TSEST Transaction Series on Engineering Sciences and Technologies

Supervisory Control of Multiple Social Robots for Conversation and Navigation

Kuanhao Zheng [1, ✉], Dylan F. Glas [2], Takayuki Kanda [3], Hiroshi Ishiguro [4], and Norihiro Hagita [5]

***Abstract*: This paper presents the modeling, implementation and validation of a human-robot system enabling a single operator to supervise multiple robots for social interactions with conversation and navigation. We developed a risk model for navigation, which can automatically tell when operation is necessary to enable safe navigation of unattended robots. We propose a utility model for evaluating the performance of interactions based on timings, and developed strategies and a planning algorithm to coordinate the tasks of the operator and robots for improving performance of the human-robot team. Finally, we conducted a field experiment in a shopping area using four robots for route-guiding services incorporating conversation and navigation to validate the effectiveness of our system.**

Keywords: Human-robot interaction, modeling, social robots

1. INTRODUCTION

Research and development of robots for social interactions have been making fast progress during recent years. Social robots have been successfully deployed in museums [1, 2], shopping malls [3, 4], and other public areas [5, 6], suggesting great potential for using social robots to communicate with people and provide useful services in the near future.

However, some major problems exist in realizing fully autonomous social robots for real-world applications. For conversation, speech recognition is still not accurate enough to support applications in noisy daily environments. It was shown in one study that a speech recognition system performing with 92.5% accuracy in 75dBA noise [7] achieved only 21.3% accuracy in a real-world environment [8].

Social interactions may often involve navigational aspects. For example, one recent study used robots in a shopping area to talk with customers as well as providing route guidance by walking together with them [9]. For autonomous navigation, although technologies such as SLAM [10] have been improving, some key problems have to be solved before we can use fully autonomous robots in real world, such as how to prevent localization errors due to dynamic features in the environment, and how to ensure safety during navigation.

Supervisory control is an effective way of compensating for the limitations of autonomous robots, wherein a human operator can control the robots during failures of automation. If a large proportion of robot tasks can be conducted autonomously, it is possible for a single operator to supervise multiple robots. Using such an approach, we can both enable robots to perform tasks which are not yet possible with full autonomy, and also leverage the efficiency of the human by enabling the operation of multiple robots at the same time.

But several challenges exist in applying supervisory control to multiple semi-autonomous robots. First, we need a reliable way to automatically determine when the robots need assistance; otherwise the operator may need to spend a lot of time in monitoring and deciding which robot to help. Second, we need an efficient way to schedule the tasks of the operator when multiple robots require operation, which is the key to improving the performance of a human-robot team.

We have developed models and algorithms to solve the problems specifically for conversation [11] and navigation [12] in previous studies. In this study, we have combined the previous models into a unified model, and implemented a working system that enables safe and efficient interactions including conversation and navigation.

In the remaining sections, we will present some related work in Section 2, and we will describe the overall model addressed in this study in Section 3. We will present the detailed algorithms and implementations through Sections 4-6, and present a field experiment in Section 7.

This work was supported by JST, CREST.

[1] *Kuanhao Zheng* (✉), *PhD Candidate, Intelligent Robotics Laboratory, Graduate School of Engineering Science, Osaka University, Japan; Intelligent Robotics and Communication Laboratories, ATR (Advanced Telecommunications Research Institute International), Japan.* (Email: zheng.kuanhao@is.sys.es.osaka-u.ac.jp).

[2,3,5] *Dylan F. Glas, Takayuki Kanda, and Norihiro Hagita, Intelligent Robotics and Communication Laboratories, ATR, Japan.* (Emails: dylan@atr.jp; kanda@atr.jp; hagita@atr.jp).

[4] *Hiroshi Ishiguro, Professor, Intelligent Robotics Laboratory, Graduate School of Engineering Science, Osaka University, Japan; Hiroshi Ishiguro Laboratory, ATR, Japan.* (Email: ishiguro@sys.es.osaka-u.ac.jp).

RECEIVED: 25, DEC., 2013; **REVISED:** 4, MAR., 2014; **ACCEPTED:** 27, APR., 2014; **PUBLISHED:** 6, JUN., 2014. ISSN: 2345-234X

010310007600922014000252

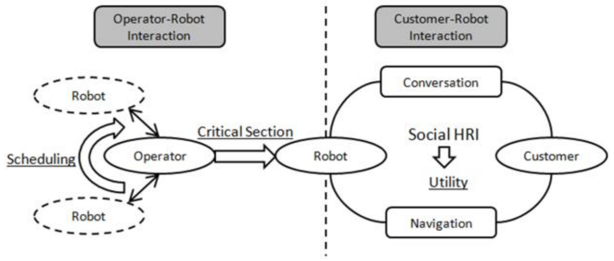


Fig.1. Interaction model overview

2. RELATED WORK

A. *Multi-Robot Teleoperation and Supervisory Control*

The concept of "fan-out" was first introduced by D. Olsen et al. [13, 14]. It describes the maximum number of robots an operator can effectively control, which can be used to evaluate the performance of a human-robot team [15]. Metrics such as interaction efficiency [16] and neglect tolerance [17] were introduced as important predictive tools for estimating the fan-out of a given system.

The strategy used in operator attention allocation greatly affects the fan-out of a human-multi-robot system [18], and various strategies for integrating human-in-the-loop have been studied [19, 20, 21]. One effective strategy is to automatically schedule the operator's tasks based on detection of each robot's task status, and then scheduling algorithms such as dSSPT [22] or MILP [23] can be applied to efficiently allocate the operator's tasks in real time. Those studies focused on tasks for coordinating multiple UAVs or rescue robots, but not for social robots.

For operator allocation for multiple social robots, an important issue is to determine when operation is needed during the task of social interactions. The concept of "critical section" [24] was proposed to define the time when operation is needed by identifying when failures in automation could occur. Modeling the structure of conversational interactions enables us to predict critical sections [11], but only conversational interaction was addressed in that study.

B. *Autonomous Localization and Navigation*

Improving autonomy in localization and navigation has been an important topic in mobile robotics. Typical algorithms for localization include Monte Carlo method [25] and grid-based Markov localization using Kalman-filter [26]. They calculate the probability distribution of the robot's position based on sensor inputs and a given map of the environment. When a map is not available, SLAM algorithms [10] can be used to simultaneously update the map and localize the robot.

One of the common features among those algorithms is that they all calculate the most-likely position of the robot based on some probability distribution. Most-likely position is useful for autonomous navigation, but may not be enough to guarantee with high confidence that robots do not enter safety-critical areas during navigation. To solve this problem, we developed an algorithm to calculate the boundary of all possible positions considering worst-case position estimates [12]. However, that study only presented a proof of concept and did not contain a formal evaluation of the algorithm's effectiveness through a comparison experiment.

3. PROBLEM DESCRIPTION

In this section, we give an overview of the interaction model for the problem scope. Two key problems will be discussed: the first is how to identify when operation is needed for supervising semi-autonomous robots, and the second is how to schedule the tasks of the operator and robots to improve team performance.

A. *Model Overview*

Fig. 1 shows an overview of the interaction model in this study. There are three types of roles in our model, namely the operator, robots, and customers. We continue to use the term "customers" to refer to the people who interact with robots as service receivers as in previous studies [11, 27]. This role differs from the role of "operator", who supervises the robots in providing services to the customers.

The overall model can be divided into "customer-robot interaction" and "operator-robot interaction". In the customer-robot interaction, the robot performs social interaction with the customer with either conversation or navigation, and some utility is produced from both types of interactions, representing the effectiveness in providing a service.

The operator's job is to assist the robots when they are unable to perform automatically, and we use the term *Critical Section* to refer to these time periods. In this study, the operator's tasks consisted of speech recognition in conversational interactions, and localization correction for navigational interactions. Assisting speech recognition is necessary because current technology is unable to reliably recognize human speech in real-world environments, and correction of localization is necessary because errors in automatic localization may cause risky situations during navigation.

Multiple robots can be supervised by switching the operator to the robots which are in critical sections. But here, primary research problems arise, which are how to correctly identify the critical section, and how to efficiently schedule the task of the operator when multiple robots are in critical sections. We will discuss these problems in the next sub-sections.

B. *Identifying Critical Sections*

We have been able to identify critical sections for conversation [11] by predicting when speech recognition is necessary in a conversation. Refer to [11] for more detailed explanations about identifying critical sections for conversation.

For navigation, there can be many important tasks which require operator assistance depending on the robot's objective. In this study, the operator's task is to ensure

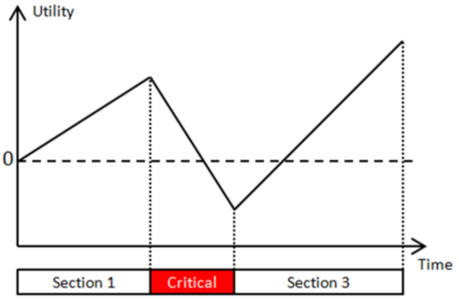


Fig. 2. The change of utility during different sections of an interaction

safety during navigation, which is closely associated with localization performance. Safety, including safety of people, safety of the robot, and safety of the environment, is one of the most important requirements for robots which interact with people, and it will not be feasible to deploy robot services in real-world social spaces until a high level of safety can be achieved.

In order to enable safe navigation, we must be able to distinguish the situations when there is any potential risk from situations when the robot can navigate autonomously. This distinction is used to define the critical sections in navigation. In this definition, it is most important to avoid the dangerous situation in which the robot navigates autonomously under the assumption that it is safe while actually it is not. We will present an algorithm to automatically identify critical sections considering the potential risk during navigation in Section 4.

C. Utility Model and Task Scheduling

Identifying critical sections enables us to determine when a robot is in need of an operator's assistance. But the next problem is how to schedule tasks when multiple robots are in critical sections, since the order for controlling the robots may greatly affect the performance of the robot team. To solve this problem, we need to first evaluate the performance from interactions in a quantitative way, which is the purpose of our utility model.

The basic idea of the utility model is that some constant rate of gain or loss in customer satisfaction over time is produced from each section of an interaction. Fig. 2 illustrates the change of utility over time in an interaction containing three different sections. In the first section, a constant increase of utility is produced as the robot's behavior satisfies the customer. Examples of such situations may include cases in which the robot is talking to the customer to present some useful information, or moving with the customer towards a goal.

Drops of utility may occur during the critical section, when the customer is waiting for a response from the robot. The total utility may become negative, meaning that an overall negative impression has been produced from the irresponsiveness of the robot. In the last section, the utility may increase again when the robot continues providing its service after operator assistance, and the final measure of utility from the interaction is the value at the last moment in the graph.

We can thus define the utility using a linear model by (1), which is the sum of utilities from each of its sections. For each section, the factor α_i defines the utility factor per unit amount of time, and T_i is the duration of the section. We will measure the utility factors for conversation and navigation in the user study in section V.

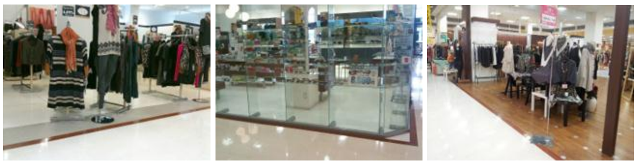

(a) (b) (c)

Fig. 3. Shops with (a) many clothing racks, (b) glass walls, and (c) invisible boundaries

$$U = \sum_{i \in Sections} \alpha_i T_i \qquad (1)$$

Given that we can calculate utilities as the outputs from interactions, the problem of multi-robot supervision can be formulated as a problem of maximizing utility. Through Sections 4-6, we will develop an algorithm to efficiently schedule the tasks of the operator and robots in order to maximize the sum of utilities produced from the human-robot team.

4. NAVIGATION RISK MODEL

In this section, we discuss how to identify the critical section for navigation by introducing a risk model. We will first describe the cause of risk in navigation, and then introduce an algorithm for estimating navigation risk.

A. Forbidden Areas

As we have introduced, the purpose of identifying critical sections for navigation is to prevent risks to safety. One of the sources of navigation risk is undetectable obstacles. While techniques for avoiding static and dynamic obstacles have been available for robots for many years, public spaces like shopping malls often include many obstacles that cannot be detected with on-board sensors.

Another source of risk is invisible boundaries defined by business or social convention, such as the boundary of an open shop or a market space. These boundaries must also be respected by robots, as a robot barging into a shop, café, or rest room could be upsetting to people in those spaces.

Together, we define the regions where the robots should not enter as *Forbidden Areas*, with some examples shown in Fig. 3. The first example is a shop with many clothing racks, which may result in incorrect distances to these features if the range finder of the robot is attached near the ground. The second example is a shop with glass walls, which cannot be detected with typical laser range finders. The third is a shop with open boundaries. Humans can easily identify the boundary of an the shop, but it is difficult for robots to recognize it due to the difficulty in incorporating humans' common sense into robots.

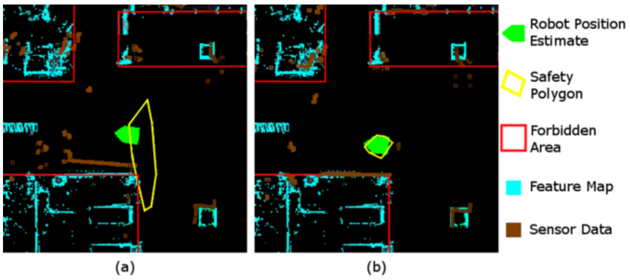


Fig. 4. Robot position estimate and safety polygon (a) before and (b) after localization by the operator

The detection of forbidden areas depends on the type and configuration of robot sensors. Different robots may be able to detect different kinds of forbidden areas due to their sensor configurations. To provide safety regardless of differences in sensor configuration, our approach is to manually define a reference map of forbidden areas, which guarantees that robots with any sensor configuration can navigate safely.

B. *Environment Map*

In order to prevent robots from entering forbidden areas, a map of the environment is required for localization of the robots as well as providing references to the forbidden areas.

In our study, the map of the environment was generated using a mobile robot equipped with a laser range finder and an offline SLAM using 3D Toolkit [28]. We drove the robot along a path covering the whole environment to log the laser and odometry data, and we generated a 2D scan map with offline SLAM using the logged data. In order to reduce inconsistencies of the map, we removed any movable objects such as benches or clothing racks from the generated map, leaving only the permanent features such as walls and pillars.

To define the forbidden areas within the generated map, we observed the real environment to identify the forbidden areas according to our definition. For the vertices defining each forbidden area, we manually measured their coordinates relative to observable features in the map whose coordinates were already known. We then padded the forbidden areas with an additional safety margin, which was necessary only for our comparison experiment, the details of which will be explained in Section 7.

Fig. 4 shows an excerpt from the feature map and forbidden areas generated for our experiment in a shopping mall.

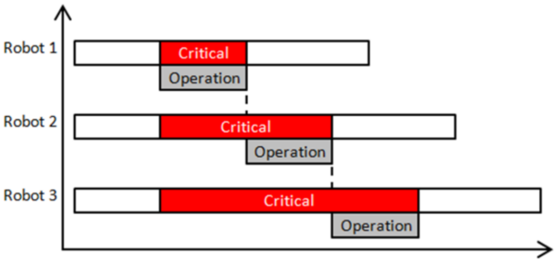


Fig. 5. Conflict of interactions when multiple robots are in the critical section

C. *Risk Estimation Algorithm*

We developed a risk estimation algorithm to predict the risk of the robots from entering forbidden areas, which is an adapted particle filter extended from [29]. Refer to Appendix A for the detailed implementation of the algorithm.

The key feature of the algorithm is to predict a *Safety Polygon*, which is the outer boundary of possible robot positions (represented by particles) considering worst-case localization errors. When the safety polygon intersects with a forbidden area, it means there is a potential risk that the robot has entered such area, and the robot automatically stops moving until the situation is handled.

We use a human operator to confirm the safety situation by re-localizing the robot when the safety polygon intersects with a forbidden area. Fig. 4 (a) shows the estimated robot position and safety polygon when a potential risk is detected. Fig. 4 (b) shows the updated safety polygon after the robot's location is corrected by the operator.

5. INTERACTION-MANAGEMENT STRATEGIES

We have presented a technique for identifying the critical section for navigation using a risk estimation algorithm, but the length of a critical section may increase when multiple robots are in need of operation at the same time and some are made to wait. Fig. 5 illustrates such a case, in which the critical sections for the second and third robots are extended while the operator is busy with other robots. The robots will be irresponsive while in the critical section, causing frustration and negative effects on customers [31]. Thus, controlling the length of critical sections is very important in terms of improving utility of interactions.

Fig. 5 shows that the length of a critical section includes two parts: 1) delay of operator assignment, and 2) the length of operation. We assume that the length of operation can be modeled as a static factor for a given human-machine system, and so we focus here on the issue of how to reduce delay of operation for multiple social robots.

Delay of operation is closely related to the starting time of critical sections, which is determined by different factors for different types of interactions. For conversation, it is determined by the time when the customer starts to ask a question (see Fig. 6). For navigation, it is determined by the time when potential risk is anticipated. We will find in this section that the starting time of critical sections can be managed if we carefully design the interactions in order to reduce delay of operations.

In the following sections, we present two categories of strategies to manage the starting time and length of critical sections. The first is to delay critical sections which an operator cannot immediately assist by using proactive behaviors, and the second is to reduce the length of critical sections in navigation by using conversation to fill the time. We will also present a human study to explore the effect of the strategies by measuring utility factors for conversation and navigation.

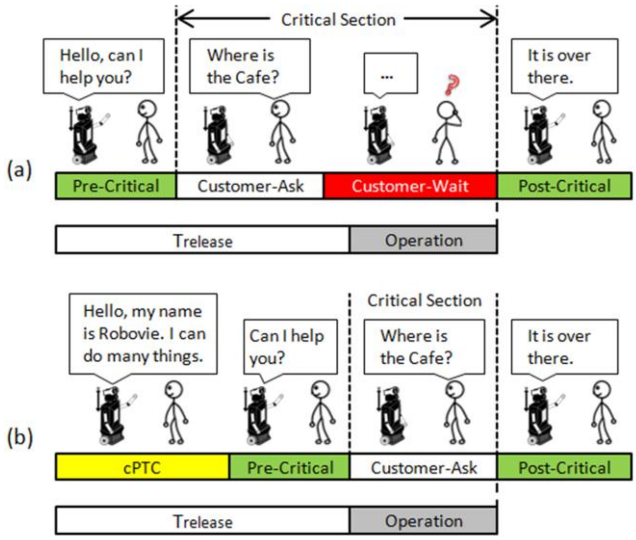


Fig. 6. Timing of interactions (a) without cPTC, and (b) with cPTC

A. Proactive Timing Control (PTC)

The first strategy we introduce is called **Proactive Timing Control** (**PTC**), which can be used to delay entrance to the critical sections. The implementations for conversation and navigation are different, and we introduce them separately.

1) PTC for Conversation (cPTC)

PTC for conversation was defined in previous studies [32, 33] based on the idea that letting customers wait before asking the question is better than letting them wait for the same amount of time after they have asked a question. For simplicity, we will refer to this strategy as **cPTC** in the remainder of the paper.

Fig. 6 illustrates example interactions with and without using the cPTC strategy. The critical section begins when the customer starts asking a question, which is the time when speech recognition by the operator will be required. We use *Pre-Critical* and *Post-Critical* sections to denote the time before and after the critical section, which are the parts when the robot can speak autonomously.

Suppose $T_{release}$ is the time before the operator can be allocated to the robot. Without cPTC, customer wait time happens when the operator cannot be allocated before the customer starts to ask, as in Fig. 6 (a). However, if we insert a cPTC section at the beginning of the conversation, as in Fig. 6 (b), the robot will keep talking using pre-defined contents until the operator is allocated, which enables customer waiting time to be avoided.

If we denote the utility factors for cPTC and customer waiting times as α_{cPTC} and α_{wait}, then we can determine if cPTC is useful in improving utility by comparing whether (2) holds, because the duration of cPTC will be equal to the length of time the customer would wait if cPTC were not used. Both α_{cPTC} and α_{wait} can be negative, but cPTC will improve interaction utility as long as the drop of utility during PTC is less than that during wait time.

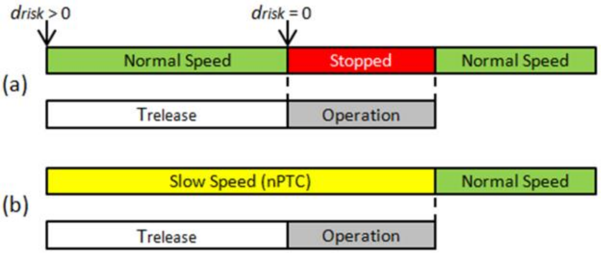


Fig. 7. Timing of interactions (a) without nPTC, and (b) with nPTC

$$\alpha_{cPTC} > \alpha_{wait} \tag{2}$$

2) PTC for Navigation (nPTC)

We can also use PTC strategy to delay the entrance to the critical section for navigation, which we will refer to as **nPTC**. In our target scenario, we assume that the customer follows the robot during navigation to some destination. The key idea for nPTC is that while moving the same distance, we can let the robot use a slower speed to increase the time it travels, which may result in delay of the critical section in the near future.

The time to entering the critical section in navigation is determined by the distance to risk (d_{risk}) and navigation speed (v), which can be expressed by (3). The equation indicates that given an estimated distance to risk, the time to the critical section is a flexible value which can be adjusted by speed control. The slower the speed is, the more time the robot can move before the operator is allocated.

$$T_{toCritical} = d_{risk}/v \tag{3}$$

Fig. 7 (a) illustrates the case when the robot moves at its normal speed and stops when d_{risk} becomes zero. Fig. 7 (b) shows the scenario when the robot moves with a slower speed in the same situation, so that the robot stays in motion until the operator is allocated, hence stopping of the robot is prevented. We define the time when the robot moves with some slower speed as nPTC.

The reducing of stopping time by nPTC comes at the cost of lowering the quality of the customer experience during slow movement. A previous study measured people's preferred speed when walking with robots [34], and we were able to measure the acceptable minimum speed from a number of subjects [12]. Using the result from [12], we chose a constant speed of 1.0m/s as normal movement speed, and 0.5m/s as slow speed, which were both rated as acceptable speeds in the human study.

We can compare the utilities with and without the use of slow speed to decide whether nPTC is effective in improving utility. Let α_{normal}, α_{slow}, and α_{stop} denote the utility factors for navigation speed of v_{normal}, v_{slow} and stopping, then the difference in utility resulting from nPTC can be calculated by (4) ~ (7).

$$U_{without\_nPTC} = \alpha_{normal}T_{normal} + \alpha_{stop}T_{stop} \tag{4}$$

$$U_{with\_nPTC} = \alpha_{slow}T_{slow} \tag{5}$$

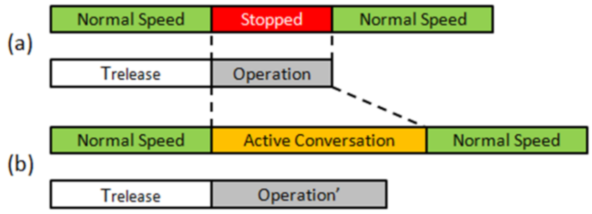


Fig. 8. Timing of interactions (a) without active conversation, and (b) with active conversation during navigation stop

$$T_{normal} = \frac{d_{risk}}{v_{normal}}; T_{slow} = \frac{d_{risk}}{v_{slow}};$$
$$T_{stop} = T_{slow} - T_{normal} \quad (6)$$

$$\Delta U = U_{with\_nPTC} - U_{without\_nPTC}$$
$$= \alpha_{slow}T_{slow} - \alpha_{normal}T_{normal} - \alpha_{stop}T_{stop} \quad (7)$$
$$= \frac{d_{risk}}{v_{slow}}\left(\alpha_{slow} - \alpha_{stop}\right) - \frac{d_{risk}}{v_{normal}}\left(\alpha_{normal} - \alpha_{stop}\right)$$

Solving for $\Delta \mathrm{U} > 0$ results in (8). If the inequality in (8) is true, then nPTC should be used. Otherwise, it is better not to use it. The left- and right-hand side of the equation can be seen as compensated utility factors for using and without using nPTC. The equation indicates that in order for nPTC to be useful, the gained utility from reduced stopping time should be larger than the loss of utility from replacing normal speed with slow speed.

$$\frac{\alpha_{slow} - \alpha_{stop}}{v_{slow}} > \frac{\alpha_{normal} - \alpha_{stop}}{v_{normal}} \quad (8)$$

B. Active Conversation

We have explained how PTC strategies can reduce the length of critical sections by delaying their start times for conversation and navigation. But stopping of locomotion still happens if the operator cannot be allocated before the robot will reach a forbidden area, even while moving at the slow speed. In this sub-section, we introduce a strategy called **Active Conversation** to reduce the length of critical sections due to stopping of locomotion.

This strategy is inspired by the fact that sometimes the robot has to stop its locomotion in order to have a conversation during navigational tasks. For example, when the robot is guiding a customer to somewhere in a shopping mall, the customer may want to stop and ask something about another shop near the route. The stopping of the robot in this case is necessary for having a conversation, thus will not cause a negative effect to the interaction. In a similar way, we can let the robot actively start a conversation whenever it has to stop navigation, so that the negative effect of stopping can be compensated by having a conversation.

Fig. 8 compares the timing of interactions with and without using active conversation. As in Fig. 8 (a), when stopping happens in navigation, the stopping time is equal to the operation time for localizing the robot. When using active conversation as in Fig. 8 (b), the stopping is replaced with a conversation. The operation time may increase because the operator has to both fix the navigation problem and assist the conversation.

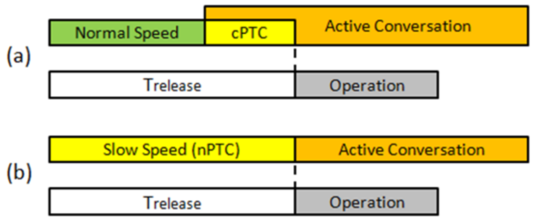


Fig. 9. Timing of interactions for (a) cPTC and (b) nPTC when active conversation is used

In order to find out wheter using active conversation is effective, we need to extend the utility model for interactions including both conversation and navigation. We use ω_{conv} and ω_{nav} to denote the weights for conversation and navigation in contributing to the overall utility. Then, we extend the utility model by (9), wherein *Conv* and *Nav* stand for the interaction sections belonging to conversation and navigation.

$$U = \omega_{conv} \cdot \sum_{i \in Conv} \alpha_i T_i + \omega_{nav} \cdot \sum_{j \in Nav} \alpha_j T_j \quad (9)$$

Using this extended utility model, we can evaluate the effectiveness of active conversation by calculating utilities from interaction timings. If we use T_{op}^{nav} to denote operation time for navigation, then the difference in utility between using and not using active conversation can be calculated by (10).

$$\Delta U = \omega_{conv} \sum_{i \in Conv} \alpha_i T_i - \omega_{nav} \alpha_{stop} T_{op}^{nav} \quad (10)$$

Active conversation can be used with a combination of PTC strategies. There are two possible ways to combine PTC with active conversation: one is to use cPTC as in Fig. 9 (a), and the other is to use nPTC as in Fig. 9 (b). In the first case, the robot keeps moving with normal speed until it has to stop, and cPTC is performed during the active conversation. In the second case, slow speed is used before active conversation, so that nPTC is part of navigation.

To compare which strategy results in higher utility, we can perform a similar calculation procedure as we did for (4) ~ (7), wherein we replace stopping time by the time for cPTC, and we add the weight factors for conversation and navigation, which results in (11). If (11) is true, it means nPTC should be used before active conversation in order to achieve higher utility. Otherwise, cPTC should be used as a part of active conversation.

$$\frac{\omega_{nav}\alpha_{slow} - \omega_{conv}\alpha_{cPTC}}{v_{slow}} > \frac{\omega_{nav}\alpha_{normal} - \omega_{conv}\alpha_{cPTC}}{v_{normal}} \quad (11)$$

C. Data Collection

We conducted a data collection from human participants to obtain empirical values for each of the utility factors. From the measured data, we verified whether the equations (2), (8), and (11) would be satisfied, which determines whether the interaction management strategies introduced in this section are effective in improving utility.

Fourteen people (9 male, 5 female, average age 20.8 with standard deviation of 2.2) participated in the data collection, and most of them were undergraduate students.

1) *Procedure*

The participants performed as customers in interactions with a humanoid robot (see Fig. 13 (a)). At the beginning, an experimenter gave instructions to the participants and expressed that the goal of the interactions was for the robot to create positive value to the shopping mall by satisfying the customers. After each interaction, the experimenter collected their evaluations by asking participants to rate the quality of interaction using an integer between [-5 ... 5], wherein the minimum and maximum values represent lowest and highest utilities. A zero score is the borderline, and a score below zero means they would prefer not to receive the service.

To measure each of the utility factors, we divided the data collection into three parts with different forms of interactions, defined as follows:

- **Part 1**

The first part was used for measuring the conversation component of the utility model. The service provided by the robot was to tell the customer about the route to the shop where the customer wants to go, but without navigating. We prepared directions to four different shops in the shopping mall and let the participants ask randomly among them in each interaction. We used eight interactions in total for this part.

To measure α_{pre} and α_{post}, we used two interactions. One interaction only included a pre-critical section, wherein the robot offered to provide guidance, but the customer said “No, thanks.” The other interaction included pre- and post-critical sections with zero waiting time. We designed the lengths of pre-critical and post-critical sections to be 8s and 6s respectively for both interactions. From the first interaction, we calculated α_{pre} by dividing measured utility by the section length. From the second interaction, α_{post} was calculated by subtracting the utility of the pre-critical section from the total, and dividing the remaining by the post-critical section length.

To measure α_{cPTC} and α_{wait} , we used three interactions, varying the cPTC lengths among 15, 30, and 45s, and another three interactions, varying wait times among 5, 10, and 15s. The orders of timings for each set of three interactions were either decreasing or increasing, and counter-balanced among all the participants. Using the mean utilities from all participants for each of the three interactions, we performed linear regressions with least square errors to calculate α_{cPTC} and α_{wait}.

We could control the exact lengths for cPTC and waiting times by preparing conversation contents for the robot with each length of cPTC, and using a timer to control the waiting time of customers before the robot answered. In order to focus on the ranges where the utility of the conversation still remains positive, we chose the maximum durations for cPTC and wait time based on the results from a previous study [11].

- **Part 2**

In this part, we measured utility factors for navigation with three route-guidance interactions. The first guidance was performed at 1m/s speed for 20s, from which we measured α_{normal} by dividing measured utility by the travelling time. The second guidance was performed at 0.5m/s for 20s, from which we measured α_{slow} with the same method. The last guidance was performed by moving at 1m/s for 20s, with a 20-second stop in the middle. Using the utility from the third interaction, we calculated α_{stop} by subtracting the expected utility from movement using α_{normal} measured above, and dividing the remaining utility by the stopping time.

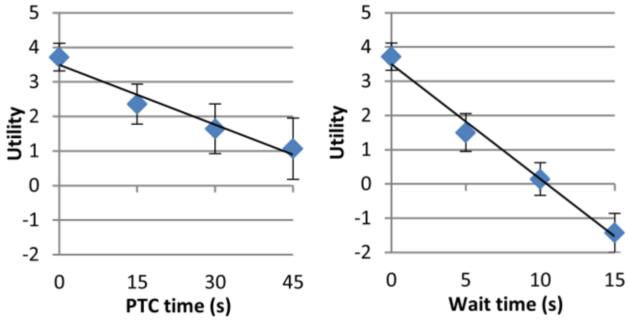
The interactions did not include any conversation; instead the experimenter told the participants where the robot would guide them to before each of the interactions. The destinations for each trial were the same for each participant.

- **Part 3**

The last part was used for measuring the weights of conversation and navigation when both occur in a single interaction. Only one interaction was used, wherein the robot conducted a 20-second conversation to determine where the customer wanted to go, and then guided the

TABLE 1
MEASUREMENT FOR UTILITY FACTORS IN THE DATA COLLECTION

	Interaction	Measure	Values
$\boldsymbol{\alpha_{pre}}$	A conversation with only 8-second pre-critical section	U_{pre}/T_{pre}	0.40
$\boldsymbol{\alpha_{post}}$	A conversation with pre- (8s) and post-critical (6s) sections	$\frac{U_{all}-U_{pre}}{T_{post}}$	0.08
$\boldsymbol{\alpha_{cPTC}}$	Conversations varying cPTC time among {0, 15, 30, 45} seconds	Linear regression	-0.06
$\boldsymbol{\alpha_{wait}}$	Conversations varying waiting time among {0, 5, 10, 15} seconds	Linear regression	-0.34
$\boldsymbol{\alpha_{normal}}$	A 20-second navigation at 1m/s	$\frac{U_{normal}}{T_{normal}}$	0.18
$\boldsymbol{\alpha_{slow}}$	A 20-second navigation at 0.5m/s	U_{slow}/T_{slow}	0.11
$\boldsymbol{\alpha_{stop}}$	A 20-second navigation at 1m/s with a 20-second stop in the middle	$\frac{U_{all}-U_{normal}}{T_{stop}}$	-0.22
$\boldsymbol{\omega_{conv}}$	20-second conversation followed by 20-second navigation at 1m/s	Direct evaluation	0.49
$\boldsymbol{\omega_{nav}}$			0.51



(a) (b)

Fig. 10. Mean, standard error of utility and linear regressions for (a) cPTC duration and (b) wait time in conversation

customer to the destination by navigating for 20s at 1m/s speed.

After the interaction, we asked the participants to provide a weighted score for the conversation and navigation elements, based on the degree to which they thought each contributed to the overall service. The weights were normalized to sum to 1.0, in order to calculate valid average weights among all participants.

2) Result

Table 1 summarizes how we measured each utility factor and the mean values measured. The total utility measured from each section is denoted by U subscripted by section name, and T denotes the section duration. Utility factors were measured in units of utility per second. Fig. 10 shows the mean and standard errors of utilities for cPTC and wait time, and also the results of linear regressions represented by straight lines. The decision coefficients R^2 are respectively 0.9572 and 0.9887 for each regression, indicating good fitting of the linear model.

From Table 1, we can see that the utility factors α_{pre}, α_{post}, α_{normal} and α_{slow} are positive, while the utility factors α_{cPTC}, α_{wait}, and α_{stop} are negative. ω_{conv} and ω_{nav} have similar values, with ω_{nav} slightly larger, meaning that navigation played a slightly more important role in the overall effect of the interaction.

By substituting the utility and weight factors into the previously defined equations, we can obtain the following three important conclusions which can be applied to the interaction scenario in the data collection:

1. Eq. (2) is satisfied. This means cPTC is effective in improving utility when waiting time would happen during conversation.
2. Eq. (8) is satisfied. This means nPTC is effective in improving utility when stopping would happen during navigation.
3. Eq. (11) is satisfied. This means using nPTC before stopping is more effective than using cPTC after stopping when active conversation is used.

These three conclusions are not guaranteed to be universally true, and they may change depending on different service scenarios. But verifying the equations for a given scenario provides useful implications for managing human-robot interactions using conversation and navigation.

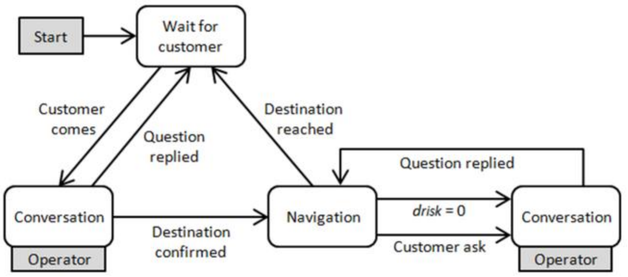


Fig. 11. The state machine of the robot for conversation and navigation

6. SCHEDULING ALGORITHM

We present an algorithm for scheduling the tasks for a human-robot team. First, we present an interaction structure combining both conversational and navigational interactions in a state machine. Then, we define an algorithm for scheduling the tasks applied to the interaction structure.

A. Interaction Structure

We define an interaction structure which can be applied to a variety of interactions combining conversation and navigation, as shown by the state machine in Fig. 11. The robot starts by waiting for the customer, and it initiates a conversation by asking for the customer's request. The conversation can be ended by the robot replying to the customer's question, or it can continue with navigation to some destination. A conversation is always performed before a navigation task so that the destination can be known from talking to the customer.

During navigation, there are two cases for insertion of a conversation: one is the active conversation initiated by the robot when d_{risk} is zero, and the other is initiated by the customer whenever she/he has some question to ask. Multiple conversations from either of these cases can be performed during a navigation task. The navigation ends when the destination is reached, and the robot starts to wait for new customers again.

For the operator, the only time when operation is needed is during conversational critical sections, because active conversation is used whenever a critical section occurs in navigation. In this case, the operator has to both correct localization of the robot and answer the question from the customer.

B. Algorithm Implementation

TABLE 2
PRIORITIES AND SORTING CRITERIA FOR EACH SECTION

Sections		Priority	Sorting Criteria
Conversation	Critical	High	--
	Pre-Critical (including cPTC)	Medium	Increasing order of section start time
Navigation	$d_{risk} > 0$	Low	Increasing order of d_{risk}

```
Scheduling-Algorithm
1:  $R_{sorted}$ = Robots sorted by task priority
2:  $T_{release}$ = Time to finish current operation

3:  for each $r$ in $R_{sorted}$:
4:      if $d_{risk}(r) > 0$
5:          Let $r$ use nPTC if $T_{toCritical}(r) < T_{release}$.
6:      else
7:          Let $r$ initiate an active conversation.
8:          Let $r$ use cPTC if $T_{pre} < T_{release}$.
9:      $T_{release} = T_{release} + \bar{T}_{op}(r)$

10: if operation is finished
11:     Allocate operator to the first robot in $R_{sorted}$.
```

Our scheduling algorithm can be applied for optimizing team performance for the interaction structure defined above. It includes sorting of the robots to determine the order of operation, and task scheduling using interaction-management strategies.

In order to decide the task sequence of the operator, the algorithm sorts the robots by increasing order of time to enter a critical section. We can use Table 2 to define priorities of the robots based on current interaction status, wherein a higher priority means earlier entry to a critical section in the future. The column of “Sorting Criteria” in Table 2 specifies how to sort the robots with the same priority. It is not necessary to sort robots in the critical section of conversation, because the use of cPTC will ensure that only one robot will be in this state.

Using the array of sorted robots, we can define the algorithm by the pseudo-code “Scheduling-Algorithm” as listed. R_{sorted} represents the robots sorted by priority from the highest to the lowest. It uses the variable $T_{release}$ to track the estimated time before the operator can be allocated to each robot, which is initialized by the remaining time of current operation.

Through lines 3~9, the algorithm iterates through each of the robots to manage their interactions. For navigating robots with positive d_{risk}, a robot is assigned to use nPTC (that is, slow speed) if $T_{toCritical}$ calculated by (3) is less than $T_{release}$. An active conversation is started whenever d_{risk} becomes 0, and cPTC is used if the time to complete the pre-critical section is less than $T_{release}$. $T_{release}$ is updated by adding the estimated operation time ($\bar{T}_{op}$) after each iteration, which is the estimated time of operator allocation for the next robot. Finally, the operator is allocated to the first robot in the array after each operation is finished.

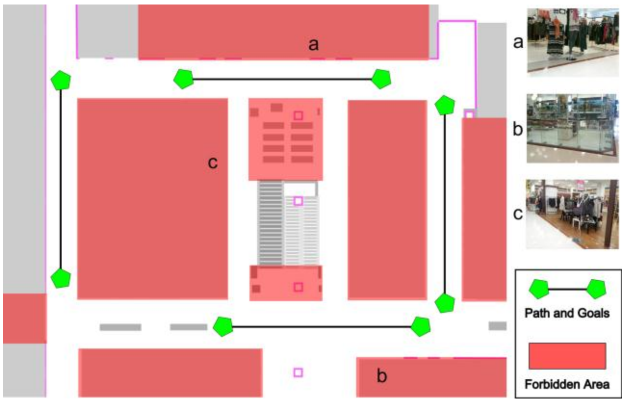


Fig. 12. The environment of the field experiment

The algorithm ensures that the robots are operated in the order of their critical sections, and the performance in terms of utility can be improved by managing interactions using PTC strategies and active conversation. We will verify the effectiveness of the scheduling algorithm in improving the performance of real robots in the field experiment presented in the next section.

7. FIELD EXPERIMENT

We conducted a field experiment aiming to answer the following questions:

1. Is our risk model effective enough to enable safe navigation of multiple robots?
2. How well does our model help in coordinating multiple robots in providing services including conversation and navigation from the customers’ perspective?

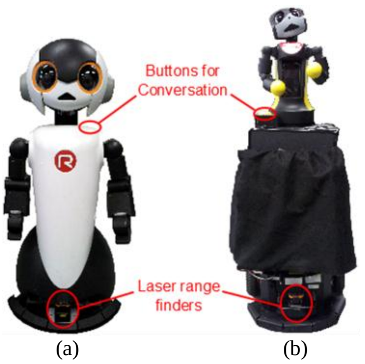
A. *Settings*

We recruited 19 participants (13 male, 6 female, average age 21.2 with standard deviation of 1.8) to act as customers. Most of them were undergraduate students, and had some experience with robots in other experiments conducted in our lab, but none of them had participated in the data collection presented in Section 5-C.

We used a researcher in our lab as the operator during the whole experiment. We did not tell the operator about our research objectives, because we wanted the operator to keep consistent effort in all experiment conditions.

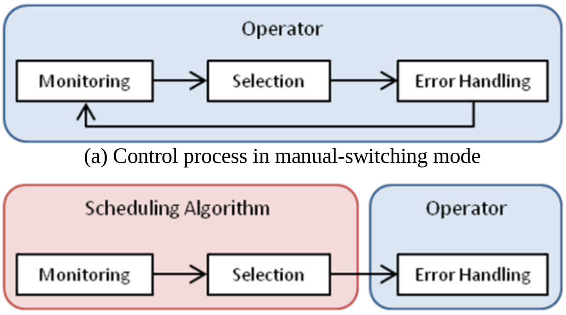
1) Environment and Robots

The environment where we deployed the robots was a shopping area as shown in Fig. 12. The forbidden areas cover the majority of the areas alongside the corridors, including the examples shown in Fig. 3. We used four robots to provide route-guidance services, and each of them traversed a path between two goal positions as shown in the figure. The lengths of the guiding paths ranged from 19.1 to



(a) (b)

Fig. 13. Robots with buttons for conversation and laser range finders



(a) Control process in manual-switching mode

(b) Control process in auto-switching mode

Fig. 14. Control processes in manual and auto switching modes

19.5 meters.

We used two Robovie-R3 humanoid robots (Fig. 13 (a)), and two Robovie-MR2 mini-humanoids mounted on top of Pioneer cart bases (Fig. 13 (b)). Each robot was equipped with a forward-facing Hokuyo UTM-30LX laser range finder at a height near the ground, which can measure the distances to obstacles with a range of 270 degrees and distance up to 30 meters. Using data from the range finders and a prebuilt environment map, each robot was able to estimate its most-likely position using particle filter localization. An additional risk estimation algorithm was used depending on the condition of experiment as will be described.

Although there were visual differences between the robots, both types of robots operated with the same conversation module, and navigated with the same ranges of speed, which were 1.0 m/s for normal navigation, and 0.5 m/s for slow speed.

The task of the robots was to guide the customers to the destinations by conducting conversational and navigational interactions as described in the interaction structure in Fig. 11. The robot greets the customer whenever the customer approaches, and it starts guiding when the destination is known. The customer can trigger a conversation at any time during navigation by touching the shoulder of the humanoid robots or a button mounted on top of the cart base (Fig. 13).

2) *Conditions*

In order to validate the effectiveness of our system, we conducted a within-participants comparison using two conditions in the experiment, namely *Manual-Switching* and *Auto-Switching*.

Fig. 14 illustrates the control processes in the two conditions. Generally, the control process consists of three steps: *Monitoring*, *Selection*, and *Error Handling*. The "Monitoring" step is for monitoring the status of each robot, and "Selection" is for selecting the robot to control based on observed statuses. "Error handling" is the step in which the operator takes actions in assisting the robot, which includes error handling for conversation, navigation, or both in sequence.

In manual-switching mode (Fig. 14 (a)), the operator monitors the robots using a user interface wherein the status of each robot (conversation state, location, sensor data, etc.) are visualized in real time. Then, the operator selects the robot which has the most urgent error to be handled based on his/her observation. The three steps form a loop, and we assume that the operator cannot process multiple steps at the same time, such as monitoring during error handling.

As a baseline condition, we did not use the risk estimation algorithm in manual-switching mode, and it was the operator's duty to ensure safety of each robot. To do so, the operator can temporarily stop navigation of a single robot or all of the robots whenever he/she perceives some error or potential risk in navigation.

In order not to cause actual damage to the experiment space, we added additional 0.5-meter buffers to each forbidden area at the time of map creation, so that even if a robot accidentally enters a forbidden area (which is larger than actual), it will not cause real damage if stopped in time. A member of our lab was assigned to each robot to immediately stop the robot when it accidentally entered a forbidden area.

In auto-switching mode (Fig. 14 (b)), the "monitoring" and "selection" processes are handled by the scheduling algorithm as described in Section 6. The operation interface is automatically switched to the robot selected by the scheduling algorithm, allowing the operator to focus on error handling for that given robot. The risk estimation algorithm was used to provide safety and also to estimate the time until danger for task scheduling.

3) *Subjective measure of Utility*

In this experiment, we used subjective measurement by letting the participants evaluate their utility after each interaction with an integer between [-5 ... 5] using the same criteria as in the data collection. We directly measured people's subjective responses in order to evaluate our model. From the utilities evaluated by customers, we used two criteria for measuring the performance of the human-robot team:

The first criterion is *Mean Utility*, which is the average utility from all interactions conducted by the robots. Mean utility describes how well the robots performed on average per individual interaction.

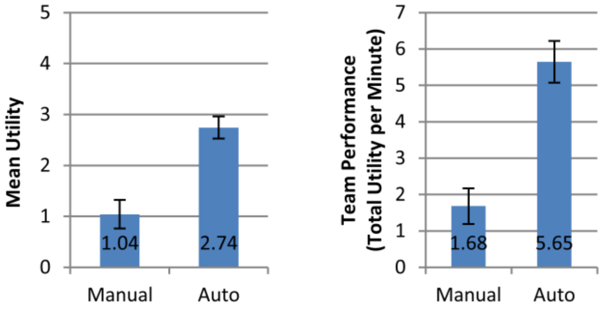
The second criterion is *Team Performance*, which is the sum of utilities from all interactions conducted by the robot team during a unit amount of time. We can use (12) to calculate team performance, where U_r^i is the utility of each interaction for each robot evaluated by the customer. We measured team performance for each minute during the experiment, which describes the overall efficiency of the robot team in producing utility through interactions.

$$Performance = \frac{\sum_{r \in Robots} \sum_{i \in Interactions} U_r^i}{Time} \qquad (12)$$

4) *Hypotheses*

To evaluate the effectiveness of our system, we proposed the following hypotheses to be tested through the experiment:

Hypothesis 1: The risk estimation algorithm can ensure



(a) (b)

Fig. 15. Mean and standard error of (a) interaction utilities and (b) team performance for manual-switching and auto-switching mode

safety of navigation by preventing robots from entering forbidden areas.

Hypothesis 2: Auto-switching mode will perform better than manual-switching mode in terms of mean utility.

Hypothesis 3: Auto-switching mode will perform better than manual-switching mode in terms of team performance.

The first hypothesis tests whether our risk model can ensure safety, which is a fundamental requirement for using robots for navigational tasks in real social environments. The second and third hypotheses test whether our interaction models incorporated into the auto-switching mode can improve the team performance compared to a manual baseline condition.

B. *Procedure*

The experiment was conducted over two days. To counterbalance the ordering of conditions, we divided trials on each day into two sessions, one with auto-switching and one with manual-switching, and their order was reversed on the second day. 10 participants performed on the first day, and 9 participants on the second day. In each session, four robots were deployed simultaneously to provide services for the defined goal positions, and each participant conducted four interactions in total, one with each of the four robots.

In order to keep the balance of conversation and navigation in each condition for making a fair comparison, we had the customers conduct at least one conversation during each navigational service is being conducted. If the robot did not initiate an active conversation before reaching the destination, the participants were instructed to trigger a conversation at any time by pressing the buttons mounted on the robots.

We counted the number of times when the robots entered forbidden areas in each condition to measure the effectiveness of the risk estimation algorithm in ensuring safety during navigation. We also took logs for the timings of interaction sections and operator activities, which were used for further analysis after the experiment.

C. *Results*

During the two days of the experiment, 70 interactions were conducted in auto-switching mode, and 71 interactions were conducted in manual-switching mode. 11 interactions were classified as failures due to technical problems such as network drop-out and battery trouble, and these interactions were not included in our data analysis.

TABLE 3
MEAN DURATIONS OF SECTIONS AND OVERALL INTERACTION

	Conversation(s)		**Navigation (s)**		**Overall Interaction (s)**	
	cPTC	***Critical***	***nPTC***	***Critical***	***Length***	***Interval***
Auto	12.9	4.3	14.0	0	79.5	11.4
Manual	11.7	4.9	0	27.9	91.2	12.0

1) *Observation of Safety*

In manual-switching mode, the robots entered forbidden areas five times during the two days of experiment. From the observation of localization history, we found that the robots entered forbidden areas because of incorrect position estimates from the particle filter localization, so that the robots considered themselves to be within the safe area while actually in forbidden areas.

When these robots entered forbidden areas, they were stopped by our staff before they could do any damage to the environment. We also suspended execution of the experiment by stopping the other robots, and we instructed the operator to manually drive the robot into a safe location in order to resume the experiment.

The robots never entered forbidden areas in auto-switching mode throughout about 1.5 km of navigation during the experiment. This result verifies our first hypothesis: that our risk model can successfully achieve safe navigation of multiple robots in auto-switching mode.

2) *Comparison of Performance*

The mean and standard error of interaction utilities measured from the participants in each mode are shown in Fig. 15 (a). We carried out analyses using linear mixed-effects models for utilities having the *condition* factor and *robots* as fixed effects and *participants* as a random effect. There was a significant main effect in *condition* (F(1, 115.195)=31.210, p<.001). The *robot* factor (F(3,115.051)=1.383, p=.251) and interaction with *condition* (F(3,114.992)=.746, p=.527) were not significant. These results verify our second hypothesis that auto-switching mode produces higher performance in terms of interaction utility.

The mean and standard error of team performance in each mode is shown in Fig. 15 (b). From the figure we can see that the mean performance in auto-switching mode is much higher than that of manual-switching mode. A t-test reported a significant difference between the two conditions (t=5.302, p<.001). These results verify our third hypothesis that auto-switching mode produces higher performance in terms of team performance.

3) *Analysis of Interaction Timings*

Since the utility model indicates that waiting times in interactions greatly affect utility, we compared the mean durations of the sections related to the waiting times, shown in Table 3. For cPTC durations, a t-test reported no

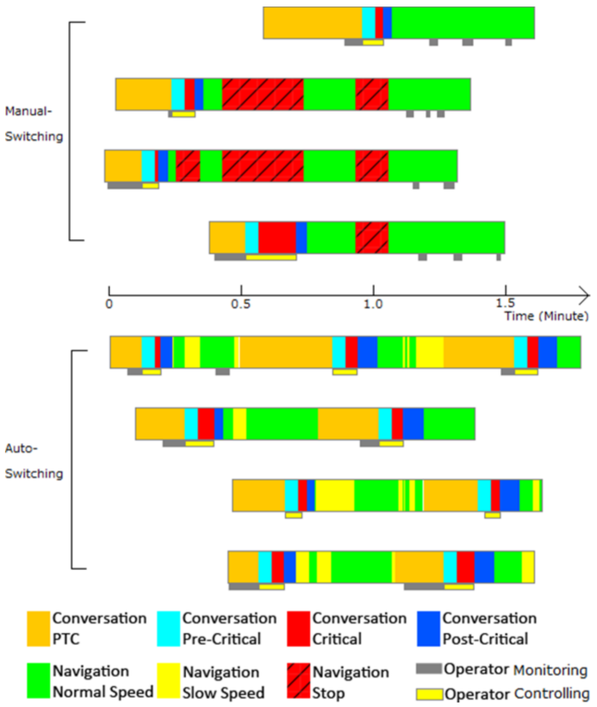


Fig. 16. Visualization of interaction timings for manual-switching and auto-switching modes during about 2-minute time periods

TABLE 4
OPERATION DATA FOR SWITCHING AND ERROR HANDLING

	(Monitoring + Selection) (s)	**Time (T_{op}) (s)**		**Handlings per Interaction (N_{op})**
		Conv.	***Nav.***	
Auto	--	3.0	1.1	2.0
Manual	9.9	4.0		1.8

significant difference between manual and auto conditions (*t*=-1.255, *p*=.211), and for conversational critical timings, a t-test also reported no significant difference (*t*=1.033, *p*=.303). Those results indicate that there is no statistical evidence showing that the robots performed differently in terms of conversation between the two conditions.

On the other hand, the robots performed differently for slowing and stopping of navigation in the two conditions. nPTC (slow speed) was used for 14.0s on average per interaction in auto-switching mode, and navigational critical section time was zero because active conversation was used whenever stopping of a robot occurred. In manual-switching mode, the critical section time for navigation was 27.9s on average during each task, and nPTC time was zero since it was not used.

The mean duration of overall interaction including conversation and navigation increased from 79.5s in auto-switching mode into 91.2s in manual-switching mode, which can be attributed to the increased stopping time in navigation. For the interval between interactions, measuring the idle time of each robot from the end of one interaction until the arrival of the next customer, a t-test reported no significant difference (*t*=-1.252, *p*=.213) between the two conditions. It means the customers (role-played by participants) arrived with similar rates in the two conditions of the experiment.

To understand what caused different waiting times in the two conditions, we visualized the timing of interactions from the logs taken in the experiment. Fig. 16 shows one example of the interaction timings of the 4 robots for each mode, covering a 2-minute time period. As shown in the figure, the operator had to stop navigation of the other robots when he was operating a robot in manual-switching mode, because he could not monitor the safety status of the other robots while focused on error-handling of one robot. Frequent switching among the robots happened during the second minute in this mode, indicating that the operator had to frequently check safety for each robot's navigation when he was not engaged in another operation task.

In auto-switching mode, operation for one robot did not cause stopping of other robots, because the risk estimation algorithm could automatically check the safety status. nPTC was used before stopping, and active conversation was used during stopping of navigation.

From the analysis of interaction timings, we can conclude that 1) the risk estimation algorithm can reduce unnecessary stopping of the robots which are not attended by the operator, and 2) nPTC and active conversation strategies can manage stopping of navigation in an acceptable way that prevents interruption of service.

4) *Analysis of Operation*

Table 4 summarizes the operation data in each condition. Since we cannot separate the time for monitoring and selecting the robot in manual-switching mode, we define *Switching Time* as a combination of the two steps, starting from the moment the operator switches to a robot until he begins error handling or switches to another robot.

The results show that the operator spent a similar amount of time for error handling in the auto-switching (4.1s) and manual-switching (4.0s) modes. But the operator spent a relatively large amount of time for monitoring and selecting robots in the manual-switching mode (9.9s), which was not necessary in the auto-switching mode.

$$Fanout = \frac{T_{interaction}}{T_{switching} + T_{op} \times N_{op}} \qquad (13)$$

Using the operation and interaction timings in Table 4 and Table 3, we calculated the fan-out by (13), which is the interaction time divided by the operation time in each interaction. The fan-out is 9 in auto-switching mode and 5 in manual-switching mode, indicating that a larger number of robots can be controlled by the operator in auto-switching mode. Note that the fan-out number does not indicate the robot team has highest performance at that team size, because the fan-out calculation does not include the utility model for calculating the interaction performance.

D. *Summary*

From the results of the experiment, we can conclude that all the hypotheses raised in A-4) were verified, which also answers the questions posed at the beginning of this section:

1. Our safety model can enable safe navigation of multiple semi-autonomous robots in a real environment such as a shopping area. This task is not feasible without using the safety model, because the safety requirement cannot be achieved by fully manual operation.
2. Our interaction models and algorithms incorporated into the auto-switching mode enabled the human-robot team to perform better in providing services compared to a baseline condition without using our models.

Because of constraints in time and number of people, we could not conduct the experiment using larger number of operators. But we think the comparison experiment provides a valid evaluation for the effectiveness of our system, considering the performance of the operator for error handling as a constant factor.

8. DISCUSSION

A. Limitations

1) Utility Model

We developed the utility model based on timing of interactions, but previous studies by Duffy [35], Sabanovic, et al. [36], and Steinfeld et al. [37] suggest that many other aspects such as human-likeness, gestures, and social etiquette can also affect customer utility in human-robot interactions. Including such factors may improve the utility model in future studies when computing the utility values at a finer grain is required.

We think that a possible extension to the utility model is to consider possible overlap between conversation and navigation, which is not included in our current model. There is a study [38] modeling the effect of simultaneous speech and locomotion for human-robot interaction, which could be used to extend our model.

2) Operator Model

In this study, we assumed that the system automatically gives tasks to the operator, which may not be the best strategy for all human-robot systems [18]. The automatic task allocation in this study is based on a reliable method (i.e. risk estimation) to detect potential system errors, which eliminates the operator's workload for monitoring the robot fleet. However, if operators are micromanaged, they have no chance to use their judgment and overall perspective to manage the robot team, especially for skilled expert operators.

We also did not consider possible mistakes in localizing the robot by the operator. Although we have measured operation accuracy and included the distribution of errors when re-localizing the robot, it does not eliminate the possibility that the operator may misdiagnose the position of the robot, which could be a risk in a noisy or highly-symmetrical environment.

B. Conclusions

We have presented a systematic mechanism to combine the management of conversation and navigation for a human-robot team. We have achieved this by extending our previous studies in two major respects:

First, we have integrated the previous studies for conversation [11], [27] and navigation [12] by proposing new hypotheses on the utility model. We have also extended the utility model for navigation by considering the relationship between speed (including stopping) and walking time. We have demonstrated that the integrated model can evaluate the task performance of interaction including conversation and navigation in a quantitative way, which could not be achieved in the previous studies when only considering conversation or navigation separately.

Second, we have evaluated the effectiveness of our integrated system in real-world settings. The risk estimation model was introduced in the previous study [12], but we did not directly evaluate its safety performance in a comparison experiment. In the current study, we have carried out a new experiment to demonstrate its effectiveness in enabling safe navigation, which was impossible when only using a traditional localization algorithm. We believe our risk estimation algorithm can be used together with existing localization algorithms in order to achieve high level of safety for people and robots.

ACKNOWLEDGMENT

We would like to thank N. Iwasaki, Gregory Cole and Dr. Satoru Satake for technical support. We appreciate the help from Alex McGilvray, Anderson Li, Phoebe Liu, Chaoran Liu, Derek Ho, Hwapyeong Cho, Deneth Kurunarathne, Philip Chan, Kanae Wada, Tayuka Kitade, Yasuhiko Hato and Ryou Murakami for the field experiment. We would also like to thank Sayaka Taniguchi and Dr. Satoshi Koizumi for organizing the data collection and experiment.

REFERENCES

[1] S. Thrun, et al., "MINERVA: A second-generation museum tour-guide robot," in *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 1999-2005, 1999.

[2] M. Shiomi, T. Kanda, H. Ishiguro and N. Hagita, "Interactive humanoid robots for a science museum," in *Proc. 1st ACM/IEEE Int. Conf. Human-Robot Interaction*, pp. 305-312, 2006.

[3] H.-M. Gross, et al., "Shopbot: progress in developing an interactive mobile shopping assistant for everyday use," in *IEEE Int. Conf. on Systems, Man, and Cybernetics*, pp. 3471-3478, 2008.

[4] T. Kanda, M. Shiomi, Z. Miyashita, H. Ishiguro and N. Hagita, "A communication robot in a shopping mall," *IEEE Trans. Robotics, 26(5)*, pp. 897-913, 2010.

[5] R. Siegwart, et al., "Robox at Expo.02: A large-scale installation of personal robots," *Robotics and Autonomous Systems, 42*, pp. 203-222, 2003.

[6] K. Hayashi, D. Sakamoto, T. Kanda, M. Shiomi, S. Koizumi, H. Ishiguro, T. Ogasawara and N. Hagita, "Humanoid robots as a passive-social medium - a field experiment at a train

station-," in *Proc. 2nd ACM/IEEE Int. Conf. Human-Robot Interaction*, pp. 137-144, 2007.

[7] C.T. Ishi, S. Matsuda, T. Kanda, T. Jitsuhiro, H. Ishiguro, S. Nakamura and N. hagita, "Robust speech recognition system for communication robots in real environments," in *IEEE Int. Conf. on Humanoid Robots*, pp. 340-345, 2006.

[8] M. Shiomi, D. Sakamoto, T. Kanda, C.T. Ishi, H. Ishiguro and N. Hagita, "A semi-autonomous communication robot-a field trial at a train station-," in *Proc. 3rd ACM/IEEE Int. Conf. Human-Robot Interaction*, pp. 303-310, 2008.

[9] D.F. Glas, S. Satake, F. Ferreri, T. Kanda and N. Hagita, "The network robot system: enabling social human-robot interaction in public spaces," *Journal of Human-Robot Interaction, 1(2)*, pp. 5-32, 2012.

[10] M. Montemerlo, S. Thrun, D. Koller and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *Proc. National Conf. on Artificial Intelligence*, pp. 593-598, 2002.

[11] K. Zheng, D.F. Glas, T. Kanda, H. Ishiguro and N. Hagita, "Designing and implementing a human-robot team for social interactions," *IEEE Trans. Systems, Man, and Cybernetics: Systems, 43(4)*, pp. 843-859, 2013.

[12] K. Zheng, D.F. Glas, T. Kanda, H. Ishiguro and N. Hagita, "Supervisory control of multiple social robots for navigation," in *Proc. 8th ACM/IEEE Int. Conf. Human-Robot Interaction*, pp. 17-24, 2013.

[13] D.R. Olsen and M.A. Goodrich, "Metrics for evaluating human-robot interactions," in *Proc. PERMIS*, 2003.

[14] D.R. Olsen and S.B. Wood, "Fan-out: measuring human control of multiple robots," in *Proc. Conf. Human Factors in Computing Systems*, pp. 231-238, 2004.

[15] J.W. Crandall, C.W. Nielsen, and M.A. Goodrich, "Towards predicting robot team performance," in *IEEE Int. Conf. on Systems, Man, and Cybernetics*, pp. 906-911, 2003.

[16] J.W. Crandall and M.A. Goodrich, "Characterizing efficiency of human robot interaction: a case study of shared-control teleoperation," In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp. 1290-1295, 2002.

[17] J.W. Crandall, M.A. Goodrich, D.R. Olsen and C.W. Nielsen, "Validating human-robot systems in multi-tasking environments," *IEEE Trans. Systems, Man, and Cybernetics – Part A: Systems and Humans, 35(4)*, pp. 438-449, 2005.

[18] J.W. Crandall, M.L. Cummings, M. Penna and P.M.A. de Jong, "Computing the effects of operator attention allocation in human control of multiple robots," *IEEE Trans. Systems, Man, and Cybernetics – Part A: Systems and Humans, 41(3)*, pp. 385-397, 2011.

[19] P. Scerri, K. Sycara, and M. Tambe, "Adjustable autonomy in the context of coordination," in *AIAA 3rd Unmanned Unlimited Technical Conference, Workshop and Exhibit*, 2004.

[20] K. Savla, C. Nehme, T. Temple, and E. Frazzoli, "On efficient cooperative strategies between humans and UAVs in a dynamic environment," in *AIAA Guidance, Navigation and Control Conference*, August 2008.

[21] K. Savla, T. Temple, and E. Frazzoli, "Human-in-the-loop vehicle routing policies for dynamic environments," in *IEEE Conference on Decision and Control*, December 2008.

[22] S. Mau and J. Dolan, "Scheduling for humans in multirobot supervisory control," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp. 1637-1643, 2007.

[23] A. Richards, Y. Kuwata, and J.P. How, "Experimental demonstrations of real-time MILP control," in *Proc. of the AIAA Guidance, Navigation, and Control Conference*, 2003.

[24] D.F. Glas, T. Kanda, H. Ishiguro and N. Hagita, "Teleoperation of multiple social robots," *IEEE Trans. Systems, Man, and Cybernetics – Part A: Systems and Humans, 42(3)*, pp. 530-544, 2012.

[25] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust Monte Carlo localization for mobile robots," *Artificial Intelligence, 128(1-2)*, pp. 99–141, 2000.

[26] J.-S. Gutmann, "Markov-Kalman localization for mobile robots," in *Proc. Int. Conf. on Pattern Recognition (ICPR)*, pp. 601-604, 2002.

[27] K. Zheng, D.F. Glas, T. Kanda, H. Ishiguro and N. Hagita, "How many social robots can one operator control?" in *Proc. 6th ACM/IEEE Int. Conf. Human-Robot Interaction*, pp. 379-386, 2011.

[28] http://slam6d.sourceforge.net

[29] S. Thrun, W. Burgard and D. Fox, Probabilistic Robotics. Cambridge, MA: MIT Press, pp.250-267, 2005.

[30] J. Borenstein and L. Feng, "Measurement and correction of systematic odometry errors in mobile robots," *IEEE Trans. Robotics and Automation, 12(6)*, pp. 869-880, 1996.

[31] T. Shiwa, T. Kanda, M. Imai, H. Ishiguro and N. Hagita, "How quickly should communication robots respond?" in *Proc. 3rd ACM/IEEE Int. Conf. Human-Robot Interaction*, pp.153-160, 2008.

[32] D.F. Glas, T. Kanda, H. Ishiguro and N. Hagita, "Simultaneous teleoperation of multiple social robots," in *Proc. 3rd ACM/IEEE Int. Conf. Human-Robot Interaction*, pp.311-318, 2008.

[33] D.F. Glas, T. Kanda, H. Ishiguro and N. Hagita, "Field trial for simultaneous teleoperation of mobile social robots," in *Proc. 4th ACM/IEEE Int. Conf. Human-Robot Interaction*, pp. 149-156, 2009.

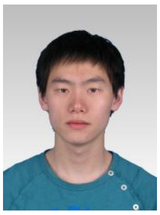
[34] E. Sviestins, N. Mitsunaga, T. Kanda, H. Ishiguro and N. Hagita, "Speed adaptation for a robot walking with a human," in *Proc. 2nd ACM/IEEE Int. Conf. Human-Robot Interaction*, pp. 349-356, 2007.

[35] B.R. Duffy, "Anthropomorphism and the social robot," *Robotics and Autonomous Systems, (42)*, pp. 177-190, 2003.

[36] S. Sabanovic, M.P. Michalowski and R. Simons, "Robots in the wild: observing human-robot social interaction outside the lab," in *Proc. 9th Int. Workshop on Advanced Motion Control*, pp. 596-601, 2006.

[37] A. Steinfeld, T. Fong, D. Kaber, M. Lewis, J. Scholtz, A. Schultz and M. Goodrich, "Common metrics for human-robot interaction," in *Proc. 1st ACM/IEEE Int. Conf. Human-Robot Interaction*, pp. 33-40, 2006.

[38] M. Shimada and T. Kanda, "What is the appropriate speech rate for a communication robot?" *Interaction Studies, 13(3)*, pp. 408-435, 2012.

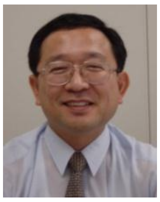


Kuanhao Zheng received his S.B. degree in computer science and technology from Tsinghua University, Beijing, China in 2008, and received his M. Eng. in engineering science in 2011 from Osaka University, Osaka, Japan. He has been an internship researcher at the Intelligent Robotics and Communication Laboratories (IRC) at the Advanced Telecommunications Research Institute International (ATR) in Kyoto, Japan since 2009. He has been a Ph.D. candidate in the Graduate School of Engineering Science at Osaka University since 2011.

Dylan F. Glas received his Ph.D. in robotics from Osaka University in 2013, under Prof. Hiroshi Ishiguro. He received his M.Eng in Aerospace Engineering from MIT in 2000 and two S.B. degrees, one in Aerospace Engineering and one in Earth, Atmospheric, and Planetary Sciences, also from MIT in 1997. From 1998-2000 he worked in the Tangible Media Group at the MIT Media Lab. He is currently group leader of the Department of Cloud Intelligence at the Intelligent Robotics and Communication Laboratories (IRC), Advanced Telecommunications Research Institute International (ATR) in Kyoto, Japan. His research interests include social human-machine interaction, cloud network robot systems, ubiquitous sensing, teleoperation for social robots, and machine learning.

Takayuki Kanda (M'04) received his B. Eng, M. Eng, and Ph. D. degrees in computer science from Kyoto University, Kyoto, Japan, in 1998, 2000, and 2003, respectively. This author became a Member (M) of IEEE in 2004. From 2000 to 2003, he was an Intern Researcher at ATR Media Information Science Laboratories, and he is currently a Senior Researcher at ATR Intelligent Robotics and Communication Laboratories, Kyoto, Japan. His current research interests include intelligent robotics, human-robot interaction, and vision-based mobile robots.

Hiroshi Ishiguro received a D. Eng. in systems engineering from the Osaka University, Japan in 1991. He is currently Professor of Department of Systems Innovation in the Graduate School of Engineering Science at Osaka University (2009–). He is Group Leader (2011–) of Hiroshi Ishiguro Laboratory and ATR Fellow (2010–) at the Advanced Telecommunications Research Institute, where he previously worked as Visiting Researcher (1999–2002). He was previously Research Associate (1992–1994) in the Graduate School of Engineering Science at Osaka University and Associate Professor (1998–2000) in the Department of Social Informatics at Kyoto University. He was also Visiting Scholar (1998–1999) at the University of California, San Diego, USA. He was Associate Professor (2000–2001) and Professor (2001–2002) in the Department of Computer and Communication Sciences at Wakayama University. He then moved to Department of Adaptive Machine Systems in the Graduate School of Engineering Science at Osaka University as a Professor (2002-2009). His research interests include distributed sensor systems, interactive robotics, and android science.



Norihiro Hagita (M'85 – SM'99) received his Ph.D. degree from Keio University (Japan) in 1986 in electrical engineering and joined Nippon Telegraph and Telephone Public Corporation (NTT) in 1978. He engaged specially in developing handwritten character recognition. He also stayed as a visiting researcher at Prof. Stephen Palmer's lab in University of California, Berkeley (Dep. of Psychology) during 1989-1990. He is currently the director of ATR Intelligent Robotics and Communication Laboratories (IRC) at the Advanced Telecommunications Research Institute International (ATR) in Kyoto, Japan.

APPENDIX A:
IMPLEMENTATION OF RISK ESTIMATION

The overall workflow of the risk estimation algorithm is illustrated by Fig. 17, which includes update of particles (left-hand side) and a human-in-the-loop (right-hand side).

A. Motion Update

This step follows the same procedure as in the particle-filter localization in [29], but the difference is that we intentionally over-estimate odometry error to spread particles considering large possible odometry errors. Here we refer to nonsystematic errors as described in [30], whose upper bounds are impossible to predict due to various unpredictable factors.

Though the upper bounds are unpredictable, we can measure the distribution of odometry error by running a robot in the environment, and over-spread the particles considering such error. The level of safety depends on the amount of over-estimation for odometry errors. For example, if the standard deviation of an error parameter is measured as σ, then using a standard deviation of 2σ to spread particles would cover all possible errors with 98% probability assuming that the odometry error is a random Gaussian variable.

B. Measurement Update

In this step, the possibilities of position estimates for each particle are updated from sensor measurement. We use a ray-casting function to calculate the distance from a particle to the closest detectable feature in a feature map of the environment. For the ray-casting, we use a feature map which includes only permanent features, created by manually removing movable features such as benches or clothing racks from the map.

Suppose the range finder returns the ranges to objects in N directions, then the incremental possibility $\delta^i(p)$ of a particle p for the measurement in the i-th direction can be calculated by (A1), where $ray^i(p)$ and $\mathrm{range}^{\mathrm{i}}$ are the distances from ray-casting and the range finder, and u is the unit step function, whose value is zero for negative arguments and one for positive arguments. ε_{range} denotes the error threshold for each measurement, considering that some laser scans will be erroneous, e.g. due to reflective surfaces.

$$\delta^i(p) = u(ray^i(p) - range^i + \varepsilon_{range}) \qquad \text{(A1)}$$

Equation (A1) means any measured range is considered to be possible if it does not exceed the ray-casting distance with some threshold. Shorter ranges are permissible because people or movable objects may be observed closer than the expected distance to the fixed features, but the opposite is impossible.

$$Possibility(p) = u(\frac{1}{N}\sum_{i=1}^{N} \delta^i(p) - \varphi_{map}) \qquad \text{(A2)}$$

Using the sensor data including N measurements, the overall possibility of a particle can be calculated by (A2), which compares the proportion of successful matches with a threshold value φ_{map} ($0 \le \varphi_{map} < 1$) representing a minimum required ratio of successful matches. Here, u is the same unit step function as for (2). A proportion of mismatches with the map (i.e. $(1 - \varphi_{map})$) is tolerated, since maps generated in real spaces are unlikely to be 100% accurate even if we use only high-confidence features.

As a result, the possibility of each particle is a binary value of either 1 or 0, representing whether the particle is a possible position estimate or not.

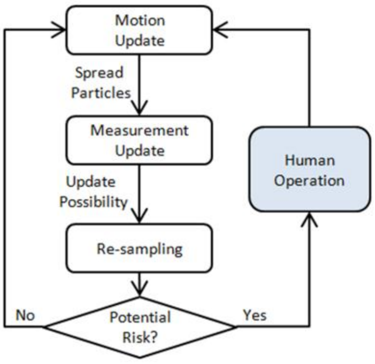


Fig. 17. Procedure of the risk estimation algorithm

C. Re-sampling

Using the possibilities of each particle calculated in the previous step, re-sampling is performed by keeping the particles whose possibility is 1, and removing those with zero possibility. To keep the same number of particles after each update, the dropped particles are replaced by the particles randomly chosen from the remaining ones. After the re-sampling, the particles are used for the next iteration of the algorithm.

$$d_{risk} = \min_{p \in Particles} \{d_{risk}(p)\} \qquad \text{(A3)}$$

Using the updated particles by re-sampling, we can calculate the shortest possible distance to a forbidden area by projecting the particles along their respective motion directions, as in (A3). Here, $d_{risk}(p)$ denotes the distance from a particle to a forbidden area in the motion direction of that particle, and $\mathrm{d}_{\mathrm{risk}}$ is the minimum among all. When $\mathrm{d}_{\mathrm{risk}}$ becomes zero, it means there is a potential risk for the robot to be entering a forbidden area.

D. Human Operation

When a potential risk is detected ($d_{risk} = 0$), the program does not know whether it is an actual dangerous situation or not. For such reason, we use a human operator to confirm the situation by manually localizing the robot. The algorithm assumes that an operator can usually perform more accurate localization than a particle filter, but with some tolerable errors. A pair of parameters $\{\sigma_{op}^{pos}, \sigma_{op}^{ang}\}$ can represent the standard deviations of positional and angular errors in operation.

After each correction by the operator, particles will be redistributed by adding random noises to positions and angles with standard deviations specified by σ_{op}^{pos} and σ_{op}^{ang}, and the algorithm proceeds to the next iteration.

APPENDIX B: TABLE OF SYMBOLS

Parameters for Utility Model

$\boldsymbol{\alpha}$		Utility factor describing change of utility per second
	$\boldsymbol{\alpha_{pre}}$	Utility factor for pre-critical section in conversation
	$\boldsymbol{\alpha_{post}}$	Utility factor for post-critical section in conversation
	$\boldsymbol{\alpha_{cPTC}}$	Utility factor for cPTC in conversation
	$\boldsymbol{\alpha_{wait}}$	Utility factor for waiting time in conversation
	$\boldsymbol{\alpha_{normal}}$	Utility factor for navigation with normal speed
	$\boldsymbol{\alpha_{slow}}$	Utility factor for navigation with slow speed
	$\boldsymbol{\alpha_{stop}}$	Utility factor for stopping time in navigation
$\boldsymbol{N_{op}}$		Number of operation per interaction
$\boldsymbol{T}$		Duration of time for different states or events
	$\boldsymbol{T_{op}}$	Duration of operation time for error handling
	$\boldsymbol{T_{release}}$	Duration of time before the operator is released to a robot
	$\boldsymbol{T_{switch}}$	Duration of time for operator switching among robots
	$\boldsymbol{T_{interaction}}$	Duration of time for a human-robot interaction
	$\boldsymbol{T_{interval}}$	Duration of time interval between human-robot interactions
$\boldsymbol{U}$		Total utility from a human-robot interaction
$\boldsymbol{v}$		Robot's navigation speed
	$\boldsymbol{v_{normal}}$	Robot's normal navigation speed
	$\boldsymbol{v_{slow}}$	Robot's slow navigation speed
$\boldsymbol{\omega}$		Weight factor for utility in different type of interaction
	$\boldsymbol{\omega_{conv}}$	Weight factor for utility in conversation
	$\boldsymbol{\omega_{nav}}$	Weight factor for utility in navigation

Parameters for Risk Estimation Algorithm

$\boldsymbol{d_{risk}}$	Estimated distance to risk from risk estimation algorithm
$\boldsymbol{\varepsilon_{range}}$	Error threshold for matching distances in ray-casting function
$\boldsymbol{\sigma_{op}^{pos}}$	Standard deviation of operation error in localizing the robot's x-y position
$\boldsymbol{\sigma_{op}^{ang}}$	Standard deviation of operation error in localizing the robot's orientation
$\boldsymbol{\varphi_{map}}$	Minimum required proportion of successful matches in ray-casting of particles

Available online at:
http://tsest.org/index.php/TCMS/article/view/252
Download full text article at:
http://tsest.org/index.php/TCMS/article/download/252/161
Cite this work as:
Kuanhao Zheng, Dylan F. Glas, Takayuki Kanda, Hiroshi Ishiguro, and Norihiro Hagita, , , "Supervisory Control of Multiple Social Robots for Conversation and Navigation," *TSEST Transaction on Control and Mechanical Systems*, Vol. 3(10), Pp. 76-92, Jun., 2014.