12-2018

Trust-Based Control of Robotic Manipulators in Collaborative Assembly in Manufacturing

Behzad Sadrfaridpour
Clemson University, bsadrfa@g.clemson.edu

Trust-Based Control of Robotic Manipulators in Collaborative Assembly in Manufacturing

---

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
Mechanical Engineering

---

by
Behzad Sadrfaridpour
December 2018

Accepted by:
Dr. Yue Wang, Committee Chair
Dr. Kapil Chalil Madathil
Dr. John R. Wagner
Dr. Ardalan Vahidi
Dr. Ian D. Walker

Abstract

Human-robot interaction (HRI) is vastly addressed in the field of automation and manufacturing. Most of the HRI literature in manufacturing explored physical human-robot interaction (pHRI) and invested in finding means for ensuring safety and optimized effort sharing amongst a team of humans and robots. The recent emergence of safe, lightweight, and human-friendly robots has opened a new realm for human-robot collaboration (HRC) in collaborative manufacturing. For such robots with the new HRI functionalities to interact closely and effectively with a human coworker, new human-centered controllers that integrate both physical and social interaction are demanded. Social human-robot interaction (sHRI) has been demonstrated in robots with affective abilities in education, social services, health care, and entertainment. Nonetheless, sHRI should not be limited only to those areas. In particular, we focus on human trust in robot as a basis of social interaction.

Human trust in robot and robot anthropomorphic features have high impacts on sHRI. Trust is one of the key factors in sHRI and a prerequisite for effective HRC. Trust characterizes the reliance and tendency of human in using robots. Factors within a robotic system (e.g. performance, reliability, or attribute), the task, and the surrounding environment can all impact the trust dynamically. Over-reliance or under-reliance might occur due to improper trust, which results in poor team collaboration, and hence higher task load and lower overall task performance.

The goal of this dissertation is to develop intelligent control algorithms for the manipulator robots that integrate both physical and social HRI factors in the collaborative manufacturing. First, the evolution of human trust in a collaborative robot model is identified and verified through a series of human-in-the-loop experiments. This model serves as a computational trust model estimating an objective criterion for the evolution of human trust in robot rather than estimating an individual's actual level of trust. Second, an HRI-based framework is developed for controlling the speed of a robot performing pick and place tasks. The impact of the consideration of the different level of

interaction in the robot controller on the overall efficiency and HRI criteria such as human perceived workload and trust and robot usability is studied using a series of human-in-the-loop experiments. Third, an HRI-based framework is developed for planning and controlling the robot motion in performing hand-over tasks to the human. Again, series of human-in-the-loop experimental studies are conducted to evaluate the impact of implementation of the frameworks on overall efficiency and HRI criteria such as human workload and trust and robot usability. Finally, another framework is proposed for the cooperative manipulation of a common object by a team of a human and a robot. This framework proposes a trust-based role allocation strategy for adjusting the proactive behavior of the robot performing a cooperative manipulation task in HRC scenarios. For the mentioned frameworks, the results of the experiments show that integrating HRI in the robot controller leads to a lower human workload while it maintains a threshold level of human trust in robot and does not degrade robot usability and efficiency.

Publications

- Behzad Sadrfaridpour and Yue Wang. Collaborative assembly in hybrid manufacturing cells: An integrated framework for human-robot interaction. In *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 3, pages 1178-1192, July 2018.

- Behzad Sadrfaridpour, Maziar Fooladi Mahani, Zhao Liao, and Yue Wang, Trust-Based Impedance Control Strategy for Human-Robot Cooperative Manipulation, ASME 2018 Dynamic Systems and Control Conference.

- Behzad Sadrfaridpour, Hamed Saiedi, and Yue Wang. An integrated framework for human-robot collaborative assembly in hybrid manufacturing cells. In *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 462–467, Aug 2016. IEEE-CASE Best Student Conference Paper Award Winner.

- Behzad Sadrfaridpour, Hamed Saeidi, Jenny Burke, Kapil Madathil, and Yue Wang. *Robust Intelligence and Trust in Autonomous Systems*, chapter 7. Modeling and Control of Trust in Human-Robot Collaborative Manufacturing, pages 115–141. Springer US, 2016.

- Hamed Saeidi, Foster McLane, Behzad Sadrfaridpour, Evan Sand, Sheng Fu, Julio Rodriguez, John R. Wagner, and Yue Wang. Trust-based mixed-initiative teleoperation of mobile robots. In *2016 American Control Conference*. IEEE, 2016.

- Sheng Fu, Hamed Saeidi, Evan Sand, Bahzad Sadrfaridpour, Julio Rodriguez, Yue Wang, and John Wagner. A haptic interface with adjustable feedback for unmanned aerial vehicles (uavs)-model, control, and test. In *American Control Conference (ACC), 2016*, pages 467–472. IEEE, 2016.

- S. M. Rahman Mizanoor, Behzad Sadrfaridpour, and Yue Wang. Trust-based optimal subtask allocation and model predictive control for human-robot collaborative assembly in manufacturing. In *ASME Dynamic Systems and Controls Conference*. American Society of Mechanical Engineers, 2015.

- Behzad Sadrfaridpour, Jenny Burke, and Yue Wang. Human and robot collaborative assembly manufacturing: Trust dynamics and control. In *RSS 2014 Workshop on Human-Robot Collaboration for Industrial Manufacturing*, July 2014.

- Behzad Sadrfaridpour, Hamed Saeidi, Jenny Burke, and Yue Wang. *Modeling and Control of Trust in Human-Robot Collaborative Manufacturing*. In *AAAI 2014 Spring Symposia*, March 2014.

Dedication

To my father, my mother and my brothers for their unconditional love and support.

Acknowledgments

I am forever grateful to my loving parents Hamid and Zahra and my wonderful brothers Amirhossein and Ehsan who have always inspired me through love, encouragement and support.

I wish to express my gratitude to my advisor Dr. Yue Wang who believed in me and supported me both financially and academically so I can reach this point as a PhD student.

I would also like to thank my committee members, Dr. Kapil Chalil Madathil, Dr. Ardalan Vahid, Dr. John Wagner, and Dr. Ian Walker for providing valuable insights during my studies as a PhD student, for serving as my committee members, and for their constructive comments and suggestions.

I am also grateful to the staff in Mechanical Engineering Department specially Ms. Trish Nigro and Ms. Gwen Dockins for their unfailing support and assistance.

Finally, I wish to express my gratitude to my friends who have supported me in different stages of my life.

This dissertation was supported in part by the National Science Foundation under Grant CMMI-1454139.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Research Motivation and Background

Conventional industrial robots have been designed for implementation inside safety peripheral equipment where only trained operators can interact with them through some external interfaces under running conditions [105]. However, the advent of light-weight and human-friendly collaborative robots (e.g. Baxter [26], UR10 [78], LBR iiwa [104], YuMi [1]) is changing manufacturing plants by more flexible and efficient robotic automation. The built-in safety features of these robots promise sound and close human-robot collaboration (HRC) in manufacturing environments. These developments improve human-robot interaction (HRI) to the extent that robots are perceived as social beings with which humans interact rather than simple tools [8]. For a light-weight, flexible, and human-friendly robot that has new HRI functionalities and interacts closely with a human co-worker, considering safety and production efficiency objectives may not suffice [108]. Social human-robot interaction (sHRI) has been demonstrated in robots with affective abilities such as Kismet, iCAT, Flobi, ERWIN, Kobian, NAO, Kamin, Ifbot, WE-3R III, Robokind, Geminoid [7, 116, 58, 77] in education, social services, healthcare, and entertainment [8]. Nonetheless, sHRI should not be limited only to those areas. Some new collaborative robots designed for manufacturing sites such as Baxter and Sawyer [25] have some social features that make the interaction more human-like and appealing. Some studies have explored how social behaviors of robots can impact human emotions. This topic is widely studied in the domain of human-computer interaction (HCI) as Affective Computing [127] which examines how interaction with an interface impacts the emotional state, feelings,

and satisfaction of the user [82]. The utilization of embodied conversation agents, human-friendly robots and facial expression are some examples of social capabilities that can be included in robotic systems for a closer human-like interaction expected by the human [7].

Human to robot trust is one of the key factors in sHRI and a prerequisite for effective HRC [30, 51]. Trust characterizes the reliance and tendency of human in using robots. Human trust can be categorized as dispositional and history-based [70]. The dispositional trust is similar to bias and defined as the initial trust an individual feels towards another being even without any interaction. The history-based trust is dynamic and is built based on the interaction. In this dissertation, we only consider the history based-trust which is dynamic and have high impacts on sHRI. Factors within a robotic system (e.g. performance, reliability, or attribute), the task, and the surrounding environment can all impact the trust dynamically [30]. Over-reliance or under-reliance might occur due to improper trust, which results in poor team collaboration, and hence higher task load and lower overall task performance [30]. Trust-based controllers demonstrate their capabilities in improving the interaction in teleoperation of mobile robots [98] and motion planning [63] scenarios in HRI.

The main objective of this dissertation is the integration of quantitative, unbiased and objective human to robot (and robot to human) computational trust models into the motion and/or force controllers of a collaborative robotic system with a manipulator in addition to the incorporation of pHRI and task performance criteria.

1.2 Contributions

This dissertation seeks to investigate and propose trust-based intelligent control frameworks for three typical tasks that require the cooperation of humans and robots and has application in flexible manufacturing. These tasks include the assembly task, robot to human handover, and, cooperative manipulation. These frameworks consider the task both from the theoretical and experimental point of view. Several objective measures such as total task completion time or robot average velocity and several subjective measures such as human perceived workload or trust are used for verification and evaluation of these frameworks. The contributions of this dissertation in modeling trust and developing trust-based controllers are as follows.

1.2.1 Human Trust in Robot

The first contribution of this dissertation is to provide a quantitative unbiased and objective measure for human to robot computational trust in assembly manufacturing task. Human-robot trust determines his/her acceptance and hence allocation of autonomy to a robot, which alter the overall task efficiency and human workload. In chapter 2, inspired by well-known human factors, a time-series trust model for human-robot collaboration tasks is developed. This trust model is a foundation for the trust-based controllers in this thesis. The major contributions of this chapter are as follows:

- A new dynamic, quantitative trust model specifically for HRC assembly manufacturing is proposed and experimentally validated.
- A neural network based robust intelligent scheme for autonomous robot speed control is developed.
- The quantitative trust models are integrated with robust intelligence for improved performance in HRC manufacturing.

1.2.2 Collaborative Assembly

Chapter 3 investigates the impacts of augmenting the combined pHRI and sHRI factors into robot controller on the joint performance of a human-robot team performing an assembly task. The chosen assembly task is usually accomplished by human workers and includes some repetitive physical movements for pick-and-place. A flexible robot can assist the human worker in doing such a task by bringing the required parts to the human worker [93, 94]. This framework allows the robot arm to select paths between the robot bin and the shared human-robot workspace based on trust evaluation and then move along the selected path while its translational velocity along the path is adjustable. The pHRI-based control condition involves prediction of human motion and synchronization of the robot motion progress with that of the human. Human trust in robot and robot emotional expressions will be considered as two main aspects of sHRI and devised in two integrated control conditions.

For this framework, a thorough statistical analysis for a set of robotic experiments with a human-in-the-loop is performed. The impacts of different control conditions on some of the well-

known HRI criteria including human perceived workload, human trust in robot, robot usability as well as objective measures in terms of robot average velocity and assembly time are evaluated and compared. The main contributions of this chapter are summarized as follows:

- A novel framework that considers pHRI and sHRI for human robot manipulation is proposed.
- Thorough statistical analysis for a set of robotic experiments with a human-in-the-loop is performed. The impacts of different control conditions on some of the well-known HRI criteria including human perceived workload, human trust in robot, robot usability as well as objective measures in terms of robot average velocity and assembly time are evaluated and compared. More details are provided in Section 3.5 and Section 3.6.
- Both human-to-robot trust and dynamic robot emotions are integrated into the framework for a more comprehensive consideration of sHRI factors. More details are in Section 3.4.4 and Section 3.4.5.
- A trust-based robot path selection strategy is devised such that if human trust in robot drops below some threshold value, the robot chooses a safer path with less chance of collision. More details are in Section 3.3.1.
- To increase transparency of HRI, an HCI is designed to show the HRC system variables and robot emotion. More details are in Section 3.4.7.

1.2.3 Robot-Human Handover

Chapter 4 proposes a trust-triggered motion planning strategy for the robot-human handovers of payloads during the collaborative assembly. More specifically, based on robot trust in human, robots handover configuration and motion of reducing are varied via kinematic redundancy to reduce potential impact forces on the human during the handover. A hybrid assembly cell is developed for a typical collaborative task. The trust-based collaborative assembly task including the trust-triggered handover is evaluated based on a comprehensive evaluation scheme. The results show that the inclusion of robot trust and trust-triggered handover improve the effectiveness in human-robot interaction and task performance through increasing safety, handover success rate, team fluency, human trust in robot, and assembly efficiency and reducing cognitive workload, with a small sacrifice in handover efficiency.

1.2.4 Cooperative Manipulation

Chapter 5 investigates human-robot cooperative manipulation. Cooperative manipulation refers to joint coordination of two or more robots handling a common object. This concept can address the typical limitations of single-arm robots in terms of dexterity and payload and open up new applications in flexible manufacturing systems and service robotics. In human-robot cooperative manipulation, a team of humans and robots coordinate together to handle a common object. Two major approaches are available for human-robot cooperative manipulation: (i) reactive or complaint approach in which the human in the leader and the robot is compliant and follows the force applied by the human to the object and (ii) proactive approach in which the robot reduces the human effort by estimating the human desired motion and force and acting as a collaborator rather than a simple follower. This dissertation proposes a trust-based control policy that balances between the human effort and the disagreement between the human and robot by dynamic or switching role allocation based on the proactive and reactive behavior of the robot.

1.3 Dissertation Outline

The next chapters of this dissertation are organized as follows. Chapter 2 presents a study on modeling trust in HRC in manufacturing. Chapter 3 presents a framework for integration of HRI factors into the robot motion controller for human-robot collaborative assembly tasks in a manufacturing hybrid cell. Chapter 4 presents a framework for trust-based handover strategy. Chapter 5 proposes a trust-based strategy for human-robot collaborative manipulation. Chapter 6 discusses the conclusions of this dissertation.

Chapter 2

Modeling Trust in Human-Robot Collaboration in Manufacturing

2.1 Introduction

Human-robot trust determines his/her acceptance and hence allocation of autonomy to a robot, which alter the overall task efficiency and human workload. Inspired by well-known human factors research, we develop a time-series trust model for human-robot collaboration tasks, which is a function of prior trust, robot performance, and human performance. The robot performance is evaluated by its flexibility to keep pace with the human coworker and is molded as the difference between human and robot speed. The human performance in doing physical tasks is directly related to his/her muscle fatigue level. We use the muscle fatigue and recovery dynamics to capture the fatigue level of the human body when performing repetitive kinesthetic tasks, which are typical types of human motions in manufacturing. The robot speed can be controlled in three different modes: manually by the human worker, autonomously through robust intelligence algorithms, or collaboratively by the combination of manual and autonomous inputs. We first simulate a typical 9-hour work day for human-robot collaborative tasks and implement the proposed trust model and the three control schemes. Furthermore, we experimentally validate our model and control schemes by conducting a series of human-in-the-loop experiments using the Rethink Robotics Baxter robot.

This chapter considers hybrid manufacturing systems [43] in which a human worker and a

peer human-friendly robot (for example, Rethink Robotics Baxter [89], KUKA LBR iiwa [5], and Universal Robots UR5 and UR10 [78]) collaborate with each other to fabricate customized products [23, 105] in the same workspace at the same time. For instance, a skilled human worker can collaborate with a lightweight, flexible, and human-friendly robot to perform an assembly operation. In such applications, human's capability in performing highly skilled tasks such as assembly is combined with the advantages of robots such as precision, performance consistency in performing repetitive jobs, data processing, sensor, and actuator based assistance [43]. The resulting collaboration between human and robot in production cells [112] is expected to lead to high productivity, flexibility, and safety, as well as balanced human working experience. However, improper HRC may cause counter effects such as misuse of a machine and/or safety issues and hence there arises a need for investigating HRC in advanced manufacturing [42]. There are potentially many issues worth addressing, but this chapter focuses on human-robot trust as a critical element in HRC manufacturing because trust will directly affect the degree of autonomy that a human delegates to the industrial robot, which determines the efficiency as well as quality of the manufacturing processes. We adopt the concept of trust among humans to study HRC in manufacturing automation [52]. Thus we investigate empirical as well as theoretical studies to utilize trust analysis [52] in HRC manufacturing. There exist two types of trust related to the automation use among different individuals, i.e. dispositional trust and history-based trust [70]. Dispositional trust reflects trust in other persons (or machines) upon initially encountering them, even if no interaction has yet taken place. In contrast, history-based trust is founded on interactions between the person and another person or machine. Due to the dynamic nature of HRC, this chapter studies the history-based trust. Several works have developed mathematical models for trust [73, 53, 37, 22]. In our previous works, inspired by Lee and Moray's (1992) trust study for an automated juice plant [49], we used a model for human-robot trust in HRC manufacturing tasks and showed examples of changing robot performance based on human's trust [92, 90]. In this chapter, we describe a time-series model of human-robot trust for real-time control allocation in HRC manufacturing tasks, a model of robot performance that ties speed to flexibility, a model of human performance that includes muscle fatigue, and a series of experimental validations to capture the impact of performance on trust within the HRC system [91]. The proposed dynamic trust model is a function of prior trust, change of robot performance, and change of human performance, as well as fault occurrence.

The robot performance can be described in terms of reliability, flexibility, dexterity, etc.

Because robot reliability is almost always guaranteed in manufacturing applications, here we will focus on understanding and improving the robot flexibility assuming the robot is reliable. Flexibility is required for factory environments with frequent changes, varying positions of transport containers, and various uses of machine tools. Flexibility is envisioned to increase productivity and humanization of the work place [109]. In fact, it is one of the advancements brought by the new generation of manufacturing robots and is achievable via instructable or adaptable robots. To model the performance of a human worker of doing a repetitive kinesthetic task, which is typical in manufacturing tasks, we adopt the muscle fatigue and recovery model [59, 60, 55, 19]. This model shows how the performance of the human worker changes as his/her muscles gradually get tired or recovered.

Artificial neural networks (ANNs) are powerful tools that can be used for realizing artificial intelligence [122]. They have been widely applied in the aviation industry, business, financial forecasting, control systems, security systems, etc. [123]. Neural networks are capable of function approximation, pattern recognition, and nonlinear mapping [69]. Their learning ability and adaptability also introduce robustness to a tool [15]. In this chapter, we are interested in the applications of neural networks in intelligent control such as black box model identification, adaptive inverse control, and model predictive control [29]. More specifically, we will use neural networks to learn the desired pattern of robot speeds in order to collaborate with a specific human worker and to use the result for autonomous adjustments of the robot's speed.

Next, we design control allocation schemes to switch between manual and autonomous modes in order to increase the human-robot trust. To do so, three approaches are designed. One way is to increase or decrease the robot performance exclusively based on manual inputs. Another way is to predict the human requests and autonomously adjust the robot performance using the neural network-based intelligent control. The last way is to use a collaborative control scheme to adjust the robot performance using both autonomous and manual inputs.

To study the trust evolution and human working pattern during HRC manufacturing, we present both a numerical example and a set of experimental validations. The numerical example is simulated for a typical 9-hour workday starting at 8 AM. The exclusively manual, exclusively autonomous, and collaborative control modes are compared. The experiments are designed as HRC assembly tasks where the robot picks the parts and places them in front of the participant and the participant assembles these parts. Such collaborations require the robot to keep pace with the human and can be applied in many manufacturing processes to partially automate the assembly

tasks.

The rest of the chapter is organized as follows. Section 2.2 introduces the time-series trust model, and robot and human performance models. Sections 2.3 develops the neural network based robust intelligence control algorithm for learning the human working pattern in controlling the robot speed. Section 2.4 discusses three control allocation schemes, i.e. exclusively manual, exclusively autonomous, and collaborative control of the robot speed. We simulate the proposed trust model and an intelligent control scheme using a numerical example of a typical work day in a manufacturing plant in Sect. 2.5. A set of experimental validations on assembly tasks are performed and major results analyzed in Sect. 2.6. We conclude the chapter in Sect. 3.7.

2.2 Trust Model

2.2.1 Time-Series Trust Model for Dynamic HRC Manufacturing

Based on Lee and Moray's (1992) time-series trust model and the more recent meta-analysis [30] and survey [33], a human's trust in the robot depends on the robot performance, human performance, and fault occurrences. In this section, we introduce a time-series dynamic model of human-robot trust for HRC manufacturing based on these results from human factors research. To clarify the manufacturing application, let us start with an example. Consider the case when a skilled human worker collaborates with a flexible robot on a product, such as inserting screws into parts or welding, in a hybrid cell. The robot picks up a part and then holds it still in specific positions and orientations near the human worker so that he/she can focus on the assembly operations. As the working speed of the human worker varies during the working hours, a constant speed of the robot will cause trust degradation of the human worker when he/she feels that the robot is working faster or slower than what he/she expects, i.e. the robot lacks the flexibility to keep the same pace as the human worker. This discrepancy indicates the robot's inflexibility. To recover trust, the robot speed should be adjustable so that the human worker feels more comfortable in the collaboration. Moreover, the human worker's performance has an influence on his/her trust in the robot. For example, due to physical and/or mental fatigue resulting from continuous work during a day, the human worker may tend to rely more on the automation and thus his/her trust in the robot increases. With this mindset, we propose the following time-series model for the dynamics of

human-robot trust

$$\begin{aligned} T(k) = AT(k-1) \quad &+ \quad B_1 P_R(k) + B_2 P_R(k-1) + C_1 P_H(k) + C_2 P_H(k-1) \\ &+ \quad D_1 F(k) + D_2 F(k-1), \end{aligned} \tag{2.1}$$

where P_R, P_H, and F are robot performance, human performance, and fault, respectively. We use k to indicate the time step. The coefficients A, B_1, B_2, C_1, C_2, D_1, and D_2 are constants to be determined through experiments. Note that we seek to obtain a computational model of a human's trust for HRC in assembly lines in general. In practice, these parameters of the trust model can be tuned for different individuals to fit their subjective trust to some extent. Moreover, similar to [49] we assume that the trust dynamics follow a lag model and there are some delays before changes of trust. As long as there is a considerable difference between the human and robot working speeds, the robot performance (P_R, flexibility) will decrease regardless of which speed is greater than the other. Therefore, the trust value decreases accordingly. In contrast, if there is no considerable decrease in robot flexibility over time, the trust will increase. We design robust intelligent control schemes to increase human trust in a robot as described in Sect. 2.3. To obtain the trust model (2.1), we need to develop the robot and human performance models as discussed in the subsequent sections.

2.2.2 Robot Performance Model

In manufacturing, machine reliability is almost always guaranteed in order to avoid huge loss under even small malfunctions. Meanwhile, for the new type of flexible manufacturing tasks, the robot needs to seamlessly collaborate with the human coworker. Hence, robot performance, in this case, can be evaluated by its flexibility in accommodating a human's work behavior. In our study, we consider especially the robot capability in adjusting its speed so as to keep the same pace as the human worker. Hence, the difference between human and robot speed will determine the robot flexibility. We denote robot working speed, $V_R \in [0, 1]$, as the normalized speed of the robot for doing a specific task where "0" represents the situation when the robot stops working, and "1" represents the situation when the robot works at its maximum speed. We denote the human working speed, V_H, correspondingly. Note that both V_H and V_R are defined as normalized non-dimensional

numbers in $[0, 1]$. Based on our definition of the robot flexibility, P_R, we can write:

$$P_R(k) = P_{R,max} - |V_H(k) - V_R(k)| . \tag{2.2}$$

Since we use the normalized values of V_H and V_R, we have $P_{R,max} = 1$ and hence P_R is always bounded between $[0, 1]$. In the ideal case when the robot works at its highest flexibility in adapting to the human worker's speed, the speed difference is minimum and $P_R = 1$. In the worst case when the robot is fully incapable of adjusting to the human worker's speed, the speed difference is maximum and $P_R = 0$.

2.2.3 Human Performance Model

A human's performance in physical tasks such as assembly manufacturing depends on his/her state of muscle fatigue or recovery. In such scenarios, a human worker usually performs repetitive kinesthetic tasks. We adopt the muscle fatigue and recovery model proposed in [60] and [19] for our human performance model. This model explains how a muscle or group of muscles get fatigued or recovered during performing physical tasks and shows how the performance of a human worker changes as his/her muscles gradually get tired or recovered. We assume that the higher the fatigue level is, the lower the performance would be. The maximum human performance occurs at the situation when he/she is not subjected to any fatigue, and the minimum value when he/she is experiencing the maximum level of fatigue. We first present the muscle fatigue and recovery model and then develop the human performance model based on the muscle fatigue and recovery model.

For the modeling of muscle fatigue and recovery, we introduce a model for isometric force generation, i.e. when the muscles do not move but they apply force. When a muscle applies some force for an amount of time, the maximum isometric force that one can produce, $F_{max,iso}(k)$, decreases. The dynamic model of fatigue for $F_{max,iso}(k)$ is a function of time, the initial maximum isometric force one can generate at rest, called Maximum Voluntary Contraction (MVC), and real-time applied force $F(k)$ [59]. On the other hand, when the muscle does not apply any force, it gets recovered. The recovery process is also a function of the time and MVC [60]. Based on [55], when the muscle fibers work, some of them become fatigued and some recover. That is to say, fatigue and recovery occur simultaneously [60]. We develop the discretized version of the combined fatigue and

recovery model in [19] using the first-order Euler approximation

$$F_{max,iso}(k) = F_{max,iso}(k-1) - C_f F_{max,iso}(k-1)\frac{F(k-1)}{MVC} + C_r(MVC - F_{max,iso}(k-1)), \quad (2.3)$$

where C_f is the fatigue constant and C_r is the recovery constant. Both C_f and C_r are individual-specific. Equation (2.3) is for isometric muscle contraction and has an equilibrium point at which the fatigue and recovery balance out. This point is the lowest limit (threshold) of the $F_{max,iso}(k)$. This threshold force, F_{th}, can be calculated by assuming that $F_{max,iso}(k) = F_{max,iso}(k-1)$ at the threshold:

$$F_{th} = MVC\frac{C_r}{2C_f}(-1 + \sqrt{1 + \frac{4C_f}{C_r}}). \quad (2.4)$$

Theoretically, at the threshold force, the fatigue and recovery occur at the same rate and one can generate this threshold force for a long time. Since the fatigue and recovery model predicts the human muscle status related to workload, this model can be used to measure the physical performance of a human worker during manufacturing tasks. Hence, we propose the following performance model for human, P_H

$$P_H(k) = \frac{F_{max,iso}(k) - F_{th}}{MVC - F_{th}}. \quad (2.5)$$

Note that in Equation (2.5), $F_{max,iso}$ varies between the minimum value F_{th} and the maximum value MVC, therefore it is a normalized value between 0 and 1. The maximum value MVC, is assumed when the human worker starts the task, i.e. $F_{iso,max}(k=0) = MVC$.

Remark 1. The threshold force, F_{th}, is the minimum value of $F_{max,iso}$. Hence, the forces below F_{th} are not theoretically achievable.

2.3 Neural Network Based Robust Intelligent Controller

The goal of using a neural network in this problem is to design a robust intelligent controller for adjusting the robot speed autonomously during the work cycle which is a black box model identification. This controller is designed so that it reduces the human worker's workload for adjusting the speed of the robot manually. To do so, a neural network with a proper method of training

and also some training data are required. One way of training the neural network is to mimic the behavior of the human worker in adjusting the robot speed manually, which can be regarded as the desired pattern for the robot flexibility when collaborating with the human worker. We performed human-in-the-loop experiments to collect the training data. In this data set, the current robot speed, human speed, and current work-cycle time index is used as the input to the neural network and the estimation of robot speed at the next cycle is the output.

The structure of the neural network used in this chapter is illustrated in Fig. 2.1. This network consists of an input layer, a hidden layer, and an output layer of neurons which form a Perceptron artificial neural network [29]. This type of neural network has the capability of approximating many nonlinear functions. The additional input "1" (as seen in the first and second layers of Fig. 2.1) represents the effect of bias in the neural network. Using bias increases the learning capability of a neural network by providing an additional degree of freedom through an adjustable offset. We utilize two different activation functions for the hidden layer and the output layer, respectively. The activation functions determine the output of the neurons in each layer as a function of the weighted sum of the inputs to that layer. The activation function of the hidden layer $\mathbf{y}$ is a tangent sigmoid function as follows

$$\text{tansig}(x_{py}) = \frac{e^{x_{py}} - e^{-x_{py}}}{e^{x_{py}} + e^{-x_{py}}}, \tag{2.6}$$

where x_{py} is the input for the tangent sigmoid function. In the neural network shown in Fig. 2.1, this variable is defined as $x_{py} = W_{py} \times [\mathbf{p}\ 1]$ where W_{py} represents the weights of the neural network that connect the input layer $\mathbf{p}$ (i.e. the current robot speed, human speed, and current work-cycle time index) to the hidden layer $\mathbf{y} = \text{tansig}(x_{py})$. The output of this function is in $(-1, 1)$ region which produces the inputs to the next (output) layer. The activation function for the output layer $\mathbf{o}$ is chosen to be the linear function according to the following

$$\text{purelin}(x_{yo}) = x_{yo}, \tag{2.7}$$

where $x_{yo} = W_{yo} \times [\mathbf{y}\ 1]$ for the output layer are the weights of the neural network that connect the hidden layer to the output layer. This layer determines the robot speed at the next work cycle. Once enough data are collected, the Levenberg-Marquardt Backpropagation training algorithm [29] is used

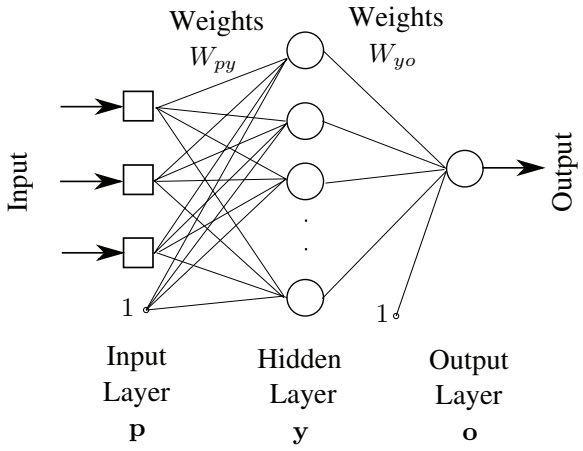


Figure 2.1: The structure of the neural network used for learning the robot speed.

to train the neural network. This algorithm is a gradient descent based optimization algorithm for minimizing the mean square estimation error of the neural network. It can be used for training either single or multi-layer neural networks. A well-trained neural network is able to do a nonlinear mapping from the input data set to the output data set.

2.4 Control Approaches

We design control allocation schemes to switch between manual and autonomous modes in order to increase human-robot trust. Since the speed of a human worker changes during the working shift, his/her expectation from the partner robot changes over time accordingly. Therefore, the human-robot trust can be increased by adjusting the robot speed according to what the operator desires. To do so, three approaches are available: (i) Increasing or decreasing the robot speed based on manual corrective requests that the human worker sends to the robot controller; (ii) Predicting the human requests at different moments and autonomously adjusting the robot performance without sending any corrective request; or (iii) Using a collaborative control scheme to adjust the robot speed using the autonomous control and manual inputs interchangeably. The prediction approach can be achieved through the robust intelligence algorithm which seeks to learn the pattern of human requests as he/she collaborates with the robot over time. Here we use the artificial neural networks as the robust intelligence algorithm as discussed in Sect. 2.3. In the collaborative mode, the robust intelligence algorithm is used to autonomously control the robot speed by default. However, the human worker can adjust the robot speed at the times when the robust intelligence fails to mimic

the human pattern in adjusting the robot performance. We now explain the details of implementation of the three different approaches for adjusting the robot speed.

2.4.1 Manual Mode

For the manual mode, a human-sensitivity based approach is adopted to predict how the human coworker adjusts the robot speed. Most of the time, the robot speed does not match the human working speed exactly. However, it is only when the difference between these two speeds exceeds a certain threshold, then the human worker would feel the significance and send some corrective commands to change the robot speed. Let this threshold be human sensitivity, H_S. With this setting, the robot speed at the next time step is adjusted by the human worker as follows

$$V_R(k+1) = V_{RH}(k), \tag{2.8}$$

where $V_{RH}(k)$ represents the manual control input whenever the human worker changes the robot speed. Other than these moments, we have $V_R(k+1) = V_R(k)$.

2.4.2 Autonomous Mode

Based on the explanations in Sect. 2.3, to train the artificial neural network, we collect data on how a human worker sends commands to the robot in the manual mode for some period of time. There are different ways to construct the neural network based on the inputs and the training algorithm. For example, we can predict the pattern of the speed commands that the human worker sends to the robot only based on time parameters or we can include other parameters in the network as well. Figure 2.1 shows the neural network with the current time, human speed, and robot speed as inputs. The output is the robot speed at the next time step. After training the neural network, it will predict the desirable robot speed based on the inputs. With this setting we have

$$V_R(k+1) = V_{RI}(k), \tag{2.9}$$

where $V_{RI}(k)$ represents the autonomous control input calculated by the neural network for the next time step. The neural network is the only source of robot speed adjustment in this mode, and thus it is used at each time step whether it generates a new command or the similar command as the

previous step.

2.4.3 Collaborative Mode

The autonomous mode reduces the human workload through the use of robust intelligence algorithms. However, the manual mode offers more accurate control over the robot speed. In the collaborative mode, we combine both advantages. The robot speed is controlled autonomously by the neural network by default and the human worker can change the robot speed whenever he/she wants to. Therefore, we can describe the process of controlling the robot speed by the following equation

$$V_R(k+1) = \sigma(k)V_{RH}(k) + (1-\sigma(k))V_{RI}(k), \tag{2.10}$$

where $V_{RH}(k)$ and $V_{RI}(k)$ are as in Eqs. (2.8)-(2.9) respectively, and $\sigma(k)$ is the activation mode

$$\sigma(k) = \begin{cases} 1 & \text{manual control} \\ 0 & \text{autonomous control} \end{cases}$$

In this setting, the robot speed at the next time step is determined either directly by the human commands or the predictions of the robust intelligence algorithms. Examples of utilizing this scheme will be presented in Sects. 2.5 and 2.6.

2.5 Simulation

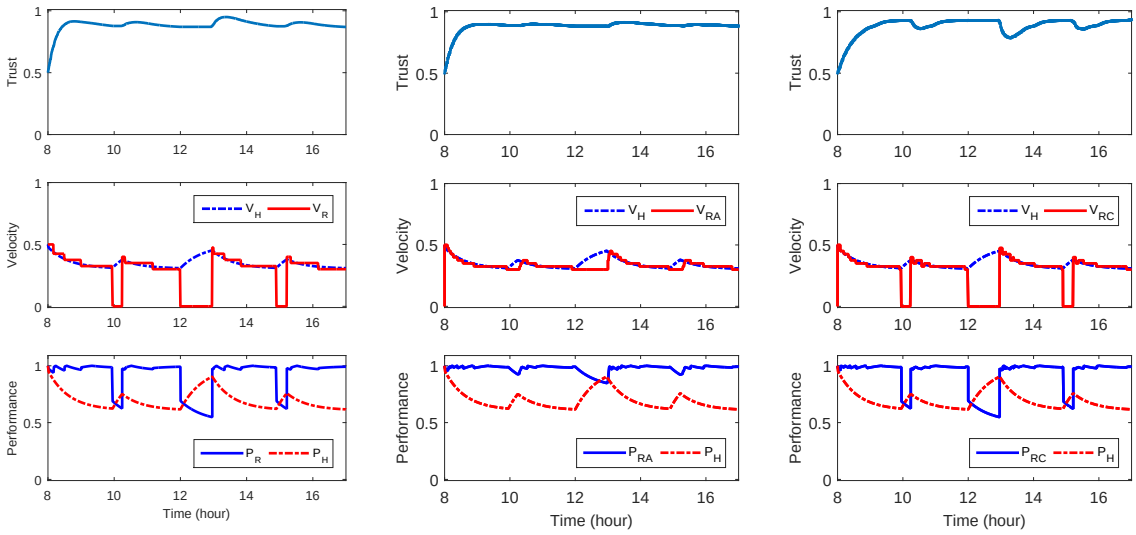
In this section, we present a numerical example using MATLAB R2014a software for three different control schemes described in previous sections. This example shows (i) how the human trust evolves according to the human and the robot performances; and (ii) how the control workload of the human worker changes. The human performance dynamics (2.5) described in Sect. 2.2.3 are simulated for a typical 9-hour workday starting at 8 AM. In the simulation we shift the time origin to 8, i.e. we use $k' = k - 8$ instead of k in all of the equations. For a fixed repetitive task we assume that the external force applied by the human worker is constant. Moreover, the human workers do not need to apply their full strength (MVC) to finish the manufacturing tasks. Therefore, we use a constant value for the external force, i.e., $F(k) = \frac{MVC}{4}$. The maximum value for both human

and robot performance is 1, i.e. $P_{H,max} = 1$ and $P_{R,max} = 1$. The human worker is assumed to start with P_H between $[0.95, 1]$. The human worker working speed, V_H is set to be half of his/her performance value, i.e. $V_H = \frac{1}{2}P_H$ in the simulation. The robot is set to start with half of the maximum robot working speed, $\frac{1}{2}V_{R,max}$. We also assume that initial trust of the human worker is the half of its maximum value. In all the simulation modes, we assume that the human worker works according to the following pattern. He/She starts to work at 8 AM and ends at 5 PM. There is an approximately one-hour lunch break around noon. There are also two short breaks (15 to 20 minutes) in mid-morning and mid-afternoon (around 10 AM and 3 PM, respectively). During such a workday, based on the Eq. (2.5) the human performance decreases from the beginning of the day through the end of the day, except for the break times and the lunch time when the human performance recovers. We simulate the three control methods in Sect. 2.4.

Based on the explanations in Sect. 2.3, to train the neural network, we simulate and collect the corresponding data for the human-robot interaction of a particular human worker for a period of 4 months. According to the data, as in Fig. 2.1, we have 3 inputs to the artificial neural network, namely month, day and time of the day, and one output which is the performance of the robot. The number of hidden layer neurons is chosen to be 10 and the Error Backpropagation training algorithm is used to train the neural network. The results for each of the three control schemes are presented in the next subsections.

2.5.1 Manual Mode

According to the explanations in Sect. 2.4.1, we set the human sensitivity as $H_S = 0.05$. The results of this simulation are shown in Fig. 2.2(a). As can be seen in this figure, at the start of the day both human and robot start fresh with high working speeds and consequently the robot performance is high. As time passes, the working speed of the human worker decreases but the robot working speed does not change, so the difference between the human and the robot speed increases and thus the robot performance decreases. The human performance also decreases during this time. Although both robot and human performance decrease, since they have high values the trust increases before 9 AM. The trust value decreases slightly when the human performance declines after 9 AM. Therefore, when the human speed decreases during the time interval 8 AM to 10 AM, the human worker sends corrective commands to decrease the robot speed. After that, the human worker takes a break and his/her speed increases when going back to work again. We use the same



(a) manual mode (b) autonomous mode (c) collaborative mode

Figure 2.2: Evolutions of human speed V_H, robot speed V_{RA}, human performance P_H, robot performance P_{RA}, and trust T in (a) manual mode, (b) autonomous mode, and (c) collaborative mode.

trend for the rest of the day with breaks at 12 PM and 3 PM, respectively. The trust value does not change during the breaks.

2.5.2 Autonomous Mode

According to the explanations in Sect. 2.4.2, we use the neural network for adjusting the robot performance autonomously. The results of this simulation are shown in Fig. 2.2(b). As shown in this figure, the autonomous mode can adjust the robot speed properly most of the times. For the autonomous mode, the trust level has a similar trend as in the manual mode except for the end of the break times, where the neural network cannot predict the desired robot speed accurately. This leads to a sudden momentary drop of trust due to a temporary difference between the human and robot speed.

2.5.3 Collaborative Mode

For simulation of this mode, we use the same configuration of the manual and autonomous control modes described in this section. We then combine them as described in Sect. 2.4.3 to simulate the collaborative mode. The results are shown in Fig. 2.2(c). The team starts to work in

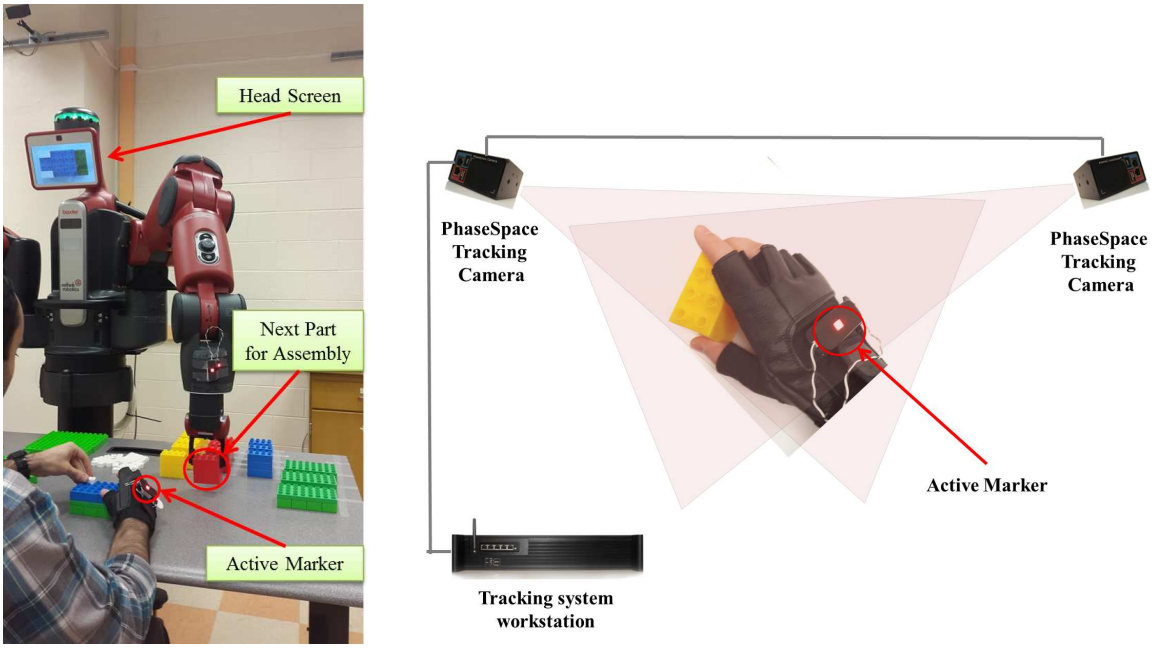
the autonomous mode at the beginning of the workday. After some time, if the robot speed does not match the human speed, the level of trust decreases. Moreover, if the robot performance is high and the human performance declines, the level of trust increases. In contrast to the autonomous mode, except for autonomous adjustment, the human worker can also switch to the manual mode by sending corrective commands. Note that the human worker sends commands whenever he/she feels that the autonomous adjustments are not correct. If the system switches back to the autonomous mode right after the manual correction, the adjustments might not be correct and hence the human worker needs to adjust the robot speed again. This leads to frequent switches back and forth between the manual and autonomous mode. To prevent such problems, once the manual mode is activated, it will be kept for a fixed time period (5 minutes) before it is allowed to switch back to the autonomous mode. After that, the system switches back to the autonomous mode and remains in the autonomous mode if no corrective commands are sent.

2.5.4 Comparison of Control Schemes

We can measure the human control workload under the manual, autonomous, and collaborative mode, respectively. The control workload for the manual mode is 100% since the human worker always changes the robot velocity by him/herself. The control workload under the autonomous mode is 0% since the human worker does not change the robot speed at all. The amount of control workload for the collaborative mode depends on the amount of time when the manual mode is activated. In our example, this value is 61.4%. We can also compare the average value of trust under these three modes. In the autonomous mode, the average trust value is 0.8803 which is lower than this value in manual mode, 0.8825. The average trust value in collaborative mode is 0.8816. This shows that using the collaborative mode, we can increase the trust compared to the autonomous mode while the control workload is smaller than the manual mode.

2.6 Experimental Validation

In this section, we provide the detailed description of our experiments to validate the quantitative trust model (2.1) and the effectiveness of the proposed control schemes. We will measure the overall task performance of the collaborative control scheme versus exclusively manual and autonomous control as well as the difference in human workload.



(a) (b)

Figure 2.3: The experiment setup: (a) collaboration of a participant and Baxter, and (b) PhaseSpace tracking system for tracking participants' hand motion.

2.6.1 Experimental Test Bed

As shown in Fig. 2.3(a), we employ a humanoid manufacturing research robot Baxter made by Rethink Robotics [27] to collaborate with the participant. The robot has two arms. Each arm provides 7 degrees of freedom. The arm joints are compliant as they are built with back-drivable motors and compliant actuators. The robot has a rotary screen at its head where informative messages or affective expressions can be displayed. It has a moveable base. The robot control program is coded in Python language and is interfaced with the robot hardware through ROS software. Baxter is very suitable for light-weight material handling and intelligent assembly, testing and sorting, and especially for small batch productions. We use the Impulse X2 motion tracking system from PhaseSpace to track the human hand for speed measurement (as shown in Fig. 2.3(b)). The tracking system includes 8 cameras, a set of active markers and a workstation for tracking rigid bodies in a 3D environment. The workstation combines the data from the cameras, which track the active markers mounted on an object (for example, a participant's hand in this study), to calculate its 3D position. The resulting position and timing information is sent to a client machine to calculate the hand motion speed.

2.6.2 Experimental Design

The experiment resembles the task that a human worker performs in the manufacturing assembly lines. In such an environment, human workers are required to perform a series of assembly tasks within a fixed period of time. For making a final product, different components need to be assembled together. Each of these components needs to be assembled by different parts as well. This procedure of component assembly is called subassembly, which is common in the airplane and automobile assembly and usually done by the human workers manually. We will consider such a subassembly task in our experiments. In such tasks, the parts need to be assembled are usually stacked near the workbench of the human worker. The human worker picks these parts and assembles them. If the component is customized, there will be a variation of choice for some of the parts. These customized parts can be delivered to the human worker by means of automatic delivery systems such as belt feeders. Once the component is assembled, it needs to be mounted on the final product. The experimental setup of this study is very similar to these tasks in a real assembly line except that there is a humanoid robot (Rethink Robotics Baxter) that collaborates with the participant. Within this collaboration, the robot helps the participant by picking up and placing the customized parts needed for the assembly task while the human worker performs tasks that robots are not capable of, e.g. assembling these parts together. The details of the experiment scenario are as follows.

2.6.2.1 Experiment Scenario

The participant is asked to perform a cooperative assembly task with Baxter within a fixed period of time. For each experiment condition, the task is assembling 10 components within 17 minutes (102 seconds per task cycle). Figure 2.3(a) shows the collaboration of a participant and Baxter. The task is to assemble a customized component (e.g. component G in Fig. 2.4) made from different parts (Lego bricks, e.g. bricks A, B, C, D, E, F in Fig. 2.4) and mount it to another component (here is another larger Lego brick, e.g. component I in Fig .2.4). The example assembly task we consider here can be found commonly in automobile and airplane assembly, e.g. center console subassembly and airplane wing spar assembly. There are 10 trials in total in each trial. The participant and Baxter share meme workspace on a table and the assembly parts are placed at different regions on the table as shown in Fig. 2.4. In this figure, the Lego bricks that need to be assembled together are A, B, C, and D. At the beginning of each task cycle, Baxter picks up a

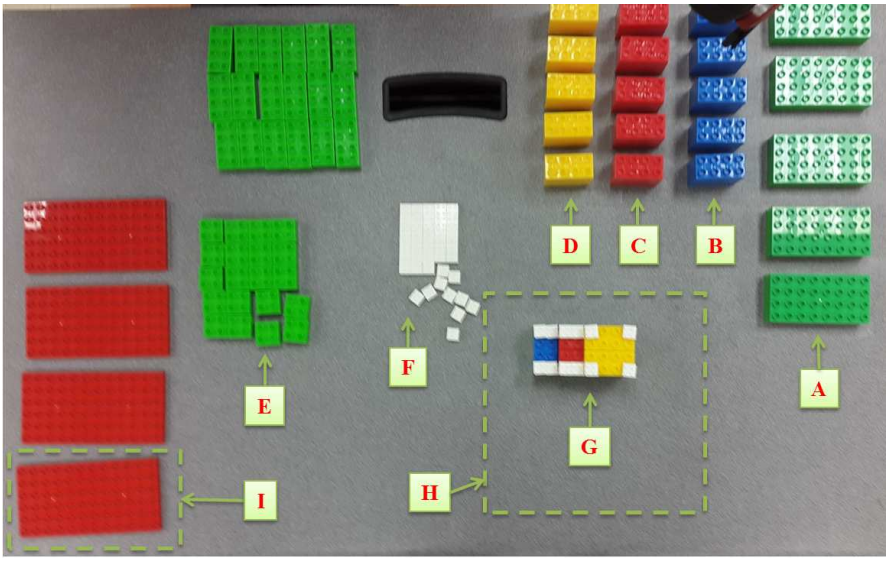


Figure 2.4: Different assembly parts and regions on the experiment table.

required part (brick A) and places it in front of the participant (region H) and displays a picture of the assembled part via its head screen (Fig. 2.3(a)). The participant is required to look at Baxter's screen and assemble the part exactly as appeared on it. The participant is also required to add fitting parts (bricks E and F in Figure 2.4) on top of the assembled Lego bricks similar to tightening screws or bolts in real manufacturing. When the participant finishes assembling the last part, he/she is required to pick and mount the whole component to another Lego brick located at the other side of the table (component I in Figure 2.4). Meanwhile, Baxter picks and places the next part in front of the participant and displays the next picture of the assembled part. The similar process is repeated until Baxter picks and places the last required part in front of the participant. Figure 2.5 shows the instruction pictures that Baxter shows to the participant in each cycle. Each of these pictures shows the correct assembly of current Lego bricks and corresponding fitting parts (needed to be mounted on the top of the Lego bricks). F and E are the fitting parts for assembling and mounting, respectively. Figure 2.6 provides a flowchart to summarize the required actions for both Baxter and the participant and their collaboration in every task cycle.

2.6.2.2 Controlled Behavioral Study

To understand the impact of the robot and human performance on the trust evolution, a 2 (robot performance - low flexibility, high flexibility) $\times$ 2 (human performance - non-fatigue, fatigue)

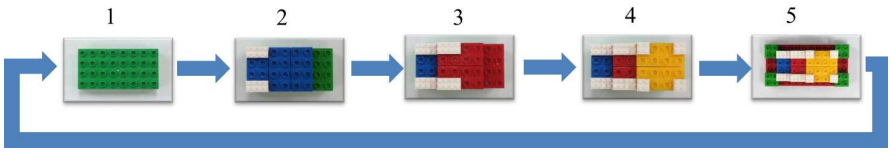


Figure 2.5: Sequence of assembly parts that Baxter shows to the participant as instruction via its head screen.

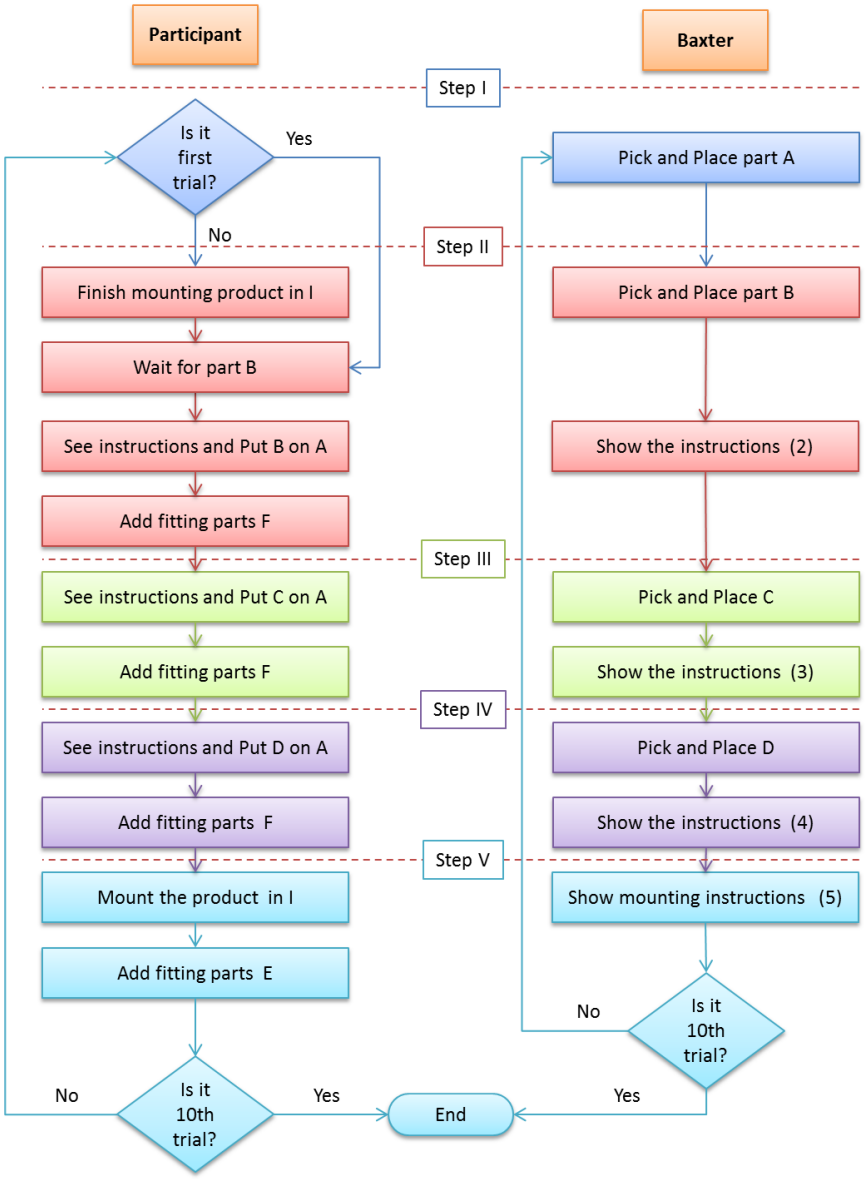


Figure 2.6: Task flowchart of one cycle of the human-robot collaborative assembly task.

mixed experimental design is employed under each control mode. In the high robot performance condition, the robot speed changes in accordance with participant's hand speed without any delay while in the low robot performance condition, the robot speed changes with some random delay plus some sudden stops of the robot. Note that the sudden stops of the robots are the faults of the robot while the random delays are the inflexibility of the robot. Here human fatigue refers to the psychically caused fatigue that commonly occurs in an assembly human worker as discussed in Sect. 2.2.3.

2.6.2.3 Imposing Fatigue

Assembly tasks usually require prolonged low-level repetitive work of the human workers which causes psychical fatigue. However, in the laboratory setting, it is difficult for a participant to perform a long 9-hour experiment to study the fatigue condition. It has been shown in [36] that the greatest effort level of shoulder muscle is required when the human worker holds a typical hand tool weighting around $15-20N$ in abducted shoulder posture (90 o vertical). A similar method as in [36] is used to impose fatigue in the experiments. In the fatigue condition, the participant is asked to warm up and then perform 10 minutes of exercises. Before doing the exercises, we need to measure the MVC as shown in Equation (2.3) in Sect. 2.2.3. The MVC level for 90 houlder posture for the dominant hand shoulder muscle of each participant is measured using a hand dynamometer. In order to measure the MVC level, the participant is asked to sit down on a chair and extend his arm fully and put his hand in the hand dynamometer (fixed under the table in front of the participant) and push it up as much as possible. The hand dynamometer value shows the maximal force which is the MVC value at the start of the experiment. We collect the data three times and use the average value. We then ask the participant to hold a weight around 30 percent of their MVC during the exercises. The exercises consist of five 2-minute intermittent static arm abduction cycles. For each cycle, the contraction duration is 90 seconds followed by 30 seconds rest. We used 166 seconds cycle time similar to the high cycle condition in [36] in our pilot study but the participants complained that it was very hard and we reduced the cycle time to 120 seconds in the final study. Note that the abduction cycle is different from the experiment cycle discussed in Sect. 2.6.2.1. The maximum isometric force of the participant's shoulder is also measured after every 10 trials.

2.6.2.4 Experiment Procedure

A participant is asked to read a written instruction on how to complete the assembly task. Verbal instructions are also given and the participant is instructed that no data will be collected during the training session. The training session consisted of 10 trials of an assembly task different from the actual experiment task. During the training session, the participant is able to change the speed of the robot using the up or down arrow keys of the keyboard at any time. In the experiment, the robot speed can be adjusted manually as well as autonomously. The adjustment of the robot speed in the manual mode during the experiment task is similar to the training session. In the autonomous control mode, the robot adjusts its speed and the participant cannot change it. In the collaborative control mode, the robot adjusts its speed autonomously while the participant is also able to change the robot speed whenever he/she wants.

The experiments were conducted over three days. In the first day, after the training, the participant performed the experiments in manual mode. The non-fatigue high robot flexible and non-fatigue low flexible conditions are the first and second experiments, respectively. Next, in order to run the experiments in the fatigue condition, the participant was asked to do the fatigue exercise as described in Sect. 2.6.2.3. The participant is then asked to perform the experiments under the fatigue high flexible and fatigue low flexible conditions in the third and fourth experiment, respectively. The data obtained in the manual mode is used to train the neural network based on the explanations in Sect. 2.3. We train the artificial neural network for all of the conditions in manual mode. The trained networks are used for the corresponding condition in the autonomous and collaborative modes. The experiments conducted in the second and third days are for the autonomous and collaborative modes, respectively.

2.6.2.5 Measurements and Scales

At the start of the first day of the experiment, the participant was asked to fill out a subjective demographic questionnaire. Moreover, at the beginning of each day, the participant was asked to rate his/her trust to Baxter. A 7-point Likert scale is used for measuring real-time subjective trust of the participant in the robot. The participant is instructed that extreme values of the trust scale—'1' and '7'— mean that they do not trust robot at all or they trust the robot completely. The real-time trust value is measured during the experiment using a separate laptop screen other than

Baxter head screen. A message on Baxter head screen pops out and asks the participant to evaluate his/her trust at the end of each trial. Moreover, the participant is informed that he can increase or decrease the trust value anytime during the experiment using the right or left arrow keys of the keyboard on the laptop. Once a participant finishes all 10 trials, we ask him to fill out a survey. The survey measures the overall workload based on the NASA TLX [31] scale.

2.6.3 Experimental Results

2.6.3.1 Trust Model Identification Procedure

We use the Autoregressive Moving Average (ARMA) Model in the MATLAB System Identification Toolbox [56] to identify the parameters of time-series trust model based on the experiment data (i.e. A, B_1, B_2, C_1, C_2, D_1, and D_2 in Eq. (2.1)). The tracking system shown in Fig. 2.3(b) is used to measure working speed of the human worker, V_H for calculating the robot flexibility in Eq. (2.2). Robot speed is the command that is sent to the robot by the computer. The real-time trust measurements are collected during the experiment.

2.6.3.2 Manual Mode

The results of the experiments are shown in Fig. 2.7. Note that we have normalized the trust level for the sake of comparison but the 7-point Likert scale can be used for analysis without difficulty. As can be seen in this figure, for the first (non-fatigue high robot flexibility) and second (non-fatigue low robot flexibility) sets of experiments, the human is not fatigued so his performance is maximum, i.e. $P_H = 1$. However, after imposing fatigue during the third (fatigue high robot flexibility) and fourth (fatigue low robot flexibility) sets of experiments, his performance decreases. In the first experiment when there is no fault, the participant's trust increases but it drops after the occurrence of faults in the second experiment. In the absence of the faults within the third experiment the trust recovers. Note that the level of trust increases with the higher rate as compared to the first experiment with the same robot flexibility condition. In the fourth experiment with low-flexible robot performance, the trust decreases but it decreases with lower pace as compared to the case with

higher human performance (non-fatigue condition). The quantified trust model in manual mode is

$$\begin{aligned} T(k) = \quad & 0.991T(k-1) + 0.014P_R(k) + 0.127P_R(k-1) + 0.046P_H(k) \\ - \quad & 0.143P_H(k-1) - 0.075F(k) + 0.003F(k-1), \end{aligned} \tag{2.11}$$

For this mode, the fit value for the ARMA model is 70.61% which shows that the model fits the data well. Equation (2.11) indicates that with low values of P_R or high values of P_H trust declines and vice versa. We also observe that since $A = 0.991$, almost 7 times the weight of the second largest parameter, the current trust is mainly dependent on the previous trust if no dramatic performance change occurs. This is consistent with the intuition that trust is highly related to prior trust and only changes when there is a large performance variation.

2.6.3.3 Autonomous Mode

The results of the experiments are shown in Fig. 2.8. As can be seen in this figure, the human and robot performance as well as the changes in the trust value are similar to that of in manual mode. For this mode, the fit value for the ARMA model is 62.34%. The time-series trust model for this mode is

$$\begin{aligned} T(k) = \quad & 0.959T(k-1) + 0.021P_R(k) + 0.015P_R(k-1) + 0.078P_H(k) \\ - \quad & 0.064P_H(k-1) - 0.045F(k) - 0.013F(k-1), \end{aligned} \tag{2.12}$$

2.6.3.4 Collaborative Mode

The results of the experiments are shown in Fig. 2.9. The fit value for the ARMA model is 45.65%. As it can be seen in Fig. 2.9, the trust value increases slowly at the start of the experiment from 0.5 to around 0.75. Fault occurrences cause a rapid trust degradation to the level of less than 0.1. Next, the participant's trust to the robot increases sharply after eliminating the faults and it decreases again after the faults occur toward the end of the experiment. Note that for the first and second half phase of the experiment, although the increasing trend of trust without faults and the decreasing trend of trust with faults are consistent, the intensity of these variations within these two phases is very different. In the former phase, trust increases very slowly but drops very fast; While

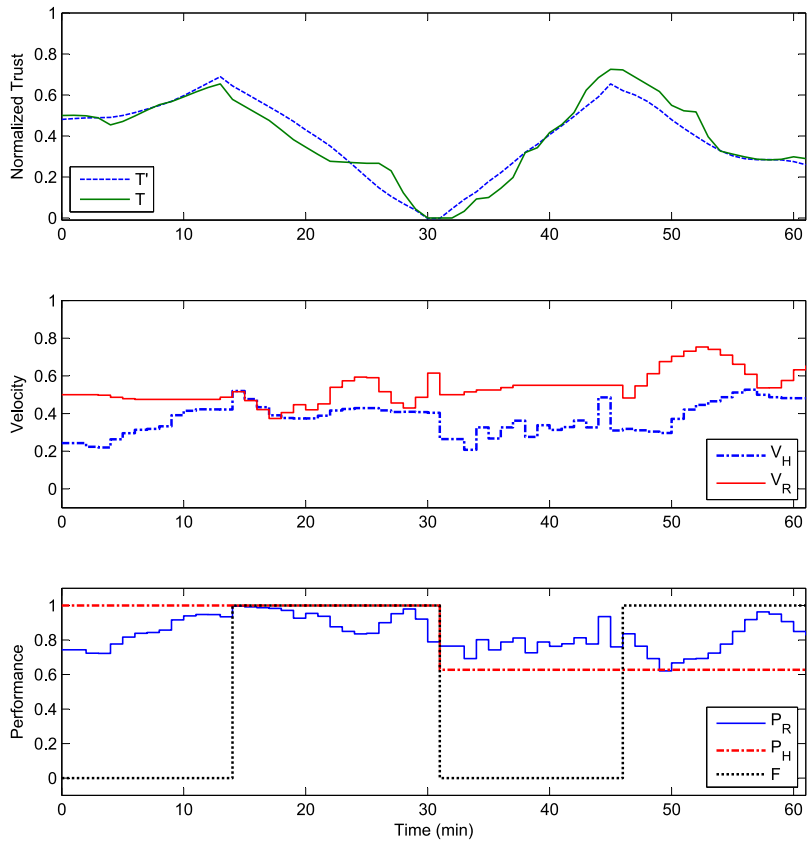


Figure 2.7: Evolution of human working speed V_H, human performance P_H, robot speed V_R, robot performance P_R, fault, trust T, and trust estimation T' using (2.11) under the manual mode.

in the latter phase, trust recovers very sharply and declines gradually. This can justify why the fit value is smaller in the collaborative mode compared to the other modes. Future work will seek models with better fitness based on validated human factor research. The time-series trust model for this mode is

$$\begin{aligned} T(k) = \quad & 0.991T(k-1) + 0.099P_R(k) + 0.033P_R(k-1) - 0.039P_H(k) \\ - \quad & 0.033P_H(k-1) - 0.062F(k) - 0.022F(k-1), \end{aligned} \tag{2.13}$$

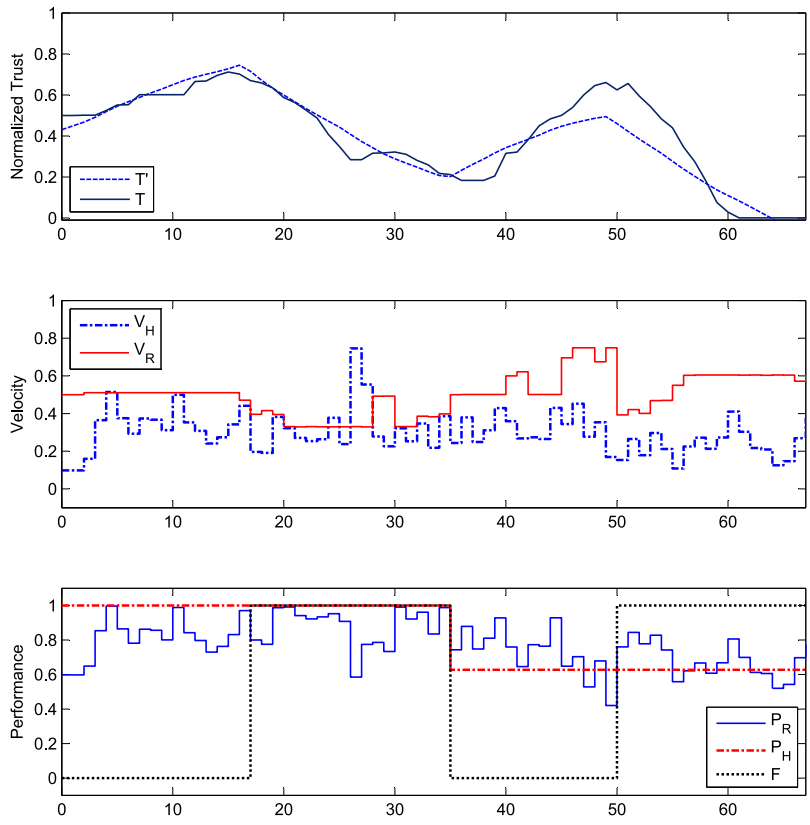


Figure 2.8: Evolution of human working speed V_H, human performance P_H, robot speed V_R, robot performance P_R, fault, trust T, and trust estimation T' using Equation (2.12) under autonomous mode.

2.6.4 Comparison and Conclusion

We measure the participant workload with NASA TLX index after each experiment. Moreover, we calculate the average values of robot speed, human speed, robot performance, human performance and trust in all of these conditions. Table 2.1 shows the comparison of these values for different experiment conditions. As can be seen in this table, for the fresh (non-fatigue) flexible condition, the overall workload of the participant is similar in all of the three control modes and it is lower as compared to the fresh inflexible condition for every control mode. Moreover, this value is lower for the fatigue flexible condition as compared to fatigue inflexible in all modes. In this table, for each experiment with a certain condition under a specific mode, ΔTrust shows the difference be-

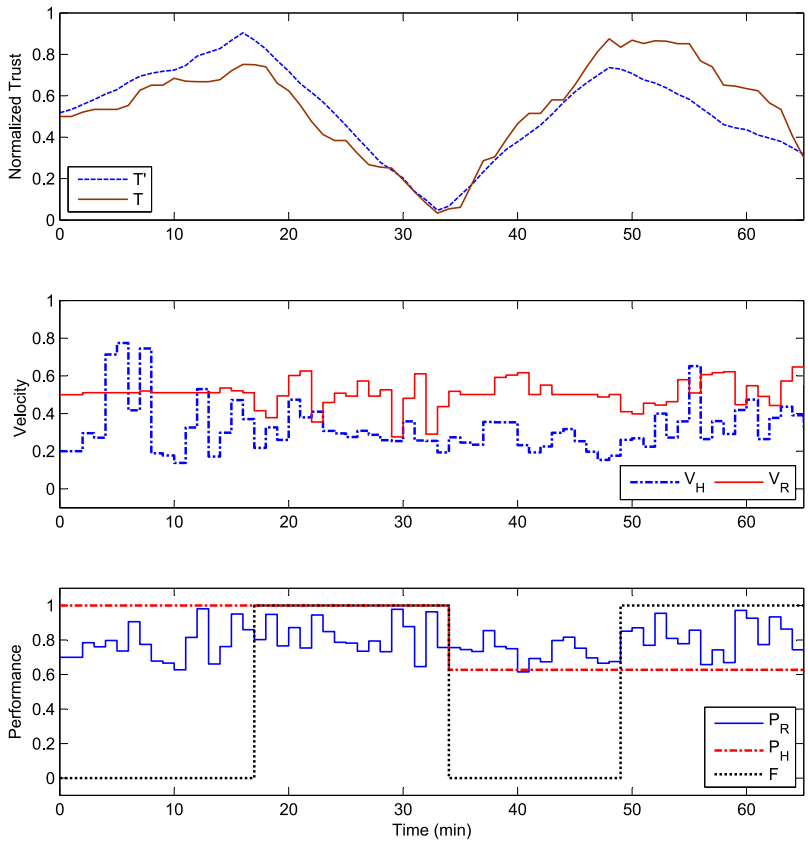


Figure 2.9: Evolution of human working speed V_H, human performance P_H, robot speed V_R, robot performance P_R, fault, trust T, and trust estimation T' using Equation (2.13) under the collaborative mode.

tween the initial and final trust. The general trend of changes of this value for all the control modes are similar: it goes up in the flexible mode and goes down in the inflexible mode. However, it can be seen that the influences of robot and human performances on trust vary for different control modes. For the fresh flexible and inflexible conditions, although the robot performances in the manual mode are higher than those in the autonomous and collaborative modes, the trust increments are lower as compared to the corespondent values in other modes.

Mode	Manual (mean values)				Autonomous (mean values)				Collaborative (mean values)			
Condition	Fresh Flexible	Fresh Inflexible	Fatigue Flexible	Fatigue Inflexible	Fresh Flexible	Fresh Inflexible	Fatigue Flexible	Fatigue Inflexible	Fresh Flexible	Fresh Inflexible	Fatigue Flexible	Fatigue Inflexible
Workload	47	67.7	46	63.3	47	53.67	38.67	59	48.3	57.7	52.7	59.7
V_H	0.32	0.42	0.32	0.43	0.32	0.32	0.32	0.24	0.37	0.30	0.26	0.36
V_R	0.48	0.49	0.54	0.64	0.51	0.37	0.59	0.55	0.51	0.47	0.53	0.52
P_R	0.84	0.92	0.78	0.80	0.81	0.88	0.73	0.69	0.77	0.82	0.73	0.82
Trust	4.24	2.88	3.01	3.64	4.59	3.41	3.48	2.32	4.75	3.19	3.80	5.08
Δ Trust	0.9	-3.5	4.4	-2.6	1.2	-2.7	3.1	-4.0	1.5	-4.3	4.9	-3.4

Table 2.1: Comparison between workload, average human and robot working velocity and performance, and trust for different experiment conditions and modes

2.7 Conclusion

In this chapter, we proposed a time-series trust model for a human worker and his/her robot coworker in a collaborative manufacturing task. We developed a performance model for robot flexibility based on the difference between the human and robot working speed. Since the tasks in manufacturing usually are repetitive kinesthetic tasks, we used the muscle fatigue and recovery model to capture the human performance. We used three methods to control the robot performance. These methods are manually by the human, autonomously by a neural network based robust intelligence controller, or collaboratively using both manual and autonomous inputs. We provided both numerical simulations and experiment validations to demonstrate the effectiveness of the proposed trust model and robust intelligent control scheme. Based on the well-known human factors result we adopted a linear trust model in this chapter. In the next chapters, we will investigate the applicability of the trust model in general HRC manufacturing and modify accordingly for specific scenarios to increase the model fitness.

Chapter 3

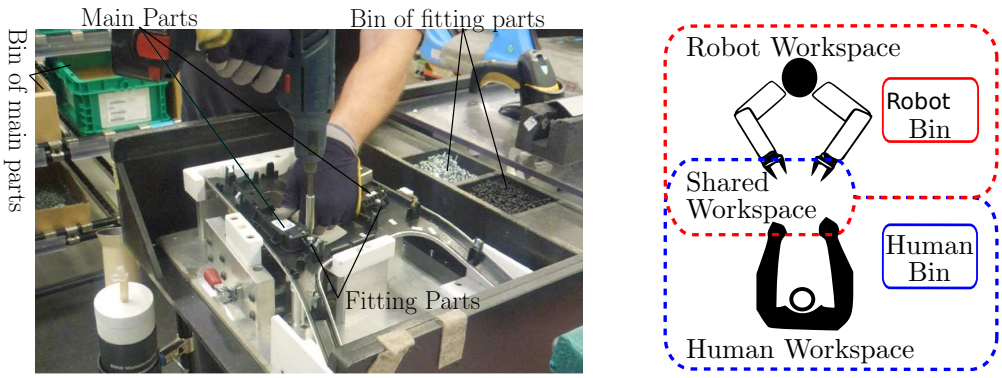
An Integrated Framework for Collaborative Assembly

3.1 Introduction

In this chapter, a novel framework for integrating HRI factors into the robot motion controller for human-robot collaborative assembly tasks in a manufacturing hybrid cell is proposed. To meet human physical demands in such assembly tasks, an optimal control problem is formulated for pHRI based robot motion control to keep pace with human motion progress. sHRI is also augmented into the framework by considering a computational model of the human worker's trust in robot as well as robot facial expressions. The human worker's trust in robot is computed and used as a metric for path selection as well as a constraint in the optimal control problem. Robot facial expression is displayed for increasing the situational awareness of the human worker. The proposed framework is evaluated by designing a robotic experimental testbed and conducting a comprehensive study with a human-in-the-loop. Results of this study show that compared to the manual adjustments of robot velocity, an autonomous controller based on pHRI, pHRI and sHRI with trust, or pHRI and sHRI with trust and emotion result in 34%, 39%, and 44% decrease in human workload and 21%, 32%, and 60% increase in robot's usability, respectively. Compared to the manual framework, human trust in robot increases by 38% and 42%, respectively, in the latter two autonomous frameworks. Moreover, the overall efficiency in terms of assembly time remains the same.

The assembly task chosen in this paper is usually accomplished by human workers and includes some repetitive physical movements for pick-and-place. Fig. 3.1(a) shows an example of a conventional manufacturing cell in which the human worker first brings the required parts from the bin of main parts and puts them together, then grabs some fitting parts such as screws or bolts from the bin of fitting parts and finally assembles the product. A flexible robot can assist the human worker in doing such a task by bringing the required parts to the human worker [93]. The goal of the human-robot team is to assemble a product in a hybrid manufacturing cell. This cell is equipped with sensory devices for safety reasons and enables the human and the robot to perform tasks collaboratively while some parts of their workspaces are shared with each other [12]. Fig. 3.1(b) shows an example of a typical hybrid manufacturing cell, where the bins of different assembly parts are assigned to the robot and human worker, respectively. The human workspace includes the shared workspace, the designated bin for the human, and the area covered by paths between them. The robot workspace is defined similarly but contains the bin of the robot. The robot picks the main parts from the robot bin and places them in the shared workspace with the human worker. The human picks the fitting parts from the human bin to the shared workspace and assembles the final product. Our proposed framework allows the robot arm to select paths between the robot bin and the shared human-robot workspace based on trust evaluation and then move along the selected path while its translational velocity along the path is adjustable. More specifically, a set of pre-planned paths are stored and the appropriate path is selected based on HRI criteria. This will be discussed in Section 3.3.1. The pHRI-based control condition will involve prediction of human motion and synchronization of the robot motion progress with that of the human. This will be discussed in Section 3.4.1. Human trust in robot and robot emotional expressions will be considered as two main aspects of sHRI and devised in two integrated control conditions. These will be discussed in Section 3.4.4 and Section 3.4.5, respectively. In Section 3.5, four control conditions are considered, i.e. the pure manual condition, the pHRI-based condition (Section 3.4.1), the integrated pHRI and sHRI condition considering collision avoidance and trust (Section 3.4.4), and the more comprehensive integrated condition considering collision avoidance, trust and emotion displays (Section 3.4.5). In the manual condition, minimal interaction is considered in the design of the robot controller. The human worker manually adjusts the robot work pattern (i.e. path and speed).

In Chapter 2 and [91], we investigate how human trust in robot can be measured during the HRC manufacturing and construct a trust model for assembly tasks. In [72], we demonstrate how the



(a) A conventional manufacturing cell (Photo courtesy of BMW Spartanburg, SC).

(b) An HRC hybrid manufacturing cell.

Figure 3.1: A conventional manual vs. a hybrid HRC manufacturing cell.

integration of dynamic emotion in a human-computer interface (HCI) can benefit sHRI in assembly tasks in manufacturing. In [88], we explore the robot redundancy and alter its configuration in handover operations during assembly based on artificial robot-to-human trust. More recently, we propose in [93] a framework for augmenting both the pHRI and sHRI factors into the robot controller in which a trust model was considered for sHRI. This work is extended based on the HRI framework in [93].

Fig. 3.2 shows our proposed framework and the experimental setup. Section 3.3 explains the detailed derivations of the robot motion controller (the Robot Controller and Path Selection blocks in Fig. 3.2). The HRC system (the Human Motion Estimation, Trust Simulator, and Facial Expressions blocks) and control laws (the NMPC Solver block) are presented in Section 3.4. The experimental study (with PhaseSpace for tracking human motion and Baxter robot in Fig. 3.2) conducted for the evaluation of our proposed framework is presented in Section 3.5. Thorough statistical analysis of the results of the experiments is presented in Section 3.6. The chapter is concluded in Section 3.7.

3.2 Related Work

HRC in assembly lines can be viewed as a twofold problem: the task scheduling problem and the task execution problem. The solution to the scheduling problem identifies when different tasks should be assigned to the human and the robot while the solution to the execution problem is robot controllers and motion planners. Both problems aim to improve safety, efficiency, cost, and productivity. In [39], a pHRI framework is proposed for task allocation and planning in HRC assem-

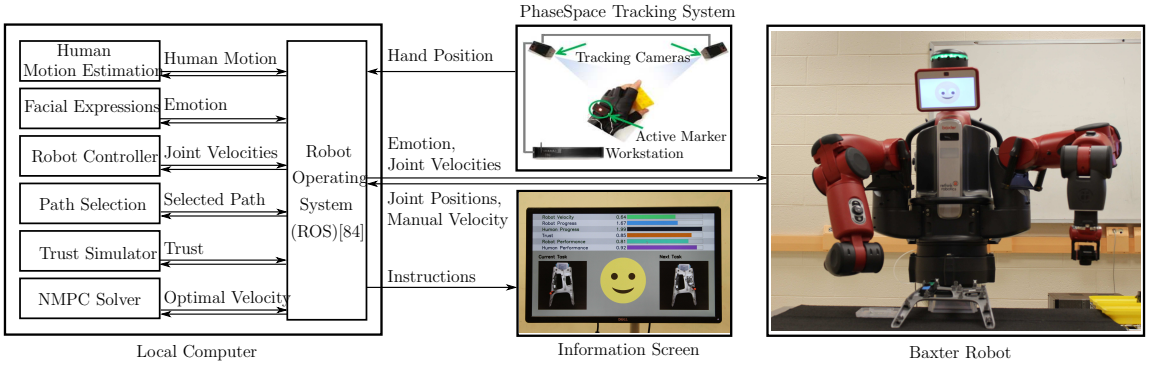


Figure 3.2: The architecture of the integrated framework and illustration of the experiment setup.

bly. It consists of a two-layer planner for the high-level abstraction and atomic level of allocation. The higher layer planner generates a coordinated skill sequence for the human-robot team. On the atomic layer, different hierarchical and concurrent state machines describe the skills of the robot. A summary of the EU project ROBO-PARTNER for the integration of automation and human capabilities in assembly operations is presented in [71] where efficient methods for task planning and execution are developed.

In [114], the human and robot agents coexist in a same cell and share tasks such as pick, place, release, and move. The task assignment is sequential and based on capability, availability, and operation time needed by either agent to perform a job. In the framework, the human and robot do not work simultaneously and their interaction is restricted by the safety consideration. A contact-based pHRI framework for assembly tasks is proposed in [13]. The task is to assemble a car joint which includes insertion of six balls in a joint's case. The robot behaves actively to reduce the load on the human and passively to comply to his/her demands. Through both risk analysis and experiment validations, it is shown that the framework is compatible with safety standards and reduces human workload. Another framework for pHRI is proposed in [80] where the robot controller adapts behavior according to the human fatigue level during the task. The task is co-manipulation and the initial interaction is a leader-follower relationship where human is the leader and robot learns skills by feedbacks from the human. When the human fatigue reaches a predetermined level, the interaction alters from collaboration to supervision. The robot takes over the task to reduce the human load and the human controls the high-level interaction behavior. A pHRI framework for hybrid manufacturing cell of cable harness assembly is proposed in [112] for ensuring safety in task execution. Task planning is performed based on a hierarchical task decomposition approach adopted

from ergonomics. Different hardware and control strategies such as designated and safe workspaces for the human and robot, safe design of the robot, and human monitoring are proposed for safety implementation. The impacts of robot motion speed and distance from the human are evaluated on system performance and human mental workload. Another HRC assembly cell for task execution is presented in [74]. The system has three key elements: a mobile robot with two manipulators for feeding the parts, production process information interface for the human, and safety management for HRC.

In sum, planning and execution are two major problems in HRC assembly in manufacturing. Some of the related works address both problems while others only consider one aspect. In this paper, we also assume that the solution to the planning problem is given and focus on the execution problem. Moreover, most existing works consider pHRI and more specifically safety. Their ultimate goal is to find safe and efficient control policies for robots to accomplish the required task (assigned by the planner) in the presence of a human worker without collision. The main contribution of our work is to integrate both pHRI and sHRI into the robot path planning and speed control for safety, efficiency, as well as balanced human experience.

3.3 Robot Motion Controllers

In this section, the design of robot motion controller including path planning and the calculation of the robot joint velocity based on the optimal velocity of the robot end-effector along the planned path will be introduced in sequence.

3.3.1 Robot Path Planning

Robot motion planning is an active research topic and a considerable amount of literature is dedicated to this field. Motion planning includes (i) path planning for searching a (possibly optimized) collision-free path in the configuration space (i.e. the set of all robot configurations) regardless of the dynamics of the robot, and (ii) trajectory planning which considers the time evolution of robot dynamics (sometimes along a planned path) for satisfying certain optimization requirements as well as differential constraints. Sampling-based planners are widely employed to construct a data structure (roadmap or tree) for representing collision-free paths [48]. Other approaches for path planning include potential field based techniques and combinatorial methods which also make roadmaps,

such as cell decompositions [48]. Trajectory planning addresses optimization criteria such as time, energy, force, effort, or the jerk minimization and differential constraints, i.e. limits on velocities, and possibly accelerations due to kinematic and dynamic considerations of the robot. Trajectory planning can be addressed either by direct methods which implement sampling-based algorithms through considering differential constraints or by decoupled approaches which first plan a path and then compute a timing function along the path [48]. In this work, our focus is on constructing a motion trajectory to improve HRI rather than other well-known optimization criteria. Hence, the decoupled approach is adopted and we formulate the problem as obtaining the motion velocity along a predefined path based on the consideration of HRI factors. The Baxter robot we use has some safety constraints that prevent solving the problem in the acceleration level. Moreover, identification of robot dynamics can be a challenging task [124]. To make the framework independent of knowledge of the robot dynamics model, this problem is solved in the velocity level.

Path planning can be done either on-line as the robot is executing the task or off-line before the task starts [48]. On-line path planning is more desirable if there are uncertainties; however; it requires more computational resources. In the manufacturing setting, since the initial and final positions of the robot end-effector, as well as the shared human-robot workspace, are given (see Fig. 3.3), the uncertainty is negligible. Thus, the problem is simplified by considering off-line obstacle avoidance and polynomial curves defined in the task-space. However, the proposed framework can be extended to consider more advanced motion planning techniques in future work.

For accomplishing the pick-and-place task, the robot end-effector is desired to pass through

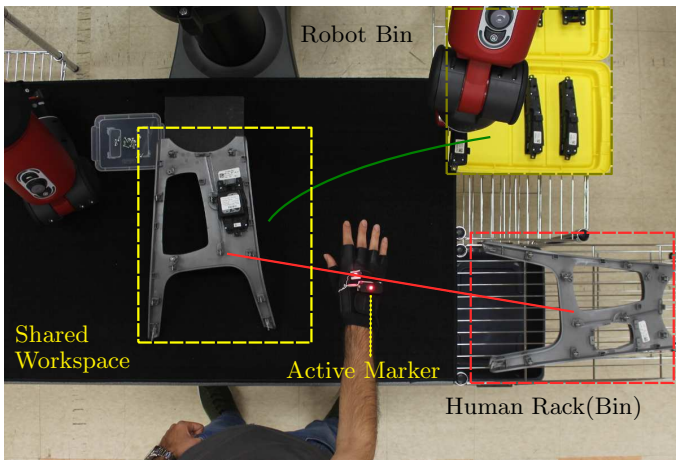


Figure 3.3: Robot (green) and human (red) sample paths.

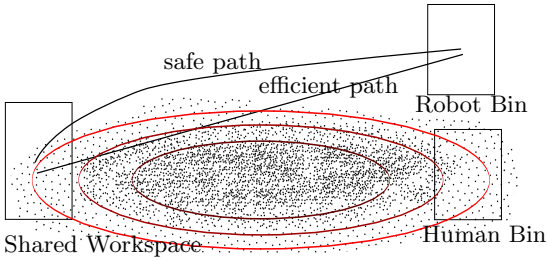


Figure 3.4: Illustration of sample data of human movements in the workspace: an efficient robot path versus a safe robot path.

a set of points of interest (POIs) defined in the task-space. This set includes the fixed initial and final POIs, and the intermediate POIs which are chosen via the high-level path-planing. Thus, given a set of n_p data points $\{\boldsymbol{d_i} = [x_i,\ y_i,\ z_i]^T \in \mathbb{R}^3,\ 1 \leq i \leq n_p\}$, we choose a path in the three-dimensional Euclidean space, $\mathbb{R}^3$, that goes through these points, i.e. $\boldsymbol{p} : s \in [0, l] \rightarrow \mathbb{R}^3$, where the path parameter s is the arc length of the distance traveled along the path. Each point on this path is given by $\boldsymbol{p}(s) = [x(s),\ y(s),\ z(s)]^T, s \in [0, l]$. We use a simple cubic polynomial to construct $\boldsymbol{p}(s)$. Moreover, based on the HRI criteria, a high-level path planning approach can be adopted for choosing a candidate path from a set of predefined paths. Fig. 3.4 shows two of the candidate paths. The dots in the figure represent some sample data of a human worker's hand position while performing a task. By fitting appropriate probability distributions to these data points, regions with different safety levels can be identified. The red contours represent three examples of data points falling within the same density distributions. We utilize the trust of the human to robot as an HRI factor in the path selection strategy. A similar approach was used in [88] by developing an artificial robot-to-human trust model and implementing trust-based arm configuration and motion planning of a collaborative robot in handover tasks. Experiment results with a human-in-the-loop in [88] show that trust-based handover strategy statistically outperforms non-trust based strategy in both pHRI and sHRI criteria. In this paper, we adopt a similar concept for selecting the robot path with differences in the detailed trust model and path planning strategy. More specifically, in [88] the robot transitional path in the task-space is fixed and it performs the motion as planned unless the trust of robot to human drops below some threshold value. In that case, the robot alters its end-effector orientation to minimize the impact force between the human and the robot. In this work, the robot end-effector orientation is fixed but we change the robot transitional path in the task-space based on the dynamic evaluation of human-to-robot trust. In the following, we justify the choice between a

conservative (safe) and an aggressive (efficient) robot path based on the analysis of human-to-robot trust. Both human and robot can cause a collision. Most works in the literature deal with collision avoidance from the robot perspective. For example, in [65], the collision probability is defined as a function of the measurement error and relative velocity between the human and the robot. As the uncertainty in robot measurement data increases, the likelihood of the collision increases. Thus, low performance of the robot increases the probability of the collision. On the other hand, with the increase of physical workload and hence performance decline, the situational awareness of human worker decreases [79] and the probability of interfering and colliding with the robot increases. Here, we consider both cases and assume that either low performance of the robot or the human increases the likelihood of the collision. According to performance-centered metrics [51], low performance of robot leads to low human trust in robot. Note that due to the close interaction of the human and the robot, the changes in the performance of the robot impacts the human's performance as well. In Section 3.4.4, we will define a computational model of human's trust in the robot (Eqn. (3.20)) which will be considered in our HRI-based motion planning. According to this model, human trust in robot depends on prior trust, robot performance, and human performance. That is, poor performances of the human and robot result in a low trust value. In turn, the computational trust level can be used to indicate both human and robot performance and hence is a criterion for choosing between safe and efficient paths. If only considering efficiency, a path with shorter length would be chosen. However, if the trust level is low, choosing a short path might result in higher probability of collision. Hence, for safety consideration, a more conservative but longer path should be chosen. If trust is high, selecting a short path might still be safe. Therefore, there are trade-offs between efficiency and safety for choosing the path. Since different levels of human trust in the robot reflect the variation of both the human and robot performances, we can adopt a trust-based path planning method. We use a look-up table to specify a corresponding pre-planned path for every range of trust values. In general, low values of trust are associated with low performances of human or the robot which suggests a more conservative path, while a more efficient path can be chosen for high values of trust.

Remark 2. The computational model of trust in this paper reflects the dynamic and temporary trust of the human worker in the robot during the interaction in terms of flexibility and efficiency as well as human's own performance. This is more specific than the general notation of trust in robot which determines the human's acceptance and hence utilization of the robot (automation) [30, 51]. •

The human arm movement data were collected in the pilot study and used for the construc-

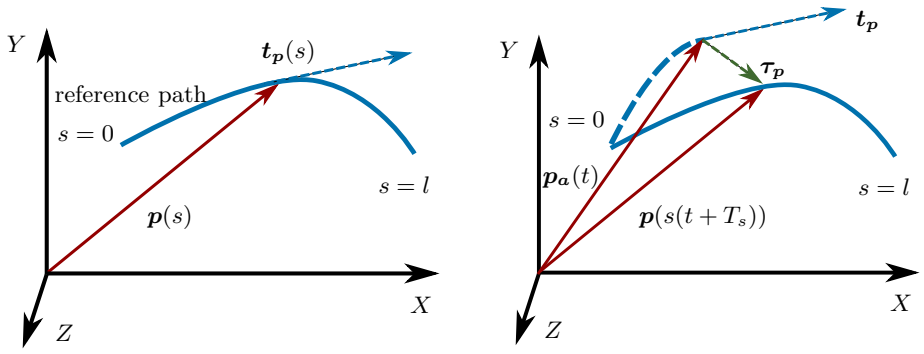
Parameter	Description
$\boldsymbol{\phi}$	End-effector orientation in task-space
$\boldsymbol{p}$	End-effector position in task-space
s	Arc length of the distance traveled along a path
$\dot{s}$	Derivative of arc length (s) with respect to time
$\boldsymbol{x}$	End-effector pose including position and orientation
$\boldsymbol{q}$	Joint positions of the robotic arm
$\boldsymbol{f_t}$	Direct kinematics function
$\boldsymbol{J}$	Task Jacobian matrix
$\dot{\boldsymbol{q}}_{\boldsymbol{0}}$	An arbitrary joint-space initial velocity
$\boldsymbol{t_p}(s)$	Tangent to the path $\boldsymbol{p}(s)$ at s
$\boldsymbol{t}(s)$	Change of end-effector desired pose $\boldsymbol{x}$ with respect to s
$\boldsymbol{\tau_p}(s)$	Vector from the actual pose of the end-effector at s to its desired pose along the path
$\boldsymbol{\tau}(s)$	Change of end-effector actual pose $\boldsymbol{x}$ with respect to s
v	Translational velocity along the path

Table 3.1: Description of the parameters in motion controller

tion of paths with lower probability of collision of the human and the robot.

3.3.2 Robot Joint Velocity Control

The problem of controlling a robot manipulator along a predefined path is widely studied in the literature. It mainly focuses on determining the robot joint velocities for moving along a path in a task-space such that it satisfies certain constraints. Most efforts made in this area solve this problem in the acceleration level [81]. However, due to the safety constraints of Baxter and lack of knowledge of robot dynamics, we solve this problem in the velocity level. The details of the robot joint controller for the proposed HRI framework is discussed in [93]. In the following, we briefly restate the solution, describe an adjustment we made for increasing the control accuracy, and finally present Algorithm 1 for the robot controller. Table 3.1 summarizes the parameters used in this section. It is desired to manipulate the robot end-effector along a given path such that its speed along the path can be dynamically adjusted. Let the configuration of the end-effector of the robot arm in the task-space be denoted by the reference point position, $\boldsymbol{p} \in \mathbb{R}^3$, and the orientation vector, $\boldsymbol{\phi} \in \mathbb{R}^3$ (such as Euler angles or the roll-pitch-yaw representation). More specifically, we define $\boldsymbol{x} = [\boldsymbol{p}, \boldsymbol{\phi}]^T \in \mathbb{R}^6$ as the vector of the pose (position and orientation) of the end-effector, where ^{T} denotes transpose. We define $\boldsymbol{q} \in \mathbb{R}^n$ as a vector of the joint positions of the robotic arm in the joint-space. The relation between the task-space and the joint-space is expressed by direct



(a) Reference trajectory (b) Compenstated trajectory

Figure 3.5: The reference vs. the compensated trajectories. $\boldsymbol{p_a}(t)$ is the actual position of the end-effector at current time and $\boldsymbol{p}(s(t + T_s))$ is the reference position at time $t + T_s$.

kinematics equation

$$\boldsymbol{x} = \boldsymbol{f_t}(\boldsymbol{q}), \quad m < n, \tag{3.1}$$

where m and n are degrees-of-freedom (DoFs) in the task-space and joint-space, respectively. In our problem $m = 6$ and $n = 7$. The function $\boldsymbol{f_t}$ represents direct (forward) kinematics. The first-order differential kinematics are

$$\dot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}, \tag{3.2}$$

where $\boldsymbol{J}(\boldsymbol{q}) = \partial \boldsymbol{f_t}/\partial \boldsymbol{q}$ is the $m \times n$ task Jacobian matrix. The general solution to Eqn. (3.2) is

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{\dagger}(\boldsymbol{q})\dot{\boldsymbol{x}} + \left(\boldsymbol{I} - \boldsymbol{J}^{\dagger}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})\right)\dot{\boldsymbol{q}}_{\boldsymbol{0}}, \tag{3.3}$$

where $\boldsymbol{J}^{\dagger}(\boldsymbol{q})$ is the Moore-Penrose pseudoinverse of $\boldsymbol{J}(\boldsymbol{q})$, $\dot{\boldsymbol{q}}_{\boldsymbol{0}}$ is an arbitrary joint-space initial velocity, and $\boldsymbol{I}$ is an identity matrix.

Remark 3. We assume that in addition to the Cartesian position coordinates and Euler angles representations introduced above, the desired positions and orientations of the end-effector can also be given for all admissible values on the curve parameterized by $s \in [0, l]$, i.e. $\boldsymbol{p}(s)$ and $\boldsymbol{\phi}(s)$. As the robot end-effector follows the given path, since $s(t)$ is a function of time, $\boldsymbol{p}$ and $\boldsymbol{\phi}$ become implicit functions of time and the pose of the robot at the time t would be $\boldsymbol{x}(s(t))$. However, note that $\boldsymbol{x}(s(t)) \neq \boldsymbol{x}(t)$. If the end-effector deviates from the reference path, the actual pose is not on the

reference path and it can not be presented by using s anymore. We first solve the problem for the ideal case where the robot exactly follows the given path and then provide a solution for the deviated case.

Now, we show the relation between the end-effector velocity in the task-space $\dot{\boldsymbol{x}}$ and the robot velocity along the path $\dot{s}$. Let us denote the following operators for derivatives with respect to s and t, respectively, as $(\ )' = d/ds$ and $\dot{(\ )} = d/dt$. A tangent to the curve can be calculated as the vector $\boldsymbol{t_p}(s) = \boldsymbol{p}'(s)$. Define the unit vector of this tangent as $\hat{\boldsymbol{t}}_{\boldsymbol{p}}(s) = \frac{1}{||\boldsymbol{t_p}(s)||}\boldsymbol{t_p}(s)$, where $||\boldsymbol{t_p}(s)||$ is the 2-norm of $\boldsymbol{t_p}(s)$. Fig. 3.5(a) shows a demonstration of these definitions. If the robot arm follows $\boldsymbol{p}(s)$, its translational velocity can be written as $\dot{\boldsymbol{p}}(s(t)) = \dot{s}(t)\boldsymbol{p}'(s) = \dot{s}(t)\ \boldsymbol{t_p}(s)$. Since $\dot{s}$ is a scalar variable, this can be rewritten as $\dot{\boldsymbol{p}} = v\ \hat{\boldsymbol{t}}_{\boldsymbol{p}}(s)$, where $v = \dot{s}||\boldsymbol{t_p}(s)||$ is a scalar number representing the velocity at which robot is moving on $\boldsymbol{p}(s)$. We call v as the path velocity along $\boldsymbol{p}(s)$. Similar to $\boldsymbol{p}(s)$, we can define the minimal description of the end-effector orientation as $\boldsymbol{\phi}(s) = [\alpha(s),\ \beta(s),\ \gamma(s)]^T$. Since in pick-and-place applications, the robot orientation along the path can remain unchanged, and the pose direction vector $\boldsymbol{t}(s)$ can be written as

$$\boldsymbol{t}(s) = \left[\hat{\boldsymbol{t}}_{\boldsymbol{p}}(s),\ \boldsymbol{0}\right]^T. \tag{3.4}$$

Note that the vector $\boldsymbol{t}(s)$ represents the change of end-effector pose with respect to s. Therefore, due to the fixed-orientation of the end-effector, the change of orientation vector is set to $\boldsymbol{0}$, i.e. the given orientations for all values of s are equal. Since $\boldsymbol{x} = [\boldsymbol{p},\ \boldsymbol{\phi}]^T$, the robot end-effector velocity in task-space can be written as $\dot{\boldsymbol{x}}(s) = v\boldsymbol{t}(s)$. Together with (3.3), the joint velocities for the robot end-effector to move along $\boldsymbol{p}(s)$ can be computed as

$$\dot{\boldsymbol{q}} = v\boldsymbol{J}^{\dagger}(\boldsymbol{q})\boldsymbol{t}(s(t)) + \left(\boldsymbol{I} - \boldsymbol{J}^{\dagger}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})\right)\dot{\boldsymbol{q}}_{\boldsymbol{0}}. \tag{3.5}$$

To account for the deviations of the robot end-effector from the reference path, we define $\boldsymbol{x_a}(t) = [\boldsymbol{p_a}(t),\ \boldsymbol{\phi_a}(t)]^T$ as the actual pose vector of the end-effector. Let $\boldsymbol{\tau_p}(t)$ be a vector connecting the actual position of the end-effector $\boldsymbol{p_a}(t)$ at time t to the next point on the reference path $\boldsymbol{p}(s(t+T_s))$, where T_s is the controller sampling time (see Fig. 3.5(b)). Define the drifted trajectory direction as

$$\boldsymbol{\tau}(t) = \left[\hat{\boldsymbol{\tau}}_{\boldsymbol{p}}(t),\ \boldsymbol{0}\right]^T, \tag{3.6}$$

where $\hat{\boldsymbol{\tau}}_{\boldsymbol{p}}(t)$ is the unit vector of $\boldsymbol{\tau}_{\boldsymbol{p}}(t)$. Now, the compensated task velocity can be defined as $\dot{\boldsymbol{x}}_{\boldsymbol{c}}(t) = v\left(\rho_1 \boldsymbol{t}(s(t))\right) + \rho_2 \boldsymbol{\tau}(t))$, where ρ_1 and ρ_2 are the weighting factors with $\rho_1 + \rho_2 = 1$, $\rho_i \in [0, 1], i = 1, 2$. Here, choices of the values of ρ_1 and ρ_2 are trade-offs between accuracy and speed of robot end-effector, respectively. For the closest distance between the robot end-effector and reference trajectory, we can set $\rho_1 = 0$ and $\rho_2 = 1$. For the fastest forwarding speed in the direction of the reference we can set $\rho_1 = 1$ and $\rho_2 = 0$. In this work, since both accuracy in reaching the goal and forwarding speed are important, we choose $\rho_1 = \rho_2 = .5$. By replacing $\dot{\boldsymbol{x}}$ with $\dot{\boldsymbol{x}}_{\boldsymbol{c}}$ in (3.5), the compensated equation for joint velocity vector can be computed as $\dot{\boldsymbol{q}} = v\boldsymbol{J}^{\dagger}(\boldsymbol{q})\left(\rho_1 \boldsymbol{t}(s(t)) + \rho_2 \boldsymbol{\tau}(t)\right) + \left(\boldsymbol{I} - \boldsymbol{J}^{\dagger}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})\right)\dot{\boldsymbol{q}}_{\boldsymbol{0}}$. Furthermore, to avoid singularities, we use the damped least-squares technique [17] by letting

$$\boldsymbol{J}^T(\boldsymbol{q})\dot{\boldsymbol{x}} = \left(\boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q}) + \lambda^2 \boldsymbol{I}\right)\dot{\boldsymbol{q}} \tag{3.7}$$

instead of (3.2) where $\lambda \in \mathbb{R}$ is the damping factor. Assuming zero initial joint velocities, i.e. $\dot{\boldsymbol{q}}_{\boldsymbol{0}} = \boldsymbol{0}$, the solution of (3.7) is

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^*(\boldsymbol{q})\dot{\boldsymbol{x}}, \tag{3.8}$$

$$\boldsymbol{J}^*(\boldsymbol{q}) = \left(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}^T(\boldsymbol{q}) + \lambda^2 \boldsymbol{I}\right)^{-1}\boldsymbol{J}^T(\boldsymbol{q}). \tag{3.9}$$

The details of this approach are described in [17]. The solution is a trade-off between accuracy and feasibility for choosing the joint-space velocity needed to achieve $\dot{\boldsymbol{x}}$. Considering the compensated task velocity $\dot{\boldsymbol{x}}_{\boldsymbol{c}}$, (3.8) can be written as

$$\dot{\boldsymbol{q}} = v\boldsymbol{J}^*(\boldsymbol{q})\left(\rho_1 \boldsymbol{t}(s(t)) + \rho_2 \boldsymbol{\tau}(t)\right). \tag{3.10}$$

Eqn. (3.10) is the equivalent of (11) in [93]. To further increase the accuracy of the robot end-effector motion along the path, we implement the closed-loop inverse kinematics (CLIK) approach [14]. Thus, (3.10) can be written as

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^*(\boldsymbol{q})\{v[\rho_1 \boldsymbol{t}(s(t)) + \rho_2 \boldsymbol{\tau}(t)] + \boldsymbol{K}[\boldsymbol{x}(t) - \boldsymbol{f}_{\boldsymbol{t}}(\boldsymbol{q})]\}, \tag{3.11}$$

where $\boldsymbol{K}$ is a constant positive-definite gain matrix. In our experiment (Section 3.5), we choose $\boldsymbol{K} = .01\boldsymbol{I}_{n\times n}$. Algorithm 1 provides a brief summary of the motion control procedure of the robot.

Algorithm 1 Moving Robot Along a Given Path

procedure MOVE_ROBOT($\boldsymbol{p}$) ▷ Moves along $\boldsymbol{p}$
  $s \leftarrow 0$
  Subscribe(v) ▷ Update path velocity from solver
  Read($\boldsymbol{q}$) ▷ Get joint positions
  while $s \leq l$ **do** ▷ Check if the robot reaches the end
    Calculate($\boldsymbol{t}(s)$) ▷ Eqn. (3.4)
    Calculate($\boldsymbol{\tau}(s)$) ▷ Eqn. (3.6)
    Calculate($\boldsymbol{J}^*$) ▷ Eqn. (3.9)
    Calculate($\boldsymbol{f}_t(\boldsymbol{q})$)
    Calculate($\dot{\boldsymbol{q}}$) ▷ Eqn. (3.11)
    Publish($\dot{\boldsymbol{q}}$) ▷ Send joint velocity commands
    Read($\boldsymbol{q}$) ▷ Get joint positions
    Calculate($\boldsymbol{x}(t)$) ▷ Eqn. (3.1)
    Calculate(s)
  end while
end procedure

3.3.3 Robot Path Velocity Controller

So far, we discuss the control of the robot end-effector along a reference path with a given translational velocity. Next, the controller for adjusting the transitional velocity along the path will be discussed. In HRC assembly scenarios, this velocity can be either fixed or varying depending on criteria such as HRI and productivity. On one hand, faster movement of the robot results in higher efficiency. On the other hand, the robot should keep pace with human for better HRI. The robot path velocity can be set manually by the human worker based on his/her preference or automatically based on objective performance measures. The manual adjustment of the robot speed can be realized through some human-machine interface (HMI). Baxter has a wheel button on both of its wrists and can be used for manual adjustment of the robot velocity. For automatic speed control, we model HRC systems based on pHRI and the integration of pHRI and sHRI, respectively. For each HRC system, we utilize the nonlinear model predictive control (NMPC) approach to solve for the optimal path velocity v. We incorporate the NMPC toolbox [24] into ROS for this purpose. The details of the HRC systems and the optimal control are described in the next section.

3.4 Human-Robot Collaboration (HRC) System

In this section, we develop robot velocity control along the path based on the pHRI and sHRI factors for the HRC system. For the pHRI system, it is desired to control robot motion so

that its motion progress can follow that of the human. For the sHRI system, the robot motion may be altered such that human trust in robot during the interaction is always higher than a threshold for effective HRC. Furthermore, robot emotion displays will be augmented in the framework for providing visual feedbacks regarding safety and performance.

3.4.1 Physical Human-Robot Interaction (pHRI) System

For assembling a product, the robot and human are required to bring r and h parts, respectively, to the shared workspace and human assembles them together. In the real factory environment, this process is continuously repeated in the assembly line. Here in the laboratory setting, we assume that the total number of finished products to be assembled is N_p (each with r and h parts from the robot and human, respectively). For each part, the robot moves along a path between the bin of parts and the shared workspace back and forth (Fig. 3.3). The sequence of the reference paths, $\{\boldsymbol{p}_i\},\ i = 1, 2, ...,$ is determined by the high-level trust-based path planning discussed in Section 3.3.1. We denote the robot's path progress, $S_R \in \mathbb{R}^+$, as follows:

$$S_R = \frac{s}{2rl_i} + \frac{c_r}{2r}, \tag{3.12}$$

where s is the arc length of distance traveled along the chosen reference path $\boldsymbol{p}_i$ with length l_i and c_r is the number of times that the robot completely traveled a path from the shared workspace and the robot bin or vice versa (see illustration in Fig. 3.6). The term $\frac{s}{2l_i}$ gives the ratio of the distance traveled by the robot end-effector along a reference path with respect to the total round trip length of the chosen reference path for picking one part. The robot starts the task of moving along a path with length l_1 from the shared workspace where initially both c_r and S_R are 0. Just before the robot reaches the bin of the required parts for the first time, we have $s = l_1$, $c_r = 0$, and $S_R = \frac{1}{2r} + \frac{0}{2r} = \frac{1}{2r}$. After the robot reaches the bin of required parts, it picks up a part and moves back towards the shared workspace on a path with length l_2. Note that at the start of this motion, the value of S_R is still the same since $s = 0$ and $c_r = 1$. When the end-effector reaches the shared workspace, $s = l_2$, $c_r = 1$, and $S_R = \frac{1}{2r} + \frac{1}{2r} = \frac{1}{r}$. The value of S_R increases by $\frac{1}{r}$ each time the robot returns to the shared workspace and increases by 1 unit every time the robot finishes bringing all r parts to the shared workspace.

In Section 3.3.2, from $v = \dot{s}||\boldsymbol{t_p}(s)||$ it follows that $\dot{s}(t) = \frac{v(t)}{\|\boldsymbol{t_p}(s(t))\|},\ 0 \leqslant v(t) \leqslant \overline{v_R}$, where

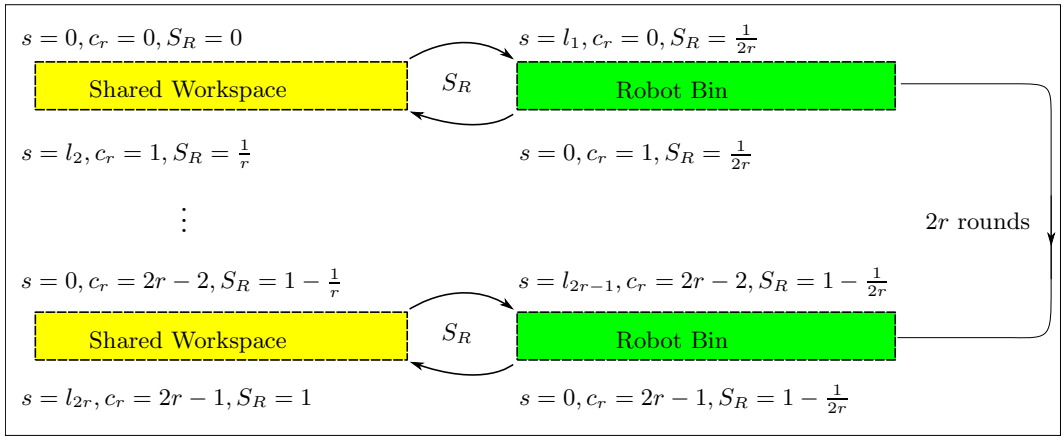


Figure 3.6: Demonstration of robot progress, S_R, for assembling the first product. This process is repeated N_p times.

$v(t)$ and $\overline{v_R}$ are the control input and its maximal value, respectively. Without loss of generality, we set $\overline{v_R} = 1$ representing the highest speed achievable by the robot. The kinematics of robot path progress, S_R, is then given by

$$\dot{S}_R(t) = \frac{\dot{s}(t)}{2rl_i} = \frac{v(t)}{2rl_i\|\boldsymbol{t_p}(s(t))\|}, \quad 0 \leqslant v(t) \leqslant \overline{v_R}.$$

Since digital controllers and sensors work in the discrete-time settings, we next consider and implement our system kinematics in discrete-time:

$$S_R(k+1) = \frac{v(k)T_s}{2rl_i\|\boldsymbol{t_p}(s(k))\|} + S_R(k), \ 0 \leqslant v(k) \leqslant \overline{v_R}, \tag{3.13}$$

where T_s and k are the sampling time and time step, respectively. Here, $S_R((k+1)T_s)$ and $S_R(kT_s)$ are written as $S_R(k+1)$ and $S_R(k)$, respectively, for the sake of simplicity.

Similarly, the path progress made by human, denoted as S_H, can be defined according to (3.13) but with a slight modification. Human hand motion has uncertainties and does not follow a specific path in general. However, the start and end points of the motion are fixed and located at the shared workspace and the human bin. We denote the length between these fixed points as l_h and consider the line that connects these two points as the human reference line. The human hand position can be measured by the PhaseSpace motion capture system as shown in Fig. 3.2. Let us denote the position of the human hand as $\boldsymbol{p_h}(k) = [x_h(k),\ y_h(k),\ z_h(k)]^T$, the position of the start point of the reference line as $\boldsymbol{p_0}(k) = [x_o(k),\ y_o(k),\ z_o(k)]^T$, and the position of the end point as

$\boldsymbol{p_f}(k) = [x_f(k),\ y_f(k),\ z_f(k)]^T$. Denote the vector that connects the start and end points of the reference line and its unit vector as $\boldsymbol{i_h}(k) = \boldsymbol{p_f}(k) - \boldsymbol{p_0}(k)$ and $\hat{\boldsymbol{i}}_{\boldsymbol{h}}(k)$, respectively. The position of the human hand along the reference line at time step k can then be computed as $\hat{\boldsymbol{i}}_{\boldsymbol{h}}(k)(\boldsymbol{p_h}(k) - \boldsymbol{p_0}(k))$. Human path progress, $S_H \in \mathbb{R}^+$, is defined as

$$S_H(k+1) = \frac{\hat{\boldsymbol{i}}_{\boldsymbol{h}}(k)(\boldsymbol{p_h}(k) - \boldsymbol{p_0}(k))}{2hl_h} + \frac{c_h}{2h}, \tag{3.14}$$

where c_h is the number of previously traveled paths by human between the shared workspace and the human bin. The first term in (3.14) is the ratio that the human hand has moved along the current reference line so far with respect to the total length that his/her hand is required to travel for bringing h parts to the shared workspace. Similar to c_r, c_h increases by 1 after each time the human reaches the shared workspace or the human bin. The details for estimating $\boldsymbol{p_h}$ are explained in the next section.

In the pHRI system, it is desired that the robot path progress follows the human path progress efficiently. This can be formulated as the following NMPC problem:

$$\min_{v(0),\ldots,v(N-1)} \sum_{k=1}^{N} \{||S_R(k) - S_H(k)||_Q + ||v(k) - \overline{v_R}||_R\}, \tag{3.15}$$

subject to (3.13) and (3.14), where N is the prediction horizon and $||\cdot||_Q$ ($||\cdot||_R$) represents the weighted norm with respect to the positive number Q (R). The first term in (3.15) addresses the human-robot synchronization of motion progress, i.e. it is desired that $S_R = S_H$ during the pick-and-place operations so that the robot brings the required assembly parts to the human in time. The second term seeks to maximize robot efficiency. Minimizing the cost has the effect of pushing S_R to S_H and v to $\overline{v_R}$. Since $v(k) \geq 0$, S_R is non-decreasing according to (3.13). If the value of Q is considerably higher than R, then the robot velocity will decrease and eventually stop when the robot progress gets ahead of human progress, i.e. $S_R(k) > S_H(k)$. These stops make the robot motion non-smooth and annoying for the participant, and thus will impact the experiments in a negative way. If the value of Q is considerably lower than R, then the robot velocity is always close to $\overline{v_R}$. This also impacts the experiments in a negative way since the robot does not adapt to the human. Hence, equal weighting of Q and R should be chosen to ensure that the robot adaptability with human motion progress and efficiency impose equal effects on the cost evaluation.

Remark 4. Here, the assembly of car center console is considered as a task that human and robot accomplish together. Nonetheless, this framework can be applied to other human-robot collaborative tasks such as hose assembly and car door assembly where the progress synchronization for pick-and-place is important. •

3.4.2 Human Kinematics Learning

We now provide details of the estimation of human hand kinematics. It should be noted that the human kinematics are required in order to predict the motion of human worker over a specified time horizon which will be used to design robot motion controller accordingly. Human hand position at the next time step can be predicted using its current velocity and position. Thus, to estimate the future values of human hand position over the horizon N in (3.15) and (3.23), the value of $\boldsymbol{v_h}$ is required. In this paper, a recursive least-square (RLS) based black-box approach is exerted in order to obviate the problem of considering the human kinematics directly and through a model-based perspective. As reported in [113], there exist many advantages for this learning algorithm including low computational burden, fast convergence to the solution, and unbiasedness when it is subject to the white noise. Our goal is to estimate the value of the human worker's hand velocity, $\boldsymbol{v_h}$ based on

$$\boldsymbol{v_h}(k+1) = \boldsymbol{\theta}^T \boldsymbol{\Phi}(k), \tag{3.16}$$

where $\boldsymbol{\theta}$ is the vector with real coefficients and $\boldsymbol{\Phi}(k)$ is the matrix with the input and past output values of the system [113]. At each time step, we consider $\boldsymbol{\Phi}(k)$ as:

$$\boldsymbol{\Phi}(k) = \begin{bmatrix} \begin{bmatrix} -\boldsymbol{v_h}(k) & \dots & -\boldsymbol{v_h}(k+1-i) \end{bmatrix}^T \\ \begin{bmatrix} \boldsymbol{p_h}(k) & \dots & \boldsymbol{p_h}(k-j) \end{bmatrix}^T \end{bmatrix}, \tag{3.17}$$

where $i \geq 1$ and $j \geq 1$ are arbitrary numbers, k indicates the present time step, and $k-o$ represents o time steps before the present. The input and output data are the position and velocity of the human hand, respectively. Obviously, the elements of $\boldsymbol{\Phi}$ consist of the previous outputs and inputs data as well as the present input value. The estimated value of human hand velocity, $\hat{\boldsymbol{v}}_{\boldsymbol{h}}(k+1)$ can be found by $\hat{\boldsymbol{v}}_{\boldsymbol{h}}(k+1) = \hat{\boldsymbol{\theta}}^T(k)\boldsymbol{\Phi}(k)$, where $\hat{\boldsymbol{\theta}}(k)$ is the estimated value of the filter at time step k. Based on the least square method, the following cost function should be minimized for

estimating $\boldsymbol{v_h}(k+1)$: $J(k) = \sum_{i=1}^{k} \left[\boldsymbol{v_h}(i) - \hat{\boldsymbol{\theta}}^T(k)\boldsymbol{\Phi}(i-1)\right]^2$. Instead of solving this equation, to reduce the computational burden, the RLS method can be used to find the filter recursively through the following equations [113]:

$$\begin{aligned}
&\hat{\boldsymbol{\theta}}(k+1) = \hat{\boldsymbol{\theta}}(k) + K(k)(\boldsymbol{v_h}(k+1) - \hat{\boldsymbol{\theta}}^T(k)\boldsymbol{\Phi}(k)) \\
&K(k) = \boldsymbol{F}(k)\boldsymbol{\Phi}(k)\left[1 + \boldsymbol{\Phi}^T(k)\boldsymbol{F}(k)\boldsymbol{\Phi}(k)\right]^{-1} \\
&\boldsymbol{F}(k+1) = (\boldsymbol{I} - K(k)\boldsymbol{\Phi}^T(k))\boldsymbol{F}(k),
\end{aligned} \tag{3.18}$$

where $\boldsymbol{I}$ is the identity matrix. The above process is repeated until the termination condition $\|\hat{\boldsymbol{\theta}}(k) - \hat{\boldsymbol{\theta}}(k-1)\| \leq \epsilon$ is satisfied, where ϵ is a sufficiently small positive value. For estimating the value of $\boldsymbol{p_h}$ at the future time steps which is needed for solving (3.15) over the horizon N, the following equations are used to recursively update the output values over the look-ahead horizon for time step $k+m,\ m = 1, 2, \cdots, N$:

$$\begin{aligned}
&\hat{\boldsymbol{p}}_{\boldsymbol{h}}(k+m) = \hat{\boldsymbol{v}}_{\boldsymbol{h}}(k+m-1)T_s + \hat{\boldsymbol{p}}_{\boldsymbol{h}}(k+m-1) \\
&\hat{\boldsymbol{v}}_{\boldsymbol{h}}(k+m) = \hat{\boldsymbol{\theta}}^T(k)\hat{\boldsymbol{\Phi}}(k+m-1) \\
&\hat{\boldsymbol{\Phi}}(k+m) = \begin{bmatrix} \begin{bmatrix} -\hat{\boldsymbol{v}}_{\boldsymbol{h}}(k+m) & \dots & -\hat{\boldsymbol{v}}_{\boldsymbol{h}}(k+m+1-i) \end{bmatrix}^T \\ \begin{bmatrix} \hat{\boldsymbol{p}}_{\boldsymbol{h}}(k+m) & \dots & \hat{\boldsymbol{p}}_{\boldsymbol{h}}(k+m-j) \end{bmatrix}^T \end{bmatrix}.
\end{aligned} \tag{3.19}$$

3.4.3 Human Trust in Robot

We now extend the optimal control formulation (3.15) by further considering human trust in robot to assure smooth and effective HRC while the assembly is efficient. Based on previous studies in human factors [49, 30], human trust in robot depends on prior trust, robot performance, human performance, and fault occurrences. In this paper, we utilize our previous results [91] of a time-series dynamic model of human-to-robot trust in HRC manufacturing. Note that in manufacturing environments, the required tasks of the robot and the environment itself are fixed, and thus it is reasonable to assume that the robot performs the tasks as planned and the main performance metric is the robot flexibility. With that mindset, we use the following computational model for

human-to-robot trust, denoted as $T(k)$:

$$T(k) = aT(k-1) + bP_R(k-1) + cP_H(k-1), \tag{3.20}$$

where P_R and P_H represent robot and human performance, respectively. The coefficients a, b, and c are constants to be determined for a specific application and individual. As we described in [91], a common method for determining these parameters is to use the Autoregressive Moving Average (ARMA) Model for the data collected during the training of the experiments. First, human and robot performances and human trust in robot are collected. The performances can be measured objectively but the human trust can only be realized subjectively. Since it is difficult to ask for subjective human trust to the robot with a high sampling frequency, this value is asked from the human worker after he/she finishes each task trial, i.e. assembles one product. Next, the trust value at each trial and the average value of performances during that trial are stored. Finally, Using ARMA in MATLAB System Identification Toolbox, the coefficients a, b, and c are fitted to (3.20) using the stored data for each individual participant in the given task. In this paper, we consider human working speed and his/her coordination with the robot coworker as the two main measures of performance, P_H. We asked two human workers who are experts in performing assembly tasks to perform the same set of operations and collected their data as the reference human working speed. This ideal speed is a function of the path progress and we denote it as $v_{ref}(S) \in [0, 1]$, where $S \in \mathbb{R}^+$ is the path progress (of the human, S_H, or the robot, S_R). Any difference between the human working speed and this reference value along the path indicates low human performance. Moreover, if the human path progress, S_H, is less than that of robot, S_R, the human is considered to not do a good job compared to his/her robot coworker. We define P_H as

$$P_H(k+1) = \overline{P_H} - w_1(S_R(k) - S_H(k))H(S_R(k) - S_H(k)) - w_2 \left| \frac{\hat{\dot{\boldsymbol{i}}}_{\boldsymbol{h}} \boldsymbol{v_h}(k)}{\overline{v_H}} - v_{ref}\left(S_H(k)\right) \right|, \tag{3.21}$$

where $\overline{P_H} = 1$ and $\overline{v_H}$ are the maximal values of human performance and human hand velocity, with $w_1 + w_2 = 1$, $w_i \in [0, 1], i = 1, 2$, and $H(.)$ is the Heaviside step function. The robot is desired to follow the human progress during the interaction. Thus, we define P_H such that it does not decrease if the human progress leads, i.e. if $S_H > S_R$. The robot performance is defined in a fashion similar to the human performance by including both robot working speed and its flexibility in keeping up

with the human co-worker. Here, any difference between S_R and S_H results in low P_R since the robot is required to follow the human worker's progress. The definition of P_R is hence

$$P_R(k+1) = \overline{P_R} - w_3|S_R(k) - S_H(k)| - w_4 \left| \frac{v(k)}{\overline{v_R}} - v_{ref}(S_R(k)) \right|, \tag{3.22}$$

where $\overline{P_R} = 1$ is the maximal value of robot performance with $w_3 + w_4 = 1$, $w_i \in [0, 1], i = 3, 4$.

Remark 5. In both human and robot performance models, the first measure depicts the quality harmony between the two agents. Lack of coordination of the progress of the agents results in frustration, higher error failure rate, and a decline of overall performance of the human-robot team [10]. However, the coordination between the agents are not sufficient. If both agents perform the task with harmony but slowly, the overall progress would be slow. Hence, the second measure depicts each agent's individual progress rate towards finishing the task. In summary, the agent performance is high only if it performs the task with a fast pace and in accordance with the other agent. •

3.4.4 Integrated Human-Robot Collaboration (HRC) System

In the integrated HRC system, the robot is required to meet the human expectations and preferences. Thus, it is desired to control the robot speed such that it follows the human path progress while human trust in robot is higher than a threshold value. This can be formulated as the following NMPC problem:

$$\min_{v(0),\ldots,v(N-1)} \sum_{i=1}^{N} \{||S_R(i) - S_H(i)||_Q + ||v(i) - \overline{v}||_R + ||T(i) - \overline{T}||_W\}, \tag{3.23}$$

subject to (3.13), (3.14), (3.20), (3.21), (3.22), and $T(i) > \underline{T}$, where $\underline{T}$ and $\overline{T}$ are the minimal threshold and maximal values of trust determined based on task specifications and individual preferences [91]. Here, minimizing cost has similar effects as of (3.15) and also pushes T to $\overline{T}$. As explained in Section 3.3.1, a high-level path planning approach is developed based on trust of human in robot. We define a look up table for selecting the robot path based on the average trust of human in robot, T_{avg}. This table specifies a path corresponding to each range of trust values. At the end of each travel (i.e. at the initial and final positions), the high-level path planner matches T_{avg} with the ranges in the table and selects the corresponding path.

3.4.5 Integrated Human-Robot Collaboration (HRC) System with Emotional Expressions

Emotions help people to interact with each other more naturally and intuitively. Many studies showed that cognition and emotion play interrelated roles in intelligent decision-making, planning, communication, social interaction, etc. [9]. For example, emotion helps to prioritize different concerns by guiding the attention towards important matters and away from distractions [83]. The possible benefits of integrating emotion into the robots resulted in the design of emotion-inspired mechanisms such as Kimset, ERWIN, Kobian, NAO, Flobi, iCAT, Robokinds, and Geminoids [7, 72]. While these robots are mostly utilized for social services [21], their applications can be effectively extended to other HRI scenarios. Thus, we borrow this idea to design an emotion-inspired robot in a hybrid manufacturing cell. In this setting, the robot emotion plays a role of non-verbal communication, informing the human about a possible safety or efficiency concern, and thus increases safety and performance.

The robot emotion is added to the integrated HRC system to make the interaction more intuitive and to alert safety or efficiency concerns to the human worker by displaying a facial expression (see Figs 3.7 and 3.9) both on the robot head screen and a computer information screen. To make the interaction more human-like, we also add eye motion to the robot facial expression and let it follow the human hand all the time. Under the nominal condition when there is no chance of immediate collision and the human performance is relatively coordinated with the robot, the robot expresses a happy face. However, the robot end-effector might collide with the human hand if the distance between them is small. We call this distance the safety index and define it as $I_S = |\boldsymbol{x} - \boldsymbol{p_h}|$, where $\boldsymbol{x}$ and $\boldsymbol{p_h}$ are the position of the robot's end-effector and human worker's hand, respectively. We define L_S as the threshold value of I_S for the safe interaction of the human and the robot. When $I_S > L_S$, the robot and information screens display a happy or bored face depending on the human and robot progress. If there is a possible collision between the human and the robot, i.e. $I_S \leq L_S$, the robot stops working and a worried face will be displayed on the robot and information screens. As soon as the human worker moves his/her hand away from the robot and $I_S > L_S$, the robot facial expression changes from worried to either happy or bored emotion and the robot will continue to move on the planned path. We also denote the difference between the robot and human progress as the efficiency index $I_E = S_R - S_H$. Let $L_E = .5$ be the threshold value of I_E for efficient interaction

(a) Happy Face (b) Worried Face (c) Bored Face

Figure 3.7: Robot emotions (facial expressions).

of the human and the robot. If $I_E > L_E$ and the robot's relative distance to human is in the safe region, the robot progress is remarkably higher than human and Baxter displays a bored face to the human expressing that his/her progress is too slow. This emotion contributes to efficiency by encouraging the participant to keep pace with the robot.

3.4.6 The Control Framework Diagram

Fig. 3.8 shows the block diagram connecting different HRC system components. All of the programs communicate through ROS. The robot controller is coded in Python. It receives the robot joint positions, $\boldsymbol{q}$, and path position velocity, v, from Baxter and the NMPC solver, respectively. It calculates the robot path progress, S_R, and corresponding joint velocities, $\dot{\boldsymbol{q}}$, using v. These joint velocity commands are sent to Baxter. The human kinematics estimator is coded in C++ and receives the time and 3D position data from the PhaseSpace tracking system workstation. It calculates human path progress, S_H, the regressand vector, $\boldsymbol{\Phi}$, and estimates the filter vector, $\boldsymbol{\theta}$. Computational trust simulator is used for the integrated experiment. The input is the human progress data S_H from the estimator, robot translational velocity v from the NMPC solver, and robot progress S_R from the robot controller. The trust simulator uses the input data and returns the human performance, P_H, robot performance, P_R, and estimated computational trust, T as the output. We add ROS interface to the NMPC toolbox written in C++ [24]. It receives all of the data and calculates the next control input, v. The facial expressions block uses values of robot end-effector position $\boldsymbol{x}$, human's hand position $\boldsymbol{p_h}$, robot progress S_R, and human progress S_H, for calculating the safety and efficiency indexes.

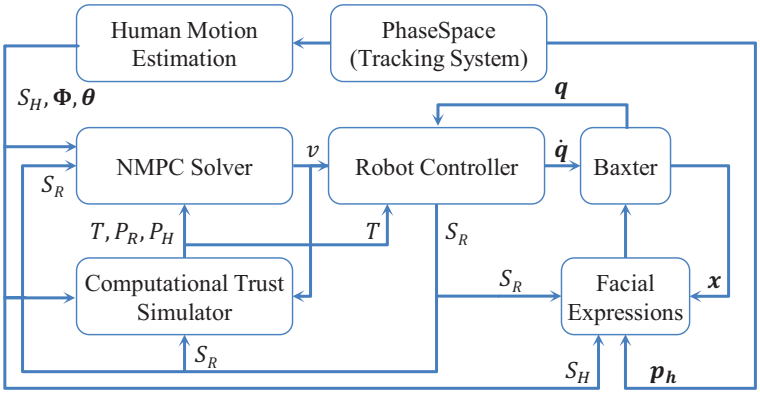


Figure 3.8: Block diagram of the HRC control framework.

3.4.7 Transparency

To keep the human aware about the robot actions and reduce confusion, we improve transparency by showing different states of the HRC systems (Section 3.4) used for controlling the robot, speed of the robot v, and computational trust of human to the robot T, through an information screen. If the speed of the robot is adjusted manually, only the robot velocity v, human progress S_H, and robot progress S_R, will be shown on the screen. Fig. 3.9(b) shows this interface for the integrated HRC system with emotional expressions.

3.5 Experiments with a Human-in-the-Loop

3.5.1 Experimental Design and Participants

We evaluated the effects of implementation of the proposed control conditions through an experimental case study. A within-subject test with Latin square design test order was performed under four different control conditions: (1) manual control condition (**C1**), (2) pHRI-based control condition (**C2**), (3) integrated pHRI- and trust-based control condition (**C3** or integrated in short), and (4) integrated pHRI- and sHRI-based control condition considering both trust and emotion display (**C4** or emotion-integrated in short). Twenty participants (6 female and 14 male) with an age ranging from 25 to 36 (average 29.9) years, participated in the experiments. Similar sample size has been used in prior works on experimental studies of human-robot interaction [110, 35, 64]. Half of the participants had no experience in working with a robot before. Based on the preliminary results of our previous work [93] and the discussed literature in Sections 3.1 and 3.2 for the collaboration

of a human and a robot under these control conditions, we hypothesize that

1. **H1**. As we move from **C1** to **C4**, the human perceived workload decreases.
2. **H2**. As we move from **C1** to **C4**, the human perceives higher trust towards the robot.
3. **H3**. As we move from **C1** to **C4**, the human perceives a higher usability of the robot.
4. **H4**. The robot average velocity and the assembly time do not change significantly in all conditions.

3.5.2 Measurements and Scales

The independent variable of this study is the control condition (**C1-C4**) utilized based on the level of the interaction. The dependent variables are the following subjective and objective measures:

- Workload: At the end of each experiment, the subjective overall workload was measured using the NASA Task Load Index (TLX) method [31]. This measure can vary from 5 to 100.
- Subjective Trust: At the end of each experiment, the subjective trust of the participant in robot were measured using a human trust in automation questionnaire [38]. The questionnaire was adjusted to be suitable for assessments of robots.
- User Satisfaction: At the end of each experiment, the subjective satisfaction with the usability of the robot was measured using the IBM usability satisfaction questionnaire [54].
- Robot Velocity: The average robot velocity (v_{avg}) is a measure of the efficiency of the human-robot team in accomplishing the task and was calculated after the experiment was finished.
- Assembly Time: This is the time spent to assemble $N_p = 3$ products, used as another measure of efficiency.

3.5.3 Apparatus

Fig. 3.2 shows the equipment and framework architecture used for the experiment. We used a humanoid manufacturing research robot, Baxter, made by Rethink Robotics for our experiment. The robot is suitable for light-weight material handling and intelligent assembly, especially for small

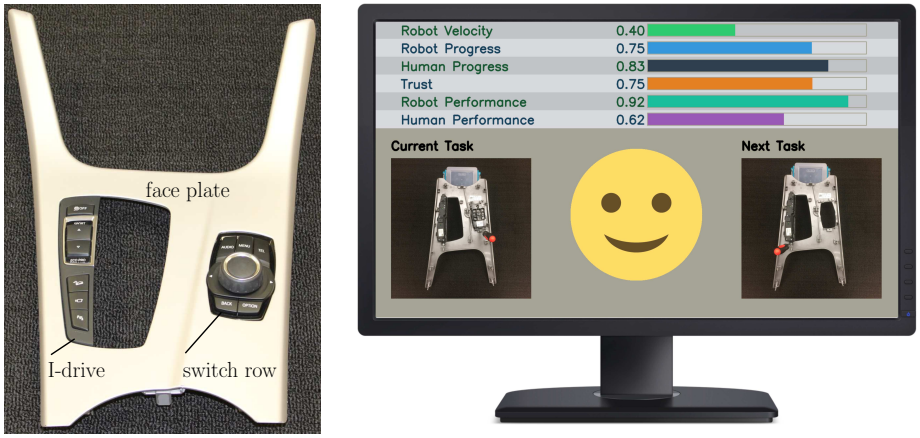
batch productions. It has a removable base and two redundant arms with 7 Degrees-of-Freedom (DoFs) on each arm. There is a rotary screen attached to the top of the robot as its head. We use the PhaseSpace motion tracking system to capture the human hand motion. The tracking system includes a set of cameras, a set of active markers, and a workstation for tracking rigid bodies in a 3D environment. The other nodes of the framework were set up on a local computer as described in Section 3.4.6.

3.5.4 Task Scenario

The selected task in the experiment is from the automotive assembly industry and the goal is to assemble three parts together to form a BMW center console. This task is similar to the assembly task that a human worker performs in the manufacturing assembly lines and requires a high level of HRC in which a human and a robot work together in a hybrid-cell [105] as shown in Fig. 3.1(b). The assembly parts include face plate, I-drive, and switch row (Fig. 3.9(a)). Each participant was asked to continuously assemble these parts together. The current and next assembly parts required to be assembled, together with some other task information, are shown to the participant via the information screen (see Fig. 3.9(b)). In the current settings, it is difficult for the robot to grab the face plate and bring it to the human due to its geometry. Therefore, the participant was required to fetch the face plate from the human bin and place it in the shared workspace. At the same time, the robot fetched the I-drive and switch row from the robot bin to the shared workspace. The participant assembled these two parts on the face plate using a screwdriver to form the final car center console. This process was repeated 3 times ($N_p = 3, h = 1, r = 2$) for each of the 4 control conditions of the experiment. Fig. 3.10 shows the collaboration of a participant and the robot during an experiment under the emotion-integrated control condition (**C4**). A brief summary of the task procedure for the integrated control condition is provided in Algorithm 2.

3.5.5 Experiment Procedure

Each participant was asked to fill out the consent form and demographic questionnaire. The information regarding the task scenario and the roles of the participant and the robot were explained in details to the participant. For training purpose, the participant was asked to collaborate with the robot until he/she feels comfortable and familiar with the task and the robot. After the training,



(a) Assembly parts (b) Information Screen

Figure 3.9: The completed assembly and information screen

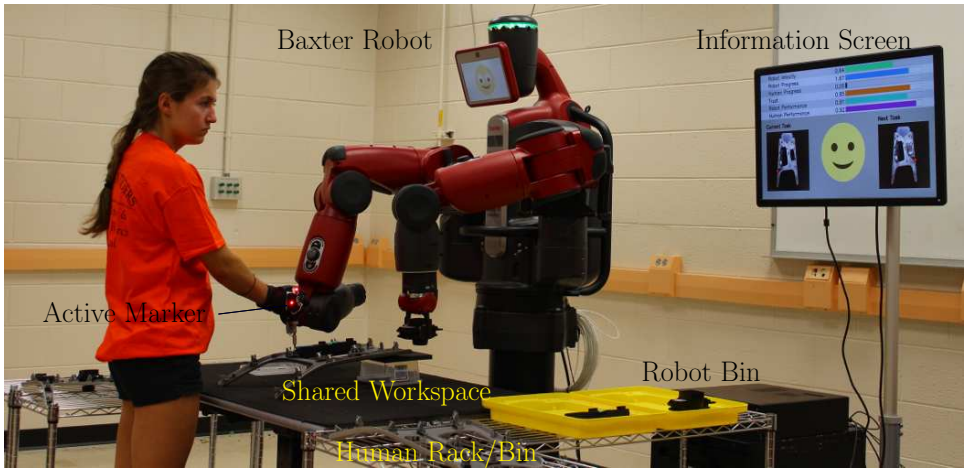


Figure 3.10: The experiment scenario.

the participant performed all four conditions of the experiment based on Latin square order. Upon completion of each condition, the participant was asked to fill out NASA-TLX, trust, and IBM usability questionnaires.

3.6 Results, Analysis, and Discussion

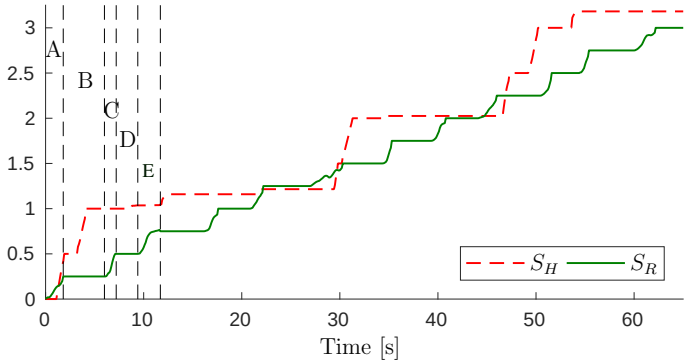
3.6.1 Sample HRC System Evolution

Fig. 3.11 shows a sample HRC system evolution for participant number 15 under emotion-integrated condition (**C4**). The robot performance P_R and robot path progress S_R do not change

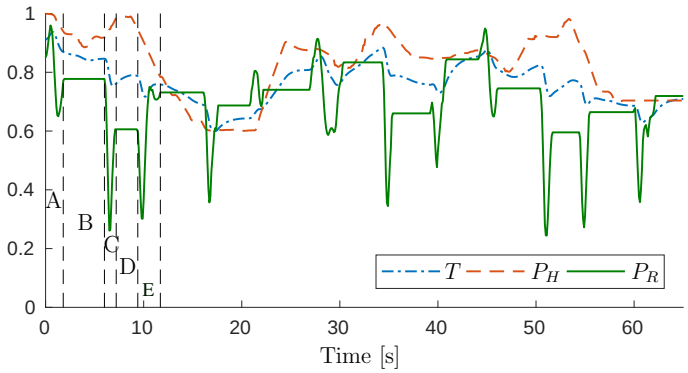
Algorithm 2 implementation of the integrated control condition in the hybrid cell

1: Move to the shared workspace ▷ Initial point
2: $c_r \leftarrow 0$ ▷ Reset the number of traveled paths
3: Subscribe(T) ▷ Subscribe to the trust simulator
4: $T_{\text{avg}} \leftarrow T$ ▷ Initialize the average trust value
5: $\boldsymbol{p} \leftarrow choose\_path\_fwd(T_{\text{avg}})$ ▷ Select forward path
6: **repeat**
7: $Move\_Robot(\boldsymbol{p})$ ▷ Move to the robot bin
8: $Pick\ up$ ▷ Pick up a part from the robot bin
9: $c_r \leftarrow c_r + 1$ ▷ Increase number of traveled paths
10: $Update(T_{\text{avg}})$
11: $\boldsymbol{p} \leftarrow choose\_path\_bwd(T_{\text{avg}})$ ▷ Select backward path
12: $Move\_Robot(\boldsymbol{p})$ ▷ Move to the shared workspace
13: $Place$ ▷ Place the part near the worker
14: $Update(T_{\text{avg}})$
15: $\boldsymbol{p} \leftarrow choose\_path\_fwd(T_{\text{avg}})$ ▷ Select forward path
16: $c_r \leftarrow c_r + 1$ ▷ Increase number of traveled paths
17: **until** $c_r == 2r \times N_p$

at the pick-and-place locations where the robot end-effector only moves vertically. At the start of the first task (Region A), both human and robot performances are high. The human moves toward the human bin to bring the face plate ($S_H = .5$) while the robot moves toward the robot bin to bring the next part ($S_R = .25$) to the shared workspace. As the human progress reaches $S_H = .5$, the difference in human and robot motion progress results in decrease of the robot performance P_R, which results in lower trust value as well. When the robot reaches the robot bin (Region B), it picks up a part (an I-drive). At the same time, human reaches the shared workspace and since S_H is greater than S_R, P_H does not change significantly. However, as the human velocity deviates from the reference velocity, P_H decreases slightly. The trust of human in robot has the same trend. Next, the robot starts to move back to the shared workspace (Region C). At this point, the robot performance gets updated. Since there is a major difference between human and robot progress, the robot performance P_R drops significantly at first and then recovers when the robot reaches the shared workspace where $S_R = .5$. As the robot places the object (Region D), human path progress is greater than robot progress. Moreover, human velocity matches the reference velocity and hence the human performance P_H is high and trust increases. When the robot starts to moves back for the next part (a switch row) (Region E), the discrepancy between the human and robot path progress is high at first but decreases and robot performance recovers. Similar trends repeat until the task is accomplished completely.

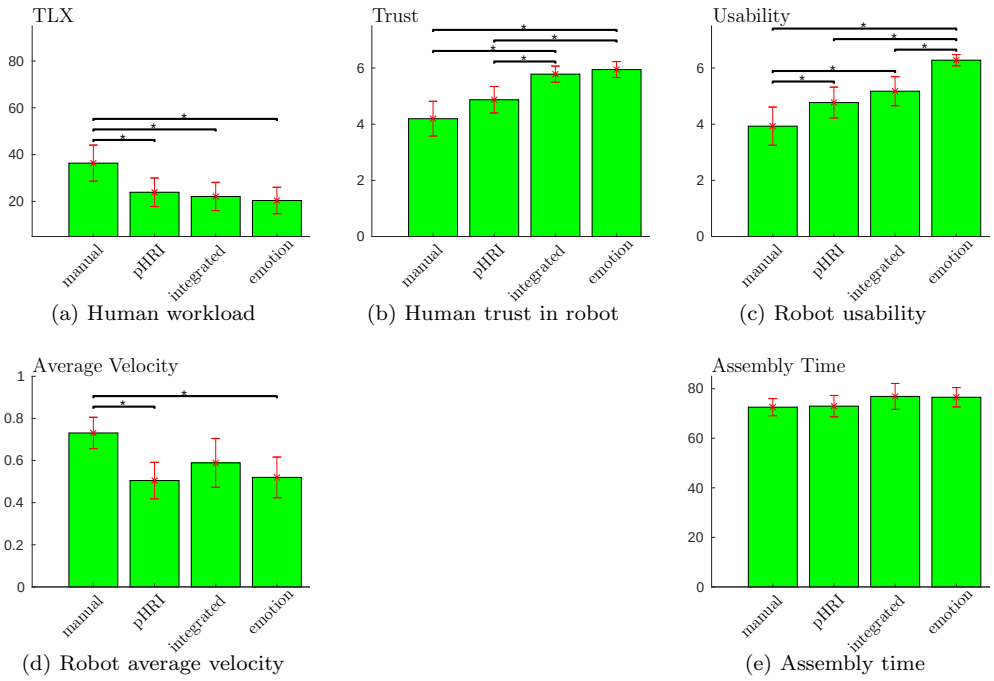


(a) Human and robot progress



(b) Trust, human and robot performances

Figure 3.11: Sample HRC system.



(a) Human workload (b) Human trust in robot (c) Robot usability

(d) Robot average velocity (e) Assembly time

Figure 3.12: Results with statistical significant changes (*).

3.6.2 Statistical Analysis

For each dependent variable explained in Section 3.5.2, a one-way repeated measure analysis of variance (ANOVA) was conducted to determine whether there were statistically significant differences in dependent variables over the interaction type. Mauchly's test of sphericity shows that the assumption of sphericity was violated for human trust in robot ($\chi^2(2) = 26.626$, $p < .001$), robot usability ($\chi^2(2) = 16.317$, $p = .006$), and perceived workload index ($\chi^2(2) = 31.523$), $p < .001$, but it is valid for robot average velocity ($\chi^2(2) = 6.744$, $p = .241$), and assembly time ($\chi^2(2) = 10.846$), $p = .055$. Therefore, Greenhouse-Geisser correction was applied ($\epsilon = .567$, $\epsilon = .634$, $\epsilon = .509$ for human trust-in robot, robot usability, and perceived workload index, respectively). The results with upper and lower bounds for the 95% confidence interval are presented in Fig. 3.12 and described as follows.

3.6.2.1 Perceived Workload

The manipulation of the interaction type elicited statistically significant changes in workload between the different conditions, $F(1.526, 29.002) = 8.930$, $p < .005$, $\eta_p^2 = .320$. Post-hoc analysis

with a Bonferreoni adjustment showed that the workload under the manual condition ($M = 36.367$, $SD = 3.686$) is significantly higher compared to the other conditions (pHRI ($M = 23.917$, $SD = 2.915$), integrated ($M = 22.117$, $SD = 2.879$), and emotion-integrated ($M = 20.383$, $SD = 2.700$)).

3.6.2.2 Human Trust in Robot

The manipulation of the interaction type elicited statistically significant changes in trust, $F(1.702, 32.338) = 17.839$, $p < .001$, $\eta_p^2 = .484$. Post-hoc analysis with a Bonferreoni adjustment showed that trust both under the integrated ($M = 5.783$, $SD = .137$) and emotion-integrated ($M = 5.946$, $SD = .136$) conditions are significantly higher compared to the trust under the manual ($M = 4.196$, $SD = .296$) and pHRI ($M = 4.874$, $SD = .225$) conditions.

3.6.2.3 Usability

The manipulation of the interaction type elicited statistically significant changes in usability, $F(1.901, 36.116) = 28.671$, $p < .001$, $\eta_p^2 = .601$. Post-hoc analysis with a Bonferreoni adjustment showed that usability under the manual condition ($M = 3.930$, $SD = .325$) is significantly lower compared to the usability under the other conditions including pHRI ($M = 4.770$, $SD = .263$), integrated ($M = 5.175$, $SD = .248$), and emotion-integrated ($M = 6.280$, $SD = .096$). Moreover, it shows that usability under the emotion-integrated conditions is significantly higher compared to all of the other conditions.

3.6.2.4 Robot Average Velocity and Assembly Time

The manipulation of the interaction type elicited statistically significant changes in robot average velocity, $F(3, 57) = 7.299$, $p < .001$, $\eta_p^2 = .278$. Post-hoc analysis with a Bonferreoni adjustment showed that the robot average velocity under the manual condition ($M = .731$, $SD = .036$) is significantly higher compared to the pHRI ($M = .505$, $SD = .041$), and integrated-emotion ($M = .520$, $SD = .046$) conditions. The interaction type did not elicit statistically significant changes in assembly time, $F(3, 57) = 2.091$, $p > .05$, $\eta_p^2 = .099$.

3.6.3 Discussion

It can be seen from the results that in general, as we augment physical and social capabilities into the framework and change the control condition from **C1** to **C4**, HRI improves and at least one

of the subjective measures (i.e. human perceived workload, human trust in robot, robot usability) improves. More specifically, although none of the hypotheses are completely true, they are partially correct. In **H1**, the human perceived workload does not constantly drop from **C1** to **C4**. However, compared to the manual adjustment of robot speed (**C1**), using any autonomous controllers (**C2** to **C4**) decreases the human perceived workload significantly. Two main reasons for this could be that under **C1** a participant has to either adapt his/her work pattern to that of the robot or pay more attention to the robot velocity and adjust it accordingly. Both of these result in more perceived workload of the participant.

Regarding **H2**, considering merely pHRI for controlling the robot velocity does not impact the human trust in the robot. However, compared to **C1** and **C2** frameworks, as we augment more social capabilities to **C3** and **C4**, trust of human in robot significantly improves. The main reason for this significant change might be that the participant becomes more confident in robot since the human and robot performances are shown to them in both **C3** and **C4**. The trust-based controller and path selection improves the trust by clarifying robot's intent and behaving more predictably and human-like.

The impact on robot usability is more obvious relative to the other subjective measures. Compared to **C1**, in any framework with autonomous controller (**C2** to **C4**), the usability increases. This, together with **H1**, suggests that an autonomous controller results in higher usability and lower perceived workload. Moreover, the utilization of emotions in **C4** results in higher robot usability compared to the rest of the conditions (**C1** to **C3**). This is probably because adding emotions makes the interaction more human-like and appealing.

In terms of efficiency, the results indicate that **H4** is not completely true. The robot average velocity is significantly higher in **C1** compared to **C2** and **C4**. We observed two major trends in **C1**: (i) some of the participants felt competitive towards the robot and intentionally tried to challenge themselves and the robot by setting the robot velocity at a high value rather than adjusting it based on their speed, and (ii) some of them simply were too engaged in the experiment and forgot to adjust the robot speed. These observations also justify the high value of human workload in **C1**. Note that the rest of the participants paid more attention to the robot velocity to determine the speed adjustments, so they also perceived high workload. The variation of assembly time in different conditions match **H4** and does not change significantly. This is probably because although participants used higher robot velocities in **C1**, they could not keep up with the robot progress and

had to spend some time at the end of the experiment to finish the assembly tasks.

3.7 Conclusion

We proposed a novel framework for HRC in manufacturing assembly lines. We described the kinematics of such HRC system and demonstrated how this framework includes both pHRI and sHRI for finding the optimal velocity of the robot. We also demonstrated the problem of moving the robot end-effector with an arbitrary velocity along a given path and implemented the control method for the HRC system. We experimentally evaluated this framework by designing an HRC testbed. The results show that the pHRI- and sHRI-based autonomous controllers can reduce human workload while maintaining the overall performance of the human-robot team compared to the manual adjustments of the robot velocity. Moreover, it is shown in our experiments that human trust in robot can be remarkably increased if sHRI factors are integrated into the pHRI-based framework. Furthermore, the robot usability can be significantly increased if emotion is added to the integrated framework while the objective measures do not show statistical significance among the automated conditions.

Chapter 4

Trust-Triggered Robot-Human Handover for Collaborative Assembly

4.1 Introduction

In this chapter, we present an experimental study based on the trust-based handover strategy. Studies on human-human handover show that during a handover people adjust their approaching posture based on their level of trust in their partners [3] to minimize the effects of impact forces in the direction of the handover [34]. A similar approach can be applied to the robot to human handover through minimizing the effects of impact forces exploiting robot's kinematics redundancy. We hypothesize that if the robot trust to the human reduces due to the human fault or low performance, the human may not be prepared for the handover and may develop error in motion planning for receiving the payload from the robot [88]. Section 4.2 explains the task scenario and hand over the task for the experimental setup shown in Fig. 4.1. Section 4.3 presents the robot to human trust model and measurement. The trust-based handover strategy is explained in Section 4.4. The experimental evaluation of the handover strategy is presented in 4.6. The chapter is concluded in Section 4.7.

4.2 The Task Description and The Handover

We focus on HRC in assembly tasks in small-scale flexible manufacturing processes that may be unstructured and requirements of materials, resources and equipment may be less predictable due to frequent changes in assembly processes and product requirements. Here, we consider the hose assembly as an example for HRC. Currently, the human manually fetches the hoses and the fitting parts (end hoses) as shown in Fig. 4.1, and assembles these together. The human adjusts the length of the hoses cutting these with a cutter if the length is found larger than the reference length. We propose that this task can be performed by a human in collaboration with a collaborative robot. We develop another hybrid cell to illustrate how the hose assembly task can be performed in collaboration between a human and a robot.

4.2.1 Task Scenario

We assume that the assembly task can be segmented into several subtasks that can be assigned to the human, the robot or both. The agents (human, robot) are to perform the subtasks assigned to them sequentially keeping pace with each other. The human uses his/her dexterous skills to perform the assigned subtasks. However, the robot must be equipped with appropriate sensing and planning strategies to perform the subtasks assigned to it. As shown in Fig. 4.1, the hose and hose ends are initially placed in the specified sections on the table. The hoses are in near the human. The hose ends are out of reach of the human but within reach of the robot. The robot is placed in front of the human on the other side of the table. A pipe with reference length is also placed within the reach of the human on the table. We divide assembly subtasks in such a way that the robot brings (picks and places) a hose end to the reach of the human. The human picks a hose and compares its length with the reference pipe. If the length of the hose is equal to the reference, the human fits the end hose to the hose and dispatches the assembled product to another section of the table. The procedure may be repeated to produce many of the hose assembly products.

4.2.2 The Handover

If the length of a hose is longer than the reference, the robot is required to hand over a cutter to the human. The cutter is initially placed out and within the reach of human and robot, respectively. When the human needs the cutter it pushes a button on the other hand of the

robot. Then, the robot finishes the current manipulation task and goes to the cutter and picks it up, then moves toward the human reach for handing over the cutter to the human based on the handover strategy. The human wears a wristband that has two mounted active markers (Fig. 4.1). The active markers can measure human hands linear and rotational speeds when the human picks the hoses and fits the parts (end hoses) respectively. We use the experimental system shown in Fig. 4.1 to investigate and justify a few novel concepts on pilot-basis including robot trust modeling and measurement, and, trust-triggered handover motion planning using kinematic redundancy, etc. Nonetheless, the setups may be scaled to realistic industrial settings in several ways:

- Realistic positioning of assembly components in the workspace layout may be ensured considering specific requirements of the task, operators health and safety, workspace ergonomics, and production flow.

- Some laboratory instruments may be replaced by more practical industrial facilities such as the keyboard command for payload handover may be replaced by voice or gesture-based commands, active markers worn in the wrist of the human may be replaced by laser position sensors, etc.

- The assembly cell may also take the advantages of the facilities of the existing factory automation system such as the assembly parts may deliver to the human and the robot through belt conveyors just-in-time (JIT), finished products may be dispatched through another belt conveyor, etc.

- The investigations can be made with assembly tasks performed in actual industrial settings

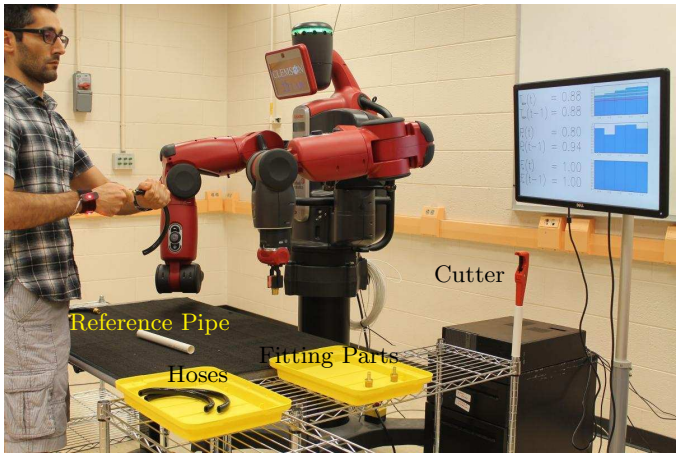


Figure 4.1: Handover task.

where multiple tools are used, and so forth.

4.3 The Trust Model and Real-Time Measurement of Trust

4.3.1 The Computational Robot-to-Human Trust Model

Most state-of-the-art trust models focus on human trust in robots, machines, automation, systems, or in humans, e.g. [30, 51], instead of on robot trust in humans except a few preliminary initiatives, e.g. [117, 86, 2]. Human trust in robot is discussed in Chapter 3, Section 3.4.4. It may be influenced by factors of robot, task, working environment and human [30]. Lee and Moray [50] proposed a time-series trust model as a function of prior trust, robot performance, and faults, and used a regression model to identify the factors of human trust in robot (automation). The time-series model is suitable for real-time quantitative trust computation [50]. We assume that manufacturing environment is well-structured, and thus uncertainty is little, and therefore such deterministic model may be suitable to capture trust, which motivates us to use the time-series model as a computational model of robot trust in human. A general computational model of human trust in robot/automation may be expressed in (4.1), where k is time step, T is human trust in robot/automation, a, b_0, b_1, c_0, c_1 are real-valued constants relevant to specific human-robot system, and q is a random noise perturbation (if any). It may be an ordinary deterministic regression model and an error-based learning algorithm, but we treat it as the computed trust [50].

$$\begin{aligned} \text{Trust}(k) = a\ \text{Trust}(k-1) \quad &+ \quad b_0\ \text{Robot Performance}(k) + b_1\ \text{Robot Performance}(k-1) \\ &+ \quad c_0\ \text{Robot Fault}(k) + c_1\ \text{Robot Fault}(k-1) + q(k), \end{aligned} \tag{4.1}$$

Trust is a perceptual issue and the human has actual perception of trust in the robot, but it is not possible to give the robot the similar perception of its trust in the human. However, we can derive a model to compute robots artificially perceived trust in human that can express the robots mental states to the human, increase transparency, reduce uncertainty in the HRC, and thus increase the effectiveness of the HRC [86]. Based on (4.1), we propose a time-series model to compute robot trust in human (denoted as T_{R2H}) as in (4.2), where P_H and F_H are reward scores for human performance and fault status respectively, and a, b_0, b_1, c_0, c_1 are real-valued constants. T_{R2H} may

be updated based on measures of P_H and F_H at every time step k.

$$T_{R2H}(k) = aT_{R2H}(k-1) + b_0P_H(K) + b_1P_H(k-1) + c_0F_H(k) + c_1F_H(k-1) + q(k), \qquad (4.2)$$

We normalize T_{R2H} value between 0 (no trust) and 1 (maximum trust) by letting $a+b_0+b_1+c_0+c_1 = 1$ and define $P_H \in [0,1],\ F_H[0,1]$. We assume that trust is calibrated and thus do not consider under or over trust of the robot as the trust measures between 0 and 1 may be proved sufficient to trigger the proposed adjustments in handover configuration and motion.

4.3.2 Real-Time Trust Measurement and Display

As observed in (4.2), we need to have real-time measurements of P_H and F_H to obtain the real-time estimates of T_{R2H} as discussed in the following sections.

4.3.2.1 Modeling and Measurement of Human Performance

Human performance, P_H, is modelled in (4.3), where V_{Hmn} is the normalized value of human hand speed for part manipulation, V_{Hm}, and V_{Han} is the normalized value of human hand speed for part gripping and releasing for manual manipulation and part attachment during assembly ,V_{Ha}, and V_{Hrn} is the normalized and absolute value of human hand angular velocity for fitting an end hose in a hose, V_{Hr}, and W_1, W_2, and W_3 are weights between 0 and 1 with $W_1 + W_2 + W_3 = 1$. The linear and rotational speeds are measured by two active markers mounted on a wristband worn by the human hand.

$$P_H(k) = W_1V_{Hmn}(k) + W_2V_{Han}(k) + W_3V_{Hrn}(k) \qquad (4.3)$$

We select human hand speed during assembly as a human performance measure because achieving high assembly efficiency largely depends on human speed (robots speed for part manipulation is kept fixed, but it can vary for handover). One of the active markers worn by the human during the assembly is used to measure hand speed (V_H) as illustrated in Fig. 4.2. V_{Hm} is identified when $V_H > V_{Hth}$, where V_{Hth} is a threshold of V_H, otherwise V_{Ha} is identified. V_{Hm} and V_{Ha} are normalized between 0 and 1 to obtain V_{Hmn} and V_{Han}, respectively, which give the measure of $P_H(k)$ according to(4.3). W_1 and V_{Hth} are determined based on the experience. P_H varies between

0 (least performance) and 1 (best performance). Effects of human fatigue and idle time are reflected in the model through hand speed measurements (performance is low if fatigue and idle time are high, and vice versa).

4.3.2.2 Modeling and Measurement of Human Fault

For the hose assembly task, a fault is considered if (i) the fitting parts are not fitted and tightened properly, and (ii) the length of the hose is not adjusted correctly, etc. If the part is correct, $F_H = 1$ is considered. The experimenter observes the assembly and assesses $F_H(k)$ subjectively using a Likert scale between 0 and 1 with a 0.1 gap between two adjacent values, and inputs the score immediately to the computer system to update trust estimation.

4.3.2.3 Real-Time Trust Measurement, Display and Update

Once P_H and F_H are measured in real-time, T_{R2H} can be measured in real-time following (4.2). However, constants of (4.2) need to be determined. The computed T_{R2H} is displayed graphically on the computer screen (Fig 4.3) placed in front of the human as in Fig. 4.1, with updates at every time step k. The trust values of five recent time steps are displayed in the graph so that the human can easily know and understand the trend of robot trust in the human. The green, yellow and red lines of trust display (Fig 4.3) indicate different warning levels to the human based on the values of robot trust in the human.

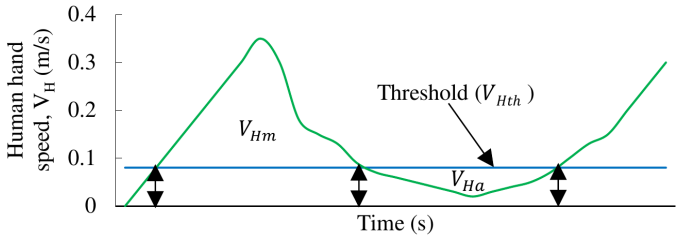


Figure 4.2: Illustration of human performance measurement during assembly.

4.4 Trust-Triggered Handover Strategy

4.4.1 The Underlying Handover Motion Planning Strategy

Given a real-time (or near real-time) measure of T_{R2H}, a key question is how to usefully exploit the trust information in modifying the robot behavior (robots handover configuration and motion). We adopt the qualitative hypothesis given in [87] for this purpose which states that if T_{R2H} reduces due to human fault and/or low speed during the collaborative assembly, the human may be unprepared for the handover and consequently may develop error in his/her own motion planning (premature and unplanned hand motion) for receiving payloads from the robot and such human error may be proportional to the amount of reduction in T_{R2H}, and vice versa.

The hypothesis means that when T_{R2H} is high, the human is mentally calm and well-prepared for the potential high impulse force between the human hand and robot end-effector through payload, and can avoid collision and thus the robot may follow the default preplanned task-optimal trajectory, which we call "normal or default handover motion". However, if the trust levels drop to below the pre-specified thresholds, the human may be uncertain about how to deal with his/her partner who has reduced trust on him/her, be unprepared for the handover, generate unplanned motion for receiving the payload from the robot, and all these human limitations may result in collision between human hand and robot end-effector that may generate impulse forces and reduce safety. To address this, a trust-triggered handover strategy is adopted for the robot so that the robot can be programmed to produce different handover configurations and motions based on T_{R2H} levels, which may reduce the effects of potential impulse forces.

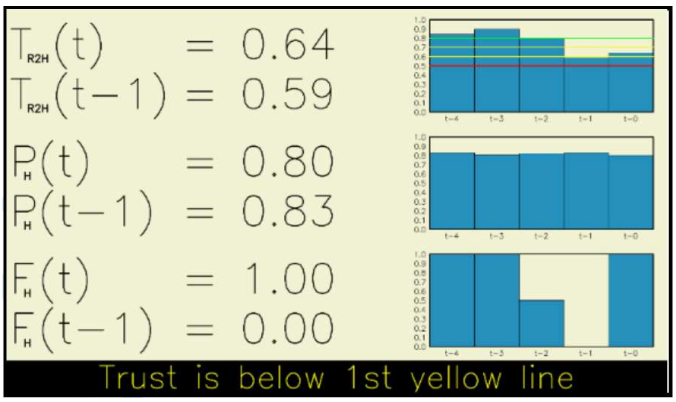


Figure 4.3: Human-computer interface for real time trust display and trust-based warnings.

4.4.2 Underlying Robot Kinematics Models

We select robot joint rates (hence robot posture as a function of time) via the manipulator Jacobian with the innovation of using robot trust to resolve kinematic redundancy [103, 118, 45]. The relation between velocities in the task-space and joint-space [106] is calculated using $\boldsymbol{J} \in \mathbb{R}^{m \times n}$ given by (4.4), where $\boldsymbol{x}(t) \in \mathbb{R}^m$ is the end-effector pose in the task-space and $\boldsymbol{q}(t) \in \mathbb{R}^n$ is the corresponding joint space configuration. $\boldsymbol{x}(t)$ and $\boldsymbol{q}(t)$ may be expressed by (4.5), where x, y, z are the end-effector position and ϕ_x, ϕ_y, ϕ_z are minimal representation of the end-effector orientation and $q_1,\ q_2,\ \ldots,\ q_n$ are the manipulator joint angles. We assume that the manipulator has $n(=$ 7 for Baxter) independently controlled axes. However, the algorithm below may be easily modified for industrial robots with fewer (4 or 5) axes. A full specification of manipulator position and orientation defined by $\boldsymbol{x}$ as above is 6-dimensional. However, in the following motion planning algorithm, we impose constraints on the specification of $\boldsymbol{x}$ by removing variables from it so that in general $\boldsymbol{x}$ is m-dimensional, where generally $m < 6$ and also $m < n$. Note that it is not necessary for $n > 6$ to have kinematic redundancy. This may occur whenever $m < n$ (the case in our analysis and experiments), providing a choice of robot configurations in the given task, which we exploit herein [103, 118, 45].

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{J}(\boldsymbol{q}(t))\dot{\boldsymbol{q}}(t), \tag{4.4}$$

$$\boldsymbol{x} = \begin{bmatrix} x \\ y \\ z \\ \phi_x \\ \phi_x \\ \phi_z \end{bmatrix}, \quad \boldsymbol{q} = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix} \tag{4.5}$$

We exploit the relationship between robot posture and the magnitude and direction of impulsive forces due to the collision at the end-effector during handover taking advantages of an existing body of understanding in robotics literature [103, 118, 45]. This understanding is based on the synthesis in end-effector space of an m-dimensional ellipsoid, which is a function of robot configuration via

the Jacobian. The directions and relative magnitudes of the ellipsoid axes illustrate the relative vulnerability of the robot to end-effector impact in these directions. In more detail, the impact ellipsoids for rigid-link robot manipulators are introduced in [118]. These dynamic impact ellipsoids are obtained by considering an impulse force acting at the tip of the robot for an infinitesimally small period of time (the time period modeling the impact of interest) in the manipulator dynamics model. Since the joint velocities and positions remain finite in such small time periods, the Jacobian and the joint torques vanish and an expression relating impulse force and change in joint velocity is obtained as in (4.6) (see [118] for the detailed derivation). In (4.6), $\boldsymbol{M}(q) \in \mathbb{R}^{n \times n}$ is the inertia matrix of the manipulator, $\boldsymbol{F} \in \mathbb{R}^m$ is the contact impulse force and $\Delta\dot{\boldsymbol{q}} \in \mathbb{R}^n$ is the vector of instantaneous changes in joint velocities caused by the potential impact.

$$\Delta\dot{\boldsymbol{q}} = \boldsymbol{M}^{-1}(\boldsymbol{q})\boldsymbol{J}^{\boldsymbol{T}}(\boldsymbol{q})\boldsymbol{F}, \tag{4.6}$$

Based on (4.6), the contact impulse force acting at the tip of the manipulator can be expressed as (4.7), where $\boldsymbol{J}^+ \in \mathbb{R}^{n \times m}$ is pseudoinverse of $\boldsymbol{J}$.

$$\boldsymbol{F} = \boldsymbol{J}^{+^T}(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q})\Delta\dot{\boldsymbol{q}} \tag{4.7}$$

Equation (4.7) specifically the Singular Value Decomposition (SVD) $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ of matrix $\boldsymbol{J}^{+^T}(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q})$ is the basis for the dynamic impact ellipsoid, where the columns of $\boldsymbol{U}$ give the directions of the principal axes of the ellipsoid. Relative magnitudes of the principal axes of the dynamic manipulability ellipsoid given by singular values of $\boldsymbol{J}^{+^T}(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q})$ in $\boldsymbol{\Sigma}$ depict relative amount of impulse forces that the tip of the manipulator may experience in the corresponding directions (column vectors in $\boldsymbol{U}$) for changes in joint velocities ($\Delta\boldsymbol{q}$). Thus, the ellipsoid is defined using (4.8) in the task space, where vector $\boldsymbol{u}_i$ defines an m-dimensional ellipse with $m = 3$ [118] as

$$\{\boldsymbol{u}_i \in \mathbb{R}^m : \boldsymbol{u}_i^T \boldsymbol{J}\boldsymbol{M}^{-2}\boldsymbol{J}^T\boldsymbol{u}_i \leq 1\}. \tag{4.8}$$

Studies of impact ellipsoids have revealed a strong correlation between the long axis of the ellipsoids (direction of largest impulse forces on impact) and the orientation of the last robot link (typically wrist/hand) [118]. This implies that to mitigate against high collision forces, the robot wrist should be bent as close to orthogonal to the approach vector (the direction collision forces are most likely

to arise in) as possible. This may be analogous to the way humans when tracing their way in dark, tend to bend their wrists to put hands up in a cautious posture with an intention to prevent them from jarring their wrists in the event on unexpected impact with something [117].

4.4.3 Trust-Based Handover Motion Planning Approach

To synthesize the cautious robot behavior in the event of low T_{R2H}, we adopt a two-step process to produce handover motions. First, we suitably modify the end-effector trajectory from the original task-optimal one. Here, the key innovation is to generate a reduced dimensional end-effector trajectory in a subset of the original task space to allow subsequent tailoring of the robot handover configuration corresponding to new end-effector motion. We retain the original geometric position of the end-effector but leave the orientation unspecified. Hence, the task-space trajectory is reduced from six to three dimensions keeping the position (but, not the pose) of the end-effector unchanged along the path. This allows generating braced configurations. A strategy is tailored to use in a new, modified (reduced dimensional) trajectory in the task space is defined as $\boldsymbol{x}_M$, which modifies and relaxes the constraints on the robot end-effector path allowing its geometric path to deviate from a direct or the most task-efficient course. The net effect may produce a cautious movement for the robot wrist/hand. We then utilize the fact that the reduced dimensionality of the modified task space causes the robot to be kinematically redundant [103, 118, 45]. This allows the robot some freedom in configurations when following the end-effector trajectory via the newly created kinematic redundancy. One approach to directly exploit the redundancy may be to solve (at velocity level) the inverse kinematics via iterative pseudoinverse-based algorithm given by

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{+}(\boldsymbol{q})\dot{\boldsymbol{x}}_M + \alpha[\boldsymbol{I} - \boldsymbol{J}^{+}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})](\nabla \boldsymbol{F})^T, \tag{4.9}$$

where $(\nabla \boldsymbol{F})^T$ is the gradient of the magnitude of $\boldsymbol{F}$. It was proved in [118] that this motion planning algorithm follows the modified end-effector trajectory via the first term in (4.9) and exploits kinematic redundancy using the second term in (4.9) to instantaneously minimize the impulse forces. The parameter α in (4.9) may be either taken from a fixed set of values (e.g., discrete values of T_{R2H}) or continuously varied values (e.g., continuously varying values of T_{R2H}). In (4.9), $\alpha = 0$ indicates no configuration compensation for T_{R2H} (i.e., T_{R2H} is high), and only the first term is used to generate handover motions solely concerned with following the modified end-effector trajectory.

However, increasingly higher values of α correspond to increasingly lower T_{R2H} that may produce greater weight on the second term and may result in changes in handover configurations to braced configurations against potential impact forces. Here, the trust-based robot motion trajectory for handover is planned using (4.9) where α is based on the trust value as:

$$\alpha = \begin{cases} \frac{T_{max}-T_{R2H}}{T_{R2H}-T_{min}} : & T_{R2H} > \frac{T_{max}+T_{min}}{2} \\ 1 \quad\quad\quad\quad\quad : & \text{otherwise} \end{cases}, \tag{4.10}$$

where $T_{max} = 1$ and $T_{min} = 0.5$. If trust is high, then $T_{R2H} = T_{max}$ and $\alpha = 0$. If trust is low, then $T_{R2H} <= \frac{T_{max}+T_{min}}{2}$ and thus $\alpha = 1$.

Note that (4.9) can be solved without finding the exact value of $\nabla \boldsymbol{F}$ using the kinematic redundancy equation given in [126] as:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{+}(\boldsymbol{q})\dot{\boldsymbol{x}}_M - k_1[\boldsymbol{I} - \boldsymbol{J}^{+}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})]\boldsymbol{H}(\boldsymbol{q} - \boldsymbol{q}_r), \tag{4.11}$$

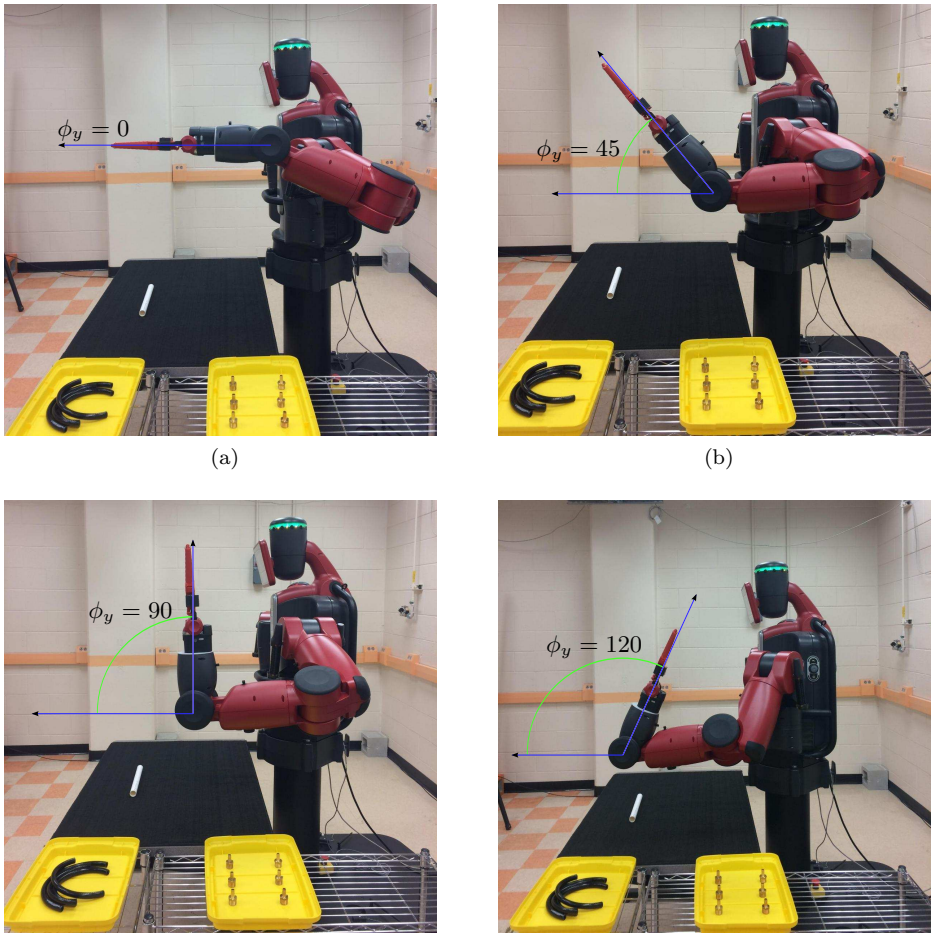
where $\boldsymbol{H}$ is a diagonal matrix with positive items, k_1 is a positive constant, and $\boldsymbol{q}_r$ is the reference configuration of the manipulator. Comparing (4.9) and (4.11) we realize that the term $(\nabla \boldsymbol{F})^T$ can be replaced by $-k_1\boldsymbol{H}(\boldsymbol{q} - \boldsymbol{q}_r)$. Here, we consider $\boldsymbol{H} = \boldsymbol{I}$ (the identity matrix), $k_1 = 1$. The reference configuration ($\boldsymbol{q}_r$) is the final configuration of the manipulator where the collision force ($\nabla \boldsymbol{F}$) is minimum, i.e. the cautious pose where trust is minimum. Thus the final trust-base handover motion can be formulated as:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{+}(\boldsymbol{q})\dot{\boldsymbol{x}}_M + \alpha[\boldsymbol{I} - \boldsymbol{J}^{+}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})](\boldsymbol{q}_r - \boldsymbol{q}). \tag{4.12}$$

4.4.4 Explanation of the Handover Approach through Examples

Fig 4.4 shows different hypothetical handover configurations of the robot based on its trust in human for the assembly. As it illustrates, more reduction in robot trust in human causes more braced (curved) configuration and more cautious (slower) handover motion. The potential impact force F_y in the contact between human hand and robot end-effector during handover is modeled as [118]

$$\boldsymbol{F}_y = (\mu \boldsymbol{n}(\boldsymbol{q})\boldsymbol{n}_y)\boldsymbol{F}, \tag{4.13}$$

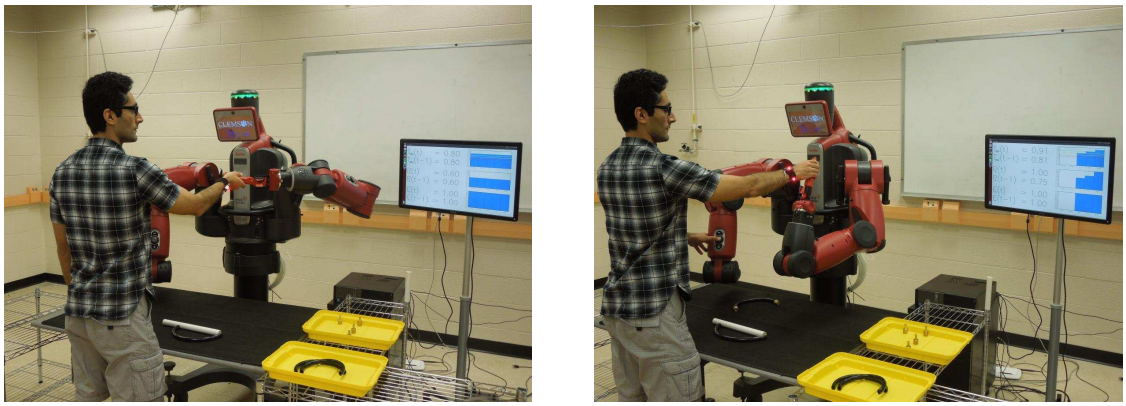


(a) (b) (c) (d)

Figure 4.4: Different configuration of the manipulator for trust-based handover.

where $\boldsymbol{n}(\boldsymbol{q})$ is a unit vector aligned with the final link (a function of robot configuration reflecting the geometry of the potential impact), $\boldsymbol{n}_y$ is a constant unit vector in the y-axis direction and μ is a constant reflecting the material quantities of the impacting bodies. The unit vector is the aspect the robot can influence with the impulsive force taking its maximum and minimum values at μ (last link aligned with y-axis) and 0 (last ink vertically aligned with respect to y-axis) respectively.

Minimizing $\boldsymbol{F}_y$ via its gradient will minimize impact forces. According to (4.13), the potential impact forces may be maximum for the handover configuration in Fig. 4.4(a) as the final link is aligned with the y-axis with a very small ϕ_y. In this case, as we hypothesize, the human may move his/her arm towards the robot in the y-axis direction with a preplanned hand trajectory to receive the payload and there is almost no possibility of any impact on human hand by the end-effector



(a) High trust (effiecent) configuration

(b) Low trust (braced) configuration

Figure 4.5: The efficient and braced configuration of the manipulator for trust-based handover.

and thus the impulse forces can be avoided. However, if T_{R2H} is increasingly low, the end-effector position is kept unchanged so that the robot can reach the payload to the human properly, but the orientation of the end-effector is left free, i.e. the modified end-effector trajectory, $\boldsymbol{x}_M$, may be simply the position trajectory $\boldsymbol{x}_M = [x,\ y,\ z]^T$. In this case, the robot may deviate its final link from y-axis so that impact between human and robot along y-axis either does not take place or reduces. Based on the varying amount of reduction in T_{R2H}, the value of ϕ_y may increase as shown in Figs. 4.4(b)-(d). In Figs.4.4(b)-(c), the handover takes place as the final link is still aligned towards the human (y-axis) with an angle ϕ_y, but the handover does not take place for Fig. 4.4(d) as the final end-effector direction totally deviates from the direction of the human (y-axis), i.e. $\phi_y \geq 90$ due to the least amount of T_{R2H}. This configuration may produce minimal impact forces. We see in (4.9) that $\boldsymbol{q}$ (joint space configuration for last link) and T_{R2H} are inversely proportional through the relationship with α, and $\boldsymbol{q}$ is related to task space configuration (ϕ_y with respect to y-axis) via manipulator Jacobian. Hence, an inverse relationship between measured ϕ_y and computed T_{R2H} in real assembly task with cautious handover motion may justify the proposed strategy. The braced configuration with minimal impact forces can be defined when $\phi_y \geq 90$. Here we pick the braced configuration as Fig. 4.4(d) as the reference braced configuration, i.e. $\boldsymbol{q}_r$ in (4.12). Fig. 4.5 shows the robot's handover configurations for the extremely high and low values of trust during an experiment.

4.5 Experimental Evaluation

4.5.1 Objectives

The objectives are to evaluate the effects of consideration of robot trust in human in the collaborative assembly and handover configuration and motion planning on overall HRI, handover and assembly performance, and task safety.

4.5.2 Experiment Design and Evaluation Scheme

We evaluated the effects of implementation of the proposed handover strategy. The independent variable is applying robot trust (T_{R2H}). A within-subject test with Latin Square design test order was performed under two different test conditions: (1) trust triggered handover (**TTH**), and, (2) trust un-triggered handover (**TUH**). The dependent variables are: (i) robot handover configuration and motion, (ii) human hand trajectory for receiving payloads during handover, (iii) HRI, (iv) handover success rate, (v) handover and assembly efficiency, (vi) impact force, and (vii) safety. Robot handover motion (position, orientation, and velocity of robot end-effector/last link) and impact force are measured by the position and force sensors embedded in the robot arm respectively. Human hand trajectory for receiving the payload is captured by the two active markers worn by the human (Fig. 4.1). The pHRI (physical HRI) for the assembly task (including handover) are expressed in a few terms as given in Table 4.1. The subject subjectively assesses the pHRI against the criteria in Table 4.1 (except the team fluency) using a Likert scale [86] (score 1 for extremely low and score 5 for very high pHRI). The experimenter records the time data using stopwatches for calculating team fluency. The cHRI (cognitive HRI) is expressed in terms of (i) human trust in the robot, and (ii) humans cognitive workload. Human trust in the robot is assessed using the Likert scale. The workload is assessed following standard NASA TLX procedures [31].

The handover success rate, ϵ_{hsr}, is expressed as in (4.14), where h_f is the total number of failed handover trials and h_t is the total number of handover trials. The safety in the handover, ϵ_s, is expressed as in (4.15), where h_c is the total number of handover trials when the human co-worker experiences collisions or large impact forces. Safety is also reflected through the magnitude of impact forces due to the collision. The handover efficiency, λ_h, is expressed in (4.16), where T_{tth} is the targeted time for a handover trial and T_{rth} is the recorded time for a handover trial. The assembly efficiency, λ_a, is expressed in (4.17), where T_{tta} is the targeted time and T_{rta} is the recorded

pHRI criterion	Description
Transparency	Physical display of contextual information (warning, messages, status) regarding human's physical performance and fault, and the resulting robot trust in the human
Naturalness	Normalcy and intuitiveness perceived by the human while physically collaborating with the robot for the assembly
Engagement	Amount/extent of humans physical involvement with the robot during the assembly
Cooperation	Extent of the sense of working together, partnership, and teamwork perceived by the human while collaborating with the robot for the assembly
Team fluency	Coordinated meshing of joint efforts and synchronization between human and robot during the collaborative assembly and handover. Four criteria (human and robot idle time, non-concurrent activity time and functional delay time) are used to objectively measure the team fluency

Table 4.1: Description of the pHRI criteria

time for an assembly (including handover) trial.

$$\epsilon_{hsr} = (1 - \frac{h_f}{h_t}) \times 100 \tag{4.14}$$

$$\epsilon_s = (1 - \frac{h_c}{h_t}) \times 100 \tag{4.15}$$

$$\lambda_h = \frac{T_{tth}}{T_{rth}} \times 100 \tag{4.16}$$

$$\lambda_a = \frac{T_{tta}}{T_{rta}} \times 100 \tag{4.17}$$

4.5.3 Subjects

Ten students (7 males and 3 females) with average age of 29.3 (and standard deviation of $SD = 3.0$) were recruited to participate in the experiments. They were performed all two conditions of the experiment. The subjects gave informed consent. The study was approved by the Institutional Review Board (IRB).

4.5.4 Experimental Procedures

In the first round practice trials, each subject was instructed about the experiment procedures and practiced the subtasks assigned to the human (Section 4.2.1). The robot performed the subtasks assigned to it. The practice trials were intended to remove learning effects of the subjects. Completion times of the entire assembly by each subject with and without robot handover were

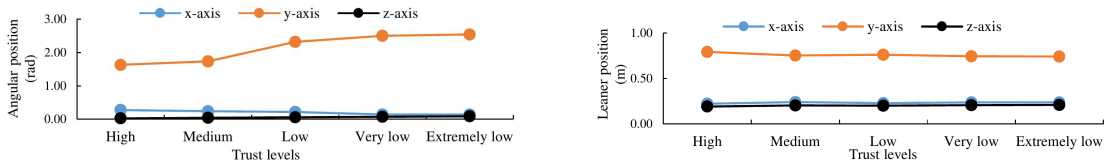
Parameter	Value
b_0	0.472
b_1	0.066
c_0	0.419
c_1	0.043
k	10s

Table 4.2: Parameters for estimating the trust

recorded and their mean values were determined, which were used as the targeted times for the assembly task including and excluding handover respectively assuming that the robot would perform the assigned subtasks in parallel with the human. Difference between target times including and excluding the handover was the target time for the handover only. Robot manipulation speed was adjusted to keep idle time zero at the beginning (ideal case). Based on the practice trials, we determined the reward scores for human performance and fault status to use for P_H and F_H for computing T_{R2H} initially following (4.2). The information on agent performance and faults in practice trials was also used to compute the constants of the trust model in (4.2) using the Autoregressive Moving Average Model (ARMAV), and to decide the time step k, as given in Table 4.2. Again, unlike the human trust, we here did not consider the prior robot trust (i.e., $a = 0$) in (4.2) as we assumed no memory of the robot. In **TTH** condition, the robot motion trajectory for handover is planned using (4.12). For both of the experiment conditions, the human subject and the robot performed the collaborative assembly. Each subject was asked separately to continue the assembly with the robot for 10 minutes (the finished assembled products were quickly disassembled and input again to keep the assembly continuing). T_{R2H} was measured and updated with a time step k and regularly displayed in the screen (Fig. 4.3) for the whole assembly period. For assembly in **TTH** condition, the robot manipulator moves based on (4.9) while the values of α updates based on the trust value, T_{R2H}. In **TUH** condition, the robot manipulator moves with regular inverse kinematic without considering the impact force, i.e. $\alpha = 0$. After each trial in each experiment condition, the collaborative assembly with handover was evaluated following the evaluation scheme, and the data were recorded separately.

4.6 Results and Evaluation

Fig. 4.6 shows the angular and linear positions of the robot end-effector for a typical trial under **TTH** condition. The results show that the linear positions along the x, y, and z axis directions were almost unchanged with trust levels, but the angular position (absolute values) along the y-axis (direction of the handover) significantly changed to produced the braced configuration. The results show that along the y-axis, there exists an inverse relationship between ϕ_y and the trust, where ϕ_y is related to $\boldsymbol{q}_y$ (joint space configuration for the last link for the y-axis) via manipulator Jacobian. The trends in ϕ_y indicate the trends in the potential impact forces between robot end-effector and human hand during handover via (4.9), which justifies the proposed handover strategy because the potential impact forces may reduce for decreasing trust values by increasing ϕ_y or $\boldsymbol{q}_y$, i.e. through more deviation of end-effector from handover direction or more braced configurations. Fig. 4.7,

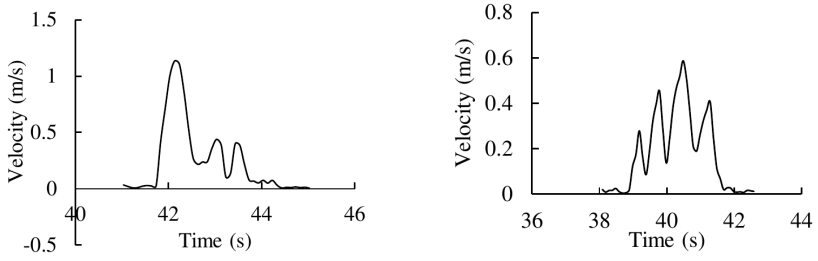


(a) Angular positions

(b) Linear positions

Figure 4.6: Angular and linear positions with respect to (along) different axes for the robot's end-effector for different trust levels for assembly under **TTH** condition.

compares the absolute measured velocity of the robots last link during handovers for high trust and low trust conditions. The results show that the robots last link velocity reduced as the robots trust in human decreased. The reduction in peak velocity provides compliance to the human in lower trust conditions as high handover velocity may generate high impact forces. Fig. 4.8 compares the



(a) High trust

(b) Low trust

Figure 4.7: Typical absolute velocity profiles of the manipulator's end-effector during handovers for high and low trust values.

Evaluation Criterion	Result for **TTH**	Result for **TUH**
Transparency	3.82(0.65)	3.89(0.53)
Naturalness	3.80(0.62)	3.47(0.41)
Engagement	3.61(0.68)	3.48(0.45)
Cooperation	3.41(0.45)	3.34(0.61)

Table 4.3: Results of physical HRI between assembly with **TTH** and **TUH**

mean impact forces between assembly with **TTH** and **TUH**. We believe that the effectiveness of the handover strategy produced lower impact forces for **TTH**. The results thus justify **Hypothesis I**. The unplanned hand motion may cause violent contact between human hand and robot end-effector that may create impulse forces. Hence, for reduced trusts, the robot saves the human by reducing impulse forces through generating braced configurations and cautious motions.

We also found that pHRI and cHRI for **TTH** were satisfactory, and better than that for **TTH**. Tabel 4.3 compares the pHRI perceived by subjects between assembly with **TTH** and **TUH**. Results show that on average the naturalness and engagement for assembly with **TTH** were slightly better than that for assembly with **TUH**. However, the transparency and cooperation were similar for both **TTH** and **TUH**. The real-time T_{R2H} display, trust-based warnings, and cautious configurations and motion of the robot made the contextual information transparent to the human and helped the human feel natural for both **TTH** and **TUH** conditions. Tabel 4.4 compares the cHRI perceived by subjects between assembly with **TTH** and **TUH**. Results show that on average the human trust in robot and cognitive workload for assembly with **TTH** were slightly better than that for assembly with **TUH**. Table 4.5 shows that handover safety, handover success rate and overall assembly efficiency for the **TTH** are better than that for the **TUH**. However, the handover efficiency reduced slightly for **TUH**.

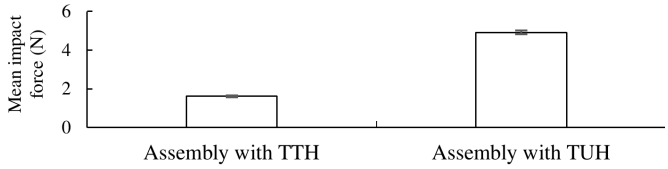


Figure 4.8: The mean impact(collision) forces between assembly with **TTH** and **TUH** conditions.

Evaluation Criterion	Result for **TTH**	Result for **TUH**
Human trust in robot	3.83(0.64)	3.56(0.59)
Cognitive workload(%)	32.1(13.64)	36.46(14.83)

Table 4.4: Results of cognitive HRI between assembly with **TTH** and **TUH**

Evaluation Criterion	Result for **TTH**	Result for **TUH**
Handover safety (%)	100	80
Handover success rate (%)	100	100
Handover efficiency (%)	96.44(0.05)	97.16(0.06)
Assembly safety (%)	96.51(0.03)	95.56(0.04)

Table 4.5: Results of objective evaluation between assembly with **TTH** and **TUH**

4.7 Conclusion

Robot trust-based human-robot collaborative assembly was proposed. A novel robot to human handover motion planning strategy was proposed so that the robot can adjust its handover configuration and motion through kinematic redundancy based on the current status of the robot-to-human trust. Such adjustments in handover configuration and motion were designed to reduce the potential impact forces between the robot end-effector and the human hand during handover and to ensure reliability and safety. Computational models of robot trust in human were derived and real-time trust measurement methods were developed. The proposed trust-based collaborative assembly including handover was evaluated using a comprehensive evaluation scheme for two types of representative assembly tasks with two different robotic platforms. In the first type setup, the handover configurations and motions with different trust levels were present. However, for the second type of assembly, the handover configurations and motions were determined in real-time with trust levels. The handover configurations were determined using kinematic redundancy. The evaluation results showed that perceived HRI, handover success rate, safety, efficiency and humans own trust in the robot increased for the trust-based assembly task with a small sacrifice in handover efficiency for both assembly tasks. The key novelties are modeling and measurement of robot trust in human and trust-triggered motion planning exploiting kinematic redundancy for handovers of parts between the robot and human for enhancement of HRI and safety in the flexible assembly in manufacturing.

The results including trust estimation are somewhat subjective in nature. However, the subjective results should be reliable as we used standard subjective methods such as the ARMAV, Likert scale, NASA TLX, etc. Statistical analyses showed nonsignificant variations in results among

the subjects, which show reliability and generality of the results. We will develop speed control algorithms for robot-human assembly and handovers to maximize human-robot bilateral trust and productivity in the near future.

Chapter 5

Trust-Based Human-Robot Cooperative Manipulation

5.1 Introduction

Cooperative manipulation refers to joint coordination of two or more robots handling a common object. This concept can address the typical limitations of single-arm robots in terms of dexterity and payload and open up new applications in flexible manufacturing systems and service robotics. In human-robot cooperative manipulation, a team of humans and robots coordinate together to handle a common object. Some research on human-robot cooperative manipulation considered a passive behavior for the robot. They presented control strategies considering human input as an exogenous input to the system [32, 101]. In contrast, a more helpful robot behaves in a proactive manner by predicting human intent [102] and minimizing human effort by ideally applying all the required forces [68]. However, the mismatch in estimating human intent may be counter effective and even harmful to the human. Thus a robot is also desired to be self-aware and has an ability for detecting and compensating for its own faults.

A collaborative robot requires its own decision-making capabilities that make seamless transitions between the different interaction paradigms. In this work, we propose a switching-based control strategy for human-robot collaborative manipulation that benefits from both of the proactive and reactive behaviors of the robot. Based on this strategy the robot starts the collaboration in

a proactive manner but it is also capable of switching to the reactive/complaint mode in case of a mismatch.

We leverage trust of human in robot for decision-making in cooperative manipulation. Trust is a key factor in any interaction [51]. It can be used as a basis of the robot decision-making strategy. As discussed in Chapter 2, human trust in robot depends on human-related, robot-related, and environmental related factors. As discussed in previous chapters, we modeled human trust in HRC in manufacturing based on prior trust and current and prior automation performance and fault occurrences [91, 94]. For the supervisory control of robots in surveillance scenario [125] and for motion planning with a human-in-the-loop [120], probabilistic trust models were developed considering causality relation between human trust and robot performance. In this chapter, we use a similar approach as in [120, 125] for developing a Dynamic Bayesian Network (DBN) model of human-to-robot trust in cooperative manipulation tasks. The probabilistic approach accommodates the uncertainties in predicting the human trust by treating it as a random variable at each time step.

The organization of this chapter is as follows. Section 5.2 summarizes a background review in role allocation problem in human-robot cooperative manipulation. An introduction to the dynamics and control problem of the cooperative manipulation systems is presented in Section 5.3. The human-robot cooperative manipulation system with the description of the reactive and proactive behaviors of the robot and model of trust of human in robot is presented in Section 5.4. A simulation study is presented in Section 5.5 and the chapter concludes in Section 5.6.

5.2 Related Work

Early works on human-robot cooperative manipulation was based on the reaction of the robot to the human operation mostly using impedance control [101, 41] and later extended by adding desired virtual constraints [111].

Another body of work has been focused on proactive behavior of the robot by means of modeling human behavior in cooperative manipulation tasks. Some works [16, 62] tried to tackle this problem analytically by following the minimum jerk principle for the human motion introduced by Flash and Hogan [20]. Some works characterized the human behavior using data from human-human experiments. In [85] trajectories for human-human cooperative manipulation were recorded

to generate human-like motion trajectories by the robot in human-robot cooperation. Later, hybrid control schemes were developed for switching between different admittance parameters based on the identification of predefined sets of motion and haptic patterns [107, 121]. A multi-class support vector machine (SVM) classification of the interaction patterns for based on force, velocity, and power related information feature sets and summary of the research on the proactive behavior of robot is provided in [61].

Another set of research has focused on effort sharing and role allocation in HRI. Namely, a switching between leader and follower roles was explored in by using a homotopy for transition between the robot control input in the leader and follower mode [18]. A similar idea [11] was proposed to switch between standalone (with solely the robot performing the task), leader and follower modes. These mentioned works did not consider when the switching should occur. Another work [44], considering a haptic board game scenario, deployed a dynamic role allocation model for transitioning to leader versus follower behavior for the computer by inferring human's intention based the applied forces by the human. In this scheme, the computer adjusts its role whenever the human applied force is out of a defined average range more than eighty percent of the last 500 milliseconds. Another dynamic role allocation and effort sharing in human-robot- cooperative manipulation was presented in [75]. The role allocation for a planar translational motion of the object was realized by adjusting a single policy parameter. The authors compared the results of experiments for three role allocation strategies including a constant uniform load distribution, dynamic scheme based on the human applied force and a binary agreement indicator, and dynamic scheme based on the agreement indicator with discretized values of the policy parameter. The role allocation problem addressed by adjusting the proactive behavior of the robot in [67, 66]. This work solved a risk-sensitive stochastic optimal control problem minimizing a cost function of a weighted sum of the disagreement between the agents, uncertainty in the estimation of the human desired trajectory and human effort.

Trust of human in robot was shown to be beneficial in dynamic role allocation and human-robot mixed-initiative haptic teleoperation scenarios of mobile robots [97, 99, 100] This chapter follows the research on the role allocation problem in human robot interaction by considering a probabilistic model of human trust in robot. Internal force, disagreement between the agents, estimation accuracy and robot effort are considered in the robot performance of the trust model. Compared to the previous works, the trust model is able to identify a faulty prediction of human intention in the proactive mode more effectively by constructing the probabilistic trust model.

5.3 Cooperative Manipulation System

In this section, we first introduce the general kinematics of cooperative manipulation in Section 5.3.1. Next, we explain the dynamics and load distribution of human multi-robot cooperative manipulation in Section 5.3.2.

5.3.1 Kinematics and Statics

Consider a human-robot cooperative manipulation system consisting of M arms – i.e. a human arm and $M-1$ robotic manipulators – that tightly grasp a rigid object. Let $\boldsymbol{x}_i = [\boldsymbol{p}_i \ \boldsymbol{\phi}_i]^T \in \mathbb{R}^q$ with $q \leqslant 6$ be the generalized position (pose) of the i-th arm coordinate frame, $\mathcal{T}_i$, with respect to a common base frame, $\mathcal{T}$. In the general case, $\boldsymbol{p}_i \in \mathbb{R}^3$ and $\boldsymbol{\phi}_i \in \mathbb{R}^3$ are the vectors of the position and the minimal representation of the orientation of $\mathcal{T}_i$ with respect to $\mathcal{T}$, respectively. The transpose of $(\cdot)$ is denoted as $(\cdot)^T$.

Let $\mathcal{T}_C$ be a coordinate frame attached to a fixed point C of the object (e.g. the mass center). The generalized position of the object coordinate frame, $\mathcal{T}_C$, with respect to the base frame $\mathcal{T}$ is given by $\boldsymbol{x} = [\boldsymbol{p} \ \boldsymbol{\phi}]^T \in \mathbb{R}^q$, where the vector $\boldsymbol{p}$ gives its position in the base frame and $\boldsymbol{\phi}$ is the minimal representation of the orientation. The relation of the object coordinate frame, $\mathcal{T}_C$, with respect to the i-th arm coordinate frame, $\mathcal{T}_i$, is expressed by a vector $\boldsymbol{r}_i$ that is denoted as the virtual stick [115]. Note for an object that is rigid and tightly grasped, each virtual stick is a constant vector and thus $\boldsymbol{p}_i + \boldsymbol{r}_i = \boldsymbol{p}$. Fig. 5.1 shows these definitions for the case of two arms. Let $\boldsymbol{h}_i = [\boldsymbol{f}_i \ \boldsymbol{n}_i]^T \in \mathbb{R}^q$ be the generalized forces (wrench) acting at the i-th arm, where $\boldsymbol{f}_i$ and $\boldsymbol{n}_i$ are the force and moment, respectively. The relation between the generalized forces acting at the tip of the i-th virtual stick (located at C), $\boldsymbol{h}_{S,i}$, with respect to the generalized forces acting at the tip of the i-th arm, $\boldsymbol{h}_i$, is given by

$$\boldsymbol{h}_{S,i} = \boldsymbol{G}_i \boldsymbol{h}_i, \tag{5.1}$$

where $\boldsymbol{G}_i$ is the Jacobian of the kinematics constraints of i-th arm denoted as the partial grasp matrix. It is defined as

$$\boldsymbol{G}_i \triangleq \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{O}_3 \\ \boldsymbol{S}(\boldsymbol{r}_i) & \boldsymbol{I}_3 \end{bmatrix},$$

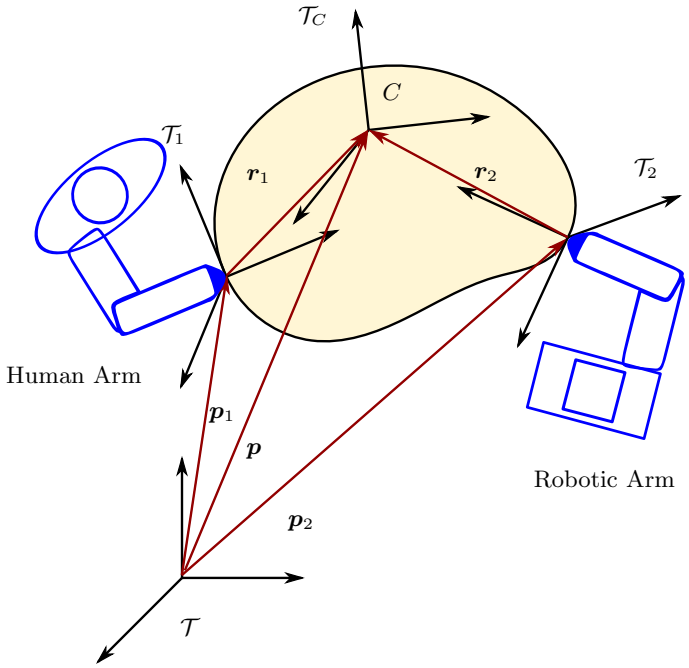


Figure 5.1: Grasp geometry for two arm (one human and one robot) cooperative manipulation.

where $\boldsymbol{I}_l$ and $\boldsymbol{O}_l$ denote the identity matrix and null matrix of $(l \times l)$ dimensions, respectively, and $\boldsymbol{S}(\boldsymbol{r}_i)$ is the skew-symmetric cross product matrix operator [76]. For an arbitrary vector $\boldsymbol{v} = [a\ b\ c]^T$ the cross product matrix operator $\boldsymbol{S}(\boldsymbol{v})$ is defined as

$$\boldsymbol{S}(\boldsymbol{v}) = \begin{bmatrix} 0 & -c & b \\ c & 0 & -a \\ -b & a & 0 \end{bmatrix}.$$

Let us denote the external force and moment applied to the object as the vector of generalized force, $\boldsymbol{h}_e$, given by

$$\boldsymbol{h}_e = \sum_{i=1}^{M} \boldsymbol{h}_{S,i} = \boldsymbol{W}_S \boldsymbol{h}_S, \tag{5.2}$$

where

$$\boldsymbol{W}_S = \begin{bmatrix} \boldsymbol{I}_q & \boldsymbol{I}_q & \dots & \boldsymbol{I}_q \end{bmatrix}, \quad \boldsymbol{h}_S = \begin{bmatrix} \boldsymbol{h}_{S,1} \\ \boldsymbol{h}_{S,2} \\ \dots \\ \boldsymbol{h}_{S,M} \end{bmatrix}_{qM\times 1}.$$

5.3.2 Dynamics and Load Distribution

The object's motion with respect to the inertia frame $\mathcal{T}$ is described by

$$\boldsymbol{M}_o(\boldsymbol{x})\ddot{\boldsymbol{x}} + \boldsymbol{C}_o(\boldsymbol{x},\ \dot{\boldsymbol{x}}) = \boldsymbol{h}_e + \boldsymbol{h}_{ext}, \tag{5.3}$$

where $\boldsymbol{M}_o \in \mathbb{R}^{q\times q}$ is the inertia matrix of the object and $\boldsymbol{C}_o \in \mathbb{R}^q$ is the sum of friction and gravitation and $\boldsymbol{h}_{ext} \in \mathbb{R}^q$ is the vector of the external generalized force applied to the object (not by the arms). Equation (5.2) calculates the object generalized force $\boldsymbol{h}_e$ given the arm generalized forces $\boldsymbol{h}_S$ acting at the tip of virtual sticks. In the cooperative manipulation, the inverse problem is usually of more interest, i.e. given the desired object generalized force, $\boldsymbol{h}_e^d$, what would be the generalized forces of the arms, $\boldsymbol{h}_S^d$? Since the number of unknown parameters, i.e. $\boldsymbol{h}_S^d \in \mathbb{R}^{qM}$, is greater than that of known parameters, i.e. $\boldsymbol{h}_e^d \in \mathbb{R}^q$, the solution of this problem is not unique. The general inverse solution of (5.2) is given by [4]

$$\boldsymbol{h}_S^d = \boldsymbol{W}_S^+\boldsymbol{h}_e^d + [\boldsymbol{I}_{qM} - \boldsymbol{W}_S^+\boldsymbol{W}_S]\boldsymbol{h}_I^*, \tag{5.4}$$

where $\boldsymbol{W}_S^+ \in \mathbb{R}^{qM\times q}$ is the generalized inverse (or pseudo-inverse) of $\boldsymbol{W}_S$. The columns of the matrix $\boldsymbol{I}_{qM} - \boldsymbol{W}_S^+\boldsymbol{W}_S$ are a basis of the null space of $\boldsymbol{W}_S$ and $\boldsymbol{h}_I^*$ represents the internal loading of the object. Therefore, the term $[\boldsymbol{I}_{(qM)} - \boldsymbol{W}_S^+\boldsymbol{W}_S]\boldsymbol{h}_I^*$ do not contribute to external forces acting on the object. The load distribution of the generalized forces among the arms depends on the choice of $\boldsymbol{W}_S^+$. In HRC scenarios, a nonuniform distribution with minimal load assigned to the human is usually desired. Let us consider the design of load sharing matrix $\boldsymbol{A}$ such that

$$\boldsymbol{h}_S^d = \boldsymbol{A}\boldsymbol{h}_e^d, \qquad \boldsymbol{h}_e^d = \boldsymbol{W}_S\boldsymbol{h}_S^d = \boldsymbol{W}_S\boldsymbol{A}\boldsymbol{h}_e^d. \tag{5.5}$$

In cooperative manipulation, internal forces that squeeze the object are often undesirable. It is shown in [68] that the family of solutions that satisfies (5.5) and without resulting in any counteracting (squeezing) generalized forces is given by

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 & \dots & \boldsymbol{A}_M \end{bmatrix}^T, \tag{5.6}$$

where

$$\boldsymbol{A}_i = diag \begin{bmatrix} \alpha_{f,i} & \alpha_{f,i} & \alpha_{f,i} & \alpha_{n,i} & \alpha_{n,i} & \alpha_{n,i} \end{bmatrix},$$

$$\alpha_{f,i},\ \alpha_{n,i} \geq 0 \quad \forall i = 1 \dots M$$

$$\sum_{i=1}^{M} \alpha_{f,i} = 1 \quad \sum_{i=1}^{M} \alpha_{n,i} = 1.$$

Note that the generalized Moore-Penrose pseudoinverse solution results in the uniform distribution of load among arms as $\{\alpha_{f,i} = \frac{1}{M},\ \alpha_{\tau,i} = \frac{1}{M}\}$.

5.4 Human-Robot Cooperative Manipulation

The objective of cooperative manipulation includes the control of both motion of the held object and the internal loading of the object [68]. The motion objective is given as a desired trajectory of the object, $\boldsymbol{x}^d(t) \in \mathbb{R}^q$, such that

$$\lim_{t \to \infty} \boldsymbol{x}(t) \to \boldsymbol{x}^d(t), \tag{5.7}$$

and the force objective is given as desired arms generalized force trajectories, $\boldsymbol{h}^d(t) \in \mathbb{R}^{(qM)}$ that are needed to avoid undesired squeeze or internal object forces such that

$$\lim_{t \to \infty} \boldsymbol{h}(t) \to \boldsymbol{h}^d(t). \tag{5.8}$$

We use impedance control approach for tracking a desired trajectory of the object motion in cooperative manipulation [101, 6]. The desired apparent object impedance renders the system as a mass-damper system given by

$$\boldsymbol{M}_v \ddot{\boldsymbol{x}} + \boldsymbol{D}_v \dot{\boldsymbol{x}} = \boldsymbol{h}_{imp} + \boldsymbol{h}_{ext}, \tag{5.9}$$

where $\boldsymbol{M}_v,\ \boldsymbol{D}_v \in \mathbb{R}^{q\times q}$ are the desired virtual inertia and damping matrices and $\boldsymbol{h}_{imp} \in \mathbb{R}^q$ is the controller input that renders the object desired impedance behavior and $\boldsymbol{h}_{ext}$ is the same generalized vector of the external force applied to the object as given in (5.3). The desired acceleration of the object is derived by (5.9) as

$$\ddot{\boldsymbol{x}} = \boldsymbol{M}_v^{-1}[\boldsymbol{h}_{imp} + \boldsymbol{h}_{ext} - \boldsymbol{D}_v\dot{\boldsymbol{x}}]. \tag{5.10}$$

From (5.3) together with (5.10), it follows that the generalized force vector applied to the object is desired to be

$$\boldsymbol{h}_e^d = \boldsymbol{C}_o - \boldsymbol{h}_{ext} + \boldsymbol{M}_o\boldsymbol{M}_v^{-1}[\boldsymbol{h}_{imp} + \boldsymbol{h}_{ext} - \boldsymbol{D}_v\dot{\boldsymbol{x}}]. \tag{5.11}$$

The other control objective, i.e. controlling the generalized forces of the arms such that they are non-squeezing (free of internal force) is achieved through the realization of the load sharing matrix, $\boldsymbol{A}$, given by Equation (5.6) in Section 5.3.2. We now consider the cooperative manipulation problem with a human-in-the-loop and indicate the superscript h for the human, i.e. the human partial grasp matrix is denoted as $\boldsymbol{G}_h$ and the human applied generalized force at the tip of his/her hand and first virtual stick are denoted as $\boldsymbol{h}_h$ and $\boldsymbol{h}_{S,h}$, respectively. With the human-in-the-loop, the control objectives of (5.7) and (5.8) are not straightforward in general due to the following challenges:

- The human desired object trajectory $\boldsymbol{x}_h^d$ is not known by the robot.
- The desired applied load sharing by the human $\boldsymbol{h}_h^d$ is unknown to the robot.

There are two general methods for addressing these challenges [68], i.e. the reactive robot behavior approach and the proactive robot behavior approach. In the reactive setting, the human is considered as an exogenous input rather than an agent of the multi-agent cooperative system. For such a system, the desired object behavior is

$$\boldsymbol{M}_v\ddot{\boldsymbol{x}} + \boldsymbol{D}_v\dot{\boldsymbol{x}} = \boldsymbol{h}_{ext} = \boldsymbol{h}_h, \tag{5.12}$$

Comparing (5.12) with (5.9), the desired generalized force for the object can be calculated using Equation (5.11), with $\boldsymbol{h}_{ext} = \boldsymbol{h}_h$ and $\boldsymbol{h}_{imp} = \boldsymbol{0}$. The human applied forces can be measured using force sensors or can be estimated using robot's joint velocities [28]. The desired generalized force acting on the manipulators at their tip of the virtual stick is calculated via (5.4)ãnd (5.5). The

actual generalized forces acting at the tip of manipulators can be calculated from (5.1) as:

$$\boldsymbol{h}_i = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{O}_3 \\ -\boldsymbol{S}(\boldsymbol{r}_i) & \boldsymbol{I}_3 \end{bmatrix} \boldsymbol{h}_{S,i}. \tag{5.13}$$

Given a robot generalized force in task space, $\boldsymbol{h}_i$, the robot's control input (joints' torques), $\boldsymbol{\tau}_i$, is calculated as

$$\boldsymbol{\tau}_i = (\boldsymbol{J}_i^{\dagger})^T \boldsymbol{h}_i \tag{5.14}$$

where $\boldsymbol{J}_i^{\dagger}$ represents the Moore-pseudo inverse of the i-th manipulator arm Jacobian. Since the human is treated as an external force, the robot is the only agent in the reactive cooperative manipulation and we do not consider the load-sharing matrix.

In the proactive setting, the robot estimates the human desired motion and force and plans accordingly. In this setting, it follows that $\boldsymbol{h}_{imp} = \boldsymbol{h}_v + \boldsymbol{h}_h$, where $\boldsymbol{h}_v$ is input from the robot. Assuming there is no external forces applied to the object, i.e. $\boldsymbol{h}_{ext} = \boldsymbol{0}$, the desired behavior of the object is given by

$$\boldsymbol{M}_v \ddot{\boldsymbol{x}} + \boldsymbol{D}_v \dot{\boldsymbol{x}} = \boldsymbol{h}_v + \boldsymbol{h}_h. \tag{5.15}$$

In ideal case the robot would apply the entire required force and the load sharing distribution would be as $\{\alpha_{f,h} = 0, \alpha_{n,h} = 0\}$ and $\{\alpha_{f,r} = 1, \alpha_{n,r} = 1\}$ where superscripts h and r denote the human and the robots, respectively. This is not possible since the human desired trajectory can only be realized when human applies some forces. However, a controller can be designed to minimize the human effort. For example, in [68] an optimal control problem was defined that finds the robot generalized forces such that the human effort is minimized. This problem needs a careful attention and is beyond the scope of this paper. Here, we find the load-sharing matrix, $\boldsymbol{A}$, at current time first by calculating the related human share matrix, $\boldsymbol{A}_h$, by using the desired applied generalized force to the object, $\boldsymbol{h}_e^d$, calculated as in (5.11) and measurement of the human applied generalized force, $\boldsymbol{h}_h$. Then the load sharing matrix is calculated using the equality constraints introduced in (5.6). After finding the load sharing matrix, the generalized forces acting at the arms can be calculated similar to the reactive mode using (5.11), (5.5) and (5.13). The robot's joints control input is calculated using (5.14).

Given an estimation of the desired trajectory for the object, $\boldsymbol{x}^d$, a common controller for $\boldsymbol{h}_v$ is a proportional derivative as

$$\boldsymbol{h}_v = \boldsymbol{K}_D \dot{\boldsymbol{e}} + \boldsymbol{K}_P \boldsymbol{e}, \tag{5.16}$$

where $\boldsymbol{e} = \boldsymbol{x}^d - \boldsymbol{x}$, and $\boldsymbol{K}_D,\ \boldsymbol{K}_P \in \mathbb{R}^{q \times q}$ are the controller damping and stiffness matrices, respectively.

5.4.1 Human Intent Estimation

Human is proficient in cooperating with others smoothly and proactively. The understanding of human action and the corresponding intent are critical in efficient human-robot co-manipulation. Different techniques are available for estimating human's action and intent. In [119], the authors point out that simple models for human motion prediction such as hidden Markov model (HMM) and linear dynamical systems (LDS) can be learned easily and effectively but limited to predict complex motions. In [57], the authors point out that Dynamic Bayesian Networks can model human motions as well due to their effectiveness in modeling temporal dynamics of motion patterns but limited in choosing appropriate model parameters. Support vector machines can also be used for both classification and regression [95, 96]. A multi-class support vector machine classification of the interaction patterns for based on force, velocity, and power related information feature sets for cooperative manipulation is provided in [61].

Gaussian process (GP) models have been proven suitable for modeling human movement [119] and become increasingly popular for modeling system dynamics [47].

A Gaussian (Normal) distribution for a random variable y is defined as $p(y) = \mathcal{N}(y; \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma}} \exp(\frac{-(y-\mu)^2}{2\sigma^2})$ with mean value μ and variance σ. A multivariate (joint) Gaussian distribution of a N-dimensional random vector $\boldsymbol{y} = [y_1, \ldots, y_N]^T$ is denoted as $p(\boldsymbol{y}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with the mean vector, $\boldsymbol{\mu}$, and the covariance matrix, $\boldsymbol{\Sigma}$, as

$$\begin{aligned} \boldsymbol{\mu} &= E\left[\boldsymbol{y}\right] = \left[E[y_1], \ldots, E[y_N]\right], \\ \boldsymbol{\Sigma} &= E\left[(\boldsymbol{y} - \boldsymbol{\mu})(\boldsymbol{y} - \boldsymbol{\mu})^T\right] = \left[Cov[y_i, y_j]; 1 \geq i, j \geq N\right], \end{aligned} \tag{5.17}$$

where $E[.]$ represents the expected value. A GP is a statistical distribution of a collection of function values $f(\boldsymbol{z}) : \mathbb{R}^n \to \mathbb{R}$ with $\boldsymbol{z} \in \mathbb{R}^n$ where any subset of finite number of samples $(\{f(\boldsymbol{z}_1) \ldots f(\boldsymbol{z}_N)\}$,

$N \in \mathbb{N}^+$) forms a multivariate Gaussian distribution [40]. A GP function $f(\boldsymbol{z})$ is denoted with its mean function $m(\boldsymbol{z})$ and covariance function $k(\boldsymbol{z}, \boldsymbol{z}')$ as

$$f(\boldsymbol{z}) \sim \mathcal{GP}(m(\boldsymbol{z}), k(\boldsymbol{z}, \boldsymbol{z}')), \tag{5.18}$$

where

$$m(\boldsymbol{z}) = E\left[f(\boldsymbol{z})\right], \tag{5.19}$$

$$k(\boldsymbol{z}, \boldsymbol{z}') = Cov(f(\boldsymbol{z}), f(\boldsymbol{z}')) = E\left[(f(\boldsymbol{z}) - m(\boldsymbol{z}))(f(\boldsymbol{z}') - m(\boldsymbol{z}'))\right]. \tag{5.20}$$

For a given noise-free training data, the prior joint distribution of observations $\boldsymbol{f} = \{f(\boldsymbol{z}_i)\}_{i=1}^N$ at input points $\boldsymbol{Z} = \{\boldsymbol{z}_i\}_{i=1}^N$ is written as

$$\boldsymbol{f} \sim \mathcal{N}(m(\boldsymbol{Z}), K(\boldsymbol{Z}, \boldsymbol{Z})), \tag{5.21}$$

where the elements of $K(\boldsymbol{Z}, \boldsymbol{Z})$ are $K_{ij} = k(\boldsymbol{z}_i, \boldsymbol{z}_j)$. The joint distribution of a set of observations $\boldsymbol{f}$ at input points $\boldsymbol{Z}$ with a zero mean normal distribution and variance of σ_n^2 for the observation noise and the predictive output $f*$ at the test input $\boldsymbol{z}_*$ is

$$\begin{bmatrix} \boldsymbol{f} \\ f_* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(\boldsymbol{Z}) \\ m(\boldsymbol{z}_*) \end{bmatrix}, \begin{bmatrix} K(\boldsymbol{Z}, \boldsymbol{Z}) + \sigma_n^2 \boldsymbol{I} & K(\boldsymbol{Z}, \boldsymbol{z}_*) \\ K(\boldsymbol{z}_*, \boldsymbol{Z}) & K(\boldsymbol{z}_*, \boldsymbol{z}_*) \end{bmatrix} \right), \tag{5.22}$$

Conditioning the distribution, the posterior (predictive) distribution, f_*, is given by [40]:

$$f_* | \boldsymbol{f}, \boldsymbol{Z}, \boldsymbol{z}_* \sim \mathcal{N}(\mu_{f_*}, Cov(f_*)), \text{ where} \tag{5.23}$$

$$\mu_{f_*} = E(f_* | m(\boldsymbol{f}), \boldsymbol{Z}, \boldsymbol{z}_*) = E(f_*) + K(\boldsymbol{z}_*, \boldsymbol{Z}) \left[K(\boldsymbol{Z}, \boldsymbol{Z}) + \sigma_n^2\right]^{-1} \boldsymbol{f} \tag{5.24}$$

$$Cov(f_*) = K(\boldsymbol{z}_*, \boldsymbol{z}_*) - K(\boldsymbol{z}_*, \boldsymbol{Z}) \left[K(\boldsymbol{Z}, \boldsymbol{Z} + \sigma_n^2 \boldsymbol{I})\right]^{-1} K(\boldsymbol{Z}, \boldsymbol{z}_*). \tag{5.25}$$

In our problem, the goal is to estimate the desired motion from the observed data. we use data form human-human cooperative manipulation to train GP models for each dimension of the motion trajectory. In training, we ask two people to jointly move an object in a task-specified environment

and record the motion and force applied to the object. We consider the input of these models as

$$\boldsymbol{\xi}(k) = \begin{bmatrix} \boldsymbol{x}(k) \\ \dot{\boldsymbol{x}}(k) \\ \boldsymbol{h}_e(k) \\ \phi \end{bmatrix}, \tag{5.26}$$

where k is the time index and ϕ is the task parameter. The outputs of the GP models are $\{\lambda_i\}_{i=1}^{2q}$, where

$$\boldsymbol{\lambda}(\boldsymbol{\xi}(k)) = \begin{bmatrix} \boldsymbol{x}(k+1) \\ \dot{\boldsymbol{x}}(k+1) \end{bmatrix}. \tag{5.27}$$

Using the GP notation we can write

$$\lambda_i(\boldsymbol{\xi}) \sim GP(\mu_i, k_i(\boldsymbol{\xi}, \boldsymbol{\xi}')), \tag{5.28}$$

where μ_i and $k_i(\boldsymbol{\xi}, \boldsymbol{\xi}')$ are the mean function and the covariance function chosen based on the prior (training) data. For a set of N observations $\boldsymbol{\Lambda}_i = \{\lambda_i(\boldsymbol{\xi}(j))\}_{j=1}^{N}$ at input points $\boldsymbol{\Xi} = \{\boldsymbol{\xi}(j)\}_{j=1}^{N}$ with a prior expected value of μ_i and a zero mean noise with variance of σ_n^2 and a test input $\boldsymbol{\xi}^*$, the posterior distribution of the desired point on the trajectory is a Gaussian distribution with the mean and variance as

$$E(\lambda_i^*) = \mu_i + K(\boldsymbol{\xi}^*, \boldsymbol{\Xi}) \left[K(\boldsymbol{\Xi}, \boldsymbol{\Xi}) + \sigma_n^2\right]^{-1} \boldsymbol{\Lambda}_i \tag{5.29}$$

$$Var(\lambda_i^*) = K(\boldsymbol{\xi}^*, \boldsymbol{\xi}^*) - K(\boldsymbol{\xi}^*, \boldsymbol{\Xi}) \left[K(\boldsymbol{\Xi}, \boldsymbol{\Xi} + \sigma_n^2 \boldsymbol{I})\right]^{-1} K(\boldsymbol{\Xi}, \boldsymbol{\xi}^*). \tag{5.30}$$

5.4.2 Trust-Based Cooperative Manipulation

Both reactive and proactive behaviors have some pros and cons. The reactive approach is intuitive and effective since the motion planning problem is done by the human partner but it can only serve as a pHRI interface and does not reduce the human effort significantly [68]. The proactive approach addresses the effort sharing problem by estimating the human desired motion but it is involved with uncertainty and disagreement challenges. In this work, we integrate the strengths of these two approaches by proposing a trust-triggered switching control policy. In this

work, for ease of formulation and demonstration, we consider a team of one human and one robot ($M = 2$) but a similar approach can be extended to a team of multiple humans and robots. We denote the human and robot with the superscript h and r, respectively instead of i as the general superscript for the agents wherever it is more appropriate. The controller estimates the human trust in robot using our proposed trust model. The trust model is based on the human performance, robot performance, and environment features. Trust is a key factor in the collaboration and can guide the reliance of human on the robot. On one hand, higher trust values reflect higher degree of collaboration and joint performance of the human-robot team in performing the cooperative tasks. This means that for higher trust values the robot is more reliable and is able to behave in a proactive manner well enough. On the other hand, lower trust values reflect the inadequate team coordination and joint performance. If the robot is not reliable enough to act in a proactive manner, the controller switches to the reactive mode. In the following sections, we first introduce the trust model and then we propose our trust-based switch-control strategy.

5.4.2.1 Human-Robot Trust Model

In this work, we use a probabilistic approach similar to [125] for developing a DBN model of human-to-robot trust in cooperative manipulation tasks. The probabilistic approach accommodates the uncertainties in predicting the human trust by treating it as a random variable at each time step. In particular, we propose the DBN model as shown in the Fig. 5.2. The human trust-to-robot, $T \in [0, 1]$, is related to the prior trust and current and previous robot performance, $P_R \in [0, 1]$ [50, 91]. The actual realization of trust value is a hidden state. Thus we maintain a belief distribution of human-to-robot trust and update it using an observation related to it. We consider a normalized magnitude of the internal force as the disagreement between human and the robot, D. The conditional probability distribution (CPD) for the relation between human-to-robot trust at time step k, $T(k)$, to previous trust, $T(k-1)$ given the current and previous robot performance, $P_R(k)$ and $P_R(k-1)$ is expressed by a Gaussian CPD with mean value μ_T and variance σ_T as:

$$p(T(k), T(k-1), P_R(k), P_R(k-1)) =$$
$$p(T(k)|T(k-1), P_R(k), P_R(k-1)) = \mathcal{N}(T(k); \mu_T(k), \sigma_T(k)), \tag{5.31}$$

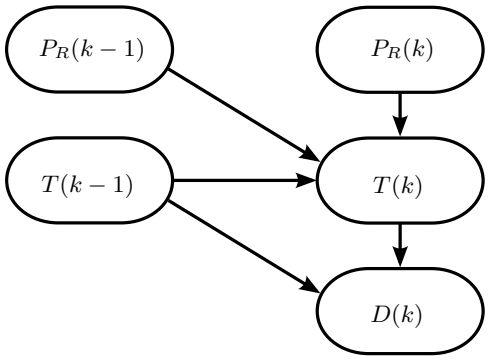


Figure 5.2: Trust model structure.

where μ_T is defined based on the causality relation between trust and performance as:

$$\mu_T(k) = aT(k-1) + bP_R(k) + cP_R(k-1). \tag{5.32}$$

The coefficients a, b and c are constants where $a+b+c=1$ and are determined for each individual in the training session.

Trust is a complex and multidimensional concept and incorporates different characterizations such as beliefs, attitudes, intentions, or behaviors [46]. In human-robot cooperative manipulation, the robot behavior can be perceived using the disagreement between the human and the robot. The internal force does not contribute to object motion and its normalized magnitude with respect to the total force magnitude can be used as a measure of disagreement. We denote $D \in [0,1]$ as the disagreement. Low disagreements correspond, e.g. $D \approx 0$, to high values of trust, e.g. $T \approx 1$, and vice versa. Moreover, increasing the disagreement yields to decline of trust, i.e. $D \propto (1-T)$. We relate the observed disagreement and the trust value at time step k with a Gaussian distribution as:

$$p(D(k), T(k)) = p(D(k)|T(k)) = \mathcal{N}(D(k); 1-T(k), \sigma_D), \tag{5.33}$$

where σ_D is a zero-mean random variable represents the disagreement uncertainty.

5.4.2.2 Trust Inference

The probabilistic belief of the human's trust, $T(k) \in [0,1]$, at the time step k can be estimated using the trust model. Both filtering and smoothing problems can be considered using

the trust model. The filtered belief at the current time step k is defined as

$$bel_f(k) = p(T(k)|P_R(1:k), D(1:k), T(0)). \tag{5.34}$$

The smoothed belief at any time step $k \in [0:K]$ given the recorded model parameter values is defined as

$$bel_s(k) = p(T(k)|P_R(1:K), D(1:K), T(0)). \tag{5.35}$$

Both of the filtered and smoothed believes can be calculated recursively as described in [125] by

$$\begin{aligned}
&\overline{bel}(T(k), T(k-1)) = p(T(k), D(k)) \\
&\cdot p(T(k), T(k-1), P_R(k), P_R(k-1)) \cdot bel_f(T(k-1)) \qquad (5.36)\\
&bel_f(T(k)) = \frac{\int \overline{bel}(T(k), T(k-1))dT(k-1)}{\iint \overline{bel}(T(k), T(k-1))dT(k-1)dT(k)} \qquad (5.37)\\
&bel_s(T(k-1)) \\
&= \int \frac{\overline{bel}(T(k), T(k-1))}{\int \overline{bel}(T(k), T(k-1))dT(k-1)} \cdot bel_s(T(k))dT(k) \qquad (5.38)
\end{aligned}$$

The initial trust belief, $T(0)$, is assumed to be uniform and maximum, i.e. $p(T(0)) = 1$. Given the training data set, the optimized model parameters $\mathbf{\Theta}^*$ (e.g. a, b, c, σ_T, $\cdots$) for each individual can be found using hard Expectation Maximization (EM) [125] as follows

$$\mathbf{\Theta}^* = \arg\max_{\mathbf{\Theta}} \max_{T(1:K)} p(T(1:K), P_R(1:K), D(1:K)|T(0)).$$

5.4.2.3 Robot Performance

We consider internal force, disagreement, accuaracy of the human motion estimation, and human effort as metrics for the robot performance, P_R. Robot performance is calculated in the window of τ time steps as

$$P_R(k) = \frac{1}{\tau} \sum_{i=0}^{\tau-1} p_r(k-i), \tag{5.39}$$

where robot instantaneous performance, $p_r(i)$ is calculated as

$$p_r(k) = \sum_{i=1}^{4} w_i f_i(k) \tag{5.40}$$

and the weights are given as $\sum_{i=1}^{4} w_i = 1$ and $w_i > 1$. The description of each of the factors contributing to robot performance are as follows:

1. Internal Force: Control design with the load sharing matrix $\boldsymbol{A}$ in Equation (5.6) results in no counteracting generalized force in the object frame. The internal force would be zero in case of multiple robots without a human-in-the-loop. However, in practice, inconsistency between the robot and human yields internal forces. Given observed generalized forces of the arms, $\boldsymbol{h}_S$, the effective and internal generalized forces are defined as:

$$\begin{aligned} & \boldsymbol{h}_S = \boldsymbol{h}_{S,eff} + \boldsymbol{h}_{S,int} \\ \text{s.t.} \quad & \boldsymbol{W}_S \boldsymbol{h}_{S,eff} = \boldsymbol{W}_S \boldsymbol{h}_S = \boldsymbol{h}_e^d \ \text{ and } \ \boldsymbol{h}_{S,int} = 0. \end{aligned} \tag{5.41}$$

The effective generalized forces only contribute in the motion of the object. From (5.6) they can be written as [68]

$$\boldsymbol{h}_{S,eff} = \boldsymbol{A}\boldsymbol{h}_e^d. \tag{5.42}$$

For a 2-agent system the effective generalized forces, $\boldsymbol{h}_{S,eff,i} = \begin{bmatrix} \alpha_{f,i}, \boldsymbol{f}^d \\ \alpha_{n,i}, \boldsymbol{n}^d \end{bmatrix}$, can be calculated by Equation (10.25) in [68]. Internal force is calculated as

$$\boldsymbol{h}_{S,int} = \boldsymbol{h}_S - \boldsymbol{h}_{S,eff} \tag{5.43}$$

The internal force factor, f_1, at time step k is calculated as

$$f_1(k) = \frac{||\boldsymbol{h}_{S,eff}(k)||}{||\boldsymbol{h}_{S,eff}(k)|| + ||\boldsymbol{h}_{S,int}(k)||}. \tag{5.44}$$

2. Agreement: The existence of internal generalized forces indicates that the generalized forces of the agents, $\boldsymbol{h}_{S,i}$ have different directions. This is considered as the disagreement between agents. We define agreement between agents, f_2, at time step k as

$$f_2(k) = \begin{cases} \frac{\boldsymbol{h}_h(k) \cdot \boldsymbol{h}_r(k)}{||\boldsymbol{h}_h(k)|| \cdot ||\boldsymbol{h}_r(k)||} & \text{if } \boldsymbol{h}_h(k) \cdot \boldsymbol{h}_r(k) > 0 \\ 0 & \text{Otherwise} \end{cases}. \tag{5.45}$$

The first case in (5.45) represents the angle between the forces. The value of f_2 is one when the human and robot apply force in the same direction and decreases as the difference between the direction of the forces of the agents increases. Here we assume if the angle between the forces is greater than the right angle the agents completely disagree with each other and hence $f_2 = 0$. The disagreement (defined in Section 5.4.2.1), is calculate as $D = 1 - f_2$.

3. Estimation Accuracy: The normalized value of the difference between the estimated object motion trajectory with the actual object motion trajectory with respect to three fold of the standard deviation, i.e.

$$\sigma_\lambda(k) = \begin{bmatrix} Var(\lambda_1(\boldsymbol{\xi}(k))) \\ \vdots \\ Var(\lambda_q(\boldsymbol{\xi}(k))) \end{bmatrix}, \tag{5.46}$$

represents the accuracy of the estimation as

$$f_3(k) = \begin{cases} 1 - \frac{||\hat{\boldsymbol{x}}(k) - \boldsymbol{x}(k)||}{3\sigma_\lambda} & \text{if } ||\hat{\boldsymbol{x}}(k) - \boldsymbol{x}(k)|| < 3\sigma_\lambda(k) \\ 0 & \text{Otherwise} \end{cases}. \tag{5.47}$$

4. Robot Effort: The normalized robot generalized force acting at the tip of virtual stick, $\boldsymbol{h}_{S,r}$ with respect to the total force presents the robot effort with respect to the effort required for moving the object and is calculated by

$$f_4(k) = \frac{||\boldsymbol{h}_{S,r}(k)||}{||\boldsymbol{h}_e(k)||}. \tag{5.48}$$

5.4.2.4 Switched-Control Strategy

In the trust-based switched control strategy, the desired behavior of the system is given by

$$\boldsymbol{M}_v \ddot{\boldsymbol{x}}(t) + \boldsymbol{D}_v \dot{\boldsymbol{x}}(t) = \boldsymbol{h}_\sigma(t), \tag{5.49}$$

$$\boldsymbol{h}_\sigma(t) = \begin{cases} \boldsymbol{h}_h(t) & \sigma(t) = 0 \\ \boldsymbol{h}_h(t) + \boldsymbol{h}_v(t) & \sigma(t) = 1 \end{cases}, \tag{5.50}$$

where $\sigma(t)$ represents the reactive or proactive behavior of the robot. The switching strategy of these two modes different and is determined as follows:

$$\sigma(t) = \begin{cases} 1 & E(bel_f) > \underline{T} \text{ and } \sigma(t-dt) = 1 \\ 0 & E(bel_f) < \underline{T} \text{ and } \sigma(t-dt) = 1 \\ 1 & E(bel_f) > \overline{T} \text{ and } \sigma(t-dt) = 0 \\ 0 & E(bel_f) < \overline{T} \text{ and } \sigma(t-dt) = 0 \end{cases}, \tag{5.51}$$

and $\underline{T}$ and $\overline{T}$ are threshold values of trust of human in robot in the proactive and reactive modes, respectively. Initially robot starts at the proactive mode, i.e. $\sigma(0) = 1$. In this mode, if the human trust's belief decreases to lower than $\underline{T}$, then the robot switches to the reactive mode. In the reactive mode, the robot behaves complaint to the human and there is no disagreement and thus the human trust in robot increases fast. If the trust is high enough, i.e. greater than $\overline{T}$, the robot is switches back to the proactive mode.

5.4.2.5 Dynamic Role Allocation Control Strategy

In the trust-based dynamic role allocation control strategy, the desired behavior of the system is given by

$$\boldsymbol{M}_v \ddot{\boldsymbol{x}}(t) + \boldsymbol{D}_v \dot{\boldsymbol{x}}(t) = \boldsymbol{h}_h(t) + \alpha \boldsymbol{h}_v(t), \tag{5.52}$$

where α is the role allocation parameter and $\alpha \in [0, 1]$. We chose α to be a function of trust belief as

$$\alpha = E(bel_f). \tag{5.53}$$

This allocation policy ensures that the proactive behavior of the robot matches the human's expectation by adjusting the role of the robot in accordance to human's trust to the robot.

5.5 Simulation Study

A team of a human and a robot moves an object from the start to the goal position. We consider a scenario where the robot incorrectly assumes that there is an obstacle between the start and goal positions due to some perception faults and behaves accordingly. The human is not aware

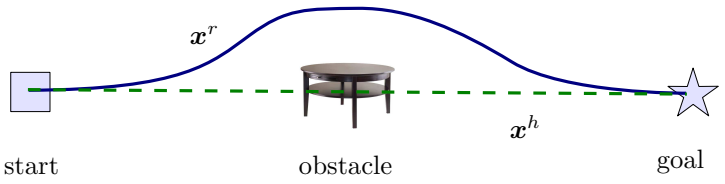


Figure 5.3: Simulation Scenario

of the misperception of the robot and his/her desired action is a to reach to the goal position via a direct trajectory from the start to the goal position. Fig. 5.3 shows the task scenario where $\mathbf{X}^h$ and $\boldsymbol{x}^r$ show the initial desired trajectory of the human and robot, respectively. The robot starts the collaboration in the proactive mode described in Section 5.4.1. The human trust in robot is estimated based on the DBN model for human-to-robot trust specified in Section 5.4.2.2. Considering the belief of the human's trust, bel_f, the robot uses the trust-based switching control strategy defined in Section 5.4.2.4 to switch between the reactive and proactive modes. Moreover, it uses the dynamic role allocation strategy defined in 5.4.2.5 for the same task. The goal of the simulation study is to evaluate the effectiveness of the proposed trust-based switching and dynamic role allocation control strategies in terms of disagreement between the human and the robot and human-to-robot trust's belief during interaction.

The robot controller implements the GP method described in Section 5.4.1 to predict the human desired trajectory in the proactive mode. We run human-human cooperative manipulation experiments to record the training data needed for constructing the GP models. We use PhaseSpace tracking system to track the position and orientation of the manipulated object. Here, we only consider the planar motion of the object in the $x-y$ plane with the translational components x and

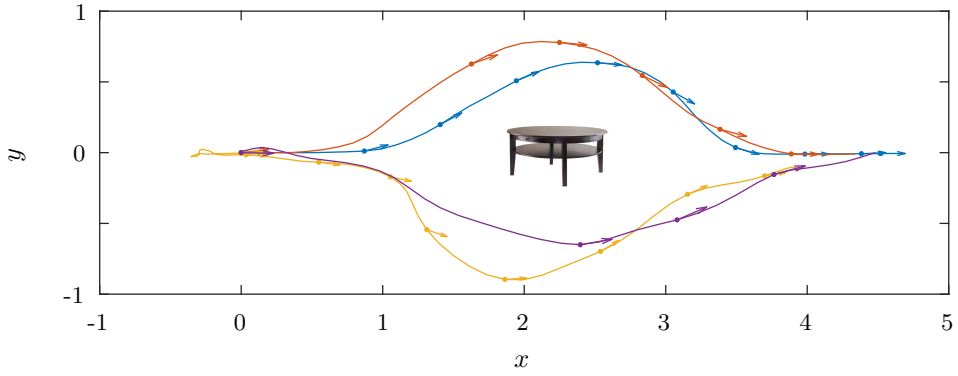
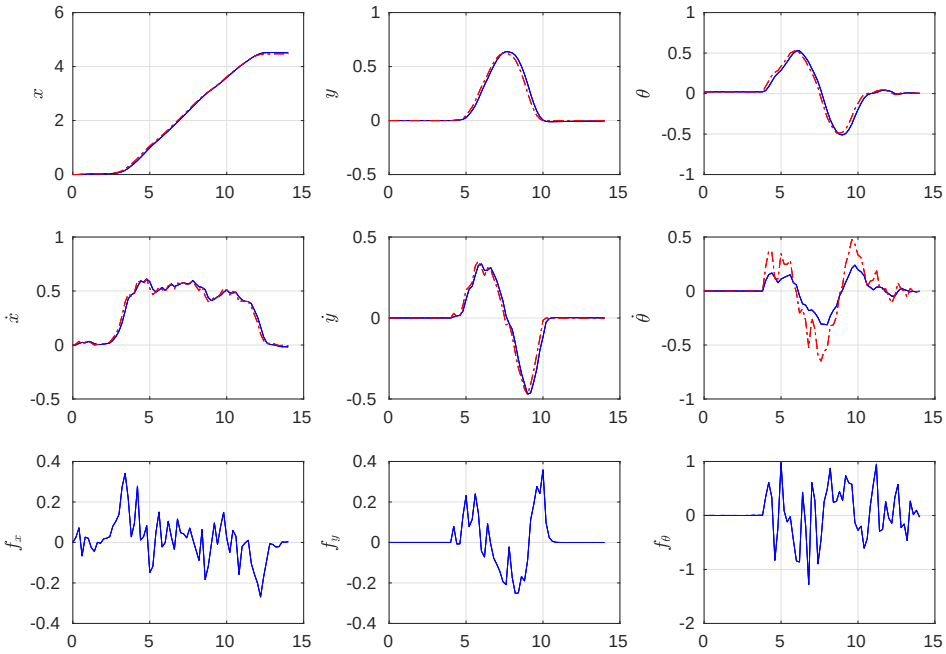


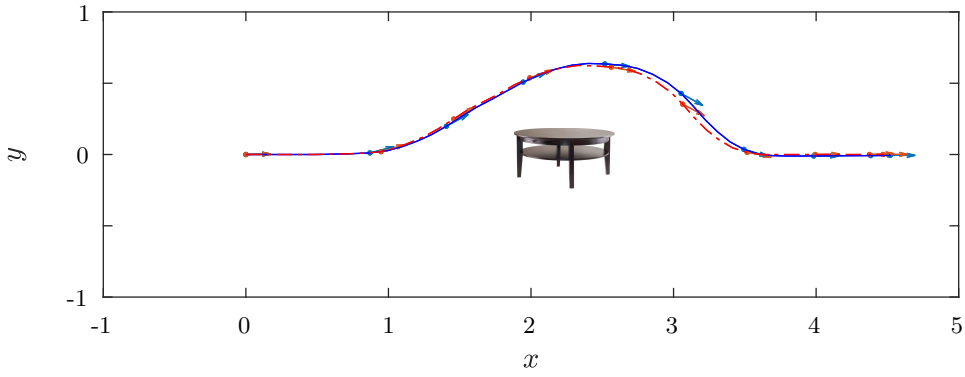
Figure 5.4: Motion trajectories from human-human cooperative manipulation

y and the rotation as θ, i.e.

$$\boldsymbol{x} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \quad \text{and} \quad q = 3. \tag{5.54}$$



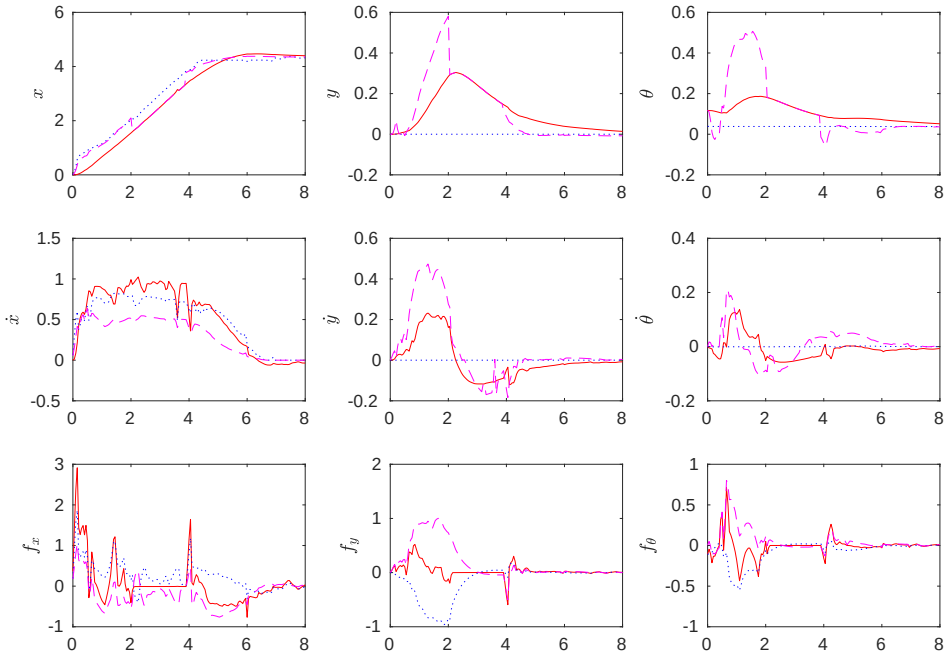
(a) Motion and force trajectories. The horizontal axes represent the time (seconds)



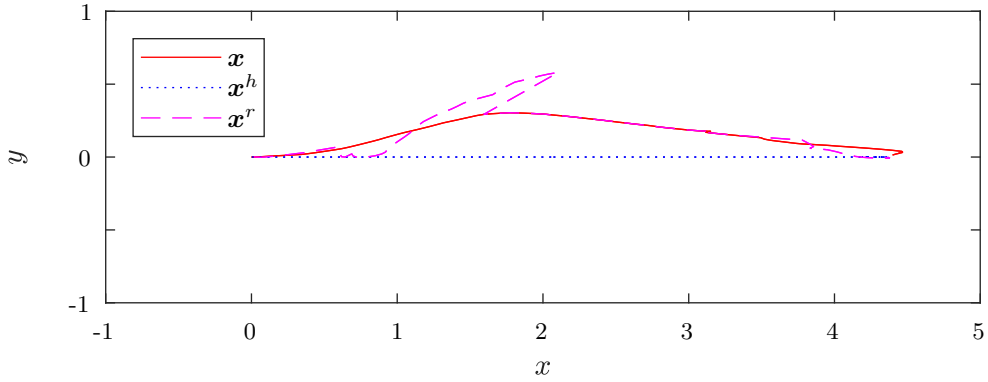
(b) Planar motion trajectory

Figure 5.5: Simulation of the applied force to the object. The dash-dotted red lines are the reference trajectories from the tracking system.

Figure 5.4 shows the motion trajectories used as the training data. The arrows depict the orientation of the object at some sample points on each trajectory and the table represents the



(a) Motion and force trajectories. The horizontal axes represent the time (seconds)



(b) Planar motion trajectoris for the object, $\boldsymbol{x}$, and desired trajectories for the human, $\boldsymbol{x}^h$, and the robot (in the proactive mode), $\boldsymbol{x}^r$.

Figure 5.6: Simulation results for the switching scenario. The solid red, dotted blue, and dashed magenta lines represent the object, human desired, and robot (desired) trajectories, respectively.

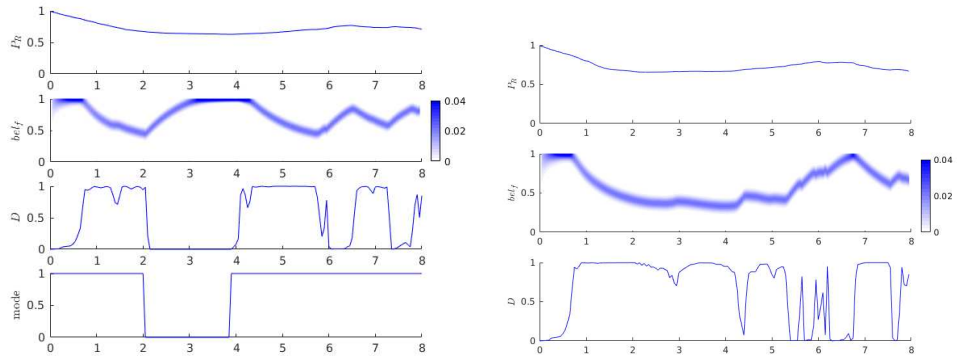
obstacle between the start and the goal positions. In order to construct the input of the GP models, $\boldsymbol{\xi}$, we need to have both motion and force trajectories. For the sake of simplicity of the training session, using the data from the motion trajectories, we simulate the forces applied to the object rather than measure them. Using a PID controller, we fit forces applied to the object based on motion data. The equation of motion of the object is considered as

$$\boldsymbol{M}_o\ddot{\hat{\boldsymbol{x}}} + \boldsymbol{C}_o\dot{\hat{\boldsymbol{x}}} = \hat{\boldsymbol{h}}_e, \tag{5.55}$$

where $\hat{\boldsymbol{x}}$ is the simulated position and $\hat{\boldsymbol{h}}_e$ is the force applied to the object, respectively. The controller is chosen as

$$\hat{\boldsymbol{h}}_e = \boldsymbol{K}_p \boldsymbol{e} + \boldsymbol{K}_i \int_0^t \boldsymbol{e}(\tau) d\tau + \boldsymbol{K}_d \dot{\boldsymbol{e}}, \tag{5.56}$$

where $\boldsymbol{e} = \boldsymbol{x} - \hat{\boldsymbol{x}}$, and $\boldsymbol{K}_p$, $\boldsymbol{K}_i$, and $\boldsymbol{K}_d$ are the gains of the PID controller. $\boldsymbol{x}$ is the reference motion of the object, i.e. the data from the tracking system. Fig.5.5 shows the results of force fitted based on data for a sample human-human training trajectory.

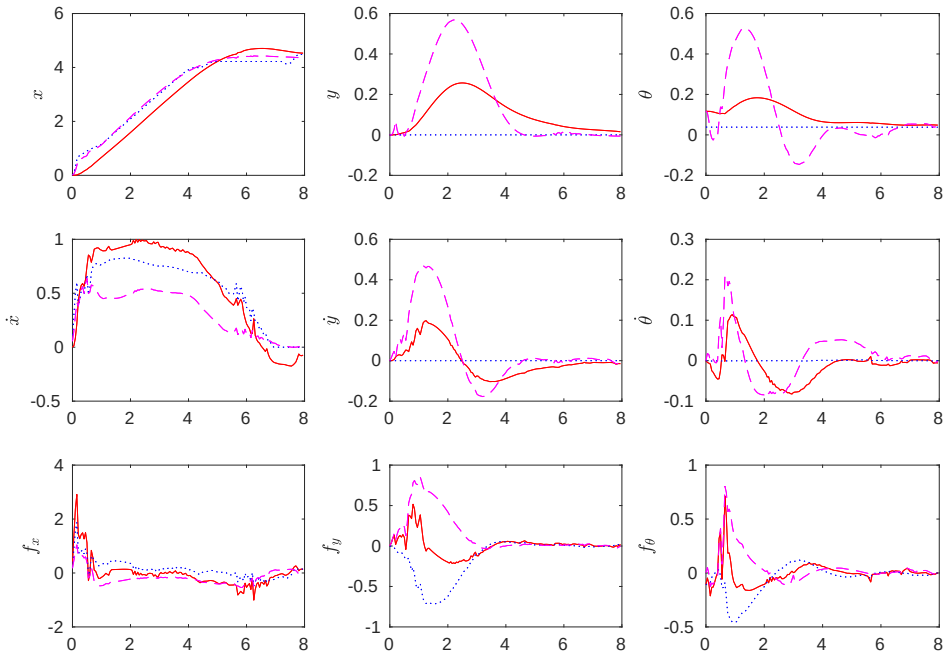


(a) (b)

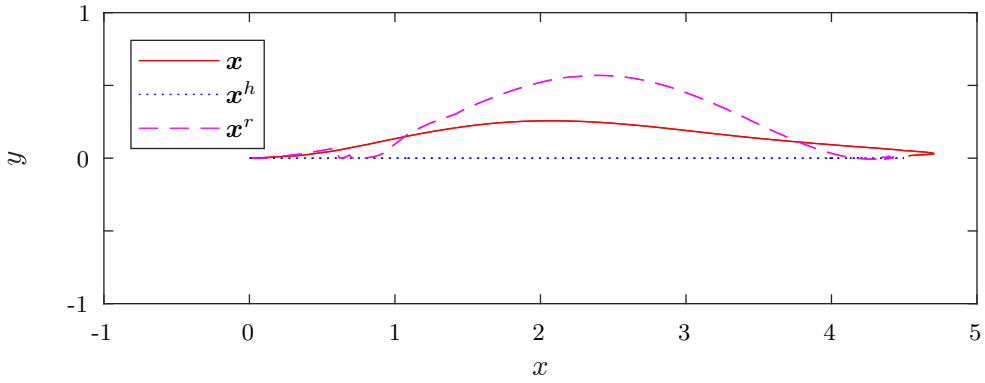
Figure 5.7: Evolution of robot performance P_R, trust belief bel_f and disagreement D for (a) switching-based and (b) dynamic role allocation scenarios.

The GP technique is also used for simulating the human desired motion and force during cooperative manipulation. Based on the scenario, the human desires to follow a straight line towards the goal. The results of the simulation study are presented in Fig. 5.6 and Fig. 5.7a for the switching role allocation study and in Fig. 5.8 and Fig. 5.7b. In the switching role allocation, the evolution of motion and force trajectories are plotted in Fig. 5.6. The human-robot team starts the cooperative manipulation in proactive mode. As there is no obstacle between the start point and the goal point in the task scenario, the desired motion predicted by the robot, $\boldsymbol{x}^r$, deviates from the human desired trajectory $\boldsymbol{x}^h$ and the actual motion of the object lies between these two trajectories. Therefore, the human and the robot have a disagreement during the scenario, since $\boldsymbol{x}^r$ and $\boldsymbol{x}^h$ are leading towards different directions.

For switching role allocation, the evolution of trust belief bel_f based on robot performance P_R and disagreement D, and control mode is plotted in Fig 5.7a. As we can see, there is a period of disagreement during the first 2 second. The trust belief drops gradually. At time instant 2



(a) Motion and force trajectories. The horizontal axes represent the time (seconds)



(b) Planar motion trajectoris for the object, $\boldsymbol{x}$, and desired trajectories for the human, $\boldsymbol{x}^h$, and the robot (in the proactive mode), $\boldsymbol{x}^r$.

Figure 5.8: Simulation results for the dynamic role allocation scenario. The solid red, dotted blue, and dashed magenta lines represent the object, human desired, and robot (desired) trajectories, respectively.

[s], the human-robot team switches to reactive mode, whereat the human takes the lead and the robot follows him/her. In this mode, the robot is completely compliant to the human and does not predict any desired trajectory. Since the robot follows the human's lead completely, there is no disagreement in this mode. The trust belief is increasing as a result and reaches close to 1 around the time instant 4 [s], where the team switches back to the proactive mode. The robot performance is high enough during the rest of the interaction and the robot stays in the proactive mode until the

goal is reached. The results indicate that the switching policy in case of faulty detection of human intent or environment, it is able to prevent further disagreements between the robot and human by being compliant to the human. Moreover, it switches back to the proactive mode once the human's trust belief is high.

For dynamic role allocation, the evolution of trust belief bel_f based on robot performance P_R and disagreement D is plotted in Fig 5.7b. As we can see, Since there is a mismatch between the feedforward prediction of and actual human desired motion there is an ongoing disagreement in the entire interaction. The trust belief drops gradually but recovers as the disagreement decreases.

5.6 Conclusion

In this chapter, a trust-based role allocation strategy for proactive behavior of the robot in human-robot collaborative manipulation was proposed. The human trust in robot was modeled using a BDN network. The human trust belief depends on previous trust, current and previous robot performance, and measured disagreement between the human and robot. The robot performance was defined based on relative internal force, disagreement, accuracy of the predicted human desired motion trajectory and robot effort. The dynamic and switching role allocation for the robot behavior was studied. In proactive role allocation, the robot adjusts its behavior from a complete reactive and follower robot to a proactive robot based on the expected value of human trust belief in robot. In switching role allocation, the robot switches between the proactive and reactive modes following the estimation of human trust in robot. A demonstration of the proposed control strategies were presented via a simulation study. Future work includes more comprehensive simulation scenarios and experimental studies.

Chapter 6

Conclusions and Future Works

6.1 Conclusions

The main contributions of this dissertation are divided in modeling human trust in robot and controlling the robot to maintain the trust and overall performance in human-robot collaboration tasks. For such tasks, human trust in robot was modeled both in deterministic and probabilistic manner. Trust-based frameworks for a team of a human and a robot performing manipulation, handover, and cooperative manipulation tasks were presented. Both performance metrics and interaction criteria were considered in the design on trust-based controllers. The results of subjective questionnaires including trust, usability, and NASA TLX (for workload) showed that integration of the interaction of the robot improves the human's overall experience. Moreover, it was shown that the performance of the trust-based controllers did not change significantly compared to the performance-based controllers. The detailed contributions of each chapter are as follows.

In Chapter 2, a time-series model for human trust in robot for HRI in performing manipulation and assembly tasks was proposed. This model was evaluated and verified through human-in-the-loop experiments for manual, autonomous, and collaborative speed control of the robot.

In Chapter 3, a trust-based framework for human-robot collaborative assembly tasks was presented. Both physical and social interaction considered in the robot controller. The framework was evaluated by conducting a set of human-in-the-loop robotic experiments.

In Chapter 4, a trust-based framework for human-robot handover tasks was proposed. A robot-to-human trust model was proposed to evaluate the performance of the human. A trust-based

handover strategy that balances safety and performance was presented for handover tasks. The framework was evaluated by conducting a set of human-in-the-loop robotic experiments.

In Chapter 5, a trust-based framework for human-cooperative manipulation tasks was proposed. Trust of human in robot was modeled using a probabilistic approach. A trust-based role allocation for the proactive behavior of the robot was proposed.

6.2 Future Work

In this dissertation we implemented the concept of trust in human-robot interaction. For modeling human trust in robot we mostly relied on motion and force comparisons for human and robot performance criteria. As a future work vision-based fault detection can be added to the trust model. Moreover, a similar probabilistic trust model described in Chapter 5, can be deployed for the manipulation framework described in Chapter 3.

As another future work, a human-in-the-loop robotic study on human-robot cooperative manipulation can be conducted to experimentally study the role allocation scheme proposed in Chapter 5. Fig. 6.1 shows a schematic of the manipulation object with design of versatile grippers with force sensors for the human and robotic arms.

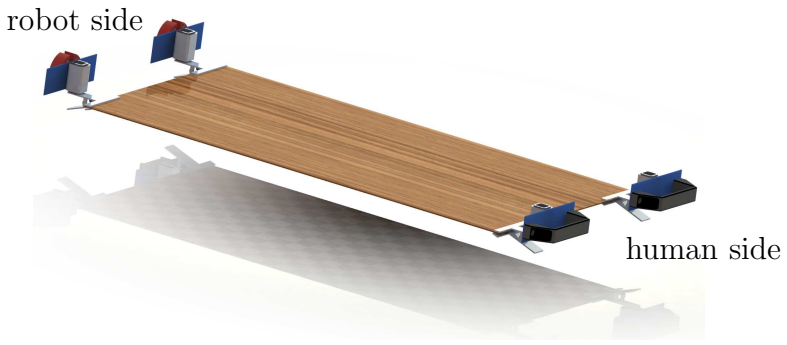


Figure 6.1: Versatile gripper for force measurement

Appendices

Appendix A Subjective Questionnaires

A.1 Demographic Questionnaire

Participant's Section: Please answer the following 5 questions

1. Are you a male or female?

   ◯ Male

   ◯ Female

   ◯ Decline to answer

2. What year were you born?

3. What is the highest level of school you have completed or the highest degree you have received?

   ◯ Less than a high school degree

   ◯ High school degree or equivalent (e.g., GED)

   ◯ Some college but no degree

   ◯ Associate's degree

   ◯ Bachelor's degree

   ◯ Graduate degree

4. Are you White, Black or African-American, American Indian or Alaskan Native, Asian, Native Hawaiian or other Pacific islander, or some other race?

   ◯ White

   ◯ Black or African-American

   ◯ American Indian or Alaskan Native

   ◯ Asian

   ◯ Native Hawaiian or other Pacific Islander

   ◯ Form multiple races

   ◯ Some other race

5. Do you have experience working with a robot?

   ◯ Yes

   ◯ No

6. 6) Occupation?

A.2 System Usability Questionnaire

Participant's Section: Please answer the following 10 questions

Note: The interface includes those items that you use to interact with the system.

1. Overall, I am satisfied with how easy it is to collaborate with this robot.

	1	2	3	4	5	6	7	
Strongly Disagree	◯	◯	◯	◯	◯	◯	◯	Strongly Agree

2. It is simple to collaborate with this robot.

	1	2	3	4	5	6	7	
Strongly Disagree	◯	◯	◯	◯	◯	◯	◯	Strongly Agree

3. I can effectively complete my work collaborating with this robot.

	1	2	3	4	5	6	7	
Strongly Disagree	◯	◯	◯	◯	◯	◯	◯	Strongly Agree

4. I am able to complete my work quickly collaborating with this robot.

	1	2	3	4	5	6	7	
Strongly Disagree	◯	◯	◯	◯	◯	◯	◯	Strongly Agree

5. I am able to efficiently complete my work collaborating with this robot.

	1	2	3	4	5	6	7	
Strongly Disagree	◯	◯	◯	◯	◯	◯	◯	Strongly Agree

6. I feel comfortable collaborating with this robot.

	1	2	3	4	5	6	7	
Strongly Disagree	◯	◯	◯	◯	◯	◯	◯	Strongly Agree

7. It was easy to learn to collaborate with this robot.

	1	2	3	4	5	6	7	
Strongly Disagree	◯	◯	◯	◯	◯	◯	◯	Strongly Agree

8. I believe I became productive quickly collaborating with this robot.

	1	2	3	4	5	6	7	
Strongly Disagree	◯	◯	◯	◯	◯	◯	◯	Strongly Agree

9. The control interface (arm and screen) of this robot is pleasant.

	1	2	3	4	5	6	7	
Strongly Disagree	◯	◯	◯	◯	◯	◯	◯	Strongly Agree

10. Overall, I am satisfied with this robot.

	1	2	3	4	5	6	7	
Strongly Disagree	◯	◯	◯	◯	◯	◯	◯	Strongly Agree

A.3 Subjective Trust Questionnaire

Participant's Section: Please answer the following 12 questions

1. The robot' motion is deceptive

	1	2	3	4	5	6	7	
Not at all	◯	◯	◯	◯	◯	◯	◯	Extremely

2. The robot moves in an underhanded manner

	1	2	3	4	5	6	7	
Not at all	◯	◯	◯	◯	◯	◯	◯	Extremely

3. I am suspicious of the robot's intent, action, or outputs

	1	2	3	4	5	6	7	
Not at all	◯	◯	◯	◯	◯	◯	◯	Extremely

4. I am wary of the robot' motion

	1	2	3	4	5	6	7	
Not at all	◯	◯	◯	◯	◯	◯	◯	Extremely

5. The robot's actions will have a harmful or injurious outcome

	1	2	3	4	5	6	7	
Not at all	◯	◯	◯	◯	◯	◯	◯	Extremely

6. I am confident in the robot

	1	2	3	4	5	6	7	
Not at all	◯	◯	◯	◯	◯	◯	◯	Extremely

7. The robot provides security

	1	2	3	4	5	6	7	
Not at all	◯	◯	◯	◯	◯	◯	◯	Extremely

8. The robot' motion has integrity

	1	2	3	4	5	6	7	
Not at all	◯	◯	◯	◯	◯	◯	◯	Extremely

9. The robot' motion is dependable

	1	2	3	4	5	6	7	
Not at all	◯	◯	◯	◯	◯	◯	◯	Extremely

10. The robot' motion is reliable

	1	2	3	4	5	6	7	
Not at all	◯	◯	◯	◯	◯	◯	◯	Extremely

11. I can trust the robot

	1	2	3	4	5	6	7	
Not at all	◯	◯	◯	◯	◯	◯	◯	Extremely

12. I am familiar with the robot

	1	2	3	4	5	6	7	
Not at all	◯	◯	◯	◯	◯	◯	◯	Extremely

Bibliography

[1] ABB. ABB introduces YuMi, world's first truly collaborative dual-arm robot, 2015.

[2] Ronald Craig Arkin, Patrick Ulam, and Alan R Wagner. Moral decision making in autonomous systems: Enforcement, moral emotions, dignity, trust, and deception. *Proceedings of the IEEE*, 100(3):571–589, 2012.

[3] Patrizia Basili, Markus Huber, Thomas Brandt, Sandra Hirche, and Stefan Glasauer. Investigating human-human approach and hand-over. In *Human centered robot systems*, pages 151–160. Springer, 2009.

[4] Adi Ben-Israel and Thomas NE Greville. *Generalized inverses: theory and applications*, volume 15. Springer Science & Business Media, 2003.

[5] Rainer Bischoff, Johannes Kurth, Günter Schreiber, Ralf Koeppe, Alin Albu-Schäffer, Alexander Beyer, Oliver Eiberger, Sami Haddadin, Andreas Stemmer, Gerhard Grunwald, et al. The kuka-dlr lightweight robot arm-a new reference platform for robotics research and manufacturing. In *Robotics (ISR), 2010 41st international symposium on and 2010 6th German conference on robotics (ROBOTIK)*, pages 1–8. VDE, 2010.

[6] RC Bonitz and Tien C Hsia. Internal force-based impedance control for cooperating manipulators. *IEEE Transactions on Robotics and Automation*, 12(1):78–89, 1996.

[7] Cynthia Breazeal. Emotion and sociable humanoid robots. *International Journal of Human-Computer Studies*, 59(1):119–155, 2003.

[8] Cynthia Breazeal. Social interactions in HRI: the robot view. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 34(2):181–186, 2004.

[9] Cynthia Breazeal and Rodney Brooks. Robot emotion: A functional perspective. *Who needs emotions*, pages 271–310, 2005.

[10] David J Bruemmer, David I Gertman, Curtis W Nielsen, Douglas A Few, and William D Smart. Supporting complex robot behaviors with simple interaction tools. In *Human Robot Interaction*. InTech, 2007.

[11] Antoine Bussy, Pierre Gergondet, Abderrahmane Kheddar, François Keith, and André Crosnier. Proactive behavior of a humanoid robot in a haptic transportation task with a human partner. In *RO-MAN, 2012 IEEE*, pages 962–967. IEEE, 2012.

[12] Fei Chen, Kosuke Sekiyama, Ferdinando Cannella, and Toshio Fukuda. Optimal subtask allocation for human and robot collaboration within hybrid assembly system. *IEEE Transactions on Automation Science and Engineering*, 11(4):1065–1075, 2014.

[13] Andrea Cherubini, Robin Passama, André Crosnier, Antoine Lasnier, and Philippe Fraisse. Collaborative manufacturing with physical human–robot interaction. *Robotics and Computer-Integrated Manufacturing*, 40:1–13, 2016.

[14] Pasquale Chiacchio, Stefano Chiaverini, Lorenzo Sciavicco, and Bruno Siciliano. Closed-loop inverse kinematics schemes for constrained redundant manipulators with task space augmentation and task priority strategy. *The International Journal of Robotics Research*, 10(4):410–425, 1991.

[15] A. Cichocki and R. Unbehauen. Robust neural networks with on-line learning for blind identification and blind separation of sources. *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, 43(11):894–906, Nov 1996.

[16] Brecht Corteville, Erwin Aertbeliën, Herman Bruyninckx, Joris De Schutter, and Hendrik Van Brussel. Human-inspired robot assistant for fast point-to-point movements. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 3639–3644. IEEE, 2007.

[17] Arati S Deo and Ian D Walker. Overview of damped least-squares methods for inverse kinematics of robot manipulators. *Journal of Intelligent and Robotic Systems*, 14(1):43–68, 1995.

[18] Paul Evrard and Abderrahmane Kheddar. Homotopy switching model for dyad haptic interaction in physical collaborative tasks. In *EuroHaptics conference, 2009 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics 2009. Third Joint*, pages 45–50. IEEE, 2009.

[19] S.Alireza Fayazi, Nianfeng Wan, Stephen Lucich, Ardalan Vahidi, and Gregory Mocko. Optimal pacing in a cycling time-trial considering cyclist's fatigue dynamics. In *American Control Conference (ACC), 2013*, pages 6442–6447, 2013.

[20] Tamar Flash and Neville Hogan. The coordination of arm movements: an experimentally confirmed mathematical model. *Journal of neuroscience*, 5(7):1688–1703, 1985.

[21] Terrence Fong, Illah Nourbakhsh, and Kerstin Dautenhahn. A survey of socially interactive robots. *Robotics and autonomous systems*, 42(3):143–166, 2003.

[22] Ji Gao and John D Lee. Extending the decision field theory to model operators' reliance on automation in supervisory control situations. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 36(5):943–959, 2006.

[23] Michael A Goodrich and Alan C Schultz. Human-robot interaction: a survey. *Foundations and Trends in Human-Computer Interaction*, 1(3):203–275, 2007.

[24] Lars Grüne and Jürgen Pannek. *Nonlinear model predictive control*. Springer, 2011.

[25] E Guizzo. Sawyer: Rethink robotics unveils new robot. *Spectrum, IEEE*, 2015.

[26] Erico Guizzo and Evan Ackerman. How rethink robotics built its new baxter robot worker. *IEEE Spectrum*, 2012.

[27] Erico Guizzo and Evan Ackerman. How rethink robotics built its new baxter robot worker. *IEEE Spectrum, http://spectrum. ieee. org/robotics/industrial-robots/rethink-robotics-baxter-robot-factory-worker (retrieved January 21, 2014)*, 2012.

[28] PJ Hacksel and SE Salcudean. Estimation of environment forces and rigid-body velocities using observers. In *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, pages 931–936. IEEE, 1994.

[29] M.T. Hagan and H.B. Demuth. Neural networks for control. In *American Control Conference, 1999. Proceedings of the 1999*, volume 3, pages 1642–1656 vol.3, 1999.

[30] Peter A Hancock, Deborah R Billings, Kristin E Schaefer, Jessie YC Chen, Ewart J De Visser, and Raja Parasuraman. A meta-analysis of factors affecting trust in human-robot interaction. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 53(5):517–527, 2011.

[31] Sandra G Hart and Lowell E Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. *Advances in psychology*, 52:139–183, 1988.

[32] Yasuhisa Hirata, Youhei Kume, Zhi-Dong Wang, and Kazuhiro Kosuge. Decentralized control of multiple mobile manipulators based on virtual 3-d caster motion for handling an object in cooperation with a human. In *IEEE International Conference on Robotics and Automation, 2003.*, volume 1, pages 938–943. IEEE, 2003.

[33] K. A. Hoff and M. Bashir. Trust in automation: Integrating empirical evidence on factors that influence trust. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 2014.

[34] Guy Hoffman and Cynthia Breazeal. Cost-based anticipatory action selection for human–robot fluency. *IEEE transactions on robotics*, 23(5):952–961, 2007.

[35] Yanjiang Huang, Yoon Seong Yong, Ryosuke Chiba, Tamio Arai, Tsuyoshi Ueyama, and Jun Ota. Kinematic control with singularity avoidance for teaching-playback robot manipulator system. *IEEE Transactions on Automation Science and Engineering*, 13(2):729–742, 2016.

[36] Hardianto Iridiastadi and Maury A Nussbaum. Muscle fatigue and endurance during repetitive intermittent static efforts: development of prediction models. *Ergonomics*, 49(4):344–360, 2006.

[37] Makoto Itoh and Kenji Tanaka. Mathematical modeling of trust in automation: Trust, distrust, and mistrust. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 44, pages 9–12. SAGE Publications, 2000.

[38] Jiun-Yin Jian, Ann M Bisantz, and Colin G Drury. Foundations for an empirically determined scale of trust in automated systems. *International Journal of Cognitive Ergonomics*, 4(1):53–71, 2000.

[39] Lars Johannsmeier and Sami Haddadin. A hierarchical human-robot interaction-planning framework for task allocation in collaborative industrial assembly processes. *IEEE Robotics and Automation Letters*, 2(1):41–48, 2017.

[40] Juš Kocijan. *Modelling and control of dynamic systems using Gaussian process models*. Springer, 2016.

[41] K Kosuge, H Yoshida, and T Fukuda. Dynamic control for robot-human collaboration. In *Robot and Human Communication, 1993. Proceedings., 2nd IEEE International Workshop on*, pages 398–401. IEEE, 1993.

[42] J Krüger, V Katschinski, D Surdilovic, and G Schreck. Flexible assembly systems through workplace-sharing and time-sharing human machine cooperative. *ISR/ROBOTIK*, 2010.

[43] J Krüger, TK Lien, and A Verl. Cooperation of human and machines in assembly lines. *CIRP Annals-Manufacturing Technology*, 58(2):628–646, 2009.

[44] Ayse Kucukyilmaz, Tevfik Metin Sezgin, and Cagatay Basdogan. Intention recognition for dynamic role exchange in haptic collaboration. *IEEE transactions on haptics*, 6(1):58–68, 2013.

[45] Bokeon Kwak, Hyunkyoo Park, and Joonbum Bae. Development of a quadruped robot with redundant dofs for high-degree of functionality and adaptation. In *Advanced Intelligent Mechatronics (AIM), 2016 IEEE International Conference on*, pages 608–613. IEEE, 2016.

[46] Morteza Lahijanian and Marta Kwiatkowska. Social trust: a major challenge for the future of autonomous systems. In *AAAI Fall Symposium on Cross-Disciplinary Challenges for Autonomous Systems. AAAI*, 2016.

[47] Muriel Lang, Martin Kleinsteuber, Oliver Dunkley, and Sandra Hirche. Gaussian process dynamical models over dual quaternions. In *Control Conference (ECC), 2015 European*, pages 2847–2852. IEEE, 2015.

[48] Steven M LaValle. *Planning algorithms*. Cambridge university press, 2006.

[49] John Lee and Neville Moray. Trust, control strategies and allocation of function in human-machine systems. *Ergonomics*, 35(10):1243–1270, 1992.

[50] John D Lee and Neville Moray. Trust, self-confidence, and operators' adaptation to automation. *International journal of human-computer studies*, 40(1):153–184, 1994.

[51] John D Lee and Katrina A See. Trust in automation: Designing for appropriate reliance. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 46(1):50–80, 2004.

[52] John D Lee and Katrina A See. Trust in automation: Designing for appropriate reliance. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 46(1):50–80, 2004.

[53] Stephan Lewandowsky, Michael Mundy, and Gerard Tan. The dynamics of trust: comparing humans to automation. *Journal of Experimental Psychology: Applied*, 6(2):104, 2000.

[54] James R Lewis. IBM computer usability satisfaction questionnaires: psychometric evaluation and instructions for use. *International Journal of Human-Computer Interaction*, 7(1):57–78, 1995.

[55] Jing Z. Liu, Robert W. Brown, and Guang H. Yue. A dynamical model of muscle activation, fatigue, and recovery. *Biophysical Journal*, 82(5):2344 – 2359, 2002.

[56] Lennart Ljung. System identification toolbox for use with {MATLAB}. 2007.

[57] Chen Change Loy, Tao Xiang, and Shaogang Gong. Modelling multi-object activity by gaussian processes. In *BMVC*, pages 1–11. Citeseer, 2009.

[58] Ingo Lütkebohle, Frank Hegel, Simon Schulz, Matthias Hackel, Britta Wrede, Sven Wachsmuth, and Gerhard Sagerer. The bielefeld anthropomorphic robot head" flobi". In *ICRA*, volume 3, pages 3384–3391, 2010.

[59] Liang Ma, Damien Chablat, Fouad Bennis, and Wei Zhang. A new simple dynamic muscle fatigue model and its validation. *International Journal of Industrial Ergonomics*, 39(1):211 – 220, 2009.

[60] Liang Ma, Damien Chablat, Fouad Bennis, Wei Zhang, and Franois Guillaume. A new muscle fatigue and recovery model and its ergonomics application in human simulation. *Virtual and Physical Prototyping*, 5(3):123–137, 2010.

[61] Cigil Ece Madan, Ayse Kucukyilmaz, Metin Sezgin, and Cagatay Basdogan. Recognition of haptic interaction patterns in dyadic joint object manipulation. *IEEE transactions on haptics*, (1):1–1, 2015.

[62] Yusuke Maeda, Takayuki Hara, and Tamio Arai. Human-robot cooperative manipulation with motion estimation. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, volume 4, pages 2240–2245. Ieee, 2001.

[63] Maziar Fooladi Mahani and Yue Wang. Runtime verification of trust-based symbolic robot motion planning with human-in-the-loop. In *ASME 2016 Dynamic Systems and Control Conference*, pages V002T24A008–V002T24A008. American Society of Mechanical Engineers, 2016.

[64] Jim Mainprice, Rafi Hayne, and Dmitry Berenson. Goal set inverse optimal control and iterative re-planning for predicting human reaching motions in shared workspaces. *arXiv preprint arXiv:1606.02111*, 2016.

[65] Jeremy A Marvel. Performance metrics of speed and separation monitoring in shared workspaces. *IEEE Transactions on Automation Science and Engineering*, 10(2):405–414, 2013.

[66] J. R. Medina, T. Lorenz, and S. Hirche. Synthesizing anticipatory haptic assistance considering human behavior uncertainty. *IEEE Transactions on Robotics*, 31(1):180–190, Feb 2015.

[67] José Ramón Medina, Satoshi Endo, and Sandra Hirche. Impedance-based gaussian processes for predicting human behavior during physical interaction. In *IEEE International Conference on Robotics and Automation*. IEEE, 2016.

[68] José Ramón Medina, Tamara Lorenz, and Sandra Hirche. Considering human behavior uncertainty and disagreements in human–robot cooperative manipulation. In *Trends in Control and Decision-Making for Human–Robot Collaboration Systems*, pages 207–240. Springer, 2017.

[69] Kishan Mehrotra, Chilukur K Mohan, and Sanjay Ranka. *Elements of artificial neural networks*. MIT Press, 1997.

[70] S. M. Merritt and D. R. Ilgen. Not all trust is created equal: Dispositional and history-based trust in human-automation interactions. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 50(2):194–210, 2008.

[71] George Michalos, Sotiris Makris, Jason Spiliotopoulos, Ioannis Misios, Panagiota Tsarouchi, and George Chryssolouris. Robo-partner: Seamless human-robot cooperation for intelligent, flexible and safe operations in the assembly factories of the future. *Procedia CIRP*, 23:71–76, 2014.

[72] R. R. Mizanoor and Yue Wang. Dynamic affection-based motion control of a humanoid robot to collaborate with human in flexible assembly in manufacturingt. In *ASME Dynamic Systems and Controls Conference*. American Society of Mechanical Engineers, 2015.

[73] Neville Moray, Toshiyuki Inagaki, and Makoto Itoh. Adaptive automation, trust, and self-confidence in fault management of time-critical tasks. *Journal of Experimental Psychology: Applied*, 6:44, 2000.

[74] Masashiro Morioka and Shinsuke Sakakibara. A new cell production assembly system with human–robot cooperation. *CIRP Annals-Manufacturing Technology*, 59(1):9–12, 2010.

[75] Alexander Mörtl, Martin Lawitzky, Ayse Kucukyilmaz, Metin Sezgin, Cagatay Basdogan, and Sandra Hirche. The role of roles: Physical cooperation between humans and robots. *The International Journal of Robotics Research*, 31(13):1656–1674, 2012.

[76] Richard M Murray, Zexiang Li, and S Shankar Sastry. *A mathematical introduction to robotic manipulation*. CRC press, 1994.

[77] Shuichi Nishio, Hiroshi Ishiguro, and Norihiro Hagita. *Geminoid: Teleoperated android of an existing person*. INTECH Open Access Publisher Vienna, 2007.

[78] Esben H Ostergaard. Lightweight robot for everybody [industrial activities]. *Robotics & Automation Magazine, IEEE*, 19(4):17–18, 2012.

[79] Carlene M Perry, Mohamed A Sheik-Nainar, Noa Segall, Ruiqi Ma, and David B Kaber. Effects of physical workload on cognitive task performance and situation awareness. *Theoretical Issues in Ergonomics Science*, 9(2):95–113, 2008.

[80] L. Peternel, N. Tsagarakis, D. Caldwell, and A. Ajoudani. Adaptation of robot physical behaviour to human fatigue in human-robot co-manipulation. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 489–494, Nov 2016.

[81] Friedrich Pfeiffer and Rainer Johanni. A concept for manipulator trajectory planning. *Robotics and Automation, IEEE Journal of*, 3(2):115–123, 1987.

[82] Rosalind W Picard. Affective computing for hci. In *HCI (1)*, pages 829–833, 1999.

[83] Rosalind W Picard and Roalind Picard. *Affective computing*, volume 252. MIT press Cambridge, 1997.

[84] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009.

[85] MM Rahman, R Ikeura, and K Mizutani. Control characteristics of two humans in cooperative task. In *Systems, Man, and Cybernetics, 2000 IEEE International Conference on*, volume 2, pages 1301–1306. IEEE, 2000.

[86] SM Mizanoor Rahman, Behzad Sadrfaridpour, and Yue Wang. Trust-based optimal subtask allocation and model predictive control for human-robot collaborative assembly in manufacturing. In *ASME 2015 Dynamic Systems and Control Conference*. ASME, 2015.

[87] SM Mizanoor Rahman, Behzad Sadrfaridpour, Yue Wang, Ian Walker, Laine Mears, Rishard Pak, and Sekou Remy. Trust-triggered robot-human handovers using kinematics redundancy for collaborative assembly in flexible manufacturing.

[88] SM Mizanoor Rahman, Yue Wang, Ian D Walker, Laine Mears, Richard Pak, and Sekou Remy. Trust-based compliant robot-human handovers of payloads in collaborative assembly in flexible manufacturing. In *Automation Science and Engineering (CASE), 2016 IEEE International Conference on*, pages 355–360. IEEE, 2016.

[89] Rethink Robotics. http://www.rethinkrobotics.com/.

[90] Behzad Sadrfaridpour, Jenny Burke, and Yue Wang. Human and robot collaborative assembly manufacturing: Trust dynamics and control. In *RSS 2014 Workshop on Human-Robot Collaboration for Industrial Manufacturing*, 2014.

[91] Behzad Sadrfaridpour, Hamed Saeidi, Jenny Burke, Kapil Madathil, and Yue Wang. *Robust Intelligence and Trust in Autonomous Systems*, chapter 7. Modeling and Control of Trust in Human-Robot Collaborative Manufacturing, pages 115–141. Springer US, 2016.

[92] Behzad Sadrfaridpour, Hamed Saeidi, Yue Wang, and Jenny Burke. Modeling and control of trust in human and robot collaborative manufacturing. In *2014 AAAI Spring Symposium Series*, 2014.

[93] Behzad Sadrfaridpour, Hamed Saiedi, and Yue Wang. An integrated framework for human-robot collaborative assembly in hybrid manufacturing cells. In *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 462–467, Aug 2016.

[94] Behzad Sadrfaridpour and Yue Wang. Collaborative assembly in hybrid manufacturing cells: An integrated framework for humanrobot interaction. *IEEE Transactions on Automation Science and Engineering*, 15(3):1178–1192, July 2018.

[95] Ehsan Sadrfaridpour, Sandeep Jeereddy, Ken Kennedy, Andre Luckow, Talayeh Razzaghi, and Ilya Safro. Algebraic multigrid support vector machines. *arXiv preprint arXiv:1611.05487*, 2016.

[96] Ehsan Sadrfaridpour, Talayeh Razzaghi, and Ilya Safro. Engineering multilevel support vector machines. *arXiv preprint arXiv:1707.07657*, 2017.

[97] H Saeidi and Y Wang. Trust and self-confidence based autonomy allocation for robotic systems. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 6052–6057. IEEE, 2015.

[98] Hamed Saeidi. *Trust-Based Control of (Semi)Autonomous Mobile Robotic Systems*. PhD thesis, 2016.

[99] Hamed Saeidi, Foster McLane, Behzad Sadrfaridpour, Evan Sand, Sheng Fu, Julio Rodriguez, John R. Wagner, and Yue Wang. Trust-based mixed-initiative teleoperation of mobile robots. In *2016 American Control Conference*. IEEE, 2016.

[100] Hamed Saeidi, John R Wagner, and Yue Wang. A mixed-initiative haptic teleoperation strategy for mobile robotic systems based on bidirectional computational trust analysis. *IEEE Transactions on Robotics*, 33(6):1500–1507, 2017.

[101] Stanley A Schneider and Robert H Cannon. Object impedance control for cooperative manipulation: Theory and experimental results. *IEEE Transactions on Robotics and Automation*, 8(3):383–394, 1992.

[102] Natalie Sebanz and Guenther Knoblich. Prediction in joint action: What, when, and where. *Topics in Cognitive Science*, 1(2):353–367, 2009.

[103] Ruben Seyboldt, Christian Frese, and Angelika Zube. Sampling-based path planning to cartesian goal positions for a mobile manipulator exploiting kinematic redundancy. In *ISR 2016: 47st International Symposium on Robotics; Proceedings of*, pages 1–9. VDE, 2016.

[104] Stuart Shepherd and Alois Buchstab. *Robotic Fabrication in Architecture, Art and Design 2014*, chapter KUKA Robots On-Site, pages 373–380. Springer International Publishing, Cham, 2014.

[105] Jane Shi, Glenn Jimmerson, Tom Pearson, and Roland Menassa. Levels of human and robot collaboration for automotive manufacturing. In *Proceedings of the Workshop on Performance Metrics for Intelligent Systems*, PerMIS '12, pages 95–100, New York, NY, USA, 2012. ACM.

[106] Mark W Spong, Seth Hutchinson, and Mathukumalli Vidyasagar. *Robot modeling and control*, volume 3. Wiley New York, 2006.

[107] Nikolay Stefanov, Angelika Peer, and Martin Buss. Online intention recognition for computer-assisted teleoperation. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 5334–5339. IEEE, 2010.

[108] Aaron Steinfeld, Terrence Fong, David Kaber, Michael Lewis, Jean Scholtz, Alan Schultz, and Michael Goodrich. Common metrics for human-robot interaction. In *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, pages 33–40. ACM, 2006.

[109] Andreas Stopp, Sven Horstmann, Steen Kristensen, and Frieder Lohnert. Towards interactive learning for manufacturing assistants. In *Robot and Human Interactive Communication, 2001. Proceedings. 10th IEEE International Workshop on*, pages 338–342. IEEE, 2001.

[110] Manida Swangnetr and David B Kaber. Emotional state classification in patient–robot interaction using wavelet analysis and statistics-based feature selection. *IEEE Transactions on Human-Machine Systems*, 43(1):63–75, 2013.

[111] Tomohito Takubo, Hirohiko Arai, Yasuo Hayashibara, and Kazuo Tanie. Human-robot cooperative manipulation using a virtual nonholonomic constraint. *The International Journal of Robotics Research*, 21(5-6):541–553, 2002.

[112] Jeffrey Too Chuan Tan, Feng Duan, Ye Zhang, Kei Watanabe, Ryu Kato, and Tamio Arai. Human-robot collaboration in cellular manufacturing: design and development. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 29–34. IEEE, 2009.

[113] Arun K Tangirala. *Principles of System Identification: Theory and Practice*. CRC Press, 2014.

[114] Panagiota Tsarouchi, Alexandros-Stereos Matthaiakis, Sotiris Makris, and George Chryssolouris. On a human-robot collaboration in an assembly cell. *International Journal of Computer Integrated Manufacturing*, pages 1–10, 2016.

[115] Masaru Uchiyama and Pierre Dauchez. A symmetric hybrid position/force control scheme for the coordination of two robots. In *IEEE International Conference on Robotics and Automation*, pages 350–356. IEEE, 1988.

[116] Albert van Breemen, Xue Yan, and Bernt Meerbeek. icat: an animated user-interface robot with personality. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 143–144. ACM, 2005.

[117] I. D. Walker, L. Mears, R. Mizanoor, R. Pak, S. Remy, and Y. Wang. Robot-human handovers based on trust. In *2nd International Conference Mathematics and Computers in Sciences and Industry*, 2015.

[118] Ian D Walker. Impact configurations and measures for kinematically redundant and multiple armed robot systems. *IEEE transactions on robotics and automation*, 10(5):670–683, 1994.

[119] Jack M Wang, David J Fleet, and Aaron Hertzmann. Gaussian process dynamical models for human motion. *IEEE transactions on pattern analysis and machine intelligence*, 30(2):283–298, 2008.

[120] Y. Wang, L. Humphrey, Z. Liao, and H. Zheng. Trust-based multi-robot symbolic motion planning with a human-in-the-loop. *ACM Transactions on Interactive Intelligent Systems*, 2018.

[121] Zheng Wang, Angelika Peer, and Martin Buss. An hmm approach to realistic haptic human-robot interaction. In *EuroHaptics conference, 2009 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics 2009. Third Joint*, pages 374–379. IEEE, 2009.

[122] Halbert White. *Artificial neural networks: approximation and learning theory*. Blackwell Publishers, Inc., 1992.

[123] Bernard Widrow, David E. Rumelhart, and Michael A. Lehr. Neural networks: Applications in industry, business and science. *Commun. ACM*, 37(3):93–105, March 1994.

[124] Andrew D Wilson, Jarvis A Schultz, Alex R Ansari, and Todd D Murphey. Dynamic task execution using active parameter identification with the baxter research robot. *IEEE Transactions on Automation Science and Engineering*, 14(1):391–397, 2017.

[125] Anqi Xu and Gregory Dudek. Optimo: Online probabilistic trust inference model for asymmetric human-robot collaborations. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, pages 221–228. ACM, 2015.

[126] Tsuneo Yoshikawa. Analysis and control of robot manipulators with redundancy. In *Robotics research: the first international symposium*, pages 735–747. Mit Press Cambridge, MA, USA, 1984.

[127] James E Young, JaYoung Sung, Amy Voida, Ehud Sharlin, Takeo Igarashi, Henrik I Christensen, and Rebecca E Grinter. Evaluating human-robot interaction. *International Journal of Social Robotics*, 3(1):53–67, 2011.