \dots, \psi_n)$. Once all of these parameters are defined for a particular task, the human-aware motion generation tool follows the algorithm depicted in Fig. 3-4.

In this algorithm, we can see that along with the human and robot transform sets, Ψ and Ω , we pass two additional values as parameters: d_{stop} and d_{slow} . These values define the distances at which our safety system orders the robot to stop and begin decelerating, respectively (see Section 2.4.3 for details about the safety system implementation). Once we provide all the parameters, the human-aware motion generation tool loops through all combinations of human and robot actions, $a_i \mid i = 1 \dots n$ and $a_j \mid j = 1 \dots n$, where $i \neq j$ (lines 2 and 3). For each action pair, we call `generateVirtualEnvironment(ψ_i, d')` to create a virtual environment con-

```

generateHumanAwareMotions( $\Psi, \Omega, d_{stop}, d_{slow}$ )
1:  $\Gamma \leftarrow \emptyset$ 
2: for  $i = 1$  to  $|\Psi|$  do
3:   for  $j = 1$  to  $|\Omega|$  do
4:     if  $i \neq j$  then
5:        $d' \leftarrow d_{slow}$ 
6:        $approved \leftarrow FALSE$ 
7:       while  $approved = FALSE$  do
8:          $success \leftarrow FALSE$ 
9:         while  $success = FALSE$  do
10:          generateVirtualEnvironment( $\psi_i, d'$ )
11:           $\gamma_{i,j}, success \leftarrow \text{runMotionPlanner}(\omega_j)$ 
12:           $d' \leftarrow d' - \delta$ 
13:        end while
14:         $approved \leftarrow \text{validateMotionPlan}(d', d_{stop})$ 
15:      end while
16:       $\Gamma \leftarrow \Gamma \cup \{\gamma_{i,j}\}$ 
17:    end if
18:  end for
19: end for

```

Figure 3-4: Pseudo-code describing the algorithm of the human-aware motion generation tool

taining a model of the human in the correct configuration for that particular action based on the transform for that action, $\psi_i \in \Psi$ (line 10). We also take into account d' which represents the buffer distance by which we extend the expected workspace occupancy in order to create motion plans which avoid not only the location of the human himself, but also as much of the safety system's robot deceleration region as possible.

Fig. 3-5 depicts a sample virtual environment generated by the tool. In the left side of the figure, we see the human workspace occupancy, defined by the transform ψ , represented by the two cocentric, green cylinders. In the right side of the figure, the volumes shown in red depict the buffer distance d' by which we are extending the occupancy model in order to avoid the robot deceleration region of the safety system when planning our human-aware motions.

After the tool generates the virtual environment, it then calls **runMotionPlanner**(ω_j) to actually generate a human-aware robot motion (line 11). In our implementation,

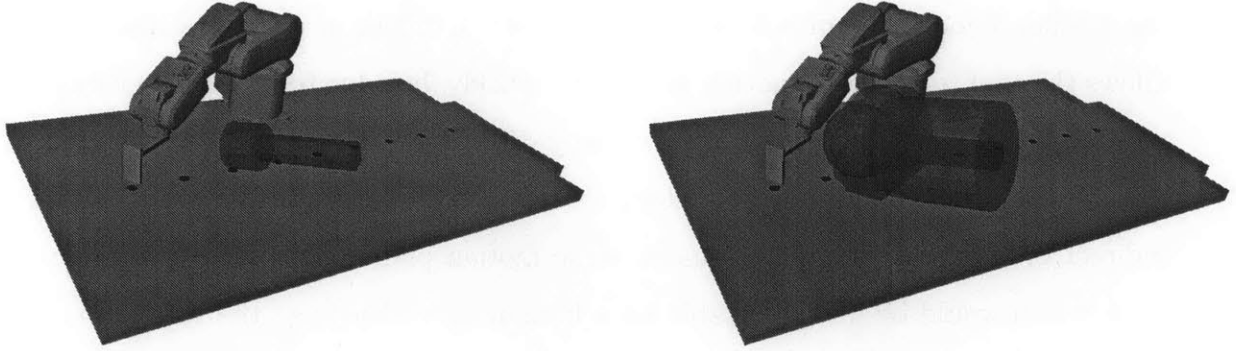


Figure 3-5: Sample virtual environment generated by the human-aware trajectory maker tool. The left side of the figure depicts the expected human workspace occupancy as two green cylinders. The right side of the figure shows the extension of this region to the safety system’s deceleration region.

we utilize the Constrained BiDirectional Rapidly-exploring Random Tree (CBiRRT) algorithm from the Constrained Manipulation Planning Suite (CoMPS) [13]. The motion planner attempts to find a solution which results in the robot configuration described by the transform ω_j while avoiding the expected human workspace occupancy extended by the slow down buffer d' . If the planner succeeds, we set the value of *success* to *TRUE* and store the generated trajectory in $\gamma_{i,j}$. If no valid solution can be found, we decrease the slow down buffer d' by δ (line 12) and try again, as this indicates that there is no possible path the robot could take to its goal which avoids the entire robot deceleration region. By decreasing d' by a small value during each iteration, the tool attempts to find a human-aware motion plan which has the shallowest penetration into the robot deceleration region. While this method allows for generated motions to guide the robot closer to the human worker than d_{slow} , it will never produce a motion which penetrates the expected human workspace occupancy, as the size of this region (depicted in green in Fig. 3-5) does not change with decreasing values of d' .

Once the planner succeeds in finding a solution, we call `validateMotionPlan(d' , d_{stop})` to determine if the motion plan is acceptable (line 14). First, the function verifies that the buffer distance by which we are extending the human workspace occupancy, d' , did not fall below the safety system’s stop distance threshold d_{stop} . Next,

the candidate motion plan is displayed to the user. Utilizing an RRT based algorithm allows the tool to generate motion plans very quickly, but due to its nondeterministic nature, the motion planner can return many different paths for the same configuration. The motion planner can, therefore, return motion plans which are unnecessarily indirect or complicated. Additionally, some motion plans might lead to robot motions which would be uncomfortable for a human co-worker (e.g., the solution might cause the robot to move too close to the human worker’s head). By displaying the computed motion plan to the user in simulation, we allow the user to decide whether or not the candidate motion plan appears acceptable. If the user approves of the candidate motion plan $\gamma_{i,j}$, the value of *approved* becomes *TRUE*, and we exit the outer loop of the algorithm. We then add $\gamma_{i,j}$ to our set of motion plans Γ (line 16) and move on to the next combination of human and robot actions. If the user does not approve of a candidate motion plan, the tool generates a new trajectory for the same action pair until a satisfactory solution is found.

3.3 Improving Robustness and Latency of the Robot Control System

Although the robot control system described in the previous sections of this chapter resulted in a functional human-aware robot, the particular hardware utilized caused some problems with robustness of the robot control system and latency of the safety system. This section describes these issues and the solutions developed to overcome them.

3.3.1 Trajectory Downsampling

The first problem encountered in the implementation of the system described in the previous section was that the trajectories generated by the CBiRRT algorithm could not be successfully executed by the ABB IRB-120 robot robustly. CBiRRT returns trajectories as time series of robot joint angles for the robot to move through. Occa-

sionally, the robot would fail to follow the path defined by the trajectory and throw an error indicating that the target configuration is too close to the current configuration. To address this issue, we developed a simple downsampling function that takes the robot trajectory γ generated by CBiRRT and a sampling rate s as parameters and returns a new trajectory that takes every s^{th} set of joint angles from the original trajectory. In order to ensure we do not lose track of the final robot configuration, the sampling is done by starting at the final joint angle configuration in γ and moving backwards through the time series. By setting the sampling rate s sufficiently high, we were able to generate robot trajectories that could be executed by our robot robustly.

3.3.2 System Latency Improvements

Another problem encountered in our implementation was related to the safety system's latency. It was noted that while the robot control system and safety system were running simultaneously, the latency of both systems was significantly higher than if either system was executed alone. While great effort was made to completely decouple the safety system from the robot control system, and the two systems are indeed completely independent software-wise, components of the two systems inevitably reside on the same physical robot hardware. After extensive troubleshooting, it was discovered that this, in fact, was the root of the problem. A simple test program was created which would connect the robot controller to the main computer used to host the core program. The program would perform a send and receive cycle 10,000 times and record how long each cycle took. When the program was running on a single robot process, it would take, on average, 0.35 ms/cycle. When two instances of the program were executed on two separate robot processes, however, this number increased to 13.66 ms/cycle for one of the tasks, which is nearly 40 times slower.

Upon consulting the documentation for the ABB IRB-120 robot, it was discovered that there are differences in how various tasks on the robot's controller are prioritized. When defining parallel tasks to be executed on the controller, only one task, called the "motion task," can be utilized to send movement commands to the robot. When

running multiple tasks simultaneously on a robot's controller, the task designated as the motion task gets priority in network communication. This prioritization scheme resulted in the aforementioned latency problems.

In our implementation, two tasks resided on the controller: the robot control algorithm, and the safety system. Since the robot control algorithm is the one sending movement commands to the robot, it was designated as the motion task. Due to the network prioritization scheme described above, the robot control task's communication with the core computer caused the safety system task's latency to increase significantly. Upon further investigation and testing, it was found that the most significant increases in safety system latency occurred when the motion task was awaiting a message on a network socket. This phenomenon is depicted in Fig 3-6. The first graph in the figure, graph (a), shows the latency of the safety system while a modified robot control algorithm is running on the second task. In this modified version, the robot performed only a single action. In this figure, one can see a region of initially high latency (mean of 57.03 ms), followed by a drop to a lower latency (mean of 17.75 ms). This discontinuity is caused by the robot control task waiting for a command to execute the action, as this requires waiting for a message on a network socket. Since this graph was produced by a modified task in which only one action was executed, the region of high latency occurs only once. In the full, unmodified version, however, this increase in safety system latency occurs before every single robot action execution, making it a serious hindrance. The second graph in the figure, graph (b), shows what the latency of the safety system looks like when it is executed alone, without the motion task taking priority of network communication. One can see that there is no initial region of high latency, and that the latency hovers around its mean of 20.87 ms. This discrepancy in safety system performance when running alone or with the robot control task in parallel was unacceptable, and thus a solution needed to be developed.

To address this problem, the robot control algorithm was modified such that it would not block and wait for a message on a network connection for extended periods of time. In this new algorithm, we added an extra communication step between the

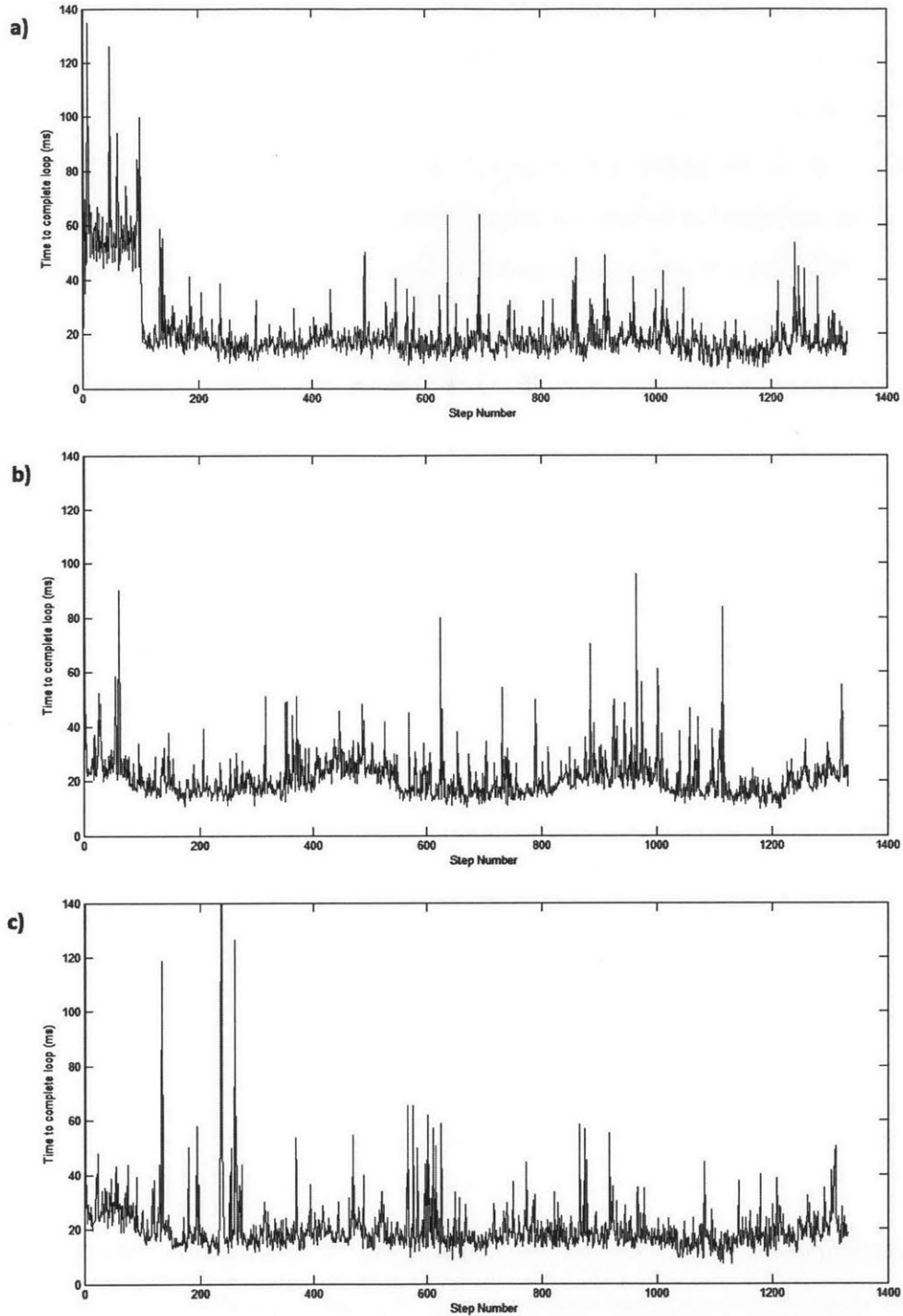


Figure 3-6: Illustration of the effects of robot task prioritization on the safety system task’s latency. Figure (a) depicts the latency of the safety system while the robot control task is running simultaneously before the fix was applied. Figure (b) shows what the latency looks like when the safety system is running alone. Figure (c) depicts what the latency looks like while both tasks are running after the implemented fix was applied.

main program and the robot's controller each time the system checks whether the next action should be executed or whether the next human and robot actions are the same and the robot should wait (line 5 in 3-3). Instead of simply waiting for the next action to be received, we added a loop to the robot controller's code which waits for messages indicating whether or not the next action is ready, pausing for 100 ms before each check. If the next action is ready to be executed, the system sends a message to the robot indicating this; if it is not, we send a different message. As a result, instead of having the robot continuously waiting for a message on a socket connection while it is waiting for the next robot action, the robot checks in only once every 100 ms, avoiding blocking the network connection and using it only momentarily. The result of this fix can be seen in the last graph of Fig. 3-6. This graph, labeled (c), shows the latency of the safety system while both it and the robot control task are active. We see that the period of high latency in the beginning is gone, and the latency plot looks much more similar to graph (b), in which the robot control task was not active at all. The latency is also similar to that shown in (b), averaging to 20.01 ms. It should be noted here that this latency was further improved later, as discussed in the previous chapter in section 2.4.4, to an average of 6.13 ms through further improvements and optimizations.

3.3.3 Final Robot Control Algorithm

After the implementation of the two fixes discussed in this section, the algorithm of the robot control system, first shown in Fig. 3-3, was changed to the final algorithm shown in Fig. 3-7. We can see the addition of the downsampling function, `downsampleTrajectory(γ)` on line 12, which stores the downsampled trajectory to be executed in the new variable γ' . We can also see the addition of the notifications sent to the robot controller, used to inform the robot whether or not the next action is ready, on lines 6 and 10.

```

robotControl( $A, \tau, \Gamma$ )
1:  $A_{rem} \leftarrow A$ 
2: while  $A_{rem} \neq \emptyset$  do
3:    $a_{robot} \leftarrow null$ 
4:    $a_{human} \leftarrow null$ 
5:   while  $a_{robot} = a_{human}$  do
6:     notifyController(false)
7:      $a_{human} \leftarrow \text{getNextHumanAction}(A, \pi_{human}, \tau)$ 
8:      $a_{robot} \leftarrow \text{getNextRobotAction}(A_{rem}, \pi_{robot}, \tau)$ 
9:   end while
10:  notifyController(true)
11:   $\gamma \leftarrow \text{getRobotTrajectory}(a_{human}, a_{robot}, \Gamma)$ 
12:   $\gamma' \leftarrow \text{downsampleTrajectory}(\gamma)$ 
13:  executeTrajectory( $\gamma'$ )
14:   $A_{rem} \leftarrow A_{rem} \setminus \{a_{robot}\}$ 
15: end while

```

Figure 3-7: Extended simplified pseudo-code describing the algorithm of the human-aware robot for selecting and performing actions in a human-robot collaborative task which implements fixes for robustness of trajectory execution and latency improvement

3.4 Conclusions and Future Work

In this chapter, we presented the design and implementation of a human-aware robot control architecture. Human-aware motion planning uses prediction of human actions together with human motion models to generate predictions of human workspace occupancy and plan robot motions which actively avoid conflict with these regions. We discussed how this type of system requires several sub-systems working in parallel, including a safety system, action detection, action override, and robot control. After discussing how each of these components was implemented, we discussed some improvements made to the system to make it more robust and operate with a lower latency.

Several improvements could be made to various aspects of our implementation in the future. As discussed in Section 3.2.5, our implementation utilizes an off-line human-aware motion planning technique. This requires a priori knowledge of action locations and appropriate human motion models for these actions. Such an off-line implementation makes it difficult to implement our architecture in a more dynamic

system, where, for example, motion models are updated throughout task execution or action locations can change in between task executions. As a result, our system stands to benefit from an on-line human-aware motion planning approach. In order to make such an implementation possible, our system would require a motion planning algorithm that produces appropriate motions consistently. As discussed in section 3.2.5, the current implementation uses an RRT based motion planner and occasionally returns motions that are overly complicated or indirect. As a result, in the current approach, our motion planning tool requires candidate robot trajectories to be approved by the user before being stored in the database. This type of pre-approval is impossible in an on-line approach, and, therefore, the current motion planning algorithm must either be augmented to produce appropriate motions consistently or replaced with a deterministic planner with an appropriately designed cost-function.

Another improvement which could be made in the motion planing component is to introduce additional factors geared toward making the motions comfortable for the human co-worker. Prior work in human-robot co-navigation has utilized factors such as the human's field of vision, posture, and positioning when planning paths for a robotic assistant [33] [32]. While these were previously used for path planning and not for close-proximity interaction, it would be interesting to investigate if these factors could prove useful in the latter domain as well.

Chapter 4

Analyzing the Effects of Human-Aware Motion Planning on Close-Proximity Human-Robot Collaboration

4.1 Introduction

A majority of the prior work mentioned in Chapter 1 aimed at bringing humans and robots closer together and allowing for close-proximity interaction focuses on the development of frameworks or algorithms without evaluating human response to these technologies. The developed systems were, for the most part, either evaluated only in simulation or with very limited experimentation with actual human participants. Without fully evaluating human response to adaptive robotic assistants through human-subject experimentation, however, it is impossible to predict whether the technologies would lead to improvements in team efficiency or human satisfaction. It is possible, for example, that the decreased predictability of an adaptive robot could cause human workers to trust it less than a robot which is preprogrammed, leading to decreased team efficiency.

From the works mentioned above which did evaluate human response to robotic assistants, they either did not deal with adaptive systems at all, or only considered adaptation on the task level. Motion-level robot adaptation, however, is critical for true close-proximity interaction, and to the author’s knowledge there has been no work thus far aimed at evaluating human response to this type of robot adaptation. The main contribution of the work presented in this chapter, therefore, was to utilize the human-aware robot architecture described in Chapter 3, and evaluate, through human-subject experimentation, whether this type of adaptation leads to more efficient teamwork and a more satisfied human co-worker.

4.2 Method

In order to investigate human response to robot adaptation on the motion planning level, we devised a human-subject experiment in which participants worked cooperatively with a robot on a collaborative task within a shared workspace. Human-Aware Motion Planning was used as the motion-level adaptation technique (see Section 3.1.1 for details). The particular robot used in our experiment was the ABB IRB-120, which is a standard 6 DOF industrial robot shown in Figure 4-1. A PhaseSpace motion capture system was used to track the human within the workspace and detect what actions he or she is taking. A real-time safety system was deployed on the robot to adjust the robot’s speed based on the separation distance between the human and robot, gradually decreasing the robot’s speed to a complete stop, as necessary (see Chapter 2 for details).

The robot was programmed such that it could operate in two motion planning modes: standard and human-aware. As described in Section 3.2.5, a database of robot motions was computed off-line for every possible combination of human and robot action using the Constrained BiDirectional Rapidly-exploring Random Tree (CBiRRT) algorithm from the Constrained Manipulation Planning Suite (CoMPS) [13]. The decision to generate robot motion plans off-line was made to ensure the robot motions were consistent throughout the experiment, since the CBiRRT algorithm is

based on the Rapidly-Exploring Random Tree (RRT) algorithm, which inherently may produce different motions each time. Both human-aware and standard motions were planned with the CBiRRT algorithm.

4.2.1 Task

The task used in the experiment is depicted in Figure 4-1. In this task, participants placed eight screws at designated locations on a table while the robot pretended to apply a sealant to each screw by dipping a brush in a centrally positioned container and then moving the brush over the screw. Each participant was instructed to twist the screw one full rotation before moving on to the next one.

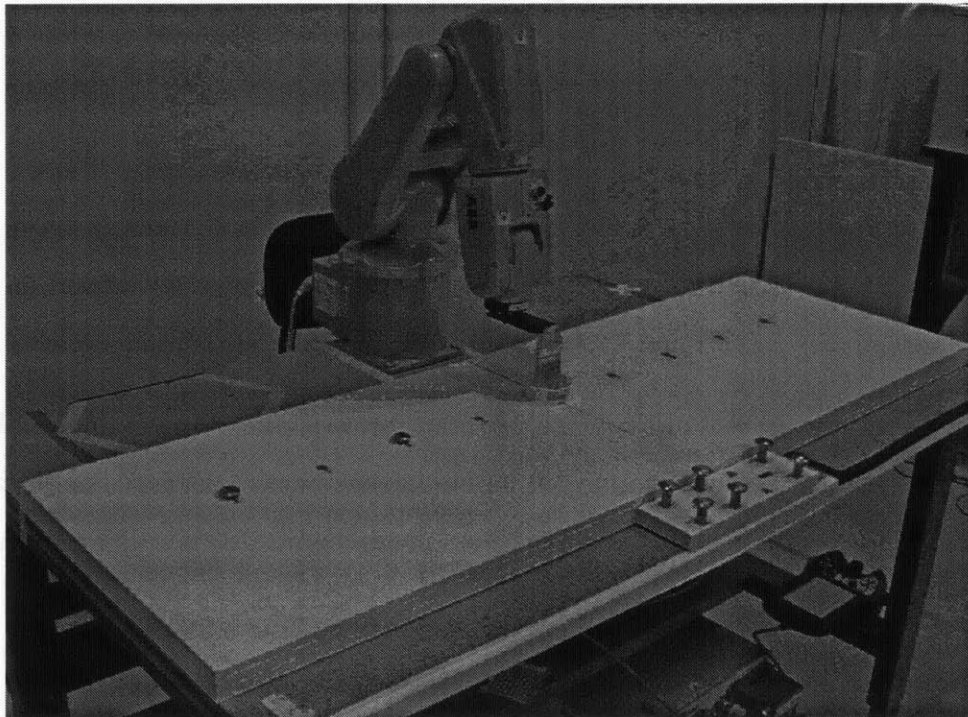


Figure 4-1: Photograph of the task setup used in the human-subject experiments.

The screws were placed by the participants in a predefined order. This, in effect, simulated perfect human action prediction, which allowed for the measuring of the effects of motion adaptation independent of the accuracy of action prediction. The screws were split into two groups by color: yellow and red. The participants first placed the four red screws, after which they were instructed to wait for a sound cue

before proceeding with the yellow set of screws. Splitting the screw placement in this manner allowed us to control what types of motion conflicts the subject would experience by not allowing for him or her to work too fast and get out of synchronization with the robot.

Numbering the screws in sequence from left to right, the particular sequence in which the screws were placed was 1, 3, 8, 6, 2, 4, 7, 5. This sequence was chosen due to its balance of conflicting and non-conflicting motion. For example, after the human worker places screw #1, the robot moves across from the centrally positioned container to this screw while the human worker moves to place screw #3. This creates a potential motion conflict, as the shortest-distance route to screw #1 from the sealant container goes above screw #3. After placing this screw, however, the human worker places screw #8 next while the robot begins working on sealing screw #3. In this case, a motion conflict does not exist. By examining the rest of the sequence in a similar fashion, one can see that the chosen screw placement order results in equal numbers of conflicted and non-conflicted motions.

4.2.2 Participants

The participant pool consisted of 20 Massachusetts Institute of Technology affiliates, including undergraduate and graduate students, postdoctoral associates, and visiting students. Of the 20 participants, 7 were female and 13 were male. The ages ranged from 19 to 39 ($M = 27.1$, $SD = 6.24$).

4.2.3 Procedure

The experiment was a repeated-measure design consisting of two randomly selected groups of subjects: those who worked with a human-aware robot first ($n = 11$) and those who worked with a robot using standard motion planning first ($n = 9$). The subjects were not informed what the two conditions were or what the experiment was attempting to measure prior to the experiment's completion.

The procedure for the experiment is depicted in Figure 4-2. First, in order to get

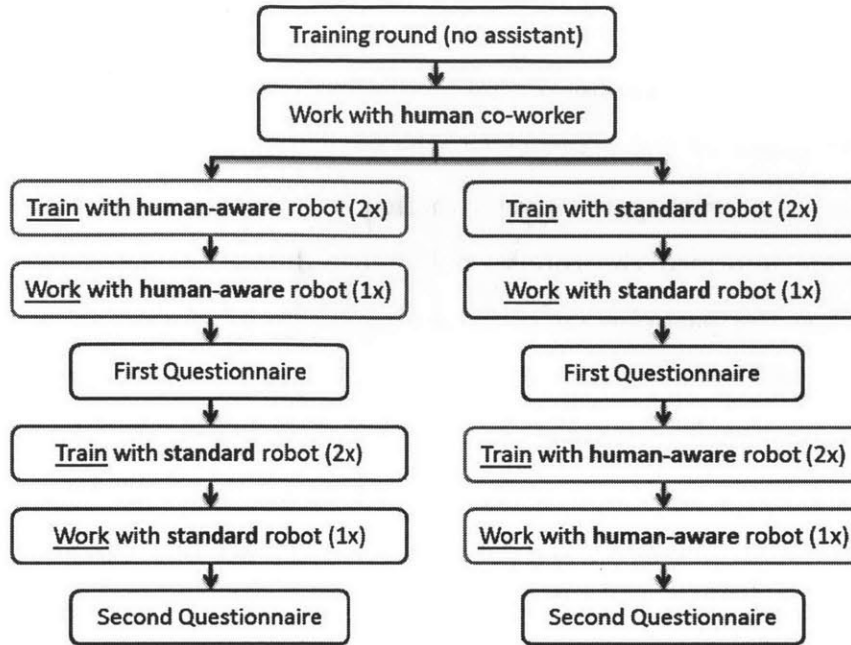


Figure 4-2: Diagram depicting the experimental procedure. The group on the left represents the “human-aware first” condition, while the group on the right represents the “standard first” condition.

accustomed to the task and practice placing the screws in the designated sequence, participants in both groups executed a training round in which they placed the screws without an assistant applying the sealant. Next, all participants performed the task with a human assistant. This was done to implicitly prime our subjects to work with a robot in subsequent task executions in a way similar to when working with a person. In order to prevent any unintentional bias from the experimenter, who acted as the human co-worker in this portion of the experiment, this first task execution was done in a double-blind fashion, with the experimenter unaware of which of the two conditions the participant was assigned to as well.

After performing the task with a human co-worker once, the participants performed the same task with a robotic assistant. The participants in the “human-aware first” condition worked with the robot in the human-aware motion planning mode while those in the “standard first” condition worked with the robot in the standard motion planning mode. Each participant performed two training task executions with the robot in order to practice working with a robotic assistant and build a mental

model of its behavior. After two training executions, the task was performed one more time, followed by an administration of the first questionnaire.

In the next phase of the experiment, the mode of the robot was switched to the opposite of what each participant experienced thus far. Prior to continuing, participants were informed the robot would move differently in the second phase in order to allow the participants to expect a change in motion so that the new robot behavior would not startle them. To avoid revealing to the participants the robot’s modes are linked to whether it adapts its motions or not, the participants were simply told that a “few robot motion parameters were changed.” After informing the participants of the change, they performed another set of training rounds with the robot and a final task execution. At this point, the participants filled out a second questionnaire that directly compared the robot behavior in the second mode experienced by each participant to that of the first mode.

4.2.4 Dependent Measures

The dependent measures considered for the evaluation of human response to human-aware motion planning, and the possible derived benefits, were split into two main groups: quantitative metrics of team fluency and subjective evaluation by the participants. The first group of metrics was based on those proposed by Hoffman and Breazeal [20] and expanded to consider additional measures. These metrics include: task execution time, amount of concurrent motion, human idle time, robot idle time, and human-robot separation distance.

The quantitative team fluency metrics were precisely defined as follows:

- **Task Execution Time:** The amount of time, in seconds, elapsed from the beginning of the first human motion toward placing the first screw until the robot finishes applying the sealant to the last screw.
- **Amount of Concurrent Motion:** The percentage of the time in which concurrent motion was observed during times in which concurrent motion was possible. Times in which concurrent motion was possible include the time span

from the beginning of the first robot motion to the end of the last human motion, with the exception of the time the human spends waiting for the sound cue indicating he or she can begin placing the second set of screws.

- **Human Idle Time:** The percentage of time during which the human worker was not moving when he or she still had actions left to perform. The time spent idle while waiting for the sound cue was, once again, removed from consideration. Additionally, any time spent not moving while twisting a screw was not considered idle time, as the person was actively working on performing the action even though his or her hand was not translating.
- **Robot Idle Time:** The percentage of time during which the robot was not moving while it still had actions left to perform. This includes both time spent waiting for the human worker to perform their portion of the joint action as well as time spent not moving due to motion conflict.
- **Human-Robot Separation Distance:** The average shortest distance, in cm, between any part of the human worker's hand or lower arm as depicted by the two cocentric cylinders of the safety system (see Chapter 2 for details) and the robot's end-effector. To prevent diluting the separation distances with large values when the human is not performing any action, only separation distances when the human passes a threshold which puts him or her within the reach of the robot (the "interaction threshold") were considered.

The second group of metrics was based on subjective evaluation of the robotic assistant based on questionnaire responses. The questions asked attempted to extract each participant's satisfaction with the robot as a teammate as well as their perceived safety and comfort. The questionnaire items are shown in Table 4.1.

Each of the questions was asked on a 5-point Likert scale ranging from "Strongly Disagree" to "Strongly Agree" in the first questionnaire and "Much Less" to "Much More" in the second questionnaire.

Table 4.1: Questionnaire Items

Satisfaction with robot as a teammate:

1. I trusted the robot to do the right thing at the right time.
 2. The robot did not understand how I wanted to do the task.
 3. The robot kept getting in my way.
 4. The robot and I worked well together.
-

Perceived Safety and Comfort:

5. I felt safe when working with the robot.
 6. The robot moved too fast for my comfort.
 7. The robot came too close to me for my comfort.
 8. I trusted the robot would not harm me.
-

4.2.5 Hypotheses

Based on the dependent measures described in the previous section, the two main hypotheses in this experiment were:

- *H1*. Utilizing human-aware motion planning will lead to more fluent human-robot teamwork, including shorter task execution time, more concurrent motion, lower human idle time, lower robot idle time, and a larger human-robot separation distance, when compared to standard motion planning.
- *H2*. Participants will be more satisfied with the human-aware robot’s performance as a co-worker and feel more comfortable and safe working with it when compared to a robot that uses standard motion planning.

4.3 Results

4.3.1 Quantitative Team Fluency Metrics

When comparing the participants’ performance working with a human-aware robot to their performance when working with a robot using standard motion planning, significant differences were found for all quantitative team fluency metrics with the use of paired t-tests. When working with a human-aware robot, participants completed the task 5.57% faster ($p = 0.038$), with 19.9% more concurrent motion ($p < 0.001$),

2.96% less human idle time ($p = 0.019$), 17.3% less robot idle time ($p < 0.001$), and a 15.1% larger separation distance ($p < 0.001$). The mean values of both robot modes for each of these metrics along with error bars depicting standard error of the mean are shown in Figure 4-3.

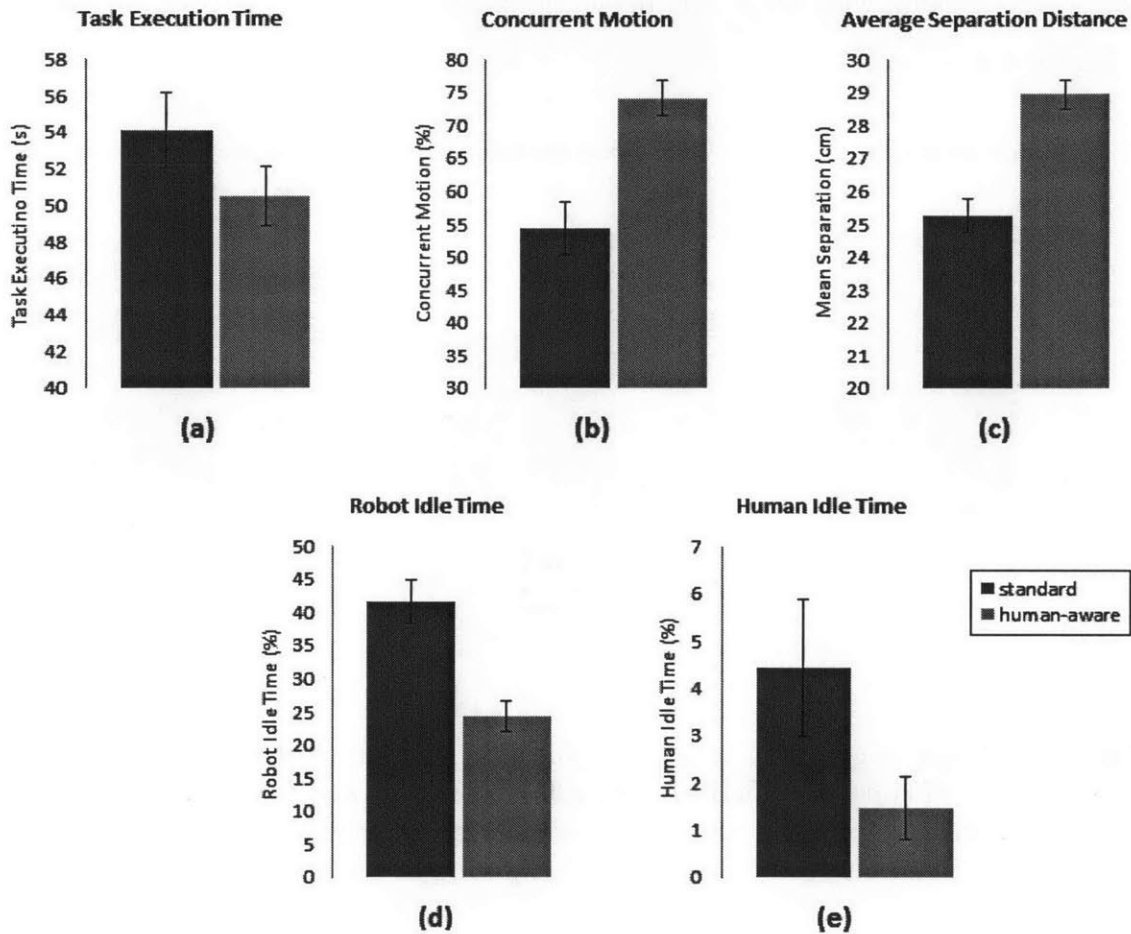


Figure 4-3: Mean values, with error bars indicating standard error of the mean (SEM), of (a) task execution time, (b) percent of concurrent motion, (c) average separation distance between the human and robot, (d) robot idle time, and (e) human idle time for the standard and human-aware robot executions

When considering data collected up to the administration of the first survey (see Figure 4-2), each participant was only exposed to a single robot type. Consequently, we can treat the team fluency metric data from the two groups of participants, those who worked with a standard robot and those who worked with a human-aware robot, as independent. Significant differences between these populations emerged for three

of the five team fluency metrics considered. The group of participants which worked with the human-aware robot completed the task with 15.0% more concurrent motion ($p = 0.028$), 14.6% less robot idle time ($p = 0.021$), and a 17.3% larger separation distance ($p < 0.001$). The mean values of both groups of participants for each of these metrics along with error bars depicting standard error of the mean are shown in Figure 4-4.

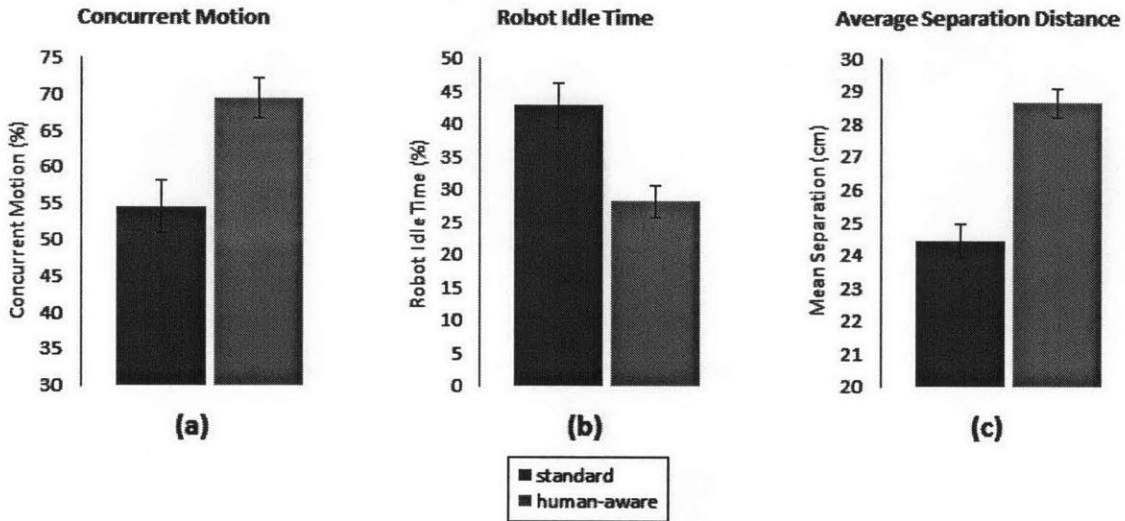


Figure 4-4: Mean values, with error bars indicating standard error of the mean (SEM), of (a) percent of concurrent motion, (b) robot idle time, and (c) average separation distance between the human and robot for groups of participants working with the standard and human-aware robots prior to exposure to the second robot type

4.3.2 Subjective Evaluation

As one can see from Figure 4-2, the first questionnaire used for the subjective assessment of the robotic assistant was given prior to each participant’s exposure to the second robot mode, regardless of whether they were in the “human-aware first” or “standard first” condition. As a result, we can treat the responses of these questionnaires as coming from two independent populations: one that worked with the human-aware robot, and one that worked with a standard robot. We therefore used the Mann-Whitney-Wilcoxon Test to determine whether these two groups provided

significantly different responses to the questions listed in Table 4.1. Using the numbers shown in this table, significant differences (at $\alpha = 0.05$) were found for three of the questions. The participants exposed to the human-aware robot disagreed more strongly with statements #2 ($p = 0.012$), #3 ($p < 0.001$), and #7 ($p = 0.05$).

After being exposed to both conditions, the participants were asked to directly compare the robot in the first mode to the robot in the second mode. Since “first mode” and “second mode” signified different modes depending on whether a participant was in the “human-aware first” or “standard first” condition, we could once again treat the two groups as independent, and test whether the responses of these groups were significantly different using the Mann-Whitney-Wilcoxon Test. All of the questions produced significantly different results with the exception of “The robot moved too fast for my comfort.” In the “human-aware first” condition, participants agreed more strongly with “I trusted the robot to do the right thing at the right time” ($p < 0.001$), “The robot and I worked well together” ($p < 0.001$), “I felt safe when working with the robot” ($p < 0.001$), and “I trusted the robot would not harm me” ($p = 0.008$) and disagreed more strongly with “The robot did not understand how I wanted to do the task” ($p = 0.046$), “The robot kept getting in my way” ($p < 0.001$), and “The robot came too close to me for my comfort” ($p < 0.001$).

4.4 Discussion

4.4.1 Differences in Team Fluency

The results presented in the previous section provide strong support for both of our hypotheses. The significant differences in favor of human-aware motion planning in all quantitative metrics of team fluency show that this type of motion planning indeed leads to more effective human-robot teamwork (H1). The major significance of this result is that these improvements can be achieved with participants who were never previously exposed to a robot capable of human-aware motion planning. After only two practice task executions and without any explanation of the system’s capability

to adapt its motion planning, participants were able to take advantage of the robot’s adaptive techniques and form a more effective team with the robot. This result suggests that given even minimal demonstration of the robot’s ability to avoid motion conflicts, human workers inherently begin to exploit this capability, just as they would when working with a human assistant who performs this type of adaptation implicitly. In light of previously mentioned research, which found some participants expect a robot to adapt on the task level just like a human would [20], it is possible that a similar expectation is placed on motion-level adaptation, and that a lack of adaptation not only leads to inefficient teamwork, as was shown by the quantitative metrics, but also an unsatisfied human worker, as discussed next.

4.4.2 Differences in Perceived Safety and Comfort

Even before exposure to both robot modes, as shown by the results of the first questionnaire, we already saw significant differences in satisfaction with the robot as a teammate. This can be seen from the fact that our participants disagreed more with statements like “The robot did not understand how I wanted to do the task” and “The robot kept getting in my way” when working with the human-aware robot. Additionally, we saw that participants were less comfortable with the robotic assistant which used standard motion planning, with more participants agreeing with the statement “The robot came too close to me for my comfort.”

Once the participants finished working with both the human-aware and standard robots and filled out the comparison questionnaire, these results were even more pronounced. In addition to the three questionnaire items which yielded significant differences in the first questionnaire, when directly comparing the two robot modes, participants agreed more with “I trusted the robot to do the right thing at the right time,” “The robot and I worked well together,” “I felt safe when working with the robot,” and “I trusted the robot would not harm me” when describing the human-aware robot. The first two of these questionnaire items once again indicates that our participants found the human-aware mode to result in a more satisfying interaction. The last two items indicate the participants also felt more comfortable and safe

when working with the human-aware robot. Collectively, these results provide strong support for our second hypothesis (H2).

The only item on the questionnaire that did not yield significantly different results was “The robot moved too fast for my comfort.” This is, in a way, an expected result, as the base speed of the robot was the same in both conditions. The safety system mentioned in the beginning of Section 4.2 did slow down and stop the robot if the distance between the human and robot workers fell below certain thresholds, but this behavior was identical in both conditions as well.

Since the safety system was running identically in both conditions, the higher perceived safety and comfort ratings when working with a human-aware robot are an interesting phenomenon. In terms of physical safety, participants in both conditions were equally at a very low risk of unwanted contact with the robot due to the use of the safety system. However, due to the robot taking evasive maneuvers in the human-aware condition, and thus having to rely less on the safety system to avoid collision, the participants were exposed to less sudden robot stops. We hypothesize that participants felt safer when working with the human-aware robot for this reason. While physical safety was the same in both conditions, having higher perceived safety is an important result, as low perceived safety can be a high stress situation for a human worker, and continuous exposure to stress has been shown to have negative long-term effects on health [26].

4.4.3 Human and Robot Idle Time

Another interesting observation can be made by comparing the percentages of human and robot idle time. As one can see from Figure 4-3, the average human idle time was far lower than robot idle time for both conditions. These differences were analyzed with a paired t-test, and statistical significance was shown for both the difference between human idle time and robot idle time for when participants worked with the standard motion planning robot ($p < 0.001$) and the human-aware robot ($p < 0.001$). This suggests that in a human-robot team where motion conflict prevents both agents from performing their tasks simultaneously, people prefer to perform their task and

make the robot wait over waiting for the robot to perform its task first. This result is similar to that observed by Unhelkar et. al. [35], in whose experiment it was shown that human workers are more likely to make a robotic assistant wait for them than doing the same to a human assistant. In the present experiment, we showed that human workers prefer to make the robot wait over themselves, and thus it appears, based on the results of the present study and the one by Unhelkar et. al., that people consider human time, whether one's own or another's, more valuable than that of a robot.

4.5 Conclusion

In this chapter, we described a human subject experiment aimed at studying human response to human-aware motion planning and quantifying its potential benefits, including quantitative team fluency metrics as well as subjective evaluation of satisfaction, safety, and comfort. Through the results of this experiment, it was shown that people learn to take advantage of human-aware motion planning even with novel tasks, with very limited training, and with no indication that the robot's motion planning is adaptive. It was shown that participants working with a human-aware robot form a more effective team, performing a collaborative task in less time, with more concurrent motion, less human and robot idle time, and while maintaining a larger separation distance. Furthermore, qualitative evaluations showed human workers were more satisfied with the human-aware robot as a teammate, and perceived it to be more comfortable and safe to work with. This signifies that human-aware motion planning leads to satisfying human-robot interaction.

The fact that human-aware motion planning leads to higher perceived safety, and thus less potential for stress-related health problems, while simultaneously improving team fluency, is a very important result. Being able to show simultaneous improvement in efficiency and human worker satisfaction and well-being makes human-aware motion planning a highly desirable tool for close-proximity human-robot interaction. These results bring us one step closer toward successful introduction of robotic as-

sistants into previously human-only domains domains, as we can show that close-proximity interaction is not only possible, but can also be made efficient and satisfying with the help of human-aware motion planning. Furthermore, by showing human-aware motion planning is effective with actual human-subject experiments instead of simulations as was done before, we can strongly motivate future research in all the facets that would make a real-time human-aware system possible, including action prediction algorithms, development of rapid motion planning techniques, human motion model development, and many others.

One important point to consider, however, is that the improvements presented in this paper were obtained with the robot having perfect knowledge of what action the human will take next, since the sequences of actions were preset, as was mentioned in the 4.2.1 section. Consequently, the improvements shown through our experiment should be viewed as an upper bound of possible increases in team fluency and human worker satisfaction. As one might imagine, these improvements would not be as pronounced with imperfect action prediction, and would depend very highly on its accuracy. Nonetheless, the improvements shown in this paper are highly significant, leading us to believe that substantial improvements can be derived even with imperfect action prediction. Evaluating how performance of a team utilizing a human-aware robot changes with varying levels of action prediction accuracy is very important, and is a planned future avenue of research.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

With the rapid expansion of the use of robots in the recent decades, especially since the turn of the century, robots have been successfully deployed in a wide variety of domains. Despite this fact, we still see many tasks in these domains being done in a purely manual fashion with no robotic assistance. Due to a necessity for a high level of judgment, dexterity, and flexible decision making that surpasses current robots' abilities, this subset of tasks requires human operators for successful completion. As such, there is a significant incentive for the development of technologies which could enable robots to safely and efficiently collaborate with people in shared workspaces.

5.1.1 Real-Time Safety System for Human-Robot Interaction

While a significant amount of research has been done in support of enabling safe human-robot interaction, the prior safety systems were found to be not capable of supporting continuous interaction at low distances of separation. These systems either required new, specialized hardware, such as new actuators or even entire robotic platforms, or used methods which utilize coarse discretizations or large "safety regions" which are incompatible with continuous, close-proximity interaction.

The first contribution of this thesis, described in Chapter 2, thus became the development and evaluation of a real-time safety system specifically designed for close-proximity human-robot interaction. This safety system, based on the use of a virtual representation of the shared workspace that is continually updated, leverages accurately known human position information from an external sensor with precise robot configuration data from the robot’s controller to allow for precise calculation of separation distance in real time. The real-time separation distance is then fed to a robot speed adjustment function, whose flexible parameters can be tuned to modify the speed reduction behavior of the robot as needed for a particular application. This safety system scheme was shown to be capable of providing safe human-robot interaction even when the stopping threshold of the system was adjusted to just 6 cm, allowing for true close-proximity interaction.

The latency of the system was also greatly reduced through various methods, including CAD model adjustment and network optimization, resulting in an average latency of 6.13 ms. A statistical analysis of the latency measurements collected for the safety system showed that system latencies are expected to fall below 9.64 ms with 95% probability, below 11.10 ms with 99% probability, and below 14.08 ms with 99.99% probability. These data allowed us to classify our safety system as soft real-time.

An important aspect to consider when reviewing these results is that this safety system performance was derived with the use of a standard industrial robot without any specialized actuators, new robot sensors, or any other robot hardware modification. This indicates that the developed safety system can be deployed on currently used industrial robots, which makes it a very simple and cost effective solution, as it does not require replacing the robots with new designs or difficult, or even impossible, retrofitting with new actuators or sensors.

5.1.2 Human-Aware Robot Control Architecture

The second main contribution of this thesis was the development of an end-to-end system to be used for human-robot collaboration, as described in Chapter 3. The

architecture uses a multithreaded structure, with several sub-components running in parallel. The first component, human action tracking, uses flexibly defined task volumes and time intervals to track what actions have been performed by a human worker. Next, an action override thread is used to provide a manual override for the detected actions in case the automatic detection fails. Another main thread is used to control what actions the robot should take and what motion plans it should use, which is based on what actions the human has taken thus far and what action he or she is expected to take next. Finally, the safety system, described in Chapter 2, runs as a separate thread as well.

The robot trajectories used by robot control thread are generated based on a motion planning technique we call Human-Aware Motion Planning. In this technique, the system predicts what action the human will take next, and then, based on an appropriate motion model, predicts what portion of the shared workspace the human will use while taking this action. The robot then picks a path to its goal which avoids this location. To ensure consistency and predictability of robot motions, the human-aware trajectories were constructed off-line with the use of a specially designed trajectory maker tool.

Due to some problems with robustness and latency caused by the particular hardware used, additional steps were taken to improve the system. First, trajectory downsampling was utilized in order to allow the ABB IRB-120 robot to follow paths generated by the trajectory maker tool consistently. Next, the issue with the latency of the safety system increasing when the robot control task is running was fixed by ensuring that socket communication in the main task on the robot's controller did not interfere with the safety system task. The implementation of these fixes resulted in a robust, low latency, end-to-end system capable of supporting human-robot collaboration with motion-level adaptation at low distances of separation.

5.1.3 Analysis of Effects of Human-Aware Motion Planning

The final major contribution of this thesis was presented in Chapter 4. In this chapter, the human-aware robot architecture was used in a full-scale human-subject experi-

ment ($n = 20$) aimed at analyzing the effects of motion-level adaptation on human-robot collaboration. Through the experiment, which involved working with a robot on a joint task of placing screws and applying sealant, it was shown that people respond well to motion-level robot adaptation, even with very limited training and with no indication of the robot’s capability to adapt. When compared to a baseline motion planning method of using shortest-path motions, when working with the robot which used human-aware motion planning, participants completed the task 5.57% faster ($p = 0.038$), with 19.9% more concurrent motion ($p < 0.001$), 2.96% less human idle time ($p = 0.019$), 17.3% less robot idle time ($p < 0.001$), and a 15.1% larger separation distance ($p < 0.001$).

In terms of subjective evaluation, when describing the human-aware robot, participants agreed more strongly with “I trusted the robot to do the right thing at the right time” ($p < 0.001$), “The robot and I worked well together” ($p < 0.001$), “I felt safe when working with the robot” ($p < 0.001$), and “I trusted the robot would not harm me” ($p = 0.008$) and disagreed more strongly with “The robot did not understand how I wanted to do the task” ($p = 0.046$), “The robot kept getting in my way” ($p < 0.001$), and “The robot came too close to me for my comfort” ($p < 0.001$).

5.2 Future Work

While the results of this thesis are very promising, the current implementation and evaluation is restricted to application on one specific industrial robot. In order to allow for greater flexibility in deploying the architecture designed in this work to other robots and domains, a modular design could be utilized with the use of a middleware such as Orocos RTT or ROS. Providing the information about the state of the human and robot could then be implemented as interchangeable nodes, independent of the source of information, making it possible to use a wide variety of robots and sensors with the architecture developed in this work.

Since the evaluation of human-aware motion planning done in this work was focused on pulling out potential benefits of this motion planning technique independent

of the accuracy of action prediction, an action prediction capability was not implemented. In order to incorporate the human-aware motion planning techniques and take advantage of their benefits, however, an effective method of predicting human intent must be developed. Based on prior work in this topic, it appears that a multimodal approach is advisable. The first layer would be based on modeling human tasks and actions in order to extract and utilize task-level knowledge for prediction purposes. One possible method is to model the tasks and actions with a MDP structure similar to [29]. In this model, we define a set of states S , a finite set of actions A , a state transition probability function T , which for every state and robot action gives a probability distribution of future world states, and a reward function R . The robot could then calculate an optimal policy with the use of dynamic programming. By formulating the system in this manner, we can incorporate prior knowledge of task sequences to generate the initial transition probability and reward functions, and then continuously update these as new sequences are observed. This formulation also allows us to derive a quantitative metric for the robot's uncertainty about the human's next action by calculating the entropy rate of a Markov Chain given by setting a specific robot policy π . One challenge with using this method is that as the number of possible states increases, the amount of data needed to teach the robot to effectively learn how people might like to perform tasks increases greatly. Therefore, a new, creative method will need to be developed to properly guide the evolution of the transition and reward functions in a more efficient way.

The second layer of modeling human intent could be based on analyzing human motion and speech. An approach similar to the one described in [25] could be utilized, where early motion is used to predict what action the person is taking. This could be done by passively recording subjects in a lab as they perform their tasks and collecting motion data, which can then be utilized to train Hidden Markov Models (HMMs) to be used for action classification. In order to enhance the accuracy of the system, new parameters, like head orientation and gaze direction, could be incorporated as new features used to train the model. Finally, research on natural language processing (NLP) could be used to analyze verbal cues that convey intent. All of these various

modalities, including both task-level information and human motion and speech, will then need to be combined to generate an effective human intent prediction system.

Bibliography

- [1] ISO 10218-1:2011 Robots and robotic devices – Safety requirements for industrial robots – Part 1: Robots. International Organization for Standardization, <http://www.iso.org>, 2011.
- [2] Dual-arm concept robot: Safety. ABB Group, <http://www.abb.us>, 2012.
- [3] World robotics industrial robots 2013 (executive summary). Technical report, International Federation of Robotics (IFR), 2013.
- [4] Baxter safety and compliance overview. Rethink Robotics, <http://www.rethinkrobotics.com>, 2014.
- [5] ISO/DTS 15066 Robots and robotic devices – Safety requirements for industrial robots – Collaborative operation. International Organization for Standardization, <http://www.iso.org>, 2014.
- [6] The Orocos real-time toolkit. The Orocos Project, <http://www.orocos.org/rtt>, 2014.
- [7] RAPID reference manual. ABB Group, <http://www.abb.com>, 2014.
- [8] Safety of human-robot systems in fixed workcell environments. The National Institute of Standards and Technology (NIST), <http://www.nist.gov>, 2014.
- [9] Terminology and notation. IEEE Technical Committee on Real-Time Systems <http://www.tcrrts.org/>, 2014.
- [10] Rachid Alami, Raja Chatila, Aurélie Clodic, Sara Fleury, Matthieu Herrb, Vincent Montreuil, and Emrah Akin Sisbot. Towards human-aware cognitive robots. In *Proc. The Fifth International Cognitive Robotics Workshop (The AAAI-06 Workshop on Cognitive Robotics)*, 2006.
- [11] T Arai, R Kato, and M Fujita. Assessment of operator stress induced by robot collaboration in assembly. *CIRP Annals-Manufacturing Technology*, 59(1):5–8, 2010.
- [12] Maren Bennewitz, Wolfram Burgard, Grzegorz Cielniak, and Sebastian Thrun. Learning motion patterns of people for compliant robot motion. *The International Journal of Robotics Research*, 24(1):31–48, 2005.

- [13] Dmitry Berenson, Siddhartha Srinivasa, and James Kuffner. Task space regions: A framework for pose-constrained manipulation planning. *International Journal of Robotics Research (IJRR)*, 30(12):1435–1460, October 2011.
- [14] Shu-Yun Chung and Han-Pang Huang. Predictive navigation by understanding human motion patterns. *International Journal of Advanced Robotic Systems*, 8(1), 2011.
- [15] Rosen Diankov and James Kuffner. Openrave: A planning architecture for autonomous robotics. *Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-08-34*, page 79, 2008.
- [16] Anup Doshi and Mohan M Trivedi. On the roles of eye gaze and head dynamics in predicting driver’s intent to change lanes. *Intelligent Transportation Systems, IEEE Transactions on*, 10(3):453–462, 2009.
- [17] Anca Dragan and Siddhartha Srinivasa. Generating legible motion. In *Robotics: Science and Systems*, June 2013.
- [18] Anca D Dragan, Kenton CT Lee, and Siddhartha S Srinivasa. Legibility and predictability of robot motion. In *Human-Robot Interaction (HRI), 2013 8th ACM/IEEE International Conference on*, pages 301–308. IEEE, 2013.
- [19] Sami Haddadin, A Albu-Schaffer, Alessandro De Luca, and Gerd Hirzinger. Collision detection and reaction: A contribution to safe physical human-robot interaction. In *Proc. IEEE/RSJ International Conference Intelligent Robots and Systems (IROS)*, pages 3356–3363. IEEE, 2008.
- [20] Guy Hoffman and Cynthia Breazeal. Effects of anticipatory action on human-robot teamwork efficiency, fluency, and perception of team. In *Proceedings of the ACM/IEEE international conference on Human-robot interaction*, pages 1–8. ACM, 2007.
- [21] Nicolas De Keijser. ABB’s smallest robot - IRB 120 for flexible & compact production. Product presentation IRB 120 Revision E, <http://www.abb.com>.
- [22] Soenke Kock, Jan Bredahl, Peter J. Eriksson, Mats Myhr, and Kevin Behnisch. Taming the robot: better safety without higher fences. *Doc no. 9AKK105152A2830* <http://www.abb.com>, 2006.
- [23] Woo Young Kwon and Il Hong Suh. A temporal bayesian network with application to design of a proactive robotic assistant. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3685–3690. IEEE, 2012.
- [24] Claus Lenz, Suraj Nair, Markus Rickert, Alois Knoll, W Rosel, Jürgen Gast, Alexander Bannat, and Frank Wallhoff. Joint-action for humans and industrial robots for assembly tasks. In *Robot and Human Interactive Communication, 2008. RO-MAN 2008. The 17th IEEE International Symposium on*, pages 130–135. IEEE, 2008.

- [25] Jim Mainprice and Dmitry Berenson. A human-robot collaborative manipulation planning framework that reasons on early prediction of human motion. In *Proc. IEEE International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2013.
- [26] Bruce S McEwen and Eliot Stellar. Stress and the individual: mechanisms leading to disease. *Archives of Internal Medicine*, 153(18):2093, 1993.
- [27] Eric Meisner, Volkan Isler, and Jeff Trinkle. Controller design for human-robot interaction. *Autonomous Robots*, 24(2):123–134, 2008.
- [28] John Nagle. Congestion control in IP/TCP internetworks. 1984.
- [29] Stefanos Nikolaidis, Przemyslaw Lasota, Gregory Rossano, Carlos Martinez, Thomas Fuhlbrigge, and Julie Shah. Human-robot collaboration in manufacturing: Quantitative evaluation of predictable, convergent joint action. In *Proc. International Symposium on Robotics (ISR)*, pages 1–6. IEEE, 2013.
- [30] Paul Rybski, Peter Anderson-Sprecher, Daniel Huber, Chris Niessl, and Reid Simmons. Sensor fusion for human safety in industrial workcells. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3612–3619. IEEE, 2012.
- [31] Manshi Shukla and Amar Nath Shukla. Growth of robotics industry early in 21st century. *International Journal of Computational Engineering Research (IJCER)*, 2 (5), pages 1554–1558, 2012.
- [32] Emrah Akin Sisbot, Luis F Marin-Urias, Xavier Broquere, Daniel Sidobre, and Rachid Alami. Synthesizing robot motions adapted to human presence. *International Journal of Social Robotics*, 2(3):329–343, 2010.
- [33] Emrah Akin Sisbot, Luis Felipe Marin-Urias, Rachid Alami, and Thierry Simeon. A human aware mobile robot motion planner. *Robotics, IEEE Transactions on*, 23(5):874–883, 2007.
- [34] Giovanni Tonietti, Riccardo Schiavi, and Antonio Bicchi. Design and control of a variable stiffness actuator for safe and fast physical human/robot interaction. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pages 526–531. IEEE, 2005.
- [35] Vaibhav V Unhelkar, Ho Chit Siu, and Julie A Shah. Comparative performance of human and mobile robotic assistants in collaborative fetch-and-deliver tasks. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, pages 82–89. ACM, 2014.