博士論文

System Architecture for Task Execution Robots with Interpersonal Situation Scenting Capability
（対人状況察知機能を備えた作業実行ロボットシステムの構成法）

平成30年12月7日提出
指導教員 稲葉 雅幸 教授

東京大学大学院 情報理工学系研究科
知能機械情報学専攻 博士課程
笹渕 一宏

Contents

Abstract

To achieve a robotic system that is capable of both task execution and interaction behaviors in our society, the research proposes a system architecture under the theme of ”scenting” the human interaction willingness from non-verbal behaviors at the beginning of an interaction. The scenting capability is applicable to different task and social situations, and solves the problems of current social robots not having control over human-robot conversations, or problems where current task robots do not have a way of initiating a task.

In chapter 1, we point out the current failures of social robots, what is currently being requested in society, and argue that, for social robots to better benefit our society, social capability should be discussed with other robotic skills e.g. manipulation and navigation.

In chapter 2, we discuss the different focuses in interaction research and explain the novelty of understanding interaction at an initiating stage under different task context. Typical human-robot interaction focus on robot behaviors, dialogue research focus on speech behaviors, however, both target mainly on what happens during a conversation *after* an interaction has been initiated. Previous systems with interaction functions do not consider the interaction willingness of the person.

In chapter 3, to achieve the scenting capability, we propose a computational method using sequential data on human behavior in relation to robot behavior. We categorize the relationship between the interaction willingness of two agents, and explain that there are nine interpersonal situation patterns. Then, we formalize the situations into a hidden Markov based probabilistic graph model. The model is then trained using human-robot interaction recordings or on runtime. We evaluate the advantages of our method in scenting interaction willingness.

In chapter 4, we summarize the type of skills that are requested in business today and the skills that are requested in near-future settings such as in robot competitions. We explain the importance of heuristic-based manipulation, constraint simplifying hardware designs, task-finite scenarios, and software minimization. This is the basis of our task system for discussion in the other chapters.

In chapter 5, we explain the detailed implementation of our proposed system architecture. Based on general-purpose dialogue patterns, we explain that for action required interactions, we must consider a task scheduler that resolve constraints in the physical context. The task scheduler is combined with the scenting capability and takes into account when and whether the person is willing to interact, listen to the person’s dialogue purpose, schedule actions depending on the purpose, postpone interactions if a primary context such as *first-come-first-served* exists between the human and robot. We show task-interaction integration in settings such as a restaurant and show how our architecture technically applies to these settings.

In chapter 6, we go over various experiments using our system to understand its effect and evaluate our system from both a technical and social perspective, including appropriate and inappropriate behavior in task postponing, how people interact with robots to use its task skills, training a robot’s initial interaction behavior, and examples where the system is beneficial in a real setting such as a guiding robot.

In chapter 7, we summarize our achievements and provide future directions.

In summary, we have proposed a task execution system with an interpersonal situation scenting capability, which handles the different interaction situations during task execution, achieves the barebone robotic skills requested in our society, and allows the robot to self-supervise interaction behaviors at an initiating stage of an interaction. From experiments inside and outside the lab, we have evaluated our concept, the technical approach, and the potential effects of a task execution robot with social capabilities including non-verbal initiating behaviors.

1

Introduction

1.1 Gap between Robots in Society and Robots in HRI

As of 2018, robots with only communicative skills are seen as *failures*. Human-robot interaction (HRI) studies have questioned and answered, ”How would a social robot be beneficial?” [99, 36]. (Here, a social robot refers to a robot that has a role, interacts with a person, and has some communicative skills.) There have even been comparisons on how a robot is better than a tablet [67, 127]. Some researchers have and are still investigating on how a robot —if had communication intelligence—would be valuable to the society using the ”Wizard of Oz” (WoZ) paradigm [112] (a research scenario is set; a person is controlling the robot from a different room, making it look as if intelligent). Yet, when we step away from these scientific scenarios, we find that the *so-called* effects and advantages of social robots that have actuated movements are not worth the extra price for consumers.

Smart speakers have captured the skills wanted by consumers; the speaker interface have provided much of the skills with a lower price when compared to social robots. This was especially the case with the Jibo robot. A similar phenomenon has been observed with the Pepper robot [98] as well, although, unlike Jibo, which is a vendor-consumer product, Pepper is more of a vendor-enterprise-consumer product. Looking at posts on the web, many consumers find that the benefits of the robot interface could be altered with a tablet. From our own investigation, people from the enterprise find that the robot should be replaced with a more human-shaped appearance (no on-board tablets or alien shaped heads). Thus, only 15% of the companies using Pepper have decided to continue using Pepper, leaving the remaining 80% to abandon the robot. The market challenges of communication robots show that the end consumer’s expectation toward robots is more toward robotic skills (e.g. navigation, manipulation). Looking at succeeding robots in the market today (with a few exceptions in the medical and rehabilitation field [116, 25]), a robot has at least one of the below functionalities: capable of manipulating objects in a fixed environment (industrial arms), or navigate in a structured environment (security robots), or navigate and carry objects (e.g. Savioke Relay [23]). The former is a vendor-enterprise product while the later two are a vendor-enterprise-consumer product.

Since communication is not a clear capability, to better benefit end consumers or enterprise, interaction should be a capability on top of core robotic skills such as manipulation and navigation.

Socialness or interaction should be a capability to enhance these skills rather than interaction being the objective.

1.2 Unsolved Technical Problems

Adding interaction to a task execution system that executes robotic skills is not simple, and is much more than just adding extra interaction components. Existing systems only trigger the interaction skill or the other robotic skills (navigation and/or manipulation) one-by-one and in a sequential manner. For example, bring a drink after speech recognition. However, in a real setting, interaction skills and other skills may happen simultaneously. A robot could respond and listen to orders from a person while continuing its chores (assuming that the robot is an open-loop *look-and-move* vision-based control system [128]). This requires developing a system that handles interaction in parallel but also a system that is able to schedule a task in relation to the interaction context. In addition, most of the existing interaction systems wait and expect for a user input. In a real setting —especially when whether the robot is ready for an interaction is uncertain to the user, which is the situation when a robot is executing a different skill—the robot must have certain control of the interaction and take the initiative (Fig. 1.1 shows an example). This requires the robot to *scent* an interaction before any verbal user input is given. Since a robot moves around in a task setting, such a scenting capability must be implemented using on-board sensors, which is a constraint that is often ignored with in-waiting social robots.

1.3 Thesis Goals and Proposed Solution

The goal of this thesis is to solve the above problem of integrating interaction capabilities and systems that execute robotic skills. The novel and core idea is to look at interaction from a total setting where the other robot capabilities are active. We provide an architecture that computationally handles the different situations that happen when an interaction is triggered or about to trigger during a task by using probabilistic models and an integrated scheduling component; not just the verbal behaviors but the non-verbal behaviors that happen at the beginning of an interaction. By modeling the effect of non-verbal robot behaviors (including a *at task* state) and its relation to a

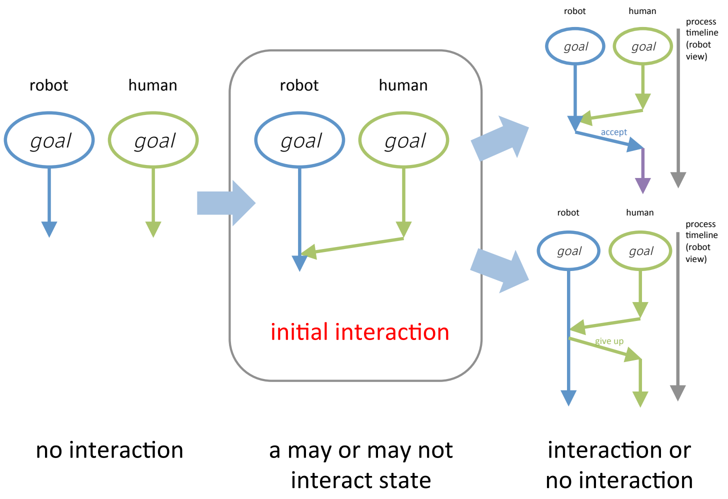


Fig 1.1: An example of an uncertain interaction beginning where the human and robot must go over an interaction initiating process. The human and robot are acting according to their own goal (whatever task they are doing), then, the human tries to change the robot's task through contact. The robot may accept the change or decline the change. (The discussion is interchangeable, and the agent that is contacting could be the robot instead of the human.)

human's willingness toward an interaction, we are able to express more precisely on the different interpersonal situation that is happening between an interacting human and a task-executing robot. Since we are using information on robot behavior, we have richer sequential information and therefore we are able to simplify the required sensor input on the human behaviors. This is an essential technique for embedding a scenting capability on a running robot with limited on-board sensors. The proposed architecture is essential for applying interaction capabilities on top of other robotic skills, but moreover, for a task robot to come out of factories and into our society. The timing of a person triggering a robot's task capability is arbitrary once outside the factory, and therefore, interaction will play a key role in handling and controlling task initiation. An overview of our proposed architecture is illustrated in Fig. 1.2 .

1.4 Prerequisite

Terminology- In this book, we will not be using the term *task* as opposed to the term *interaction*. A **task** could be defined as *a robot action or list of actions that are conducted for a certain purpose*.

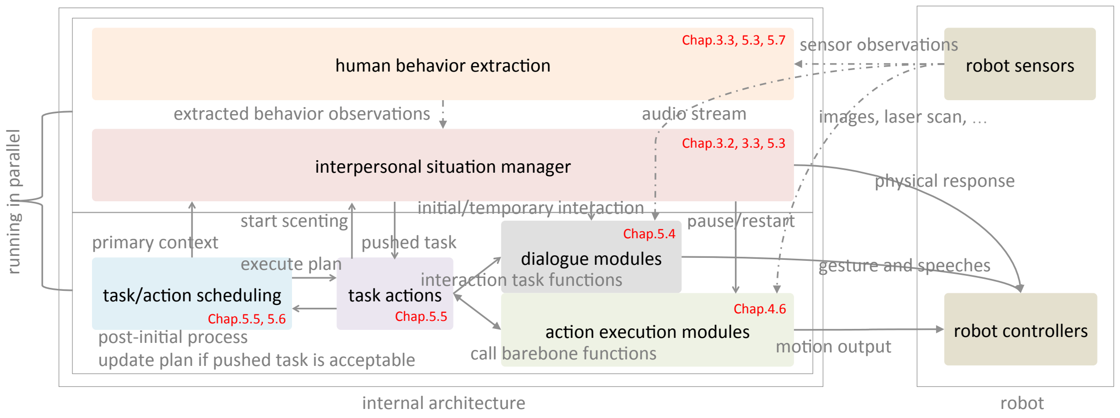


Fig 1.2: The proposed architecture handling the initiation and transition of an interaction task in parallel to task execution. Dotted lines indicate streaming information.

A handover, for example, could be a task but more specifically an interaction task.

We will refer to the interaction (especially non-verbal interactions such as responding) that happens between two agents (human and robot) before beginning an interaction task as an **initial interaction** (A more concrete definition will be provided in Chapter 3). Note that the term may sound similar to *initial contacts*. However, the word *initial contact* is used in the context where a contact is supposed to lead to an interaction. In contrast, as we will see later in the book, the term *initial interaction* also refers to situations where the contact between human and robot may be unintended.

Author's Standpoint Toward Robots- Human-robot interaction has a characteristic similar to human-human interaction but also typical to human-robot situations. For example, although many people will approach a robot and see a robot's head movement as a response (just like in a human-human interaction), people may also stare and walk around the robot before beginning an interaction (which would be impolite and unlikely with a human-human interaction). Therefore, in this book, we will assume that human-robot interaction is similar to human-human interaction at its core; yet, the human-robot interaction will also have robot-unique features that differ from human-human interaction. This means that, although we model human-robot interaction from a human-human interaction perspective, we will train the model from a real human-robot interaction dataset in order to achieve a better expression of the robot typical interaction.

Moreover, in this book, we will treat a robot as a device to assist people, but a device that must also act reasonable to the users. Here, reasonable could mean *a human-like manner* but more concretely, understandable and comfortable to the users. The unique part of a robot device is that, not only are they capable of physical assistance, but, since robots are able to give physical responses or physical actions, a robot could change its interface to a more optimal interaction style through training with runtime data. This is very different from tablets or speakers, which have a pre-programmed fixed interface.

1.5 Thesis Structure

The structure of the book is shown in Fig. 1.3 . Below we describe the details.

In Chapter 2, we go over the broad field of interaction and explain the novelty of our perspective toward interaction. We explain interaction, introduce a ten-dimensional graph explaining how previous works —including systems with interaction capability—have captured aspects of interaction, and the aspects we capture opposed to the other works.

In Chapter 3, we introduce our solution for modeling the scenting of interpersonal situations at an initial interaction stage and how to computationally estimate a person's interaction willingness in relation to the robot's behavior. We apply real human-robot interaction recordings to train our model and evaluate its effect through experiments held inside and outside the lab.

In Chapter 4, we explain the required robotic skill components especially those that are being requested in society. One of the problems with current interaction robots was the hardware quality and lack of robotic skills. We look at the recent tasks asked by the enterprise, and the requests we see in competitions. We summarize the required barebone robotic skills we should focus on for further discussion on interaction-task integration, as well as some of the solutions needed for executing skills under the more severe non-table environments a robot would face in our daily lives.

In Chapter 5, we introduce our solution for scheduling interaction tasks that consider both the interaction context and the robot's current task context. We propose the idea of task acceptance, integrate the interpersonal situation scenting model, and explain the details of the entire architecture. We evaluate the performance of our architecture on different person dialogue objectives. We

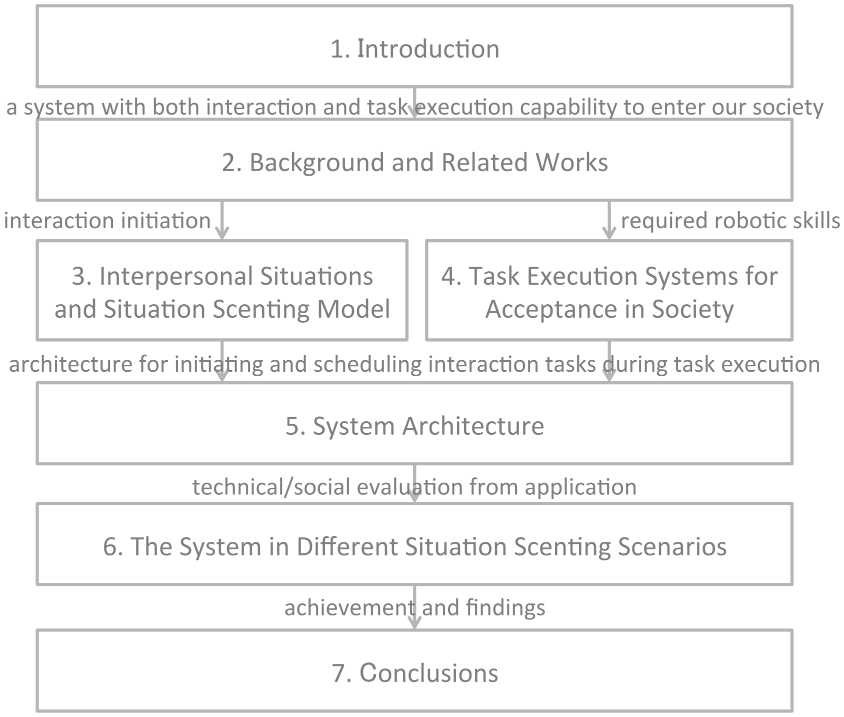

1. Introduction
a system with both interaction and task execution capability to enter our society
2. Background and Related Works
interaction initiation
required robotic skills
3. Interpersonal Situations and Situation Scenting Model
4. Task Execution Systems for Acceptance in Society
architecture for initiating and scheduling interaction tasks during task execution
5. System Architecture
technical/social evaluation from application
6. The System in Different Situation Scenting Scenarios
achievement and findings
7. Conclusions

Fig 1.3: Structure of thesis.

also show how our architecture handles a layered scenting scenario where a chance of interaction is scented during a different interaction task.

In Chapter 6, we evaluate our architecture from both a technical and social perspective to understand the actual type of scenarios in society where our architecture would be most valuable. As a technical evaluation, we show that the system can be used to change the robot's belief on a human's interaction willingness and provide better interaction behaviors throughout time. As a social evaluation, we look at how people try to interact with a robot under a task other than waiting for an interaction. We summarize the way people perceive task robots and the feedbacks we had on robot behavior during an interaction.

In Chapter 7, we conclude our findings and achievement.

2

Background and Related Works

2.1 Interaction from Theory in Different Fields

To understand the total setting of interaction in robots, and its relation or uniqueness compared to other interaction research, we must first understand what is an interaction. By definition, interaction is related to communication, and therefore, we begin by understanding what is communication.

2.1.1 Looking at Interaction from Communication and Dialogues

Robert T. Craig [24] has summarized different theories in communication, and has indicated seven communication theories. These theories look at communication as a flow of information, gap between viewpoints of subjects, influence on others, and/or to the more macro scale i.e. developing the functions of society. A common idea among some of these theories is that, communication is used to fill in the gaps and generate a consensus between individuals. While these theories provide understanding of communication in general, to capture the actual relation between the gaps, or the settings that produce the gaps, we must look more deeply into the type of speeches and actions that happen during a communication.

A known field that covers such area is the dialogue or dialogue act. Harry Bunt [16] has shown two general-purpose functions of a dialogue, which are information-transfer functions and action-discussion functions. Information-transfer functions can further be seen as information-seeking functions and information-providing functions. Action-discussion functions can further be seen as commissives and directives. There are many other ways in which researchers have explained dialogues and category of dialogues. Yet, these categories can be looked as a different angle of Bunt's explanation of information-transfer and action-discussion functions, and therefore, we will mainly refer to Bunt's theory.

Taking into account Bunt's viewpoint, we may say that the gaps being filled through dialogues are the information and agreement of actions. Bunt further looks at the information or agreement being updated, and explains the importance of looking at these information from the five-component context model which are: linguistic context, semantic context, cognitive context, physical/perceptual context, and social context. The linguistic context is the discourse plan dependent on the knowledge of the agent. The semantic context is the goals of each agent and the belief of goals on the other agent. The cognitive context refers to the cognitive process and production of utterances of each agent. The physical/perceptual context is the physical situation that the dialogue is being held. The social context is the context that generated the social acts such as greetings or apologies.

The discussions on information updates by Bunt provide us hints about interaction or the contexts behind that lead to a communication or dialogue in an interaction. The linguistic context and cognitive context are more about the characteristic or knowledge of an agent, while the semantic context, physical/perceptual context, and social context is more about the situation between the agents. The term *semantic* is tied to a linguistic meaning, but when we look at the context as relating to goals and beliefs of other agent's goals, we may say that the semantic context is more generally an *objective context* or an *objective frame*. The frame refers to what the agent is trying to do and the agent's belief on what the other agent is trying to do.

2.1.2 Looking at Interaction from Psychology and Objectives

Not surprisingly, we may look at the situational contexts leading to interactions also from the psychology field. After all, psychology is the field that helps us understand the relations between our internal states (objective frame) and actions (physical and social context).

Here we will introduce William T. Powers explanation on goals and goal-oriented behaviors from his theory on perceptual control (PCT) [104, 81]. Powers uses the term *living system* for agents, and explains a living system as anything that has the capability to behave. In his theory, he explains that living systems are goal-oriented and that their behaviors are controlled under perceptual input from the external environment. For example, a human driver takes an action of stop, accelerate, and handle, to fulfill his goal of parking. The driver selects his actions according to the difference between the goal (parked state) and current state (what the driver perceives).

From Powers' idea, it is apparent that an agent's action is determined by both the goal and physical environment, that is, the difference between the two. The goal corresponds to the objective frame, and the physical environment corresponds to the physical context. The two are a different concept, but are a common effect to the agent's behavior.

Extending Powers' idea, we may explain the social context from what happens when multiple systems try to fulfill their goal. Here is an example. A human worker at an office takes an action of wait and talk, to fulfill his goal of handing over a paper to his busy colleague. The worker selects his action according to the difference between the goal (interrupt colleague at a suitable timing) and current state (colleague is busy to be interrupted, or, colleague maybe interruptible). The worker may take additional actions such as ”cough” to catch his colleague’s attention and create a bigger cough depending on the colleague’s response (the perceptual feedback). In this example, the worker accomplishes his goal by disturbing his colleague’s goal i.e. his colleague’s work. PCT describes such disturbing situations of one’s goal as conflict. Using this idea from PCT, we may understand an interpersonal conflict as a situation where A) to accomplish the goal of one system, a disturbance to a goal of another system is essential, and B) therefore, at least one of the system desires to re-organize (terminology as used in PCT) the other system’s goal. In our above example, the worker must disturb his colleague’s work in order to handover his papers (A) and therefore he desires to re-organize his colleague’s goal from ”focusing on work” to ”talk with the worker” (B).

An agent’s action is determined by both the goal and the social situation between the agents. Similar to the relation between the objective frame and physical context, the relation between the objective frame and the social context are a common effect to the agent’s behavior. The difference between the physical context and the social context is that, a social context exists only under the situation with multiple agents.

2.1.3 Relation to HCI

It is also interesting to note that the aspects of interaction discussed so far also follows the human-computer interaction (CHI) ideas explained by Bonnie A Nardi [90]. Nardi explains that CHI can be represented as situated action models, activity theory, or distributed cognition. The situated action models can be seen as the relation between physical context and behaviors. The activity theory can be seen as the relations between objective frame, physical context, and behaviors. The distributed cognition can be seen as the relations between the objective frame, social context, and behaviors. Nardi explains that the activity theory and distributed cognition are similar and that the difference is whether the human and agent are seen as equal or not. We see that the two are indeed similar also from our discussion in that, both perspectives somehow connect the objective frame with behaviors. In addition, Nardi explains that the representation of activity theory is deeper than the situated action models. We see a similar structure from our discussion as the activity theory discusses an additional objective frame, whereas, the situated action model only discusses the relation between the physical context and behaviors.

2.1.4 Representing Interaction as a Ten Dimensional Graph

From our discussion, an interaction is what leads to an exchange in gap of information or agreement of actions. Such exchange is triggered from the objective, physical, and social context. While the dialogue act literature focuses mainly on dialogues or speech behaviors, it is apparent that, there is also non-verbal communication involved in an interaction such as gaze [59]. The five categories of gaze behavior discussed in [59] can be explained from a similar discussion of objective, physical, and social context.

Summing up our discussion and different perspectives from different fields, we may summarize the total view of an interaction as in Fig. 2.1 . There are two agents (here we focus on machine and human interaction) exchanging information and agreement of actions through verbal and nonverbal behaviors, which are generated from the objective, physical, and social context. Note that the social context only happens when there are multiple agents. Likewise, a verbal behavior happens as an exchange between multiple agents. Therefore, these two aspects of interaction meet together as shown in the figure. In contrast, the objective and physical context, as well as a nonverbal behavior may happen within a single agent.

An interaction research captures the different aspects of interaction with different focuses, and the different focus weights on each aspect. In the next sections, we will look at the different focuses of interaction, and compare how they are similar or different from our approach of interaction.

2.2 Research in Interaction

2.2.1 Typical HRI

In the 2018 HRI conference, there were four main categories that were announced by the conference organizers. We will use slightly different terms, but the categories were: HRI user studies, technical HRI, HRI robot design, and HRI analysis. HRI user studies include works such as, investigating robot applicable scenarios and finding their effect [142, 36], applying human-to-human

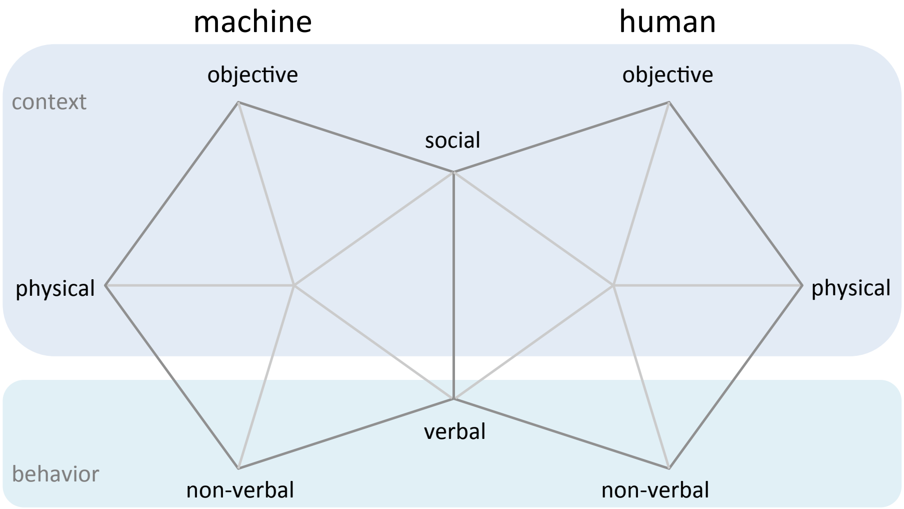
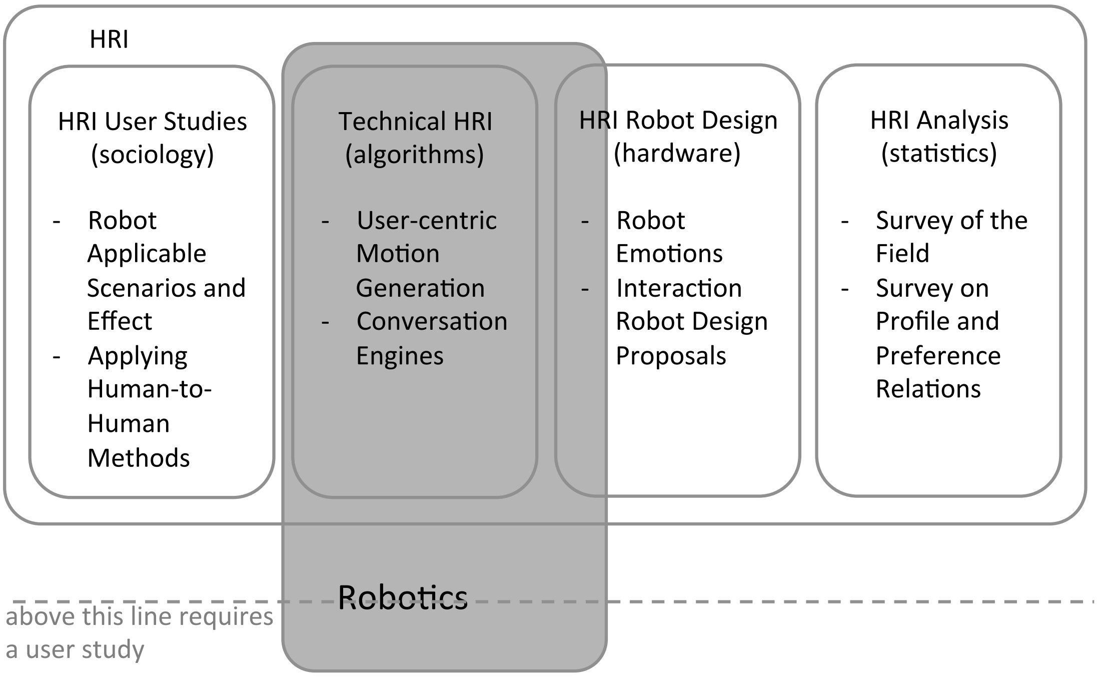


Fig 2.1: Ten-dimensional interaction focus graph. A summarized representation of aspects shaping an interaction. The aspects come from a summary on interaction-related theories and perspectives of different fields.


HRI

HRI User Studies (sociology)
- Robot Applicable Scenarios and Effect
- Applying Human-to-Human Methods

Technical HRI (algorithms)
- User-centric Motion Generation
- Conversation Engines

HRI Robot Design (hardware)
- Robot Emotions
- Interaction Robot Design Proposals

HRI Analysis (statistics)
- Survey of the Field
- Survey on Profile and Preference Relations

Robotics

above this line requires a user study

Fig 2.2: An overview of the HRI field and its relation to the robotics field.

interaction methods to human-to-robot interaction and evaluating validity [3, 52]. Technical HRI include works such as designing user-centric motion algorithms and comparison between motions using the algorithm versus motions not using the algorithm [29, 88, 20], conversation engines and evaluation on whether the engine provides better results to specific problems [18]. Robot design include works such as developing robot emotions and evaluating whether users perceived emotions correctly [77], but also proposals of interaction robot hardware designs and evaluate satisfactory of the design [39]. HRI analysis include works such as survey of the field [63], survey on user profile and its relation towards acceptance of robots [71].

Despite the category, an HRI study requires some form of user study in the end to evaluate the study proposal. This is what makes HRI unique from other themes in robotics. HRI is not always solving a new problem but find scientific facts about user acceptance toward robots. In contrast, robotics require solving new problems, different approaches to problems, and summarize findings in a way that will benefit the field. Due to these different goals of the field, robotics and HRI only overlap partially. The technical HRI category include both a robotics and HRI aspect. The robot design category usually use simple reproducible hardware designs, and therefore, may not be much of a benefit to robotics.

The other categories focus on sociology or statistics and are more towards science. HRI user studies focus and weight the social context. HRI robot design is more of a design research rather than an interaction research (it evaluates appearance readability without no or little context), therefore, does not fit in the ten-dimensional graph. HRI analysis is more of a research on interaction research, and also does not fit in the graph.

Given these different characteristics of each category, our approach toward the total setting falls closely to the technical HRI. We develop an algorithm to autonomously scent interpersonal situations, and we go over user studies to evaluate its performance. However, as we will see in the next section, our approach differs from other technical HRI approaches in that, the focus includes the social context of what happens before, or at the beginning of an interaction. We summarize the HRI categories and its relation to robotics in Fig. 2.2 .

2.2.2 Technical HRI Approaches with Physical Context

Popular themes in interaction with physical task context include human-robot manipulation and human-robot navigation. For example, human-robot manipulation may include human-robot handover [80, 135, 85, 1, 53], human-robot cooperation [86, 30, 20], human readable (including expression of incapability) manipulation motion designs [31, 131, 69], or human-robot teaching [42, 7]. Human-robot navigation may include human-robot avoidance [107], robot-to-human approaching [26, 122], or even combination of handover and navigation [126].

In most manipulation related scenarios, the human and robot are already in interaction with a shared goal or role (for example, human teaching and robot learning). [135] discusses part of before handover phase and its relation in the handover context. Yet, the situation assumes that the person accepts the handover. Although the themes of prior work include the physical context, the social context is somewhat weak. There is no social background or verbal communication that influences the interaction between the human and robot.

The navigation scenarios, on the other hand, may seem more like a discussion on interaction initiation. However, the focus of these scenarios is mainly on *appropriate* approaching behaviors. Such navigation problems focus on the physical context (e.g. distance or formation between the human and robot) and robot's action in response to that context. Again, there is no social background or verbal communication that influences the interaction. It does not question whether the person will respond when the robot approaches.

In one of our experiments from Chapter 3, we see that sometimes, people are not aware of the robot's approach. In Chapter 6, we even see situations where a person passes by the robot by ignoring its approach. In some long-term experiments [138], we see that in fact, people are not always interested in the robot and people do not notice some of the robot's signals. In this book, we assume that on top of the robot's willingness, the interpersonal situation is dependent on the interaction willingness of the human.

Most of the approaches above solve a mathematical optimization problem with human-in-the-loop cost functions, and then evaluate whether the cost functions were appropriate through surveys. The cost is usually between user experience and task performance. [86] and [53] both discuss that user experience and task performance do not always correlate. Whether we should maximize user experience or maximize task performance is a recent question even in autonomous car driving behaviors [117].

The initial interaction problem in the total setting is somewhat different to behavior optimization. As we see in the other chapters, there are situations where the robot is not able to interact, not because it is trying to maximize performance, but because its motions are under some task constraint and must resolve the constraint before beginning any interaction. Or even, the person may ask the robot to come to his office later, or bring a drink once the robot has finished its current task. As in this example, the person may tell the robot how he or she wants it to behave per situation, rather than having the robot to autonomously optimize what it thinks is the correct cost function.

2.2.3 Engaging and Scenting

Some topics focus more on the social intent of a person. Here we will discuss approaches to intent in relation to robot behavior. One of the questions in this book is *how will a robot know if a person really wants to interact?* To this particular topic of *intent*, terms such as ”engagement” or ”attention” [70] have been used. Engagement refers to both before and during interaction interest of a person toward a machine. It has been an important topic of discussion [129, 4]. [13, 14] detected engagement using visual information of human face and location. Method and evaluation on intent understanding of different situations at a shopping mall has been conducted in [65]. [73] discusses with-me-ness of an interaction. However, the term ”engagement” focuses on whether a *person* is engaged with the robot or not. How about whether the *robot* is engaged with the person or not? Most researches discussing engagement, do not consider the robot's objective or physical context, and is mostly focused only on the human-side of interactions. We must step one step further into engagement, where a robot may be under a different desire than to interact.

A robot may not always be ready to *detect* engagement. It might be focusing on a task, and a person may be out of the robot's view. Instead, a robot may *scent* that a person who may be engaged is coming close by. (The robot could use its base lasers for this particular situation) **Scenting** is different from *detection* in that, what is being observed is a chance of truth rather than the most likely truth. What is *scent* could be very uncertain but provide hints to how the robot should react in the next time frame. For example, if a robot *scents* that a person is engaged, it could try to look back at the person. By doing so, the robot could then *detect* whether the person *was* actually engaged. One of the goals of this book is to understand this procedure of scenting, responding, and detecting; and how this procedure could be represented on an autonomous robot. This covers a wider scenario than engagement, which only discuss about detection.

Some studies have discussed situations where the human initiates an interaction. [154] has analyzed initial human actions in elderly day cares. However, technical implementation for detecting these initial human actions has not been achieved on a fully autonomous robot. A more generalized discussion on interaction situations is required for technical implementations; especially for a re-usable system.

Several researches have discussed situations where the human may or may not interact with the robot, and have focused on how to initiate an interaction. An approach using a participation zone concept was discussed in [125]. Human-to-human interaction strategies for handling fliers were discussed in [126]. These discussions provide methods for initiating an interaction in situations where the robot approaches the human. These researches provide scenario-based interacting methods for concrete situations. Meanwhile, the focus of this book is more on understanding situation patterns in general. The role (objective) of the robot is fixed for the usual robot-to-human scenario, and the robot is often the initiator of the interaction. In contrast, the robot could be a listener or an initiator in our approach of the total setting, and the role could change from estimations in the initial interaction. Therefore, we focus more on the objective frame when compared to previous research.

2.2.4 Comparison to Dialogue Research

Robots focus mainly on non-verbal behavior, therefore are quite different from the dialogue research field. Yet, robots also go through verbal communication after initiating an interaction. The problem of turn taking during an interaction has common interests in both fields. The fields are also similar in that, statistical models are used to automate and understand interaction. Here we will go over few of the recent approaches in dialogue, and how they are similar or different to the total setting approach of human-robot interaction.

[132] focus on a temporal model for estimating who has floor in a turn-taking dialogue, and reports that for their scenario, data must be trained on human-machine interaction rather than human-human interaction. Interestingly, the approach is similar to our approach in estimating will-

ingness, in that, the model takes in temporal binary data for estimation, uses run time probabilistic scores, and trains on human-machine interaction. The difference is that, instead of willingness, the dialogue model tries to detect a hold or a switch in conversation. However, the more important difference is that, while a dialogue tries to detect a change in intent, in our scenario with robots, we try to change the behavior of the person (get a response from a person who was not willing) or change the behavior of the robot (stop trying to initiate an interaction if it finds out the person was not willing) from the scented intent. In our case, there are situations where no one may have floor of the interaction (it has not yet been initiated). This leads to considering different and more types of estimation states but also different discussions on how to integrate model estimations to the entire system. How to map the estimation outcomes and machine behavior is not as direct as the turn-taking scenario.

In terms of objective differences, [93] focuses on a dialogue version of the total setting where two agents have different goals, and try to negotiate through a dialogue. The authors propose three negotiation patterns: individualistic, cooperative, and competitive. This is similar to some of our interpersonal situation patterns. The individualistic may be seen as an agreement where both agents fulfill their own goal. The cooperative may be seen as an agreement where either agent is following the goal (cooperating) with the other agent. The competitive may be seen as a conflict where both agents are trying to push their goal over the other agent's goal. However, we also realize that the setting in [93] is actually scoped in terms of possible interpersonal situations. The goal differences discussed in their paper refer to the difference of the same topic: two agents trying to get fruits from one market. The setting starts from where both agents are aware of each other, and try to predict the other agent's goal within the topic. We find that there are more possible interpersonal situations if we step one step back, consider the total setting from before the two agents are aware of each other. In fact, the speech behavior of concern is quite different if we take this step.

We notice that it is common in the dialogue research field to consider the objective of both agents. This is quite different from HRI where the objectives of agents are often ignored, or the objective of the human is the main concern. Yet, dialogue is what happens after an interaction is initialized. To this extent, our focus of robots is slightly different and must consider a broader range of situations including unintended interactions and integration under different context of non-willing situations.

2.2.5 Approaches to Reading Human Intent

The problem of understanding human actions in the computer vision field and the robotic field is slightly different. Human action recognition in computer vision is a problem of detecting human actions from images or videos and are evaluated through datasets [101]. Some other works focus on action segmentation to extract key frames [58]. Others try to capture group conversation behaviors such as detecting F-formation [124].

In contrast to the above problems, human actions in the robotic field are tied more toward understanding actions in relation to daily life objects, forward predicting actions on real time, or understanding under first person interaction. (Although, some computer vision approaches do consider understanding human-object pose relations [43, 56, 156].) To understand human behavior, anticipation gesture based reasoning approaches have successfully represented human actions in daily life task situations [68]. Other approaches detect human intention from grasp inference [133], or from object reaching inference [108]. [37] proposes that beyond intent, there is the human's rationale that is estimated by: understanding activity (e.g. pour cereal), understanding motion (e.g. pick, place), and understanding intent (e.g. domain knowledge such as object presence). Some works target specifically on human interruptibility to decide the timing for robot-to-human interaction [8]. These approaches each capture understanding of human behavior from different perspectives in robotics but without the context of the robot. Even approaches that do include a robot in the scene, the robot just sits and waits, and there is no specific context [153, 84].

Our interest is not only to capture human behavior, but also understand human behavior in relation to robot behavior. [9] have studied on whether a robot should take the initiative for a collaborative task. We question about unintended interactions that initiatives may cause. [152] have proposed robot centric understanding of human intent under a limited set of interaction actions. Regarding machine-human interruption, interrupting computer behaviors has been discussed in the CHI field [82]. However, the CHI approaches largely rely on robot speech behavior, and do not consider physical actions of the machine. Likewise, the dialogue field also discusses interruption and initiation of an interaction using speech behavior [92]. [92] points out that preferred speech behavior is different among people in an interruption setting. However, the results could be different if there is physical activity involved (e.g. the approaching activity is a continuous interruption while a speech is a sudden interruption). Beside visual approaches, there are text-based approaches to understand human intent [35, 46]. These techniques are helpful for understanding human intent verbally after an interaction has begun, but since our discussion is more on interaction beginnings, these techniques are slightly out of scope.

There are many approaches to understanding human behavior and intent. It is possible to use these approaches to understand human intent for initiating interactions. However, reading the intent does not answer the process that occurs when initiating an interaction, and some approaches are even tied to scenarios where there must be some specified object the human is using. We provide a more general framework that may be applied with or without object context.

2.3 Robot System Architectures

In this section, we will compare the total setting from a system perspective.

2.3.1 Different Task Execution System Designs

Various robot system architecture designs have been proposed over the years. While designs differ depending on expected level of autonomy, most systems are composed with general robotic components and a sequence-managing layer. How components are divided and what components are integrated largely depend on what the system finds as important, or what is the system's objective.

In terms of autonomy, [49] discusses five level of automation in relation to the problem structure the system will solve. [49] describes fully modeled problems solved with pre-defined repeating actions as level 0, problems requiring check after action as level 1, problems requiring sensing before action as level 2, problems requiring feedback control as level 3, and problems requiring prediction as level 4. Below, we will see that most near-future manipulation problems such as the ones from competitions fall into the level 2 automation requiring a *scene-plan-act* system. A manipulation-centered system may focus on a one-way connection between vision, manipulation planning, and servoing [48, 134, 47]. However, other systems may connect components in various ways. For example, the PR2 system [12] has a vision, manipulation, and navigation component. These components are not connected in one single way but are triggered at different timings depending on the application. Such applications may be written using state machines such as SMACH [11]. Whether represented by state machines or not, the components are often triggered from task descriptions. [95] describe three types of descriptions: without failure recovery, with local recovery, and with local and global recovery. In their paper, strategies such as fail and then abort (switch plans) fall into the first type of task description, while repeating actions is referred as local recovery. Failure recovery may also be handled on the component side rather than at task description or planning level. For example, recovery behaviors are seen in the ROS navigation stack. These type of component based error handling are more like strategies rather than logical plans. Whether error handling should be done on the component side or on the task description side may be determined by whether the plan is a logical recovery or a human insight. In cases such as rapid prototyped systems or user-designed applications, components are combined without recovery behavior [54]. In other systems like ASIMO, components are triggered from events rather than from task descriptions [118].

Some systems such as teleoperation systems do not have task descriptions but instead a user interface [158, 57, 6, 47]. How components are divided differ from automated systems. For example, [158] have divided components so that multiple operators may operate the robot in parallel. The components they describe are trajectory design, execution managing, and perceptual. An automated robot would plan and then act, however, in teleoperated systems as the one mentioned above, an operator prepares the next plan while another operator tries to fix the current plan in action.

For systems that expect HRI, there are some unique components related to HRI or even a semantic knowledge component [60]. We also see such HRI components within architectures proposed by [118] and [136]. These architectures are mostly event driven parallel running behavior modules. Another important part of these architectures, are that, each component accesses to a knowledge database related to the task. The system either collects knowledge such as a human face, or some predefined knowledge such as a building map is applied. Speech recognition, text to speech, face recognition, and gesture recognition are common type of functions that compose the HRI component. Surprisingly, the HRI component does not handle how to *initiate* an interaction but only the basic functions that would be required for a person to talk with a robot *during* an interaction. Perhaps such limitation comes, as some of these systems were developed toward solving the RoboCup@Home [149] problem. In the competitions, it is assumed that the robot is always willing to begin an interaction, and in some situation, even waits for a person to come by. In addition, the robot somehow has control of the world, people do not bother the robot, nor do the people ignore the robot when questioned. More non-integrated interaction robot systems or middleware based systems such as the NAOqi [103] focus on managing more low-level components such as gestures and speech in both sequential and parallel ways. More complicated interaction specialized robots such as ERICA consider complex decisions for speech dialogues [83]. They use multiple channels such as utterance and event (e.g. long term silence) to trigger a dialogue, which considers part of the initiation problem especially those that are tied to dialogue events.

Most systems explained above handle a structured or semi-structured environment. However, when we look toward problems in other fields, we see that some components must dynamically adapt to the environment. For example, an agricultural robot reported by [87] points out that a GPS approach cannot be used reliably at all times especially when the sorghum crop is tall, therefore, requiring different combination of multiple sensors for navigation depending on the environmental situation. Such component design would be too much for an indoor scenario but would be essential for an outdoor scenario.

The systems introduced so far do not use any planning or only use some form of deterministic planning. Meanwhile, probabilistic action decision has enabled information gathering and robots to conduct tasks under uncertain environments [61]. [44] proposes a more advanced system with a three-layer architecture and switches between deterministic planning and decision-theoretic planning under uncertainty. In their three-layer structure, the semantic component (relational map) is placed in the belief layer between the task description (deliberative layer) and components (competence layer). Although the deliberative layer is more complicated, failures are reasoned, and the plans are re-planned; the layer will decide a set of actions the robot should conduct from a given goal. Therefore, such layers still manage a sequence and are connected to general robotic components, however, the difference is that actions triggered from each component is fed back to the manager rather than just being stored as semantic knowledge for the components to access.

There are many different ways systems are designed. Teleoperation based systems lead to user interface connected components. Automated systems have managing layers instead of user interfaces. Competition based or problem based systems lead to strategic plans rather than logical plans. Specific manipulation based systems lead to one-way sequential component connections. General-purpose systems lead to multiple connections between components. HRI involved systems lead to having semantic components. Systems that tackle practical problems lead to more complex component design. Systems that tackle high-level uncertainty lead to more complex management, logical plans, as well as more complicated connections to the semantic components.

2.3.2 Systems from Evaluation Benchmarks vs. Systems Asked by Society

When we look at how the above systems were developed, some systems were developed through benchmarks such as RoboCup. Some were developed from predefined problem settings such as teleoperation. Some were developed as an experimental platform to tackle more uncertain problems.

The problem with benchmarks is that they fall into specific evaluations such as grasping [145] and not the robot functionality as a whole (e.g. manipulating in the clutter). In addition, functional tests try to cover tasks not in depth but rather in variety [105]. These variety do not answer whether these are the functions required by society or whether these are only technical tests.

To this extent, competitions (sponsored by the government or enterprise) have better settings as they are truly a simplified version of a real problem in society, and the developed systems provide likely solutions to the problem. The winning team of the DRC developed a hardware that switches between walking and driving with standing and kneeling postures [94]. The winning team of the APC in 2016 used a hybrid approach of suction and gripping [48]. However, we must also be aware that solutions for one competition may only be beneficial for the specific problem, and for business, a generalized solution helps in reducing cost as they target more domains. Suction with large compressors may relax a specific problem but is not suitable for passing through doors or moving on terrains. To avoid such local maximum and to achieve more scalable solutions, we propose at looking through several competitions instead of one. This not only helps us avoid local maximum but understand the required technologies that are in common among different tasks and the technologies that are specific to the problem.

Yet, competitions are still a simplified version of a future challenge. Before the robot is able to achieve the real challenge, it will take a few years. For new technologies to get accepted, the technology must gradually enter our society. This helps in avoiding over expectations, but also help find out critical problems at an early stage. Therefore, we must keep in mind what is currently possible, what is expected in the future, and what stage of the technology lies in between. In this sense, experimental platforms lie too far in the future, making it difficult to capture the steps in which the technology will advance. With systems developed for experimental research purposes, we might be going in a direction away from what will be accepted in society.

2.3.3 Mapping Complexity of Systems

We may summarize the different type of system architectures as in Fig. 2.3 . The systems on the left side are more *automated* systems. The systems on the right side are more *autonomous* systems. The systems on the top handle simpler situations, the systems on the bottom handle more complex situations. The components depend on the scope of the system including its scope toward task variety and depth of the task.

From the figure, we see that the total setting tries to tackle more complex situations but there is room to enhance level of autonomy (we have much room for discussion on whether we actually want such autonomy).

Complex systems may be more experimental rather than practical at the current stage. However, some systems have to be complex to be practical. Especially in a setting where an arbitrary customer comes to use the robot, the robot must first understand that the customer wants the robot's response and then handle the customer. The timing that a customer comes and interacts with the robot is arbitrary, and the robot may not be ready for an interaction if doing a task. Therefore, understanding interaction decisions and management, is an essential part of a system for robots entering our society. We must challenge the complexity, and the first step is to build an automated system handling interactions during a task.

2.4 Approach Background

This section provides some of the backgrounds on our method for understanding interpersonal situations and producing robot behaviors. The section is supplementary for understanding why we

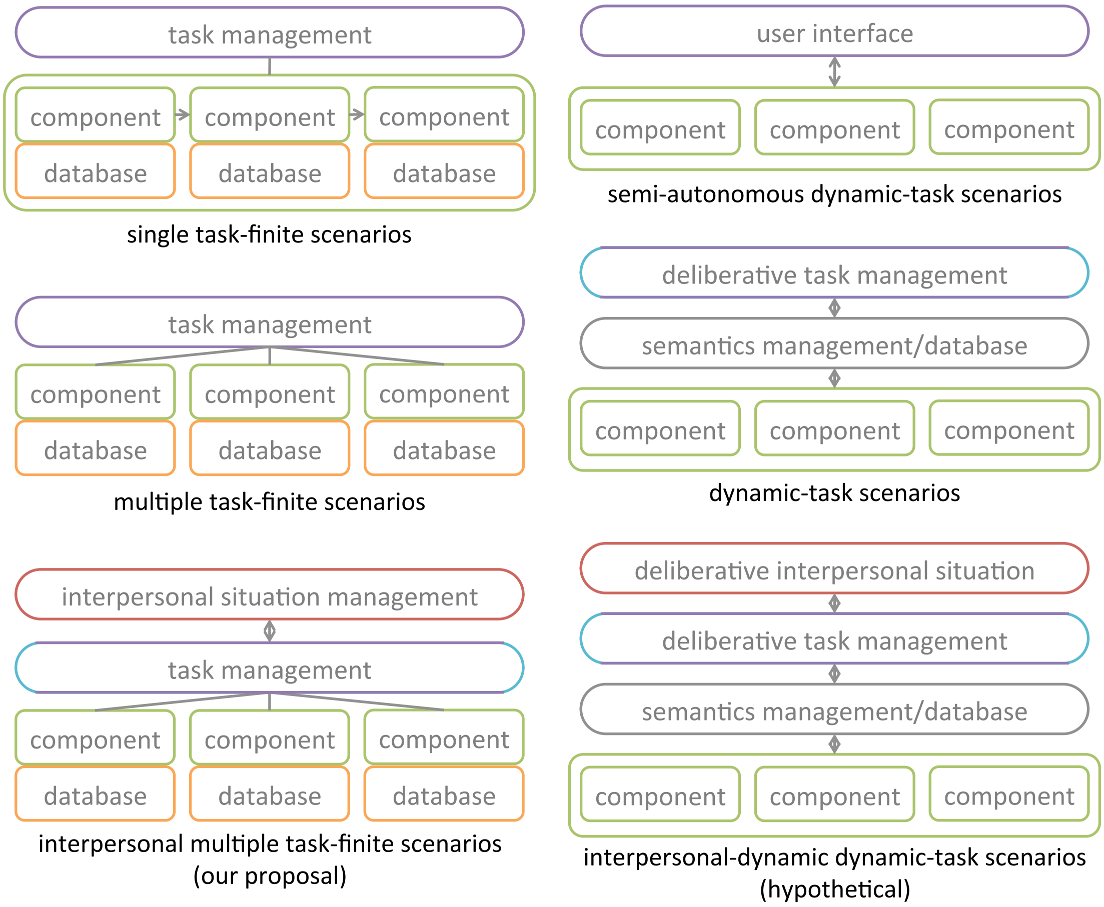

task management
component
component
component
database
database
database
single task-finite scenarios
user interface
component
component
component
semi-autonomous dynamic-task scenarios
task management
component
component
component
database
database
database
multiple task-finite scenarios
deliberative task management
semantics management/database
component
component
component
dynamic-task scenarios
interpersonal situation management
task management
component
component
component
database
database
database
interpersonal multiple task-finite scenarios
(our proposal)
deliberative interpersonal situation
deliberative task management
semantics management/database
component
component
component
interpersonal-dynamic dynamic-task scenarios
(hypothetical)

Fig 2.3: A way of looking at the different system architectures. Colors match the component colors in Fig.1.2. Note, task action/scheduling and task actions are combined as a task manager component for easier comparison.

choose our assumptions and methods in this book.

2.4.1 Human-human Assumption in Human-machine Interaction

In our method, robots will produce looking behaviors or speech behaviors, and act accordingly to the outputs of our situation engine model. The assumption that underlie in these behaviors, are that, robots have their own goals, a robot will look toward which ever direction that is related to their goal, and a robot will try to avoid conflicted interaction states.

This kind of behavior comes from human behavior models. One may question why we apply human behaviors to robots. HRI studies have shown that applying human-like behaviors lead to more natural and preferred interaction behaviors. For example, [3] reports human-to-human gaze aversion are effective when applied to human-robot conversations. [109] point out the effect of non-linguistic utterances by robots to enhance contextual meanings. In addition, adapting human-like behaviors help robots to look intellectual. It has been reported by [79] that without intellectual behavior, robots may look uncanny. Although the above was studied on robot androids, even for robots like the PR2, there is data that supports the relation between behavior preference and intellectual behaviors, that is, anticipation behaviors [140]. In this example, Disney principles [143, 102] were applied instead of human-likeness. This shows that human-like behaviors are not the only path to being intellectual, but rather, human understandable behaviors are the key to acceptable robot behaviors. Some also report the importance of exaggerated motions for interaction [38]. In one of our previous work, we introduce that robot specific-mechanisms such as an actuated hair may be perceived as context emphasizing motions and are understandable [120]. Yet, some behaviors and its relation to intellectualness are not always obvious, and may depend on who is interacting. [5] points out that robot-to-human touching behaviors give an impression that the robot is more capable of a job, however results differ depending on the subject's gender.

We have explained in the previous chapter that we assume that human-robot interaction will be slightly different from human-human interaction but will have similarities to human-human interaction at its core. The above works in the HRI field have shown some supportive data on this assumption. A person will expect some level of human-like intellectualness for better HRI, however, the expected intellectualness is more about understandable behaviors rather than being exactly human.

2.4.2 Usage of Sequential Data for Situation Understanding

In this book, we use a Hidden Markov Model (HMM) based model structure for understanding human-robot interpersonal situations. Hidden Markov Model although proposed in the 1960s have been seen effective in many applications that use sequential inputs [32]. HMMs are generative models and unlike discriminative models, they capture the joint probability instead of directly finding the conditional probability between input and outputs. In some comparisons between generative models and discriminative models, it has been seen that generative models reach its asymptotic error faster with smaller data [91].

Advances in machine learning have introduced other ideas for handling sequential data. Discriminative models such as Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) [51] are becoming applicable to practical applications such as handwriting [41] and speech recognition [40]. In the speech recognition literature, LSTMs capture longer context than RNNs, and RNNs have richer expressions than HMMs for large state-space [40]. However, the comparison of such methods are not as simple, and depending on required learning time or data amount in the problem scope, HMMs tend to be more appropriate than LSTMs [100]. Other ideas include the idea of using attention mechanisms that are computation efficient in longer sequences [147]. Attention mechanisms have been used in combination with RNNs and LSTMs in the machine translation literature [78] or speech recognition [22], but more recent work use *only* attention mechanisms (namely Transformer) and also handle intra-input and intra-output relations [147]. Whether an attention mechanism is necessary depend on the problem. For example, [113] reports N-gram RNN models works as well as sophisticated attention integrated models for dataset problems that only require local context such as predicting verbs and prepositions [19] in the Children Book Test [50].

Whether to use a generative model or discriminative model, whether to use attention mechanisms or no, depends on the number of data, required efficiency, and whether understanding of sequences require a global look of the sequence. In terms of interaction, the interaction patterns may differ among situations and we do not have enough data that benefit from using the more sophisticated models. In addition, the larger data the better may not apply for an interaction. In the later chapters, we show that interpersonal situations reach an asymptotic probability at around forty-to-sixty interaction datasets. Instead of being able to handle large datasets, it might be better to quickly adapt to new data and new interaction patterns the robot faces. Moreover, typical HRI have shown the importance of statistically understanding interaction between the human and robot. It is important to technically achieve a short hand answer like discriminative models, but it is also as important to understand what underlies in the answers like the generative models.

Let us step a little bit further into our discussion of handling sequential data. The examples so far are hand writing recognition, speech recognition, and machine translation, which are all relevant to context understanding. However, interaction not only requires the understanding of context, but gaining information from the visual is also an important part of interaction. Some approaches treat sequential problems similar to computer vision problems by mapping input and output as a two-dimensional grid [33]. In this approach, contextual attention is expressed similarly to a visual attention. Other problems such as image captioning are by nature a combination of a context and visual understanding problem. In such problems, an idea such as using both bottom-up and top-down attention has been introduced [2]. [2] handles the contextual problem with the softmax top-down attention (the ones we have seen so far), and the visual problem with the ResNet-101 [45] based Faster-RCNN [111] trained bottom-up attentions. In vision only problems, [74] uses a bottom-up and top-down structure to capture smaller objects in an image.

The model we use for understanding interpersonal situation scenting is somewhat similar to the top-down bottom-up attention integration explained above. In a human to robot interaction, a bottom-up attention of someone who might be willing to interact is passed to a top-down attention of who to (or who not to) interact. In a robot to human interaction, the robot may try to interact through a top-down attention. In our system, the bottom-up and top-down mechanism is divided as the required data size for the two attentions differs. The top-down mechanism requires only about forty-to-sixty interaction recordings. The bottom-up mechanism responds to more generalized features and will require more data. Between the two mechanisms, a binary structured data is passed. One might question why not directly feed images to process sequences. The one limitation of directly feeding images is that we are tied to a specific sensor. A robot has multiple sensors. When a person is out of view, it may scent a person coming by using its base laser. A robot may have multiple sensors and may *scent* with one sensor, but *detect* with a different sensor. Direct sensor feeding may limit the capabilities of our system. Yet, if a higher context understanding (such as person is not understanding the robot's behavior or the person is not understanding the spoken language of the robot) were required for the interaction, then, some kind of direct sensor feeding (perhaps integrated with semantic knowledge) might benefit. In our experiments, 70% of the time, we did not require understanding of such higher context to move from task to interaction. Moreover, we may just want to train on the top-down attention but reuse the bottom-up mechanism. Transfer learning or fine-tuning [157] have similar concepts of reusing the general feature extraction and train on top of what is already learnt.

2.5 Conclusion

The total setting perspective is balanced between the social, physical, and objective context. This is different from usual works in HRI, where the perspective is either more towards social (HRI user studies), or more towards physical (technical HRI). Although areas in technical HRI such as cooperation are similar to our focus in that, the robot tries to understand a person's intent, and behaviors are changed according to intent estimations, the main context is physical, and the objectives between human and robot are aligned. In contrast, our focus considers situations where a human may not want to interact and therefore, must scent the social context on top of a physical context (e.g. whether a robot is in work, or the distance between the human and robot).

In terms of social context understanding, our focus overlaps with the field of engagement. However, in usual engagement, the robot's behavior is not considered for estimation, and approaches try to understand a person's intent mainly from visual observation and in a robot waiting setting. We step one step ahead of engagement and generalize different interpersonal situations including robot-to-human interaction. The difference of our approach and other robot-to-human interaction research is that, we consider situations where a robot could become a listener, and the initiator of an interaction may alter depending on situation.

The total setting looks at what happens before an interaction, which may potentially connect to the dialogue field once an interaction has been initiated. The difference between dialogue research and our robot perspective of an interaction is that, the objective of the human and agent could be completely different and related more towards physical context (whether a robot looks, or whether a robot has shown a physical response of looking back).

Although there are several overlaps between our approach and other research fields, none of them have covered interaction in a balanced way as we do. Most research focus on specific context, and some are more towards social, some are more towards physical, some do not consider the objective frame, some do not take into account the effects of robot behavior. In order for a robot to interact in different task situations, we must not look at one, but multiple situations and how they relate to other situations.

3

Interpersonal Situations and the Situation Scenting Model

3.1 Introduction

In the total setting, a robot may or may not begin an interaction. A decisional process is handled from the interpersonal situation. The chapter will introduce the **willingness state** to explain the possible interpersonal situations related to decisions. The core idea behind is that, the different context and number of interpersonal situations can be simplified to nine patterns if used the willingness state. We will discuss the total setting problem from both an HRI (human-robot interaction) and technical perspective. The HRI perspective allows us to find the underlying characteristics behind interaction situations. For example, will there be a difference in human behavior when it is the situation where the robot desires to interact, compared with when it is the person who desires to interact? Which interaction situations are rare, and what happens in these rare situations? Will a robot's behavior influence a person's interaction desire? Sub questions to this question would be: Will a person give up interacting if a robot keeps on ignoring him (her)? Will a person more likely respond to a robot that tries to catch attention by talking, or more likely respond to a robot that tries to wait politely without talking? The technical perspective on the other hand, allows us to understand how to computationally model the characteristic findings into an autonomous procedure.

As a result, we achieve a **situation scenting model** that includes underlying characteristics of various interaction situations and context. The chapter will also point out the rare interaction situations such as **unintended interactions** and why they occur in real human-robot interaction scenarios. At the end of the chapter, we show that different situations occur depending on human behavior, and that a robot can understand these situation differences using the scenting model. We show that the robot can then behave according to each situation using the proposed computational model. (This chapter has been written based on our dual track oral presentation conference paper [119]. In this book we add details that were excluded due to page limitations and have reformatted the structure in order to clarify the relations with the other chapters. We have also added extra discussion to avoid possible confusion on the abstracted nine interaction situations, and how they relate to real interaction scenarios.)

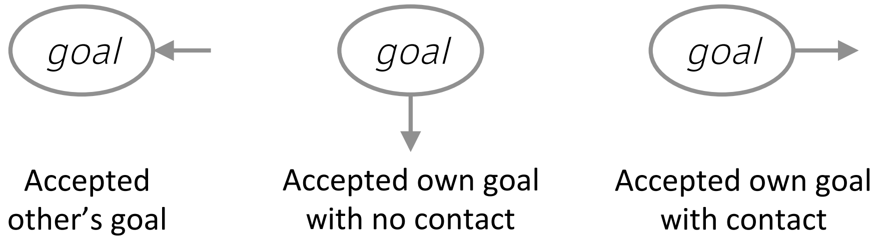

goal
goal
goal
Accepted other's goal
Accepted own goal with no contact
Accepted own goal with contact

Fig 3.1: The three types of willingness state representing situated goal acceptance.

3.2 Interpersonal Situations

3.2.1 Willingness State

To begin with, we will assume two interacting agents. One is the human and one is the robot. Both agents are looked as equal (similar to the distributed cognition approach in HCI).

Looking at the ways a goal (objective context) by an agent can be tied to the goal of the other agent, we find that there are three patterns. The patterns are based on *how the current goal by the agent is in relation to the goal of the other agent* (Fig. 3.1). We will refer to these relations as connected arrows and name this discretized representation of a goal as a **willingness state**. The patterns are: 1) Agent's goal is irrelevant to the other agent (non-connecting arrows, the agent is not trying to initiate the interaction nor accepting the other agent), 2) Agent's goal is an action to the other agent (outward connecting arrows, the agent is trying to initiate the interaction), 3) Agent's goal is a reaction to the other agent (inward connecting arrows, the agent accepts that the other agent is trying to initiate the interaction).

Note that when it is only two agents that are interacting, the first agent's goal is prioritized in the non-connecting state and the outward connecting state. Whereas, the other agent's goal is prioritized in the inward connecting state. In addition, the state is named *willingness* as the state indicates whether the agent is willing to accept a change in goal (or the *re-organization* as explained in Chapter 2) from the other agent.

In a simple question-and-answer information-seeking situation, it is easy to understand that the outward connecting arrow corresponds to questioning, and that the inward connecting arrow corresponds to answering. In an information-providing situation where one agent believes that the other agent may want the information, and therefore informs, the outward connecting arrow corresponds to informing, and the inward connecting arrow corresponds to receiving.

3.2.2 Willingness in Different View Points

The discussion becomes tricky when one agent is trying to seek information, while the other agent is waiting to provide information e.g. a situation at an information desk. One way to look at this situation is, both are at a willingness state of an outward connection (let us denote this way of understanding the situation as view A). The other way of looking at the situation is, the information-seeking agent was willing to interact before the information-providing agent noticed the other agent. The information-providing agent only became willing once after noticing the other agent, and therefore, the information-seeking agent is outward connecting but the information-providing agent is inward connecting (view B).

From a third person view, both views A and B are correct. A third person does not know the truth of which willingness happened first, and therefore, it can be either situation.

From a first person view, there is only one answer. If the information-providing agent scented an outward connection of the other agent before its own change in willingness state, the agent believes its view as B. If the information-providing agent scented the other agent's willingness state as a non-connecting state, the agent believes its view as A.

Now let us say there is a god that knew each agent's belief on the other agent's willingness. If one agent believed that the other agent was willing beforehand and if the other agent believed that the other agent reacted to its own willingness, the view is B. If both agents believed that the other agent was not willing, the view is A.

Since we are handling a mental model that is not observable, the idea of willingness often causes confusion depending on which view point we are looking at the problem. To clarify, we will use the following view points for each of the following topics: when we are theoretically discussing interpersonal situations, we will take the god view, when we are annotating interpersonal situations, we will take the third person view, when we are implementing interpersonal situation understanding to robots, we will take the first person view (estimated god view on the robot).

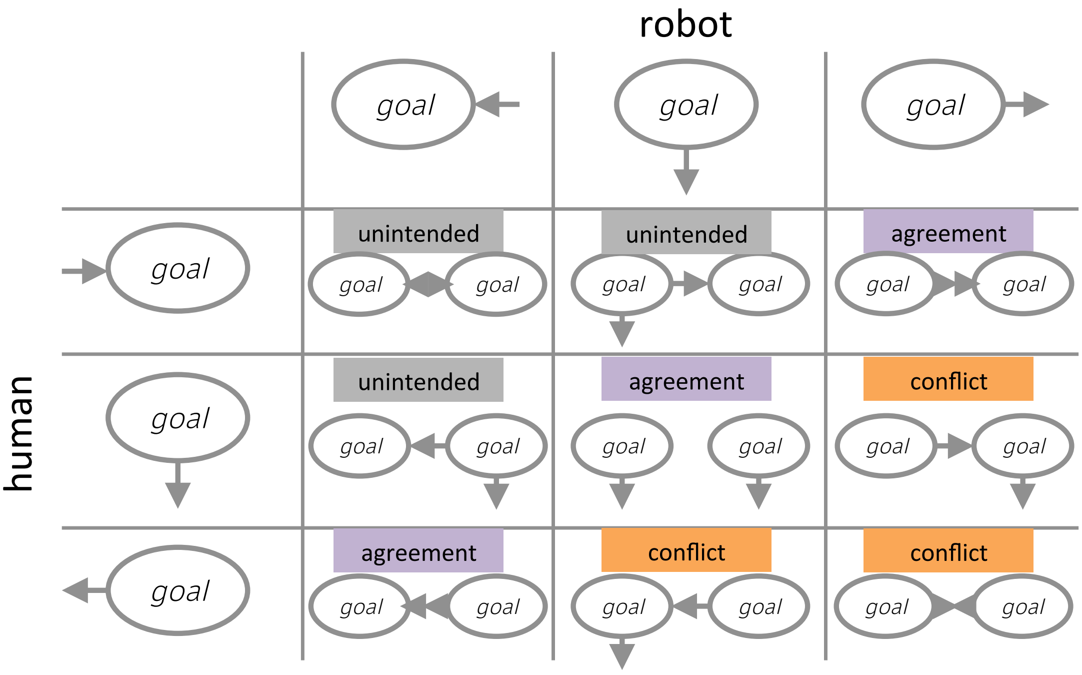


Fig 3.2: Nine interpersonal situations from willingness state. A god's view of situations.

3.2.3 Conflict and Agreement

In the previous section, we have explained the three willingness state of an agent. Depending on the willingness state of each agent in god view, there are nine possible situations as shown in Fig. 3.2 . Notice that while the two goals are not conflicting in some of the situations, the goals are still in conflict in others. For example, in situation 2-3, the robot's goal is now aligned with the human's goal. In situation 1-1, although the robot's goal and human's goal differ, they are not conflicting and is under an **agreement** to not bother each other. However, in situation 2-1, the human is trying to re-organize the robot's goal, while the robot is continuing its own goal.

Note that the interpersonal situation could alter from a conflict situation to an agreement situation, and then to a different agreement situation. For example, the human and robot may have agreed to discuss the details of the goal to proceed (which is either situation 2-3 or 3-2). The human and robot may end up finding out the goal is impossible to accomplish, and therefore decides not to proceed the goal (which is situation 1-1). For discussion purposes, we will distinguish the first agreement with any agreement that happens after the first agreement. We will define any interpersonal situations that happen between the previous task and until the first agreement as an **initial interaction** and any situation happening after the initial interaction until the end of an interaction task as a **during interaction**. For most simple task interruptions, the final agreed situation is equal to the agreed situation of the initial interaction (as we show in the experiments section). Therefore, we will mainly focus on the initial interaction.

An interesting situation is situation 2-2. This is the situation where a human tries to re-organize a robot's goal while at the same time, the robot tries to re-organize the human's goal. A real example of this situation can be given at a restaurant. A waiter tries to approach a customer to ask whether he or she requires a new drink for an empty glass. At the same time, a customer may try to ask for additional food orders. If the waiter was a robot, the robot should understand that the customer has something to say, and —depending on the situation —listen to the customer's request first. Such a situation is not characterized in any of the engagement detectors discussed in Chapter 2, which shows how understanding the total setting is different from simple engagement.

Other interesting situations are 1-3, 3-1, and 3-3. These situations are unique situations we find from discussion on willingness. We may call these situations as **unintended** situations. Unintended situations are unintended conflicts that would otherwise have been a 1-1 agreement. An example would be when a person asks, ”did you say something to me?” even when no one was calling him or her. For robots, such situations can often happen due to recognition failures. Table. 3.1 summarize the interaction type of each situation. Example of actual situations are shown in Fig. 3.3 .

3.2.4 Multiple Agents

The discussion can easily be extended to multiple agents. For example, if a robot is talking to person-A while a different person person-B is trying to talk with the robot, the situation between the robot and person-A is 2-3 (3-2), the situation between the robot and person-B is 2-1, and the situation between person-A and person-B is 1-1. The point is that the current accepted goal of the robot is relevant to person-A but irrelevant to person-B. Likewise, when the robot, person-A, and person-B are all talking together, they are all under agreement.

Table 3.1: Interaction type of each situation.

state human-robot	interpersonal situation
1-1 (H_n-R_n)	agreement: no interaction
1-2 (H_n-R_p)	conflict: robot to human interaction
1-3 (H_n-R_r)	unintended: mistaken by robot
2-1 (H_p-R_n)	conflict: human to robot interaction
2-2 (H_p-R_p)	conflict: purpose-crossing interaction
2-3 (H_p-R_r)	agreement: interacting for human's goal
3-1 (H_r-R_n)	unintended: mistaken by human
3-2 (H_r-R_p)	agreement: interacting for robot's goal
3-3 (H_r-R_r)	unintended: mistaken by both

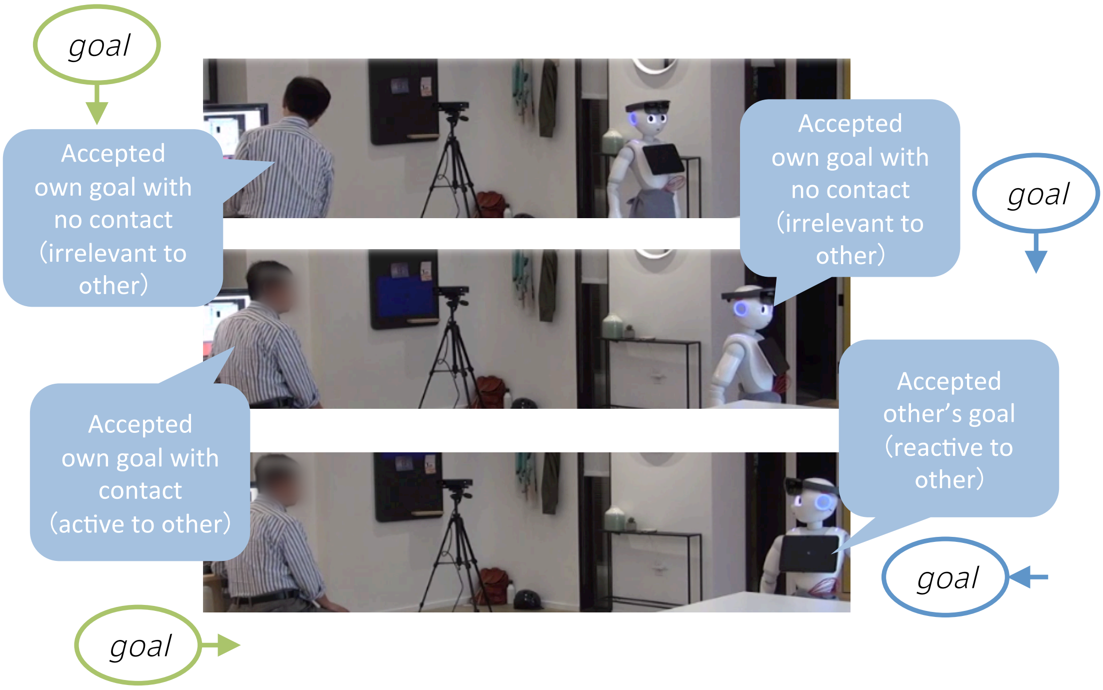

goal
Accepted own goal with no contact (irrelevant to other)
Accepted own goal with no contact (irrelevant to other)
goal
Accepted own goal with contact (active to other)
Accepted other's goal (reactive to other)
goal
goal

Fig 3.3: Example pictures of interpersonal situations.

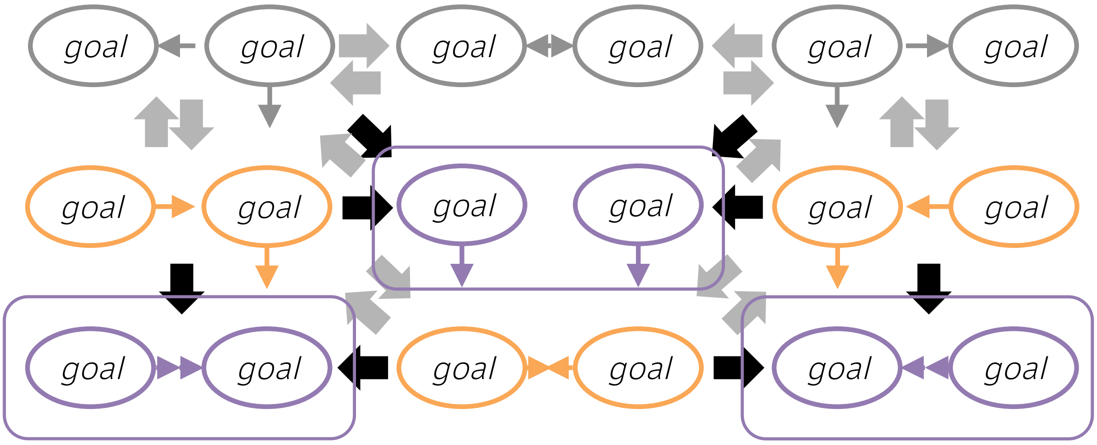


Fig 3.4: Transition between each interpersonal situation at an initial interaction.

3.2.5 Transitions between Interpersonal Situations

Fig. 3.4 shows the possible transitions to reach an agreement at an initial interaction. Here, we assume that only one goal is changed one at a time. Also, since we are focusing on an initial interaction, the two agents will no longer change their goals once reached an agreement.

An interaction situation is determined from the god view willingness state of the robot and the god view willingness state of the human. A robot may start from a willingness state depending on the flow of the task. For example, if the robot is doing its task, it is at a non-connected (irrelevant) state. If the robot needs to contact a human before continuing the task, it is at an outward connected (active) state. However, a willingness state will not start from an inward connected (reactive) state, as this is a goal outside of the task flow. Likewise, a human's willingness will also start from either a non-connected or outward connected state. Depending on the starting situation, the robot and human will change the situation based on their behavior until reached an agreement situation. In the next section, we will go over in detail, how such transitioning of situations can be modeled as a probability graph for computation.

3.3 Situation Scenting Model

3.3.1 Modeling the Decision Process

In order for a robot to understand its current interaction situation, the robot must know the human's willingness state. However, this is hidden to the robot and must be guessed from sensor observations on human behavior. On the other hand, the robot knows its own state. In an interaction situation, a robot will conduct some behavior related to the robot's desire (primary willingness). The robot behavior will influence a human's willingness state, and thus, also influence the perceived human behavior. Likewise, the human behaviors —which implicitly indicate a change or no change in human's willingness state —will influence the robot's next behaviors. Using a time index i, we can illustrate such relations of human willingness state x_i, robot primary willingness y_0, robot behavior a_i, and human behavior o_i as in Fig. 3.5 . We name this graph representation of human-robot interaction from the perspective of willingness as **the situation scenting model**. (In [119], we use the term "agreement model". The term is specifically used under the HRI context. However, in this book, we will be speaking of interaction in relation to the total setting. Therefore, a more general terminology "situation scenting" is used instead of "agreement".) We will simplify the graph representation by implicitly representing the robot's willingness y_i with a_i ($a_i \propto y_i$). The model assumes that behavior and willingness relations are Markov i.e. there are no latency in behavior and influence on one's willingness. We will discuss whether this assumption is valid later on in the discussion section.

3.3.2 Transition and Emission Probabilities

Although we take into account human behaviors for controlling our robot behaviors, we assume that the robot behaviors are independently controlled and that we are able to ignore the probabilistic relation of human behavior causing robot behavior. From this assumption, the agreement model is nothing more than a Hidden Markov Model (HMM) by defining a joint willingness $X_i = (x_i, a_i)$ as states, and defining human behavior o_i as observations. Therefore, we are able to solve the problem of understanding interaction situations as an HMM problem. Our probabilities of interest are the transition probability $P(X_i|X_{i-1}) = P(x_i, a_i|x_{i-1}, a_{i-1})$ and emission probability $P(o_i|X_i) = P(o_i|x_i, a_i)$. For the transition probability, we are able to further rewrite the probabil-

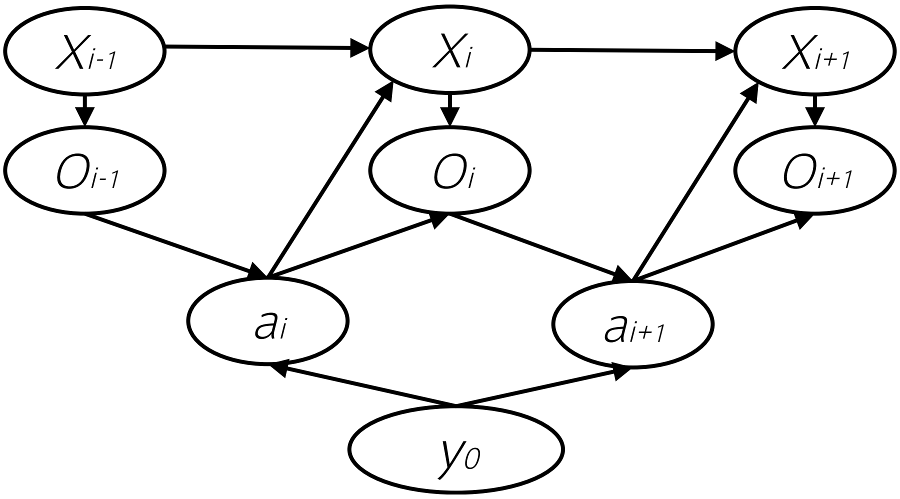


Fig 3.5: The situation scenting model. x_i are the hidden human willingness state. o_i are the human behaviors observable by the robot. a_i are the robot behaviors. y_0 is the robot's primary willingness.[119]

ity using Bayes ball, and conditional independence between a_{i-1}, x_i given joint (x_{i-1}, a_i). That is (denoting time index $i-1$ as j)

$$\begin{aligned} P(x_i, a_i|x_j, a_j) &= \frac{P(a_j|x_i, a_i, x_j)P(x_i|a_i, x_j)P(a_i|x_j)}{P(a_j|x_j)} \\ &= \frac{P(a_j|x_j, a_i)P(x_i|x_j, a_i)P(x_i|a_i, x_j)P(a_i|x_j)}{P(a_j|x_j)} \\ &= P(a_i|a_j)P(x_i|a_i, x_j) \end{aligned} \tag{3.1}$$

As the robot behavior is independent from the previous behavior, $P(a_i|a_{i-1}) = P(a_i)$. $P(a_i)$ represents a scaling factor. Thus, $P(x_i, a_i|x_{i-1}, a_{i-1}) \propto P(x_i|a_i, x_{i-1})$. In summary, we will analyze human-robot interaction situations based on the two probabilities $P(x_i|a_i, x_{i-1})$ and $P(o_i|x_i, a_i)$.

3.3.3 Observations

From a technical viewpoint, no observations are easy to achieve when it comes to human behavior. However, human head directions have been used as an approximation of human gaze (perceptual) direction for engagement detection ([4, 73]). Moreover, gaze has an important connection with PCT; which is the basis from the psychology field we are using to explain interpersonal situations.

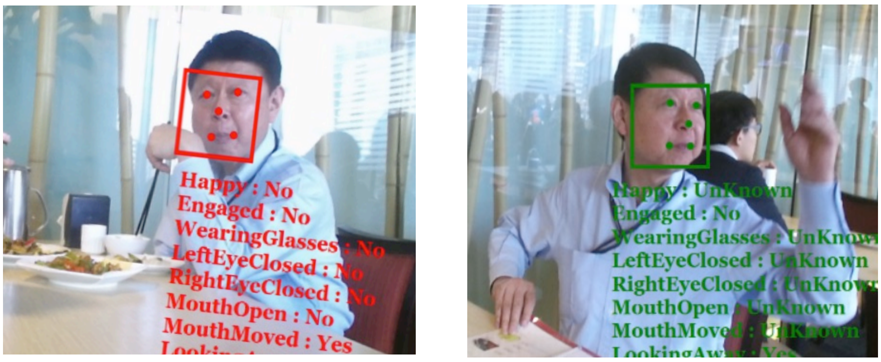


Fig 3.6: Two examples where a person is looking away from the table. Both are looking away, however the person is under an internal state irrelevant to the another agent with the left image, but is acting to another agent with the right image.

In the PCT assumption, an agent behaves according to what it perceives. Therefore, the observation *where the person is looking (eye gaze)* most likely correlates to the human's current goal. As we defined the willingness state in relation to goals, it is not a surprise that the observation of human gaze or head direction tells us the willingness state of the person. However, the human gaze direction does not exactly tell the willingness state and contains uncertainty. When a person behaves according to an internal decision (e.g. emotions, thoughts), the gaze direction is irrelevant to goals of the other agents. Therefore, even if the person was looking toward the robot, there is a chance that the gaze could be an acceptance of another agent's goal but also a chance that the gaze is a result of an internal state. The willingness state is remained hidden. Real examples are shown in Fig. 3.6 . Although in both pictures, the person is looking away from the table, one picture indicates a non-connecting willingness state while the other indicates an outward connecting willingness state.

Although we will be using gaze by default, the important logic behind is that, there is some type of observation that indicate a higher probability of acceptance on a different agent's goal (disturbance) over the current agent's goal. In general, such an observation could be a binary observation of *more toward the other agent's goal* and *more toward its own goal*. In the case of gaze, the binary is *the person is looking toward the robot* and *the person is looking away from the robot*.

3.3.4 Robot Behaviors

As discussed in the previous sections, a robot's behavior a is an implicit representation of the robot's willingness y, where y is one of the following: 1) Agent's goal is irrelevant to the other agent (R_n), 2) Agent's goal is an action to the other agent (R_p), 3) Agent's goal is a reaction to the other agent (R_r). (Likewise, we will use H_n, H_p, H_r to indicate the human's willingness. See section 3.4.2 for details.)

The concrete behavior itself can be of any motion by the robot. In addition, the above behavior categories are not specific to interaction behaviors as they are categorized according to goals. Any task action will implicitly indicate one of the behavior categories. For example, if a robot is not interacting and is doing a task action, the goal is irrelevant to the other agents, and therefore, the task actions are all R_n behaviors despite the content of the action.

In a multiple people setting, a robot could be talking with person-A while person-B is trying to interrupt the conversation. The robot's behavior is R_r to person-A but R_n to person-B. As in this example, a robot's behavior category is not dependent on the robot's motion and may differ depending on the opponent.

Although concrete behaviors depend on the objective, social, or physical context, we may define a generalized behavior category-transitioning pattern for the robot. In section 3.2, we went over the possible transitions from a conflict situation to an agreement situation. The objective of the situation engine (especially for the initial interaction) is to resolve a conflict situation and enter an agreement situation. From the transitions in Fig. 3.4 , we are able to define a behavior category transitioning pattern to meet an agreement situation. The pattern and procedure is shown below.

First, we have stated that a behavior is decided from human behavior o and primary willingness y_0. y_0 is determined by task flow. When a robot is doing its task or not interacting, y_0 is R_n. When a robot must approach a human to begin an interaction, y_0 is R_p (including situations where a robot is actively moving toward the human). In addition, using our situation scenting model, we are able to estimate human willingness x' from observation o. That is, we calculate the posterior probability (using a recursive function f) of each possible willingness state, and select the most likely state as our estimate.

$$\begin{aligned} x'_i &= \operatorname*{argmax}_x P(x|o_{0:i}, a_{0:i}) \\ &\propto \operatorname*{argmax}_x P(x, a_i|o_{0:i}, a_{0:i}) \\ &\propto \operatorname*{argmax}_x P(o_i|x, a_i) \sum_j f_j(i-1) P(x, a_i|x_j, a_{i-1}) \end{aligned} \tag{3.2}$$

Using x' we are able to understand the interpersonal situation, thus, a robot's willingness changes from y_0 to a different state y if in conflict. When a robot is capable of interacting (which is decided by task flow) and x' is likely H_p, y will change to R_r to resolve the conflict. When a robot fails to interact with R_p (x' does not change from H_n), y will change to R_n or will stay R_p depending on task flow. When a robot detects x' as H_r while y is R_p, y will change to any state defined by task flow.

3.3.5 Agreement Level in Behaviors

If we compare different behaviors of the same category, we will notice that some behaviors may lead to an agreement quicker. For example, *stare at the person*, *stare and wave the robot's hand*, or even *grab a person's shoulder and force him or her to look toward the robot* are all R_p actions but will have a different effect when leading the situation to an agreement. We may define such differences as an **agreement level**. In general, we may model multiple behaviors of the same category but with different levels. However, the more levels we add, the more complex our model will become. For simplicity, we will consider only two types of level in this book: *normal basis (steady) level* and one *resolve faster level* that uses speech. To distinguish different level behavior, we will use the following notations in this book: R_{n0}, R_{p0}, R_{p1}, R_{r0}, R_{r1}, where 0 indicates the steady level (no speech behavior) and 1 indicates the resolve faster level (speech behavior). R_{n1} is not included as this was not observed from the dataset we used to train the model (although theoretically, R_n may also have a speech level behavior such as speaking ”go away”).

3.3.6 Forward Guessing

One of the problems with using HMM based models on an application running in real time, is that, the HMM may not find the best estimate from the current observations. For example, from one observation, if the person is looking away, the model might estimate x_0 as H_n. However, after three observations, the model might update its estimate on x_0 as H_p after seeing the other observations. Although this is not a problem if we were to only estimate the situation (we may keep on observing until reached a confidence threshold), it is a problem for the situation scenting model since the robot needs to decide its next actions from the most recent estimation. The inaccuracy of the estimate with few observations will delay the robot's response. It may take a while for the robot to realize that x is actually H_p instead of H_n. We do not want the robot to wait till it is confident, but rather have the robot behave so that it can gain more confidence on the situation. The robot should behave whether it is confident or not, but at the same time, should behave in a reasonable way. This is partly why we call it *scenting* rather than *detecting*.

To solve this problem we use a technique we call **forward guessing**. We assume that the latest observation continues for a while in the future. That is, if the current observation was *look away, look toward*, we will assume that the observations in the coming future would be *look away, look toward, look toward, look toward.* For deciding the robot's next behavior, we use the estimation from the forward guessed sequence. (We only use the forward guessed sequence for deciding the behavior. For understanding the situation, we will use the estimation from the raw observation sequence.) The strategy is naïve, yet is simple enough to apply to any observation sequence.

A question related to forward guessing, is *to what magnitude should we seek into the future?* When the current observation is *look away, look toward*, how many *look toward* should we add to the raw sequence? The less we seek into the future, the more similar result to the raw sequence we will get. But what happens if we seek too much in the future? There is an interesting and reasonable result regarding this question. Using the probabilities we discuss in section 3.4, let us assume the situation where a robot is R_p trying to re-organize the human's goal, and we observe a *looking toward* after an R_{p1} followed by an R_{p0} behavior. An intuitive guess would be that x is H_r as the person looked back at the robot after the robot's speech. With a reasonable seek in the future, our model also returns an intuitive guess H_r. However, if we seek long into the future, our model returns H_p. There is a reasonable and intuitive explanation for this result. The key is that the final robot behavior was R_{p0}. This means that in the seeked future, the person is looking toward the robot for a long time although the robot is not in speech. It would be more intuitive for the person to return to his or her goal if the robot had nothing to say. If that goal were irrelevant to the robot, the person would most likely look away from the robot (at least from the PCT perspective). However, if the goal was relevant to the robot, it is understandable that the person is not looking away from the robot. Such a situation is H_p. As shown in this result, a long seek into the future may give a sequence that would have a completely different meaning from the current situation. Therefore, we should not seek too close but not too far in the future. In the experiments, we seek three frames of the future.

3.3.7 Backward Trimming

The other problem with using HMM models on runtime is, the observations may be continuous. For example, let us say that a person is nearby the robot and the robot is able to observe the person the whole time. When there is no interaction, the person is looking away from the robot. This may lead to a very long observation of *look aways*. Usually (and for the recordings and experiments) this was not a problem as a person appeared and disappeared from the robot's view. The sequence was re-initialized every time the person re-appeared in the view. However, this might not be always the case in some situations. For example, think of a receptionist-helping robot. The robot is always next to the receptionist and responds only when the receptionist is about to give an order. In such a scenario, the initial state would refer to the situation that was happening first thing in the morning. The initial state would most likely give a wrong effect if this was used for estimating something happening in the afternoon. (The model will try to estimate the human willingness from the global situation rather than what is happening locally at the moment.)

To solve this problem, we use a technique we call **backward trimming**. When we observe a *look toward* observation after a *look away* observation, we will look back at the sequence and see how long the person was *looking away*. If the *look away* sequence was very long, we will trim the current sequence so that the current sequence looks as if a new person appeared in the scene to begin an interaction.

How far in the past should we seek? We must be careful, as a *look away* between two *look toward* may sometimes be valuable. For example, ”that person was looking toward me a second ago, but looked away quickly, now the person is looking to me again” is a valuable information in that, it could lead to a guess of ”this person may have a habit of looking around, so this *looking* behavior may just be a coincidence rather than a sign of engagement.” Therefore, we should not

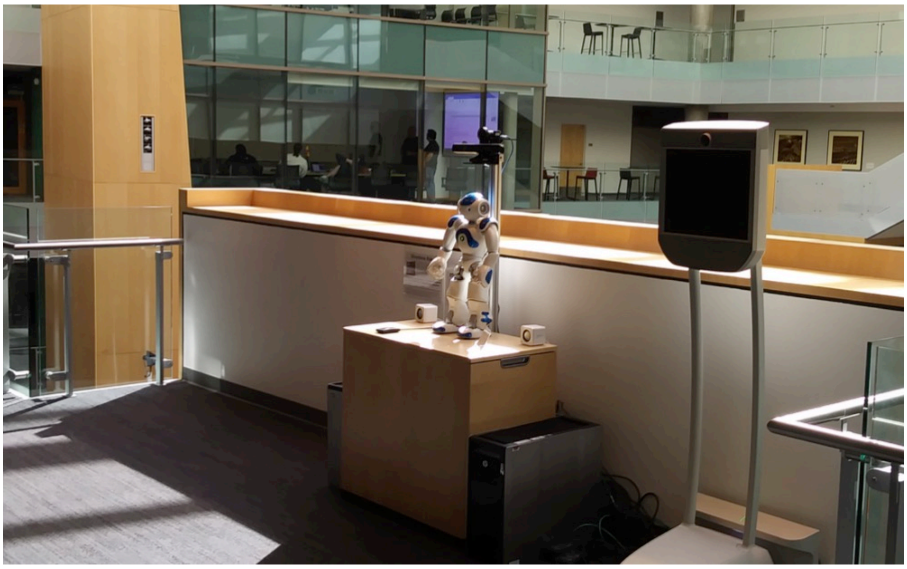


Fig 3.7: The directions robot setting in the HRI video recordings used for training our model.

decide to trim by only looking into a few shots of the past. Yet, at the same time, we must decide to trim before looking too far into the past. We should not leave an interaction context that happened far in the past in the current sequence. In the experiments, we seek five frames of the past. If the observation of the last five frames is *look away*, we will reset the observation sequence with leaving only the most recent *look away*.

3.4 Understanding Model Characteristics from Recordings

3.4.1 Setup

First we will discuss how our model applies to a real HRI scenario. For the HRI scenario, we will use in-the-wild HRI video recordings from the directions robot [14]. An image of the setting is shown in Fig. 3.7 . The recordings capture a quantitative number of daily-situated interaction beginnings and endings, both successful and not successful. In the recordings, a Nao robot positioned in front of an elevator gave directions to people who came by close to the robot. Each recording began with a person (or group of people) coming in toward the robot and ended with a walking out from the recording camera. Most people interacted with the robot by asking directions or by playing with the robot, while others refused to interact after a robot speech. In most occasions, the robot had floor for starting the conversation and —although people sometimes were in groups—interacted with one person who was most engaged. The robot was autonomous, used an external wide-range 2D-camera to capture human faces, an external Kinect microphone array for speech recognition, but used default speaking and motion capability of the robot. We use 93 of the recordings and add labels to the recordings from the perspective of willingness. These labels will help understand the characteristics of willingness in a real HRI situation. Also, the labels will be used to train the situation scenting model for the other experiments.

3.4.2 Labeling Human Willingness on Recordings

As discussed in section 3.2, a human willingness state x has three categories: 1) Agent's goal is irrelevant to the other agent (H_n), 2) Agent's goal is an action to the other agent (H_p), 3) Agent's goal is a reaction to the other agent (H_r). Therefore, we will label each video with one of these states. Note that the state could change multiple times in one video.

One problem with labeling human willingness state on existing video recordings is, we do not have the ground truth from the participants (although, the participants would not be able to concentrate on the interaction if we were to ask for the ground truth). There are no pre-defined labeling rules as the human behavior may vary in many ways. However, by observing multiple videos of similar situation, we are able to extract a speech or motion pattern of the human. We will begin by finding whether there is a possible willingness state over the finite number of patterned human behaviors. The found patterns in the recordings are listed below.

1. A human ignores a robot speech.

2. A human conducts a behavior not related to the context of the robot speech.

3. A human talks to the robot before the robot speech.

4. A human waits for an answer from a robot.

5. A human behaves according to a robot's favor (e.g. ”swipe a badge”).

6. A human responds with an utterance (e.g. ”uh...”).

7. A human quickly responds with a request after the robot speech.

8. A human quickly responds with a negative response.

9. A human starts playing with a robot after a negative response.

In some of the patterns, the willingness can be easily labeled as the situation matches the definition of one of the states. In pattern 1 and 2, it is obvious that the goal of the robot is not being accepted, nor is the person willing to interact with the robot. Therefore, the willingness state x must be H_n. In pattern 3, the person is trying to contact the robot to re-organize its goal. In pattern 4, the person is waiting for a re-organization. Therefore, in these two patterns x is H_p. In pattern 5, it is obvious that the person is accepting the robot's goal and therefore x must be H_r.

Other patterns require a discussion before labeling. For instance, pattern 6 is an interesting reaction we see in real HRI. The person may be thinking of how to word his or her goal and x could be H_p. However, in the recordings, the first question by the robot was ”can I help you find something?” which can be immediately answered by a ”yes” if x was H_p. Therefore, we have labeled pattern 6 as H_r. In contrast —from our above discussion—pattern 7 is H_p as the person immediately responded to the robot's question. Pattern 8 is a tricky one. The global willingness is H_n as the person is trying to end up with a 1-1 agreement. However, to reach a 1-1 agreement, the person first responds to the robot's question, which is a 3-2 agreement. We will label this pattern as a change from H_n to H_r, and then back to H_n to distinguish between no reaction and with reaction. Note that in reality, a robot will not immediately accept an agreement situation. Although a robot understands the situation as an agreement, it will not accept the situation as a real agreement until there is enough confidence. If trained correctly, the model should detect a 3-2 situation but with low confidence for patterns like 8. It should only accept the final 1-1 agreement and therefore, the whole interaction can be seen as a continuation of an initial interaction. The last pattern is a unique pattern found in the recordings. A person answers ”no” to the robot's question ”can I help you find something?” but is actually willing to interact with the robot. Again, the global willingness is H_p but the local response is H_r. Therefore, the pattern can be seen as a change from H_r to H_p.

Besides the above patterns there were some ambiguous patterns where a person was looking around near the robot but not speaking. We have labeled all ambiguous patterns as H_n. The reason is that, from the perspective of PCT, it is less likely for a person to look around if the goal is targeted toward the robot. It is most likely that the person is under an internal input or a different goal irrelevant to the robot.

[55] discusses the labeling of engagement and points out that there exist labeling differences depending on the annotator's character. However, their focus is mainly on a conversational interaction. We will see from the labeling in our recordings that, when there is a task skill that the robot provides, the engagement (willingness) of a person does not alter as frequently as in a conversational context. Therefore, we do not take into account an annotator's character and mainly label according to pre-defined rules.

3.4.3 Labeling Observations on Recordings

For the observations, a human annotator looked through each recording and annotated whether the human head was facing toward the robot (O_{tw}), or whether the head was facing away (O_{aw}), or whether the head was facing downwards such as looking at cellphones (O_{dw}). All ambiguous directions were counted as O_{tw} ($\kappa = 0.693$). The head direction was annotated and not the actual gazing direction.

3.4.4 Labeling Robot Behavior on Recordings

As explained in the previous section, the robot behaviors depend on the context of the interaction rather than the action content. A robot behavior is an approximation of the robot's objective context that is hidden to the annotator, and the annotator may only see the interaction as a third person view. We have already discussed that an interpersonal situation may be multiple in the third person view. Therefore, we will double sample some of the videos depending on the possible interpretations of the situation. That is, for videos that may have multiple possible contexts (specifically two different context for the recordings we used), we will label the same video in two different ways. In the first label, we label the context as *the robot is not willing to interact before the interaction, but begins an interaction because a person has approached the robot*. In the second label, we label the context as *the robot is willing to request an interaction, and tries to encourage approaching people to ask directions*. Although the sampled video is exactly the same, the labeled behavior is completely different. In the first case, y_0 is R_n changing to R_r. In the second case, y_0 is R_p.

In most interactions, y_0 has only one context R_n. However, when the human willingness x_0 is H_n, we also sample the interaction as a $y_0 = R_p$ context. The exceptional interactions where y_0 is not a sample of R_n and only R_p, is when *a person has asked a question before a robot's first speech, but the robot begins a greeting and encourages the person to ask for directions*. This situation is 2-2 —although both the human and robot are willing to interact—the conversation is conflicting. For all other y, we annotate R_p or R_r depending on speech context of the robot (until the next robot speech, y does not change). We annotate the robot behaviors from annotated y and from whether the robot is talking at the moment or not. When a robot is not talking (R_{p0}, R_{r0}), the robot silently gazes toward the human. When the robot is not gazing and is at a neutral state, it is R_{n0}.

3.4.5 Results on Human Behaviors from Recordings

When all o, x, a are not changing for more than a second, we have divided the interaction into multiple segments. We analyze and count up the occurring situation patterns from these segments.

We summarize our results in Table. 3.2 . Four sets of 23 recordings were randomly chosen from the 93 interaction recordings (no duplicates between sets). The table represents the mean probability of the four sets and its standard deviation. Paired t-test was conducted for analysis.

We first summarize our results during the initial interaction. In the recordings, we have assumed that an agreement was established after the first two greeting speeches. In our discussion with labeling human willingness states, we have explained that a robot only accepts an agreement situation when it is confident. Even if our labels indicate an agreement, this does not indicate that the robot has detected an agreement with confidence. While the robot is greeting, we have assumed that the situation is: the robot is trying to understand the current interaction situation better. That is, a greeting is not part of the interaction content of the agreed situation therefore, while the greeting is on going, the robot has not yet reached an agreement. When an interaction ends with only greeting speeches, we assume that there were no during interactions and only an initial interaction ending with a 1-1 agreement.

The table shows that O_{tw} is higher when x is H_p and y is R_n, but lower when x is H_n and y is R_p ($p < .05$), i.e. **a human looks toward the robot more when it is the human to request**

an interaction, compared with when it is the robot that requests an interaction. When the situation is purpose-crossing (H_p-R_p), a human looks toward even more ($p < .05$). The table also indicates the different effect of robot speech. When it is the human who is requesting to interact, speech or no speech has little effect on human looking behavior (p=.602). However, when it is the robot that is requesting to interact, a human may (but not significantly) look less toward the robot when the robot is speaking (p=.117). In the recordings, the robot sometimes spoke to people who were talking over a phone. In these situations, people were looking at the robot but faced away once the robot had started speaking. The probabilities also indicate that people are sometimes not looking toward the robot even though they are requesting to interact. In the recordings, some people looked around while approaching the robot. These also include situations where people had glanced at another person when someone else was coming close by. Others were looking at their phone before interacting (they were checking for the room number they were finding).

The second part of the table summarizes the end of an interaction where a human is finishing an interaction but a robot is still requesting to interact. Although a person is no longer willing to interact, the person may sometimes keep looking at the robot while leaving the conversation. The table indicates that **when a robot's willingness is R_r at an end of an interaction, and if the robot was speaking, the more likely that a human will look back at the robot** ($p < .10$). **However, when a robot's willingness is R_p such results were not found** (p=.468). In the recordings, there were several situations where a person heard a robot's request, reacted to the request while the robot was still in speech, and left (without looking back) before the robot had finished its speech.

The third part of the table summarizes the transition probability of human's willingness x at the beginning of an interaction. The table indicates that **a human will more likely change from a non-willing to interact state when a robot speaks** ($p < .05$, excluding one set with no goal change samples). When a human is not willing to interact (H_n) and the robot is only staring (R_{p0}, R_{r0}), the chance of a human state change is the same as not interacting (R_{n0}) (p=.398,.513). For the recordings we used, the probability that a human changes his or her willingness H_n, H_p is low. In the dataset, people usually had a mind set before interacting (those who wanted to interact approached the robot for interaction purpose, those who did not want to interact were coincidently standing near the robot). Another indication from the table is that, H_r changes even if the robot is still requesting to interact (R_p). However, this is due to how we analyzed the data. People often responded a ”no” which is —in the long context—a declaration of not willing to interact. We have analyzed ”no” responses as H_r to distinguish from ignoring behaviors. Probability is lower when a robot is speaking, due to the fact that people often respond ”no” right after robot speech.

The results also show probabilities for rare unintended situations discussed in section 3.2. These are situations where a robot mistakenly reacts and a human starts interacting due to the mistaken reaction. Such situations were analyzed using our double sampling method. The results indicate that, **when a robot's willingness is R_r, the chances of a person looking toward a robot is the same despite whether the human's willingness is H_r or H_p** (no significant difference as p=.333,.588). In addition, observations on when a robot speaks toward a person with H_r varied among the recording sets. As we discuss in the next section, this is partly due to the lack of recordings of such situation.

3.4.6 Limited Sample Situations in Recordings

In most of the recordings, a person did not change his or her goal throughout the interaction. We did not find as many H_r-R_p situations where a robot had a desire of giving directions and a person who did not care about directions started listening for directions. However, such cases were not zero. In two recordings, a person changed from H_p to H_r. A person tried shaking hands with the robot, but the robot tried to give directions. The person gave up shaking hands and decided to ask for directions. The other situation happened when the robot could not catch where the person wanted to go. The person first tried to correct the robot but then decided to —although the directions were not the answers she had intended—listen to the robot's directions. In nine recordings, a person changed from H_n to H_r. The most common pattern of this type was: a person had left but noticed that the robot had something additional to say and then came back to interact. Another common pattern was: two or more people were talking with each other but the robot was interrupting their conversation. As they had been interrupted, one of the persons decided to talk with the robot.

Due to the recordings we used, we do not have enough results for when a robot keeps ignoring a person who is requesting to interact. However, 11 recordings finished from robot speech recognition failure. In these recordings, people had retried recognition two to three times but finally gave up interacting as the robot kept on failing to understand human speech. In three recordings, the

Table 3.2: Situation and Probabilities from 92 Recordings (4 sets of 23 recordings each) [119]

probability	situation	mean	stddev
emission O_{tw}	H_n-R_{n0}	.301	±.0472
at beginning	H_n-R_{p0}, R_{p1}	.392, .303	±.0215,±.0954
	H_n-R_{r0}, R_{r1}	.398, .293	±.0104,±.123
	H_p-R_{n0}	.617	±.139
	H_p-R_{p0}, R_{p1}	.772, .798	±.200,±.124
	H_p-R_{r0}, R_{r1}	.788, .767	±.0242,±.0796
	H_r-R_{n0}	.833	±.319
	H_r-R_{p0}, R_{p1}	.853, .552	±.153,±.332
	H_r-R_{r0}, R_{r1}	.884, .635	±.148,±.423
emission O_{tw}	H_n-R_{p0}, R_{p1}	.258, .204	±.0746,±.154
at end	H_n-R_{r0}, R_{r1}	.206, .316	±.0990,±.108
transition of	$H_n \rightarrow H_n$-R_{n0}	.975	±.00713
no change in	$H_n \rightarrow H_n$-R_{p0}, R_{p1}	.951, .850	±.0419,±.104
willingness	$H_n \rightarrow H_n$-R_{r0}, R_{r1}	.961, .871	±.0333,±.0830
	$H_p \rightarrow H_p$-R_{n0}	.986	±.00946
	$H_p \rightarrow H_p$-R_{p0}, R_{p1}	.948, .964	±.0553,±.0684
	$H_p \rightarrow H_p$-R_{r0}, R_{r1}	.991, 1.0	±.00604,±.0
	$H_r \rightarrow H_r$-R_{n0}	.612	±.209
	$H_r \rightarrow H_r$-R_{p0}, R_{p1}	.732, .417	±.200,±.479
	$H_r \rightarrow H_r$-R_{r0}, R_{r1}	.766, .666	±.171,±.471

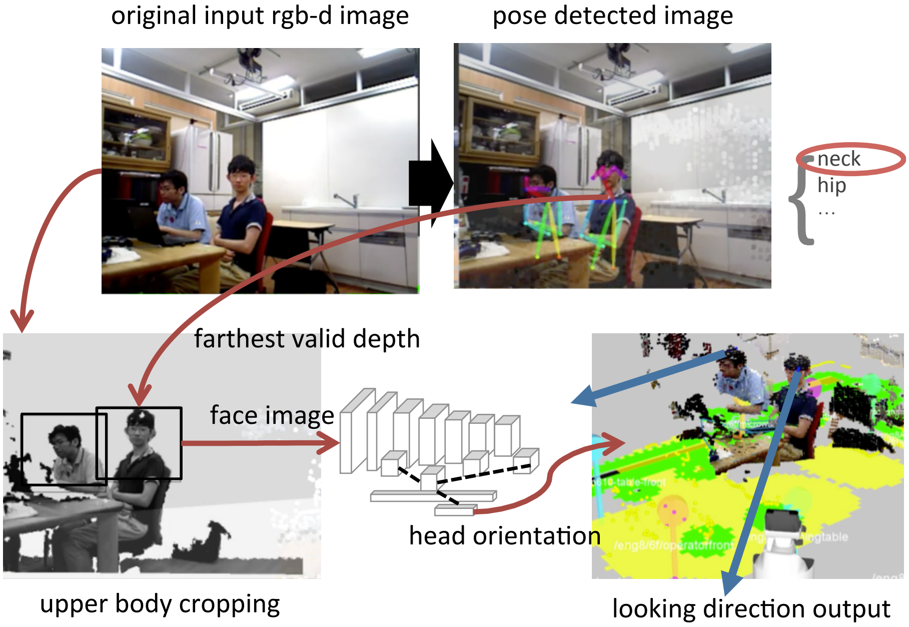


Fig 3.8: The detection pipeline used in the experiments for detecting human behavior. Note that a more accurate, fast, less computational, and stable detection pipeline is used for the other chapters.

robot accidently finished the interaction by misrecognizing ”uh...” as ”nah...”. In these situations, people tried to re-initiate the interaction with the robot, but the robot kept on ignoring them. The accidental finishes sometimes lead a person with willingness H_r to change to H_p. As in these samples, people do not easily give up interaction; however, the robot did not look busy when its goal was R_n. Different results might be observed when a robot actually *does* look busy when a person is trying to re-initiate an interaction.

Other limited samples include the purpose-crossing situation, which was found in only five of the recordings. In these recordings, the conversation was conflicting. Most of the time, the robot agreed with the human’s goal after speech recognition. Exceptions are the H_p to H_r recordings that we have already discussed.

3.5 Experiment

3.5.1 Applying Model on Real Robots

In the experiment, we will see how our situation scenting model will work on an autonomous interacting robot. For estimating the interaction situations, we will use the probabilities found from the recordings in the previous section. We will wait at least for two interaction segments before giving estimation. (A segment is until a change in observation or robot behavior is detected OR when all model variables o, x, a are not changing for more than a second.) Before, going on to the experiments, we will go over the technical settings to use our model in the scenario.

In order to capture whether a human is looking toward a robot, we either mounted an RGB-D camera or used the robot's internal camera. The placement of the camera was different depending on the robot; however, we used the same algorithm to detect human behavior. The algorithm consists of three parts. The first part uses RGB data and Open Pose [17] to detect human joint positions in pixels. Depth information is added to the neck joint of the human, and any pixel with greater depth value is subtracted as background. The second part crops the head boundary using the extracted pixel positions. An upper body Haar cascade detector is used on the subtracted image to find head boundaries. The third part uses the HyperFace [106] network to estimate human head poses from images. The cropped head image is passed to HyperFace and the estimated roll, pitch, yaw values are used to estimate head orientation in 3D space. A threshold was applied to distinguish looking toward and away from the orientation values. Note that we have used the head orientation as an approximation of gaze direction. (The state of the art eye gaze recognition is not yet reliable when faces are far from the camera. They require calibration or high-resolution cameras, as well as camera distance constraints [97]. Head movements are alternative cues to eye gaze, and the robustness and accuracy of head detection itself is much more reliable.) Although we use state-of-the-art technology to capture observations, the method has limitations in view range and timing. Depending on the camera resolution, at worst case, a human must be within 1[m] of the camera range for accuracy. In addition, we rely on external GPU machines for calculation. Depending on the robot's CPU power and network condition, a 1[sec] delay occurs while image is being processed. Although timing is not strictly handled and proximity is limited, we are still able to capture an overall scope on the occurring interaction situations. The detection pipeline is shown in Fig. 3.8 . The algorithm was used for the experiments in this section. In the later chapters, we use a more accurate, fast, less computational, and stable pipeline using single-shot multibox detectors [76] and datasets from [155].

Regarding the robot behaviors, we have already mentioned the decision process in section 3.3.4. However, we have not yet mentioned how the agreement level is decided. That is, the decision on whether the robot should speak or not. We will apply a decision rule based on findings from our analysis, which we will discuss more deeply in section 3.6. Here, we will explain the basic decision rule as below: a robot will speak once at the beginning when R_p, and not speak for R_r. As an exception, when a behavior is applied by task context (e.g. moving toward a person) the robot may begin with R_{p0}.

The concrete behaviors depend on the context of each experiment especially for R_n. For R_{p0} and R_{r0}, the robot will turn its head toward the person. For R_{p1} and R_{r1}, the robot will turn its head toward the person and also speak.

3.5.2 Experiments

We will do one experiment in a non-public in-lab domain and two experiments in a public domain. One of the public domain experiments was interaction-only similar to the recordings, but the robot may or may not have conversational floor. The other included a task situation. In each experiment, we implemented our model on top of a scenario flow designed by an engineer. (The purpose of the experiments was to see the effects of the situation scenting model, and therefore, the task side of the system was simplified. Complex task interrupting situations such as the one discussed in Chapter 5 were not part of the experiment.) A different engineer designed each scenario. When there were multiple people, the robot faced toward one of the persons for interaction.

In-lab Experiment

The first experiment was held in-lab where one participant (the task flow engineer) interacted in the scenario. The participant was debugging the task side of the scenario, and interacted as part of the task flow. The engineer was not informed how the robot would reason a situation; therefore, the setting captured natural human behaviors worth discussing.

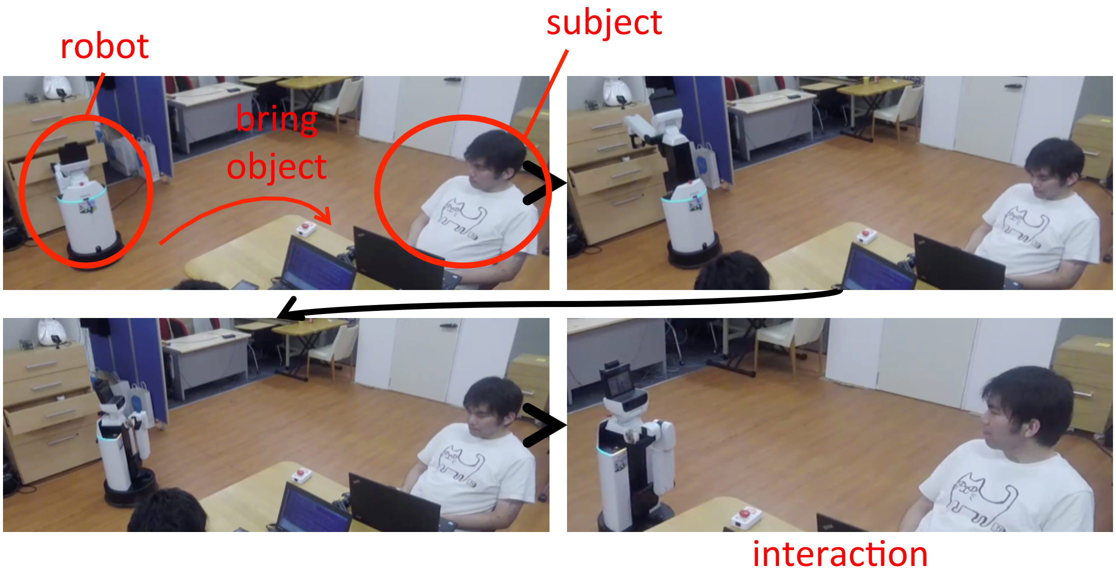


Fig 3.9: The in-lab experiment task flow.

The task had four phases. 1) The robot was looking around and moved close to a person who seemed to have had a request (R_r), 2) the robot listened to the person's command, 3) the robot conducted the command of finding and delivering an object, 4) after finding the object, the robot approached back to the person and confirmed an interaction before handing over the object (R_p). In phase 4, the robot said ”hey” to confirm an interaction with a maximum of three times when the situation was uncertain. Pictures of phase 4 are shown in Fig. 3.9 .

The participant was not always looking toward the robot at phase 4, and was sometimes looking at debug logs when the robot approached with an object. The participant caused different response timing. We experimented whether our situation scenting model would understand the different situations and thus, lead to different robot behaviors.

Estimation results are shown in Fig. 3.11 and the actual interaction in Fig. 3.10 . From the graphs and interaction images, we can see that the robot distinguished the two situations. In the first situation, the participant was focusing on the logs and looked back at the robot after it had said ”hey.” The robot estimated this situation as H_r-R_p. In the second situation, the participant was not focusing on the logs and looked back at the robot as it was approaching him. The robot estimated this situation as H_p-R_p.

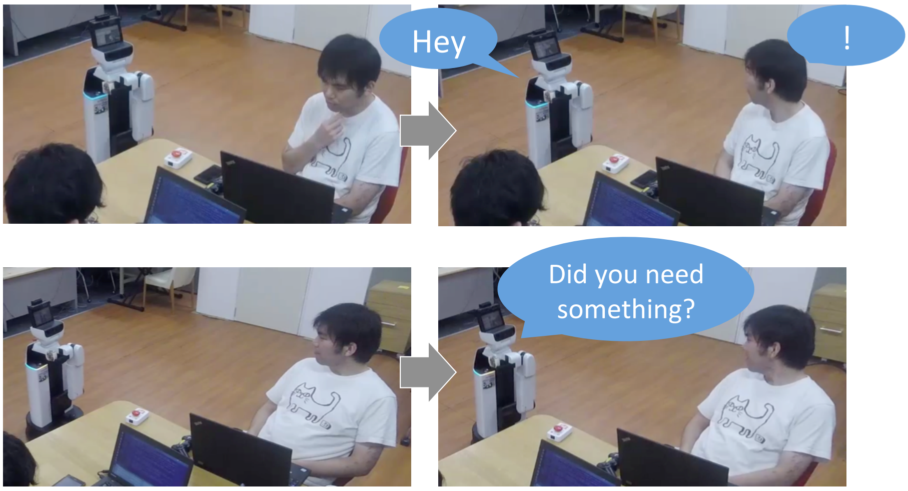


Fig 3.10: The two different interactions observed in the experiment. One where the participant's attention is toward the display (above) and one where the participant's attention is toward the robot (below) when the robot is approaching.

From the graphs, we also observe that the robot spoke ”hey” multiple times. Since the camera used was mounted on the robot's head, depending on the planned mobile path trajectory, the participant suddenly appeared in front of the robot's view. In such situations, the situation was sudden and uncertain to the robot. Therefore, the speech was triggered multiple times until there was enough certainty on the agreement situation.

Public Domain Social Context Only Experiment

This experiment was held indoors near an entrance of a building as shown in Fig. 3.12 . The robot had chatting capability and an engineer designed the chatting flow. Observations were captured using two external Kinect V2 cameras. Using our model, the robot either responded to people to start the chatting flow (R_r) or encouraged people to chat by calling ”hey” (R_p). When no one was nearby, the robot stayed still (R_n). The robot also tried detecting the end of an interaction using our probabilities. However, the flow was also programmed so that when the phrase ”bye” was detected, the interaction would forcefully finish. Also, when a person was lost for a few seconds, this also triggered an end of an interaction. We collected a total of 14 interactions in one day (including one where a robot had mistaken a person passing by as requesting an interaction).

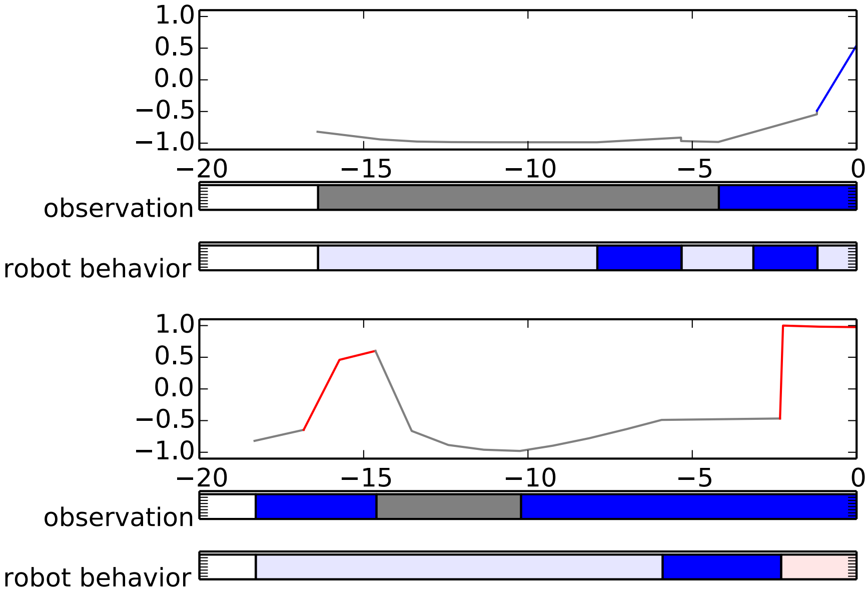


Fig 3.11: Different situation estimation from different human behavior. The two graph show the estimated posterior probability of the human state (-1.0 not willing, 1.0 willing). Gray, blue, red line denote H_n, H_r, H_p. Blue blocks in observation indicate looking toward, gray blocks away. Blue blocks in robot behavior indicate R_{p1}, light blue R_{p0} (moving toward human), and pink R_{r0}. In the first graph, the person responds after the robot speech thus H_r is estimated. In the second graph, the person was looking before the robot speech thus H_p. [119]

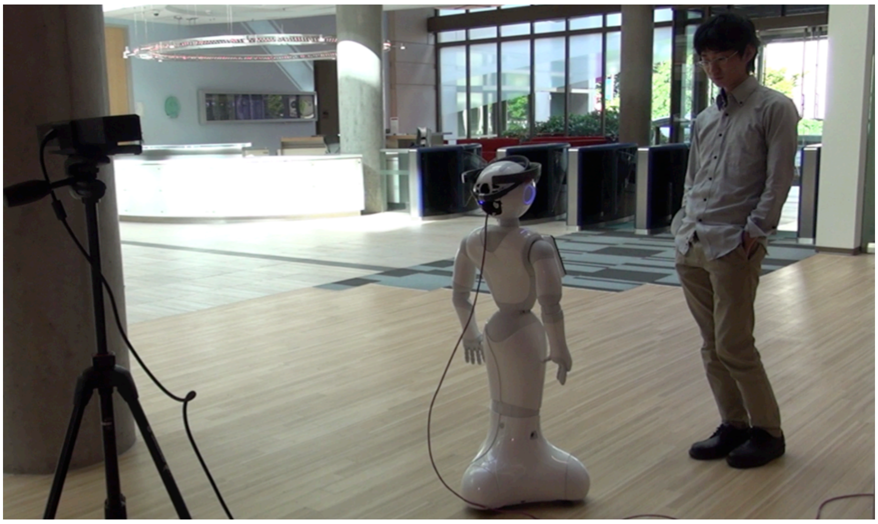


Fig 3.12: Picture image of the public domain social context only experiment.

As the appearance of the robot looked interactive, in 10 interactions people were H_p, and in three interactions people quickly responded with a ”no” and went away. People were looking toward the robot (including those with ”no” responses) and therefore, the robot reasoned all situations as H_p-R_r and agreed to interact. In most of the interactions, the person walked away at the end of the interaction or said ”bye.” These cues triggered the end of an interaction before any detection using probabilities. However, in one interaction, two people started talking with each other and the robot detected this as an end of a conversation (H_n).

Public Domain with Physical Context Experiment

This experiment was held at a robot exhibition. The robot demonstrated its task capabilities of picking objects from a shelf. Group of people walked by to see what the robot was doing. The robot interacted optionally when it was not picking objects and —after greetings to confirm the situation—explained the demonstration to any audience that seemed H_p or H_r and asked whether the robot should pick something from the shelf. A low-wall barrier between human and robot was set for safety. A rotating Kinect V2 camera on the robot torso always faced the audience and captured the situation. We collected a total of 56 interactions in one day.

The task had five phases as shown in Fig. 3.13 . 1) The robot was at start position (between the shelf and audience) and optionally moved close to the audience, then interacted (R_p) depending on the situation, 2) the robot went to pick an object, 3) the robot turned back to show what it had picked to the audience, 4) the robot placed the object back to the shelf, 5) the robot went back to start position. Pictures of the experiment are shown in Fig. 3.13 .

As the exhibition setting attracted people, most of the time, people were looking toward the robot. The looking caused similar H_p or H_n estimations as in the interaction-only experiment. The robot succeeded in agreeing with 39 interactions (including agreeing not-to-interact with non-interested people walking away), leading to an F1 score of 0.821 (precision 0.813, recall 0.830). However, we also observed failures in situation understanding. In four interactions, the audience requested what the robot should pick from the shelf while the robot was at phase 3 or 4. This was observed from people who copied interaction of other audiences. To the audience this situation was $H_p - R_r$ while to the robot the situation was $H_p - R_n$. The robot's behavior had not looked R_n, and thus, caused this difference in understanding of the situation. Another four interaction

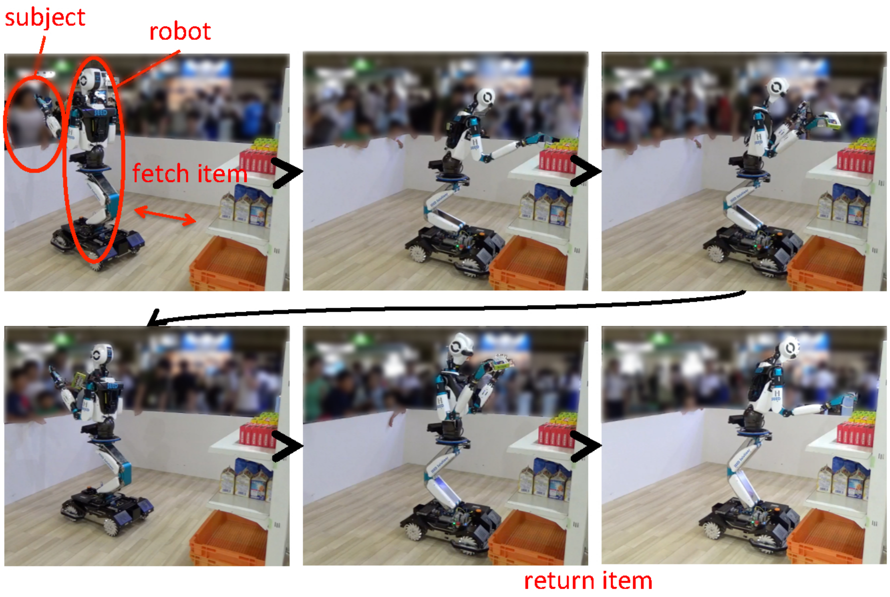


Fig 3.13: Flow of the public domain with physical context experiment. The robot finds an engaged user then fetches a requested item from the shelf, comes back, and shows the requested item. After showing the item, the robot returns the item to the shelf.

failed, as head directions were not representing actual gaze directions and the robot thought H_n as H_p. Other failures included people who could not understand the language of the robot and were confused.

3.6 Discussion

pros and cons of robot speaking behavior—While a speech helps to gain information on an interaction situation, and also helps getting people to interact, we also found drawbacks. As a speech would change a person's goal and interaction situation, the situation becomes more uncertain. For example, when a person is looking toward and a robot reactively speaks "hi," the information *person is looking toward* could indicate either a situation H_p-R_r or H_r-R_r. No significant probabilistic difference in observation was found for the two situations. In contrast, when a robot does not speak, it is unlikely that a human's goal will change. Therefore, the human's willingness is either H_n or H_p. The two has a significant probabilistic difference in observation. Perhaps one of the learning from our discussion on conflict and agreement are that, an interaction could unintendedly happen. An unintended interaction could confuse both the human and robot. It may be better for the robot to not immediately say ”hi,” but wait a while, be skeptical, and be sure of the situation before responding with greetings.

situation of surprised latency—We have assumed that there are no latency between robot behavior and the influence on human’s goal. For the most part, the assumption held true. In the recordings, people reacted quickly from the robot speech and answered ”no”. They quickly responded to a robot’s request including utterances such as ”uh....” However, we also observed rare cases where this assumption may not be true. In an accidental finish found in the recordings, people were surprised when the robot suddenly ended the interaction. In such situations, it took people some time before they could react to the robot’s behavior. Perhaps the worst type of interaction is this type of situation where even the human is not able to understand the situation. When an interaction is not understandable to both the human and robot, the robot will no longer be capable of reasoning the situation from probabilities.

agreeing and engagement—In the task scenarios, the robot was able to reason conflict and agreement situations —including H_p-R_p—in different scenarios using the same probability model. This was possible by using sequence of human head observations and robot behavior information instead of person location. The model was able to handle both people approaching and people in place (robot approaching). Our results indicate that, especially for public domains, agreeing was mostly the same as understanding engagement. The in-lab experiment, however, indicates how understanding purpose-crossing situation might benefit in non-public domains where people behave in various timing.

fire and forgetting interaction—In the interaction-only scenario, the engineer was concerned about detecting the end of an interaction. However, in the task scenarios, the engineers did not care much about the end. Rather, the engineers preferred a timeout when the robot was not receiving a command or an answer. In scenarios where the robot was doing a task, the robot ended the interaction by start going for a pick, or after finishing a handover. Instead of reasoning the end of an interaction for a robot under physical context, perhaps we could begin (fire) but purposefully end (forget) an interaction. The advantage of this strategy is that, if the end were purposeful, it would not be accidental (to the person, why the robot stopped interacting is understandable).

controlling expectation—In the interaction-only experiment, some people asked to play the game of rock, paper, and scissors. Other people asked to have a walk with the robot. The requested physical skill was diverse. In contrast, in the task experiment, people asked to fetch an object from the shelf. It was clear to the audience what to expect of the robot. Results show that the task context provides more control over user expectations.

3.7 Conclusions

Intuitively, a robot's speaking behavior should have an effect on the person's willingness, and whether the person was looking after a robot's action should provide a valuable hint on a person's willingness. We found from our analysis that our intuition was true. The probability of a person's looking behavior and willingness changes depending on the robot's speaking behavior. The probability of a person looking toward the robot after a robot's response is significantly higher when the person is willing to interact with the robot. Therefore, modeling robot action provides more information of the situation when compared to only using human observations.

From a technical viewpoint, using a probabilistic model allows us to directly implement the statistical results found about interpersonal situations. We have found that the implementation was successful from several experiments. The in-lab experiment has shown that the robot was able to distinguish whether the person was looking at the robot or responded to the robot using the probabilities. The public domain experiment showed that the robot was able to successfully distinguish between people who wanted to interact with the robot and people who did not with a 70% success rate and an F1 score of 0.821.

From an HRI viewpoint, we have found patterns on how people behave in the total setting. First, a person close to a robot does not necessarily indicate that the person wants to use the robot. This indicates that understanding a person's willingness and having the robot to only respond when it scents an interaction, is a very important function. Second, a robot with a purpose (physical context) will lead to a much smoother social interaction than a robot with only a social context. It is easier for the robot to have control over the conversation flow as what the person will request is bounded, and the robot is more likely to fulfill the person's desire.

4

Task Execution Systems for Acceptance in Society

4.1 Introduction

In order to evaluate the real possibilities of robots, we must get away from game-like settings we often see in HRI research. In this chapter, we try to capture the essence of required physical robotic skills in our society. We approach the problem from multiple perspectives. The first perspective comes from a survey with a vendor. We will look at some of the business-to-business requests that the vendor has faced before. The requests provide us an actual image of what is already being asked in business today. The second perspective comes from an investigation on multiple robot competitions. Robot competitions provide a small-scaled version of actual tasks asked by the enterprise. Participating in these competitions provide us a more concrete image of required physical settings and possible skills in near-future business of robotics. The third perspective comes from what type of intelligence and automation is being researched or is still a challenge in current robotics. The stories provide us hints on what type of business would be difficult for robotics. By the end of the chapter, we will summarize the found elements from the different perspectives. (The second perspective in this chapter covers the topics written in our journal paper [121]. In this book we discuss the lessons in relation to other perspectives.)

4.2 Perspectives from Vendor

According to the vendor we surveyed, the most often requests are automation in information centers and teleoperation devices toward entertainment (however, this may be due to how the vendor commercialized its robot product). While the automation at an information center is a job of a social robot, what was asked was a human-shaped barebone stack of motors. Social skills were not requested on the robot but requested on the backyard system. Teleoperation is another field that requests barebone robot devices, and there is no real need for robotic skills, except actuating the motors.

In contrast, a more robotic skill that seems to be catching attention is navigation. Although navigation is requested from the enterprise, they are more of an experimental skill the enterprise is trying to investigate on. Such skills have their needs for object transportation in the industry, or for dismantling in datacenters. It is also interesting to note that, these requests also require some sort of manipulation skill of handling objects. However, precise manipulation in the open world is still a challenge and some may prefer an automated infrastructure solution for preciseness. A different vendor [23] points out that, although not having a manipulation skill, navigation skill may still provide value to object transportation. Beside combination with navigation, manipulation alone was not requested except from research institutes. This may be due to the fact that the vendor was selling a human shaped robot or a mobile base, and those who need manipulation-only skills go for industrial arms.

To this point, we do not know the general hardware qualities that are expected in our society. Navigation is preferred, reliable barebone hardware is preferred for interaction, but these are tied to specific settings and usage of robots. Manipulation is an uncertain field that is being hesitated. So far, there is no one design that is accepted for the different settings, perhaps due to its cost. One of the problems of these requests is that, they are framed to how the enterprise envisions robots from current state-of-the-art (or at least what is believed to be state-of-the-art in our society). The requests demand solutions as soon as possible, and do not challenge the long-term boundaries of robotics. Therefore, we must also look at what would be requested of robotics possibly in the near future.

4.3 Perspectives from Robot Challenges

Robotic competitions try to tackle a specific problem but an existing problem that may be handled by robotics in the near future. The Darpa Robotics Challenge (DRC) held in 2015 targeted a disaster response setting. The Amazon Picking Challenge (APC) held in 2016 targeted a warehouse setting. The Tomato Robot Challenge (TRC) held in 2015 targeted an agricultural setting. The Future Convenience Store Challenge (FCSC) held in 2017 targeted a daily life shop setting. These challenges were organized by enterprise or government, or sponsored by a specific company. Although we may not be able to capture the whole picture of what is required in our society with just four competitions, we will be able to capture some of the common essence that will draw possible conclusions on hardware and software skills/designs that will benefit the society. The approach is bottom-up, which is different from the usual top-down approach when designing robot hardware and systems. Note that although we are aware of other competitions, we will exclude some competitions from our discussion for specific reasons. For example, we will exclude

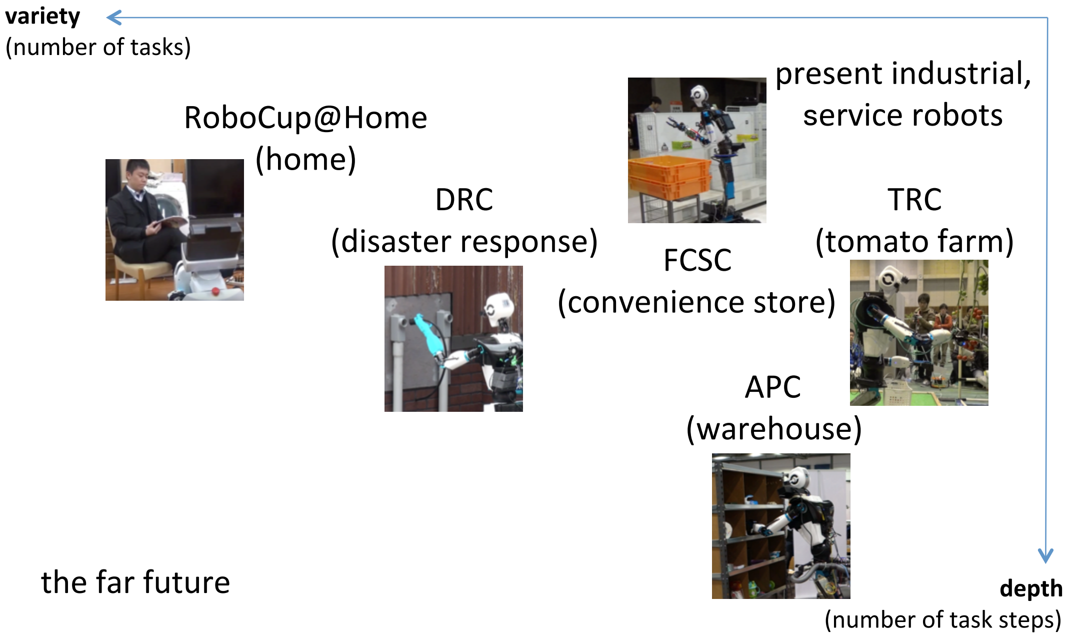


Fig 4.1: Mapping of different problem settings at competitions.

the RoboCup@Home [149] as the competition does not target a real problem proposed by society. The rules are strictly tied to skill benchmarks rather than to actual settings. We also exclude competitions that focus more on developing standard platforms for a specific problem (usually virtual competitions, e.g. the Space Robotics Challenge). These competitions are framed toward a specific design solution and the discussions would be too platform-specific. We will also exclude discussions on competitions that focus on multiple robots e.g. the MBZIRC. Although multiple robots could be one way of entering society, a full discussion on multiple robots would be out of the scope of this book.

Mapping the different competition settings from its characteristic in required skill (task) variety and depth, we find that some problems require handling of more variety while others require handling more depth of the skill (Fig. 4.1). The variety shows the number of tasks (e.g. navigate to a room, pick and place an object, answer user questions, etc.) a single system must handle. The depth shows the number of steps in that task (e.g. number of actions required in the task, number of objects) the system must handle. As a comparison, the present practical systems we see in industrial settings mostly fall into the right hand corner where both the number of tasks and the depth of the task is limited. (The plot in the figure may slightly differ depending on how we define a task as a single task or skill. Here, we will define two skills as different tasks when the robot changes its workspace or when there is a change in the task problem setting.)

4.3.1 The Darpa Robotics Challenge Finals (DRC)

The competition was held in 2015 in the United States. There were eight continuous tasks that were held outdoors under wind and sunlight including: driving, egress, door opening, turning a valve, cutting a wall, traversing debris/walking over terrain, climbing stairs, and a surprise task (operating a switch/plugging socket). The eight tasks had to be completed within a one-hour time limit. Teleoperation was allowed and the robots were semi-autonomous. However, the operators were far away from the robot and only had feedback from the robot's sensors. When compared to the other challenges, the unique part of the challenge was that, the field condition was the toughest and required locomotion and mobility challenges. Power and durability was especially required for the operations.

The disaster setting required the robot body size to be compact to pass a door but have power to open the door and be durable at the same time. The robot had to be designed so that it was durable against fall downs. Falls happened as the robot had to climb stairs and move on a slopped ground condition (or even thick dirt if the driving task was skipped). In the competitions, the three main approaches in robot design was 1) transform the robot body (e.g. RoboSimian [47]) or 2) to use a biped humanoid robot close to human size (e.g. nedo-jsk [62]) or 3) to create a lightweight compact robot with power (e.g. Momaro [123]). Transformation allows the robot to change its body size. Bipeds are compact but capable of producing power. One problem with bipeds is that, they may not be durable in case of a fall over. Lightweight and power are often tradeoffs. One way to overcome this problem is to use the robot's whole body to produce power instead of only producing power with its arm. However, this strategy may cause unintended overloads to some of the robot's body part during operation. The robot must be durable and have a safety mechanism against overloads. An example solution would be the use of stepper motors (See appendix A.1 for details).

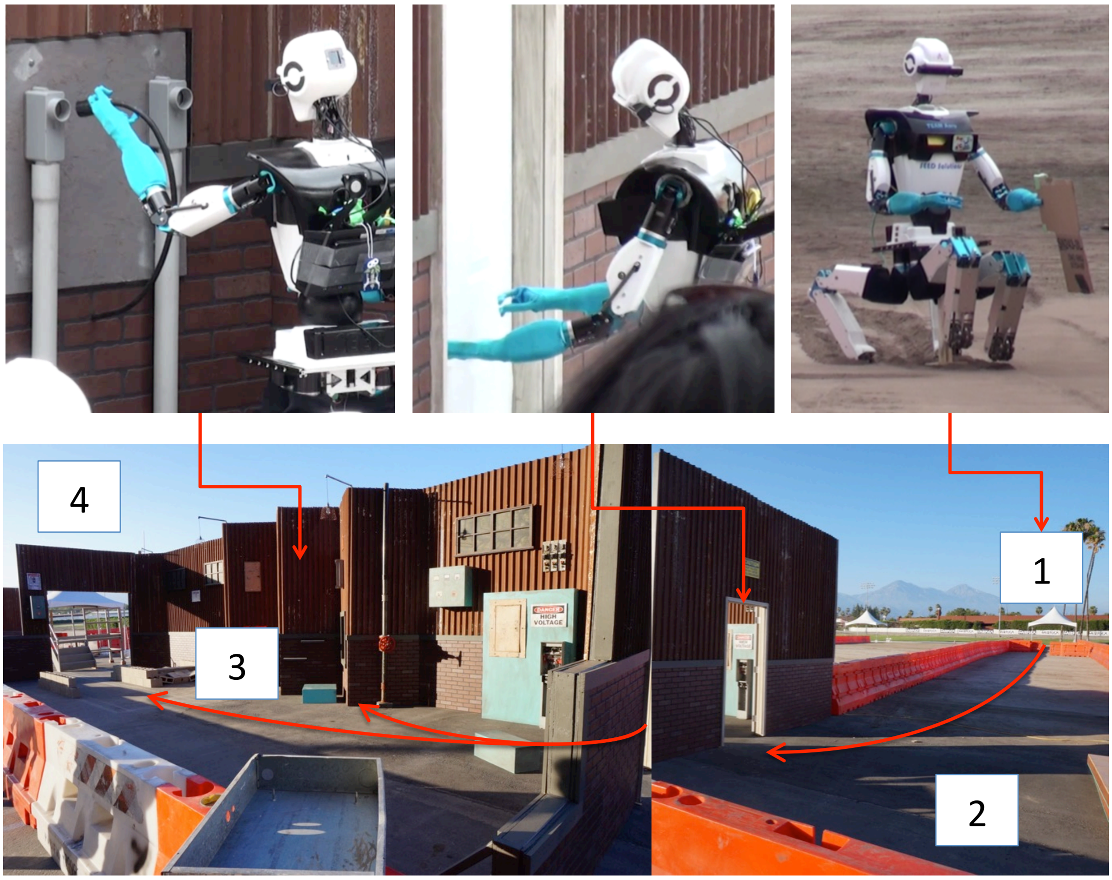


Fig 4.2: The field and flow of the DRC competition. The robot starts from position 1 and then moves by either traversing dirt or by driving a car to position 2. The robot then opens the door and does any of the tasks in area 3. Finally, the robot ends the task by climbing the stairs to position 4.

The disaster setting also required the robot to work in different workspace and heights. The robot had to be compact in one task, but had to reach higher or further in a different task. A lower height was preferred for traversing the debris but required higher height for the valve task. This can be thought as a problem of moving the relative position of the upper body from the base (foot or mobile base) position. We will name a body structure that fulfills this function as a *middle mobility layer*. The function enhances the robot to work in workspaces of different scale. Such a function can be expressed as the following equation:

$$(\boldsymbol{p_{root}}, R_{root}) = Mobility_{l_1,...,l_n}(\boldsymbol{q_{mobility}}) \tag{4.1}$$

where $l_i(i = 1, ..., n)$ are the link parameters of the middle mobility layer, $\boldsymbol{p_{root}} \in \mathbb{R}^3$ is the position of the upper body root joint in the robot base coordinate, R_{root} is the orientation of the upper body root joint in the robot base coordinate which is fixed, $\boldsymbol{q_{mobility}}$ the joint angles of the middle mobility layer. Fig. 4.3 provides an actual example of a middle mobility layer using parallel links. The parallel link structure fulfills the function with the following equation:

$$(\boldsymbol{p_{root}}, R_{root}) = Leg_{l_1,l_2}(\boldsymbol{q_{leg}}) \tag{4.2}$$

where l_1, l_2 are the two link length parameters of the leg, $\boldsymbol{p_{root}} = [p_x \; p_z + z_0] \in \mathbb{R}^2$ where z_0 is an offset from the base, $\boldsymbol{q_{leg}} = [q_a \; q_b] \in \mathbb{R}^2$ the joint angles of the leg.

A common topic regarding robot design is the sensor placement and the number of required sensors. In the disaster setting, a human operator had to operate the robot from a remote location. To this extent, a camera mounted on the head was suitable. The head is the highest point on the robot that is able to gain the most field of view (if the ”eyes” were on the body, it would gain less view). In addition, the head is isolated from the other body parts and can be in action throughout the whole task (if the ”eyes” were on the hands, it cannot be in action if the arm is in action). Beside the head sensor, in the disaster setting, the robot had to traverse over deep sand and it was essential to have a camera in the middle of the four crotch to see whether any of the legs were stuck in the sand or not. When the operator detected that the robot was stuck in the sand, the operator had moved the legs to ”walk” out of the sand hole (Fig. 4.4).

For all the tasks in the competition, the size of the manipulating objects was known beforehand. Intuitively, using a model-based approach (pre-calculated according to the structure of the target

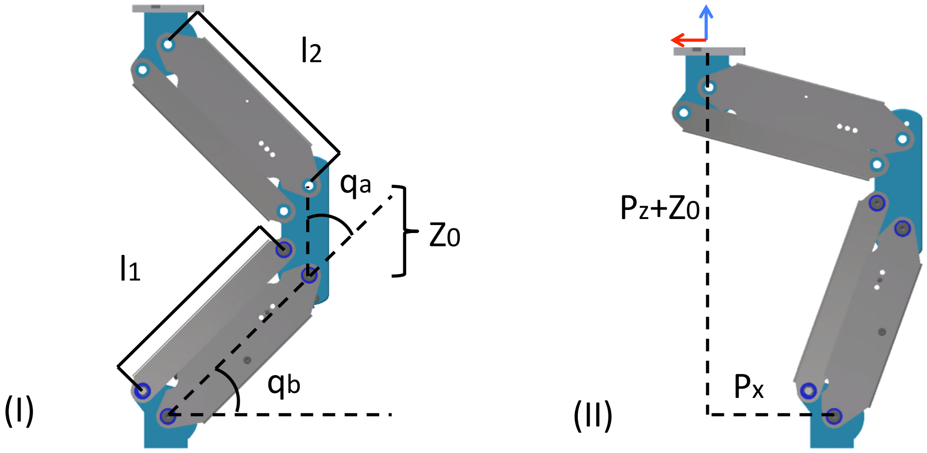
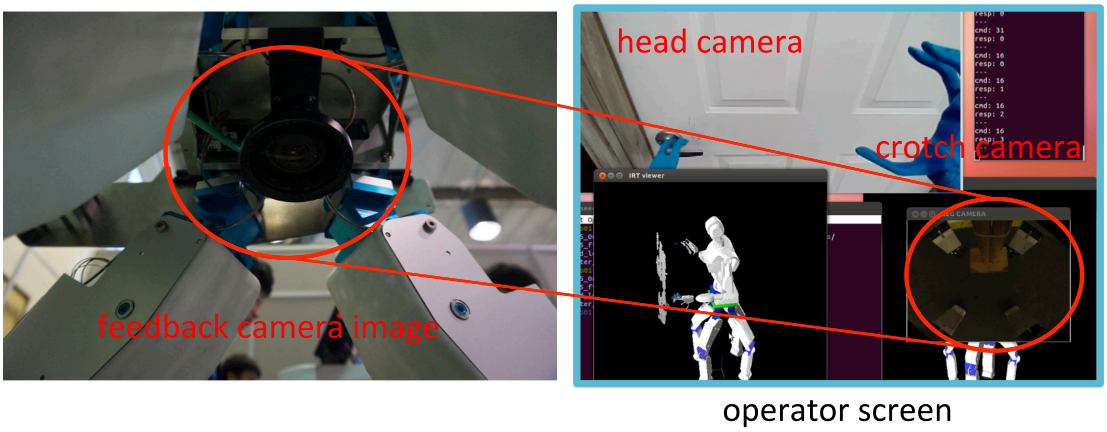


Fig 4.3: An example of a middle mobility lifter on the Seednoid platform. (I) The parallel link structure. (II) An example motion. [121]



```
resp: 0
---
cmd: 31
resp: 0
---
cmd: 16
resp: 0
---
cmd: 16
resp: 1
---
cmd: 16
resp: 2
---
cmd: 16
resp: 3
---
```

Fig 4.4: Crotch camera for the operator to decide whether to use the wheels or walk out of the sand was necessary.

model and the model motion dynamics) will allow us to apply all of what we know about the problem setting. However, for a robot that lack sensors and are only capable of open loop motions, a motion approximation was more practical then using a full model based motion approach. For example, let us look at the door-opening task that consists of turning a knob and then pushing the door. An end-effector trajectory following the turning of the knob and an end-effector trajectory following the movement of the opening door could be designed from the model knowledge. Unfortunately, these type of precise trajectory requires the real robot and environment to match the model state when beginning the motion. A small error between the model and real will lead to a mismatch in trajectory, therefore, not turning the knob enough to open the door (the hand could get stuck in the middle of the trajectory as the expected physical conditions may not be matching with the real motion). There are slight errors from vision, and these errors are hard to adjust by a remote operator as the operator may only see in a hand-occluded and limited camera view. In contrast, the door-opening problem could be solved using an approximate motion of vertically pushing down the knob (perhaps move the hand a little bit inward as the hand goes down) and then push the door by moving the robot body forward. The point is that we are not solving any orientation of the end-effector nor are we following the actual trajectory of the object motion dynamics. This removes the assumed model-based motion constraints. We have a less chance of getting stuck during a motion as we are not relying on the state of the model but choosing a motion that most likely works (pushing down the knob will always turn the knob despite the state of the knob). Fig. 4.2 shows how the approximation approach worked at the competition finals. Similarly, many teams have approached the valve task by rotating the center of the valve instead of actually holding and rotating the handle part of the valve. We find that a model-based approach is not always used for generating manipulation motions in practical settings.

4.3.2 The Tomato Robot Challenge (TRC)

The competition was held in 2015, Japan. There was one task: to pick as many valid tomatoes from real tomato branches and place them in a cage (attaching a small cage to the robot was allowed). Valid tomatoes were pure red, while the non-valid tomatoes were green or not as red. Experts judged whether the tomatoes were valid. Robot operation was either semi-autonomous or fully autonomous, had the option to either move toward to the tomato branches on a flat ground, or

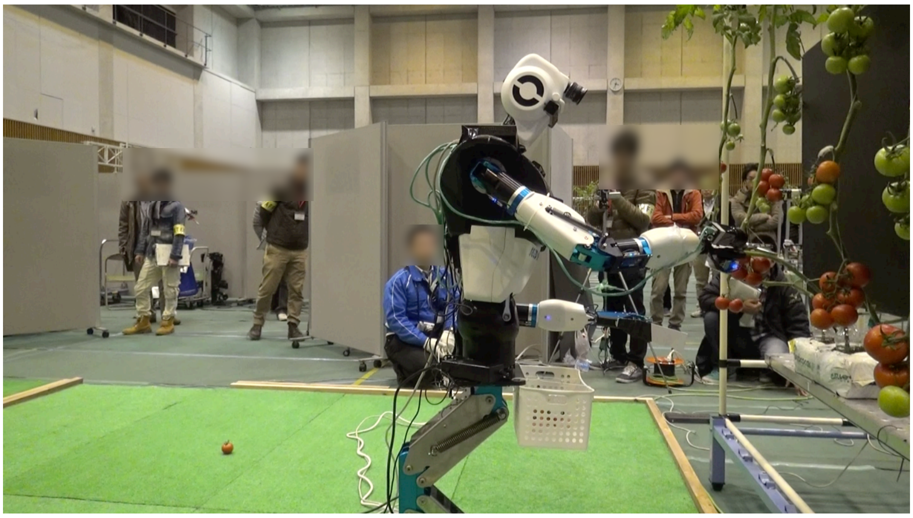


Fig 4.5: Picture of the TRC.

to use linear guide assistance for mobility. Damaging tomatoes were not allowed. 10 minutes time limit was provided. Unlike the other challenges, the field was not an artificially made environment and the environment gained uncertainty through physical contact.

The tomato task required a specialized end effector design: the end effector had to be small enough to maneuver between tomato branches, and cutting the tomato required handling of swinging branches. One approach to the problem was to use a scissor end-effector that holds the cut tomatoes, while other approaches were to use end-effectors that pull the tomatoes or spin the tomatoes. Feature based approaches were possible for detecting the tomatoes but required segmentation in occluded branches. In terms of body structure, the requirements were not as intense compared to the disaster setting.

There were many similarities with the tomato task and the disaster response task. The tomatoes hung vertically requiring manipulating on a ”wall”, the operators only had feedback from cameras, an additional camera was required for the operator to presume the task. Yet, there were slight differences on where the feedback camera should be placed. The task required precision and the operator had to see whether the scissors were close enough to the tomatoes to cut the tomatoes (Fig. 4.6). As the tomato swung while putting the scissors in the tomato branches, it was necessary for the operator to get feedback of what was happening. An extra hand camera for the disaster

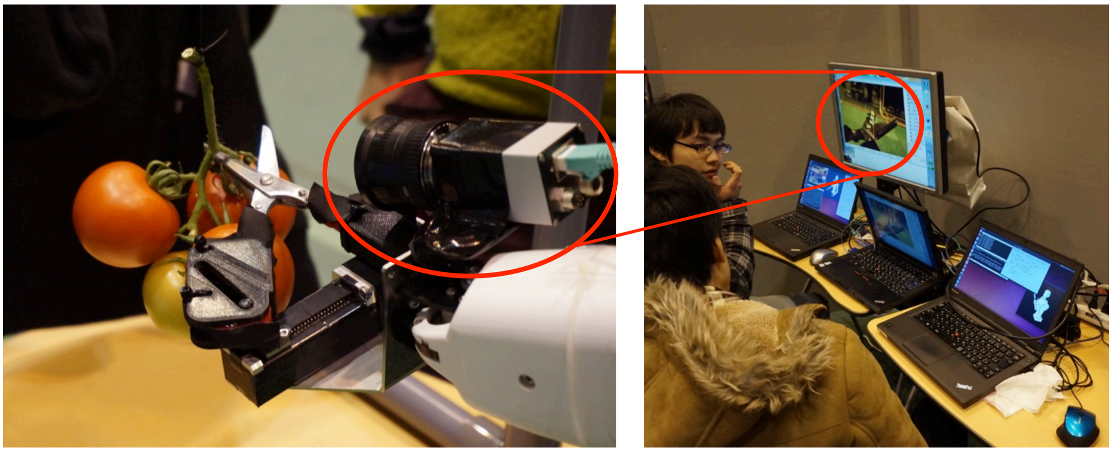


Fig 4.6: Hand camera for the operator to decide whether to cut the tomatoes or whether the hand must get closer to the tomatoes.

setting would have bothered many of the disaster response tasks, however, for the tomato setting, as long as the camera did not get in the way of the scissors, there was no problem attaching a hand camera.

One comparative lesson we may have is the difference between an arm manipulator (Fig. 4.7) on a mobile platform, compared with a human-shaped (semi-humanoid) platform. All conditions except the upper body and control interface is the same between the two: exact same mobile base, middle mobility layer structure, end-effector, and semi-autonomous competing condition. The advantage of the humanoid platform was it had two arms and one arm could be used to hold the tomatoes from hanging away when a scissor tried to cut the tomato. However, the results were that the arm manipulator scored second and the semi-humanoid platform scored third. The dual arm advantage was not well used by the operator, and controlling two arms only resulted to operation complexity. Beside the fact that the dual arm was not beneficial for teleoperation, we believe that the score difference more comes from the control interface (GUI versus a direct joint control device), and there is no real meaning to comparing scores for the particular competition (for a practical solution to the tomato problem, we would want the robot to be autonomous or at least semi-autonomous so that the operator only has to push a button once in a while which was not the case for these direct controlling robots), but we have one fact that a humanoid might be an over

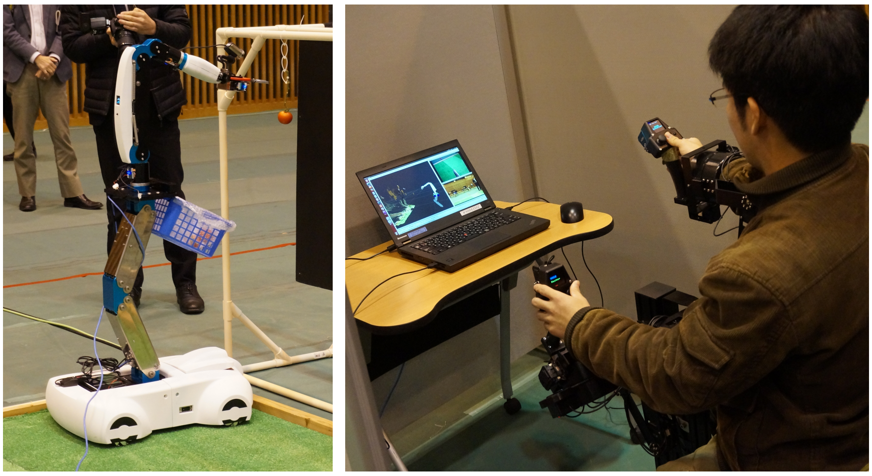


Fig 4.7: Single arm version of the Seednoid platform and the operation device.

specification for the problem. A single arm robot with fewer joints would work well enough to cut the tomatoes.

4.3.3 The Amazon Picking Challenge (APC)

The competition was held in 2016 in Germany. There were two tasks: stow and pick in a warehouse environment. In the stow task, the robot picked up items from a tote and placed them in a shelf bin. In the pick task, the reverse operation was conducted. All 38 objects and geometric features of the tote and shelf were known, however, the position of the shelf was slightly moved by a referee after the robot was placed at start position. The items that were in each bin were given at start time, but how they were placed was not known. There was no restriction in the placement of the tote. Robots were fully autonomous except for a start signal. 15 minutes time limit was provided for each task. Unlike the other challenges, it was not essential for the robot to move its base, but the robot had to operate in bins placed at high positions. The number of items and how they were placed were the toughest among the competitions.

One of the requirements was to scale a robot to a large environment. This lead to tool-attached designs, large designs, or scalable designs using a middle mobility layer. In addition, the task of manipulating in small messy bins required a relatively thin and compact tool axis. For most

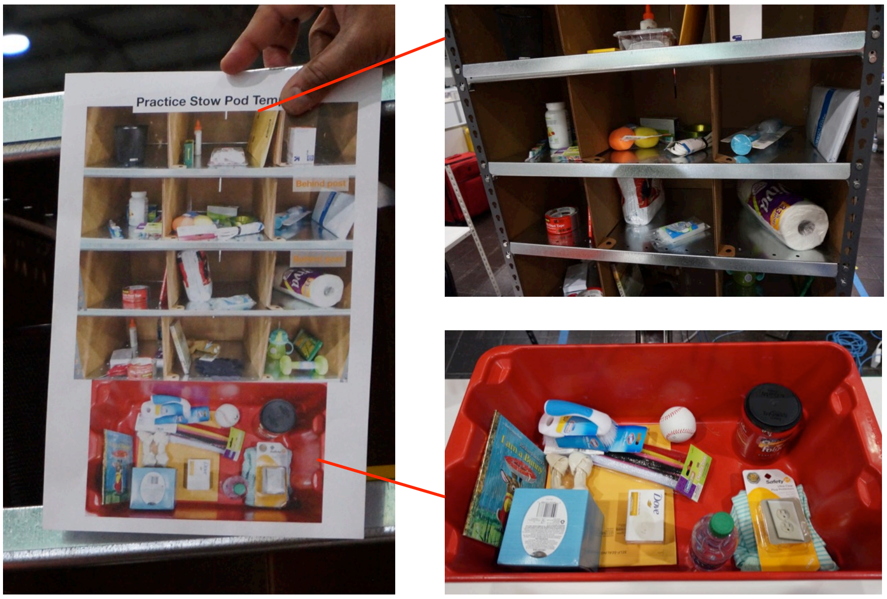


Fig 4.8: Images of the starting conditions in the stow task. Setup instruction panel for the competition staff (left). Actual setup from the instructions (right).

teams, the tough challenge resulted in relaxing the problem as much as possible by using a suction type end effector and removing base mobility of the robot. This has lead to similar designs among teams of using a large industrial robot. On the software side, the task of operating *known* objects has lead to strategies such as learning, reducing uncertainty using pre-computation and/or model based approaches. Picking of different shaped objects required testing of multiple end effector design.

In the warehouse setting, larger objects had to be grasped from the front, and smaller objects from the above. The robot had to grasp various objects, which required different grasp directions. However, all directions had to be conducted under similar restrictions and not under free space like a tabletop condition. Both small and large objects were placed in the same sized bin. Grippers, if used, had to be compact in both grasping conditions. In this sense, a non-centered fingertip design might have suited the problem (Fig. 4.9). (See appendix A.2 for details.)

Regarding motion planning, a pre-defined bin model was necessary to check the collision of the motion. However, checking collision between the end effector and items inside the bin was not always necessary. If taken an approach of picking the most nearest item, we would not have to

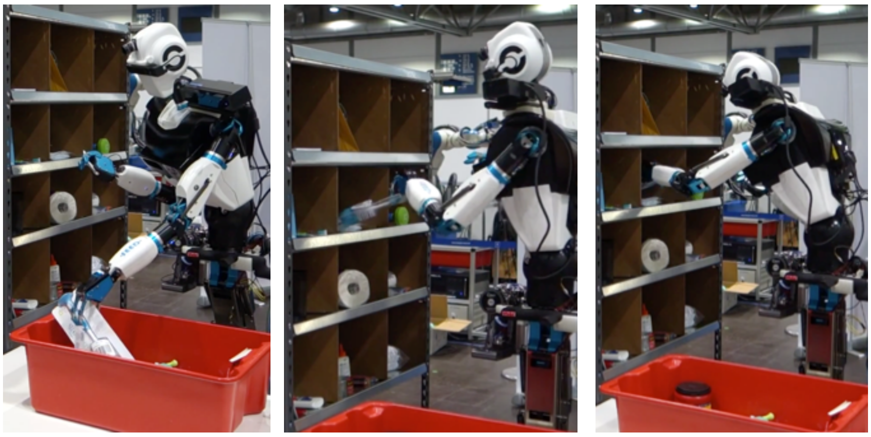


Fig 4.9: Grasping an object with non-centered fingertips at the actual competition.

consider collisions with the other items. Moreover, in some situations items had to be pushed away in order to pick the target item. Therefore, there was not much benefit in checking collision with the bin items. For the motion plans, there was no guarantee that we would have a solution with a complete search-based planner. Some teams have pre-computed possible solutions beforehand [34], while others took inverse kinematics sampling approaches using an initial guess of pre-defined reaching poses.

4.3.4 Future Convenience Store Challenge (FCSC)

The competition was held in 2017 in Japan. There were two tasks: the storing task and the disposal task in a convenience store environment. The robot had to move between two shelves of which each task was conducted. In the storing task, the robot carried a tote to the shelf, picked items from the tote (three rice balls, three cylinder shaped drinks, and three bentos (lunch box)) and placed them on a shelf. In the disposal task, the robot picked five randomly placed sandwiches on a shelf, checked the expiration date of the sandwich (not actual letters but alternative markers), and re-placed the sandwich if it was still edible but collected the sandwich if it had been expired. All items were known beforehand but slightly changed in color and package design at the competitions. There was no restriction in how the items should be placed inside the tote and where/how the tote should be placed. However, there was a restriction in where the items should

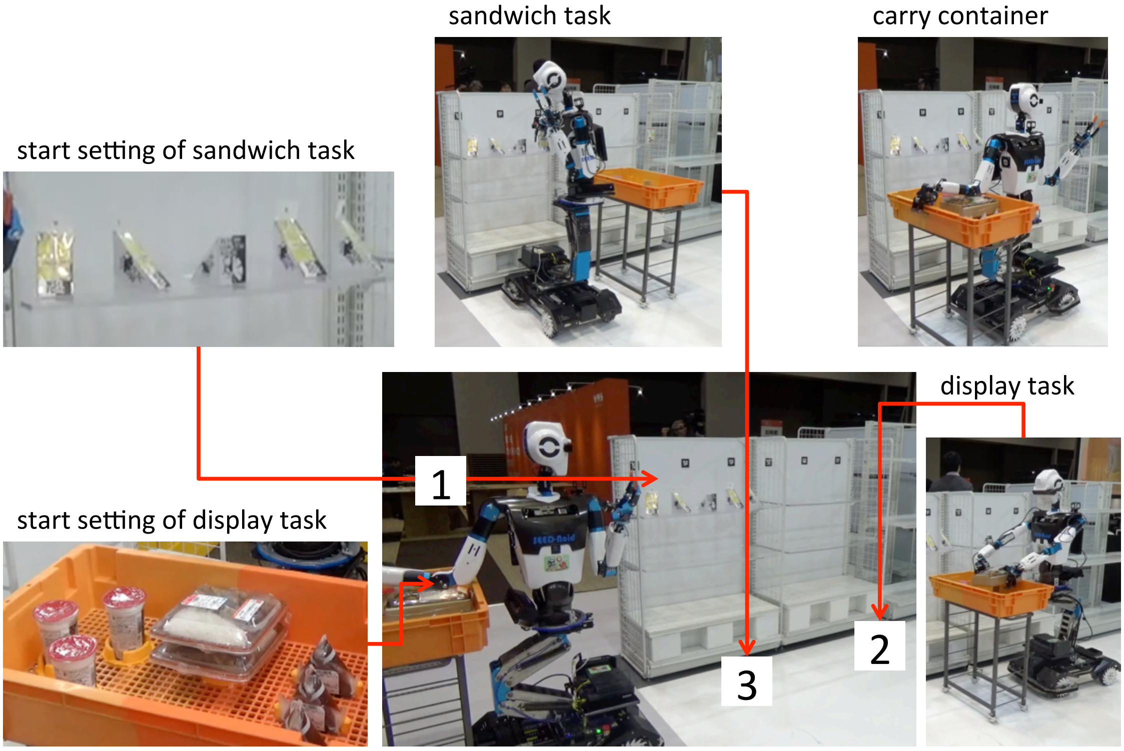


Fig 4.10: The field and flow of the FCSC competition. The robot starts from position 1 and finishes by returning to this position. The robot first goes to position 2 and displays the items in the tote. The robot then goes to position 3 to carry back expired sandwiches and organize the non-expired sandwiches.

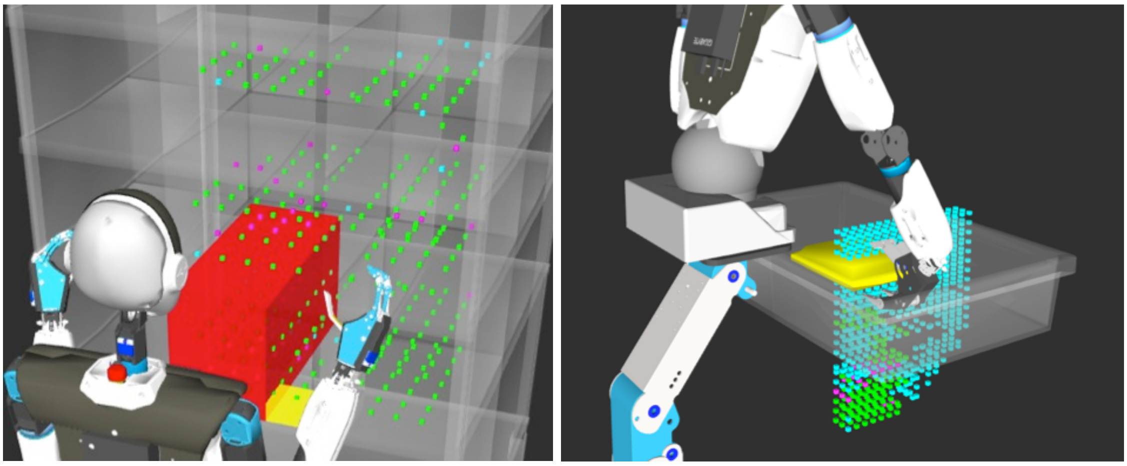


Fig 4.11: Reach solvable positions of the two wrist structures in two tasks. Blue dots indicate positions solved by the industrial structure, purple indicate positions solved by the human structure, green indicate positions solved by both. [121]

be placed on the shelf (e.g. each item should be placed within an area box and on two shelf boards of different height). Five sandwiches in the disposal task were placed and oriented randomly by a referee. Robot operation was fully autonomous and had to finish both tasks within 20 minutes. Infrastructure reformation (e.g. changing shelf height, adding a line trace, but not changing shelf positions) was allowed as long as the reforming and task was completed within the 20 minutes time limit. Unlike the other challenges, placing of objects required neatness and accuracy.

A robot had to move between different work shelves and required accurate mobility. As infrastructure reformation was possible, some teams approached the problem using multiple robots and a robotic shelf. A navigation-based approach used the usual AMCL localization with laser based feedback. However, for confirming correctness of position, visual based feedback approaches using AR markers posted on the shelf were also used in combination. Vision was used as a final adjustment between the robot's current position and desired position. Other approaches reformed the floor by adding a line for tracing. The solution was specific to the simplified competition setting, but would be a problem if the robot had to avoid people in a real convenience store. The difficulty of the sandwich and bento discouraged usage of grippers over suction, leading to similar design approaches as the warehouse setting.

Regarding required manipulation, the convenience store setting was similar to the warehouse setting in that, the robot had to operate in a narrow environment (e.g. a tote filled with items). In these conditions, the end effector pose is crucial. Interestingly, we found that the pose of grasping a bento from a tote was actually more difficult to create than the pose of grasping an item from a warehouse bin. The narrow settings make us think more carefully about appropriate joint arrangements. For example, let us compare a roll-pitch-yaw human wrist structure (B_h) and a roll-pitch-roll PUMA type industrial wrist structure (B_r). A comparative result is shown in Fig. 4.11 . The figure shows the different non-colliding reachable grasp positions of the two wrist structures at a fixed base position (we used the BKPIECE [10, 137] with MoveIt to find the path between a solved goal pose and initial pose before grasping). Blue points indicate item positions that are solved with the B_r structure and the purple indicate positions that are solved with the B_h structure. Green points are positions solved by both. In the bento task, the B_r solves for 822 test points and B_h solves for 244. In the warehouse task, the B_r solves for 262 test points and B_h solves for 279. As there exists collision (the red box in Fig. 4.11), B_h solves for slightly more points in the warehouse task. However, in the warehouse task, the robot is able to move its direction sideways and position itself to reach the non-solved points. In contrast, there is no way (unless the robot is enlarged) for the B_h structure to reach the bento when the tote is at table height. Therefore, B_r has a work range that can generalize to more problems while keeping a compact structure. (See appendix A.4 for a theoretical comparison on the two structures).

The fact that objects could be placed in any way inside the container for the storing task meant that object position and environmental condition was predictable. Motion plans that most likely avoided collisions were pre-computed and therefore did not require any realtime modeling of the environment. Placing the object was more of a navigation problem. As long as the relative position between the robot, the object, and the environment was nearly the same as the pre-computed motion plans, the robot was able to grasp and place the object. In addition, such an approach generalized to grasping the bento that required dual arm picking if used a gripper. As a dual arm pose has an extra constraint between the two arm positions, finding a solution in given time is difficult with a search based motion planning approach. (Strictly speaking, the bento task can be solved relatively easily with a motion planning approach. In case of the bento, the left and right arm has a symmetric pose. Therefore, an approach of solving for one arm and copying the result to the other is possible. However, this is a special case of dual arm manipulation.)

By looking through different competitions, we have a much clearer view of the type of hardware and software required for robots to be successful in the near future. We see that many problem simplification techniques are essential for a promising manipulation and accurate task achievement. However, problem simplification may limit the possibilities of robots. Next, we would like to know whether some of these simplifications would be unnecessary in the future, or whether robots for the next ten or so years will most likely be successful in the simplified scopes.

4.4 Perspectives from Research

Unlike the competition settings where *how to relax a problem* was the main focus of building robot hardware and systems, the field of science tries to handle various uncertainties by either searching over several possibilities [66], or from decision using large amount of trained data [21]. However, the scope of such state-of-the-art approaches is still limited. We are still handling how to reduce the search space to be computationally feasible [148], discussing abstraction [144], and figuring out cost functions [150]. Unfortunately, the approaches only work under limited state and action space, which actually pull us away from real problem settings that are possibly more rigorous in terms of required actions. In the current stage, there seems to be a tradeoff between handling tasks with many procedures and handling tasks under uncertainty. This is not a surprise. As the number of task steps becomes large, so does the search space on uncertainty. Although we may gain situational robustness by handling uncertainty, we see that this will in fact limit the possible scale of the task.

4.5 Discussion

hardware requirements—Although requests to vendors have hesitated in using manipulation skills, it is apparent from competitions that, in the long-term future business of robotics, a manipulation skill is inevitable. Yet, in the current stage of society, manipulation is not seen as reliable as navigation, and we found many difficulties in manipulation through the competition settings. There seems to be a gap between what was believed to be essential for manipulation and what was really missing to solve real problems. Directions in research have gone for handling broader situational uncertainty, while real problems asked for robustness toward long task procedures. To understand our lessons, we must first briefly look over what was believed or known about manipulation.

For the degrees of freedom of manipulation, we often see an upper body structure with 7 degrees-of-freedom (DoF) for each arm (e.g. PR2 [151], HRP [64]). Joint arrangements are similar to the human structure that is a 3-1-3 shoulder-elbow-wrist structure. There are reasons to why 7-DoF manipulators is preferred over six, which can be explained by how each joint is expected to contribute to a task. From the perspective of analytic inverse kinematics, the last three joints (the wrist) of a 7-DOF arm are used to get a desired tool axis [130]. From analysis on the human upper body arm motions, the other four joints are used to position the arm. According to [139], a combination of the shoulder roll, shoulder pitch, and especially the elbow joint is used for a one direction reaching action. In contrast, the shoulder yaw joint is not used when an arm is being reached in one direction. [139] explain that the shoulder yaw is used for inward and outward motions, and interestingly, we use different shoulder yaw values depending on where we place the hand in the height direction. Some motions are easier to achieve using different combinations of the joints. Therefore, a redundant manipulator is expected to solve more type of different reaching problems. Another possible reason why redundant joints are preferable is because we achieve faster motions [96].

Although, such discussion on degrees of freedom is correct, the arrangements of roll, pitch, yaw has not been systematically evaluated, especially under a narrow environment. Most hardware design follows either the human-like structure design or industrial structure design. By observing the actual tasks requested by company-sponsored competitions, an additional lesson about manipulation was that, the industrial structure better suits for settings with collision, which is the case in our society. Surprisingly, there were no tabletop tasks in many of the settings and most were against a wall or against the shelf. Even for tasks that were table view, the operation was in a container. The type of applications required by the challenges proved that being able to work in free space is not enough for manipulation in our lives. (A minor exception would be the convenience store task. Some teams have approached the problem by pulling out the shelf board, leading to a tabletop manipulation task. Yet, the solution must still handle the container constraint.)

Another important gap between what is being researched for task robots and what was really required was that, manipulation is not only about arms. Most settings required operation in workspace of different heights. Sometimes the robot must operate at a lower height, other times the robot must operate at a higher position. It is essential for a robot to consider a middle mobility layer that will adapt to different work heights or reaching length. Although, robots such as the HSR already consider a middle mobile lifter that moves to different heights, it does not consider the reaching length, and depending on the height, the robot's reach length is often limited due to environmental collision. Real problems tackle a narrower environment with more motion constraints and a larger work area.

Regarding end effector designs, we were not able to find answers to an appropriate one design. Grippers had much more difficulty in solving some of the problems when compared to suction based approaches. The problem with grippers is that, the designs usually simplify the characteristics of a human hand. [15] classifies the usage of the human hand during a task. Of the detailed classification, the common prehensile grasp modes that are actually implemented on robots are the fingertip and encompassing grasp [89]. These functions were obviously not enough to handle manipulation in the cluttered situation. It is also obvious that some actions require non-prehensile actions (see appendix A.5 for details). Perhaps a hybrid suction and gripper approach would suit most problems, yet, such solutions would lead to extra costs. Although the HSR has such hybrid structure, the power of the suction cup only work against flat cards. For cost, there requires a decision between suction and gripper. If picking and storing various objects in open space were the main skill required, then a suction solution would be appropriate. If the robot had to open a hatch before picking objects, or if there is a handover between the human and robot, then, a gripper solution might be more suitable (handover timings would be difficult with a suction end effector).

In terms of number of arms, this will also depend on the end effector design or problem domain. A suction approach is usually capable of picking items with a single arm, thus reducing the hardware cost. A gripper approach requires the usage of two arms for picking up the larger items. For some problem domain such as picking tomatoes, two arms increases the complexity of the system with not much difference in performance over a single arm. In a picking scenario, two arms is mostly beneficial for supporting the pick. If the end effector is powerful enough and does not require any support, then, maybe we should reduce the cost with only having one arm. Yet, when there is social context, the answers may be not as clear.

software requirements—In current business requests, sometimes the enterprise does not require a software, or only require minimum software for teleoperation: streaming images from camera and actuating the robot. This was partly true for some of the more teleoperated semi-autonomous competitions.

Object recognition was not a severe demand in current business requests. Perhaps this is because, recognition will not have much meaning without a manipulation or navigation capability. However, we also learn that recognition is essential for most other automated manipulation settings (e.g. disposing a sandwich or picking objects from a bin). In object recognition, there are mainly three types of problem settings. 1) All objects or state of objects are known. 2) Some objects are unknown but we have time to register new objects. 3) Some objects are unknown but we do not have time to register the objects. After attending several competitions, we found that from a practical perspective, situation 3 —although the most general approach and a problem of interest to researchers—may be a rare situation. The warehouse setting in APC 2016 was situation 1. The warehouse setting in APC 2017 and the home setting in RoboCup@Home was situation 2. For situation 2, there are fine-tuning [157] approaches and we may also use recent cloud services such as Microsoft Custom Vision, which only require 15 images and can be trained in seconds. In all situations, the basic approach is to combine a segmentation problem and detection problem (however, there are also end-to-end approaches for situation 1 [110, 76]). In general, the segmentation problem can be seen as a bottom-up attention problem (or saliency problem) to roughly find regions that are interesting but well parted from other regions. The detection problem can be seen as a labeling problem. Both situation 1 and 2 handles these two problems using deep training techniques. In contrast, situation 3 must map semantic knowledge with recognition in the order of seconds without any training. Being able to handle situation 3 requires a more complex software of semantic knowledge management and a larger database (see appendix A.3 for possible approaches). Perhaps situation 3 is not much of an interest to the society, since, we know the job we want the robot to do, and we want the robot to perform its job accurately. If situation 3 is rarely asked, it might be better for robotic systems to focus on situations 1 and 2.

Regarding manipulation, it seems that there are hardware problems we must first over come before going into the software issues. From our discussion, changing the hardware structure has significantly increased the number of possible solutions in a tote manipulation. Yet, once we have the appropriate hardware, difficulty still remains in software. Unlike factory automation, where the task workspace is designed for the robot, robots entering our society have a shared workspace with people. The physical context is much more rigorous and computational solutions struggle against the large search space. For these more difficult problems that require more manipulation steps, heuristic approximated plans do better than computational solutions. Instead of focusing on having the robot learn manipulation from scratch, perhaps it is better to focus on how we automate heuristics for the more complicated scenarios. We must also consider the fact that, manipulation is not just about arms, and problems can be simplified using the mobile base or middle mobility layer. We must balance between heuristics, hardware provided solutions, and automation.

4.6 A Proposed Minimum Task Execution System to Fit Our Society

We summarize the required skills as a system structure shown in Fig. 4.12 . Note that this is more of a minimum operating system (OS) structure that we are proposing, rather than a full system to handle various tasks. Below we will briefly explain each component.

4.6.1 Vision Component

Vision is used in combination with manipulation. Although vision may be used in combination with navigation for data collecting, perhaps from a business perspective, clients would want a navigation-only robot, and then collaborate with a company with a vision-based data-collecting specialty. Unlike manipulation, the navigation and vision scenario does not require a high integration of the two. In manipulation, the main settings are object recognition with a known list of items, or non-realtime registration of items. We have explained that in these settings, some type of recognition model is learned from data and a recognition algorithm (e.g. Ssd [76], mask-RCNN [114], or communicate with cloud services) outputs the detection results.

The output results are then passed to an information extractor (e.g. in the convenience storage task it was finding the nearest item, in the convenience sandwich task it was finding the sandwich in the most right followed by an update of regions-of-interest, in other tasks this could connect to a principal component analysis using point clouds). The information extractor largely depends on

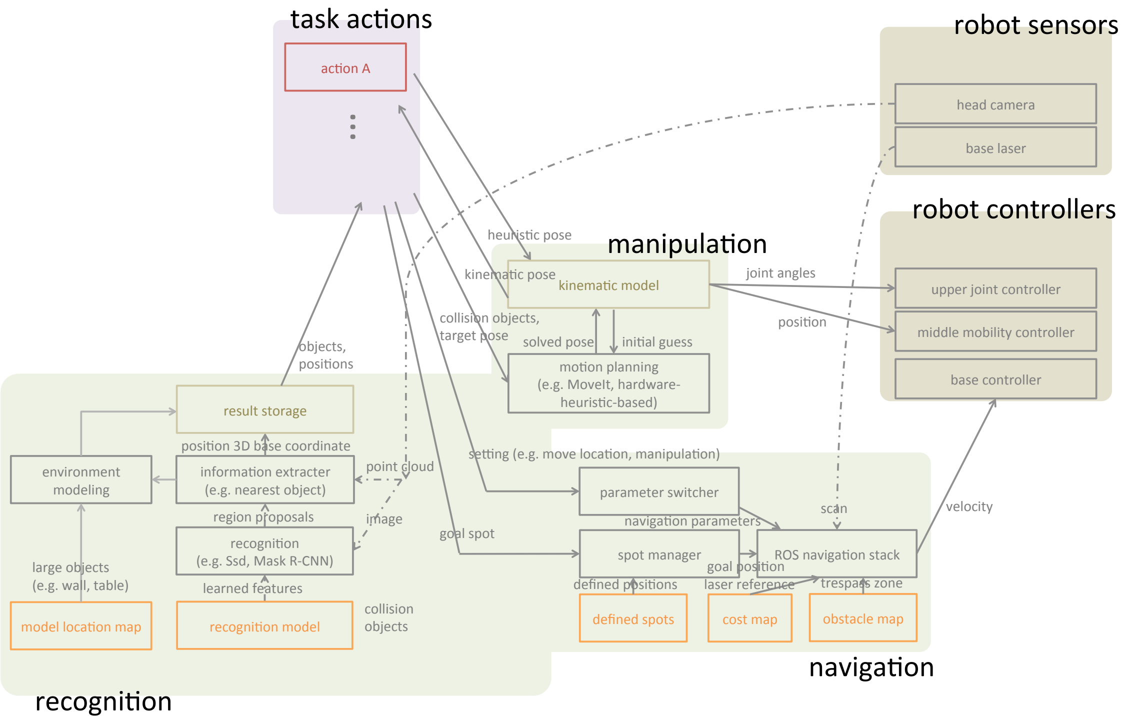


Fig 4.12: The minimum required task execution system from current and near-future needs. In addition to the ROS navigation stack, a parameter switching node is added to use navigation for both mobility and manipulation purposes. For motion planning, initial guesses and hardware (middle mobility layer) based heuristic solutions are used to simplify calculation under constraints. For recognition, online information is used to model small objects and offline information is used to model large objects.

the task whereas the recognition algorithm could be re-used to different tasks. The extracted target is then stored to memory.

Note, that the result does not always directly connect to the motion planning component. It may connect to navigation (location adjustment) before generating manipulating motions. In addition to what has been requested about vision, the vision component has an online modeling module for visualizing the recognition results. This provides feedback to the users, and is an important component at development stage. Although online modeling would also help in calculating collision, in a cluttered situation, the access to the object is limited and there is not much we can do by knowing the surrounding environment of the object. The possible grasping poses are restricted and we may even require push-grasp [28] solutions instead of collision avoidance. (See appendix A.6 for details on the visualization.) Note that the vision component explained is the minimum requirement for robots to do a task in society. It is mostly a *vision for manipulation*. Depending on the task, other task-specific vision components such as checking the expiration date of an object might be necessary. However, to reduce software costs, such components should be an extra application and not included in the basic software (OS of the robot).

4.6.2 Manipulation Component

To enter society, manipulation requires preciseness. However, this is not the preciseness of the motion, but the preciseness of achieving the manipulation task. The most important part is repeatability. Since the type of problem we have seen assumed some knowledge of the problem in a structured or semi-structured environment, we find that open loop strategies are applicable, simpler, and more stable. (Closed-loop feedback based approaches are parameter sensitive. Closed-loop is better for handling uncertainty but not repeatability.)

Most tasks that seem to be of concern is pick, carry, place, and perhaps opening of hatches and drawers. The need for manipulation seems to be related to transportation of objects, which was a request to vendors but what was also required in the warehouse and convenience store setting. To achieve such a pipeline, we propose a general motion generation pipeline as follows: 1) Reach: some pre-defined heuristic pose is applied as an initial guess for the robot, and the planning algorithm generates a kinematic (or navigated) solution that ”fits” the pose in relation to object position. 2) Pick: repeat 1, except under a more strict kinematic constraint of handling

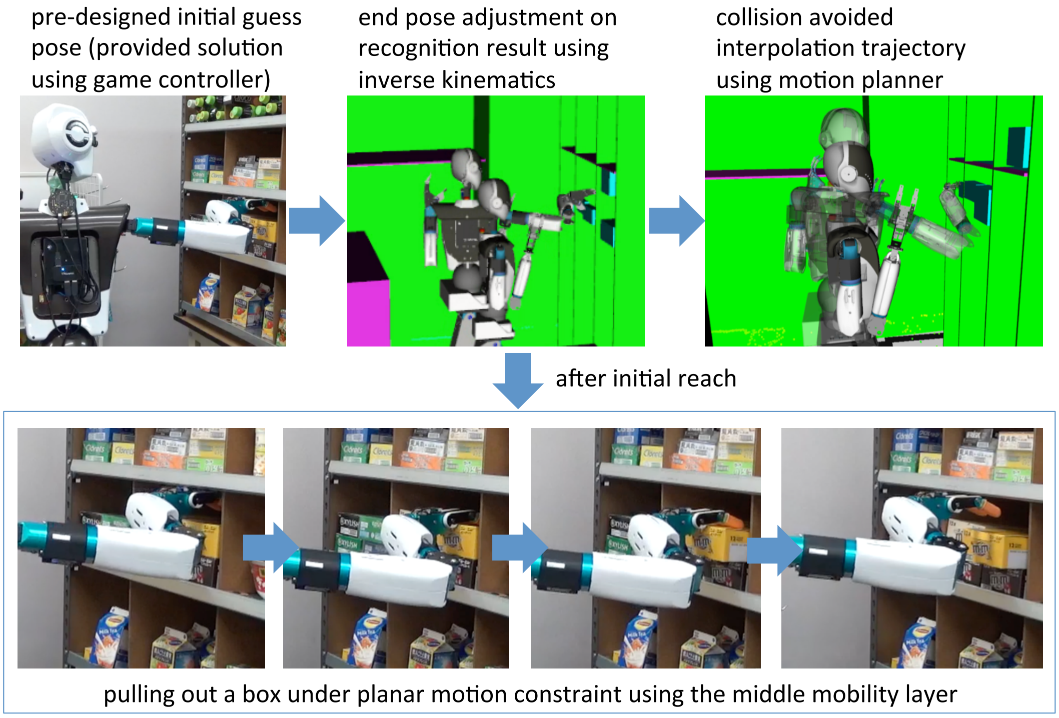

pre-designed initial guess pose (provided solution using game controller)
end pose adjustment on recognition result using inverse kinematics
collision avoided interpolation trajectory using motion planner
after initial reach
pulling out a box under planar motion constraint using the middle mobility layer

Fig 4.13: Example of a manipulation pipeline for a complex picking task. Above row shows the reaching process: a pre-defined pose for an initial guess, solved end pose, and the solved trajectory. Bottom row shows the picking process using the middle mobility layer to solve joint continuity constraints after the initial reach.

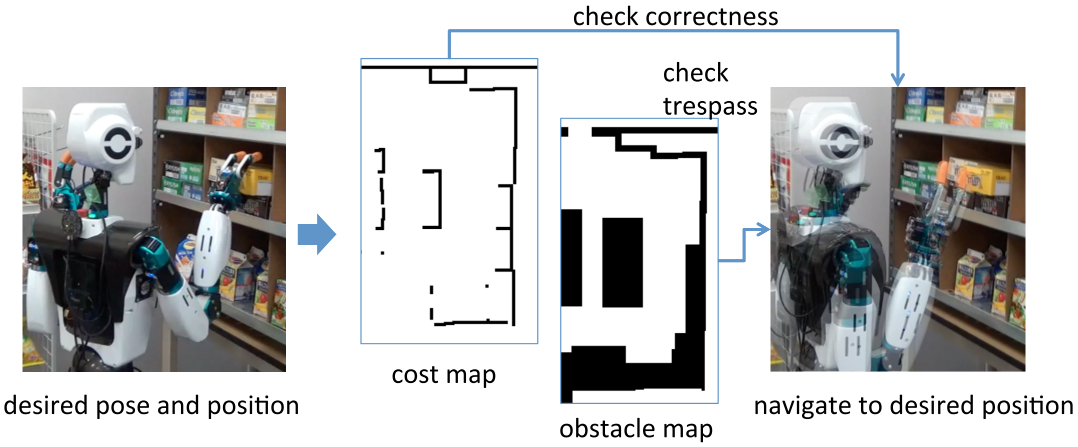


Fig 4.14: Manipulating from a pre-defined pose by navigating to a desired offset position.

object orientation. 3) Place: repeat 2. This basically leads to the modules illustrated in the figure with one algorithm module and storage of poses. Targeted environment in our lives are much more restricted than open table top tasks, and require heuristics to handle computation efficiency or to achieve stable solutions. The difficulty of manipulation is perhaps this applying of heuristics. The picking and placing usually has the following planar constraint on the object in addition to joint continuity:

$$M_o^w(x, z) = M_{o_0}^w Trans(x\boldsymbol{v_1}) Trans(z\boldsymbol{v_2}) \tag{4.3}$$

or in case of a non-prehensile picking:

$$M_o^w(x, z, \theta) = M_{o_0}^w Trans(x\boldsymbol{v_1}) Trans(z\boldsymbol{v_2}) Rot(\boldsymbol{n}, \theta) \tag{4.4}$$

However, if we look carefully at the function of the middle mobility layer, we realize that such constraints can be directly solved using the middle mobility layer. The middle mobility layer keeps the upper body pose (joint continuity, object orientation constraint) but allows the robot to move forward/backward or upward/downward (object movement). Fig. 4.13 shows an example of applying heuristics using the middle mobility layer for solving a picking task of a large item from a bin. Fig. 4.14 shows an example of combining heuristics with navigation.

For more specific tasks such as a tomato task, we would need a more task-specific end-effector solution, but the motion generation pipeline should not be much different from what is being proposed here (the required constraints are usually the same). Even for the disaster response setting, the motion generation could be based on heuristics (as we have already discussed), therefore, following the proposed pipeline. In the default manipulation software, what is required is this pipeline of applying heuristics and connection with the other components.

4.6.3 Navigation Component

Most business requests on navigation are structured settings e.g. inside a building. Although, companies may provide their own navigation algorithms, these own developments add to extra costs. Simple navigation problems can mostly be solved using the open source ROS navigation stack, if chosen the correct navigation planner and parameters. For example, the timed elastic band [115] planner creates a smooth path for forward and rotating directions, and suits for long distance navigation that may require avoiding of people. However, in order to use the planner in navigation-manipulation integrated scenarios, we must change parameters and disable local obstacles as the planner often tries to avoid obstacles using a long path even when what is wanted is the robot to move a few centimeters to the left. Therefore, we add a parameter-switching node that sets the appropriate navigation settings for the different usages. Regarding the common parameters, the AMCL parameters should be set to update every centimeter. The default parameters only update every twenty centimeters leading to a large error in localization especially for robots that navigate in a small workspace (other than the hallway). In addition to parameters, we separate the obstacle map and the localization cost map. The obstacle map defines where a robot should not enter, while the localization cost map is used for estimating the robot's current location (usually created using any kind of laser-based SLAM algorithm). For example, a carpet may be colored black in the obstacle map but white in the localization cost map.

4.7 Experiments using the Proposed System

various picking—Fig. 4.15 shows examples of various picking using the system. The top row in the figure shows a grasping of a 500[ml] bottle from a neatly cluttered shelf. Unlike the APC, the robot is not able to push-grasp the item, and the only available handle is the bottle cap. Trying to pull out the item directly will get the bottle stuck, and the robot must tilt the bottle to get it out. The robot successfully achieves this motion by using the shelf guard as a fulcrum, and adjusting its movement using the middle mobility layer. The success rate is 8 out of 20 trials with slight noise in position, tuned parameters, but without feedback. Failure is mostly due to fingers colliding with bottles on the side when the modeled position differs from the bottle's actual position. The middle and bottom row shows other examples where a middle mobility layer solves the problem. The success rate is 14 out of 20 trials for both situations, slightly better than the bottle as collision on the side was less severe. The results prove that the system is able to easily integrate heuristics to achieve various complex manipulations, and that the actions are repeatable.

evaluation from competitions—Table. 4.1 is a summary on the competition results using our proposed system or a former version of the system. From the results, we find that the system is able to score and scale to different problem settings. Looking more closely to our scores in FCSC (the convenience store challenge), our system accomplished 50% of the task. We have dropped

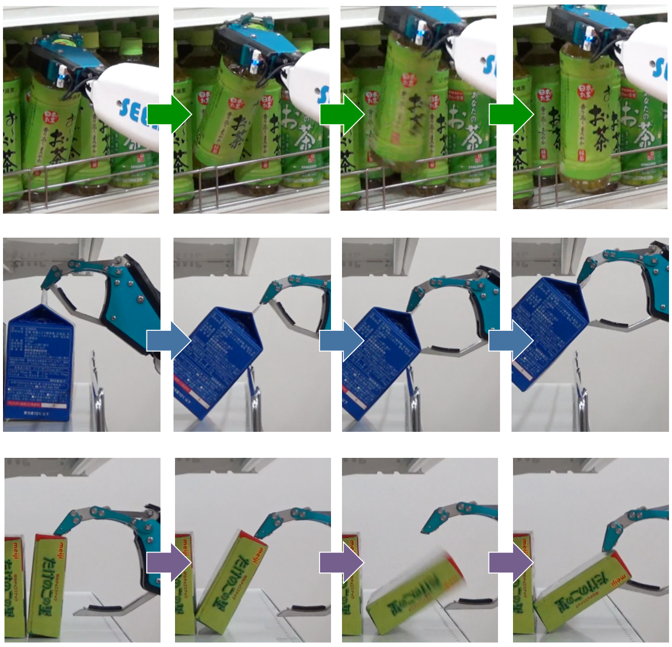


Fig 4.15: Common constraints in picking solved with a combination of the middle mobility layer and grasping modes. [121]

Table 4.1: Actual scores in the competition using the proposed system.

competition	results
DRC	0pt (due to skipping of desert challenge)
TRC	44pt, 3rd/14
APC	pick 16pt, stow 3pt
FCSC	50pt, 2nd/9, Most Challenging Award, Seven Eleven Award

34% of the storing task and 22% of the sandwich task due to the time limit, leading to a total of 28% of the whole task. The other 22% were failures of which 5% came from the storing task and the other 17% from the sandwich task. The failure in the storing task was due to a non-stable grasp from a vision error. We may avoid this type of error by capturing objects more toward the center of field of view, which result to more sensor detection accuracy. The failure of the sandwich task was more due to the fact that we did not prepare appropriate end effectors for the task. The system was not targeted specific for the task, and we have challenged the system's capability by not attaching the tote to the robot (which was the approach for most teams including the winning team). Despite the disadvantage, the system scored second place, did best within the other multi-purpose robots e.g. HSR, and scored 0.476 more points per team member than the winning team.

4.8 Conclusion

The findings from this chapter indicate that manipulation —especially the transporting and delivering of tools or objects—is an important skill requested in society. We find that the challenges of manipulation are its reliability and targeted scope. We have shown that applying heuristics instead of computational approaches increases reliability. We have shown that the approach was sufficient enough to handle various complex picking situations including grasping from a narrow shelf, grasping from a container, and grasping from an organized cluttered environment with 70% success rate in most cases. To apply the heuristics, integrating hardware solutions such as using a middle mobility layer that simplifies the joint continuity constraint in constrained environments was key.

In addition, we were able to accomplish 50% of the convenience store challenge despite the system not being targeted for the specific task, scoring second place out of nine teams, and receiving two awards. The convenience store challenge was balanced in the number of tasks, and the number of procedures in each task. The result shows the generality of the proposed task execution system and its effectiveness especially for task-finite scenarios requested in society (scenarios where the request by a user, the role of the robot, and the scope of the problem, are all defined and controlled to some extent).

5

System Architecture

5.1 Introduction

The purpose of this chapter is to solve the problem of interaction and task integration by building a system (Fig. 5.1) using the situation scenting capability from Chapter 3 and a task scheduler proposed in this chapter. The scheduler is unique in that it not only considers robot-centric physical constraints but also human-centric dialogue objectives. This chapter will begin by explaining this idea with examples, and then explain how to achieve the components required for this idea. In the end of the chapter, we will show how our system achieves a restaurant scenario task, which include a mixture of task skills presented in the previous chapter, but also various interaction situations.

5.2 Interpersonal Situations During a Task

Interpersonal situations happen in parallel to a task. A robot may be picking, placing, navigating, or transporting an object but also reacts and listens to a person. Even when it is the robot that is trying to initiate an interaction, task and interaction happen in parallel. A robot may greet a person as it is approaching the person. The speech timing is dependent on the scented interpersonal situation. Therefore, the situation engine is always running on background, and then triggers a robot behavior at any timing depending on the interaction willingness of the robot, and estimated interpersonal situation. However, the discussion on robot willingness is rather complicated as the discussion involves looking at the physical, social, and objective context of the task.

From Chapter 3, the process of entering (or not entering) an interaction during a task is described as below: assuming that unintended situations are avoided, the human and robot are first at an H_n-R_n agreement situation which at some point may turn to a H_p-R_n or H_n-R_p conflict situation. In this section, we will go over the H_p-R_n conflict situation and its relation to the different context.

5.2.1 Simple Interaction without Context Constraints

When there are no constraints, there is no reason for the robot to not be willing, and therefore, the robot may simply change to R_r to enter an interaction agreement. An example of such a situation is shown in the Chapter 6 guiding robot experiment, where the robot is in idle mode when there are no users willing to use the robot.

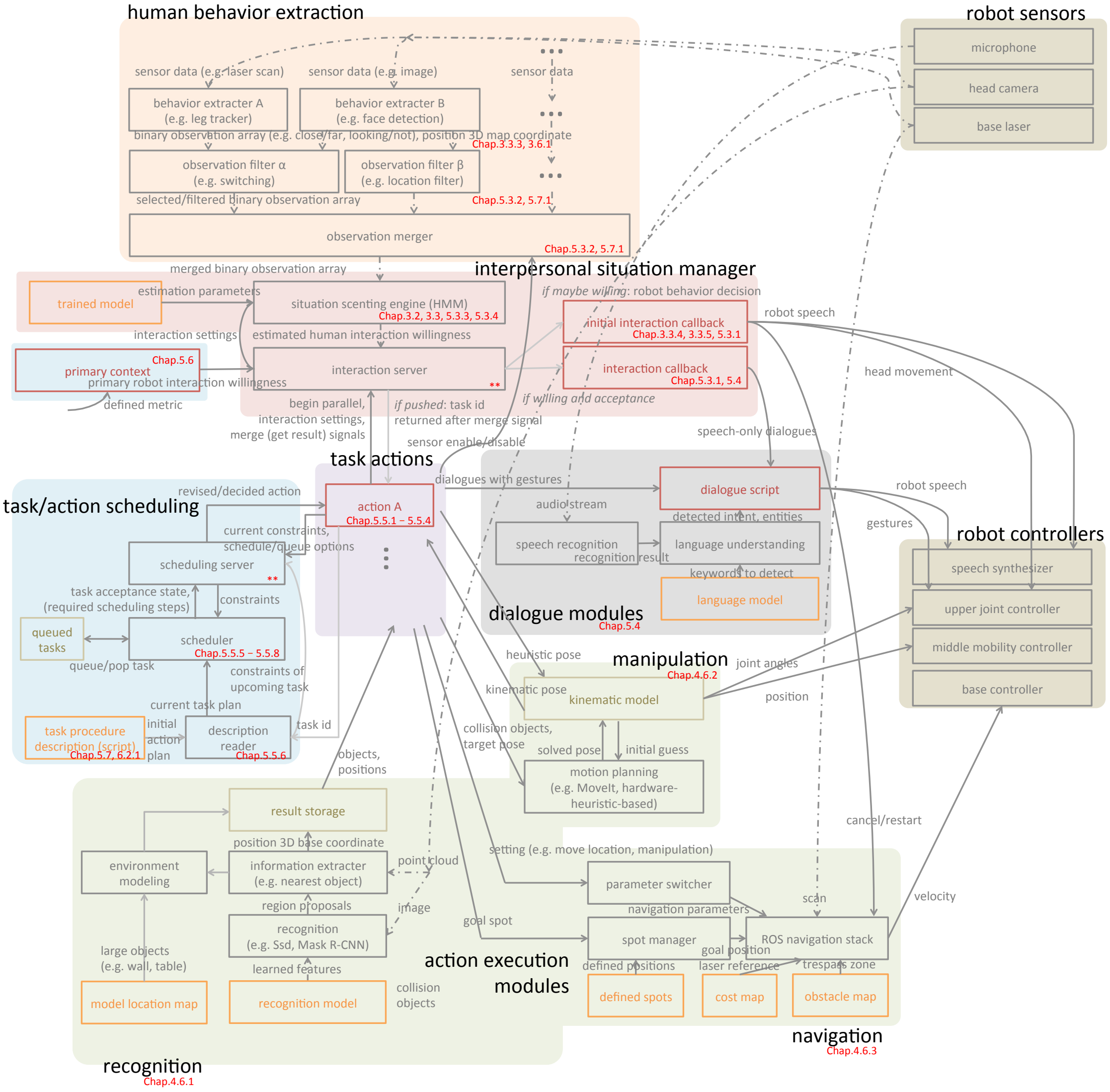


Fig 5.1: Summary and detail of the proposed system and the running nodes. The system autonomously handles the initiation of dialogue-based interaction tasks while executing automated object-transportation/navigating tasks. Information is transferred through ROS based communication. Dotted lines indicate streaming information. Whited lines indicate information that stream under certain conditions. Orange boxes indicate prior or pre-trained knowledge used by the system. Red boxes indicate nodes that require scenario-based implementation. Gold boxes indicate stored memory. ** server nodes act as a proxy layer and therefore detailed implementation are not explained in this book.

5.2.2 Social Context Before an Interaction

For an object transportation or navigating task, the robot might be moving around while a human user may be at a positioned location. Examples include the public domain with physical context experiment from Chapter 3 or the restaurant experiment from this chapter. In these situations, there is distance between the human and robot, and it would sometimes be inappropriate to talk over such a long distance, especially from the perspectives of proxemics [141]. It may be better for the robot to say ”in a moment” and move close by to the person instead of accepting an interaction and talking over long distances.

Moreover, in the example from Chapter 3, the robot may be interacting with one customer, and therefore, may not be in the state to handle a second customer. In fact, in the experiment from Chapter 3, the robot is both at distance and doing a prioritized object transportation task for the first-come-first-served customer.

From the above examples, we see that social constraints sometimes provide reasons to postpone an interaction. Often, these constraints are where the customer is also accepting to wait. Therefore, the situation is more of keeping the H_p-R_n conflict rather than trying to reach a H_n-R_n agreement.

5.2.3 Robot-centric Physical Context

The robot may be under a physical constraint where the robot may not be able to turn its head, as it must keep a look at its current task. The robot may also be busy during the picking and placing process of the task (which we will discuss more in the later sections). Unlike social constraints, physical constraints are robot-centric conditions; therefore, these constraints try to reach a H_n-R_n agreement. As this is robot-centric, this may be uncomfortable for the user. See Chapter 6 for experimental results on the side effects.

5.2.4 Human-centric Dialogue Objectives

Even if the robot is under a physical constraint, there are times where a robot may accept an interaction. This depends on the objective of the interaction (the concrete goal behind the H_p state). From the perspective of dialogues there are mainly two types of user objectives: information-transfer and action-discussion. While the action-discussion objective requires handling of physical constraints, the information-transfer objective could be handled despite the physical state of the robot. A person may ask something to the robot, and the robot could simply provide answers while pausing the task. When the robot does not know whether the user's objective is information-transfer or action-discussion, the robot may accept an interaction once (H_p-R_r), and then decide to postpone (go back to a H_p-R_n conflict) if the objective was an action-discussion, and then accept the task once the physical constraint is removed.

From the above discussion on constraints, the human and robot usually reaches the H_p-R_r agreement directly or through an extension of the H_p-R_n conflict. It should be rare to have to reach a H_n-R_n agreement from the robot side. However, when we look at the situation from the human perspective, we will see that the user may try to reach a H_n-R_n agreement. Such results have been seen in the dataset in Chapter 3.

5.2.5 Required System for Handling Interaction

By looking more deeply into the type of interpersonal situations that happen during a task, we see that an integrated interaction-task system must have the following features: 1) A parallel running interpersonal situation understanding function that lead to an interaction for understanding the dialogue objectives. 2) A task scheduler that handles removing of the physical constraint. 3) A parallel running robot willingness module that updates the willingness state by keeping track of the current social and objective constraints. The third feature depends on the task scenario, therefore we will not provide general implementations but instead, general input/output connections required for the integration.

5.3 Parallel Interpersonal Situation Managing

5.3.1 Interpersonal Situation Manager Component

This component (Fig. 5.1 red region) detects whether there is a chance of interaction using the situation engine from Chapter 3. The role of the situation engine was two folds: 1) understand interpersonal situations and 2) to decide a goal through interactions (whether to interact or not).

In Chapter 3, we have explained that the robot during an initial interaction will output a behavior with a category of R_{p0}, or R_{p1}, or R_{r0}, or R_{r1} where 0 indicates no speech (a looking behavior) and 1 indicates a with-speech behavior. The initial interaction can be seen as a **callback** connecting to the situation engine. This is just like recent computer systems that use asynchronous callbacks to handle user input events. The parallel initial interaction callback is triggered when the engine scents a situation, and then enters a different callback (parallel interaction callback) when the engine is confident of an agreement situation. Similar structures can be seen in recent speech recognition systems that have a callback to return partial results and a final result. The implementations of these callbacks are arbitrary. The initial interaction callback receives the behavior category, and the concrete movement can be designed per task. Usually it should be the actuation of head movements and triggering of speech-to-text. Therefore, these callbacks are usually connected to the robot controller and speech synthesizer directly or through a dialogue module component. In contrast, the situation engine is a higher-level decision layer that does not directly connect to the actuators, but manages decisions relating to interaction.

5.3.2 Human Behavior Extraction Component

This component (Fig. 5.1 orange region) is used to generate an abstracted human behavior observation required for the input to the interaction situation handling component. We have explained in the other chapters that the abstracted observation is binary information that provides hints to estimating human interaction willingness. The implementation of the component depends on the task and may combine multiple nodes to extract different binary cues such as person distance (close or far away from robot) or person face directions (looking at the robot or not). We have shown an example using face directions in Chapter 3. We will show an example of the multiple combined implementation at the end of this chapter in the restaurant example.

5.3.3 The Flow of the Interaction Situation Handling Component

In this section, we will explain the parallel process that is happening in the situation engine, and how it is achieved. The key to the implementation is the multi-threaded structure that is sometimes paused and processed together. Each person in the scene will be applied its own willingness estimation thread. These threads are always running asynchronously, but are paused and synchronized when an interaction target is being searched. Below we explain this threaded process in

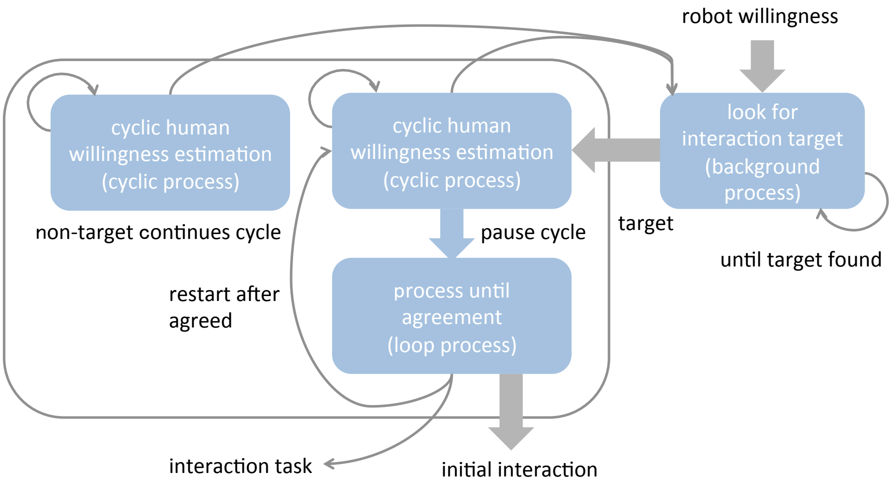

robot willingness
cyclic human willingness estimation (cyclic process)
cyclic human willingness estimation (cyclic process)
look for interaction target (background process)
non-target continues cycle
pause cycle
target
until target found
restart after agreed
process until agreement (loop process)
interaction task
initial interaction

Fig 5.2: How the process works inside the situation engine.

more detail. An image of how the process of the situation engine works is shown in Fig. 5.2 . The actual code flow of the situation engine is presented in Algorithm 5.1 and 5.4.

The first algorithm represents the cyclic process for one of the estimation threads (tied to one of the persons detected in the robot's view). The process runs inside the situation engine at a constant rate (e.g. 10 Hz) and continuously estimates a person's willingness. The robot's willingness is by default R_n during this cycle. Once an update in a robot's primary willingness y_0 is reported to the engine, the procedure in the second algorithm is triggered on a background process. This background process temporarily stops all cyclic processing threads and tries to set an interaction target (select one of the threads; possible selection strategy is passed as one of the interaction settings in Fig. 5.1). (The stopping of all processes is done before triggering the background process therefore not written in the algorithm.) The cyclic processes restart once the above target setting procedure has finished. (Strictly speaking, the cyclic process starts once the engine finds out that the thread is not a possible target, which is not written in the algorithm for simplicity.) The background process will stop if found a target or a *merge signal* (see Fig. 5.1) is provided from the task side. (The merge signal is usually called at the end of a task action.)

First, we will explain the cyclic process running in each thread in more detail. In the cyclic process (when an interaction target is not set or the process is not the thread of the interaction

Algorithm 5.1 Cyclic process of the situation engine.

```
if thread is already running then
    return
start new thread for person $thread$
$observation \leftarrow GetObservation(thread)$
if no person in $observation$ then
    $StackSequence(thread)$
    $ParameterUpdate(thread)$
    $Reset(thread)$
    $SyncOnce(thread)$
    if $thread$ is interaction target then
        $SetTarget(NULL)$
if $observation$ is different from previous frame then
    $Trim(thread)$
    $Process(thread, observation)$
    reset time
if time has elapsed then
    $ObserveFrom(thread)$
    reset time
$SyncOnce(thread)$
```

Algorithm 5.2 Process(thread, observation) function in cyclic process.

```
append $observation, thread.robotWillingness$ to $thread.observations$
$ParameterUpdate(thread)$ if any sequence in stack
$humanWillingnessEstimate \leftarrow GetState(thread.observations)$
if $thread$ is not interaction target then
    $SyncOnce(thread)$
    return $thread.robotWillingness$
if not $Conflict?(thread.robotWillingness, humanWillingnessEstimate)$ then
    if $humanWillingnessEstimate$ is not $H_n$ then
        $StackSequence(thread)$
    return $thread.robotWillingness$
$nextWillingness \leftarrow DecideAction(thread.robotWillingness, humanWillingnessEstimate)$
if $nextWillingness$ is the same as $thread.robotWillingness$ then
    return $thread.robotWillingness$
$thread.robotWillingness \leftarrow nextWillingness$
return $ObserveFrom(thread)$
```

Algorithm 5.3 ObserveFrom(thread) function in cyclic process.

```
if level(thread.robotWillingness) is not 0 OR thread.robotWillingness is different from previous frame then
    ExecuteAction()
    while robot behavior has not finished do
        SyncOnce(thread)
        observation ← GetObservation(thread)
        if observation is different from previous frame then
            append observation, thread.robotWillingness to thread.observations
            ParameterUpdate(thread) if any sequence in stack
observation ← GetObservation(thread)
Trim(thread)
return Process(thread, observation)
```

Algorithm 5.4 Background process when the robot's primary willingness is updated.

```
interactionType ← from parameters passed with primary willingness
expectedTarget ← from parameters passed with primary willingness
candidates ← $\phi$
if inteactionType is parallel (check human to robot) interaction then
    for thread in threads do
        if thread.observations is not empty then
            humanWillingnessEstimate ← GetState(thread.observations)
            if Conflict?(thread.robotWillingness, humanWillingnessEstimate) AND humanWillingnessEstimate is $H_p$ then
                push thread to candidates
    if candidates not empty AND interaction target is not set then
        SetTarget(SearchTarget(expectedTarget, candidates))
        thread.robotWillingness ← DecideAction(thread.robotWillingness, targetThread.humanWillingnessEstimate)
        pause targetThread
        ObserveFrom(targetThread)
        restart targetThread
else if interactionType is sequential (robot to human) interaction then
    SetTarget(SearchTarget(expectedTarget))
    pause targetThread
    ObserveFrom(targetThread)
    restart targetThread
```

target), an observation is updated when 1) the robot conducts a new action, 2) a change in observation occurs, or 3) constant time has elapsed. When the robot is at task state, the robot is R_n, and thus, the observation will proceed with only conditions 2 and 3. When a person is lost, $Reset()$ is called to initiate observations and estimations. Observations from sensors are queued and the latest observation is popped with a $SyncOnce()$ function. The latest observation is then called from $GetObservation()$, the human state is estimated with $GetState()$, and the state confidence and conflict termination (whether the robot has reached an agreement) is evaluated with the method $Conflict?()$. If the conflict has not yet reached an agreement, the next action is decided with $DecideAction()$, and if there is a change in action decision (meaning the robot will conduct a new action through the parallel initial interaction callback), a signal is sent to the interaction callback with $ExecuteAction()$.

Note that for the $GetState()$ function, only the target thread uses the robot's willingness for estimation and the rest continues the cyclic process with robot's willingness set as R_n.

After an interaction target is set, the cyclic process of the target thread will pause and the thread will instead loop over $ObserveFrom()$ (updating observations and robot action) and $Process()$ (updating estimations) until reached an agreement. Once reached an agreement, the thread will go back to the cyclic process with the robot's willingness set to whatever reached agreement state (could be any of R_n, R_p, or R_r).

5.3.4 Runtime Model Training

The $StackSequence()$ and $ParameterUpdate()$ function in the algorithm stores the current interaction sequence and updates estimation probabilities on runtime from how an actual interaction went. Update rules are built as addons and are embedded in the situation engine. Results of using such runtime updates are shown in Chapter 6. However, in the rest of the experiments, this function is turned off. Even when the function is turned off, all interaction data are stored as logs in case a model is wanted to be trained later offline.

5.3.5 Remarks on Detecting Interaction Finishes

It is possible to check the end of an interaction using the situation engine if needed. In this case, the interaction callback will communicate with the situation engine by providing the robot's willingness state every time it changes. (Basically a state will change when there is a change in who has floor. The willingness could automatically be mapped from speech content.) This is not illustrated in the figure as we find this connection as optional, and not used for most of the task execution tasks we encounter in this book. The end of an interaction can usually be reported from the task side. For example, an interaction only task could finish when the person walks away from the robot, and a task related interaction could finish from a known context flow (a conversation with a waiter could finish by asking ”anything else?” and a following ”no” response could trigger the end of an interaction). Finish flags are passed as one of the merge signals.

5.3.6 Remarks on Robot-to-Human Interaction

Estimations in robot-to-human scenario may also run in parallel. The process is similar to human-to-robot interaction, except the robot tries to estimate between H_r and H_n and possibly H_p (depending on the scenario) instead of between H_p and H_n. The other difference is that the robot-to-human interaction setting usually transitions to a sequential task prepared by the robot. This means that although the initiation happens in parallel (physically approaching and speech timing happen separately), the initiated interaction will happen as a sequential flow. Therefore, instead of triggering the interaction callback, in the robot-to-human scenario, the situation engine will return whether the task is proceedable or not to the task actions. A task might not be proceedable if a person ignores a robot (H_n), or the person instead asks a request (H_p). In the former case, the robot will have to abort its task. In the later case, the robot will enter the parallel interaction callback.

5.4 Dialogue Modules

Dialogue modules are triggered in the following cases: 1) interaction callback after scenting an interaction, 2) from the task actions. In the first situation, the robot is listening to the user's request to find out the dialogue objective, or providing short answers that do not require any gestures. Using gestures may require scheduling of actions (next section) and should not be used in this situation. In the second situation, the robot is doing an interaction task. In this situation, the robot may use gestures or connect to manipulation actions such as fetching a drink. The situation is the usual situation we see in task-interaction integrated systems where both the human and robot are ready for an interaction at the beginning of the task. Modules include a speech recognition,

Table 5.1: Some of the command level action sequences from the APC stow task.

command	purpose
move(lifter)	potential collision avoidance
pose()	init a pose
move(torso)	pose for recognition
move(left-shoulder)	occlusion avoidance
move(lifter)	pose for recognition
move(left-shoulder)	prepare reach pose
pose()	prepare reach pose
move(torso)	prepare reach pose
move(lifter)	reach collision avoidance
trajectory()	reach by inverse kinematics
move(lifter)	reach collision avoidance
grasp()	
move(lifter)	withdraw collision avoidance

language understanding, and a dialogue script module. Speech recognition converts human speech to text and the language understanding extracts the intent of the text (e.g. ”Where is the office?” is a speech, *find location* is the intent). By extracting the intent we may map the speech to the dialogue objective (information-transfer or action-discussion). These modules are provided as open cloud services (e.g. Microsoft LUIS, Google Dialogue Flow). The dialogue script is a mapped list of human-intent and robot-response pairs. A predefined speech and gesture by the robot is triggered from the detected human intent.

5.5 Task Scheduling and Constraints

5.5.1 Robot-centric Physical Context and Task Acceptance Types

Table. 5.1 and Table. 5.2 lists some of the actual command level action sequences that were required in the APC stow task and FCSC display task from the previous chapter. From our analysis on motion commands and the purpose of the command, we find that there are three types of actions regarding task acceptance (acceptance from task to a different task, especially an action-discussion request).

Table 5.2: Some of the command level action sequences from the FCSC display task.

command	purpose
pose()	pose for recognition
pose()	init a pose
move(base)	adjust to recognition
openhand()	
trajectory()	reach by plan
grasp()	
trajectory()	withdraw by undo
move(base)	undo adjust

The first type of action is **acceptance hard**. This is the case where a robot is not able to freely move its body (e.g. potential collision, holding an object with both hands, etc.) and therefore, must proceed or backward the task before it is able to accept an action-discussion driven task. For example, such examples are seen when the robot is putting its hand in the APC shelf. The robot must either backward its actions and remove its hand out of the shelf or proceed and grasp an object and then move its hand out before it is able to begin the next task. Another example that falls into an *acceptance hard* condition is when the robot is opening a fridge, or is manipulating under an opened fridge. The robot must close the fridge first before it can accept any new tasks, and therefore, cannot move freely.

The second type of action is **acceptance conditional**. In this type of action, a robot may move freely, but must redo some previous actions when it returns to its current task. For example, in the APC task, the robot positions to look inside the tote. The robot may move to a different task as it is not under any potential collision, but when it returns to the task, the robot must redo the looking.

The third type is **acceptance**. The robot may move freely and does not have to redo actions when returning to its task. For example, in the FCSC, the robot is just about to move to the sandwich shelf to begin a sandwich task. In this situation, the robot will begin from moving to the sandwich shelf despite whether it was interrupted with a different task. Pictured examples of acceptance hard and acceptance conditional is provided in Fig. 5.3 .

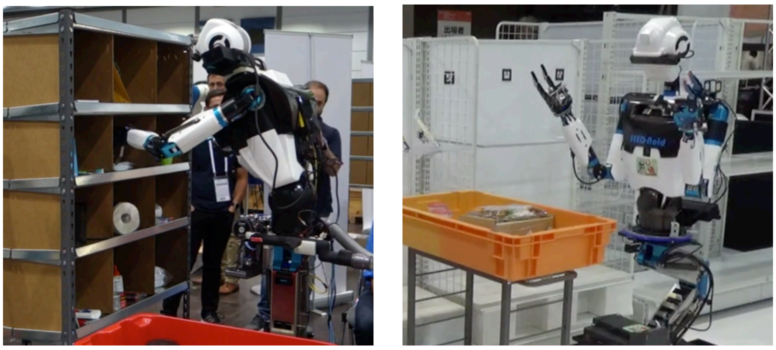


Fig 5.3: Picture example of an acceptance hard action (left) and an acceptance conditional action (right).

5.5.2 Percentage of Task Acceptance Types in Object Transportation Tasks

Fig. 5.4 shows the actual percentage of each acceptance type in the two competitions. From the graph, we see that an action-discussion driven task is actually not immediate acceptable in most cases. Tasks are mostly *acceptance hard* and if not, mostly *acceptance conditional*. Since the examples are both object transportation tasks, these would be common situations in most targeted applications of task robots entering society.

5.5.3 Exceptions to Acceptance Hard and Forward Looking at Actions

Below we provide an example exception to acceptance hard conditions that may transit to an action-discussion task. Let us assume an action sequence that is composed of the following list of actions: *moveLocation*, *pose*, *doRecognition*, *moveArmVisionContext*, *grasp*, *release*, *lookPersonContext*, *moveArmPersonContext*. *moveLocation* moves the robot to a pre-defined work spot. *pose* directly moves the joint angle of the robot to create some pose. *doRecognition* does the recognition and creates a model environment. *moveArmVisionContext* moves the joint angles from kinematic calculation in relation to the model environment. *grasp* closes the hand. *release* opens the hand. *lookPersonContext* looks at a person. *moveArmPersonContext* moves the joint angles from kinematic calculation in relation to a person.

Let us assume the current action sequence (we will name this sequence A) is a pick and place

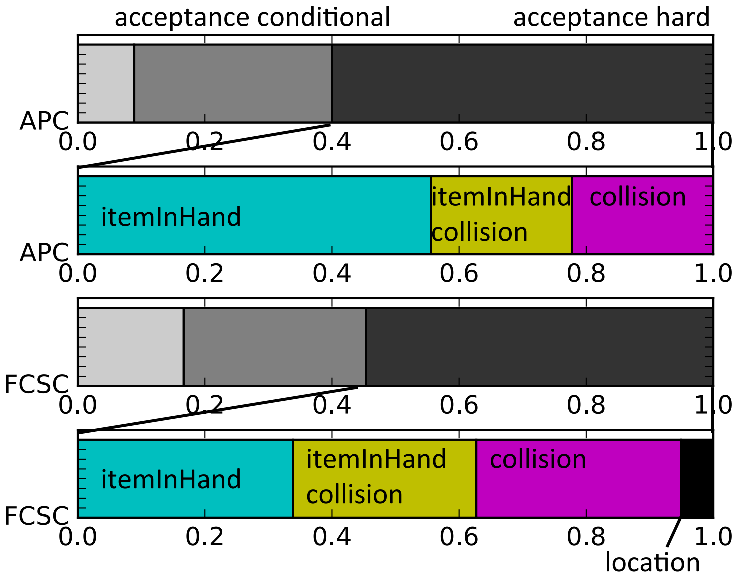


Fig 5.4: Percentage graph of each acceptance type in the APC and FCSC task, followed by the percentage of constraint types of acceptance hard.

task on a table, where a big open box and a drink is on the table. The actions are *moveLocation(table1), pose(for recognition), doRecognition, moveArmVisionContext(drink), grasp(power grasp, drink), pose(move arm away from table1), moveArmVisionContext(box), release(drink), pose(move arm out of box)*. Let us assume that during *pose(move arm away from table1)* we encounter an interaction.

In this interaction task, the robot is not sure what the person is going to ask, and therefore, accepts to talk with the person. During the talk, the robot finds out that the person wants the robot to do the following action sequence (we will name this sequence B): *lookPersonContext, moveArmPersonContext, release(drink)*. As the robot is holding a drink, it is not able to move freely and is under an *acceptance hard* state. However, the task that is being asked requires the robot to do the next task at the particular moment despite the fact that the upcoming task requires the robot to *move* its body. The exception here is that, the constraint that is keeping the robot from accepting a new task is removed by the upcoming task. We will assume that there is a pre-defined constraint that the robot may check to decide whether the robot may move freely using information on the upcoming task

5.5.4 Defining Actions

For smooth transitions between tasks, we must divide task actions so that one action does not include multiple acceptance types. For example, rather than setting a ”pick” as one action including pre-posing and recognition processes, the ”pick” should at least be divided to ”getting ready for a pick” and ”pick”. This will allow a person to interrupt the robot before it begins a pick, and smoothly transition to the next task. Elsewise, the person may have to wait until the robot finishes placing an item. Dividing actions depending on acceptance types will allow more chance of smooth transitions between tasks. In addition, actions may have to be divided according to a change in acceptance hard constraint conditions (increase or decrease in constraints) depending on the task scenarios the robot accepts. This was the case for the exception in the previous section.

5.5.5 Processing Task Acceptance

From the discussions in the previous sections, there are three things that must be checked when switching tasks under an action-discussion. First, we must check the acceptance type of the current action in the task. If the robot is under *acceptance hard* it must proceed or backward its current task. If the robot is under *acceptance conditional* or *acceptance* it may consider proceeding to the upcoming task. Second, if the robot is under *acceptance hard* it must check the constraints of the current task and the upcoming task. If there is a current constraint or any constraint in the future that is removed in the upcoming task, the robot may accept the task at the moment the constraint is removed. Third, if the robot was at *acceptance conditional*, we must check for a redo or skip of the task actions.

5.5.6 Task/Action Scheduling Component

This component (Fig. 5.1 blue region) handles the conditions regarding acceptance. We assume that some scripted information on acceptance type and constraint of each action in a task is provided. The scheduling module parses this script to inform the system on the current acceptance state as well as the required number of steps to forward/backward before starting a new task. The input of the module is the script, but also the result from the situation engine (whether the robot is free of being R_n or the robot must try to enter R_r; here, the result includes information such as what the next task should be). The scheduling module returns a list of actions to do (forwarding/backwarding) before an interrupted action-discussion task, or a list when returning from the interrupted task to the original task. If there is no need to handle any task transitions (free of being R_n), the scheduling module will execute the list of actions in the current task one-by-one.

We summarize our procedure for switching tasks from script information in Algorithm 5.5 and 5.6. The algorithm assumes that the interaction (getting a request of the next task) happened and finished before finishing the current action, or the current action waited for the interaction to finish before proceeding to the next.

5.5.7 Limitations and Relation to Task Planning

In some cases, the previous task may abort once finished the new task. For example, in the case of the person wanting to take over the current task of the robot and instead wanting to order something different e.g. take away what the robot is delivering and ask to go over and handle a customer. In more complicated situations, a robot may be doing a looped pick and place task and sometimes, a person may ask to handover an item instead of placing it. In these cases, a pointer to the start of the task (in this case, the start of the loop which is a pick) may be set when going back to the previous task. Or, a robot may start from a middle of a task as if the *acceptance conditional* situation. If the robot requires an action that is not in the action list of the previous task, then, that is a problem that requires global planning such as using SMACH [11]. Planning in some sense can be seen as an online generation of a task script. For our framework to work on a generated script, we would have to automatically annotate constraints and acceptance type. Yet, there are many interaction related tasks or switching between various tasks that do not require such planning complexity. For example, postponing and coming back later is one of those tasks that do not require handling complex transitions. In another example, a robot pushing a cart may be stopped by a person, and the person may add additional items to the cart for the robot to carry. These are interactions that require scenting in a middle of a physical task but do not require any motion from the robot.

In other cases, the robot may not have information on the upcoming task. Such situations can be

Algorithm 5.5 Interaction acceptance and task scheduling.

```
$actionList, acceptanceType_t, allowedProcess_t$ ← read current task script
**if** $acceptanceType_t$ is not ACCEPTANCE HARD **then**
    **return** SUCCESS
$nextActionList$ ← read task script of upcoming task
**for** $a$ in $nextActionList$ **do**
    $nextConstraintStatus$ ← add information of $a$
$backingPlan \leftarrow \phi$, $proceedingPlan \leftarrow \phi$, $backingPlanComplete$ ← FALSE
**if** $allowedProcess_t$ includes backward planning **then**
    **for** $a$ in descending range [t, 0] of $actionList$ **do**
        push $a$ to $backingPlan$
        $constraintStatus_i$ ← read information of $a$
        $acceptanceType_i$ ← read information of $a$
        **if** $constraintStatus_i$ is removed in $nextConstraintCondition$ OR $acceptanceType_i$ is not ACCEPTANCE HARD **then**
            $backingPlanComplete$ ← TRUE
            **break**
$proceedingPlanComplete$ ← FALSE
**if** $allowedProcess_t$ includes forward planning **then**
    **for** $a$ in ascending range [t, ] of $actionList$ **do**
        push $a$ to $proceedingPlan$
        $constraintStatus_i$ ← read information of $a$
        $acceptanceType_i$ ← read information of $a$
        **if** $constraintStatus_i$ is removed in $nextConstraintCondition$ OR $acceptanceType_i$ is not ACCEPTANCE HARD **then**
            $proceedingPlanComplete$ ← TRUE
            **break**
**if** $proceedingPlanComplete$ is TRUE **then**
    **if** $backingPlanComplete$ is FALSE OR length of $proceedingPlan$ is shorter than $backingPlan$ **then**
        do $proccedingPlan$
        **return** SUCCESS
    **else if** $backingPlanComplete$ is TRUE **then**
        do $backingPlan$
        **return** SUCCESS
**else if** $backingPlanComplete$ is TRUE **then**
    do $backingPlan$
    **return** SUCCESS
**else**
    **return** FAILED
```

Algorithm 5.6 Return to task after acceptance conditional.

```
$redoActions \leftarrow \phi$
$actionList \leftarrow$ read current task script
for $a$ in descending range [t, 0] of $actionList$ do
    stack $a$ to $redoActions$
    $acceptanceType_i \leftarrow$ read information of $a$
    if $acceptanceType_i$ is ACCEPTANCE then
        break
while $redoActions$ is not empty do
    do top of $redoActions$
    pop $redoActions$
```

found in a teaching-by-demonstration scenario. The person interrupts the robot and tries to teach or revise a skill. In this situation, the forwarding and backwarding of actions could be commanded by a person instead of automatically handled by the algorithm. These type of settings are passed as parameters to the scheduling component, but also require extra nodes (not in the figure) to auto-generate action lists.

5.5.8 Relation to Task Queuing

In addition to checking the task acceptance state and handling forwarding/backwarding of a task, the task scheduler must have a queuing function. It is apparent that the module requires such a function since the scheduler also handles returning of a task for an acceptance conditional task. Here we will briefly explain its mechanism.

There are five types of queued tasks. The next task, tasks queued with priority, tasks that were queued due to interruption, tasks escaped from an error state, and routine tasks. The next task will always happen after the current task. Tasks queued with priority will happen before the other task types if it has a high priority but after other types if low priority. Tasks that were interrupted will be popped before the other tasks unless the other tasks have a higher priority setting. Tasks escaped from an error are queued in occasions such as failing a robot-to-human interaction. These have lower priority as an error might be under fix by someone else or the robot has to come back later when the person might be ready. These will only be popped when at least one different task from queue is selected first, or when there are no other tasks in queue or all tasks in queue are error-escaped tasks. The routine task will be selected last unless the routine tasks have priority or the task was interrupted. The routine tasks will be popped in whatever order specified by the user.

When an interruption happens during a task, the current task will be queued unless specified an abort option. An abort option may be specified from the task or the interaction callback. An example of an abort from task will be when the robot fails a robot to human interaction. However, the robot also has a choice of queuing this task as an error and pops and retries the task later on. An example of a callback abort is when a person asks the robot to do something else, and the person takes over the current robot's task.

5.6 Primary Context Node

The primary context node sets the primary robot's willingness of reaching a H_n-R_n agreement or reaching a H_p-R_r agreement (responding with a ”please wait” or ”yes?”), as well as, settings during a robot-to-human interaction.

In the human-to-robot interaction situation, there are three types of settings: accept an interaction (reach a H_p-R_r situation and listen now), or postpone an interaction (reach a temporary H_p-R_r situation but then reach H_n-R_n at the moment and listen to the details later), or completely avoid an interaction (try to reach H_n-R_n without a temporary agreement). The settings are related to the social context such as distance between the human and robot.

Note that not all social contexts are handled in the primary context node and some are related to task acceptance and physical constraint. For example, a robot may be throwing away some garbage as a person comes by asking for directions. The request itself is an information-transfer dialogue. Yet, the robot may have to point toward some direction. It may be rude for the robot to have garbage in its hand while pointing. This is more of a post-process manner decision and is expressed as a hand usage constraint. Once the robot finds out that it needs to point directions, it will forward its actions of throwing the garbage before answering the person's question.

Beside, social contexts, a more robot-centric metric could also be applied. For example, the robot could use the acceptance state information and speak that it is busy, despite whether the person will ask for an information-transfer or an action-discussion. This may sound inappropriate, but when the robot is under a long action, such behavior may be required; especially if there is a chance that the robot will keep the person waiting for quite a while, after knowing that the request is an action-discussion.

In the robot-to-human interaction situation, the node may set settings such as to just stare at the person to catch attention, catch attention by speaking, or catch attention if the estimation score seems not to change. By setting such settings in an independent node, we may control the interaction timing independent from the task state. For example, a robot could have a task where the robot approaches a person. The node could set the robot's behavior to *just stare* but then change the behavior to *catch attention* once estimated a high score of a person willing to interact. The change in setting is then passed to the situation engine as a change in primary robot's willingness. The change will at the end, affect the type of behavior to conduct in the initial interaction callback. Since the changing of a robot's willingness is dependent on the scenario of the task, such setting of willingness is defined outside the situation engine.

5.7 Interaction-Task Integrated Example using the System

5.7.1 Achieving the Restaurant Task

In this section, we prove that our system works and achieves a complex setting such as a restaurant scenario. The robot is capable of the following tasks: *routineTask*, *goAndListen*, *entranceCustomerHandling*, *storeMenuToShelf*, *passTheMenu*. The restaurant task is composed of four parts and uses a combination of three sensors: base laser, head camera and torso camera. The four parts are shown in Fig. 5.5 and is running one or two of the capable tasks. The task procedure description (scripted list of task actions) for the *goAndListen* task is provided at the end of the section as an actual coded example.

The first part is handling of an interruption during the robot's *routineTask*. In this routine task, the robot is picking up a dish to clean the table and tries to put the dish in a carrying container. This is similar to the FCSC storing task in that we are picking items from the narrow (the goal position of fingers are constrained between the object and the table). The robot succeeds the task by using the techniques from Chapter 4. For the picking, the planar constraint holds, and therefore, we use the middle mobility layer to move the fingers under the dish. In addition, the navigated solution ”fits” a heuristic pose using both arms to pick up a dish. Since the robot is holding a large dish

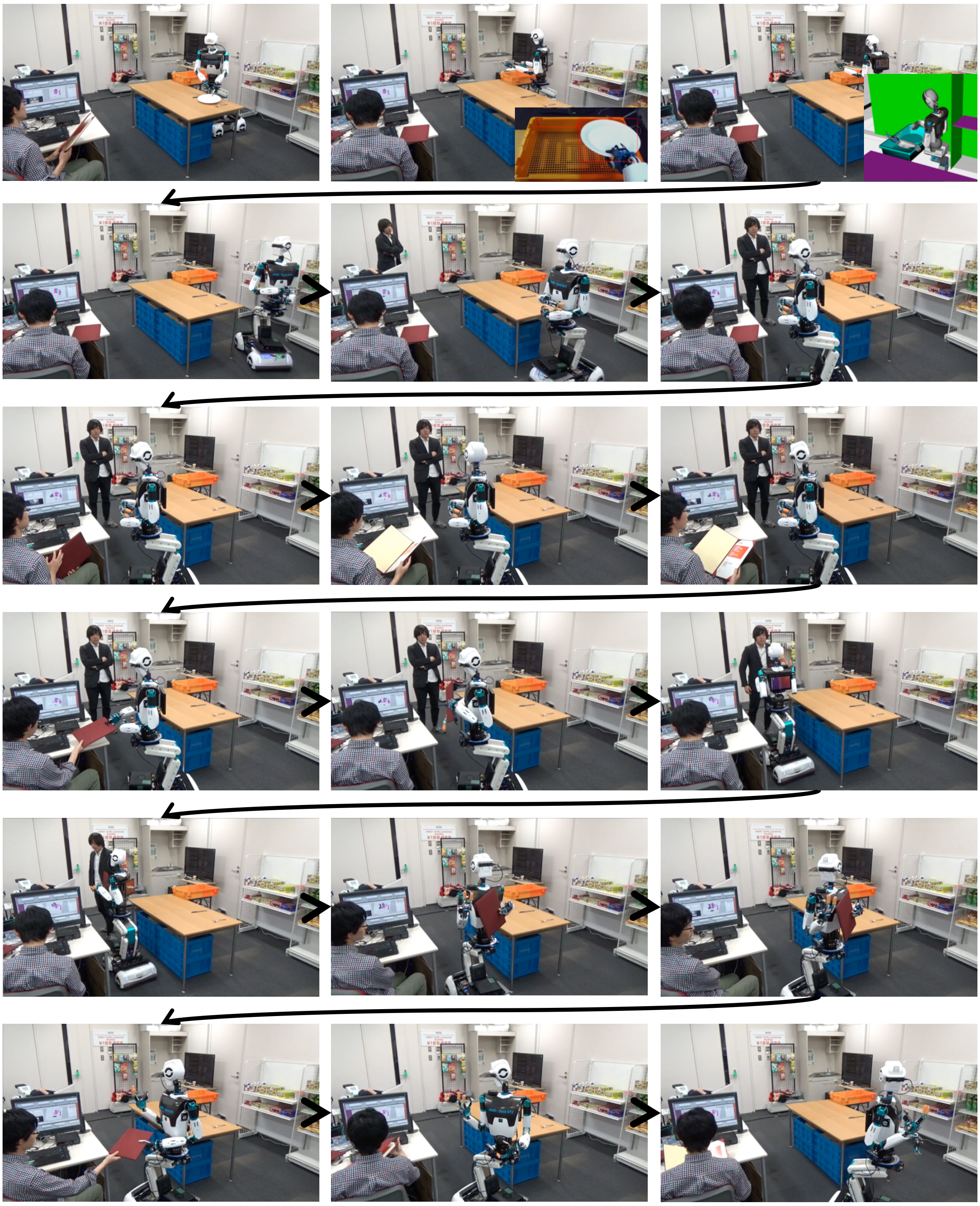


Fig 5.5: The restaurant task. In the top row, the robot is cleaning up the dish from the table. In the second row, the robot approaches the person and tries to initiate an interaction. In the third row, the robot interacts between two customers and acts according to the scented willingness. In the fourth row, the robot receives the menu and starts guiding the second customer. The timing of receiving the menu is decided from the customer's willingness state. In the fifth row, the robot finishes the guiding task and scents an interaction on its way back. In the sixth row, the robot hands back the menu. The move away timing is decided from the customer's willingness state.

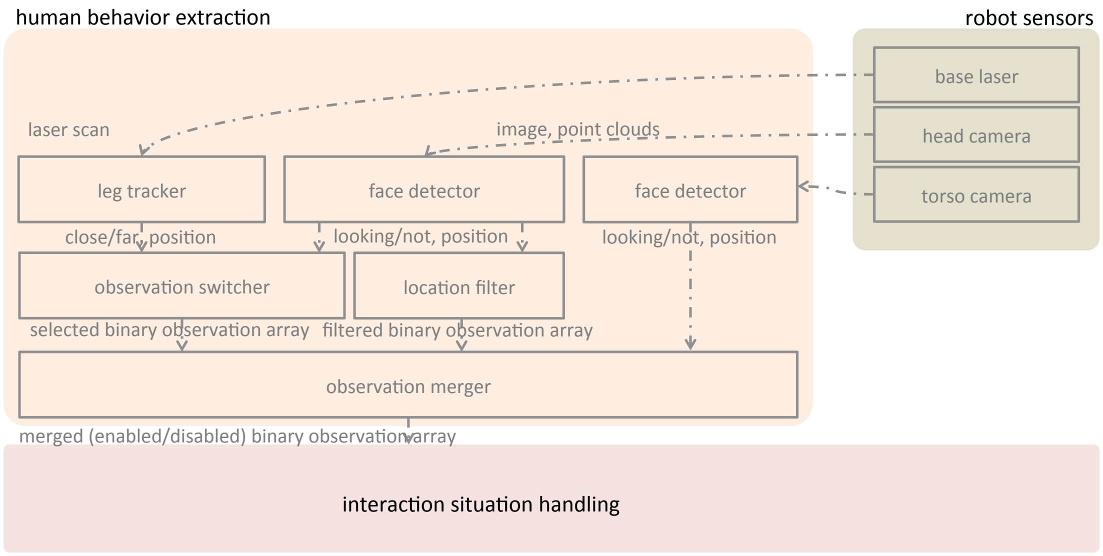


Fig 5.6: The behavior observation module with multiple sensor gateways. This allows more complex scenting but does not change the structure of the situation engine nor the base architecture.

that is almost the same size as the container, the problem of placing the dish in the container is difficult to solve with online planning. The images in the figure show that our approach of using predefined motions and the navigated solution simplifies such planning and achieves the solution.

While the robot is doing this routine task, the robot is constantly scenting human engagement from a sitting customer (the customer is at a different table). We use the torso sensor for this detection. The primary context uses the distance metric. If there is any possibility of a human to robot interaction, the robot will enter a temporary agreement but ask the user to wait until the robot reaches close by. After knowing that the robot must approach the person, the robot will check the task acceptance state. If the robot finds that it is in an acceptance or conditional state, it will go to the customer first. Else, the robot will forward or backward its task depending on which is faster (have fewer action steps) to accomplish. The images in the figure show that the robot successfully scents a customer and reaches back after finishing the dish-placing task.

The second part is approaching the customer at the table. The robot starts a new task *goAndListen* that was set when scenting a temporary agreement. The robot navigates itself using the

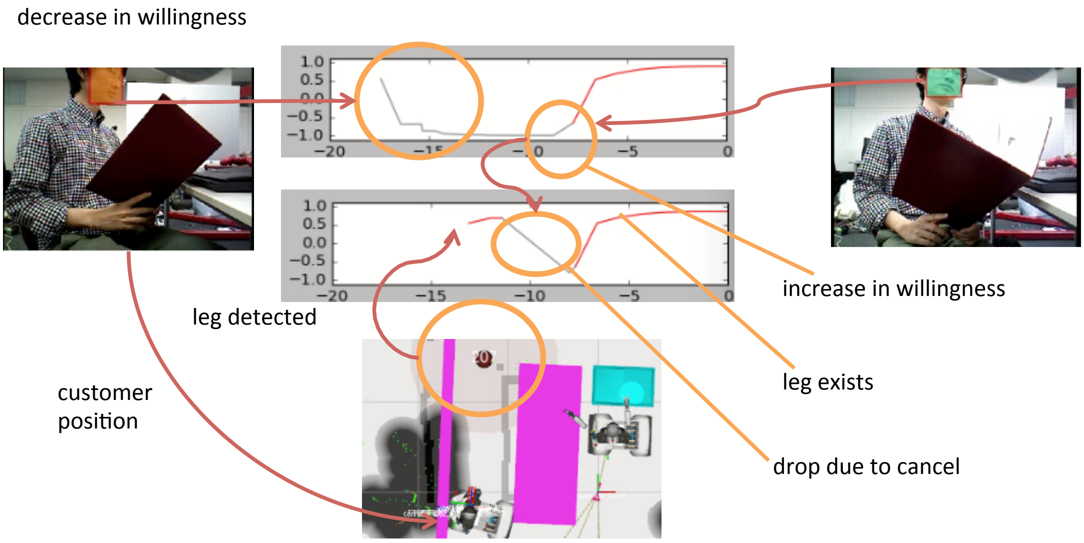


Fig 5.7: The willingness estimate between the two customers. The face detection results indicate the customer who is requesting orders. When the willingness from the face detection decreases, the robot handles the scented leg. When the willingness from the face detection increases, the robot re-handles the customer requesting orders.

navigation stack and predefined map information. The robot will not accept any new interaction at this phase (the robot's willingness is strictly R_p). The robot uses observations from its head sensor, and passes the observations to the situation engine to sense whether the person is ready to interact with the robot. After the robot has successfully interacted with the customer, it will listen to the customer's request (ordering from a menu). This is shown in the second row in the figure.

While the robot is taking orders, the robot will either focus on the current customer, or if the robot notices that a different customer is waiting at the entrance and the current customer is busy looking at the menu, the robot will give eye contact to the customer waiting at the entrance. This is an example of handling interaction situations during interaction tasks.

The handling is done by using multiple sensor inputs and comparing willingness scores (Fig. 5.6). The customer already under interaction with the robot is observed using the torso sensor. The second waiting customer near the entrance is observed using a leg to camera switching observation. When the willingness score (posterior probability) of the current customer becomes low and the score of the entrance customer is high, the robot will give an eye contact to the customer at the entrance. However, if the current customer's score increases, the robot will stop (cancel) the eye contact motion and look back at the current customer. Fig. 5.7 shows that the system successfully estimates and achieves the above situation.

At the end of the interaction task, the robot will take away the menu. For the timing of initiation of the handover, we will again use our engine. If the human's willingness score rises, the robot will suggest taking away the menu.

Since the robot's hand becomes occupied, it will gain a physical constraint. In order to remove constraints before doing other tasks, the robot must queue a *storeMenuToShelf* task with high priority to remove the constraint. However, if interrupted (in this case, a customer at the entrance is waiting), a robot may set the *next task* as *entranceCustomerHandling*. As we have explained, the system handles the queue in the order of: next task first, then whatever is prioritized in the queue. This is successfully done as shown in the fourth and fifth row of Fig. 5.5 .

The third part of the task is handling the entrance customer in the *entranceCustomerHandling* task. Again, the robot will try a robot to human interaction. If it fails the interaction (e.g. the person was already seated) the robot will continue with the queued task (store the menu). Here we assume that the robot has knowledge that the *entranceCustomerHandling* task does not have any conditions relating to the hand occupancy constraint (in this task, we assume the menu for the new customers are already prepared on the table and does not require any hand over actions from the robot). The scheduler will therefore, accept this task.

The fourth and final part is the *storeMenuToShelf* task. The robot is navigating itself to place the menu in-hand back into the shelf. This time, a customer tries to catch the robot's attention as the robot passes by. Therefore, the distance is close enough that the robot will accept the interruption immediately. The user will ask that he wants to see the menu again. This is an example of handling constraints that is removed in the upcoming task (*passTheMenu*). The sixth row in Fig. 5.5 show that the task scheduler successfully handles this situation.

5.7.2 Limitations

A person's attention toward a robot may decrease as he or she looks at the menu or looks at his or her phone. Our default trained scenting model does not take into account such context, but instead, tries to distinguish long-term loss of attention and short term loss such as gaze aversions.

Our model will not distinguish an end of an interaction (H_n-R_n agreement) opposed to long-term loss of attention. They are both a long term loss. Such high level context would be required during an interaction that is complex as the restaurant task. We see that there is room for enhancing the current architecture. We would need to integrate task-situation scenting on top of interpersonal situation scenting.

Although we have experimented a restaurant scenario, one may question how well the experiment condition would apply to a real restaurant. We have looked at five tasks: cleaning up a dish from the table, taking orders, guiding a customer, storing the menu, and handing over the menu. Each task in the restaurant task was represented as a sequence of actions. We have assumed that the type of dish and scenery is possible of knowing beforehand, and that such knowledge is applicable. We have handled object location uncertainty using the movement of the base, which should be an applicable solution at real restaurants as long as there are enough landmarks to laser scan match and localize the robot's position. Technically, the task falls under the paradigm of pick, carry, and place of known objects; which is what we have explained are the basic and required functions for a task execution system to enter society.

However, the problem with the restaurant task is that most of these tasks were triggered through interpersonal situations rather than from the robot's task state space. We may say that this is why our architecture is technically important for developing these type of tasks, but at the same time, this is why we may say that the restaurant task will not yet enter our society. There are too many expectations on the robot in the restaurant task. The robot must take an order, might need to re-fill a drink, might need to pick a dropped item, there are so many things a person could ask the robot. We have seen with the current social robots that, over expectations must be taken with care.

```
# go_and_listen (ac indicates acceptance conditional, ah indicates hard)
0,0,init,_get_agreed:false
0,0,acSetupScentingSensors
0,0,acPose
0,0,acLookAtPersonAction,_agree_point:0,_agree_type:prioritized
0,0,acMoveToPersonAction,_agree_point:0,_agree_type:prioritized
0,0,ahInteractWithCustomer,_agree_point:3,_agree_type:prioritized
0,1,ahGraspMenuAction
1,1,ahPose
```

5.7.3 Achieving the Garbage Throwing Task

Our system also achieves other scenarios such as taking away a tray and throwing away garbage for a person at a food court. While the restaurant task has shown that our system is able to handle various initiation scenes, the garbage task shows that the system is able to handle various initiation timings and different objectives that happen in a single scene. Successful results are shown in Fig. 5.8 .

5.8 Conclusion

There are three types of interpersonal agreements, but also, two distinct dialogue functions; which make interaction-task integration complex. To solve the different contexts that underlie, we have introduced task acceptance and its relation to the physical context and human objectives. By analyzing the competition tasks, we have found that in object transportation tasks, more than 50% of the time, a robot may not be able to accept an interaction task, but if the system checks between the constraints of the current task and the upcoming task, the robot might be able to accept for more than 30% of the acceptance-hard task state.

We have shown the capabilities of our system through a restaurant and garbage task scenario. The task setting had a more close-to-real assumption when compared to the restaurant task at competitions e.g. RoboCup@Home. The robot is always doing some routine work and not just waiting. Multiple customers may be waiting for the robot. A person may be looking at his or her cellphone when the robot approaches. A passing of a menu is done between the person and robot. The manipulation is much more rigorous and so are the interpersonal situations. We have achieved such a complex setting using the manipulation simplifying techniques from the previous chapter, combining a parallel running situation scenting capability to the task execution nodes, and adding a task scheduler that handles task switching from both a robot-centric and human-centric view.

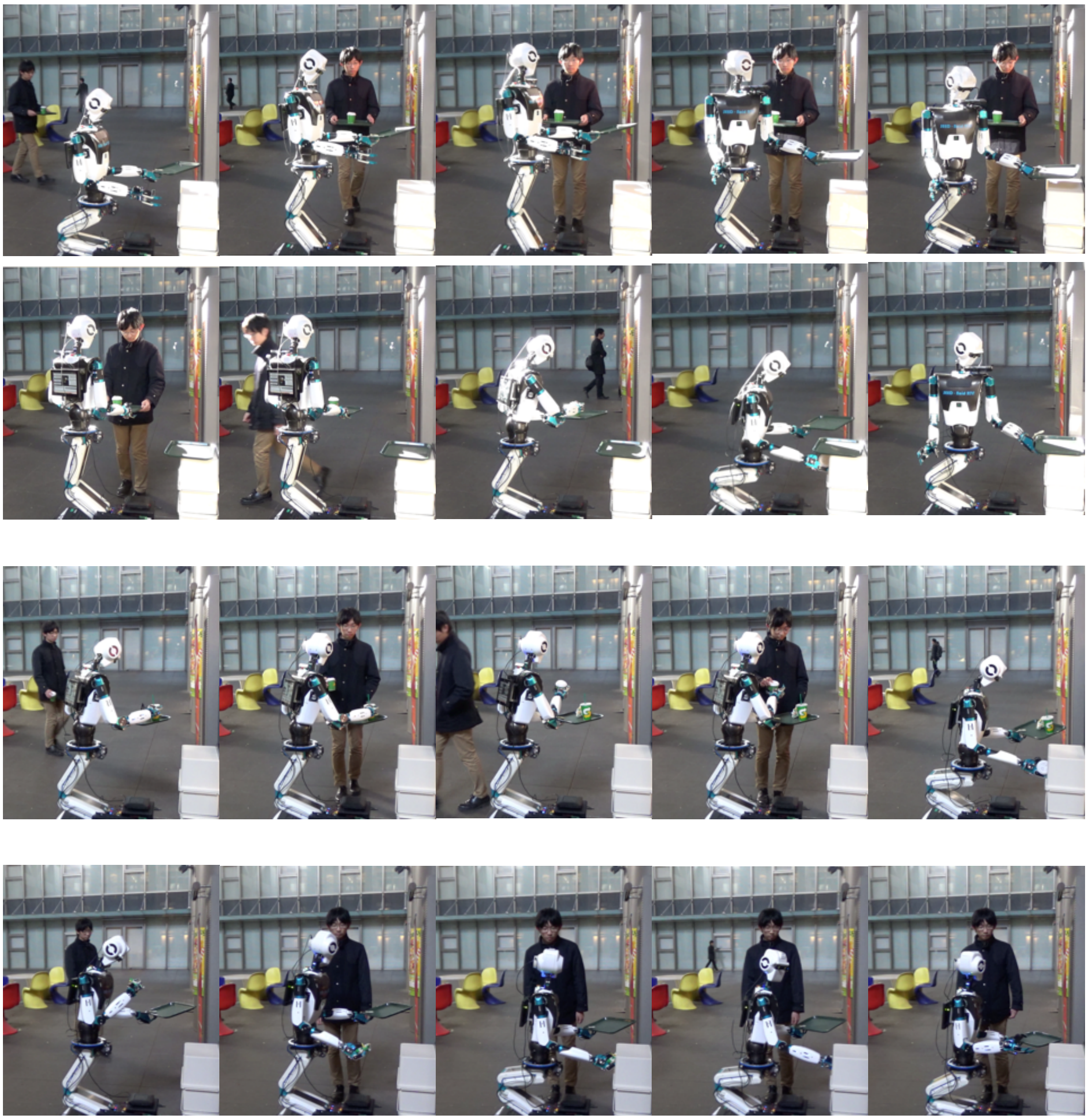


Fig 5.8: An example of a task with different initiation patterns. Top two row shows accepting once and then postponing an interaction to receive a tray. Middle row shows backwarding current action to accept an interaction of receiving garbage. Bottom row shows accepting an interaction and then forwarding a task to meet a social context.

6

The System in Different Situation Scenting Scenarios

6.1 Introduction

In this chapter, we will test our system under different scenarios to find out how and where our system would be beneficial to our society. We will experiment the reaching of the three agreement situations in possible future robotic applications related to navigation and object transportation.

The first part of the chapter experiments no-interaction agreement situations by adding a payment handling interaction to the FCSC competition task. Although such agreement situations are rare for a practical application, and most of the time the robot should politely extend a conflict situation, we will see the effects of reaching such an agreement, whether or not such an agreement is acceptable for the user, and if not, what are the causes of the non-acceptance. We have already explained in the previous chapter that most of the FCSC task is acceptance hard. The payment handling is an action-discussion type of interaction and therefore requires scheduling after the interruption. Instead of reaching a temporary human-to-robot agreement (which would be the usual case when the robot is not at an acceptance state), we will purposefully try to reach the no-interaction agreement unless the robot is at an acceptance state. The robot does not respond but instead reports that it is busy when a conflict is detected.

In the second part of the chapter, we will experiment the human to robot agreement in a guiding task scenario. The application was a practical application that was actually required due to shortage of guiding staffs at an exhibition. We will see whether an autonomous changing in a robot's interaction behavior would be beneficial in the particular agreement scenario. Note that although the robot often started the speech, the situation is human to robot as the robot only tries to interact when a person seems to be trying to initiate an interaction.

The third part of the chapter looks at a robot to human interaction, especially one where a robot proactively acts to begin an interaction. Unlike in the other experiments where the robot initiated an interaction in a situation where a person was usually engaged due to the setting (exhibition) or due to the previous context (robot coming back from a previous order), in this section, we discuss whether robot to human interaction would be acceptable when there is no sign of engagement. After going through the three sections, we will conclude with the possibilities of robots with integrated skills, and evaluate how our system architecture would be beneficial to the society.

Table 6.1: Task procedure description of the payment and sandwich task.

action	description of acceptance
payment	
initialize	required: no collision
pose body toward person	
loop until $request = finish$	
handle payment	
loop	
undo pose body toward person	
sandwich	
initialize	required: both hands free
loop until $items = \phi$	
pose and recognize in $shelf$	conditional
pull *sandwich* with *rightarm*	hard, collision constraint
recognize with *rightarm*	hard, collision constraint
rotate *sandwich* with *rightarm*	hard, collision constraint
pick *sandwich* with *rightarm*	hard, right hand occupied constraint
place *sandwich* to *var*	acceptance
loop	

6.2 No Interaction Agreement Convenience Store Experiment

6.2.1 Task Setting

The procedure of the task and the acceptance state of each procedure action is scripted in the table. Note that the acceptance state and constraint described in the description refers to the state after the action has finished. There are two scripts; the sandwich task that is the default non-interaction task the robot is doing, and the payment task that is the interrupting user request.

For the non-interaction task, we apply hand occupancy constraints and collision constraints. As mentioned in the previous chapter, an action should be divided whenever there is a change in constraint. The script in the table follows this rule. To experiment no interaction agreement conditions, we have set actions in the sandwich task as mostly collision constraints. This enables a long manipulation scenario where a robot does not react most of the time. Here, we have assumed that the robot is continuously operating between shelves until picked a sandwich. Note that the experimental constraint condition is different from the actual sandwich task, and was a purposeful setting for the experiment. The actual sandwich task was not between shelves and there were open space like a tabletop task (assuming that shelves in convenience stores can be pulled out during manipulation). If we were to follow the actual setting, the *pull sandwich* is acceptance and *recognize with arm* is conditional.

We tested the above task as part of an open demonstration that introduced the capabilities of the robot. A presenter of the demonstration who had little knowledge of the system tried to interrupt the non-willing robot during the sandwich task. Since the task was done at an open demonstration with some audience, the presenter had to seriously interact with the robot. The presenter however, had knowledge that the camera was doing some scenting of human-robot interaction, and that the robot was running autonomously. What the presenter did not know was when the robot would actually interact, and therefore, interacted with the robot at random timings where the presenter thought the robot would react. Two presenters explained the demonstration with each presenter doing a minimum of at least three trials.

6.2.2 Remarks on Implementation on Postponing Behavior

Every time the robot detects a H_p-R_n conflict, the situation will trigger the initial interaction callback. The actual postponing behavior (e.g. say ”please wait”) is implemented in this initial interaction callback (remember that by definition, this is indeed an initial interaction that tries to reach a H_n-R_n agreement from a conflict state). The interaction callback (after the initial callback) will never be called when trying to reach the H_n-R_n agreement.

6.2.3 Technical Results

We provide the outcome images of one of the trials in Fig. 6.1 . In the second image in the figure, we see that the presenter stares at the robot. In the third image, the robot scents this staring and speaks ”please wait.” In the fourth image, we see that the presenter decides to pick some item on the shelf before trying to re-interact with the robot. In the fifth image, the presenter re-tries to interact with the robot. The robot scents this situation again, but this time, the robot finds that it is at an acceptance state and therefore, accepts the interaction with ”yes ma’am.” The robot handles

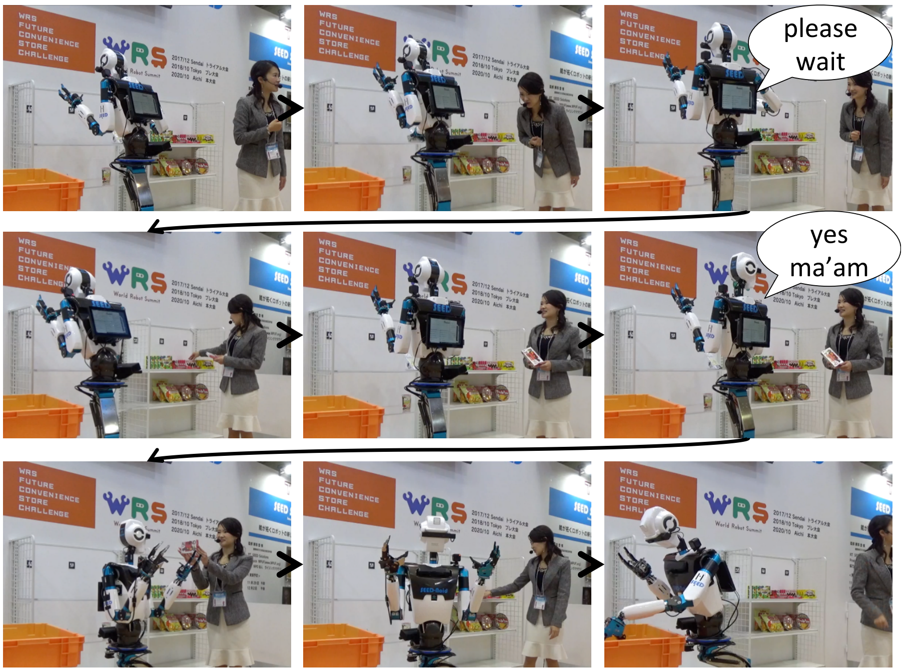


Fig 6.1: A robot postponing an interaction with a non-expert.

the payment task in the seventh image and returns to its task in the eighth and ninth image. Since the robot has finished the sandwich task, it begins a queued storing task.

From a technical viewpoint, we see that our system successfully postpones an interaction task and accepts an interaction with a non-expert. In addition, we found some interesting results regarding the situation estimation. In the demonstration, the presenter looked back and forth between the robot and the audience. This was a natural head movement at a demonstration, as the presenter had to look toward the audience to explain what was going on, but also look at the robot for shared attention with the audience. The estimation results for such situation are shown in Fig. 6.2 . From the bottom graph in the figure, we see that the robot scents the situation correctly; the presenter is not willing to interact even though the presenter looks toward the robot several times. This is one of the advantages of using our proposed method when compared to more naïve approaches where a robot starts interacting once it sees a person looking toward the robot. In the upper graph, we see that the robot scents that the presenter may be willing to interact, and therefore, responds by looking back at the presenter. However, since the presenter was no longer looking at the robot, or looked away from the robot just before the robot had responded, we see a drop in the posterior

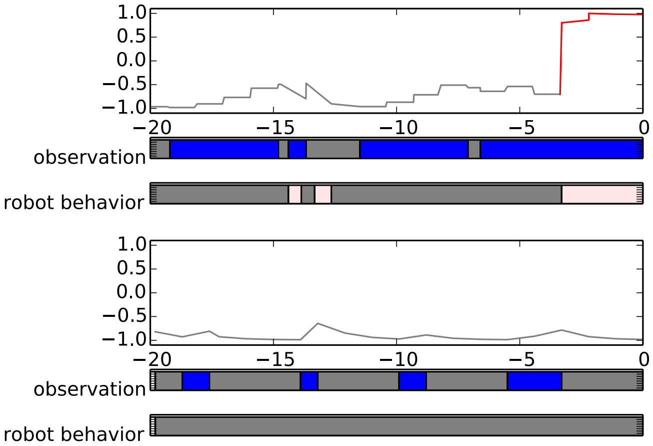


Fig 6.2: Estimation of the presenter's willingness (posterior probability) while the presenter looked back and forth between the robot and the audience. -1.0 indicates not willing and 1.0 indicates willing. Gray line denote H_n, red line denote H_p. Blue blocks in observation indicate looking toward, gray blocks away. Gray blocks in robot behavior indicate R_n and pink R_{r0}. The graph shows that our system avoids naïve interaction timing and enables accurate task decisions.

probability. This shows that accounting robot behavior for situation estimation is beneficial when a person may be looking in multiple directions.

6.2.4 Study Results

We collected feedbacks from the two presenters. We provided open questionnaires. In one of the questions, we asked how many times would postponing an interaction be acceptable. In the experiment, we have assumed a manipulation task setting with mostly constraints. Therefore, the robot said, ”please wait” zero to three, four times depending on how it scented the interpersonal situation, and depending on the timing of the presenter to do the interaction. While one presenter did not have concerns on the number of times the interaction was postponed, the other replied that up to two postponing was acceptable but no more. A comment we had from one of the presenters was, *I was puzzled sometimes as the robot did not enter the payment task even right after the robot had said ”Yes ma'am.”* (comment translated from Japanese).

This happened when the interaction was much shorter compared to the non-interaction task (in this case, the interaction was the ”Yes ma'am” response and queuing of the payment task when accepted a H_p-R_r agreement). The user had to wait a full one action to finish before the next task (payment task) began. The learning here is that, some human-to-robot interactions are not actually parallel and have a severe timing. Indeed, the phrase ”Yes ma’am.” indicates that the robot is going to listen to the person, and it would be strange if the robot kept on doing its task in parallel.

The results of this experiment indicate that a non-willing robot is accepted under some circumstances. What is more important than avoiding non-willing responses, is to avoid false willing responses. How much action remains in a task should be tracked and the robot should honestly report that it is going to take some time before entering an interaction.

6.2.5 Discussion

Perhaps the result is limited to repeating customers or the customer’s personality. In the experiment, the robot immediately entered an interaction in some trials, therefore, the presenter may had the impression that she was not interacting in the right timing in the trials where the robot postponed the interaction.

Perhaps the critical part of the agreeing-but-continuing-task situation was that, to the presenter, why she was kept waited was not understandable. When the robot was warned that it was busy, this was understandable to the presenter, and gave time to the presenter to do something else e.g. talk over the audience or taking an item from the shelf.

6.2.6 Handling Long Constrained Actions

There are few solutions we may come up with for the problem we encountered on *false acceptance*. First, we may set the action as acceptance hard if that action is mostly under an acceptance hard state. In the experiment, we have set the acceptance state of an action according to what would be the state *after* the robot finishes its action. A more appropriate tagging would be to set the action as acceptance hard followed by a dummy action that is acceptance. The problem with this approach, is that, depending on timing, the robot may say ”please wait” then immediately ”Yes ma’am.” However, this is better than saying, ”Yes ma’am” and having the person wait for an interaction.

A second approach to the problem would be to pause task actions in the initial interaction phase, and then add extra phrases such as ”In a moment” if the task has more to progress (this is similar to the extending conflict strategy, except, instead of looking at the remaining action sequence, the robot has to look at the sequence within the action). The problem with the second approach is that, the progress may not be possible of obtaining with some of the control systems.

A third approach would be to make each action shorter (instead of sending joint trajectories at once, divide each trajectory to sub actions). For the third approach, the robot should continue moving at the initial interaction phase and only pause when an interaction is accepted. Otherwise, the robot may stop its motions too often, every time it scents a chance of interaction. When the third approach is not possible, the general approach would be the first one.

6.3 Training Human to Robot Agreement Guiding Experiment

6.3.1 Task Setting

The task setting of the experiment is described below: a robot is in idle mode. When the robot scents a H_p-R_r situation, the robot will begin the guiding task. Once the robot finishes its task, it will return to its home position and idle mode. Pictures of the task is shown in Fig. 6.3 .

From which direction a person would approach the robot was unknown beforehand. The field of view of the cameras was not enough to capture people coming from different directions. Since our engine allows arbitrary binary hints for observations, we have used a switching observation between a robot's base laser and camera. That is, use a base laser that detects nearby legs to scent a person approaching from any direction, and then, after looking back at that direction, use the head camera to confirm the situation. For the leg detection, we have used the methods by Leigh et al. [72]. Pictures of the used observations is shown in Fig. 6.4 .

Theoretically, the probability model for this switching observation should slightly differ from the camera-input only models used in the other experiments. However, we may also say that the probabilities should be similar as, in this approaching case, a person's standing position (leg detection) has high relevance to that person's goal.

We have ran a runtime training of the model to see how the scenting model will change overtime. In addition, we evaluate whether the training improves the performance of the robot, both from a technical perspective using F1 scores, and from a user-centered perspective by rating whether the interaction went well.

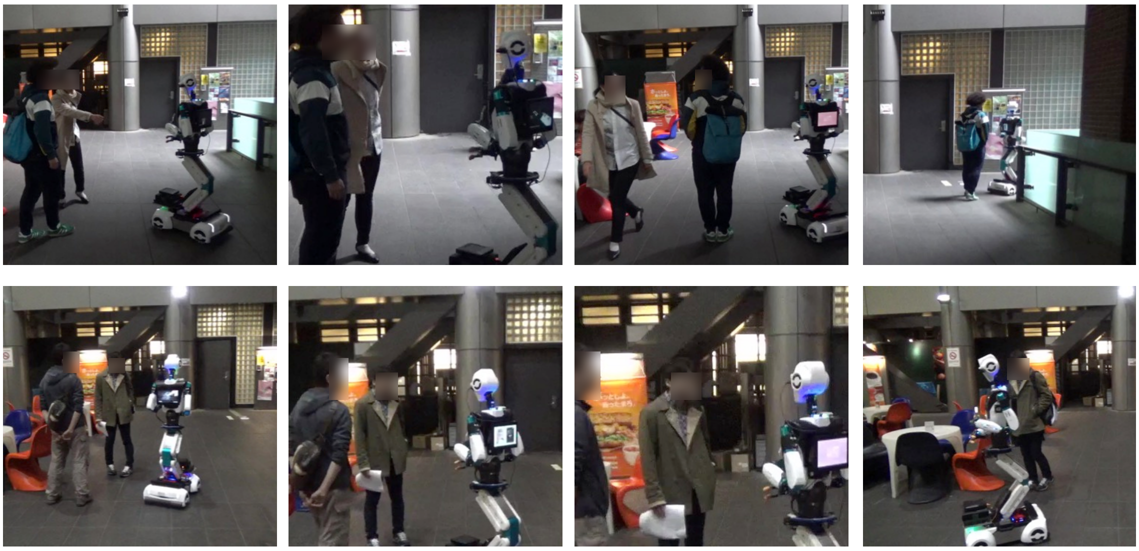


Fig 6.3: Pictures of the guiding task. The top row shows an instructor leading a customer to the robot, and the robot initiates a guiding task. The bottom row shows the robot returning from guiding a different customer, and then immediately initiates the guiding task for a waiting customer.

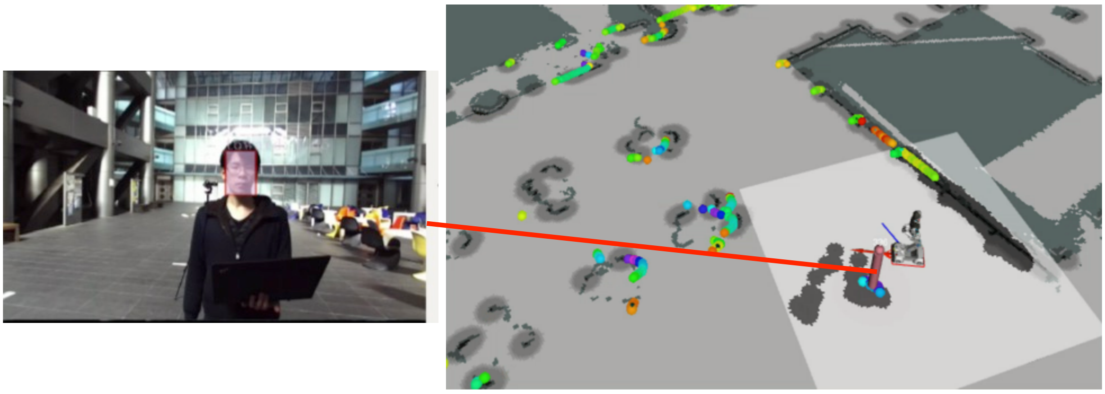


Fig 6.4: Screen capture of the guiding task observations from base laser and head camera.

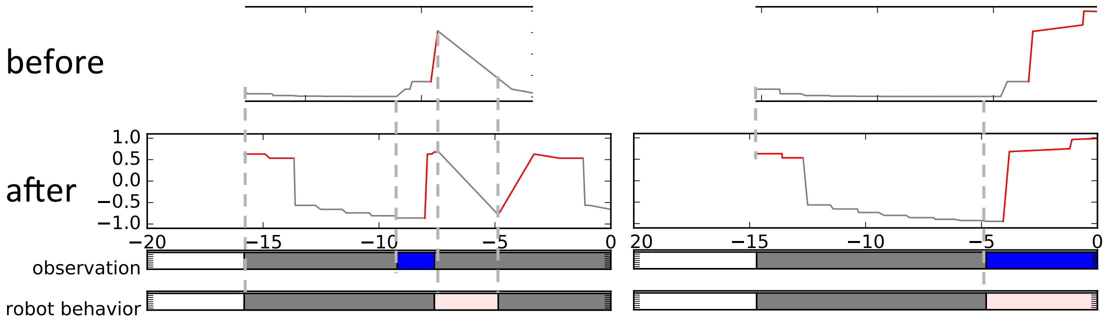


Fig 6.5: Comparison of estimation results before and after the training. The top row shows estimation of before the training and the bottom row shows the estimation after the training. The left side shows the estimations for a non-willing person, and the right side shows the estimations for a willing person.

6.3.2 Training Setting

We recorded the observation sequence and the timing of the robot's action sequence. We provided interaction true labels when a person was in front of the robot for a while after the initial interaction. We provided false labels when the person went away after the initiation. Note that these labels were automatically annotated from the system. The system updated the model after every five interactions. When the robot failed to initiate an interaction the first time but succeeded the second time right after the first failure, we provided true labels for the failed interaction. This meant that the robot revised its estimation of its failures.

6.3.3 Overall Results

We collected a total of 172 interaction initiations in three days (excluding people who just passed by or people who came at in invalid robot state such as while changing the batteries). The interactions included situations such as: the person was guided, the robot failed to initiate the interaction, and where the initiation was successful but the person decided not to use the guiding robot. In some of the interactions, a staff led a visitor in front of the robot.

First, we compare estimation results before the training and after the training. Fig. 6.5 shows estimation on the same observation sequence. Here, we use an actual observation sequence that was collected from some of the visitors. From the figure, we see a clear difference. Before the training (the model from the original dataset), the probability that a person is willing to interact increases only after observing a few frames. In contrast, after the training (the model using parameters collected after running a guiding task for three days), the model estimates a high chance that a person is willing to interact once it detects a person, but then the estimation drops toward not willing afterwards. In addition, we notice that before the training, the model estimated the willingness as false if the person was not looking toward the robot after the robot looked back. After the training, the model shows a similar estimation once, but will increase the chance of human willingness right afterwards.

From these results, we see that, first, when a person suddenly comes close by (which happened when two people were approaching and one person was hidden behind the other), the trained model reacts faster than the model used before training. Second, the robot keeps the characteristics that lead to correct estimations. For example, we see a similar rise in the estimation when the person is willing to interact for both models. Third, the trained model tries to recover from a chance of (initiation) failure when it detects a human non-willing state. Example situations where this happened was when a person approached the robot but then started looking at a nearby sign, then, re-approached the robot afterwards.

After the training, the robot responded faster to these types of interactions. From the result, we see that how the robot interprets these observations change due to training. Also, we see that the robot trains itself especially on the estimations before and after showing a sign of reaction. This shows the importance of collecting data not only when the robot is reacting, but also when a robot is doing a non-interactive behavior.

6.3.4 Technical Results

Table 6.2: The task initiation F1 score during the training of the guiding robot.

	day1	day2	day3
F1 score	0.85(0.72)	0.88(0.84)	0.89(0.84)
precision	0.87(0.75)	0.86(0.82)	0.86(0.80)
recall	0.84(0.69)	0.90(0.86)	0.92(0.88)

Table. 6.2 shows the F1 score of the first day (starting the training), the second day (middle of the training), and the third day (finishing the training). We see that the scores become slightly

higher as the training proceeds. The scores in parentheses indicate the score when excluding the people who decided not to use the robot after listening to the robot's speech. The result shows that the model is becoming more accurate in scenting whether a person is willing or not.

6.3.5 Study Results

Table 6.3: Rating percentage of the interaction by two researchers.

	day1(good:fair:bad)	day2(good:fair:bad)	day3(good:fair:bad)
rater 1	0.28 : 0.24 : 0.48	0.28 : 0.36 : 0.36	0.33 : 0.44 : 0.23
rater 2	0.38 : 0.31 : 0.31	0.57 : 0.20 : 0.23	0.68 : 0.17 : 0.15

Two researchers have looked over all the interaction that happened in the three days and rated the interaction initiation as good, fair, or bad. An example of a good interaction was where the interaction went smoothly and the person and robot directly entered an interaction. An example of a fair interaction was where the person had to try to catch the robot's attention before entering an interaction. Bad interactions included not responding, responding too late and users going away, and reacting to people passing by. Table. 6.3 show the rate percentage of each day. From the table, we see that the raters rated fewer interactions as bad as the training proceeded (Cohen's kappa score of 0.51, 0.69, 0.41 for each day respectively). This means that a robot's interaction behavior can be trained to increase the number of acceptable interactions, by using probabilities on sequential observations.

6.3.6 Discussion on Raters

When we look at the number of good and fair interactions, we see different opinions depending on the rater. One rater rated more interactions as good as the model was trained. In contrast, the other rater rated the same amount of good for the model in the middle of training and the model finishing the training. Although the scenting accuracy increased, one rater found that the speech timing could be faster (should enter an interaction with a shorter initial interaction duration). This is one of the drawbacks of trying to be accurate. In general, taking more observations allow more accurate estimations, however, the person will lose interest in the robot if the decisions are slow. The trained model sometimes collected observations as much as possible, but has made decisions right before the user would lose interest. Although such a strategy would increase the F1 score, it might not increase user preference scores. However, this rater had more experience with the field of human-robot interaction. The other rater with less experience found that if the robot interacted before the person went away, it was good.

Another interesting difference between the raters is that, the rater with more experience found that in some situations, not interacting was reasonable (a person looked at the robot for a few seconds and soon went away) and therefore rated the interaction as fair. The rater with less experience found that the robot should also interact in such cases, therefore rated the interaction as bad. Training does indeed correct the robot's scent towards an interaction; however, preference might dependent on different factors such as a user's previous experience or expectance toward robots.

6.3.7 Discussion on Application Usage

When people are guided by the robot, some are actually impressed with the robot's capability. Yet, about half the visitors decided not to use the robot once hearing the robot's capability after an initial interaction. This included situations where people did not understand the robot's language, but also situations where people just wanted to take a picture of the robot. In addition, since the experiment was done at an art exhibition, some people have looked at the robot as one of the art works and may have lost interest because they found that the robot was not the art they were looking for.

The training results did not correlate with how much the robot was actually used for its physical capability. The number of people guided was highest in the second day where the model was in the middle of the training. Perhaps this is because the reasons for the people not using the robot was due to speech content or a person's initial mindset toward the robot, rather than the robot's behavior at an initiation. Similar results have been seen in Chapter 3 as well.

To make a robot more acceptable, perhaps there could be a staff to introduce the robot. When a staff introduced the robot as a tool to help guide the visitor, all visitors accepted the robot and were successfully guided.

If a staff initiates the interaction between the visitor and the robot, it may seem as if we would not need an automated initiation function. However, we have observed in our experiment that the staff will have more time talking and explaining things to the visitor if the robot initiates the interaction automatically. Unlike a tablet interface that takes away a staff's attention from the visitor to the screen, a robot interface less distracts human-human conversations and could seamlessly begin the guiding task just by being introduced. Not only does removing the tablet lower the cost of the robot, but it also allows a more smoothly acting interface.

6.3.8 Benefits of Training

When the robot is not responding, people do not speak to the robot (only in a few occasions, a person will say ”hello”). The first thing people (mainly adults) do is look around the robot as if trying to find a switch to activate the robot. However, once the robot responds, people often wait and stay still until a robot speaks a phrase. In this sense, a robot has certain control over the interaction if initiated the interaction at correct timing. However, if the robot fails to initiate, the human behavior becomes unpredictable. Both the human and robot will enter a confused state and most likely fail to enter an interaction. This was often seen in the first day at start of the training. We may look at this result the other way around. Revising estimations revise a robot’s behavior, and a revised robot behavior will provide better estimations.

As we have seen, the main advantage of using a complex model that trains and estimates interpersonal situations, is that, the robot could better fit to interaction scenarios, and self-supervise itself when it encounters an interaction failing pattern. Such failures may happen, as how a person will act in front of the robot is not known beforehand.

6.3.9 Limitations

We must be aware that the model was trained in an open space and a lot of people were passing by the robot. The open space also meant that the robot would detect people (legs) approaching from a far away distance. In addition, the model captured some of the flow patterns where a hidden person suddenly appeared, or when a person looked at a sign before starting the interaction with the robot. These were typical flow patterns in the scenario. Depending on the scenario, we may have different flow patterns. We may also have different observation patterns even when using the same combination of sensors. For example, if this was a small room and far away detection was not possible, we would have completely different observation patterns. Therefore, the trained model most likely fits only the scenario that it was trained. However, as we did in the experiment, we may use an already trained model to generate initial interaction behaviors to begin with.

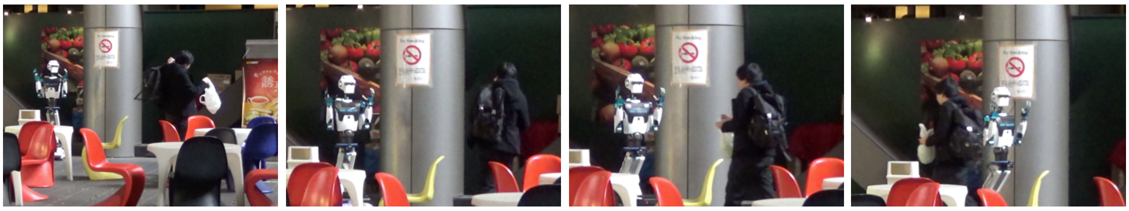


Fig 6.6: An example where an approaching robot fails to initiate an interaction.

6.4 Approaching Robot to Human Agreement Experiment

6.4.1 Experiment and Results

The last part of this chapter will discuss robot to human interaction. Unlike the experiments in the other sections with specific task settings, we will pick up some of the other experiments we have done but have not yet discussed in this book. These experiments were those that *failed*. We will discuss mainly on these failures and conclude with the limitations of robot to human interaction.

The first experiment was done as a side experiment of the Chapter 3 interaction-only public domain experiment. In the Chapter 3 experiment, the robot starts an interaction with R_n, whereas, the side experiment started with a R_p willingness state. The robot tried to catch a walking-by person's attention by speaking. The robot even tried moving toward people passing by. However, no one interacted with the robot and almost all people ignored the robot. One person stopped and looked back at the robot but soon walked away. Perhaps the problem was that both the person and robot were moving around which made the interaction initiation difficult. In addition, people were walking by, meaning that they might have been busy, and the situation was not right for interacting. Even in human-human interaction, we sometimes ignore people giving out fliers out in the road. However, our second experiment rejects this hypothesis.

The second experiment was done at a forum in the university (the same location where the robot navigated people in the previous section). We have used the same robot as the other experiments in this chapter. Instead of navigating, the robot moved toward a person standing up from a table and asked if there was any garbage the robot could throw away for the person. This time, the robot was moving around and the person was mostly at the table preparing to leave. Some people started walking away from the table, but this was after the robot started approaching them.

Again, no one interacted with the robot. One person faced away from the robot when the robot approached. Another person was walking in the direction where the robot was approaching from, but passed through the side of the robot as if trying to avoid it (Fig. 6.6). One group noticed the robot and looked back, but seemed surprised and became speechless. Unlike the first experiment, people were in place and had a chance of interacting. Compared to the situation in the first experiment, there would have been more chance of an interaction happening if it were a human-human interaction. Yet, the results show that people do not interact with approaching robots.

6.4.2 Discussion

It seems that people feel odd about an approaching robot. An approaching machine is out of their expectations, they are facing something unfamiliar, and they are puzzled with the situation. In addition, not everyone accepts a robot. We have seen that even in human to robot interaction scenarios, some people are not willing to interact with the robot and walk away once seeing the robot respond. There is a chance that we are trying to interact with the non-accepting people (which could be the majority), and therefore, an approaching robot might be inappropriate for our society.

However, in some cases, it is beneficial for a robot to approach a person. For example, in the Chapter 3 in-lab experiment, the engineer asked the robot to bring an item, but was not aware that the robot brought back the item. In this situation, the person asked for the robot's help and the approaching machine was within his expectation. This is a successful example of a robot-to-human interaction.

The other possibility of robot-to-human interaction is to reach an interaction faster. If a robot comes near a person while a person approaches the robot, the person will have to walk a shorter distance to reach the robot. This may seem as if the robot must know the intent of the person. However, we know that most of the time, people will just ignore the approaching robot if they do not need to use the robot. There would not be many side effects if the robot correctly decides not to begin an interaction while it is approaching the person. Therefore, we may have some kind of trigger to have the robot proactively start moving, and then we may use the situation engine to find out if the proactive action was correct. For example, in a garbage-collecting scenario at a food court, we may have the robot start moving once detecting a person leaving the table. While the robot approaches the person, the robot could observe the looking direction of the person. If the person seems to be not willing from the observations, the robot could stop its approach and decide not to talk. If the person seems to be willing, the robot could proceed to reaching an interaction agreement.

We have found in the experiments that as people start to familiarize with the robot, they try to have more control over the interaction with the robot (they try to initiate the interaction with the robot instead of waiting for the robot to speak). In order for the robot to still have some control over the interaction, a robot might need to act beforehand.

6.5 Conclusions

The experiment results show that the robot must react and initiate the conversation. The reaction helps users to use the robot more intuitively, but also helps the robot to have more control over the interaction. In order to achieve such control, scenting is a very important function as it allows the robot to react at an appropriate timing where the interaction can be controlled, but does not stress users who are not willing to use the robot. The system structure is indeed essential for providing better usability in initiating robot task skills.

However, by evaluating our system in different situations, we find that not all interaction situations are appropriate for the robot to have control. The robot should only have control over interaction situations when people are able to predict the robot's behavior. For example, robot to human interaction in the wild is one of those situations where people are not able to predict or understand why the robot is approaching the person. We have not found any successful tries in such situation. Another example is the false agreement.

The other finding from this chapter was that, scenting can be trained per scenario to provide better control over interaction patterns in a particular scenario. The recall score has increased by 0.08 in total, and increased by 0.19 for the users who actually used the robot's task skill. In addition, the number of bad-rated interactions has decreased by an average of 22%.

7

Conclusion

7.1 Conclusion and Findings

In Chapter 1, we have pointed out the limitations and failures of current social robots as not meeting customer expected skills, and therefore, our goal of integrating interaction skills with other robotic skills such as navigation and manipulation.

In Chapter 2, we have clarified the novelty of solving interaction under a total setting where other robotic skills are active. The unsolved problems included managing of physical and non-verbal interaction in the initiating stage under the situation of interaction goals not being aligned.

In Chapter 3, we have achieved a model that is able to estimate the interpersonal situation between the human and robot at an interaction initiating stage of non-verbal behaviors. The proposed method of taking into account the probabilistic relations between the human behavior and the robot's behavior has succeeded in estimating a switch in initiative, scent willing users, detect non-willing users, and achieve an F1 score of 0.821 despite using only a sequence of simple binary observations, and has been proved effective.

In Chapter 4, generality reduces cost, and we have indicated the barebone task architecture that provides baseline solutions to the task skills requested in our society. By clarifying a hardware solution of using a middle mobility layer that minimizes the number of required joints to execute a continuous position control for picking and placing objects in a structured environment, the system has succeeded in reducing on-board calculation time by only dropping 28% of the convenience store task, and holding the task error rate to 30%, placing second in the competitions despite not targeting for the specific task.

In Chapter 5, we have achieved a scheduling algorithm that processes the required transition from task to interaction under the physical, social, and objective context. The proposed method of combining constraint-based task acceptance, a person's dialogue objective, and the robot's interaction willingness from prior context has shown to be practical in preparing an interaction task in the following situations: removing social constraints during information-transfer objectives, planning the minimum number of actions to remove physical constraints for action-discussion objectives, and postponing interactions under inappropriate social proximity.

In Chapter 6, we have proved by going through a total of 172 interactions in a real guiding application that, the robot taking the initiative through physical behaviors help prepare people for an interaction. In addition, we have indicated that the scenting mechanism trains and fits itself to the specific task context through self-update and self-revising, which has increased the recall score in the particular scenario by 0.08 and decreasing the number of bad-rated interactions by an average of 22%.

As a result of the above achievements, we have achieved an architecture that is able to autonomously initiate and automatically prepare an interaction from different task states under different interpersonal and task situations including those under non-verbal behaviors. As we have indicated from our experiments in the wild, such architecture is inevitable for robots to enter our society, provide an entrance to using the physical capabilities of the robot, and be socially acceptable.

7.2 Outcomes

Filling the gap between society. Interaction task-based scenarios using the most fundamental capabilities of robots have not been well discussed over the past. This book has covered such missing areas to fill in the gap between task-based expectations in society and handling of interactions in those task-based scenarios.

Beyond current social robots. Adding task skills will obviously be beyond current social robots. Yet, even as a social capability, our solution provides more than current robots. Using our solution, a robot may lead the conversation instead of waiting for its name to be called; which provide more control over user expectations. A robot may reason between who is more engaged when there are multiple people, and provide eye contact from person to person. These are important social skills to provide better usage and better acceptance toward society.

Answers to questions in the industry. As part of a graduate school leading program (GCL-GDWS), we have investigated on what people from the industry (hardware vendors in the country) find difficult when developing software for multi-purpose robot systems. Our investigation has found that, what people from the industry find most difficult is how to write codes with lot of *ifs* and how to achieve complex tasks. For example, how to stop a robot passing by and request something different. The work in this book has provided the answers for combining non-interaction and interaction tasks by providing a situation scenting capability embedded system under the total setting. We have shown that different interaction scenarios can be achieved with this single technology of situation scenting. In addition, we have confirmed that the architecture works on several robots including the Seednoid, HSR, Pepper, and Fetch.

A gateway to the next generation of robot business. We have shown through our experiments that people are still not at the stage of accepting robot-to-human interaction, at least in a public setting. However, results also show that more than 50% of public users accept human-to-robot interaction, and 100% under guidance by a human staff. The achieved technology will act as a core to such scenarios in the business market. We may estimate at least $5 billion market size in Japan that would benefit from this technology (assuming the robot's capability to be 20% of the human's capability for 50% expense in the first year, and assuming 5% of the market uses the robot).

7.3 Limitations and Future Work

Although we have clarified on how the social context influences the robot's interaction acceptance or postponing in the total setting, the approach is not yet autonomous. An open area in research is teaching robots the social manners appropriate for a particular setting. While the usual theme in this area is based on preferable motions, we believe from our findings that manners for postponing an interaction are also as important, especially for a task robot.

One remaining issue that exists for robots to enter society is, automating the generation of a task skill. Although, vendors provide code samples for using the robot, *how to code* depends on each scenario, and the vendor has to *mentor* each client to provide the appropriate solution. This leads to an over cost on human resource on the vendor side. As the robotic skills become advanced, a more data-driven solution where the enterprise feeds data, and the flow to be generated from that data is necessary. To reach the consumer in an actual market, we must solve the problems between the vendor and enterprise. This is another challenge we must investigate before marketing.

Acknowledgement

This thesis was written and supervised under professor Masayuki Inaba. Inaba sensei has provided the author many valuable advices, directions, and questions challenging the author’s thoughts. There were many nights where we discussed for hours just before the last train. Some of the important advices were to have other people evaluate the author’s system, and to do an experiment that prove the importance of the system’s core element by showing how the element functions to solve the many problems in one gigantic scenario. The advices have lead to the experiments in Chapter 5 and of course, the title of the thesis. Inaba sensei has also helped in organizing the author’s work so that it fulfills both as a project of GCL and as a research of the department.

Some of the works in this thesis was supervised under Katsushi Ikeuchi at Microsoft. Ikeuchi sensei has helped in theoreticizing and clarifying the ideas in Chapter 3. His well-known comment ”I cannot understand” was the most simplist and powerful word in helping the clarification. In addition, the name ”situation engine” was named by Ikeuchi sensei. Without Ikeuchi sensei’s help, the author would never have had connections with the people at Microsoft.

The author also thanks the support by The University of Tokyo Social ICT Global Creative Leader (GCL) program and the project organizer professor Yasuo Kuniyoshi. The comments by Kuniyoshi sensei during the GCL presentations have encouraged the author to focus on the main theme of initial interactions. GCL has also helped in organizing workshops to understand the connection of the author’s work and the actual society. In addition, GCL has helped in making connections with other researchers and students of different fields. Without the help of GCL and professor Tohru Asami, the author would not have met Ikeuchi sensei.

In March 2014, associate professor Fumihide Tanaka from University of Tsukuba organized a trip to the HRI conference under the support of GCL. The conference has provided the author to learn about the field. Since then, Tanaka sensei has helped the author connect with the other researchers in HRI. The author thanks Tanaka sensei for his valuable support.

At Microsoft Beijing, the author got support from Zhaoyuan Ma, Fan Shi, Masaru Takizawa. Without Zhaoyuan and Fan’s help, the author would never have survived Beijing. At Microsoft Redmond, the author got support from the following team members: David Baumart, Yutaka Suzue, John Lee, Jordan Kravitz, Ryo Ishikawa, Jun Takamatsu, and Meghna Gupta. The author also got support from other groups in Microsoft: Sean Andrist and Dan Bohus. Sean snd Dan has helped in sharing the interaction videos used in Chapter 3.

The THK Company has helped in providing the hardware discussed in Chapter 4. The author and the company have had a really nice relationship. While THK has helped in achieving the hardware structures we requested, the author and the JSK lab has helped in evaluating THK's hardware and find its potential. The author especially thanks Yasuto Shiigi for his all time support, Takuya Saiki for developing the rotating torso sensor the author had requested, and Masaki Nagatsuka for designing such a wonderful hardware.

And of course, the author thanks the people at the JSK lab. Taichi Higashide has helped the author's experiment remotely while the author was at Redmond. In addition, Higashide has helped in trying out the proposed system, providing feedback, and also with the guiding robot experiment in Chapter 6. Iori Yanokura has helped in some of the experiments and revising some of the code. Yuki Furuta has helped in discussing the unique parts of each other's system and has helped experimenting the situation engine in Chapter 3. Shintaro Hori has helped with experiments at exhibitions and development of some of the tasks discussed in Chapter 4. Yohei Kakiuchi has helped in maintaining the Seednoid repository. Hiroaki Yaguchi, Kotaro Nagahama, Hiroto Mizohana, and Wesley P Chan have helped with the competitions.

The author also thanks the organizers of the competitions discussed in this thesis. Without the competitions, the author would never have been able to evaluate his system. The author also thanks the reviewers of the conference and journal papers submitted by the author. The reviewers have surely had an impact on some of the experiments and some of the terminology.

The author greatly thanks Hirose sensei, Kuniyoshi sensei, Harada sensei, and Okada sensei for providing valuable insights to the author for finalizing this thesis. Hirose sensei and Kuniyoshi sensei has provided advice on discussing more towards what it takes for robots to be accepted in our society, and clarifying how the author's ideas solve for such acceptance. This has helped the author revise the overall structure of the thesis and especially Chapter 1, 5 and 7. Harada sensei has provided advice on looking more into the dialogue field, which has allowed deeper discussions on interaction in Chapter 2. Okada sensei has asked for explaining more on how tasks influence interaction, which has lead to deeper thoughts in the conclusions.

Last but not least, the author thanks his family and parents for their support.

12.7.2018 Kazuhiro Sasabuchi

Appendix

A.1 Evaluation of a Robot using Ball Screws and Stepper Motors

Stepper motors have two advantages when compared to brushless motors: 1) stepper motors require less maximum input of 5 to 10[W], thus it is easier to achieve a longer battery life when the motors are in stall. 2) The platform is able to detect high loads and collision from motor missteps during manipulation. Compared to other sensorless collision detection [27], a mechanical approach ensures the platform to safely stop its movements before breaking its hardware.

Table. A.1 shows the force tolerance of using ball screws and stepper motors. We experimented two poses, elbow bended to 90 degrees (B), and arm straightened at shoulder height (S). The percent in the table represents used electrical current. *f, t, h, w, e* each denote the part detecting a misstep as finger, torso-yaw, hand-roll, wrist-pitch, elbow-pitch. No misstep indicates a tumbling risk. The wheels were servoed during the experiments and measured using a force gauge (FGPX-250H). The values were sufficient for pull-opening doors, picking up a 3lb. warehouse item, as well as bottle drinks.

In addition, the power produced during collision was experimented. The robot descended the right forearm toward a robot impact sensor (KMG-300-75) until a motor misstep was detected. The produced force was measured starting 60[mm] above the sensor at a maximum stroke speed of 60[mm/s] in the elbow actuator. The maximum impact measured was 65[N] at the hand. The value is within doubled safety range explained by [146]. This also provides evidence on how damage is reduced by using stepper motors. The disadvantage of stepper motors is its speed.

Table A.1: The force tolerence of the seednoid platform. Unit in [N]. [121]

direction	30%B	100%B	30%S	100%S
forward	69-f	79-f	72-f	72-f
inward	38-t	83-	23-t	51-t
outward	26-t	46-f	22-t	42-w
upward	45-h	44-e	32-h	46-e
downward	12-h	53-e	30-e	46-h

A.2 Constraints Inside Narrow Bins and Non-centered Fingertips

In general, a grasping surface $S_\varphi \in \mathbb{R}^2$ by a planar hand movement is represented as the following equation:

$$\boldsymbol{S_\varphi} = \begin{pmatrix} L_h \sin\theta + r\cos(\theta - \theta_0) \\ L_h \sin\theta + r\sin(\theta - \theta_0) \end{pmatrix} \tag{A.1}$$

where L_h is the distance between the wrist to the virtual root joint of the fingers (a virtual root joint is the summed center of all finger root joints), r is the distance between the virtual root joint to the fingertip grasping position, θ_0 an angular offset of the grasping position (for example, in a common parallel gripper $\theta_0 = 0$ while $theta_0 = 0.78539$ for the gripper in Fig. A.1), and θ is the orientation (pose) of the hand in the grasping plane. Let us assume the situation where the hand is grasping from a narrow shelf. A required fingertip workspace may have an upper bound φ_+ and a lower bound φ_-. Assuming that the robot does not operate on a ceiling, the picking of an object from front side is conducted near φ_+ and the picking of an object from above is conducted near φ_-. The fingertip height from the wrist to φ_+ and φ_- is expressed as the following by using the above equation:

$$z_+ = L_h \sin\theta_0, \ z_- = -r - L_h \cos\theta_0 \tag{A.2}$$

When picking an object from a shelf, we want a smaller $|Z_+|$ for a frontal grasp, and a smaller $|z_-|$ for grasping from above. A centered fingertip ($\theta_0 = 0$) will minimize $|z_+|$ but maximize $|z_-|$. In contrast, $\theta_0 > 0$ would balance $|z_-|$ and $|z_+|$. The human like gripper ($\theta_0 > 0$) is more suitable when compared to an industrial gripper ($\theta_0 = 0$) in this narrow shelf condition.

The disadvantage of the human like gripper is that, it has difficulty in rotating objects. For a robot to rotate an object, the simplest way would be to grasp an object from the top, and then rotate the hand yaw joint. This type of strategy was also seen with the valve task in the DRC by many of the robots. The strategy would also have been beneficial for the FCSC sandwich task. A human does not have this yaw joint but instead is able to do in-hand manipulations.

A.3 Recognition Pipeline in the APC without Training of Items

Fig. A.2 shows an approach of detecting objects without training using pixel segmentation and object detection via a cloud database. This assumes that a robot has some knowledge of its physical context and tries to map the knowledge with a new label. In the APC, this knowledge was the location of the item (which bin the item was in), the color, the shape, and a matching to a similar known object. For the known object matching we used the COCO dataset [75] as our base. For example, a teddy bear was already a known object to the COCO dataset, but the brush was unknown and was referred to as a monitor. Therefore, the robot labeled a *monitor like object that was blue and long and is in bin D* as a brush. During the competition, we were able to detect the glove (a shirt like object that is black and small and was in bin B) using this approach. However, we were only able to grasp one object with this approach, as the location of the object was too vague due to inaccurate bounding from general segmentation approaches.

A.4 A Theoretical Comparison of Two Compact Wrist Structures

This section is a supplementary material to the discussion of wrist structures in Chapter 4. In general, when a spherical assumption holds for the wrist, the pose (tool axis) R_h of the hand can be expressed as the following quaternion:

$$R_h = \cos\frac{\theta_1}{2} + (S_{x_{\theta_2,\theta_3}}\hat{i} + S_{y_{\theta_2,\theta_3}}\hat{j} + S_{z_{\theta_2,\theta_3}}\hat{k})\sin\frac{\theta_1}{2} \tag{A.3}$$

where θ_1, θ_2, θ_3 are the root (wrist) joint angles, and $S = [S_x\ S_y\ S_z]$ a unit normal vector on a spherical surface. θ_2 and $theta_3$ must rotate around a different axis. From the above equation, the number of pose solutions is dependent on the joint limits; especially the joint with the smallest range among θ_1, θ_2, and θ_3. These limits are tied to the hardware structure. It may seem reasonable to design a hardware so that $\theta_1, \theta_2, \theta_3 \in [-2\pi, 2\pi]$. However, this is not always possible especially if we are trying to develop a compact hardware.

With the roll-pitch-yaw structure (Fig. A.3 -I), $\theta_1 = \psi \in [0, \pi]$, $\theta_2 = \phi_y \in [-0.08\pi, 0.14\pi]$, $\theta_3 = \theta \in [-0.39\pi, 0.36\pi]$, therefore, the number of solutions is dependent on ϕ_y. Likewise, with the roll-pitch-roll (Fig. A.3 -II) structure, $\theta_1 = \phi_r \in [-\pi, \pi]$, $\theta_2 = \psi \in [0, \pi]$, $\theta_3 = \theta \in [-0.39\pi, 0.36\pi]$, therefore, the number of solutions is dependent on θ. Comparing ϕ_y and θ, the roll-pitch-roll provides more functionality.

A.5 Hand Designs for Non-prehensile Manipulation

We have explained that common prehensile grasp modes are the fingertip and encompassing grasp. Likewise, we may define two major grasp modes for the non-prehensile actions, depending on the usage of the thumb. The **hook mode**, which uses the thumb for post-contact after a non-prehensile action (e.g. topple and catch), and the **thumbless mode**, which stores the thumb so that the thumb does not disturb the operating finger (e.g. sliding a thin object). We see that the Seednoid gripper has each grasp mode Fig. A.4 . This allows combining various actions in the narrow. For example, using non-prehensile actions will allow the robot to pull out objects that are deep inside the shelf.

A.6 Environment Modeling

The modeling module uses both online and offline information to visualize the environment. Offline prior knowledge of the problem environment is applied via an environment map. Moreover, some objects are better to be defined prior rather than on runtime. A stereo camera is not able to detect walls, and because position of walls often do not change, it is better to input such models as prior knowledge. Likewise, a shelf is often occluded with objects and is also better to be generated from prior knowledge. Other smaller objects should use runtime detection as the position often changes in the scene. However, since only the frontal face of an object is observed through the sensor, sometimes it is better to use prior knowledge to model the shape and use runtime detection for modeling object location. For objects that are modeled for collision, we may not need the precise shape and create a bounding box by analyzing point cloud information.

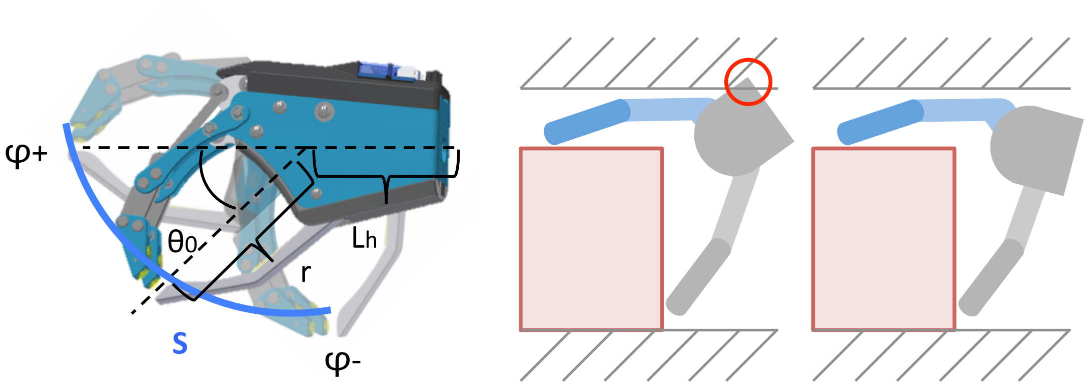


Fig A.1: The Seednoid gripper with non-centered fingertips.

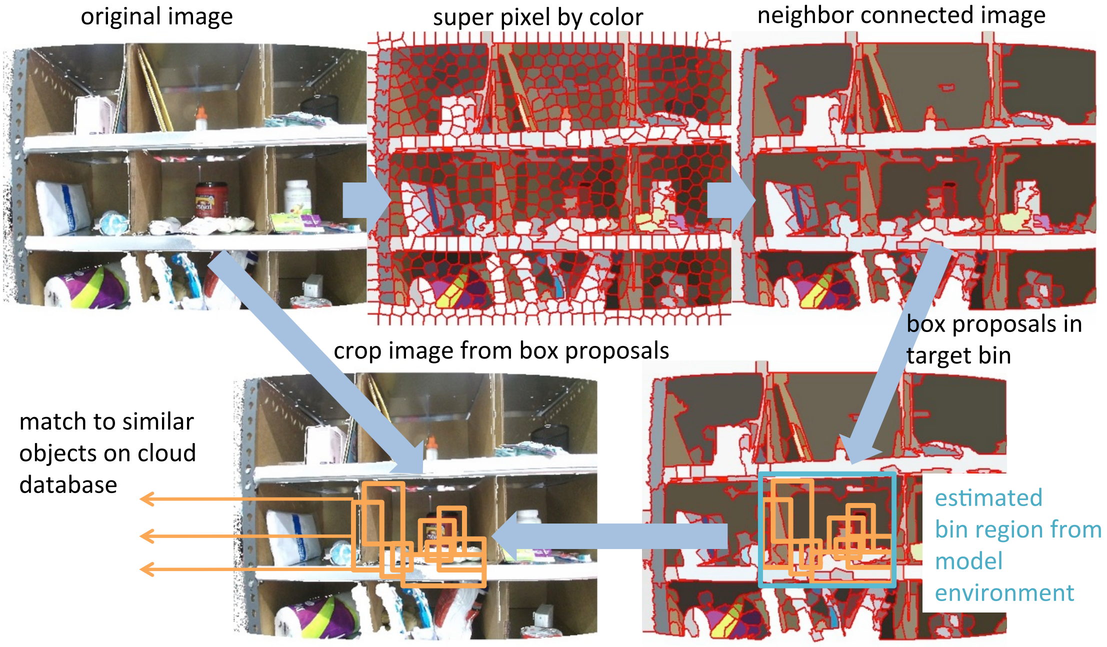

original image
super pixel by color
neighbor connected image
box proposals in target bin
crop image from box proposals
match to similar objects on cloud database
estimated bin region from model environment

Fig A.2: Vision process using pixel segmentation and object detection via Microsoft Cognitive Services.

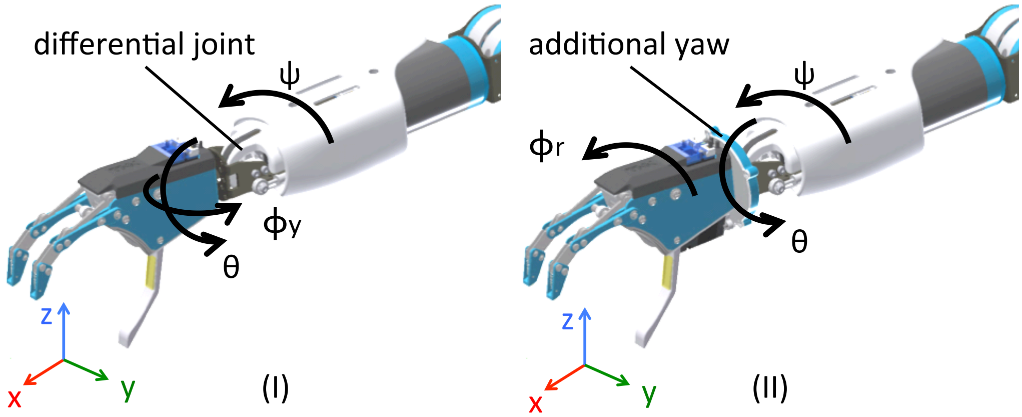


Fig A.3: Comparison of two wrist joint structures. [121]

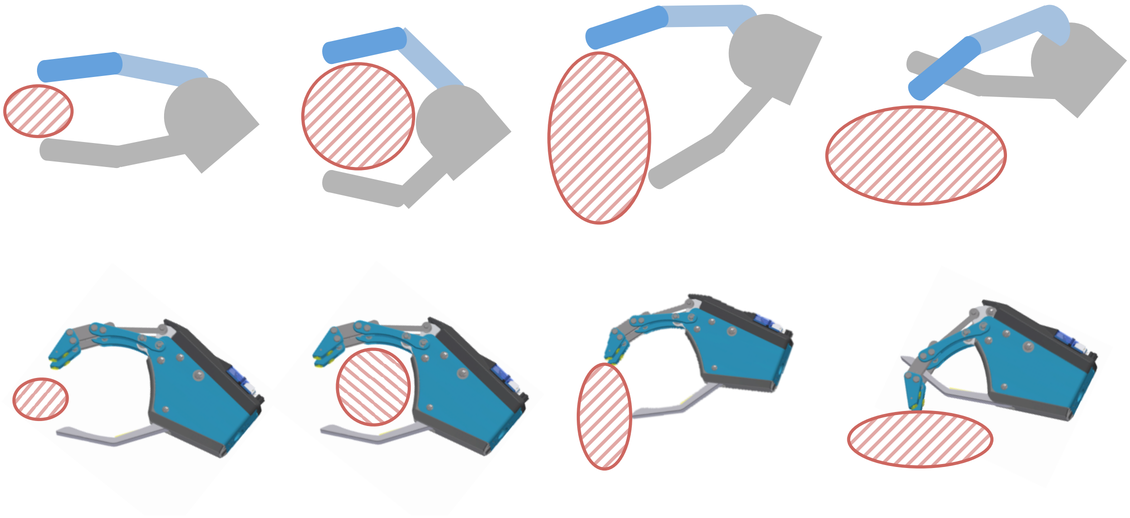


Fig A.4: Grasp modes of the Seednoid gripper based on human anatomy.

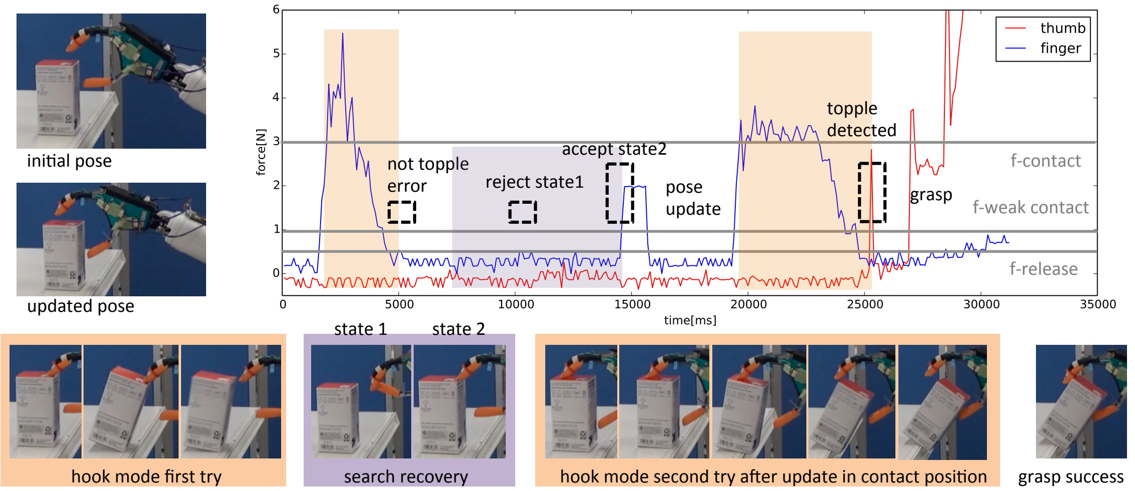


Fig A.5: Example of picking an object with only tactile feedback using a non-prehensile action of topple and catch.

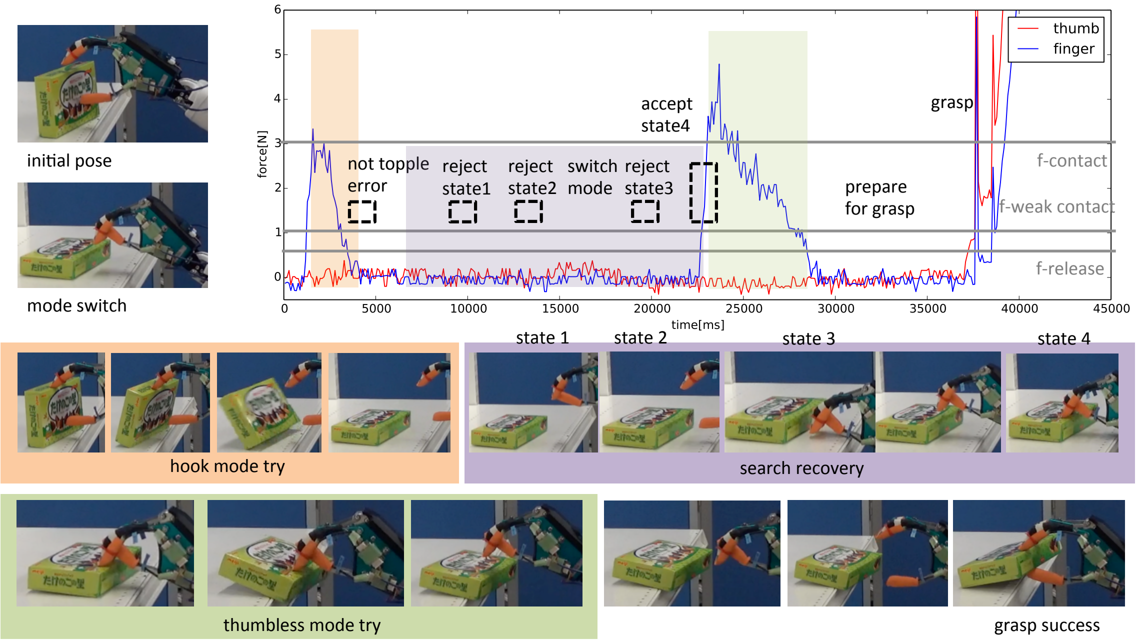


Fig A.6: Example of picking an object with only tactile feedback using a non-prehensile action of sliding then grasp.

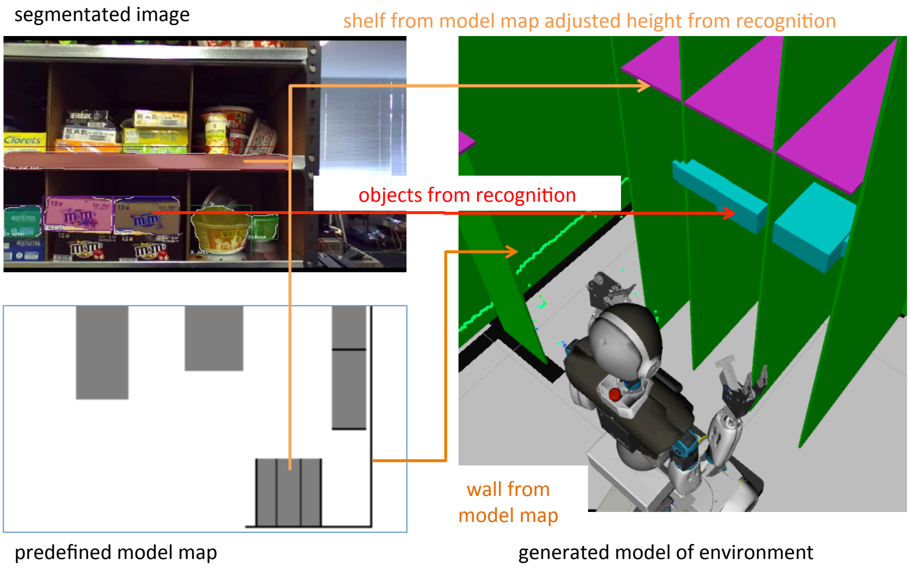

segmentated image
shelf from model map adjusted height from recognition
objects from recognition
wall from
model map
predefined model map
generated model of environment

Fig A.7: Example of modeling an environment from a combination of pre-defined maps and online model generation from vision.

Publications

First Author

International Journal

1. Kazuhiro Sasabuchi, Hiroaki Yaguchi, Kotaro Nagahama, Shintaro Hori, Hiroto Mizohana, Masayuki Inaba. The Seednoid Robot Platform: Designing a Multi-purpose Compact Robot from Continuous Evaluation and Lessons from Competitions. IEEE Robotics and Automation Letters, Vol. 3 No. 4 pp. 3983-3990, 2018.

International Conference

2. Kazuhiro Sasabuchi, Katsushi Ikeuchi, Masayuki Inaba. Agreeing to Interact: Understanding Interaction as Human-Robot Goal Conflicts. In *Companion of The 2018 IEEE/RSJ International Conference on Human-Robot Interaction (HRI2018)*, pp. 21-28, 2018.

3. Kazuhiro Sasabuchi, Yohei Kakiuchi, Kei Okada, Masayuki Inaba. Design and implementation of multi-dimensional flexible antena-like hair motivated by 'Aho-Hair' Japanese anime cartoons: Internal state expressions beyond design limitations. In *Proceedings of The 2015 IEEE International Conference on Robot and Human Interactive Communication (RO-MAN2015)*, pp. 223-228, 2015.

国内学術講演会

4. 笹渕　一宏, 中西　雄飛, 岡田 慧, 稲葉 雅幸. CG顔を実機で作るための複合線路を利用する曲面追従機構を備えた口表情デバイスの実現. 日本機械学会ロボティクス・メカトロニクス講演会'13 講演論文集, 2P1-Q12, 2013.

Awards

5. C-team. *Sevel-Eleven Japan Award*, World Robot Summit Future Convenience Store Challenge Trial Competition 2017, 2017.12.21.

6. C-team. *Most Challenging Award*, World Robot Summit Future Convenience Store Challenge Trial Competition 2017, 2017.12.21.

Co-Author

International Journal

1. Yuto Nakanishi, Shigeki Ohta, Takuma Shirai, Yuki Asano, Toyotaka Kozuki, Yuriko Kakehashi, Hironori Mizoguchi, Tomoko Kurotobi, Yotaro Motegi, Kazuhiro Sasabuchi, Kei Okada, Masayuki Inaba. Design approach of biologically-inspired musculoskeletal humanoids. International Journal of Advanced Robotic Systems, Vol. 10 No. 4 pp. 216, 2013.

International Conference

2. Yuki Furuta, Kazuhiro Sasabuchi, Yusuke Niitani, Kotaro Nagahama, Hiroaki Yaguchi, Kei Okada, Masayuki Inaba. Bring me manju from the drawer: Task Acquisition Framework under Incompleteness and Ambiguity using Interaction and Semantic Knowledge-enabled Perception. In *The 2017 IEEE/RSJ IROS Workshop Machine Learning Methods for High-Level Cognitive Capabilities in Robotics*, 2017.

3. Yaguchi Hiroaki, Kazuhiro Sasabuchi, Shintaro Hori, Kotaro Nagahama, Masayuki Inaba. A research on autonomous loading/unloading of consumer products using a dual-arm robot. In *The 2017 IEEE ICRA Workshop Warehouse Picking Automation Workshop*, 2017.

4. Hiroaki Yaguchi, Kazuhiro Sasabuchi, Wesley Patrick Chan, Kotaro Nagahama, Takayuki Saiki, Yasuto Shiigi, Masayuki Inaba. A design of 4-legged semi humanoid robot aero for disaster response task. In *Proceedings of The 2015 IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 61-66, 2015.

国内学術講演会

5. 東出 泰治, 笹渕 一宏, 矢野倉 伊織, 垣内 洋平, 岡田 慧, 稲葉 雅幸. 車輪移動双腕ロボットによる店舗環境における整列操作のための物体認識,. 第36回日本ロボット学会学術講演会講演論文集, 3J3-02, 2018.

6. 堀慎太郎, 笹渕 一宏, 長濱 小太郎, 矢口 裕明, 稲葉 雅幸. 視覚情報を用いた把持計画と置き場所探索に基づく品出しロボットの研究. 情報処理学会第79回全国大会講演論文集, pp. 161-162, 2017.

7. 長濱小太郎, 笹渕 一宏, 矢口 裕明, 稲葉 雅幸. 等身大人型ロボットプラットフォームを用いた農業作業支援への取り組み. 日本機械学会ロボティクス・メカトロニクス講演会’16講演論文集, 2P2-17a1, 2016.

8. 長濱小太郎, 笹渕 一宏, 矢口 裕明, 稲葉 雅幸. 等身大人型ロボットの農作業タスクプログラミングと収穫競技会を通した知能ロボットソフトウェア教育. 日本機械学会ロボティクス・メカトロニクス講演会’16講演論文集, 2P2-17a2, 2016.

9. 矢口 裕明, 笹渕 一宏, 長濱小太郎, 稲葉 雅幸. 生活支援研究プラットフォーム Aero の競技会を通じたシステム評価. 第 34 回日本ロボット学会学術講演会講演論文集, 3F1-05, 2016.

10. 矢口 裕明, 笹渕 一宏, 長濱小太郎, 稲葉 雅幸. 多種物品取り出しタスクのための EusLisp を用いた幾何学モデル構築と把持・動作計画手法. 第 34 回日本ロボット学会学術講演会講演論文集, 3E3-04, 2016.

11. 大坪　論史, 古田 悠貴 , 稲垣 祐人, 笹渕 一宏, 室岡 雅樹, 垣内 洋平, 岡田 慧, 稲葉 雅幸. 動力学シミュレータを用いたホームアシスタントロボットによる家事支援タスク遂行. 第 33 回日本ロボット学会学術講演会講演論文集, 1J3-06, 2015.

Bibliography

[1] Henny Admoni, Anca Dragan, Siddhartha S Srinivasa, and Brian Scassellati. Deliberate delays during robot-to-human handovers improve compliance with gaze communication. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, pages 49–56. ACM, 2014.

[2] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and vqa. *arXiv preprint arXiv:1707.07998*, 2017.

[3] Sean Andrist, Xiang Zhi Tan, Michael Gleicher, and Bilge Mutlu. Conversational gaze aversion for humanlike robots. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, pages 25–32. ACM, 2014.

[4] Salvatore M Anzalone, Sofiane Boucenna, Serena Ivaldi, and Mohamed Chetouani. Evaluating the engagement with social robots. *International Journal of Social Robotics*, 7(4):465–478, 2015.

[5] Thomas Arnold and Matthias Scheutz. Observing robot touch in context: How does touch and attitude affect perceptions of a robot's social qualities? In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pages 352–360. ACM, 2018.

[6] Christopher G Atkeson, Benzun P Wisely Babu, Nandan Banerjee, Dmitry Berenson, Christoper P Bove, Xiongyi Cui, Mathew DeDonato, Ruixiang Du, Siyuan Feng, Perry Franklin, et al. No falls, no resets: Reliable humanoid behavior in the darpa robotics challenge. In *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*, pages 623–630. IEEE, 2015.

[7] Andrea Bajcsy, Dylan P Losey, Marcia K O'Malley, and Anca D Dragan. Learning from physical human corrections, one feature at a time. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pages 141–149. ACM, 2018.

[8] Siddhartha Banerjee and Sonia Chernova. Temporal models for robot classification of human interruptibility. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, pages 1350–1359. International Foundation for Autonomous Agents and Multiagent Systems, 2017.

[9] Jimmy Baraglia, Maya Cakmak, Yukie Nagai, Rajesh Rao, and Minoru Asada. Initiative in robot assistance during collaborative task execution. In *Human-Robot Interaction (HRI), 2016 11th ACM/IEEE International Conference on*, pages 67–74. IEEE, 2016.

[10] Robert Bohlin and Lydia E Kavraki. Path planning using lazy prm. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 1, pages 521–528. IEEE, 2000.

[11] Jonathan Bohren and Steve Cousins. The smach high-level executive [ros news]. *IEEE Robotics & Automation Magazine*, 17(4):18–20, 2010.

[12] Jonathan Bohren, Radu Bogdan Rusu, E Gil Jones, Eitan Marder-Eppstein, Caroline Pantofaru, Melonee Wise, Lorenz Mösenlechner, Wim Meeussen, and Stefan Holzer. Towards autonomous robotic butlers: Lessons learned with the pr2. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 5568–5575. IEEE, 2011.

[13] Dan Bohus and Eric Horvitz. Models for multiparty engagement in open-world dialog. In *Proceedings of the SIGDIAL 2009 Conference: The 10th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 225–234. Association for Computational Linguistics, 2009.

[14] Dan Bohus, Chit W Saw, and Eric Horvitz. Directions robot: in-the-wild experiences and lessons learned. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 637–644. International Foundation for Autonomous Agents and Multiagent Systems, 2014.

[15] Ian M Bullock, Raymond R Ma, and Aaron M Dollar. A hand-centric classification of human and robot dexterous manipulation. *Haptics, IEEE Transactions on*, 6(2):129–144, 2013.

[16] Harry Bunt. The semantics of dialogue acts. In *Proceedings of the Ninth International Conference on Computational Semantics*, pages 1–13. Association for Computational Linguistics, 2011.

[17] Zhe C, Tomas S, Shih-En W, and Yaser S. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

[18] Joyce Y Chai, Lanbo She, Rui Fang, Spencer Ottarson, Cody Littley, Changsong Liu, and Kenneth Hanson. Collaborative effort towards common ground in situated human-robot dialogue. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, pages 33–40. ACM, 2014.

[19] Danqi Chen, Jason Bolton, and Christopher D Manning. A thorough examination of the cnn. *Daily Mail Reading Comprehension Task*, 2016.

[20] Min Chen, Stefanos Nikolaidis, Harold Soh, David Hsu, and Siddhartha Srinivasa. Planning with trust for human-robot collaboration. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pages 307–315. ACM, 2018.

[21] Hao-Tien Lewis Chiang, Aleksandra Faust, Marek Fiser, and Anthony Francis. Learning navigation behaviors end to end. *arXiv preprint arXiv:1809.10124*, 2018.

[22] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. Attention-based models for speech recognition. In *Advances in neural information processing systems*, pages 577–585, 2015.

[23] Steve Cousins. Building a service robotics business-challenges from the field. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pages 3–3. ACM, 2018.

[24] Robert T Craig. Communication theory as a field. *Communication theory*, 9(2):119–161, 1999.

[25] Kerstin Dautenhahn, Chrystopher L Nehaniv, Michael L Walters, Ben Robins, Hatice Kose-Bagci, N Assif Mirza, and Mike Blow. Kaspar–a minimally expressive humanoid robot for human–robot interaction research. *Applied Bionics and Biomechanics*, 6(3-4):369–397, 2009.

[26] Kerstin Dautenhahn, Michael Walters, Sarah Woods, Kheng Lee Koay, Chrystopher L Nehaniv, A Sisbot, Rachid Alami, and Thierry Siméon. How may i serve you?: a robot companion approaching a seated person in a helping context. In *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, pages 172–179. ACM, 2006.

[27] Alessandro De Luca and Raffaella Mattone. Sensorless robot collision detection and hybrid force/-motion control. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 999–1004. IEEE, 2005.

[28] Mehmet Dogar and Siddhartha Srinivasa. A framework for push-grasping in clutter. *Robotics: Science and systems VII*, 1, 2011.

[29] Anca Dragan and Siddhartha Srinivasa. Familiarization to robot motion. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, pages 366–373. ACM, 2014.

[30] Anca D Dragan. Robot planning with mathematical models of human state and action. *arXiv preprint arXiv:1705.04226*, 2017.

[31] Anca D Dragan, Kenton CT Lee, and Siddhartha S Srinivasa. Legibility and predictability of robot motion. In *Human-Robot Interaction (HRI), 2013 8th ACM/IEEE International Conference on*, pages 301–308. IEEE, 2013.

[32] Rakesh Dugad and UDAY B Desai. A tutorial on hidden markov models. *Signal Processing and Artifical Neural Networks Laboratory, Dept of Electrical Engineering, Indian Institute of Technology, Bombay Technical Report No.: SPANN-96.1*, 1996.

[33] Maha Elbayad, Laurent Besacier, and Jakob Verbeek. Pervasive attention: 2d convolutional neural networks for sequence-to-sequence prediction. *arXiv preprint arXiv:1808.03867*, 2018.

[34] Clemens Eppner, Sebastian Höfer, Rico Jonschkowski, Roberto Martın-Martın, Arne Sieverling, Vincent Wall, and Oliver Brock. Lessons from the amazon picking challenge: Four aspects of building robotic systems. *Proceedings of Robotics: Science and Systems, AnnArbor, Michigan*, 2016.

[35] Juan Fasola and Maja J Matarić. Interpreting instruction sequences in spatial language discourse with pragmatics towards natural human-robot interaction. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 2720–2727. IEEE, 2014.

[36] Julia Fink, Séverin Lemaignan, Pierre Dillenbourg, Philippe Rétornaz, Florian Vaussard, Alain Berthoud, Francesco Mondada, Florian Wille, and Karmen Franinović. Which robot behavior can motivate children to tidy up their toys?: Design and evaluation of ranger. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, pages 439–446. ACM, 2014.

[37] Dirk Gehrig, Peter Krauthausen, Lukas Rybok, Hildegard Kuehne, Uwe D Hanebeck, Tanja Schultz, and Rainer Stiefelhagen. Combined intention, activity, and motion recognition for a humanoid household robot. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 4819–4825. IEEE, 2011.

[38] Michael J Gielniak and Andrea L Thomaz. Enhancing interaction through exaggerated motion synthesis. In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, pages 375–382. ACM, 2012.

[39] Randy Gomez, Deborah Szapiro, Kerl Galindo, and Keisuke Nakamura. Haru: Hardware design of an experimental tabletop robot assistant. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pages 233–240. ACM, 2018.

[40] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pages 6645–6649. IEEE, 2013.

[41] Alex Graves and Jürgen Schmidhuber. Offline handwriting recognition with multidimensional recurrent neural networks. In *Advances in neural information processing systems*, pages 545–552, 2009.

[42] Jonathan Grizou, Inaki Iturrate, Luis Montesano, Pierre-Yves Oudeyer, and Manuel Lopes. Interactive learning from unlabeled instructions. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence (UAI)*, number EPFL-CONF-205138, 2014.

[43] Abhinav Gupta and Larry S Davis. Objects in action: An approach for combining action understanding and object perception. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[44] Marc Hanheide, Moritz Göbelbecker, Graham S Horn, Andrzej Pronobis, Kristoffer Sjöö, Alper Aydemir, Patric Jensfelt, Charles Gretton, Richard Dearden, Miroslav Janicek, et al. Robot task planning and explanation in open and uncertain worlds. *Artificial Intelligence*, 2015.

[45] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[46] Keliang He, Morteza Lahijanian, Lydia E Kavraki, and Moshe Y Vardi. Towards manipulation planning with temporal logic specifications. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 346–352. IEEE, 2015.

[47] Paul Hebert, Max Bajracharya, Jeremy Ma, Nicolas Hudson, Alper Aydemir, Jason Reid, Charles Bergh, James Borders, Matthew Frost, Michael Hagman, et al. Mobile manipulation and mobility as manipulation—design and algorithms of robosimian. *Journal of Field Robotics*, 32(2):255–274, 2015.

[48] Carlos Hernandez, Mukunda Bharatheesha, Wilson Ko, Hans Gaiser, Jethro Tan, Kanter van Deurzen, Maarten de Vries, Bas Van Mil, Jeff van Egmond, Ruben Burger, et al. Team delft ’s robot winner of the amazon picking challenge 2016. In *Robot World Cup*, pages 613–624. Springer, 2016.

[49] Carlos Hernandez, Mukunda Bharatheesha, Jeff van Egmond, Jihong Ju, and Martijn Wisse. Integrating different levels of automation: Lessons from winning the amazon robotics challenge 2016. *IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS*, pages 1–11, 2018.

[50] Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. The goldilocks principle: Reading children’s books with explicit memory representations. *arXiv preprint arXiv:1511.02301*, 2015.

[51] Sepp Hochreiter and Jürgen Schmidhuber. Lstm can solve hard long time lag problems. In *Advances in neural information processing systems*, pages 473–479, 1997.

[52] Guy Hoffman, Gurit E Birnbaum, Keinan Vanunu, Omri Sass, and Harry T Reis. Robot responsiveness to human disclosure affects social impression and appeal. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, pages 1–8. ACM, 2014.

[53] Chien-Ming Huang, Maya Cakmak, and Bilge Mutlu. Adaptive coordination strategies for human-robot handovers. In *Proceedings of Robotics: Science and Systems*, 2015.

[54] Justin Huang, Tessa Lau, and Maya Cakmak. Design and evaluation of a rapid programming system for service robots. In *The Eleventh ACM/IEEE International Conference on Human Robot Interaction*, pages 295–302. IEEE Press, 2016.

[55] Koji Inoue, Divesh Lala, Katsuya Takanashi, and Tatsuya Kawahara. Engagement recognition by a latent character model based on multimodal listener behaviors in spoken dialogue. *APSIPA Transactions on Signal and Information Processing*, 7, 2018.

[56] Yun Jiang, Marcus Lim, and Ashutosh Saxena. Learning object arrangements in 3d scenes using human context. *arXiv preprint arXiv:1206.6462*, 2012.

[57] Matthew Johnson, Brandon Shrewsbury, Sylvain Bertrand, Tingfan Wu, Daniel Duran, Marshall Floyd, Peter Abeles, Douglas Stephen, Nathan Mertins, Alex Lesman, et al. Team ihmc’s lessons learned from the darpa robotics challenge trials. *Journal of Field Robotics*, 32(2):192–208, 2015.

[58] Simon Jones and Ling Shao. Linear regression motion analysis for unsupervised temporal segmentation of human actions. In *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*, pages 816–822. IEEE, 2014.

[59] Mathis Jording, Arne Hartz, Gary Bente, Martin Schulte-Rüther, and Kai Vogeley. The “ social gaze space ”: A taxonomy for gaze-based communication in triadic interactions. *Frontiers in psychology*, 9:226, 2018.

[60] Fabrice Jumel, Jacques Saraydaryan, Raphael Leber, Laëtitia Matignon, Eric Lombardi, Christian Wolf, and Olivier Simonin. Context aware robot architecture, application to the robocup@ home challenge. In *RoboCup symposium*, 2018.

[61] Leslie Pack Kaelbling and Tomas Lozano-Perez. Integrated robot task and motion planning in belief space. 2012.

[62] Yohei Kakiuchi, Kunio Kojima, Eisoku Kuroiwa, Shintaro Noda, Masaki Murooka, Iori Kumagai, Ryohei Ueda, Fumihito Sugai, Shunichi Nozawa, Kei Okada, et al. Development of humanoid robot system for disaster response through team nedo-jsk's approach to darpa robotics challenge finals. In *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*, pages 805–810. IEEE, 2015.

[63] Alisa Kalegina, Grace Schroeder, Aidan Allchin, Keara Berlin, and Maya Cakmak. Characterizing the design space of rendered robot faces. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pages 96–104. ACM, 2018.

[64] Kenji Kaneko, Fumio Kanehiro, Shuuji Kajita, Kazuhiko Yokoyama, Kazuhiko Akachi, Toshikazu Kawasaki, Shigehiko Ota, and Takakatsu Isozumi. Design of prototype humanoid robotics platform for hrp. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 3, pages 2431–2436. IEEE, 2002.

[65] Yusuke Kato, Takayuki Kanda, and Hiroshi Ishiguro. May i help you?: Design of human-like polite approaching behavior. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, pages 35–42. ACM, 2015.

[66] Sung-Kyun Kim and Maxim Likhachev. Parts assembly planning under uncertainty with simulation-aided physical reasoning. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 4074–4081. IEEE, 2017.

[67] Takanori Komatsu and Yukari Abe. Comparing an on-screen agent with a robotic agent in non-face-to-face interactions. In *Intelligent Virtual Agents*, pages 498–504. Springer, 2008.

[68] Hema S Koppula and Ashutosh Saxena. Anticipating human activities using object affordances for reactive robotic response. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 38(1):14–29, 2016.

[69] Minae Kwon, Sandy H Huang, and Anca D Dragan. Expressing robot incapability. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pages 87–95. ACM, 2018.

[70] Sebastian Lang, Marcus Kleinehagenbrock, Sascha Hohenner, Jannik Fritsch, Gernot A Fink, and Gerhard Sagerer. Providing the basis for human-robot-interaction: A multi-modal attention system for a mobile robot. In *Proceedings of the 5th international conference on Multimodal interfaces*, pages 28–35. ACM, 2003.

[71] Hee Rin Lee and Selma Sabanović. Culturally variable preferences for robot design and use in south korea, turkey, and the united states. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, pages 17–24. ACM, 2014.

[72] Angus Leigh, Joelle Pineau, Nicolas Olmedo, and Hong Zhang. Person tracking and following with 2d laser scanners. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 726–733. IEEE, 2015.

[73] Séverin Lemaignan, Fernando Garcia, Alexis Jacq, and Pierre Dillenbourg. From real-time attention assessment to with-me-ness in human-robot interaction. In *The Eleventh ACM/IEEE International Conference on Human Robot Interaction*, pages 157–164. IEEE Press, 2016.

[74] Tsung-Yi Lin, Piotr Dollár, Ross B Girshick, Kaiming He, Bharath Hariharan, and Serge J Belongie. Feature pyramid networks for object detection. In *CVPR*, volume 1, page 4, 2017.

[75] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[76] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[77] Diana Löffler, Nina Schmidt, and Robert Tscharn. Multimodal expression of artificial emotion in social robots using color, motion and sound. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pages 334–343. ACM, 2018.

[78] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

[79] Karl F MacDorman and Hiroshi Ishiguro. The uncanny advantage of using androids in cognitive and social science research. *Interaction Studies*, 7(3):297–337, 2006.

[80] Jim Mainprice, Mamoun Gharbi, Thierry Siméon, and Rachid Alami. Sharing effort in planning human-robot handover tasks. In *RO-MAN, 2012 IEEE*, pages 764–770. IEEE, 2012.

[81] Richard S Marken and Warren Mansell. Perceptual control as a unifying concept in psychology. *Review of General Psychology*, 17(2):190, 2013.

[82] Daniel McFarlane. Comparison of four primary methods for coordinating the interruption of people in human-computer interaction. *Human-Computer Interaction*, 17(1):63–139, 2002.

[83] Pierrick Milhorat, Divesh Lala, Koji Inoue, Tianyu Zhao, Masanari Ishida, Katsuya Takanashi, Shizuka Nakamura, and Tatsuya Kawahara. A conversational dialogue manager for the humanoid robot erica. In *Advanced Social Interaction with Agents*, pages 119–131. Springer, 2019.

[84] Christophe Mollaret, Alhayat Ali Mekonnen, Frédéric Lerasle, Isabelle Ferrané, Julien Pinquier, B Boudet, and Pierre Rumeau. A multi-modal perception based assistive robotic system for the elderly. *Computer Vision and Image Understanding*, 149:78–97, 2016.

[85] AJung Moon, Daniel M Troniak, Brian Gleeson, Matthew KXJ Pan, Minhua Zeng, Benjamin A Blumer, Karon MacLean, and Elizabeth A Croft. Meet me where i'm gazing: how shared attention gaze affects human-robot handover timing. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, pages 334–341. ACM, 2014.

[86] 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.

[87] Tim Mueller-Sim, Merritt Jenkins, Justin Abel, and George Kantor. The robotanist: a ground-based agricultural robot for high-throughput crop phenotyping. In *IEEE International Conference on Robotics and Automation (ICRA), Singapore, Singapore*, 2017.

[88] Ryo Murakami, Luis Yoichi Morales Saiki, Satoru Satake, Takayuki Kanda, and Hiroshi Ishiguro. Destination unknown: walking side-by-side without knowing the goal. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, pages 471–478. ACM, 2014.

[89] John Napier. The evolution of the hand. *Scientific American*, 207(6):56–65, 1962.

[90] Bonnie A Nardi. Studying context: A comparison of activity theory, situated action models, and distributed cognition. *Context and consciousness: Activity theory and human-computer interaction*, 69102, 1996.

[91] Andrew Y Ng and Michael I Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in neural information processing systems*, pages 841–848, 2002.

[92] Leah Nicolich-Henkin, Carolyn Rose, and Alan W Black. Initiations and interruptions in a spoken dialog system. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 148–156, 2016.

[93] Elnaz Nouri and David Traum. Initiative taking in negotiation. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 186–193, 2014.

[94] Paul Oh, Kiwon Sohn, Giho Jang, Youngbum Jun, and Baek-Kyu Cho. Technical overview of team drc-hubo@ unlv's approach to the 2015 darpa robotics challenge finals. *Journal of Field Robotics*, 34(5):874–896, 2017.

[95] Kei Okada, Yohei Kakiuchi, Haseru Azuma, Hiroyuki Mikita, Kazuto Murase, and Masayuki Inaba. Task compiler: Transferring high-level task description to behavior state machine with failure recovery mechanism. In *ICRA Workshop on Combining Task and Motion Planning*, 2013.

[96] Hisashi Osumi. Basic knowledge of robot motion control. *Journal of the Japan Society for Precision Engineering*, 73(10):1123–1126, 2007.

[97] Oskar Palinko, Francesco Rea, Giulio Sandini, and Alessandra Sciutti. Eye gaze tracking for a humanoid robot. In *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*, pages 318–324. IEEE, 2015.

[98] Amit Pandey and Rodolphe Gelin. A mass-produced sociable humanoid robot: Pepper: The first machine of its kind. *IEEE Robotics & Automation Magazine*, (99):1–1, 2018.

[99] Caroline Pantofaru, Leila Takayama, Tully Foote, and Bianca Soto. Exploring the role of robots in home organization. In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, pages 327–334. ACM, 2012.

[100] Maximilian Panzner and Philipp Cimiano. Comparing hidden markov models and long short term memory neural networks for learning action representations. In *International Workshop on Machine Learning, Optimization and Big Data*, pages 94–105. Springer, 2016.

[101] Ronald Poppe. A survey on vision-based human action recognition. *Image and vision computing*, 28(6):976–990, 2010.

[102] Tom Porter and Galyn Susman. On site: creating lifelike characters in pixar movies. *Communications of the ACM*, 43(1):25, 2000.

[103] Emmanuel Pot, Jérôme Monceaux, Rodolphe Gelin, and Bruno Maisonnier. Choregraphe: a graphical tool for humanoid robot programming. In *Robot and Human Interactive Communication, 2009. RO-MAN 2009. The 18th IEEE International Symposium on*, pages 46–51. IEEE, 2009.

[104] William T Powers. Quantitative analysis of purposive systems: Some spadework at the foundations of scientific psychology. *Psychological Review*, 85(5):417, 1978.

[105] Ana Huamán Quispe, Heni Ben Amor, and Henrik I Christensen. A taxonomy of benchmark tasks for robot manipulation. In *Robotics Research*, pages 405–421. Springer, 2018.

[106] R Ranjan, V Patel, and R Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *arXiv preprint arXiv:1603.01249*, 2016.

[107] Photchara Ratsamee, Yasushi Mae, Kazuto Kamiyama, Mitsuhiro Horade, Masaru Kojima, and Tatsuo Arai. Social interactive robot navigation based on human intention analysis from face orientation and human path prediction. *ROBOMECH Journal*, 2(1):1–18, 2015.

[108] Harish Chaandar Ravichandar and Ashwin Dani. Human intention inference and motion modeling using approximate em with online learning. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 1819–1824. IEEE, 2015.

[109] Robin Read and Tony Belpaeme. Situational context directs how people affectively interpret robotic non-linguistic utterances. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, pages 41–48. ACM, 2014.

[110] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[111] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[112] Laurel D Riek. Wizard of oz studies in hri: a systematic review and new reporting guidelines. *Journal of Human-Robot Interaction*, 1(1):119–136, 2012.

[113] Tim Rocktäschel, Johannes Welbl, and Sebastian Riedel. Frustratingly short attention spans in neural language modeling, 2017.

[114] Bernardino Romera-Paredes and Philip Hilaire Sean Torr. Recurrent instance segmentation. In *European Conference on Computer Vision*, pages 312–329. Springer, 2016.

[115] Christoph Rösmann, Wendelin Feiten, Thomas Wösch, Frank Hoffmann, and Torsten Bertram. Trajectory modification considering dynamic constraints of autonomous robots. In *Robotics; Proceedings of ROBOTIK 2012; 7th German Conference on*, pages 1–6. VDE, 2012.

[116] Selma Sabanovic, Casey C Bennett, Wan-Ling Chang, and Lesa Huber. Paro robot affects diverse interaction modalities in group sensory therapy for older adults with dementia. In *Rehabilitation Robotics (ICORR), 2013 IEEE International Conference on*, pages 1–6. IEEE, 2013.

[117] Dorsa Sadigh, Nick Landolfi, Shankar S Sastry, Sanjit A Seshia, and Anca D Dragan. Planning for cars that coordinate with people: leveraging effects on human actions for planning and active information gathering over human internal state. *Autonomous Robots*, 42(7):1405–1426, 2018.

[118] Yoshiaki Sakagami, Ryujin Watanabe, Chiaki Aoyama, Shinichi Matsunaga, Nobuo Higaki, and Kikuo Fujimura. The intelligent asimo: System overview and integration. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 3, pages 2478–2483. IEEE, 2002.

[119] Kazuhiro Sasabuchi, Katsushi Ikeuchi, and Masayuki Inaba. Agreeing to interact: Understanding interaction as human-robot goal conflicts. In *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pages 21–28. ACM, 2018.

[120] Kazuhiro Sasabuchi, Yohei Kakiuchi, Kei Okada, and Masayuki Inaba. Design and implementation of multi-dimensional flexible antena-like hair motivated by 'aho-hair' japanese anime cartoons: Internal state expressions beyond design limitations. In *Robot and Human Interactive Communication (RO-MAN), 2015 24th IEEE International Symposium on*, pages 223–228. IEEE, 2015.

[121] Kazuhiro Sasabuchi, Hiroaki Yaguchi, Kotaro Nagahama, Shintaro Hori, Hiroto Mizohana, and Masayuki Inaba. The seednoid robot platform: Designing a multi-purpose compact robot from continuous evaluation and lessons from competitions. *IEEE Robotics and Automation Letters*, 3(4):3983–3990, 2018.

[122] Satoru Satake, Takefumi Kanda, Dylan F Glas, Masayoshi Imai, Hiroshi Ishiguro, and Norihiro Hagita. A robot that approaches pedestrians. *Robotics, IEEE Transactions on*, 29(2):508–524, 2013.

[123] Max Schwarz, Tobias Rodehutskors, David Droeschel, Marius Beul, Michael Schreiber, Nikita Araslanov, Ivan Ivanov, Christian Lenz, Jan Razlaw, Sebastian Schüller, et al. Nimbro rescue: Solving disaster-response tasks with the mobile manipulation robot momaro. *Journal of Field Robotics*, 34(2):400–425, 2017.

[124] Francesco Setti, Chris Russell, Chiara Bassetti, and Marco Cristani. F-formation detection: Individuating free-standing conversational groups in images. *PloS one*, 10(5), 2015.

[125] Chao Shi, Michihiro Shimada, Takayuki Kanda, Hiroshi Ishiguro, and Norihiro Hagita. Spatial formation model for initiating conversation. In *Robotics: science and systems*, volume 11, 2011.

[126] Chao Shi, Masahiro Shiomi, Christian Smith, Takayuki Kanda, and Hiroshi Ishiguro. A model of distributional handing interaction for a mobile robot. In *Robotics: Science and Systems*, 2013.

[127] Kazuhiko Shinozawa, Futoshi Naya, Junji Yamato, and Kiyoshi Kogure. Differences in effect of robot and screen agent recommendations on human decision-making. *International Journal of Human-Computer Studies*, 62(2):267–279, 2005.

[128] Yoshiaki Shirai and Hirochika Inoue. Guiding a robot by visual feedback in assembling tasks. *Pattern recognition*, 5(2):99–106, 1973.

[129] Candace L Sidner, Christopher Lee, Cory D Kidd, Neal Lesh, and Charles Rich. Explorations in engagement for humans and robots. *Artificial Intelligence*, 166(1-2):140–164, 2005.

[130] Giresh K Singh and Jonathan Claassens. An analytical solution for the inverse kinematics of a redundant 7dof manipulator with link offsets. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 2976–2982. IEEE, 2010.

[131] Emrah Akin Sisbot, Luis Felipe Marin-Urias, Rachid Alami, and Thierry Simeon. A human aware mobile robot motion planner. *Robotics, IEEE Transactions on*, 23(5):874–883, 2007.

[132] Gabriel Skantze. Towards a general, continuous model of turn-taking in spoken dialogue using lstm recurrent neural networks. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 220–230, 2017.

[133] Dong Song, Nikolaos Kyriazis, Iason Oikonomidis, Chavdar Papazov, Antonis Argyros, Darius Burschka, and Danica Kragic. Predicting human intention in visual observations of hand/object interactions. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 1608–1615. IEEE, 2013.

[134] Anthony Stentz, Herman Herman, Alonzo Kelly, Eric Meyhofer, G Clark Haynes, David Stager, Brian Zajac, J Andrew Bagnell, Jordan Brindza, Christopher Dellin, et al. Chimp, the cmu highly intelligent mobile platform. *Journal of Field Robotics*, 32(2):209–228, 2015.

[135] Kyle Wayne Strabala, Min Kyung Lee, Anca Diana Dragan, Jodi Lee Forlizzi, Siddhartha Srinivasa, Maya Cakmak, and Vincenzo Micelli. Towards seamless human-robot handovers. *Journal of Human-Robot Interaction*, 2(1):112–132, 2013.

[136] Jorg Stuckler, Dirk Holz, and Sven Behnke. Robocup@ home: Demonstrating everyday manipulation skills in robocup@ home. *IEEE Robotics & Automation Magazine*, 19(2):34–42, 2012.

[137] Ioan A Şucan and Lydia E Kavraki. Kinodynamic motion planning by interior-exterior cell exploration. In *Algorithmic Foundation of Robotics VIII*, pages 449–464. Springer, 2009.

[138] Dag Sverre Syrdal, Kerstin Dautenhahn, Kheng Lee Koay, and Wan Ching Ho. Integrating constrained experiments in long-term human–robot interaction using task-and scenario-based prototyping. *The Information Society*, 31(3):265–283, 2015.

[139] Satoshi Tadokoro, Makoto Aikawa, and Kazuhiro Ichimichi. An analysis of human upper extremity motions. *The Japanese Journal of Ergonomics*, 26(1):41–47, 1990.

[140] Leila Takayama, Doug Dooley, and Wendy Ju. Expressing thought: improving robot readability with animation principles. In *Proceedings of the 6th international conference on Human-robot interaction*, pages 69–76. ACM, 2011.

[141] Leila Takayama and Caroline Pantofaru. Influences on proxemic behaviors in human-robot interaction. In *Intelligent robots and systems, 2009. IROS 2009. IEEE/RSJ international conference on*, pages 5495–5502. IEEE, 2009.

[142] Fumihide Tanaka, Toshimitsu Takahashi, Shizuko Matsuzoe, Nao Tazawa, and Masahiko Morita. Telepresence robot helps children in communicating with teachers who speak a different language. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, pages 399–406. ACM, 2014.

[143] Frank Thomas, Ollie Johnston, and Frank. Thomas. *The illusion of life: Disney animation*. Hyperion New York, 1995.

[144] Marc Toussaint, K Allen, K Smith, and J Tenenbaum. Differentiable physics and stable modes for tool-use and manipulation planning. RSS, 2018.

[145] Stefan Ulbrich, Daniel Kappler, Tamim Asfour, Nikolaus Vahrenkamp, Alexander Bierbaum, Markus Przybylski, and Rüdiger Dillmann. The opengrasp benchmarking suite: An environment for the comparative analysis of grasping and dexterous manipulation. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 1761–1767. IEEE, 2011.

[146] Deutsche Gezetzliche Unfallversicherung. Bg/bgia risk assessment recommendations according to machinery directive, design of workplaces with collaborative robots. Technical report, Report, 2009.

[147] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[148] Erli Wang, Hanna Kurniawati, and Dirk P Kroese. An on-line planner for pomdps with large discrete action space: A quantile-based approach. 2018.

[149] Thomas Wisspeintner, Tijn Van Der Zant, Luca Iocchi, and Stefan Schiffer. Robocup@ home: Scientific competition and benchmarking for domestic service robots. *Interaction Studies*, 10(3):392–426, 2009.

[150] Markus Wulfmeier, Dushyant Rao, Dominic Zeng Wang, Peter Ondruska, and Ingmar Posner. Large-scale cost function learning for path planning using deep inverse reinforcement learning. *The International Journal of Robotics Research*, 36(10):1073–1087, 2017.

[151] Keenan A Wyrobek, Eric H Berger, HF Machiel Van der Loos, and J Kenneth Salisbury. Towards a personal robotics development platform: Rationale and design of an intrinsically safe personal robot. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 2165–2170. IEEE, 2008.

[152] Lu Xia, Ilaria Gori, Jake K Aggarwal, and Michael S Ryoo. Robot-centric activity recognition from first-person rgb-d videos. In *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*, pages 357–364. IEEE, 2015.

[153] Qianli Xu, Liyuan Li, and Gang Wang. Designing engagement-aware agents for multiparty conversations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2233–2242. ACM, 2013.

[154] Keiichi Yamazaki, Michie Kawashima, Yoshinori Kuno, Naonori Akiya, Matthew Burdelski, Akiko Yamazaki, and Hideaki Kuzuoka. Prior-to-request and request behaviors within elderly day care: Implications for developing service robots for use in multiparty settings. In *ECSCW 2007*, pages 61–78. Springer, 2007.

[155] Shuo Yang, Ping Luo, Chen-Change Loy, and Xiaoou Tang. Wider face: A face detection benchmark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5525–5533, 2016.

[156] Bangpeng Yao and Li Fei-Fei. Recognizing human-object interactions in still images by modeling the mutual context of objects and human poses. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(9):1691–1703, 2012.

[157] Jason Yosinki, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014.

[158] Matt Zucker, Sungmoon Joo, Michael X Grey, Christopher Rasmussen, Eric Huang, Michael Stilman, and Aaron Bobick. A general-purpose system for teleoperation of the drc-hubo humanoid robot. *Journal of Field Robotics*, 32(3):336–351, 2015.

1p～ 198p fin

博士論文

平成30年12月7日提出

東京大学大学院 情報理工学系研究科
知能機械情報学専攻 博士課程
笹渕 一宏