Proceedings of the ASME 2014 Dynamic Systems and Control Conference
DSCC 2014
October 22-24, 2014, San Antonio, Texas, USA

DSCC2014-6048

CONTROL IN A SAFE SET: ADDRESSING SAFETY IN HUMAN-ROBOT INTERACTIONS

Changliu Liu
Mechanical System Control Lab
Department of Mechanical Engineering
University of California
Berkeley, California 94720
Email: changliuliu@berkeley.edu

Masayoshi Tomizuka
Mechanical System Control Lab
Department of Mechanical Engineering
University of California
Berkeley, California 94720
Email: tomizuka@me.berkeley.edu

ABSTRACT

Human-robot interactions (HRI) happen in a wide range of situations. Safety is one of the biggest concerns in HRI. This paper proposes a safe set method for designing the robot controller and offers theoretical guarantees of safety. The interactions are modeled in a multi-agent system framework. To deal with humans in the loop, we design a parameter adaptation algorithm (PAA) to learn the closed loop behavior of humans online. Then a safe set (a subset of the state space) is constructed and the optimal control law is mapped to the set of control which can make the safe set invariant. This algorithm is applied with different safety constraints to both mobile robots and robot arms. The simulation results confirm the effectiveness of the algorithm.

INTRODUCTION

Human robot interactions (HRI) have been a topic of both science fiction and academic speculation for a long time [1]; however, it is the development of modern robotics and artificial intelligence that constitutes a solid ground for the analysis and applications of HRI.

Nowadays, physical human-robot interactions take place in great abundance. In factories, robots are leaving their cages and starting to work cooperatively with human workers. Car giants, such as Volkswagen and BMW, have launched cooperative robots in their factories in 2013 [2]. These manufacturers employ cooperative robots to take the advantage of both the human's flexibility and the robot's productivity, to meet the needs of flexible production [3]. Additionally, in view of the success of Google's autonomous cars, many car makers have been investing in driverless vehicles recently. Human drivers and autonomous cars will interact on the road, which poses new challenges in road safety [4]. 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 very close physical interactions between the patient and the robot [5]. Others like nursing robots [6] or robot guide dogs [7] are in great demand and involve HRI.

Safety is one of the biggest concerns in HRI. The difficulty of designing a safe robot controller comes from (i) sophistication of human motion and (ii) real time implementation of the algorithm. Control theory offers plenty of methods to deal with disturbances. However, when humans are in the loop, the disturbances introduced by human motions are much more complicated. They are time varying with unknown models. Moreover, it seems counterproductive to view human motions as disturbances in the robot control, as the system should be human-oriented. Deep learning of human behaviors is needed to make robots compliant. Even with robots learning human behaviors, another concern is that a complex control algorithm may not be executed fast enough to ensure quick responses of robots in emergencies.

To address safety, some authors use potential field methods, e.g. introducing a virtual force that is generated by an obstacle [8]. Some use numerical optimization methods which put the safety requirements in hard constraints [9, 10]. Others use sliding mode methods to locally modify the reference trajectory in the presence of obstacles [11]. However, the motion patterns of human subjects are much more complicated than that of general obstacles, as humans make decisions and react to the robots. To account for human decisions, a method is proposed in [12], which uses a joint probability distribution to describe the interaction between a group of people and a robot. However, the probability distribution for human motions is generally very hard to construct.

A safe set method is proposed in this paper to address the real time safety in HRI. All humans and robots in the system are regarded as agents. This method consists of two steps. Firstly, the closed loop dynamics of all human agents are estimated online to account for human decisions and to avoid the difficulty in getting the exact model of human motions. Secondly, control is applied in a safe set to account for various constraints and to enable fast online computation. The safe set is a subset of the state space that is constructed to be invariant under a feedback-like control law.

The remainder of this paper is organized as follows: firstly, a multi-agent interaction model will be introduced. Based on that model, the safe set method will be proposed with theoretical guarantees that the system will always stay in the safe region. Finally, the safe set method will be applied in three different cases to validate the effectiveness of the algorithm.

INTERACTION MODEL

There are two types of interactions, space sharing and time sharing. In space sharing interactions, humans and robots are doing their respective works in a shared workspace. In time sharing interactions, humans and robots are jointly finishing one task. In this paper, only space sharing interactions are studied.

State Space Model

To describe human robot interactions, a multi-agent model is needed. Robots and humans are all regarded as agents in this model. Suppose there are N agents in the system and are indexed from 1 to N. Let R be the set of indices for the robots, and H be the set of indices for the humans. We denote each agent's state by $x_i \in \mathbb{R}^{nx_i}$ for $i = 1,...,N$, where nx_i is the dimension of x_i. The control input of each agent is $u_i \in \mathbb{R}^{nu_i}$ for $i = 1,...,N$, where nu_i is the dimension of u_i. Then the state of the system is the union of all agents' states, i.e. $x = [x_1^T, x_2^T, \ldots, x_N^T]^T \in \mathbb{R}^{nx}$, where $nx = \sum_{i=1}^{N} nx_i$. For simplicity, we write x_R as the union of the states of all robots and x_H as the union of the states of all humans. Let $X \in \mathbb{R}^{nx}$ be the set of all possible values in the system's state space. 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.

If the agents are independent with respect to each other, the open loop dynamics of x_i is solely determined by x_i, u_i and the noise w_i. Then the open loop system can be decoupled as

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

This case arises in human-robot cooperative manufacturing [13], since the dynamics of either the human or the robot does not depend on the other. However, in general, the open loop system cannot 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. Then the robot's input enters the human's dynamic equation, which renders the open loop system indecomposable. In this paper, only decomposable systems are studied.

Every agent has its measurement of the system

$$y_i = h_i(x, v_i), \forall i = 1,...,N \tag{3}$$

where v_i is the measurement noise. Assume there is no direct communication among the agents. Then for any agent i, he can only choose his control u_i based on the measurement y_i and his reference goal $G_i \in \mathbb{R}^{nx_i}$, i.e.

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

Since we assume that the system is space sharing but not time sharing, G_i specifies space relations instead of time relations. Denote the system goal as $G = \left[G_1^T, ..., G_N^T\right]^T \in X$. Then the closed loop system is

$$\begin{aligned}\dot{x} &= f(x, g_1(h_1(x, v_1), G_1), ..., g_N(h_N(x, v_N), G_N), w) \\ &= f'(x, G, v_1, ..., v_N, w)\end{aligned} \tag{5}$$

Since the agents will react to all system state variables that they observe, the closed loop system is no longer decomposable. The system block diagram for the general multi-agent system is shown in Fig.1. The block diagram for the decomposable system is shown in Fig.2. The interaction in the decomposable system is easier to deal with as the interaction happens only in the measurement and controller side, instead of in the open loop dynamics.

Information Structure

In a multi-agent system, information structure is important as it determines the system's type [14]. Suppose every agent

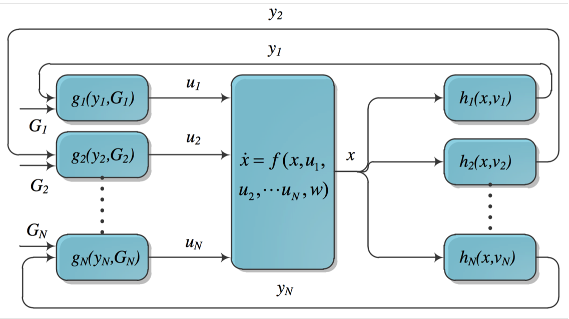


FIGURE 1: Multi-Agent System Block Diagram

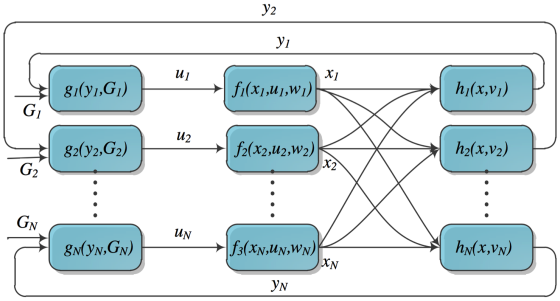


FIGURE 2: Decomposable Multi-Agent System Block Diagram

i is trying to minimize a cost function $J_i(x)$. Since J_i depends on the system state x, all agents can affect J_i through their controls. Thus the control u_i that minimizes J_i is actually a function of all other u_j's for $j \neq i$. Before choosing the control u_i, every agent i needs to 'guess' or observe the 'strategies' the others will play (i.e. the control signal u_j's that others would choose). When all agents are operating on the same information set (i.e. no one knows the others' strategies for the next move) before they take the next move, this is a simultaneous game and the optimal solution of this system defines a Nash equilibrium. 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. The agents being observed are considered as leaders and the agents that take the observation are considered as followers.

We assume that safety-oriented HRI can be modeled as a sequential game and the robot being controlled is always a follower. There are several advantages in doing so. First, it is a conservative strategy to play, thus good from the safety point of view. Second, we assume that robots have a smaller reaction time than humans, thus they can adapt very fast to human motion. In this way, robots are qualified as followers. Last but not least, to analyze a human's control strategy, we need to know his 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.

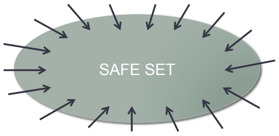


FIGURE 3: Invariant Safe Set

SAFE SET APPROACH

During human-robot interactions, the system needs to be safe, e.g. no collisions. In this section, the safe set approach to ensure safety is discussed. A general framework is set up first with a naive safe set approach. Then three modifications are made to make the algorithm realizable and to account for humans in the loop and uncertainties. In the last, the discrete time implementation are discussed and two examples of designing safety indices are given.

Naive Safe Set Approach

Let X_S be a subset of the state space X that is collision free. X_S represents the safe set. A safety index $\phi_0 : X \rightarrow \mathbb{R}$ is defined to be a functional on the state space such that $\phi_0(x) \leq 0$ if and only if $x \in X_S$. To be safe, the set X_S must be invariant with respect to time, i.e. $\phi_0(x(t)) \leq 0$, $\forall t \geq t_0$, where t_0 denotes the start time. One way to enforce invariance is by making X_S the region of attraction (ROA) of the whole space, i.e.

$$\dot{\phi}_0 < 0 \; when \; \phi_0 \geq 0 \tag{6}$$

where $\dot{\phi}_0$ is the derivative of ϕ_0 with respect to time t[1]. The intuition behind this control law is that once the system deviates from the safe set, the control flow will pull it back as illustrated in Fig.3. By Lyapunov's theorem, the system will asymptotically converge to the safe set X_S under this control law. Moreover, once the system enters the safe set, it will never go out. Let $U_S^i(t)$ be the set of the control u_i for agent i at time t that is consistent with the control law in Eq.(6), i.e.

$$U_S^i(t) = \left\{ u_i(t) : \dot{\phi}_0 z(\phi_0) \leq -\eta_i z(\phi_0) \right\} \tag{7}$$

where $z(\phi_0) = \frac{1}{2}(sgn(\phi_0) + 1)$, $z(0) = 1$. $\eta_i \in \mathbb{R}^+$ is a safety margin. To make the system safe, all robots should choose their control inputs from U_S^i. For $i \in R$, when the dynamic equation is

[1] ϕ_0 may not be everywhere differentiable. But we do not need to worry about this situation since we can always convolve ϕ_0 with a smooth function to make it differentiable. This property enables us to deal with complicated constraints.

affine, i.e. $\dot{x}_i = f^*_{ix}(x_i, w_i) + f^*_{iu}(x_i)\, u_i$, $U^i_S(t)$ is a closed convex set for all t, since Eq.(7) can be written as

$$U^i_S(t) = \{u_i(t) : L(t) u_i(t) \leq S(t)\} \tag{8}$$

where $S(t) = \left[-\eta_i - \sum_{j \neq i} \frac{\partial \phi_0}{\partial x_j} \dot{x}_j - \frac{\partial \phi_0}{\partial x_i} f^*_{ix}(x_i, w_i)\right] z(\phi_0)$, $L(t) = \frac{\partial \phi_0}{\partial x_i} f^*_{iu} z(\phi_0)$. Thus $U^i_S(t)$ is either a half space or the whole space. Suppose we have an optimal control law $u^o_i(t)$ for the robot i, which drives the robot to its goal. Then we map $u^o_i(t)$ to the set of safe control $U^i_S(t)$ according to the following cost function

$$J_i(u_i) = \frac{1}{2}(u_i - u^o_i)^T Q (u_i - u^o_i) \tag{9}$$

where $Q \in \mathbb{R}^{nu_i \times nu_i}$ is positive definite. Thus the final control satisfies $u^*_i(t) = \arg\min_{u_i \in U^i_S(t)} J_i(u_i)$. To solve the optimization problem, we need to consider the following two cases:

$u^o_i \in U^i_S$: then $u^*_i(t) = u^o_i(t)$.

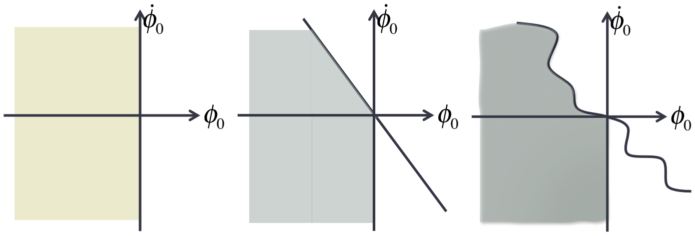
$u^o_i \notin U^i_S$: we need to invoke the Lagrange method. Since $U^i_S(t)$ is a half space, the optimal solution should be on the boundary $Lu_i = S$. Let λ be the Lagrange multiplier. The new cost function is $J^*_i = J_i + \lambda(Lu_i - S)$. The optimal solution satisfies $\frac{\partial J^*_i}{\partial u_i} = \frac{\partial J^*_i}{\partial \lambda} = 0$, i.e. $Q(u_i - u^o_i) + \lambda L^T = 0$ and $Lu_i - S = 0$. By solving these two constraints, we have $u^*_i = u^o_i - \lambda Q^{-1} L^T$ and $\lambda = \frac{Lu^0_i - S}{LQ^{-1}L^T}$.

Define $c = \inf_{u \in U^i_S(t)} |L(u - u^o_i)|$. When $u^o_i \in U^i_S(t)$, $c = 0$, otherwise, $c = \lambda LQ^{-1}L^T$. Thus for any u^o_i, we have

$$u^*_i(t) = u^o_i(t) - c\frac{Q^{-1}L^T}{LQ^{-1}L^T} \tag{10}$$

However, there are three problems in this naive safe set approach. First, $u_i(t)$ may not enter the expression $\dot{\phi}_0 z(\phi_0)$ explicitly if the relative degree from ϕ_0 to u_i in the sense of lie derivative is greater than one, i.e. $\left\| \frac{\partial \phi_0}{\partial x_i} f^*_{iu} \right\| = 0$. Then $U^i_S(t)$ is either an empty set or the whole space, which is not desirable for safe control. The second problem is that U^i_S may depend on u_j for $j \neq i$ since U^i_S depends on $\dot{x}_j$. This is a common situation in multi-agent systems. To solve this problem, we need to incorporate the information structure of the system. Thirdly, there are noise terms in U^i_S.

To address these problems, a new way in designing the safety index, which ensures that the safety index has relative degree one to the robot's input, will be introduced. Then we consider the closed loop behavior of all human agents to avoid the difficulty in reasoning about u_H. This can be justified when we assume that humans are leaders in the game. Robots are just playing a reactive strategy. For uncertainties, an upper bound will be added.



(a) $X(\phi_0)$ (b) $X(\phi_n)$ (c) $X(\phi^*_n)$

FIGURE 4: Relationship between $X(\phi_0), X(\phi_n), X(\phi^*_n)$

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

Designing a Safety Index

The problem is that we want to enforce invariance in the set $X_S = \{x : \phi_0 \leq 0\}$, but the relative degree from ϕ_0 to u_i may not be one. Then we can try the following safety index where the relative degree from $\phi^{(n)}_0$ to u_i is one:

$$\phi_n = \phi_0 + k_1 \dot{\phi}_0 + \ldots + k_n \phi^{(n)}_0 \tag{11}$$

where $k_1, \ldots, k_n$ are real-valued coefficients. Define $X(\phi)$ to be the set of all reachable states under the control law $\dot{\phi} < 0$ *when* $\phi \geq 0$ and with the initial condition $x(t_0) = x_0 \in X_s$. Obviously, $X(\phi_0) = X_S$. We say ϕ defines an invariant set in X_S if and only if $X(\phi) \subset X_S$. For simplicity, denote $\phi(x(t))$ by $\phi(t)$.

Theorem 1. *ϕ_n in Eq.(11) defines an invariant set in X_S, if (i) all roots of $1 + k_1 s + k_2 s^2 + \ldots + k_n s^n = 0$ are on the negative real line; (ii) ϕ_0 are n-th order differentiable. Moreover, suppose ϕ^*_0 defines the same set as ϕ_0 does, i.e. $\{x : \phi^*_0 \leq 0\} = X_S$, then*

$$\phi^*_n = \phi_n - \phi_0 + \phi^*_0 \tag{12}$$

also defines an invariant set in X_S.

The relationship between $X(\phi_0)$, $X(\phi_n)$ and $X(\phi^*_n)$ in the two dimensional case is illustrated in Fig.4. In (a), the shaded area is the set $X_S = X(\phi_0)$. In (b), the area below the line is the set defined by $\phi_n \leq 0$. But all the flows on the set

$\{(\phi_0, \dot{\phi}_0) : \dot{\phi}_0 \leq 0, \phi_0 = 0\}$ are pointing to the left, i.e. making ϕ_0 smaller. Since the initial condition states that $\phi_0(t_0) \leq 0$, the triangle area defined by $\{\phi_n \leq 0, \phi_0 > 0\}$ can never be reached. Thus the shaded area represents $X(\phi_n)$, which is an invariant subset in X_S. In (c), we introduce nonlinearity to the safety index by substituting ϕ_0 with ϕ_0^*. Since ϕ_0 and ϕ_0^* defines the same set, only the shape of the boundary is changed. The shaded area $X(\phi_n^*)$ is still an invariant subset in X_S. Rigorous proofs are as follows.

Proof. (i) ϕ_n defines an invariant set in X_S.

$\mathbf{n = 1}$: Since $1 + k_1 s = 0$ has a negative root, then $k_1 > 0$. Suppose $X(\phi_n)$ is not a subset of X_S, i.e. $\exists t_2$, s.t. $\phi_n(t_2) \leq 0, \phi_0(t_2) > 0$. Then $\phi_0(t_2) + k_1\dot{\phi}_0(t_2) \leq 0 < \phi_0(t_2)$, which implies $\dot{\phi}_0(t_2) < 0$. Since $\phi_0(t)$ is differentiable, $\phi_0(t)$ is continuous. Thus $A = \phi_0^{-1}([0,\infty]) \cap [t_0, t_2]$ is a closed set. Pick the largest connected set in A containing t_2 and denote it by $[t_1, t_2]$. Thus $\phi_0(t_1) = 0$ and $\phi_0(t) \geq 0$, $\forall t \in [t_1, t_2]$. Then $\dot{\phi}_0(t) \leq 0$ $\forall t \in [t_1, t_2]$ since $\phi(t) \leq 0$. So $\phi_0(t_2) = \phi_0(t_1) + \int_{t_1}^{t_2} \dot{\phi}_0(t)dt \leq \phi_0(t_1) = 0$. However, this contradicts with $\phi_0(t_2) > 0$. Thus we can conclude $X((1+k_1 s)\phi_0) \subset X_S$.

$\mathbf{n > 1}$: Decompose ϕ_n as $\phi_n = \phi_0 + k_1\dot{\phi}_0 + ... + k_n\phi_0^{(n)} = (1+a_1 s)(1+a_2 s)...(1+a_n s)\phi_0$. Then we can conclude, $X(\phi_0) \supset X((1+a_n s)\phi_0) \supset X((1+a_{n-1}s)(1+a_n s)\phi_0) \supset X((1+a_1 s)...(1+a_n s)\phi_0)$. So $X(\phi_n) \subset X(\phi_0) = X_S$. [2]

(ii) ϕ_n^* defines an invariant set in X_S.

If the statement is not true, then $\exists t_2 > t_0$ s.t. $\phi_n^*(t_2) \leq 0$, $\phi_0(t_2) > 0$. For the same reason in (i), there exists $t_1 \in [t_0, t_2]$ s.t. $\phi_0(t_1) = 0$ and $\phi_0(t) \geq 0$ for $t \in [t_1, t_2]$. Since ϕ_0 and ϕ_0^* defines the same set, $\phi_0^*(t_1) = \phi_0(t_1) = 0$. Thus $\phi_n(t_1) = \phi_n^*(t_1) \leq 0$ by definition. As proved in (i), when $\phi_0(t_1) = 0$ and $\phi_n(t_1) \leq 0$, it must follow that $\dot{\phi}_0(t_1) \leq 0$ (otherwise, it will contradict with the fact that ϕ_n defines an invariant set in X_S). Thus it is impossible for ϕ_0 to go beyond 0, contradicts with $\phi_0(t_2) > 0$. Indeed, ϕ_n^* also defines an invariant set in X_S.

From now on, we will denote the index that is properly designed simply as ϕ.

Closed Loop Human Behavior

To address the problem mentioned in the naive safe set approach, we need to study the closed loop behavior of the humans.

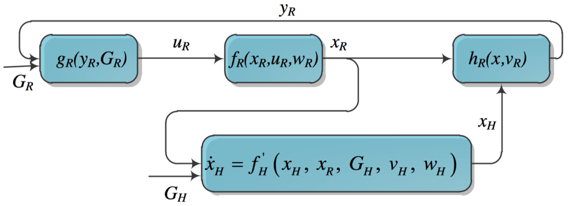


FIGURE 5: Block Diagram with Closed Loop Human Behavior

The closed loop dynamic equation for the humans is:

$$\begin{aligned}\dot{x}_H &= f_H(x_H, u_H) = f_H(x_H, g_H(h_H(x, v_H), G_H), w_H) \\ &= f_H'(x_H, x_R, G_H, v_H, w_H) \qquad (13)\end{aligned}$$

Since all humans are coupled together, we can take x_H as the state for a single human agent. In this way, the system in Fig.2 reduces to Fig.5. Notice that $y_R = h_R(x, v_R)$, suppose the measurement can be decoupled as

$$y_R^H = h_R^H(x_H, v_R^H) \qquad (14)$$

$$y_R^R = h_R^R(x_R, v_R^R) \qquad (15)$$

where y_R^H is the robot's measurement of the human, y_R^R is the robot's measurement of itself. v_R^H and v_R^R are the corresponding measurement noises. Equations (13) and (14) form a nonlinear Gaussian system with unknown parameters. Then we can do online parameter adaptation to learn the system properties. Unfortunately, nonlinear systems are hard to deal with in general, so an alternative choice is to discretize and linearize the model and run parameter adaptation algorithm (PAA) to estimate the linearized closed loop parameters online. This method can be justified if we assume that a human does not change very fast with respect to the sampling rate. By linearizing and discretizing Eq.(13) and substituting x_R with the robot's state estimate $\hat{x}_R$ from a state estimator (i.e. Kalman Filter), we have

$$x_H(k+1) = A_H(k)x_H(k) + B_H(k)u_H^{cl}(k) + w_H^*(k) \qquad (16)$$

where $u_H^{cl}(k) = \left[\hat{x}_R(k|k)^T, ..., \hat{x}_R(k-m|k)^T, G_H^T(k)\right]^T$ and m represents the delay in the human reaction. $w_H^*(k)$ is a noise term assumed to be zero-mean Gaussian and white. We explicitly consider the human's reaction to the robot's state in Eq.(16). Methods for inferring $G_H(k)$ are discussed in [13]. In this paper, we assume it is known. Assume that Eq.(14) can be written as

$$y_R^H(k) = x_H(k) + v_R^H(k) \qquad (17)$$

[2] Notice that if $1 + k_1 s + k_2 s^2 + ... + k_n s^n = 0$ has roots outside the negative real line, the statement above is generally not true (even if the roots are all stable). Consider the boundary defined by the index ϕ_n, i.e. $\phi_n = \phi_0 + k_1\dot{\phi}_0 + ... + k_n\phi_0^{(n)} = 0$. If there are unstable roots, then $\phi_0 \to \infty$ as $t \to \infty$, which is not desired. Suppose all roots are stable, $\phi_0 \to 0$ as $t \to \infty$. Since $\phi_0(t_0) < 0$, this is like a step response from $\phi_0(t_0)$ to 0. If the damping ratio of the characteristic equation is less than one, then there will be overshoots, i.e. $\exists t > t_0$ s.t. $\phi_0(t) > 0$. Thus ϕ_n no longer defines an invariant set in X_S.

Equations (16) and (17) then form a linear time varying (LTI) Gaussian system with unknown parameters. Let $\hat{x}_H(k|k-1)$ be the *a priori* estimate of the human's state at time k, $\hat{x}_H(k|k)$ the *a posteriori* estimate of the human's state at time k and $\hat{A}_H(k), \hat{B}_H(k)$ the estimates of the matrices given the information up to the k-th time step. At $k+1$-th time step, we first update the estimate of $\hat{x}_H$ according to the closed loop dynamics in Eq.(18), then incorporating the measurement information to get the *a posteriori* estimate in Eq.(19). Since the system is time varying, a constant update gain $\alpha \in (0,1)$ is chosen to ensure that the measurement information is always incorporated.

$$\hat{x}_H(k+1|k) = \hat{A}_H(k)\hat{x}_H(k|k) + \hat{B}_H(k)u_H^{cl}(k) \tag{18}$$

$$\hat{x}_H(k+1|k+1) = (1-\alpha)\hat{x}_H(k+1|k) + \alpha y_R^H(k+1) \tag{19}$$

Then the closed loop matrices are estimated using recursive least square (RLS) PAA as shown in Eq.(20):

$$\begin{aligned}\left[\hat{A}_H(k+1), \hat{B}_H(k+1)\right] &= \left[\hat{A}_H(k), \hat{B}_H(k)\right]\\ &+ \left(\hat{x}_H(k+1|k+1) - \hat{x}_H(k+1|k)\right)\varphi(k)^T F(k+1)\end{aligned} \tag{20}$$

where $\varphi(k) = \left[\hat{x}_H(k|k)^T \; u_H^{cl}(k)^T\right]^T$. F is the learning gain with the update equation:

$$F(k+1) = \frac{1}{\lambda}\left[F(k) - \frac{F(k)\varphi(k)\varphi^T(k)F(k)}{\lambda + \varphi^T(k)F(k)\varphi(k)}\right] \tag{21}$$

$\lambda > 0$ is a forgetting factor. We choose $\lambda < 1$ to make sure that the parameter adaptation algorithm is always active [15]. Since RLS algorithm minimizes the prediction error, the prediction performance from this algorithm is good as shown later.

Bounding Uncertainties

By redesigning the safety index, we can make $\left\|\frac{\partial\phi}{\partial x_i}f_{iu}^*\right\| \neq 0$ for $i \in R$. Then, by estimating the closed loop behavior of the humans, we can obtain the estimate $\dot{\hat{x}}_j$ for $j \in H$. For the robot, we estimate $\hat{x}_R$. In this way, Eq.(8) can be written as

$$\begin{aligned}U_S^R(t) = \Bigg\{ u_R(t) : &\sum_{i\in R}\frac{\partial\phi}{\partial x_i}f_{iu}^*(\hat{x}_i)u_i z(\phi) \\ &\le \left[-\eta_R - \lambda_R - \sum_{j\in H}\frac{\partial\phi}{\partial x_j}\dot{\hat{x}}_j - \sum_{i\in R}\frac{\partial\phi}{\partial x_i}f_{ix}^*(\hat{x}_i)\right] z(\phi)\Bigg\}\end{aligned} \tag{22}$$

where $\lambda_R \in \mathbb{R}^+$ is introduced to bound the uncertainties in PAA and state estimation. $\dot{\hat{x}}_j$ is approximated by $\left(\hat{x}_j(k+1|k) - \hat{x}_j(k|k)\right)/T_s$. Equation (22) is a realizable version of Eq.(8). The set $U_S^R(t)$ is also a closed convex set. The final control law is calculated by Eq.(10).

Discrete Time Implementation

In the discrete time, the set of safe control is only calculated at each time step k. Suppose the sampling time is T_s. Denote $\phi(x(k))$ by $\phi(k)$. Then

$$\begin{aligned}&U_S^R(t = kT_s)\\ &\approx \left\{u_R(k) : \frac{\phi(k+1)-\phi(k)}{T_s}z(\phi(k)) \le (-\eta_R - \lambda_R)z(\phi(k))\right\}\\ &= \left\{u_R(k) : \phi(k+1)z(\phi(k)) \le (\phi(k) - \eta_R T_s - \lambda_R T_s)z(\phi(k))\right\}\\ &\supset \left\{u_R(k) : \phi(k+1) \le -\eta_R T_s - \lambda_R T_s\right\}\end{aligned} \tag{23}$$

Let $\phi^d = \phi + \eta_R T_s + \lambda_R T_s$. The safety index ϕ^d takes into account the uncertainties and have a safety margin $\eta_R T_s$. Define the set of safe control in the discrete time as

$$U_S^R(k) = \left\{u_R(k) : \phi^d(k+1) \le 0\right\} \subset U_S^R(t = kT_s) \tag{24}$$

It is shown in Eq.(24) that the set of safe control $U_S^R(k)$ in the discrete case is a subset of the safe control $U_S^R(t = kT_s)$ in the continuous case. The convexity of $U_S^R(k)$ follows from $U_S^R(t)$ when $T_s \to 0$. Thus Eq.(10) still applies.

Safety Indices for Typical Safety Constraints

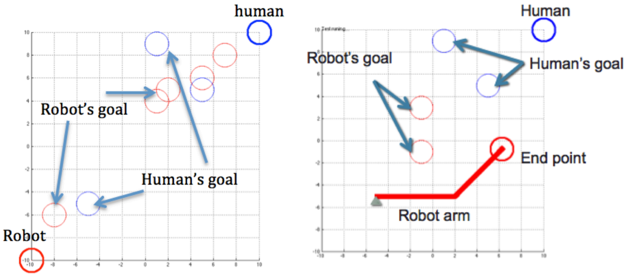
Safety constraints are usually dependent on the smallest relative distance d between the human H and the robot R. One typical constraint is $d \geq d_{min}$, where d_{min} is the minimum acceptable distance between the human and the robot. An intuitive safety index for this constraint is $\phi_0 = d_{min} - d \le 0$. Since the robot plant usually includes a double integrator, the relative degree from d to u is two. So $n = 1$ and $\dot{\phi}_0$ needs to be included. Define $\phi_n = \phi_0 + k_1\dot{\phi}_0 = d_{min} - d - k_1\dot{d}$. A heavier punishment on the relative velocity is desired when the relative distance is small. Thus we introduce a nonlinear term. Notice that $\phi_0^* = d_{min}^2 - d^2$ defines the same safe set as ϕ_0 does. Let $\phi_n^* = \phi_n - \phi_0 + \phi_0^* = d_{min}^2 - d^2 - k_1\dot{d}$. By taking into account the safety margin and let $k_1 = 1$, we have

$$\phi^d = D - d^2 - \dot{d} \tag{25}$$

where $D = d_{min}^2 + \eta_R T_s + \lambda_R T_s$.

Another typical safety constraint is $d_{min} \le d \le d_{max}$ when the robot is required to stay close to the human but not collide with the human. d_{max} is the maximum acceptable distance between the human and the robot. An intuitive safety index is $\phi_0 = (d - d_{min})(d - d_{max})$. By introducing $\dot{\phi}_0$, and considering the safety margin, we have the safety index as

$$\phi^d = \phi_0 + \dot{\phi}_0 + \eta_R T_s + \lambda_R T_s = D_1 + d^2 - D_2 d + 2d\dot{d} - D_2\dot{d} \tag{26}$$



(a) Mobile Robot Simulation Environment (b) Robot Arm Simulation Environment

FIGURE 6: Simulation Environments

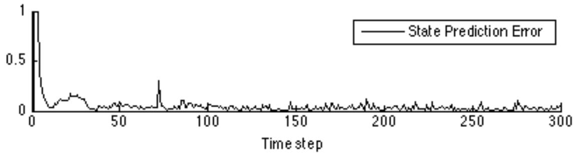


FIGURE 7: Prediction Performance of the Human's State

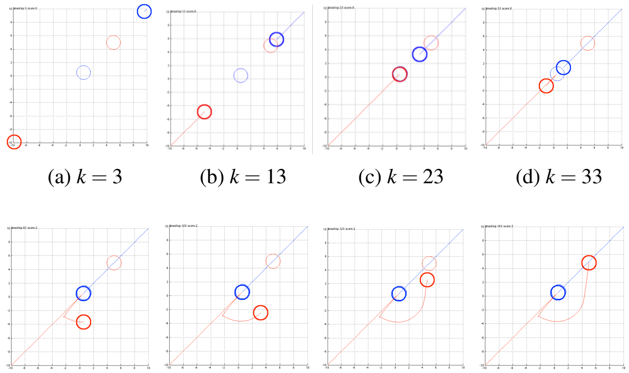
where $D_1 = d_{min}d_{max} + \eta_R T_s + \lambda_R T_s$, $D_2 = d_{min} + d_{max}$. The performance of the two safety indices will be shown in the case study.

CASE STUDY

In this section, simulation studies are conducted in three cases involving both mobile robots and robot arms. We consider two agents in the system, a robot and a human. The simulation environment (20×20) for mobile robots is shown in Fig.6a. Both the human and the robot are represented by circles of radius 1, while the robot is in red and the human in blue. The thin blue and the thin red circles are the goals for the human and the robot respectively. The simulation environment for the robot arm is shown in Fig.6b. The two-link robot arm is represented as the thick red broken line. A human subject can control the thick blue circle using a multi-touch pad during the simulation. Define $x_H = [h_x, \dot{h}_x, h_y, \dot{h}_y]^T$, where h_x is the human's x position, $\dot{h}_x$ the human's x velocity, h_y the human's y position and $\dot{h}_y$ the human's y velocity. The estimation of the human's closed loop dynamics follows from Eq.(18)-(21). The state prediction error $\|x_H(k+1) - \hat{x}_H(k+1|k)\|$ is shown in Fig.7. The error converges quickly with the PAA algorithm as shown in the figure. The parameter estimation performance is shown in [13].

Case 1: Mobile Cooperative Robot

The interaction between humans and a 2D robot will arise in factories as described in [13]. For simplicity, we only model the robot's kinematics. Let $x_R = [r_x, \dot{r}_x, r_y, \dot{r}_y]^T$ where r_x is the robot's x position, $\dot{r}_x$ the robot's x velocity, r_y the robot's y position and $\dot{r}_y$ the robot's y velocity. Let $u_R = [\ddot{r}_x, \ddot{r}_y]^T$. The kinematics of the robot is



(a) $k = 3$ (b) $k = 13$ (c) $k = 23$ (d) $k = 33$

(e) $k = 83$ (f) $k = 103$ (g) $k = 123$ (h) $k = 143$

FIGURE 8: Simulation of the Mobile Cooperative Robot

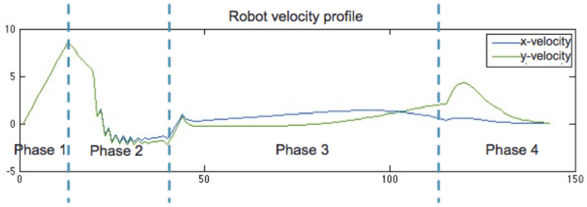


FIGURE 9: Velocity Profile of the Mobile Cooperative Robot

$$x_R(k+1) = A_R x_R(k) + B_R u_R(k) + w_R(k) \tag{27}$$
$$y_R^R(k) = x_R(k) + v_R(k) \tag{28}$$

where $A_R = \begin{bmatrix} 1 & T_s & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 1 \end{bmatrix}$, $B_R = \begin{bmatrix} \frac{1}{2}T_s^2 & 0 \\ T_s & 0 \\ 0 & \frac{1}{2}T_s^2 \\ 0 & T_s \end{bmatrix}$. w_R, v_R are independent zero mean white Gaussian noises. The robot's state is estimated by a Kalman Filter. The safety requirement is $d_{min} - d \leq 0$. Thus the safety index in Eq.(25) is adopted.

Safe Control The safety index is defined over the relative distance, which is dependent on the relative state between the human and the robot, i.e. $dx(k) = x_R(k) - x_H(k)$. The relative distance can be calculated as in Eq.(29); the product of the relative distance and the relative velocity can be calculated as in Eq.(30).

$$d^2(k) = (r_x - h_x)^2 + (r_y - h_y)^2 = dx(k)^T P_1 dx(k) \tag{29}$$

$$
\begin{aligned}
d(k)\,\dot d(k) &= (r_x - h_x)\left(\dot r_x - \dot h_x\right) + (r_y - h_y)\left(\dot r_y - \dot h_y\right)\\
&= dx(k)^T P_2 dx(k) \qquad (30)
\end{aligned}
$$

where $P_1 = \begin{bmatrix} 1\,0\,0\,0 \\ 0\,0\,0\,0 \\ 0\,0\,1\,0 \\ 0\,0\,0\,0 \end{bmatrix}, P_2 = \begin{bmatrix} 0 & 0.5 & 0 & 0 \\ 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 \\ 0 & 0 & 0.5 & 0 \end{bmatrix}$.

Due to noises, we must estimate the relative state, i.e.

$$
\begin{aligned}
\hat{dx}(k+1|k) &= \hat x_R(k+1|k) - \hat x_H(k+1|k)\\
&= I(k) + B_R u_R(k) \qquad (31)
\end{aligned}
$$

where $I(k) = A_R \hat x_R(k|k) - \hat x_H(k+1|k)$.

Notice that as $T_s \to 0$, $P_1 B_R \to 0$ and $B_R P_2 B_R \to 0$. Using the estimate (31) in Eq.(29) and (30), we obtain

$$\hat d^2(k+1|k) \approx I^T(k) P_1 I(k) \qquad (32)$$

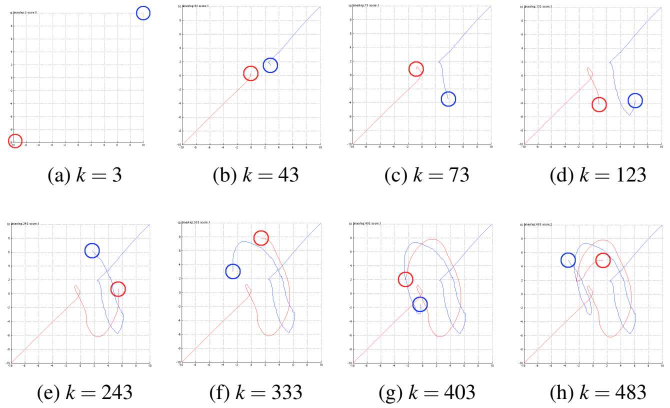
$$\hat d \hat{\dot d}(k+1|k) \approx I^T(k) P_2 I(k) + 2I^T(k) P_2 B_R u_R(k) \qquad (33)$$

Solving $\phi^d(k+1) \leq 0$ (where ϕ^d is defined in Eq.(25)) by Eq.(32) and (33), we have:

$$
\begin{aligned}
2I^T(k)P_2B_Ru_R(k) \geq D\sqrt{I^T(k)P_1I(k)}\\
-\left(I^T(k)P_1I(k)\right)^{\frac{3}{2}} - I^T(k)P_2I(k) \quad (34)
\end{aligned}
$$

The set of safe control $U_S^R(k)$ is still a half plane. Suppose the optimal control is $u_R^o = -K(x_R - G_R)$, where $K = \begin{bmatrix} 1\,2\,0\,0 \\ 0\,0\,1\,2 \end{bmatrix}$ is the control gain. The final control at time k is obtained from Eq.(10) where $L(k) = -2I^T(k)P_2B_R$ and $S(k) = -D\sqrt{I^T(k)P_1I(k)} + \left(I^T(k)P_1I(k)\right)^{\frac{3}{2}} + I^T(k)P_2I(k)$.

Safety Performance A simulation study was conducted for the environment shown in Fig.6a with $T_s = 0.1s$. A human subject controlled the blue circle through a multi-tough pad. As shown in Fig.8 and Fig.9, the robot successfully avoided the human under the safe set algorithm. In phase 1, the robot was executing the optimal control since it was far away from the human. In phase 2, the robot observed that the human was approaching, so it slowed down to meet the safety constraints. In phase 3, the human approached his goal and stayed there. The robot learned that the human would stay there for sometime through PAA. So it detoured to avoid hitting the human. In phase 4, the robot approached its goal successfully.


(a) $k = 3$ (b) $k = 43$ (c) $k = 73$ (d) $k = 123$

(e) $k = 243$ (f) $k = 333$ (g) $k = 403$ (h) $k = 483$

FIGURE 10: Simulation of the Robot Assistant

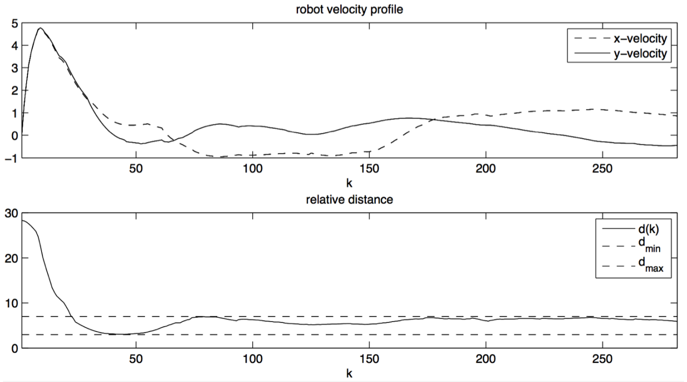


FIGURE 11: Velocity Profile and Relative Distance Profile of the Robot Assistant

Case 2: Mobile Robot Assistant

As we are gradually stepping into an aging society, more and more robot assistants are needed to assist the aged people. The requirement for robot assistants is that they should stay close to the human in order to take orders, while far enough to avoid collision. The system model and simulation environment in this case is the same as in case 1. The only difference is that there is no specific goal for the robot and the human. The safety index defined by Eq.(26) is adopted.

Safe Control Solving $\phi^d(k+1) \leq 0$ (where ϕ^d is defined in Eq.(26)) by Eq.(32) and (33), we have

$$2\left[D_2 - 2\sqrt{I^T(k)P_1I(k)}\right] I^T(k)P_2B_Ru_R(k) \geq H(k) \qquad (35)$$

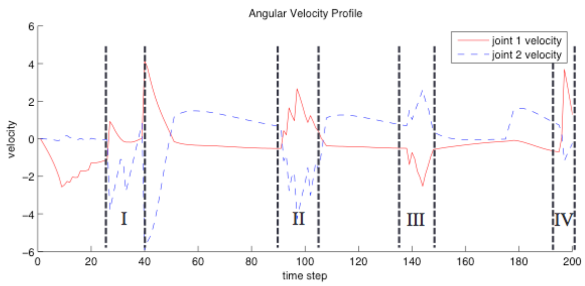
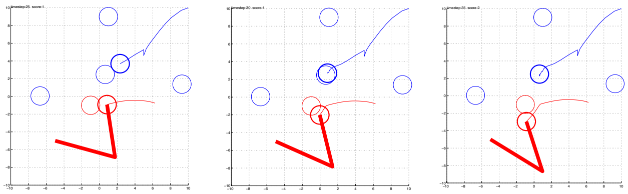
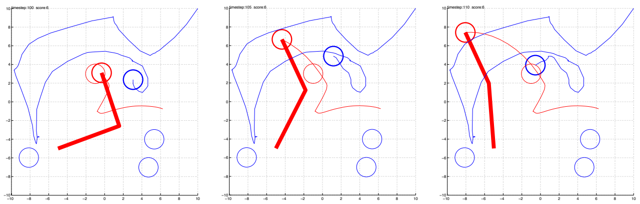


FIGURE 12: Velocity Profile of the Robot Arm



(a) $k=25$ (b) $k=30$ (c) $k=35$

FIGURE 13: Simulation of the Robot Arm, Scenario 1



(a) $k=100$ (b) $k=105$ (c) $k=110$

FIGURE 14: Simulation of the Robot Arm, Scenario 2

where $H(k) = D_1\sqrt{I^T(k)P_1I(k)} + (I^T(k)P_1I(k))^{\frac{3}{2}} - D_2I^T(k)P_1I(k) - \left[D_2 - 2\sqrt{I^T(k)P_1I(k)}\right]I^T(k)P_2I(k)$.

The optimal control law is designed as $u_R^o = -Kx_R$ where $K = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$. This is a stabilizing control law which will slow down the robot. Then the final control law follows from Eq.(10) where $L(k) = 2\left[2\sqrt{I^T(k)P_1I(k)} - D_2\right]I^T(k)P_2B_R$ and $S(k) = -H(k)$.

Safety Performance In the simulation, $d_{min} = 3$, $d_{max} = 7$. Figure 10 shows the simulation result and Fig.11 shows the velocity profile of the robot and the relative distance profile between the robot and the human. From the beginning of the simulation, the robot and the human were at two opposite corners. Then the robot moved towards the human, while the human also moved towards the robot. When the robot was very close to the human, it slowed down and turned to the up-left direction to avoid collision. Meanwhile, the human moved to the bottom-right corner. When detecting this, the robot changed its direction to follow human (at $k = 73$). Then the human started to go up. The robot followed the human closely from time step $k = 173$ to $k = 383$. At around 400-th time step, the human suddenly changed direction to move towards the robot. Then the robot gave way to the human by turning to the right. As shown in Fig.11, the distance is always in the safety range except in the beginning.

Case 3: Cooperative Robot Arm

The interaction between humans and a robot arm often arises in factories. Both the cooperative robots from Volkswagen and BMW are robot arms. The safety index in Eq.(25) is used with d being the smallest distance from the robot arm to the human.

The robot arm has two joints. Let θ_1 be the angular position of the first joint, $\dot{\theta}_1$ the angular velocity of the first joint, θ_2 the angular position of the second joint and $\dot{\theta}_2$ the angular velocity of the second joint. The control input is the angular accelerations of the two joints, i.e. $u_R = \left[\ddot{\theta}_1, \ddot{\theta}_2\right]^T$. The end point position of the robot is (p_x, p_y). The closest point to the human is (m_x, m_y). The optimal control law is designed to be

$$u_R^o = J_p^{-1}\left\{K_p\begin{bmatrix} p_x - g_x \\ p_y - g_y \end{bmatrix} + K_v\begin{bmatrix} \dot{p}_x \\ \dot{p}_y \end{bmatrix} - H_p\right\} \tag{36}$$

where J_p is the Jacobian matrix of p and $H_p = \dot{J}_p\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$. (g_x, g_y) is the goal point in the work space. $K_p \in \mathbb{R}^{2\times 2}$ and $K_v \in \mathbb{R}^{2\times 2}$ is the control gain. Equation (36) is valid when J_p is invertible. When J_p is not invertible, we perturb the system out of the singularity.

Safe Control Define the robot arm's state to be $x_R = \left[m_x \; \dot{m}_x \; m_y \; \dot{m}_y\right]$ with the state equation:

$$x_R(k+1) = A_R x_R(k) + B_R J_m u_R(k) + B_R H_m(k) \tag{37}$$

where J_m is the Jacobian matrix of m and $H_m = \dot{J}_m\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$. A_R, B_R is the same as in case 1. The prediction for the relative state is

$$\hat{dx}(k+1|k) = I(k) + B_R J_m u_R(k) \tag{38}$$

where $I(k) = A_R\hat{x}_R(k|k) + B_R\hat{H}_m(k|k) - \hat{x}_H(k+1|k)$. The set of safe control satisfies

$$2I^T(k)P_2B_RJ_mu_R(k) \geq D\sqrt{I^T(k)P_1I(k)} - \left(I^T(k)P_1I(k)\right)^{\frac{3}{2}} - I^T(k)P_2I(k) \quad (39)$$

The final control follows from Eq.(10) where $L(k) = -2I^T(k)P_2B_RJ_m$ and $S(k) = -D\sqrt{I^T(k)P_1I(k)} + \left(I^T(k)P_1I(k)\right)^{\frac{3}{2}} + I^T(k)P_2I(k)$.

Safety performance In this simulation, we assign several goals for the human. Before doing parameter adaptation, the robot performs inference on the human's current goal [13]. $T_s = 0.1s$. Figure 12 shows the velocity profile of the robot arm in one simulation. The robot tried to avoid human in parts I, II, III, and IV in the figure and executed the optimal control in the other parts. The human avoidance behavior of part I is shown in Fig.13. In (a), the human and the robot were both near their goals. But since the human had a large velocity towards the robot, the robot went backward in (b) and (c). Figure 14 shows the robot behavior under miscellaneous human behavior. In (a), the human suddenly changed his course. Although all of his goal points are in the lower part of the graph, the human started to go up. By observing that, robot went away from the human in (b) and (c).

Conclusion

In this paper, a multi-agent model in describing human-robot interactions was proposed. Based on that model, we discussed the safe set method to address the safety in physical HRI. By assuming that humans were always leaders in the system, we estimated the closed loop behavior of all human agents using PAA online. Then a safe set was constructed in the state space, which was defined by a safety index. The design method of the safety index was discussed. By mapping the optimal control to the set of control that made the safe set invariant, we got a feedback-like control law to guarantee the real time safety of the system. The case studies confirmed the effectiveness and efficiency of the algorithm.

The advantage of the proposed algorithm is that it is (1) faster than a numerical optimization approach (2) more flexible than an potential field method for its ability to deal with various constraints. In the future, this algorithm is to be implemented on real robots, and more agents are to be introduced in the study.

ACKNOWLEDGMENT

The authors thank Xingxing Cai, Kevin Haninger and Angela Liu for their helps in editing this paper.

REFERENCES

[1] Asimov, I., 1942. "Runaround". *Astounding Science Fiction,* ***29***(1), pp. 94–103.

[2] Leber, J., 2013. At volkswagen, robots are coming out of their cages, 9.

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

[4] Ozguner, U., Stiller, C., and Redmill, K., 2007. "Systems for safety and autonomous behavior in cars: The darpa grand challenge experience". *Proceedings of the IEEE,* ***95***(2), pp. 397–412.

[5] Kong, K., Bae, J., and 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), pp. 105–118.

[6] Park, H. K., Hong, H. S., Kwon, H. J., and Chung, M. J., 2001. "A nursing robot system for the elderly and the disabled". *International Journal of Human-friendly Welfare Robotic Systems (HWRS),* ***2***(4), pp. 11–16.

[7] Tachi, S., and Komoriya, K., 1984. "Guide dog robot". *Autonomous Mobile Robots: Control, Planning, and Architecture*, pp. 360–367.

[8] Park, D.-H., Hoffmann, H., Pastor, P., and Schaal, S., 2008. "Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields". In Humanoid Robots, 2008. Humanoids 2008. 8th IEEE-RAS International Conference on, IEEE, pp. 91–98.

[9] Schulman, J., Ho, J., Lee, A., Awwal, I., Bradlow, H., and Abbeel, P., 2013. "Finding locally optimal, collision-free trajectories with sequential convex optimization". In Robotics: Science and Systems (RSS).

[10] Du Toit, N. E., and Burdick, J. W., 2012. "Robot motion planning in dynamic, uncertain environments". *Robotics, IEEE Transactions on,* ***28***(1), pp. 101–115.

[11] Gracia, L., Garelli, F., and Sala, A., 2013. "Reactive sliding-mode algorithm for collision avoidance in robotic systems". *Control Systems Technology, IEEE Transactions on,* ***21***(6), Nov, pp. 2391–2399.

[12] Trautman, P., and Krause, A., 2010. "Unfreezing the robot: Navigation in dense, interacting crowds". In Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, IEEE, pp. 797–803.

[13] Liu, C., and 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.

[14] Basar, T., and Olsder, G. J., 1995. *Dynamic noncooperative game theory*, Vol. 200. London: Academic press.

[15] Goodwin, G. C., and Sin, K. S., 2013. *Adaptive filtering prediction and control.* Courier Dover Publications.