Designing the Robot Behavior for Safe Human Robot Interactions

Changliu Liu and Masayoshi Tomizuka

Abstract Recent advances in robotics suggest that human robot interaction (HRI) is no longer a fantasy, but is happening in various fields such as industrial robots, autonomous vehicles and medical robots. Human safety is one of the biggest concerns in HRI. As humans will respond to the robot's movement, interactions need to be considered explicitly by the robot. A systematic approach to design the robot behavior towards safe HRI is discussed in this chapter. By modeling the interactions in a multi-agent framework, the safety issues are understood as conflicts in the multi-agent system. By mimicking human's social behavior, the robot's behavior is constrained by the 'no-collision' social norm and the uncertainties it perceives for human motions. An efficient action is then found within the constraints. Both analysis and human-involved simulation verify the effectiveness of the method.

Key words: Human Robot Interactions (HRI), Robot Safety, Motion Planning, Human-in-the-Loop Control, Multi-Agent System

1 Introduction

Recent advances in robotics suggest that human robot interaction (HRI) is no longer a fantasy, but is happening in various fields. In factories, robots are leaving their cages and starting to work cooperatively with human workers [16]. The manufacturers are interested in combining human's flexibility and robot's productivity in flexible production lines [15, 18]. On the other hand, as automated driving is widely viewed as a promising technology to revolutionize today's transportation system [5], substantial research efforts are directed into the field from research groups and com-

Changliu Liu
University of California at Berkeley, Berkeley, CA 94720, e-mail: changliuliu@berkeley.edu

Masayoshi Tomizuka
University of California at Berkeley, Berkeley, CA 94720, e-mail: tomizuka@berkeley.edu

panies [1]. As a consequence, human drivers and autonomous vehicles will interact on the road, which poses new challenges in road safety [25]. Another example is in the field of rehabilitation. In order to rebuild the sensory connection of a patient after a stroke, robots or exoskeletons are needed to guide and assist the patient in walking. There are close physical interactions between the patient and the robot [14]. Others like nursing robots [27] or robot guide dogs [31] are also in great demand and involve HRI.

1.1 The Safety Issues and Existing Solutions

Human safety is one of the biggest concerns in HRI [32]. Two different approaches can be used to address the safety issues. One way is to increase the intrinsic safety level of the robot through hardware design, so that even if collision happens, the impact on the human is minimized [11]. The other way is to let the robot behave safely, which is called 'interactive safety' as opposed to the intrinsic safety [33]. In this chapter, interactive safety will be addressed in the context of decision making, motion planning and control of the robot during HRI.

Conventional approach to address the interactive safety is conservative, which slows down the robot when human is nearby, hence sacrifices productivity for the sake of safety. However, to make the interaction desirable both safety and efficiency need to be considered in designing the robot behavior.

Less conservative methods in the context of obstacle avoidance in robot motion planning have also been used to address safety in HRI. Some authors use potential field methods, e.g. introducing a virtual force which pushes the robot away from the obstacles [26, 13]. Some use sliding mode methods to locally modify the reference trajectory in the presence of obstacles [10]. These two methods result in closed-form analytical control laws; but they do not emphasize optimality (or efficiency). Some authors formulate the problem as an optimization or optimal control problem with hard constraints to represent the safety requirements [7]. Unfortunately, these non-convex optimization problems are generally hard to solve analytically [12] and different approximations and numerical methods are used [29, 30].

However, as humans will respond to the robot's movement, interactions need to be considered explicitly in the design of the robot behavior, which implies that humans cannot be simply treated as obstacles. Moreover, the algorithms must be designed such that it can be executed fast enough for timely responses in emergency situations, which implies that those numerical methods may not be desirable for online applications.

These requirements are pushing for new perspectives in understanding and dealing with HRI. The robot should be designed intelligent enough to conduct social behavior [34] to interact with humans safely and efficiently even in emergency situations.

1.2 Safety Problems in HRI: Conflicts in Multi-Agent Systems

By designing the robot behavior, the safety issues need to be understood in the human robot systems. Since the intelligent robots are sharing the space with humans, the system can be viewed as a multi-agent system (MAS), where all humans and robots in the environment are regarded as agents [28]. This multi-agent situation complicates the interactions as the individual optima usually does not coincide with the system optima [23]. An agent's interest is to be efficient, i.e. finish the task as soon as possible, while staying safe. For example, an automated vehicle's interest is to go to the destination in minimum time without colliding with surrounding vehicles. The safety is the mutual interests for all agents in the system, while the efficiency goals may conflict with one another, as the individual tasks are not identical and the resources are limited. Figure 1 shows the conflict between two vehicles during lane merging. The two vehicles cannot pass the conflict zone at the same time (since the road resources are limited), which implies that one of the vehicle must yield (i.e. sacrifice efficiency) to ensure safety. Figure 2 shows the conflict between an autonomous guided vehicle (AGV) and a human worker as their future trajectories intersect. The intersection point is a potential conflict zone if neither the AGV or the human detours.

There are plenty of methods to make a robot efficient [6]. However, literature is limited on solving the conflicts among agents when there is no designated controlling agent. In the real world, conflicts of interests can be solved by active communication and negotiation [24], whereas in a human-robot co-existing environment, communications are limited. Classic game theory [3] offers reasoning strategies for cooperation to evolve in limited communications when the behavioral rules

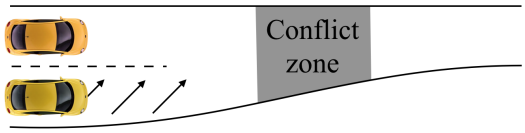


Fig. 1: Conflict between two vehicles during lane merging.

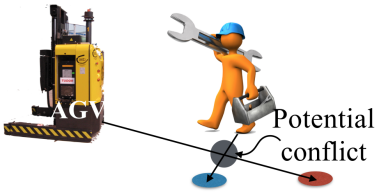


Fig. 2: Conflict between a human worker and an autonomous guided vehicle (AGV) in a factory floor.

of the agents are known. However, in the real world, agents are "strangers" to each other. This situation runs into the bayesian game regime [2]. To ensure safety, the robot should actively learn the human behavior and only seek to maximize its utility within a set of actions that is safe with respect to what human would do and how human would react to the robot motion.

1.3 Safe Control and Exploration

Take human's social behavior as an example, to avoid conflicts of interests, a human's behavior is usually constrained. The constraint comes from at least two factors: the social norm and the uncertainties that he perceives for other people. The social norm guides behavior in a certain situation or environment as 'mental representations of appropriate behavior'. For example, in the AGV case, the social norm will suggest the AGV to keep a safe distance from human and detour if necessary. In practice, the perceived uncertainties also affect human's decision. Similarly, whether the AGV needs to detour and how much the AGV should detour highly depends on how certain the AGV is about its prediction of the human's trajectory. Moreover, the uncertainties can be attenuated through active learning (refer to uncertainty reduction theory [4]). A common experience is: a newcomer tends to behave conservatively in a new environment due to large uncertainties. But through observing and learning his peers, he will gradually behave freely due to the reduction of uncertainties.

Based on these observations, a safety oriented method is introduced to design the robot controller. The HRI is modeled in the MAS framework. A safe set is defined in the system state space (which contains both the robot state and the human state) to represent the social norm. The safe set algorithm (SSA) [17] enforces invariance in the safe set according to the predicted human behavior. The safe exploration algorithm (SEA) further constrains the robot motion by uncertainties in the predictions [19]. By actively learning human behaviors, the robot will be more 'confident' about its prediction of human motion. The robot can access a larger subset of the safe set when the uncertainty is smaller.

The remainder of the chapter is organized as follows: in section 2, a multi-agent interaction model will be introduced, followed by the discussion of the safety oriented design methodology in section 3. The safe set algorithm (SSA) and the safe exploration algorithm (SEA) will be discussed in section 4 and section 5 respectively. A method to combine SSA and SEA will be discussed in section 6. Section 7 points out directions for future works. Section 8 concludes the chapter.

2 Modeling the Human Robot Interactions

2.1 The Agent Model

Suppose there are N agents in the system and are indexed from 1 to N. R denotes the set of all indices for the robots, and H for the humans. Denote agent i's state as x_i, its control input as u_i, its information set as π_i and its goal as G_i for $i = 1, ..., N$. For simplicity, write x_R as the union of the states of all robots and x_H as the union of the states of all humans. Denote the system state as $x = [x_R^T, x_H^T]^T \in X$ where X is the state space of the system. The open loop system dynamics can be written as

$$\dot{x} = f(x, u_1, u_2, \ldots, u_N, w) \tag{1}$$

where w is a Gaussian noise term.

In the case that there is no physical contact among humans and robots as shown in Fig.1 and Fig.2, the system is decomposable[1], i.e.

$$\dot{x}_i = f_i(x_i, u_i, w_i), \forall i = 1, ..., N \tag{2}$$

Moreover, it is assumed that the dynamics of all robots are affine in the control term, i.e.

$$\dot{x}_i = f_{ix}^*(x_i, w_i) + f_{iu}^*(x_i) u_i, \forall i \in R \tag{3}$$

Agent i chooses the control u_i based on the information set π_i and its goal G_i. The information set is a combination of the measured data and the communicated information. In this chapter, it is assumed that there is no direct communication among the defined agents. If a group of agents do communicate with each other, then they can be coordinated and will be treated as one agent. In this way, agent i's information set at time T contains all the measurements up to time T, i.e. $\pi_i(T) = \{y_i(t)\}_{t \in [t_0, T]}$ where

$$y_i = h_i(x, v_i), \forall i = 1, ..., N \tag{4}$$

and v_i is the measurement noise. The controller can be written as

$$u_i = g_i(\pi_i, G_i), \forall i = 1, ..., N \tag{5}$$

2.2 The Closed Loop System

Using (4) (5) in the open loop dynamics (1) or (2), the closed loop dynamic equation becomes

[1] In certain cases, the open loop system may not be decoupled. For example, in the case of rehabilitation, the robot can affect the human's dynamics directly by assisting the human to accomplish special tasks, such as walking. When the robot's input enters the human's dynamic equation, (2) does not hold.

$$\dot{x} = f^{cl}\left(x, G, v_1, ..., v_N, w_1, ...w_N\right) \tag{6}$$

where $G = \left[G_1^T, ..., G_N^T\right]^T \in X$ is the system goal. The system block diagram for the general multi-agent system is shown in Fig.3 based on (1), (4) and (5). The block diagram for the decomposable system is shown in Fig.4 based on (2), (4) and (5). Due to measurement feedbacks, all agents are coupled together in the closed loop system no matter it is decomposable or not in the open loop. The difference is that in the decomposable system, interaction happens only in the measurement and controller side instead of in the open loop dynamics. In this chapter, the decomposable system will be studied.

2.3 Information Structure

In a multi-agent system, an agent should be considerate to other agents. Before choosing the control u_i, agent i needs to consider the 'strategies' that others will play (the control signal u_j's that others would choose).

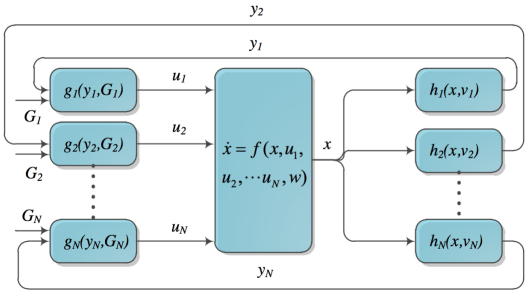


Fig. 3: Multi-Agent System Block Diagram

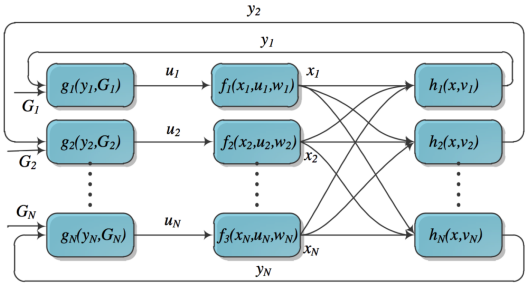


Fig. 4: Decomposable Multi-Agent System Block Diagram

Information structure [3] specifies what information is available to which agent when the agents are deciding on their control strategies. When all agents are operating on the same information set but independently before they take the next move, this is a simultaneous game and the optimal solution of this system defines a Nash Equilibrium (or a Bayesian Nash Equilibrium if the utilities of the agents are not publicly known). When some agents know the others' u_j (by observing x_j) before deciding on their own controls, this is a sequential game and the optimal solution defines a Stackelberg equilibrium (or a Perfect Bayesian Equilibrium if the utilities of the agents are not publicly known). The agents being observed are considered as leaders and the agents that take the observation are considered as followers.

It is assumed that safety-oriented HRI can be modeled as a sequential game and the robot being controlled is always a follower [17]. There are several advantages in doing so. First, it is a conservative strategy, thus good from the safety point of view. Second, it is assumed that robots have a smaller reaction time than humans. Thus they can quickly adapt to human motion, which qualifies them as followers. Moreover, to analyze a human's control strategy, the robot needs to know the human's cost function, which, however, is hard to obtain. So it is better to assume all humans' strategies are revealed in their past moves before the robot takes the move. Then the robot must be a follower, which plays a reactive strategy.

3 The Safety-Oriented Behavior Design

The robot behavior is determined by the robot controller g_i for $i \in R$. The goal in designing g_i is 1) to find the bounds on robot actions which are safe with respect to what human would do and how human would react; and 2) to maximize the robot's utility or minimize the robot's cost within the bounds.

When there are more than one robot in the system, they can be coordinated by a central controller. In this way, all robots can be treated as one agent. If it is not possible to design a coordinator as in the case of autonomous vehicles, those robots that cannot be coordinated are simply regarded as human agents. Then the number of effective robots in the system is reduced to one, denoted by R.

3.1 The Safety Principle

Denote the system's safe set as X_S, which is a closed subset of the state space X that is safe, e.g. collision free. Then the state space constraint R_S for the robot depends on humans' states, e.g. $R_S(x_H) = \{x_R : [x_R^T, x_H^T]^T \in X_S\}$. If humans are followers and will take care about the safety, then the safe set for the robot is

$$R_S^1 = \{x_R : x_R \in R_S(x_H) \text{ for some } x_H\} \tag{7}$$

However, based on the discussion in section 2.3, humans are assumed to be the leaders, so the safety problem should be taken care of by the robot. In the case that the robot knows human's next move $\hat{x}_H$, the safety bound for the robot becomes

$$R_S^2 = \{x_R : x_R \in R_S(\hat{x}_H)\} \tag{8}$$

However, due to noises and uncertainties, the estimate $\hat{x}_H$ may not be accurate. The human state x_H may lie in a set Γ_H containing $\hat{x}_H$. Then the robot motion should be constrained in a smaller set

$$R_S^3 = \{x_R : x_R \in R_S(x_H), \forall x_H \in \Gamma_H\} \tag{9}$$

Figure 5 illustrates the safe set X_S and the state space constraints R_S^1, R_S^2 and R_S^3. It is clear that $R_S^3 \subset R_S^2 \subset R_S^1$.

The Safety Principle: the function $g_i(.,.), i \in R$ should be chosen such that X_S is invariant, i.e. $x(t) \in X_S$ for all t, or equivalently, $x_R(t) \in R_S^3(t)$ for $\Gamma_H(t)$ which accounts for almost all possible noises $v_1, ...v_N, w_1, ..., w_N$ and human decisions $g_i(.,.), i \in H$ (those with negligible probabilities will be ignored).

Figure 6 illustrates the expected outcome of the robot behavior under the safety principle. In view of the potential conflict, the robot re-plans a trajectory in the safe region R_S^3.

3.2 The Safety Index

The safety principle requires the designed control law to make the safe set invariant with respect to time. In addition to constraining the motion in the safe region R_S^3, the robot should also be able to cope with any unsafe human movement. Given the current configuration in Fig.5, if the human is anticipated to move downwards, the robot should go left in order for the combined trajectory to stay in the safe set.

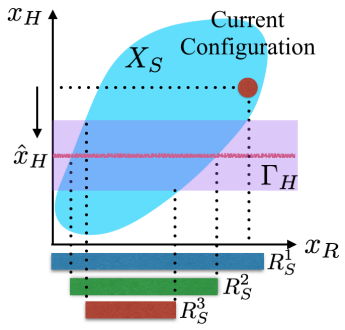


Fig. 5: The State Space Safety Constraints X_S, R_S^1, R_S^2 and R_S^3

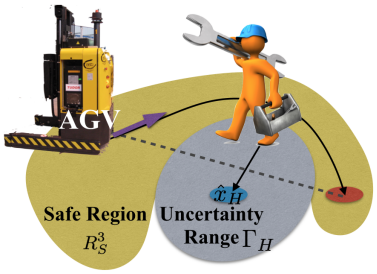


Fig. 6: Solving the conflict by re-planning in the safe region R_S^3.

To make the robot better understand the dynamics, a safety index is introduced as shown in Fig.7. The safety index $\phi : X \to \mathbb{R}$ is a function on the system state space such that

1. ϕ is differentiable with respect to t, i.e. $\dot{\phi} = (\partial\phi/\partial x)\dot{x}$ exists everywhere;
2. $\partial\dot{\phi}/\partial u_R \neq 0$;
3. The unsafe set $X \setminus X_S$ is not reachable given the control law $\dot{\phi} < 0$ *when* $\phi \geq 0$ and the initial condition $x(t_0) \in X_S$.

The first condition is to ensure that ϕ is smooth. The second condition is to ensure that the robot input can always affect the safety index. The third condition presents a quick criteria to determine whether a control input is safe or not, e.g. all the control inputs that drive the state below the level set 0 are safe and unsafe otherwise.

Lemma: (Existence of the Safety Index) The function ϕ satisfying all three conditions exists for any $X_S = \{x : \phi_0(x) \leq 0\}$, where $\phi_0(x)$ is a smooth function on the state space.[2]

To ensure safety, the robot's control must be chosen from the set of safe control $U_S(t) = \{u_R(t) : \dot{\phi} \leq -\eta_R \; when \; \phi \geq 0\}$ where $\eta_R \in \mathbb{R}^+$ is a safety margin. By (3), the derivative of the safety index can be written as

$$\dot{\phi} = \frac{\partial\phi}{\partial x_R} f_{Ru}^* u_R + \frac{\partial\phi}{\partial x_R} f_{Rx}^* + \sum_{j\in H} \frac{\partial\phi}{\partial x_j}\dot{x}_j \tag{10}$$

Then the set of safe control is

$$U_S(t) = \{u_R(t) : L(t)\,u_R(t) \leq S(t, \dot{x}_H)\} \tag{11}$$

where

[2] The Lemma is proved in [17]. ϕ can be constructed in the following procedure: first, check the order from ϕ_0 to u_R in the Lie derivative sense, denote it by n; then define ϕ as $\phi_0 + k_1\dot{\phi}_0 + \ldots + k_{n-1}\phi_0^{(n-1)}$. The coefficients $k_1, \ldots, k_n$ are chosen such that the roots of $1 + k_1 s + \ldots + k_{n-1}s^{n-1} = 0$ all lie on the negative real line.

$$L(t) = \frac{\partial \phi}{\partial x_R} f_{Ru}^* \tag{12}$$

$$S(t, \dot{x}_H) = \begin{cases} -\eta_R - \sum_{j \in H} \frac{\partial \phi}{\partial x_j} \dot{x}_j - \frac{\partial \phi}{\partial x_R} f_{Rx}^* & \phi \geq 0 \\ \infty & \phi < 0 \end{cases} \tag{13}$$

$L(t)$ is a vector at the 'safe' direction, while $S(t, \dot{x}_H)$ is a scalar indicating how much control effort is needed to be safe, which can be broken down into three parts: the margin, the term to compensate human motion and the term to compensate the inertia of the robot itself. In the following arguments when there is no ambiguity, $S(t, \dot{x}_H)$ denotes the value in the case $\phi \geq 0$ only. As could be expected, under different assumptions of the human behavior, $S(t)$ varies. The sets of safe control correspond to R_S^1, R_S^2 and R_S^3 are

$$U_S^1(t) = \{u_R(t) : L(t) u_R(t) \leq S(t, \dot{x}_H) \text{ for some } \dot{x}_H\} \tag{14}$$

$$U_S^2(t) = \left\{u_R(t) : L(t) u_R(t) \leq S\left(t, \hat{\dot{x}}_H\right)\right\} \tag{15}$$

$$U_S^3(t) = \left\{u_R(t) : L(t) u_R(t) \leq S(t, \dot{x}_H) \text{ for all } \dot{x}_H \in \dot{\Gamma}_H\right\} \tag{16}$$

where $\hat{\dot{x}}_H$ is the velocity vector that moves the current configuration x_H of human to $\hat{x}_H$ and $\dot{\Gamma}_H$ is the set of velocity vectors that move x_H to Γ_H. Computationally, $\hat{\dot{x}}_H = \frac{\hat{x}_H - x_H}{T_s}$ where T_s is the sampling time. Obviously $U_S^3 \subset U_S^2 \subset U_S^1$. When the uncertainties in the estimation of $\hat{\dot{x}}_H$ reduces, U_S^3 converges to U_S^2.

The difference between R_S and U_S is that R_S is static as it is on the state space, while U_S is dynamic as it concerns with the 'movements'. Due to the safety index, the non convex state space constraints R_S are transformed to convex state space constraints U_S. For example, in Fig.6, the safe region R_S^3 for the AGV is the space outside the uncertainty range. But the set U_S^3 according to (16) is a half space.

According to the safety principle, the robot control should be restricted to the set U_S^3. However, the choice of the uncertainty bound $\dot{\Gamma}_H$ differs. One way is to use a constant bound based on the mean prediction error. Another way is to let the uncertainty bound depend on the level of uncertainties in real time. The first way

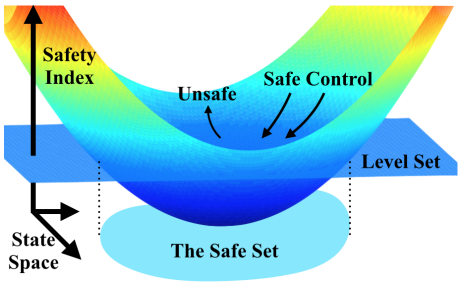


Fig. 7: The Safety Index

corresponds to the safe set algorithm (SSA) [17] while the second one corresponds to the safe exploration algorithm (SEA) [19], which will be discussed in the following two sections.

4 The Safe Set Algorithm (SSA)

The safe set algorithm offers a fast online solution concerning the safety principle. In this section, the control algorithm and the learning algorithm in SSA will be discussed, followed by an application on a robot arm.

4.1 The Control Algorithm

In SSA, a constant $\lambda_R^{SSA} \in \mathbb{R}^+$ is introduced to bound the noises and uncertainties in the estimation, i.e.

$$S^{SSA}\left(t,\hat{x}_H\right) = S(t,\hat{x}_H) - \lambda_R^{SSA} = -\eta_R - \lambda_R^{SSA} - \sum_{j\in H}\frac{\partial\phi}{\partial x_j}\hat{x}_j - \frac{\partial\phi}{\partial x_R}f_{Rx}^* \tag{17}$$

where $\hat{x}_j = \frac{\hat{x}_j(k+1|k)-\hat{x}_j(k|k)}{T_s}$ for all $j \in H$. k represents the time step of the last measurement before t, while $k+1$ is the time step for the next anticipated measurement. T_s is the sampling time. $\hat{x}_j(p|k)$ is the estimate of $x_j(p)$ given information up to time step k. $L(t)$ and $S(t)$ will also be written as $L(k)$ and $S(k)$ to denote that the last measurement is taken at k-th time step. The control algorithm can also be designed in discrete time as discussed in [17] which is more conservative.

The optimal control problem for robot motion planning can be posed as [18]:

$$\min \;\; E(J(x_R,u_R,G_R)) \tag{18}$$

$$s.t. \;\; u_R \in U_S^{SSA}, u_R \in \Omega, x_R \in \Gamma_R \tag{19}$$

where $U_S^{SSA}(t) = \{u_R : L(t)u_R \leq S^{SSA}(t,\hat{x}_H)\}$. J is the cost function that evaluates the efficiency of the robot motion. Ω is the control saturation and Γ_R is the state space constraint such as joint limits and speed limits. Ω and Γ_R is usually convex, while J can be designed to be convex. Then (18-19) is a convex optimization problem, which is easy to solve online.

An analytic control law that solves the optimization approximately can be obtained by breaking down the problem (18-19). Assume the state space constraint Γ_R is considered in the safe set X_S. Suppose $u_R^o(t)$ is the solution of the optimal control problem without the safety constraint $u_R \in U_S^{SSA}$ (which can be solved offline). Then the safe control law $u_R^*(t)$ can be found by mapping $u_R^o(t)$ to the set of safe control $U_S^{SSA}(t)$ according to the following cost function

$$u_R^* = \min_{u_R \in U_S^{SSA} \cap \Omega} \frac{1}{2} (u_R - u_R^o)^T Q (u_R - u_R^o) \tag{20}$$

where Q is a positive definite matrix. Let $c = \min_{u \in U_S^{SSA}(t)} |L(t)(u - u_R^o(t))|$. By (20), the safe control law $u_R^*(t)$ when Ω is not tight is [17]:

$$u_R^*(t) = u_R^o(t) - c \frac{Q^{-1} L(t)^T}{L(t) Q^{-1} L(t)^T} \tag{21}$$

4.2 Online Learning and Prediction of Humans' Dynamics

Since only one effective robot is considered, the system in Fig.4 can be reduced to the system in Fig.8 by combining all the blocks for human agents to the closed loop block $\dot{x}_H = f_H^{'}(x_H, x_R, G_H, v_H, w_H)$. The objective is to find the estimate $\hat{x}_H(k+i|k)$ to minimize the expected prediction error, e.g.

$$\hat{x}_H(k+i|k) = \arg\min_a E_{x_H(k+i)}(\|x_H(k+i) - a\|^2) \tag{22}$$

The human's closed loop dynamics $f_H^{'}$ need to be estimated for state prediction. The nonlinear continuous time dynamic function $\dot{x}_H = f_H^{'}(.)$ can be linearized and discretized as

$$x_H(k+1) = A_H(k) x_H(k) + B_H(k) u_H^c(k) + w_H^*(k) \tag{23}$$

where $u_H^c(k) = \left[\hat{x}_R(k|k)^T, G_H^T(k)\right]^T$[3]. $\hat{x}_R$ is the estimate of the robot state from a state estimator (e.g. Kalman Filter). $A_H(k)$ and $B_H(k)$ are time varying parameters. $w_H^*(k)$ is a noise term assumed to be zero-mean Gaussian and white. Assume the robot's measurement of the human is:

$$y_R^H(k) = x_H(k) + v_R^H(k) \tag{24}$$

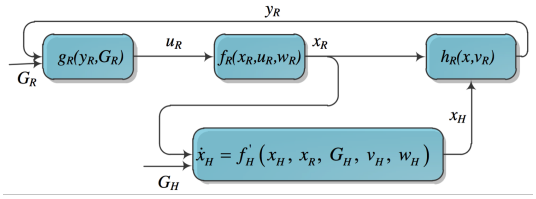


Fig. 8: Simplified System Model

[3] Methods for inferring $G_H(k)$ are discussed in [18]. In this chapter, it is assumed to be known.

Equations (23) and (24) form a linear time varying (LTV) Gaussian system with unknown parameters. A recursive least square parameter adaptation algorithm (RLS-PAA) [9] is developed to identify the system online such that the prediction $\hat{x}_H(k+1|k)$ minimizes the expected prediction error (22). Define $\hat{A}_H(k)$ and $\hat{B}_H(k)$ to be the estimates of the matrices given the information up to the k-th time step.

- State Estimation
  At $k+1$-th time step, $\hat{x}_H$ is first updated according to the closed loop dynamics in (25). Then the measurement information is incorporated in the *a posteriori* estimate in (26). A constant update gain $\alpha \in (0,1)$ is chosen to ensure that the measurement information is always incorporated.

$$\hat{x}_H(k+1|k) = \hat{A}_H(k)\hat{x}_H(k|k) + \hat{B}_H(k)u_H^c(k) \tag{25}$$
$$\hat{x}_H(k+1|k+1) = (1-\alpha)\hat{x}_H(k+1|k) + \alpha y_R^H(k+1) \tag{26}$$

- Parameter Estimation
  The closed loop matrices are estimated using RLS-PAA:

$$\begin{aligned}&\left[\hat{A}_H(k+1), \hat{B}_H(k+1)\right] = \left[\hat{A}_H(k), \hat{B}_H(k)\right] \\ &+ \left(\hat{x}_H(k+1|k+1) - \hat{x}_H(k+1|k)\right)\varphi(k)^T F(k+1)\end{aligned} \tag{27}$$

  where $\varphi(k) = \left[\hat{x}_H(k|k)^T \; u_H^c(k)^T\right]^T$. F is the learning gain such that

$$F(k+1) = \frac{1}{\lambda}\left[F(k) - \frac{F(k)\varphi(k)\varphi^T(k)F(k)}{\lambda + \varphi^T(k)F(k)\varphi(k)}\right] \tag{28}$$

  where $\lambda \in (0,1)$ is a forgetting factor.

4.3 Applications

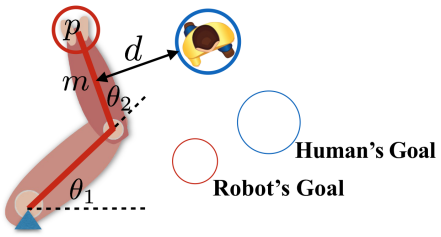


Fig. 9: The Interaction between a Robot Arm and a Human

In the factories of the future, there will be close interactions among human workers and robot arms [20]. In this section, the SSA is applied to a planar robot arm to demonstrate the effectiveness of the algorithm.

The environment is shown in Fig.9 where both the robot arm and the human need to approach their respective goals in minimum time without colliding with each other. The robot arm has two links with joint positions θ_1 and θ_2 and the joint velocities $\dot{\theta}_1$ and $\dot{\theta}_2$. Let $\theta = [\theta_1, \theta_2]^T$. Considering the kinematics of the robot arm, the control input is defined to be the joint accelerations, i.e. $u_R = \left[\ddot{\theta}_1, \ddot{\theta}_2\right]^T$. The end point position of the robot is denoted as $p = (p_x, p_y)$. The optimal control law is designed to be

$$u_R^o = J_p^{-1}\left\{K_p\begin{bmatrix} p_x - g_x \\ p_y - g_y \end{bmatrix} + K_v\begin{bmatrix} \dot{p_x} \\ \dot{p_y} \end{bmatrix} - H_p\right\} \tag{29}$$

where J_p is the Jacobian matrix at p and $H_p = \dot{J_p}\dot{\theta}$. (g_x, g_y) is the goal point in the work space. $K_p \in \mathbb{R}^{2\times 2}$ and $K_v \in \mathbb{R}^{2\times 2}$ are the control gain.

The closest point to the human is $m = (m_x, m_y)$. Define the robot state as $x_R = [m_x, m_y, \dot{m}_x, \dot{m}_y]^T$. The state space equation is

$$\dot{x}_R = A_R x_R + B_R J_m u_R + B_R H_m \tag{30}$$

where

$$A_R = \begin{bmatrix} 0 & I_2 \\ 0 & 0 \end{bmatrix}, B_R = \begin{bmatrix} 0 \\ I_2 \end{bmatrix}$$

and J_m is the Jacobian matrix at m with $H_m = \dot{J_m}\dot{\theta}$.

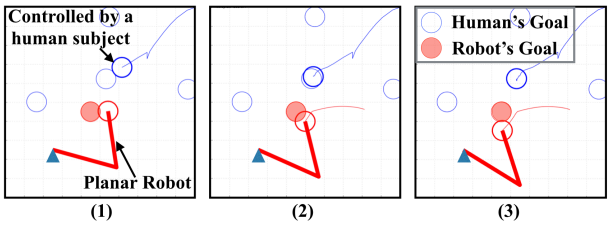
The human is simplified as a circle, whose state is taken as $x_H = [h_x, h_y, \dot{h}_x, \dot{h}_y]$ where h_x and h_y is the position and $\dot{h}_x$ and $\dot{h}_y$ is the velocity. Define the safe set as $X_S = \{x : d \geq d_{min}\}$ where d measures the smallest distance between the human and the robot arm and d_{min} is a constant. Based on the discussion in section 3.2, the safety index is designed as $\phi = D - \dot{d}^2 - k_\phi \dot{d}$ where $D = d_{min}^2 + \eta_R T_s + \lambda_R^{SSA} T_s$ and $k_\phi > 0$ are constants [17]. Let the relative distance, velocity and acceleration vectors be $\mathbf{d} = \left[I_2\ 0\right](x_R - x_H)$,$\mathbf{v} = \left[0\ I_2\right](x_R - x_H)$ and $\mathbf{a} = \left[0\ I_2\right](\dot{x}_R - \dot{x}_H)$. Then $d = |\mathbf{d}|$ and

$$\begin{aligned} \dot{\phi} &= -2d\dot{d} - k_\phi \ddot{d} = -2\mathbf{d}^T\mathbf{v} - k_\phi \frac{\mathbf{d}^T\mathbf{a} + \mathbf{v}^T\mathbf{v} - \dot{d}^2}{d} \\ &= -2\mathbf{d}^T\mathbf{v} - k_\phi \frac{\mathbf{d}^T(J_m u_R + H_m) - \mathbf{d}^T\left[0\ I_2\right]\dot{x}_H + \mathbf{v}^T\mathbf{v}}{d} + k_\phi \frac{(\mathbf{d}^T\mathbf{v})^2}{d^3} \end{aligned} \tag{31}$$

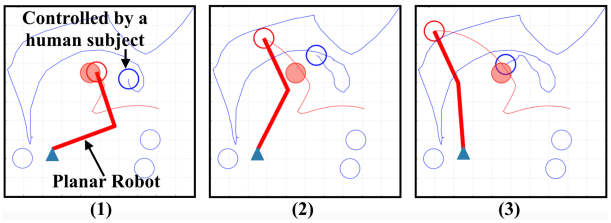
Hence

$$L(t) = -k_\phi \frac{\mathbf{d}^T}{d} J_m \tag{32}$$

$$S(t, \dot{x}_H) = -\eta_R + 2\mathbf{d}^T\mathbf{v} + k_\phi \frac{\mathbf{d}^T H_m - \mathbf{d}^T\left[0\ I_2\right]\dot{x}_H + \mathbf{v}\cdot\mathbf{v}}{d} - k_\phi \frac{(\mathbf{d}^T\mathbf{v})^2}{d^3} \tag{33}$$



(a) Scenario 1



(b) Scenario 2

Fig. 10: Two scenarios in the simulation of SSA on a planar robot arm. The objective: the human and the robot arm should approach their respective goals in minimum time. Safety requirement: the minimum distance between the human and the robot arm should be greater than a threshold.

In the simulation, several goals were assigned for the human. Before parameter adaptation, the robot inferred on the human's current goal first [18]. Sampling time $T_s = 0.1s$. Figure 10a shows the human avoidance behavior of the robot. In (1), the human and the robot were both near their respective goals. However, since the human was heading towards the robot in high speed, the robot went backward in (2) and (3). Figure 10b shows the robot behavior under unexpected human behavior. In (1), the human suddenly changed his course. Although all of his goal points were in the lower part of the graph, the human started to go up. By observing that, the robot went away from the human in (2) and (3). The simulation results confirms the effectiveness of the algorithm.

5 The Safe Exploration Algorithm (SEA)

One of the limitation of SSA is that the bound for the uncertainties (i.e. λ_R^{SSA}) is a constant. However, the mean squared estimation error (MSEE) of the human's state is changing from time to time. A larger bound is needed if the MSEE is larger. To capture this property, the safe exploration algorithm (SEA) is introduced, where the control strategy changes for different levels of uncertainties.

Table 1: Definitions of Notations in State Estimation

	State Estimate	Estimation Error	MSEE
a posteriori	$\hat{x}_j(k\|k)$	$\tilde{x}_j(k\|k)$	$X_j(k\|k)$
a priori	$\hat{x}_j(k+1\|k)$	$\tilde{x}_j(k+1\|k)$	$X_j(k+1\|k)$

5.1 The Safe Set in the Belief Space

In a belief space [7], the state estimate of x_j for $j \in H$ is no longer a point but a distribution, i.e. $\mathcal{N}(\hat{x}_j, X_j)$ where X_j is the covariance which represents the level of uncertainties in the estimation. All the distributions are assumed to be Gaussian. Since $x_j \sim \mathcal{N}(\hat{x}_j, X_j)$, the covariance can be written as $X_j = E\left[(x_j - \hat{x}_j)(x_j - \hat{x}_j)^T\right]$, which is the mean squared estimation error (MSEE).

The definition of the *a priori* and *a posteriori* estimates, estimation errors and MSEEs are shown in Table 1, where $\tilde{x}_j(k|k) = x_j(k) - \hat{x}_j(k|k)$ and $\tilde{x}_j(k+1|k) = x_j(k+1) - \hat{x}_j(k+1|k)$. At the k-th time step, the best prediction for $x_j(k+1)$ has the following distribution

$$\mathcal{N}(\hat{x}_j(k+1|k), X_j(k+1|k)) \tag{34}$$

In the belief space, since the distribution of $x_j(k+1)$ is unbounded, the inequality in (16) is ill-defined. Indeed, u_R needs to satisfy a probability constraint

$$P\left(\{x_j(k+1) : L(k)u_R \le S(k, x_H)\}\right) \ge 1-\varepsilon, \forall j \in H \tag{35}$$

where $\varepsilon > 0$ is a small number. A bounded set $\Gamma_j(k)$ can be defined for $j \in H$ such that the probability density of $x_j \notin \Gamma_j(k)$ is small and $P(x_j \in \Gamma_j(k)) \ge 1-\varepsilon$. For a Gaussian distribution, the probability mass lying within the 3σ deviation is 0.997. Set $\varepsilon = 0.003$ and let $\Delta x_j = x_j - \hat{x}_j(k+1|k)$, then the set Γ_j can be defined as

$$\Gamma_j(k) = \left\{x_j : \Delta x_j^T X_j(k+1|k)^{-1} \Delta x_j \le 9\right\} \tag{36}$$

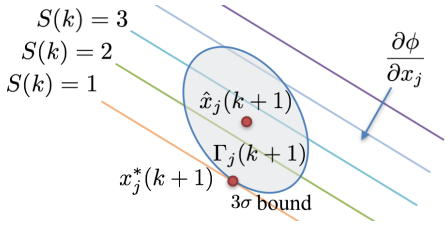


Fig. 11: The Constraint in the Belief Space

By (16) and (35), the constraint in U_R^3 is equivalent to the following optimization problem,

$$L(k)u_R \leq S^{SEA}(k) = \min_{x_j(k+1)\in\Gamma_j(k),\forall j\in H}\{S(k)\} \tag{37}$$

By (13), the RHS of (37) can be decoupled as a sequence of optimization problems, i.e. for all $j \in H$,

$$\min_{x_j(k+1)\in\Gamma_j(k)} \frac{\partial\phi}{\partial x_j}x_j(k+1) \tag{38}$$

By Lagrangian method[4], the optimal solution $x_j^*(k+1)$ for all $j \in H$ is

$$x_j^*(k+1) = \hat{x}_j(k+1|k) + \frac{3X_j(k+1|k)\left(\frac{\partial\phi}{\partial x_j}\right)^T}{\left[\left(\frac{\partial\phi}{\partial x_j}\right)X_j(k+1|k)\left(\frac{\partial\phi}{\partial x_j}\right)^T\right]^{\frac{1}{2}}} \tag{40}$$

Using (40), $S^{SEA}(k)$ can be expressed as

$$S^{SEA}(k) = S(k,\hat{x}_H) - \lambda_R^{SEA}(k) = -\eta_R - \lambda_R^{SEA}(k) - \sum_{j\in H}\frac{\partial\phi}{\partial x_j}\hat{x}_j - \frac{\partial\phi}{\partial x_R}f_{Rx}^* \tag{41}$$

where

$$\lambda_R^{SEA}(k) = \frac{3}{T_s}\sum_{j\in H}\left[\left(\frac{\partial\phi}{\partial x_j}\right)X_j(k+1|k)\left(\frac{\partial\phi}{\partial x_j}\right)^T\right]^{\frac{1}{2}} + \lambda_R^o \tag{42}$$

and $\lambda_R^o \in \mathbb{R}^+$ is the bound for other uncertainties. All other equations follow from the safe set algorithm except for the learning and prediction part, where new methods are needed to estimate $X_j(k+1|k)$ online.

5.2 Learning in the Belief Space

In this section, the MSEE propagation algorithm in parameter adaptation will be discussed, followed by the discussion of its application to human motion prediction. To simplify the notation, the LTV system is assumed to be

[4] The objective function is linear while the constraint function defines an ellipsoid as shown in Fig.11. The optimal solution must lie on the boundary of the ellipsoid. Let γ be a Lagrange multiplier. Define the new cost function as:

$$J_j^* = \frac{\partial\phi}{\partial x_j}x_j(k+1) + \gamma\left[9 - \Delta x_j^T X_j(k+1|k)^{-1}\Delta x_j\right] \tag{39}$$

The optimal solution satisfies $\frac{\partial J_j^*}{\partial x_j(k+1)} = \frac{\partial J_j^*}{\partial\gamma} = 0$, i.e. $(\frac{\partial\phi}{\partial x_j})^T - 2\gamma X_j(k+1|k)^{-1}\Delta x_j = 0$ and $9 - \Delta x_j^T X_j(k+1|k)^{-1}\Delta x_j = 0$. Then (40) follows.

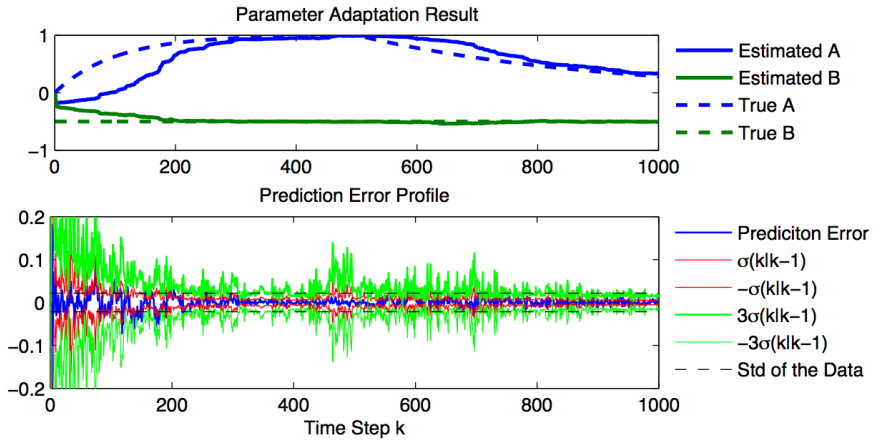


Fig. 12: Simulation Result of the MSEE Propagation Algorithm

$$x(k+1) = A(k)x(k) + B(k)u(k) + w(k) \tag{43}$$

where $x(k) \in \mathbb{R}^n$, $u(k) \in \mathbb{R}^m$ and $w(k)$ is the dynamic noise assumed to be zero-mean, Gaussian and white with covariance W. It is assumed that the state $x(k)$ is fully known. $A(k)$ and $B(k)$ are the unknown parameters that need to be estimated online. Define the parameter matrix $C(k) = [A(k), B(k)] \in \mathbb{R}^{n\times(n+m)}$ and the data vector $\varphi(k) = \left[x^T(k), u^T(k)\right]^T \in \mathbb{R}^{n+m}$. Since random matrices are hard to deal with, transform matrix $C(k)$ to a vector $\vartheta(k)$. Suppose the row vectors in $C(k)$ are $C_1(k), C_2(k), \ldots, C_n(k) \in \mathbb{R}^{1\times(n+m)}$. Define

$$\vartheta(k) = [C_1(k), C_2(k), ..., C_d(k)]^T \in \mathbb{R}^{n(n+m)\times 1} \tag{44}$$

Define a new data matrix $\Phi(k)$ as

$$\Phi(k) = \begin{bmatrix} \varphi^T(k) & 0 & \cdots & 0 \\ 0 & \varphi^T(k) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \varphi^T(k) \end{bmatrix} \in \mathbb{R}^{n\times n(n+m)} \tag{45}$$

Using $\Phi(k), \vartheta(k)$, the system dynamics can be written as

$$x(k+1) = \Phi(k)\vartheta(k) + w(k) \tag{46}$$

Let $\hat{\vartheta}(k)$ be the estimate of $\vartheta(k)$ and $\tilde{\vartheta}(k) = \vartheta(k) - \hat{\vartheta}(k)$ be the estimation error.

- State estimation

The *a priori* estimate of the state and the estimation error is

$$\hat{x}(k+1|k) = \Phi(k)\,\hat{\vartheta}(k) \tag{47}$$

$$\tilde{x}(k+1|k) = \Phi(k)\,\tilde{\vartheta}(k) + w(k) \tag{48}$$

Since $\hat{\vartheta}(k)$ only contains information up to the $(k-1)$-th time step, $\tilde{\vartheta}(k)$ is independent of $w(k)$. Thus the *a priori* MSEE is

$$X_{\tilde{x}\tilde{x}}(k+1|k) = E\left[\tilde{x}(k+1|k)\tilde{x}(k+1|k)^T\right] = \Phi(k)\,X_{\tilde{\vartheta}\tilde{\vartheta}}(k)\,\Phi^T(k) + W \tag{49}$$

where $X_{\tilde{\vartheta}\tilde{\vartheta}}(k) = E\left[\tilde{\vartheta}(k)\,\tilde{\vartheta}(k)^T\right]$ is the mean squared error of the parameter estimation.

- Parameter estimation
  In the standard PAA, the parameter is estimated as

$$\hat{\vartheta}(k+1) = \hat{\vartheta}(k) + F(k+1)\,\Phi^T(k)\,\tilde{x}(k+1|k) \tag{50}$$

where $F(k+1)$ is the learning gain in (28) with $\varphi(k)$ replaced by $\Phi^T(k)$. The parameter estimation error is

$$\tilde{\vartheta}(k+1) = \tilde{\vartheta}(k) - F(k+1)\,\Phi^T(k)\,\tilde{x}(k+1|k) + \Delta\vartheta(k) \tag{51}$$

where $\Delta\vartheta(k) = \vartheta(k+1) - \vartheta(k)$. Since the system is time varying, the estimated parameter is biased and the expectation of the error can be expressed as

$$\begin{aligned} & E\left(\tilde{\vartheta}(k+1)\right) \\ &= \left[I - F(k+1)\,\Phi^T(k)\,\Phi(k)\right] E\left(\tilde{\vartheta}(k)\right) + \Delta\vartheta(k) \\ &= \sum_{n=0}^{k} \prod_{i=n+1}^{k} \left[I - F(i+1)\,\Phi^T(i)\,\Phi(i)\right] \Delta\vartheta(n) \end{aligned} \tag{52}$$

The mean squared error of parameter estimation follows from (51) and (52):

$$\begin{aligned} & X_{\tilde{\vartheta}\tilde{\vartheta}}(k+1) \\ &= F(k+1)\,\Phi^T(k)\,X_{\tilde{x}\tilde{x}}(k+1|k)\,\Phi(k)\,F(k+1) \\ & -X_{\tilde{\vartheta}\tilde{\vartheta}}(k)\,\Phi^T(k)\,\Phi(k)\,F(k+1) - F(k+1)\,\Phi^T(k)\,\Phi(k)\,X_{\tilde{\vartheta}\tilde{\vartheta}}(k) \\ & +E\left[\tilde{\vartheta}(k+1)\right]\Delta\vartheta^T(k) + \Delta\vartheta(k)\,E\left[\tilde{\vartheta}(k+1)\right]^T - \Delta\vartheta(k)\,\Delta\vartheta(k)^T + X_{\tilde{\vartheta}\tilde{\vartheta}}(k) \end{aligned} \tag{53}$$

Since $\Delta\vartheta(k)$ is unknown in (52) and (53), it is set to an average time varying rate $d\vartheta$ in the implementation.

Fig.12 shows the simulation result of the proposed learning algorithm on a first order system with a noise covariance $W = 0.005^2$. A forgetting factor $\lambda = 0.98$ is used. The solid and dashed blue lines in the upper figure are $\hat{A}(k)$ and $A(k)$, while the solid and dashed green lines are $\hat{B}(k)$ and the constant parameter B respec-

tively. As shown in the figure, the time varying parameter $A(k)$ is well approximated by $\hat{A}(k)$, while $\hat{B}(k)$ converges to B. In the lower figure, the blue curve is the one step prediction error $\tilde{x}(k|k-1)$. The green curves are the 3σ bound ($\sigma = \sqrt{X_{\tilde{x}\tilde{x}}(k|k-1)}$). The black dashed line is the statistical standard deviation (Std) of the data $\tilde{x}(k|k-1)$ from $k=1$ to $k=1000$. As shown in the figure, the 3σ value offers a good bound for the prediction errors as all measured errors lie between the green curves. Moreover, the MSEE is larger when the parameter is changing faster, which captures the time varying property of the system. On the other hand, the statistical standard deviation does not give a good description of the data in real time.

In section 4.2, a RLS-PAA algorithm is adopted in identifying the closed loop behavior of the humans. In the safe exploration algorithm, the MSEE of $\hat{x}_j(k+1|k)$ for all $j \in H$ also needs to be estimated. The system follows from (23) and (24), which is different from (43) in that the state is not exactly known. But it is assumed that the measurement noise is small, thus can be neglected. So the system is approximated by

$$y_H^R(k+1) = A_H(k) y_H^R(k) + B_H(k) u_H^c(k) + w_H^*(k) \tag{54}$$

This is equivalent to setting $\alpha = 1$ in (26). The prediction algorithm then follows from (47-53). In the implementation, the covariance of the noise W, the time varying rate $d\vartheta$ and the initial values are hand-tuned.

5.3 A comparative Study between SSA and SEA

In this section, a comparative study between SSA and SEA is performed on an autonomous vehicle model shown in Fig.13. The vehicle's state is denoted by $x_R = [R_x, R_y, v_R, \theta_R]^T$ where R_x is the x-position of the vehicle, R_y the y-position, v_R the speed and θ_R the direction. The control input of the vehicle is $u_R = [\dot{v}_R, \dot{\theta}_R]^T$ (saturations apply: $|\dot{v}_R| \le a_{max}$ and $|\dot{\theta}_R| \le \omega_{max}$, where a_{max}, ω_{max} are positive constants). The state equation is

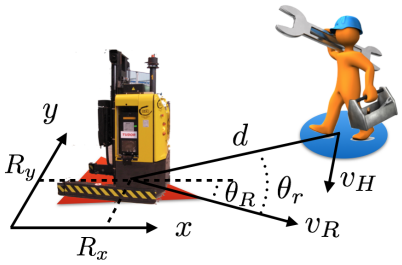


Fig. 13: The Interaction between an AGV and a Human Worker

$$\dot{x}_R = f^*_{Rx}(x_R) + Bu_R \tag{55}$$

where $f^*_{Rx}(x_R) = \begin{bmatrix} v_R\cos\theta_R & v_R\sin\theta_R & 0 & 0 \end{bmatrix}^T$, $B = \begin{bmatrix} 0 & I_2 \end{bmatrix}^T$.

The vehicle can measure its own state directly. It can also measure the relative distance d and the relative direction θ_r towards the nearby human as illustrated in Fig.13. The human's state is $x_H = [h_x, h_y, \dot{h}_x, \dot{h}_y]^T$, which is calculated based on the measurements and the robot state. Suppose that the goal point of the robot is $[G_x, G_y]$. The baseline control law is designed as [19]:

$$\dot{v}_R = -\left[(R_x - G_x)\cos\theta_R + (R_y - G_y)\sin\theta_R\right] - k_v v_R \tag{56}$$

$$\dot{\theta}_R = k_\theta \left[\arctan\frac{R_y - G_y}{R_x - G_x} - \theta_R\right] \tag{57}$$

where $k_v, k_\theta \in \mathbb{R}^+$ are constants.

The safety index $\phi = D - d^2 - k_\phi \dot{d}$ is chosen as the same as in section 4.3. In SSA, D is set to be $d^2_{min} + \eta_R T_s + \lambda^{SSA}_R T_s$. In SEA, D is set to be $d^2_{min} + \eta_R T_s + \lambda^{SEA}_R(k) T_s$. The relative distance, velocity and acceleration vectors are

$$\begin{aligned}
\mathbf{d} &= [d\cos(\theta_r + \theta_R), d\sin(\theta_r + \theta_R)]^T \\
\mathbf{v} &= [v\cos(\theta_R) - \dot{h}_x, v\sin(\theta_R) - \dot{h}_y]^T \\
\mathbf{a} &= \begin{bmatrix} \cos\theta_R & -v_R\sin\theta_R \\ \sin\theta_R & v_R\cos\theta_R \end{bmatrix} u_R - \begin{bmatrix} 0 & I_2 \end{bmatrix} \dot{x}_H
\end{aligned}$$

Similar to (31), the time derivative of the safety index is

$$\begin{aligned}
\dot{\phi} &= -2\mathbf{d}^T\mathbf{v} - k_\phi \frac{\mathbf{d}^T\mathbf{a} + \mathbf{v}^T\mathbf{v} - \dot{d}^2}{d} \\
&= -2\mathbf{d}^T\mathbf{v} - k_\phi \frac{[d\cos\theta_r, -dv_R\sin\theta_r]\, u_R - \mathbf{d}^T \begin{bmatrix} 0 & I_2 \end{bmatrix} \dot{x}_H + \mathbf{v}^T\mathbf{v}}{d} + k_\phi \frac{(\mathbf{d}^T\mathbf{v})^2}{d^3}
\end{aligned} \tag{58}$$

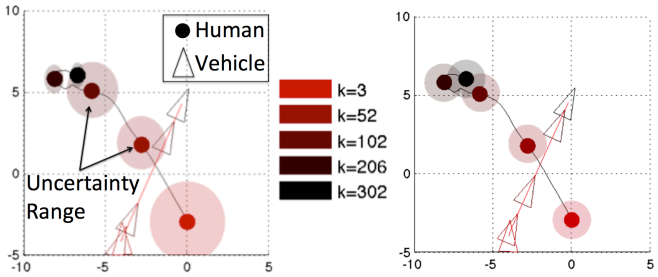
which implies

$$L(t) = k_\phi \left[\cos\theta_r, -v_R\sin\theta_r\right] \tag{59}$$

$$S(t, \dot{x}_H) = -\eta_R + 2\mathbf{d}^T\mathbf{v} + k_\phi \frac{\mathbf{v}^T\mathbf{v} - \mathbf{d}^T \begin{bmatrix} 0 & I_2 \end{bmatrix} \dot{x}_H}{d} - k_\phi \frac{(\mathbf{d}^T\mathbf{v})^2}{d^3} \tag{60}$$

Then S^{SSA} and S^{SEA} follow from (17) and (41) respectively, and the final control follows from (21).

Figure 14 shows the vehicle trajectories under SSA and SEA. The vehicle needed to approach (0,5) from (-5,-5) while the human went from (0,-3) to (-5,5). Five time steps are shown in the plots: $k = 3, 52, 102, 206, 302$ from the lightest to the darkest. The solid circles represent the human, which was controlled by a human subject through a multi-touch trackpad in real time (notice there was overshoot as the control was not perfect). The triangles represent the vehicle. The transparent circles in Fig.14a represent the set $\Gamma_H(k)$ in (36) mapped into 2D, which is shrinking grad-



(a) Simulation with SEA (b) Simulation with SSA

Fig. 14: Simulation of Human Vehicle Interaction under SEA and SSA

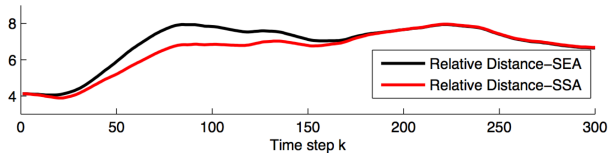
ually due to the reduction of uncertainties as an effect of learning. In Fig.14b, the transparent circles represent the equivalent uncertainty levels introduced by λ_R^{SSA}, thus the radius remain constant throughout the time.

Figure 15 shows the distance profiles and the vehicle velocity profiles under SSA and SEA. Due to large initial uncertainties, the vehicle only started to accelerate after $k = 50$ (when the relative distance is large) in SEA. However, in SSA, the vehicle tried to accelerate in the very beginning, then decelerated when the relative distance to the human decreased. The velocity profile in SSA was serrated, while the one in SEA was much smoother. Meanwhile, in both algorithms, the relative distance was always greater than $d_{min} = 3$. However, before $k = 150$, the relative distance was kept larger in SEA than in SSA, since the vehicle was more conservative in SEA due to large uncertainty. Fig.16 shows that the *a priori* MSEE provides a perfect bound for the prediction error, while the prediction error reduces gradually which validates the learning algorithms.

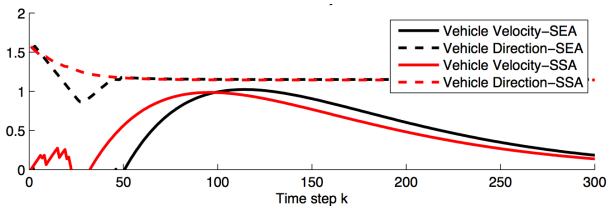
In conclusion, the behavior in SSA is: move and modify; while in SEA, it is: move only if confident. The behavior under SEA is better for a new comer, while the behavior under SSA is better if the robot is already very familiar with the environment, i.e. with low uncertainty levels.

6 Combining SSA and SEA in Time Varying MAS Topology

In real world applications, the system topology is usually time varying, e.g. the robot will encounter different agents at different time in different locations [8]. Mathematically, that means some agents will be decoupled from the system block diagram in Fig.4 and others will join from time to time. The robot is not faced with the 'same' system throughout the time. This scenario is common for mobile robots and automated vehicles [21].



(a) The relative distance profiles in SEA and SSA



(b) The velocity profiles of the vehicle in SEA and SSA

Fig. 15: Comparison between SEA and SSA

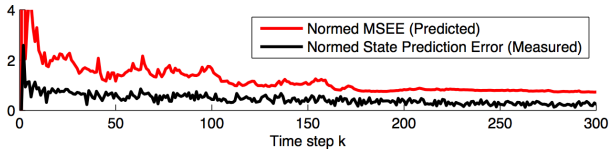


Fig. 16: Performance of the *a priori* MSEE as a Bound of the State Prediction Error

As the robot needs to deal with new agents, SEA is more appropriate than SSA. However, when the number of agents increase, the computation complexity with SEA increases dramatically. In this section, a method to combine SSA and SEA is discussed in order to balance the performance and the computation complexity.

6.1 The Control Algorithm

Due to limited sensing capability, the robot can only track humans that are within certain distance. Every agent within this range will be assigned a special identification number. Let $H(k)$ denotes the collection of those identification numbers at time step k. A safety index ϕ_j is designed for each agent $j \in H(k)$, as it is hard to design one analytical safety index that satisfies all the requirements for time varying $H(k)$. In this way, u_R needs to be constrained by

$$U_R^3 = \bigcap_{j \in H(k)} U_{R,j}^3 = \{u_R : L_j(k)u_R(k) \leq S_j(t, \dot{x}_j) \text{ for all } \dot{x}_j \in \dot{\Gamma}_j\} \tag{61}$$

where L_j and S_j are calculated with respect to ϕ_j. The uncertainty bound Γ_j is chosen according to SEA only for new agents. Once the MSEE converges, the algorithm is switched to SSA for that agent. The idea is illustrated in the following algorithm, where Π is a set that records the identification numbers of the agents who are no longer considered as new agents.

Initialize $\Pi = \emptyset$, $k = 0$;
while *Controller is Active* **do**
  $k = k + 1$;
  Read current $H(k)$ and $y_R^j(k)$ for $j \in H(k)$;
  for $j \in H(k)$ **do**
    calculate the estimate $\hat{x}_j(k+1|k)$ based on measurements $y_R^j(k)$;
    if $j \notin \Pi$ **then**
      calculate the MSEE $X_j(k+1|k)$;
      if X_j *converges* **then**
        $\Pi = \Pi \cup \{j\}$;
      end
    end
  end
  for $j \in H(k)$ **do**
    if $j \in \Pi$ **then**
      $U_{R,j}^3 = U_{R,j}^{SSA} = \{u_R : L_j(k) \leq S_j^{SSA}(k)\}$ (Apply SSA to ϕ_j);
    else
      $U_{R,j}^3 = U_{R,j}^{SEA} = \{u_R : L_j(k) \leq S_j^{SEA}(k)\}$ (Apply SEA to ϕ_j);
    end
  end
  $U_R^3 = \bigcap_{j \in H(k)} U_{R,j}^3$;
  Choose control u_R^* by optimizing over U_R^3;
end

Algorithm 1: The Algorithm Combining SSA and SEA

6.2 The Learning Algorithm

When the system topology is time varying, it is hard to learn humans' closed loop dynamics in (23). When the number of agents increases, the computation complexity regarding (23) increases exponentially. That is because the correlation among agents are over estimated, e.g. an agent's motion may only be affect by several surrounding agents instead of by all agents. When the system topology is time varying, it is more appropriate to learn agents' dynamics separately and use low dimension features to represent the correlations among agents. It is assumed that an agent j's motion will be affected by several features f_j^p for $p = 1, 2, \cdots$, e.g. the distance to the nearest agent and distance to the goal point. Then the linearized closed loop dynamics of agent j can be written as

$$x_j(k+1) = A_j(k)x_j(k) + \sum_p B_j^p(k) f_j^p(k) + w_j^*(k) \tag{62}$$

where $w_j^*(k)$ is the noise. Then the parameters $A_j(k)$ and $B_j^p(k)$ can be identified as discussed in section 4.2 and section 5.2.

6.3 Performance

The strategy is tested on robot navigation in a crowded environment with multiple humans as shown in Fig.17. The robot is modeled as a double integrator whose input is the acceleration in x and y directions. The position of the humans are controlled in real time by several human subjects who observe the virtual environment though a screen. The humans do not have specific tasks and are just 'wandering' in the environment. The robot is required to approach its goal while avoiding humans. The safety index is the same as in section 5.3. The features in (62) are chosen to be the distance to the closest human and the distance to the robot.

The simulation result is shown in Fig.18, Fig.19 and Fig.20. Before the 50-th time step, the robot was trying to approach its goal. It detoured when the blue agent came close. At the same time, the green agent which was previously hidden by the blue agent showed up in the robot's view. The new agent surprised the robot, as there was a large peak in the robot velocity profile in Fig.19. Algorithm-wise, it was the large uncertainty of the green agent that 'pushed' the robot away. The constraint U_R^3 was effective after the 50-th time step as evidenced in Fig.19. The relative distance between the robot and every human agent was always maintained greater than d_{min} as shown in Fig.20.

Fig. 17: The Simulation Environment for Robot Navigation.

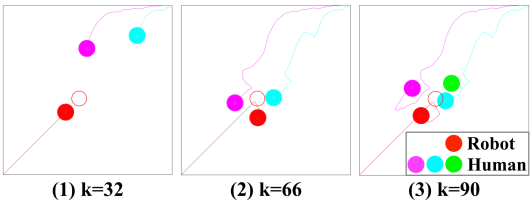


Fig. 18 Multi-human simulation results. Objective: the robot needs to navigate to its goal point in a crowded environment. Safety requirement: all relative distances are greater than d_{min}.

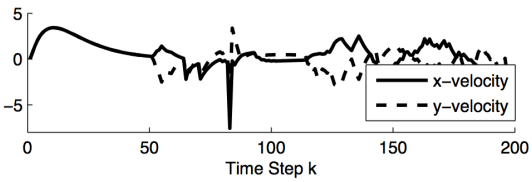


Fig. 19 The robot velocity profile in the multi-human simulation. After $k = 50$, the curves are no longer smooth due to the enforcement of the safety constraints.

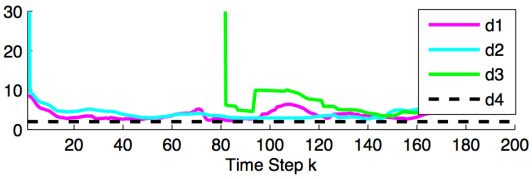


Fig. 20 The relative distance profile in the multi-human simulation. $d1, d2, d3$ are the distance between the robot and the respective human. $d4$ is the threshold d_{min}.

7 Discussions

Although derived from multi-agent interaction point of view, the safety-oriented behavior design method can be viewed as an energy based method, which is closely related to other energy based methods such as the potential field method and the Lyapunov method. In section 7.1, these methods will be compared. Moreover, although the proposed algorithms can handle a range of typical interactions such as robot navigation in a crowded environment and space sharing interactions among robot arms and human workers, there are certain limitations which will be discussed in section 7.2.

7.1 The Energy Based Methods

In an energy based method, a scalar energy function is usually defined such that the control objective (e.g. safety) is with low energy. So the desired control should drive the energy function in the negative direction. The potential function, the Lyapunov function and the safety index discussed in section 3.2 can all be regarded as energy functions. For a rough comparison, the emphasis in the potential field is the 'virtual repulsive force'; the emphasis in the Lyapunov theory is asymptotic convergence to the control objective; while the emphasis in the safe set algorithm is the time invariance regarding the control objective. To some extent, the potential field

method can be regarded as a 'force' control, while the safe set method is a 'position' control. The safety index is even closely related to the Lyapunov function, as the index can be transformed to an Lyapunov function by setting all negative values to be zero. The novel part of the proposed method is the introduction of multi objectives, e.g. the efficiency objective versus the safety objective. There are various ways to be safe, the robot needs to find the best one.

The multiple objectives is handled in the framework of constrained optimal control. Nonetheless, due to the introduction of the energy method (e.g. the safety index), the non-convex state space safety constraint is transformed to a convex control space constraint, which reduces the computation complexity comparing to other optimization methods. Thus the safe set method takes the advantage of both the energy method and the optimal control method.

7.2 Limitations and Future Work

Long Term Behavior versus Short Term Behavior

Both SSA and SEA are concerned with short term reactive behavior. It is possible that the robot get stuck in local optima. Long term planning is needed for the robot to get out of local optima. However, the uncertainties in the predictions of other agents will accumulate in the long term, which makes such predictions unreliable. In order to make the interactions smoother, the design method of the long term behavior will be studied in the future.

Structured Environment versus Unstructured Environment

Unstructured environments are also known as free spaces, while certain rules apply for structure environments. A typical structure environment is the traffic lanes shown in Fig.1. Both SSA and SEA work well in unstructured environment. In structured environment, since the environment is discretized, it is important to include discrete choices in the robot control and planning, such as the choice for lane changing or lane following for an automated vehicle [21].

Cooperation versus Coexistence

Cooperation usually involves physical contacts among agents, which is fundamentally different from coexistence. HRI which involves contacts can only be modeled by the indecomposable model in Fig.3. Then the human dynamics can not be separated from the robot dynamics as in Fig.8, which makes the learning and prediction even harder. Moreover, the safety issues in cooperation no longer come from conflicts as the objectives of the agents are identical, but from mis-interpretation or

mis-understandings instead [22]. For example, it is dangerous when the robot motion is unexpected by the human. Cooperation will be analyzed in the future for a better understanding of interactions.

8 Conclusion

This chapter discussed the general methodology in designing the robot controller for safe HRI. By modeling the system in a multi-agent framework, the safety issues were understood as conflicts in the multi-agent system. To solve the conflicts, the robot's behavior was constrained according to the 'social norm' and the uncertainties it perceived for the other agents (including humans and other robots). Two algorithms were discussed under the framework: the safe set algorithm (SSA) and the safe exploration algorithm (SEA). In both algorithms, the robot calculated the optimal action to finish the task while staying safe with respect to the predicted human motion. The difference was that SEA actively tracked the uncertainty levels in the prediction and incorporate that information in robot control, while SSA did not. As shown in the human-involved simulations, SEA was better when the uncertainty levels change from time to time, especially in the early stages of human robot interactions. On the other hand, SSA was better when the predictions were more accurate, e.g. when the robot was 'familiar' with the human, as SSA was more computationally efficient than SEA. Finally, a method to combine both algorithms was proposed to take the advantage of both algorithms. Several case studies were presented and demonstrated the effectiveness of the method.

In the future, the long term behavior, the safety issues in structure environments and the safety issues in human robot cooperation will be studied.

Acknowledgements This work was supported in part by a Berkeley Fellowship awarded to Changliu Liu.

References

[1] (2015) Driverless car market watch. URL http://www.driverless-future.com
[2] Armbruster W, Böge W (1979) Bayesian game theory. In: Game Theory and Related Topics, vol 17, p 28
[3] Basar T, Olsder GJ (1995) Dynamic noncooperative game theory, vol 200. London: Academic press
[4] Berger CR, Calabrese RJ (1975) Some explorations in initial interaction and beyond: Toward a developmental theory of interpersonal communication. Human communication research 1(2):99–112
[5] Burns LD (2013) Sustainable mobility: a vision of our transport future. Nature 497(7448):181–182

[6] Craig JJ (2005) Introduction to robotics: mechanics and control. Pearson/Prentice Hall
[7] Du Toit NE, Burdick JW (2012) Robot motion planning in dynamic, uncertain environments. Robotics, IEEE Transactions on 28(1):101–115
[8] Franchi A, Secchi C, Son HI, Bülthoff HH, Giordano PR (2012) Bilateral teleoperation of groups of mobile robots with time-varying topology. Robotics, IEEE Transactions on 28(5):1019–1033
[9] Goodwin GC, Sin KS (2013) Adaptive filtering prediction and control. Courier Dover Publications
[10] Gracia L, Garelli F, Sala A (2013) Reactive sliding-mode algorithm for collision avoidance in robotic systems. Control Systems Technology, IEEE Transactions on 21(6):2391–2399
[11] Haddadin S, Albu-Schaffer A, Eiberger O, Hirzinger G (2010) New insights concerning intrinsic joint elasticity for safety. In: Proceedings of Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, IEEE, pp 2181–2187
[12] Hartl RF, Sethi SP, Vickson RG (1995) A survey of the maximum principles for optimal control problems with state constraints. SIAM review 37(2):181–218
[13] Khatib O (1986) Real-time obstacle avoidance for manipulators and mobile robots. The International Journal of Robotics Research 5(1):90–98
[14] Kong K, Bae J, Tomizuka M (2009) Control of rotary series elastic actuator for ideal force-mode actuation in human–robot interaction applications. Mechatronics, IEEE/ASME Transactions on 14(1):105–118
[15] Krüger J, Lien T, Verl A (2009) Cooperation of human and machines in assembly lines. CIRP Annals-Manufacturing Technology 58(2):628–646
[16] Leber J (2013) At volkswagen, robots are coming out of their cages. URL http://www.fastcoexist.com/3016848/at-volkswagen-robots-are-coming-out-of-their-cages
[17] Liu C, Tomizuka M (2014) Control in a safe set: addressing safety in human robot interactions. In: ASME 2014 Dynamic Systems and Control Conference, ASME, p V003T42A003
[18] Liu C, Tomizuka M (2014) Modeling and controller design of cooperative robots in workspace sharing human-robot assembly teams. In: Intelligent Robots and Systems (IROS), 2014 IEEE/RSJ International Conference on, IEEE, pp 1386–1391
[19] Liu C, Tomizuka M (2015) Safe exploration: Addressing various uncertainty levels in human robot interactions. In: American Control Conference (ACC), pp 465 – 470
[20] Liu C, Tomizuka M (2016) Algorithmic safety measures for intelligent industrial co-robots. In: Proceedings of Robotics and Automation (ICRA), 2016 IEEE International Conference on, IEEE, p to appear
[21] Liu C, Tomizuka M (2016) Enabling safe freeway driving for automated vehicles. In: 2016 American Control Conference, IEEE, p to appear

[22] Liu C, Tomizuka M (2016) Who to blame? Learning and control strategies with information asymmetry. In: 2016 American Control Conference, IEEE, p to appear
[23] Liu J (2001) Autonomous Agents and Multi-Agent Systems: Explorations in Learning, Self-Organization, and Adaptive Computation. World Scientific
[24] Mohr J, Spekman R (1994) Characteristics of partnership success: partnership attributes, communication behavior, and conflict resolution techniques. Strategic Management Journal 15(2):135–152
[25] Ozguner U, Stiller C, Redmill K (2007) Systems for safety and autonomous behavior in cars: The darpa grand challenge experience. Proceedings of the IEEE 95(2):397–412
[26] Park DH, Hoffmann H, Pastor P, Schaal S (2008) Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields. In: Humanoid Robots, 2008 IEEE-RAS International Conference on, IEEE, pp 91–98
[27] Park HK, Hong HS, Kwon HJ, Chung MJ (2001) A nursing robot system for the elderly and the disabled. International Journal of Human-friendly Welfare Robotic Systems (HWRS) 2(4):11–16
[28] Russell S, Norvig P (1995) Artificial Intelligence: A Modern Approach. Prentice Hall
[29] Schulman J, Ho J, Lee A, Awwal I, Bradlow H, Abbeel P (2013) Finding locally optimal, collision-free trajectories with sequential convex optimization. In: Robotics: Science and Systems (RSS), vol 9, pp 1–10
[30] Sisbot EA, Marin-Urias LF, Broquere X, Sidobre D, Alami R (2010) Synthesizing robot motions adapted to human presence. International Journal of Social Robotics 2(3):329–343
[31] Tachi S, Komoriya K (1984) Guide dog robot. Autonomous Mobile Robots: Control, Planning, and Architecture pp 360–367
[32] Tadele TS, Vries TJd, Stramigioli S (2014) The safety of domestic robots: a survey of various safety-related publications. Robotics and Automation Magazine, IEEE pp 134–142
[33] Tsai CS, Hu JS, Tomizuka M (2014) Ensuring safety in human-robot coexistence environment. In: Proceedings of Intelligent Robots and Systems (IROS), 2014 IEEE/RSJ International Conference on, IEEE, pp 4191–4196
[34] Young JE, Hawkins R, Sharlin E, Igarashi T (2009) Toward acceptable domestic robots: Applying insights from social psychology. International Journal of Social Robotics 1(1):95–108