Algorithmic Safety Measures for Intelligent Industrial Co-Robots

Changliu Liu and Masayoshi Tomizuka, *Fellow, IEEE*

***Abstract*— In factories of the future, humans and robots are expected to be co-workers and co-inhabitants in the flexible production lines. It is important to ensure that humans and robots do not harm each other. This paper is concerned with functional issues to ensure safe and efficient interactions among human workers and the next generation intelligent industrial co-robots. The robot motion planning and control problem in a human involved environment is posed as a constrained optimal control problem. A modularized parallel controller structure is proposed to solve the problem online, which includes a baseline controller that ensures efficiency, and a safety controller that addresses real time safety by making a safe set invariant. Capsules are used to represent the complicated geometry of humans and robots. The design considerations of each module are discussed. Simulation studies which reproduce realistic scenarios are performed on a planar robot arm and a 6 DoF robot arm. The simulation results confirm the effectiveness of the method.**

I. Introduction

In modern factories, human workers and robots are two major workforces. For safety concerns, the two are normally separated with robots confined in metal cages, which limits the productivity as well as the flexibility of production lines. In recent years, attention has been directed to remove the cages so that human workers and robots may collaborate to create a human-robot co-existing factory [1]. Those robots working in a human-involved environment are called co-robots.

The potential benefits of co-robots are huge and extensive, e.g. they may be placed in human-robot teams in flexible production lines [2] as shown in Fig.1, where robot arms and human workers cooperate in handling workpieces, and automated guided vehicles (AGV) co-inhabit with human workers to facilitate factory logistics [3]. Automotive manufacturers Volkswagen and BMW [4] have took the lead to introduce human-robot cooperation in final assembly lines in 2013.

In the factories of the future, more and more interactions among humans and industrial robots are anticipated to take place as shown in Fig.2. In such environments, safety is one of the biggest concerns [5], which attracts attention from standardization bodies [6], as well as from major robot manufacturers including Kuka, Fanuc, Nachi, Yaskawa, Adept and ABB [7]. However, most of these researches are focused on intrinsic safety, i.e. safety in mechanical design [8], actuation [9] and low level motion control [10]. Safety during social interactions with humans, which are key to intelligence (including perception, cognition and high level motion planning and control), still needs to be explored.

*This work was supported by the Berkeley fellowship awarded to Changliu Liu and by FANUC Corporation.

C.Liu and M.Tomizuka are with University of California, Berkeley, CA 94720 USA (e-mail: {changliuliu, tomizuka}@berkeley.edu).

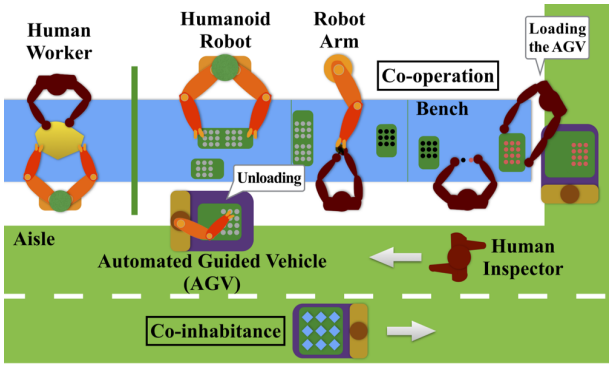


Fig. 1: Flexible production lines in the future, which involve human robot co-operation and co-inhabitance.

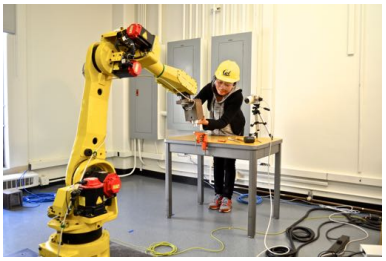


Fig. 2: Human robot interactions

On the other hand, several successful implementations of non-industrial co-robots have been reported, e.g. home assist robots [11] and nursing robots [12]. Complex software architectures are developed to equip the robots with various cognition, learning and motion planning abilities. However, those robots are mostly of human-size or smaller size with slow motion, which may not be cost-efficient for industrial applications. To fully realize a human-robot co-existing factory, the software design methodology for fast co-robots, especially those that are large in size, with multiple links and complicated dynamics, needs to be explored.

In order to make the industrial co-robots human-friendly, they should be equipped with the abilities [13] to: (1) collect environmental data and interpret such data, (2) adapt to different tasks and different environments, and (3) tailor itself to the human workers' needs. The first ability is a perception problem, while the second and third are control problems that are of interest in this paper.

The challenges for control are (i) coping with complex and time-varying human motion, and (ii) assurance of real

time safety without sacrificing efficiency. An constrained optimal control problem is formulated to describe this problem mathematically. And a unique modularized controller architecture will be proposed to solve the problem. The controller architecture is based on two online algorithms proposed by the authors: the safe set algorithm (SSA) [14] and the safe exploration algorithm (SEA) [15], which confine the robot motion in a safe region regarding the predicted human motion. The modularized architecture 1) treats the efficiency goal and the safety goal separately and allows more freedom in designing robot behaviors, 2) is compatible with existing robot motion control algorithms and can deal with complicated robot dynamics, 3) guarantees real time safety, and 4) are good for parallel computation.

The remainder of the paper is organized as follows: in section II, the constrained optimization problem will be described; in section III, the controller architecture in solving the optimization problem will be proposed, together with the design considerations of each module. Case studies with robot arms are performed in section IV. Section V concludes the paper.

II. Algorithmic Safety Measures: The Optimization Problem

As shown in Fig.1, co-robots can co-operate as well as co-inhabit with human workers. In this paper, safety in co-inhabitance and contactless co-operation will be addressed as they form basic interaction types during human robot interactions. Since the interaction is contactless, robots and humans are independent to one another in the sense that the humans' inputs will not affects the robots' dynamics in the open loop. However, humans and robots are coupled together in the closed loop, since they will react to others' motions [14].

A. Problem Formulation

Denote the state of the robot of interest as $x_R \in \mathbb{R}^n$ and the robot's control input as $u_R \in \mathbb{R}^m$ where $n, m \in \mathbb{N}$. Assume the robot dynamics is affine[1] , i.e.

$$\dot{x}_R = f(x_R) + h(x_R)u_R \tag{1}$$

The task or the goal for the robot is denoted as G_R, which can be 1) a settle point in the Cartesian space (e.g. a workpiece the robot needs to get), 2) a settle point in the configuration space (e.g. a posture), 3) a path in the Cartesian space or 4) a trajectory in the configuration space.

The robot should fulfill the aforementioned tasks safely. Let x_H be the state of humans and other moving robots in the system, which are indexed as $H = \{1, 2, \cdots, N\}$. Then the system state is $x = [x_R^T, x_H^T]^T$. Denote the collision free state space as X_S, e.g. $X_S = \{x : d(x_H, x_R) > 0\}$ where d measures the minimum distance among the robot, the humans and all other moving robots. Figure 3 illustrates X_S in blue, which is the off-diagonal area in the system's state space. Given the human configuration, the constraint on the robot's state space $R_S(x_H)$ is a projection from X_S, e.g. $R_S(x_H) = \{x_R : [x_R^T, x_H^T]^T \in X_S\}$, which is time varying with x_H. Hence, two steps are needed to safely control the robot motion: 1) predicting the human motion; and 2) finding the safe region for the robot (green area in Fig.3) based on the prediction.

[1]Any system can have an affine form through dynamic extension. Suppose $\dot{x}_R = F(x_R, u_R)$. Define $x_R^e = [x_R^T, u_R^T]^T$. Let the new control input be $u_R^e = \dot{u}_R$. Then the new system

$$\dot{x}_R^e = \begin{bmatrix} F(x_R^e) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_R^e$$

is affine.

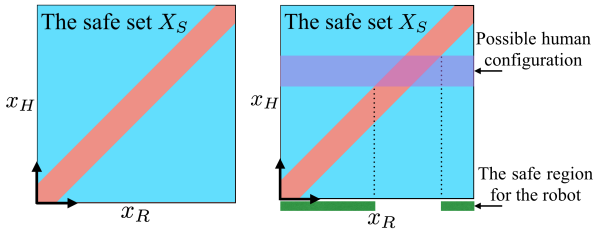


Fig. 3: Illustration of the safe set X_S (blue area) on the system's state space and the safe region on the robot's state space (green area) according to the human configuration.

B. The Optimization Problem

The requirement of the co-robot is to finish the tasks G_R efficiently while staying in the safe region $R_S(x_H)$, which leads to the following optimization problem [16]:

$$\min \quad J(x_R, u_R, G_R) \tag{2}$$
$$s.t. \quad u_R \in \Omega, x_R \in \Gamma, \dot{x}_R = f(x_R) + h(x_R)u_R \tag{3}$$
$$x_R \in R_S(x_H) \tag{4}$$

where J is a goal related cost function to ensure efficiency, Ω is the constraint on control inputs, Γ is the state space constraint (e.g. joint limits, stationary obstacles). The problem is hard to solve since the safety constraint $R_S(x_H)$ is nonlinear, non-convex and time varying with unknown dynamics.

There are numerical methods in solving non-convex optimizations, e.g. sequential convex optimization [17], A* search [18] and Monte-Carlo based rapidly-exploring random trees (RRT) method [19]. However, the computation loads are too high for online applications on industrial co-robots. On the other hand, analytical methods such as potential field methods [20] and sliding mode methods [21] have low computation loads. But they generally do not emphasize optimality. Moreover, the motion patterns of human subjects (or other intelligent robots) are much more complicated than those of general obstacles due to interactions, e.g. x_H may be a function of x_R. To solve the problem, a safe set algorithm (SSA) was proposed to identify the dependency of x_H on x_R online and regulate the control input of the robot in a supervisory loop so as for the system state to stay in the safe set X_S [14]. A safe exploration algorithm (SEA) was built upon SSA to reflect the uncertainties in the prediction of x_H in robot motion control [15]. These two methods

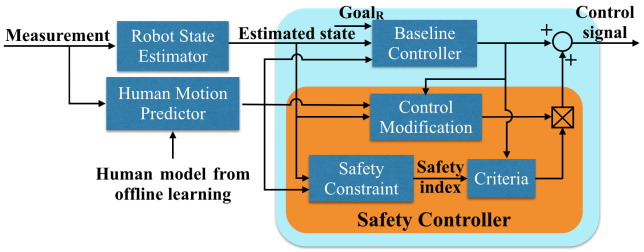


Fig. 4: The controller architecture

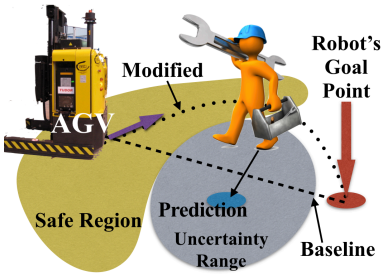


Fig. 5: Illustration of the controller applied on an AGV.

will be generalized and a modularized controller architecture that can handle 3D interactions will be proposed in the next section.

III. Algorithmic Safety Measures: The Controller Architecture

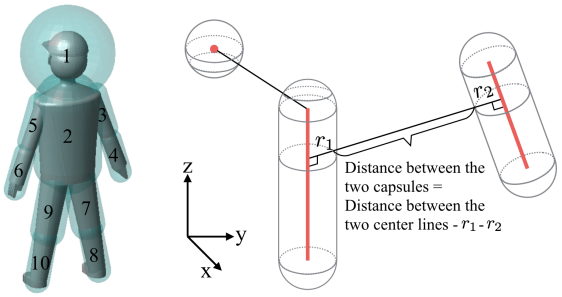
A. The Controller Architecture

The proposed controller will be designed as a parallel combination of a baseline controller and a safety controller as shown in Fig.4. The baseline controller solves (2-3), which is time-invariant and can be solved offline. The safety controller enforces the time varying safety constraint (4), which computes whether the baseline control signal is safe to execute or not (in the "Safety Constraint" and the "Criteria" module) based on the predictions made in the human motion predictor, and what the modification signal should be (in the "Control Modification" module). Each module will be elaborated below.

The expected outcome of this controller structure is shown in Fig.5 on an AGV. In that scenario, the baseline controller will command the AGV to go straight towards its goal. However, the human motion predictor predicts that the human will go to the blue dot and he will be very likely to show up in the gray area. Since the baseline trajectory is no longer in the safe region, the safety controller generates a modified trajectory towards the goal and avoids the human.

B. The Baseline Controller

The baseline controller solves (2-3), which is similar to the controller in use when the robot is working in the cage. The cost function is usually designed to be quadratic which penalizes the error to the goal and the magnitude of the control input, e.g. when G_R is a trajectory, $J = \int_0^T [(x_R - G_R)^T P (x_R - G_R) + u_R^T R u_R] dt$ where P and



(a) The human model (b) The distance between capsules

Fig. 6: The human model and the capsules

R are positive definite matrices. The control policy can be obtained by solving the problem offline. Basic collision avoidance algorithms [17] can be used to avoid stationary obstacles described by the constraint $x_R \in \Gamma$. This controller is included to ensure that the robot can still perform the tasks properly when the safety constraint $R_S(x_H)$ is satisfied.

C. The Human Model and the Human Motion Predictor

In different applications, human body should be represented at various levels of details. For AGVs, mobile robots and planar arms, since the interactions with humans happen in 2D, a human can be tracked as a rigid body in the 2D plane with the state x_H being the position and velocity of the center of mass and the rotation around it. For robot arms that interact with humans in 3D, the choice of the human model depends on his distance to the robot. When the robot arm and the human are far apart, the human should also be treated as one rigid body to simplify the computation. In the close proximity, however, the human's limb movements should be considered. As shown in Fig.6a, the human is modeled as a connection of ten rigid parts: part 1 is the head; part 2 is the trunk; part 3, 4, 5 and 6 are upper limbs; and part 7, 8, 9 and 10 are lower limbs. The joint positions can be tracked using 3D sensors [22]. The human's state x_H can be described by a combination of the states of all rigid parts.

The prediction of future human motion x_H needs to be done in two steps: inference of the human's goal G_H and prediction of the trajectory to the goal. Once the goal is identified (using the method in [23]), a linearized reaction model can be assumed for trajectory prediction [24], e.g.

$$\dot{x}_H = A x_H + B_1 G_H + B_2 x_R + w_H \tag{5}$$

where w_H is the noise, A, B_1 and B_2 are unknown matrix parameters which encode the dependence of future human motion on his current posture, his goal and the robot motion. Those parameters can be identified using parameter identification algorithms [14], while the prediction can be made using the identified parameters. Note that to account for human's time varying behaviors, the parameters should be identified online. This method is based on the assumption that human does not 'change' very fast. Moreover, to reduce the number of unknown parameters, key features that affect human motion can be identified through offline analysis of

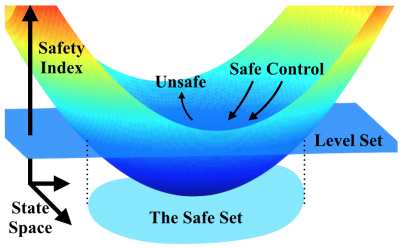


Fig. 7: The safety index over the system's state space.

human behavior. Those low dimension features $\{f_i\}$ can be used in the model (5) to replace the high dimension states x_H and x_R, e.g. $\dot{x}_H = \sum_i a_i f_i + B_1 G_H + w_H$.

D. The Safety Index

The safe set X_S is a collision free subspace in the system's state space, which depends on the relative distance among humans and robots. Since humans and robots have complicated geometric features, simple geometric representations are needed for efficient online distance calculation. Ellipsoids [25] were used previously. However, it's hard to obtain the distance between two ellipsoids analytically. To reduce the computation load, capsules (or spherocylinders) [26], which consists of a cylinder body and two hemisphere ends, are introduced to bound the geometric figures as shown in Fig.6a, Fig.9b and in Fig.12b. A sphere is considered as a generalized capsule with the length of the cylinder being zero. The distance between two capsules can be calculated analytically, which equals to the distance between their center lines minus their radiuses as shown in Fig.6b. In the case of a sphere, the center line reduces to a point. In this way, the relative distance among complicated geometric objects can be calculated just using several skeletons and points. The skeleton representation is also ideal for tracking the human motion.

Given the capsules, the design of X_S is mainly the design of the required minimum distances among the capsules. The design should not be too conservative, while larger buffer volumes are needed to bound critical body parts such as the head and the trunk, as shown in Fig.6a. The safe set in the 3D interactions can be designed as:

$$X_S = \{x : \frac{d(p_{ij}, x_R)}{d_{ij,min}} > 1, \forall i = 1, \cdots, 10, \forall j \in H\} \quad (6)$$

where $d(p_{ij}, x_R)$ measures the minimum distance from the capsule of body part i on the human (or the robot) j to the capsules of the robot R. $d_{ij,min} \in \mathbb{R}^+$ is the designed minimum safe distance. $d_{1j,min}$ should be large since the head is most vulnerable.

To describe the safe set X_S efficiently, a safety index ϕ is introduced, which is a Lyapunov-like function over the system's state space as illustrated in Fig.7. The safety index needs to satisfy three conditions: 1) the relative degree from ϕ to the robot control u_R in the Lie derivative sense is one (to ensure that the robot's control input can drive the state to the safe set directly); 2) ϕ is differentiable almost everywhere; 3) the control strategy that $\dot{\phi} < 0$ if $\phi \geq 0$ will make the set X_S invariant, i.e. $x(t) \in X_S, \forall t > t_0$ if $x(t_0) \in X_S$. Such a safety index can be constructed as shown in [14]. For example, the safety index for the safe set in (6) can be designed as:

$$\phi = 1 + \gamma - (d^*)^c - k_1 \dot{d^*} - \cdots - k_{l-1}(d^*)^{(l-1)} \quad (7)$$

where $d^* = \frac{d(p_{i^*j^*}, x_R)}{d_{i^*j^*,min}}$ and the capsule i^* on the human (or the robot) j^* is the capsule that contains the closest point (the critical point) to the robot R . $l \in \mathbb{N}$ is the relative degree from the function $d(\cdot, x_R)$ to u_R in the Lie derivative sense. In most applications, $l = 2$ since the robot's control input can affect joint acceleration. $c > 1$ is a tunable parameter, while larger c means heavier penalties on small relative distance. $\gamma > 0$ is a safety margin that can uniformly enlarge the capsules in Fig.6a. $k_1, \cdots, k_{l-1}$ are tunable parameters that need to satisfy the condition that all roots of $1 + k_1 s + \cdots + k_{l-1} s^{l-1} = 0$ should be on the negative real axis in the complex plane. The higher order terms of d^* are included to make sure that the robot does not approach the boundary of the safe set in a large velocity, so that the state can always be maintained in the safe set even if there are constraints on the robot control input, e.g. $u_R \in \Omega$.

E. The Criteria

Given the safety index, the criteria module determines whether or not a modification signal should be added to the baseline controller. There are two kinds of criteria: (I) $\phi(t) \geq 0$, or (II) $\phi(t + \Delta t) \geq 0$. The first criterion defines a reactive safety behavior, i.e. the control signal is modified once the safety constraint is violated. The second criterion defines a forward-looking safety behavior, i.e. the safety controller considers whether the safety constraint will be violated Δt time ahead. The prediction in the second criterion is made upon the estimated human dynamics and the baseline control law. In the case when the prediction of future x_H has a distribution, the modification signal should be added when the probability for criteria (II) to happen is non-trivial, e.g. $P(\{\phi(t+\Delta t) \geq 0\}) \geq \epsilon$ for some $\epsilon \in (0, 1)$.

F. The Set of Safe Control and the Control Modification

The set of safe control U_R^S is the equivalent safety constraint on the control space, i.e. the set of control that can drive the system state into the safe set as shown in Fig.7. By construction, the robot can always drive the system state into the safe set through the safety index, i.e. by choosing a control such that $\dot{\phi} < 0$. Since

$$\begin{aligned} \dot{\phi} &= \frac{\partial \phi}{\partial x_R}\dot{x}_R + \frac{\partial \phi}{\partial x_H}\dot{x}_H \\ &= \frac{\partial \phi}{\partial x_R} f(x_R) + \frac{\partial \phi}{\partial x_R} h(x_R) u_R + \frac{\partial \phi}{\partial x_H}\dot{x}_H \end{aligned} \quad (8)$$

then the set of safe control when $\phi \geq 0$ is

$$U_R^S = \{u_R : \frac{\partial \phi}{\partial x_R} h(x_R) u_R \leq -\eta - \frac{\partial \phi}{\partial x_R} f(x_R) - \frac{\partial \phi}{\partial x_H}\dot{x}_H\} \quad (9)$$

where $\eta \in \mathbb{R}^+$ is a margin and $\dot{x}_H$ comes from human motion predictor. When $\dot{x}_H$ has a distribution, let Π be the compact set that contains major probability mass of $\dot{x}_H$, e.g. $P(\{\dot{x}_H \in \Pi\}) \geq 1-\epsilon$ for a small ϵ. Then the inequality in (9) should hold for all $\dot{x}_H \in \Pi$ [15], as illustrated in Fig.3.

The non convex state space constraint $R_S(x_H)$ is then transferred to a linear constraint on the control space in (9). In this way, the modification signal is the optimal value to be added to the baseline control law such that the final control lies in the set of safe control,

$$\Delta u_R = \quad \arg\min_{u_R^o + u \in U_R^S \cap \Omega \cap U_\Gamma} \quad u^T Q u \qquad (10)$$

where $Q \in \mathbb{R}^{m \times m}$ is positive definite which determines a metric on the robot's control space. To obtain optimality, Q should be close enough to the metric imposed by the cost function J in (2), e.g. $Q \approx d^2 J / d u_R^2$ where J is assumed to be convex in u_R. U_Γ is the equivalent constraint on the control space of the state space constraint Γ, which can be constructed following the same procedure of constructing U_R^S. Equation (10) is a convex optimization problem and is easy to solve. In the case that $U_R^S \cap \Omega \cap U_\Gamma$ is empty, a smaller margin η can be chosen so that the feasible control set becomes nonempty.

IV. Case Studies

Simulation studies are performed to evaluate the safety measures on scenarios shown in Fig.1. The cases for AGVs and mobile robots are studied in [23]. In this paper, the interactions among robot arms and humans will be studied. The architecture of the simulation system is shown in Fig.8, which consists of the human loop, the robot loop and the environment. The human subject is in the human loop, who can observe the virtual environment through the screen and whose reaction will be captured by the sensors (e.g. touchpad or Kinect). The human animator reads the tracking data from the sensors and sends the human figure to the environment for display. In the robot loop, the robot animator reads the noisy human data from the environment, computes the safe and efficient trajectory and then sends the real time robot figure to the environment.

A. Planar Robot Arm

The planar robot arm is shown in Fig.9a. Denote the joint angle as $\theta = [\theta_1, \theta_2]^T$. The dynamic equation of the robot arm is $M(\theta)\ddot{\theta} + N(\theta, \dot{\theta}) = \tau_R$ where $M(\cdot)$ is the generalized inertia matrix and $N(\cdot,\cdot)$ is the Coriolis and

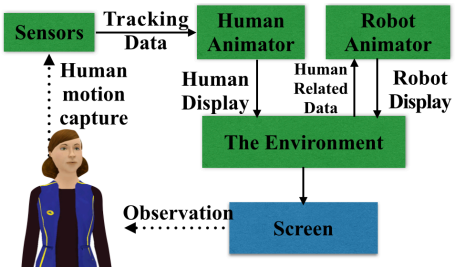
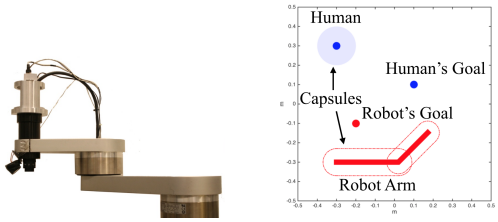


Fig. 8: The simulation system



(a) The planar robot arm (b) The simulation environment

Fig. 9: The planar robot arm and the simulation environment

centrifugal forces [27]. Both functions depend on the robot state $x_R = [\theta^T, \dot{\theta}^T]^T$. $u_R = \tau_R$ is the torque input. The state space equation of the planar robot is affine:

$$\dot{x}_R = \begin{bmatrix} \dot{\theta} \\ -M^{-1}(\theta)N(\theta,\dot{\theta}) \end{bmatrix} + \begin{bmatrix} 0 \\ M^{-1}(\theta) \end{bmatrix} u_R \qquad (11)$$

The simulation environment is shown in Fig.9b where the robot arm is wrapped in two capsules. The vertical displacement of the robot arm is ignored. The human is shown as a blue circle, which is controlled in real time by a human user through a multi-touch pad. Both the human and the robot need to approach their respective goal points in minimum time. New goals will be generated when the old one is approached.

The baseline controller is designed as a computed torque controller with settle point G_R. The safety index is designed as $\phi = D - d^2 - \dot{d}$, where d measures the minimum distance between the human and the robot arm and $D = d_{min}^2(1+\gamma)$. The sampling frequency is $20hz$. Due to the limitation of bandwidth, both reactive and forward-looking criteria are used, in order not to violate the safety constraints between two samples. The set of safe control $U_R^S(k)$ at time k is the intersection of the two sets: $U_1 = \{u_R(k) : \dot{\phi}(k) \leq \eta$ when $\phi(k) \geq 0\}$ and $U_2 = \{u_R(k) : \phi(k+1) < 0\}$. The computation of U_1 follows from (9). The computation of U_2 is similar and is discussed in details in [15]. The metric Q is chosen to be $M(\theta)$, which puts larger penalties on the torque modification applied to heavier link, thus is energy efficient.

The simulation result is shown in Fig.10 and Fig.11. The first plot in Fig.10 shows the critical point on the arm that is the closest to the human capsule. The orange area represents the first link ($y = 0$ is the base) and yellow area represents the second link ($y = 0.55m$ is the endpoint). The second plot shows the distance from the robot endpoint to the robot's goal position. The third plot shows the relative distance d between the robot capsules and the human capsule, while the red area represents the danger zone $\{d < d_{min}\}$. The bars in the fourth plot illustrate whether the safety controller is active (green) or not (white) at each time step. During the simulation, the robot was close to its goal at $k = 55$ and at $k = 110$ before it finally approached it at $k = 220$. However, since the human was too close to the robot in that two cases, going to the goal was dangerous. Then the safety controller went active and the robot arm detoured to avoid

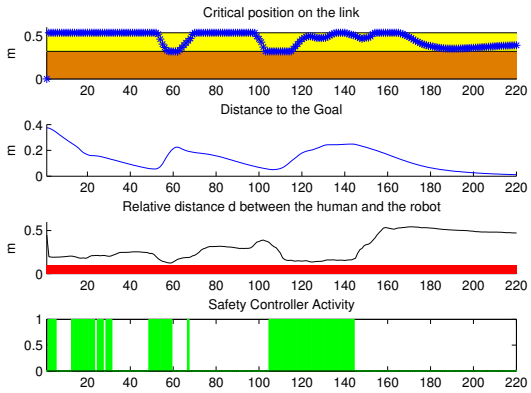
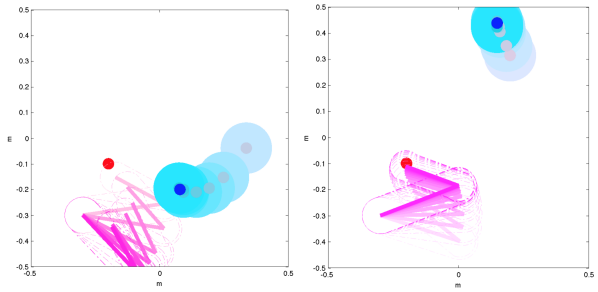


Fig. 10: The simulation profile of the planar robot



(a) $k = 110 : 5 : 140$ (b) $k = 160 : 5 : 220$

Fig. 11: The simulated response of the planar robot

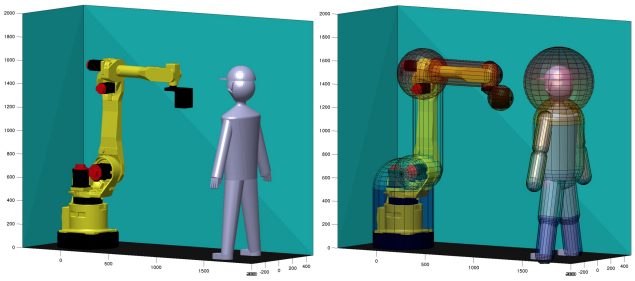
the human. This scenario is also illustrated in Fig.11(a), the 5-times down-sampled snapshots from time step 110 to 140, denote as $k = 110 : 5 : 140$. Lighter color corresponds to smaller k. Due to the safety controller, the relative distance was always maintained above the danger zone. Figure 11(b) shows the snapshots at $k = 160 : 5 : 220$. As the human was far from the robot arm, the safety controller was inactive and the robot finally approached its goal.

B. Six Degree of Freedom Robot Arm

In this case study, the Fanuc M16iB robot arm is used as shown in Fig.2 and the simulation environment is shown in Fig.12a. Capsules are calculated for both the human and the robot as shown in Fig.12b. The radius of the capsules are designed such that one uniform minimum distance requirement $d_{min} = 0.2m$ can be used for all capsules. Denote the robot state as $x_R = [\theta^T, \dot{\theta}^T]^T$ where $\theta = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6]^T$ are the joint angles. $u_R = \ddot{\theta}$ is the joint acceleration. The control modification is done in the kinematic level. A perfect low level tracking controller is assumed. The state space equation of the robot arm is linear:

$$\dot{x}_R = Ax_R + Bu_R \tag{12}$$

where $A_R = \begin{bmatrix} 0_{6\times6} & I_{6\times6} \\ 0_{6\times6} & 0_{6\times6} \end{bmatrix}$ and $B_R = \begin{bmatrix} 0_{6\times6} \\ I_{6\times6} \end{bmatrix}$.



(a) The 3D simulation environment (b) The capsules

Fig. 12: The 3D simulation environment

G_R is to follow a path in the Cartesian space. The baseline controller is a feedback and feedforward controller. The human is moving around the robot arm. The safety index is the same as in the previous case, e.g. $\phi = D - d^2 - \dot{d}$, where d is computed analytically [28]. The sampling frequency is $20hz$. The forward-looking criteria is used. The set of safe control is $U_R^S(k) = \{u_R(k) : \phi(k+1) < 0\}$, and $Q = I$.

The simulation results are shown in Fig.13, Fig.14 and Fig.15. The first plot in Fig.13 shows the critical capsule ID on the robot arm that contains the closest point to the human and the second plot shows the critical capsule ID on the human that contains the closest point to the robot. During interactions, those critical points changed from time to time. The minimum distance between the human and the robot is shown in the third figure, which was maintained above the danger zone during the simulation. The tracking error is shown in the fourth plot. When the human was far from the robot, perfect tracking can be achieved from $k = 100$ to $k = 200$. When the human went close to the robot at $k = 230$, the safety controller took over and moved the robot arm away from the human, at the cost of large tracking error. The snapshots at $k = 230 : 5 : 250$ are shown in Fig.14. Another human avoidance behavior at $k = 310 : 10 : 350$ is shown in Fig.15, with the solid spheres representing the reference path at each time step. The robot stopped tracking the path that moved towards the human by moving backward. In this simulation, the human subject can only control the planar movement of the dummy. The simulation that captures human's whole body movement using Kinect is shown in the video attachment.

The algorithms are run in Matlab (using .m file) on a macbook of 2.3 GHz using Intel Core i7. The running time of the safety controller is shown in Table I. The average running time of the safety controller is 9.5ms, which is dominated by the time in finding the critical points, e.g. calculating the minimum distance between the robot and the human. This is because finding the critical points involves 6×10 distance calculations between capsules. If only the first three joints of the robot arm are considered, e.g. only three robot capsules are used in calculation, the running time is reduced to 5.5ms. If the number of human capsules is reduced to two, the running time for the safety controller is reduced to 2.7ms. Moreover, the running time of the

TABLE I: Running time of the safety controller

Robot arm: degree of freedom	Human model	Running time of the safety controller	Running time in finding critical points
6 DoF	10 capsules	9.5ms	8.8ms
3 DoF	10 capsules	5.5ms	5.0ms
6 DoF	2 capsules	2.7ms	2.0ms
3 DoF	10 spheres	0.77ms	0.40ms

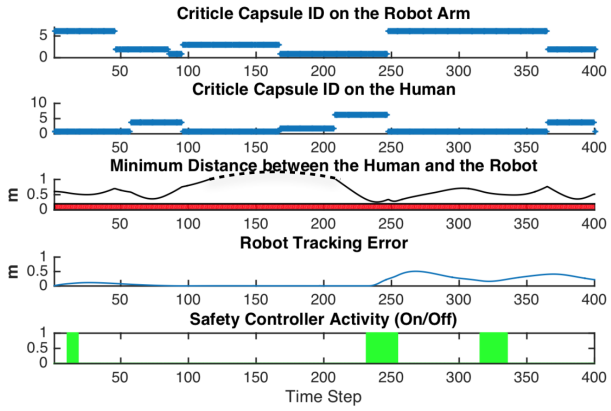


Fig. 13: The simulation profile of the 6DoF robot arm

safety controller is only 0.77ms if the human geometry is represented using spheres. However, the spheres cannot describe the geometry as accurate as the capsules do and may be too conservative. In conclusion, current algorithms can support at least $100Hz$ sampling frequency and the computation time can be further reduced if faster algorithms are developed for distance calculation.

V. Discussions and Conclusions

This paper discussed the algorithmic safety measures for industrial robots working in a human-involved environment. The control problem was posed as a constrained optimal control problem and a unique parallel controller structure was proposed to solve the problem. The control problem was separated into two parts: the efficiency goal with time-invariant constraints and the time-varying safety constraint. The first part was solved by the baseline controller and the safety constraint was enforced by the safety controller. This separation is ideal due to the following reasons:

- There is no need to solve the original problem in a long time horizon, since the uncertainties of the human motion will accumulate. And the safety constraint $R_S(x_H)$ is only active in a small amount of time as evidenced in the simulations. The separation respects different natures of the constraints, by allowing the baseline controller to do long term planning without the time varying constraint and letting the safety controller to do local modification regarding the time varying constraint.
- This separation can also be validated by analytically solving the optimal control problem. Suppose $G_R = \{x_R = 0\}$ and $J = \int_0^T (x_R^T P x_R + u_R^T R u_R) dt$. Let Ω, Γ be the whole space and $R_S(x_H) = \{x_R : \phi(x_R, x_H) < 0\}$. Assume $h(x_R) = B$. Then the Lagrangian [29] of the optimal control problem (2-4) is

$$L = x_R^T P x_R + u_R^T R u_R + \lambda(f(x_R) + B u_R) + \eta\dot{\phi} \quad (13)$$

where λ, η are adjoint variables and $\eta = 0$ if $\phi < 0$. The partial derivatives from L to u_R is $L_u = 2(Ru_R)^T + \lambda B + \eta\phi_{x_R} B$. Setting $L_u = 0$, the optimal control law becomes

$$u_R = -\frac{1}{2} R^{-1} B^T \lambda^T - \frac{1}{2} \eta R^{-1} B^T \phi_{x_R}^T \quad (14)$$

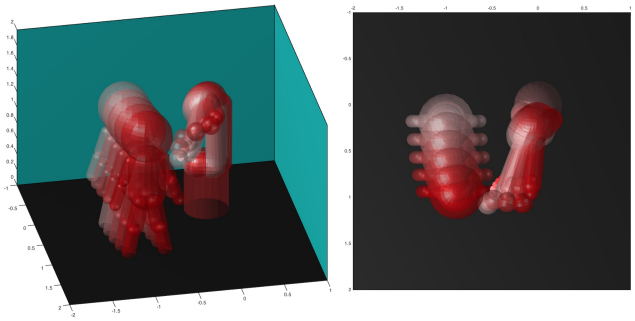
where the first term on the RHS is not related to the safety constraint, which can be viewed as the baseline control law; the second term is concerned with the safety constraint, which is nontrivial only if $\phi \geq 0$, e.g. the safety constraint is violated. Nonetheless, the optimality of this separation will be studied for more complicated problems in the future.

Moreover, the separation offers more freedom in designing the robot behavior and is good for parallel computation.

In conclusion, the controller design procedure is:

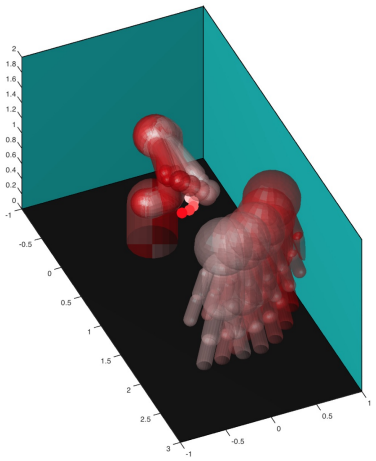
1) Design the baseline controller that can handle the goal and the time invariant constraints.
2) Wrap every moving rigid body with a capsule to simplify the geometry.
3) Design the safe set X_S which specifies the required distance among capsules.
4) Design the safety index ϕ based on the safe set X_S and the robot dynamics.
5) Choose the control modification criteria and design the control modification metric Q.
6) Design the human motion predictor.

To fully realize the scenario in Fig.1, more aspects in the controller design needs to be investigated. For example, as the number of agents in the system increases, the non-convexity of the problem will increase. Methods to avoid local optima need to be developed. Moreover, the safe control method for human robot cooperation that involves contacts also needs to be studied.

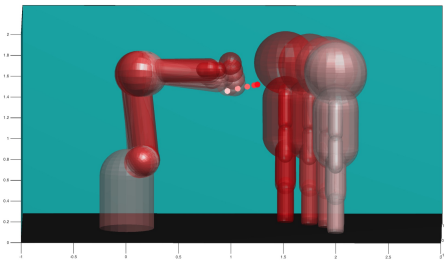


(a) The lateral view (b) The top view

Fig. 14: The simulated response of the 6DoF robot arm: scenario 1



(a) The lateral view



(b) The left view

Fig. 15: The simulated response of the 6DoF robot arm: scenario 2

Nonetheless, the controller structure proposed is of importance as it is a method to handle constraints of different natures and to deal with multiple objectives, whose effectiveness is demonstrated both in simulation and in analysis.

REFERENCES

[1] G. Charalambous, "Human-automation collaboration in manufacturing: Identifying key implementation factors," in *Proceedings of Ergonomics & Human Factors, 2013 International Conference on*. CRC Press, 2013, p. 59.

[2] J. Krüger, T. Lien, and A. Verl, "Cooperation of human and machines in assembly lines," *CIRP Annals-Manufacturing Technology*, vol. 58, no. 2, pp. 628–646, 2009.

[3] G. Ulusoy, F. Sivrikaya-Şerifoğlu, and Ü. Bilge, "A genetic algorithm approach to the simultaneous scheduling of machines and automated guided vehicles," *Computers and Operations Research*, vol. 24, no. 4, pp. 335–351, 1997.

[4] "Working with robots: Our friends electirc," *The Economist*, Sep 2013. [Online].

[5] T. S. Tadele, T. J. d. Vries, and S. Stramigioli, "The safety of domestic robots: a survey of various safety-related publications," *Robotics and Automation Magazine, IEEE*, pp. 134–142, Sep 2014.

[6] C. Harper and G. Virk, "Towards the development of international safety standards for human robot interaction," *International Journal of Social Robotics*, vol. 2, no. 3, pp. 229–234, 2010.

[7] T. M. Anandan. (2014, Sep) Major robot OEMs fast-tracking cobots. [Online].

[8] G. Hirzinger, A. Albu-Schaffer, M. Hahnle, I. Schaefer, and N. Sporer, "On a new generation of torque controlled light-weight robots," in *Proceedings of Robotics and Automation (ICRA), 2001 IEEE International Conference on*, vol. 4. IEEE, 2001, pp. 3356–3363.

[9] M. Zinn, B. Roth, O. Khatib, and J. K. Salisbury, "A new actuation approach for human friendly robot design," *The International Journal of Robotics Research*, vol. 23, no. 4-5, pp. 379–398, 2004.

[10] R. C. Luo, H. B. Huang, C. Yi, and Y. W. Perng, "Adaptive impedance control for safe robot manipulator," in *Proceedings of Intelligent Control and Automation (WCICA), 2011 9th World Congress on*. IEEE, 2011, pp. 1146–1151.

[11] K. Yamazaki, R. Ueda, S. Nozawa, M. Kojima, K. Okada, K. Matsumoto, M. Ishikawa, I. Shimoyama, and M. Inaba, "Home-assistant robot for an aging society," *Proceedings of the IEEE*, vol. 100, no. 8, pp. 2429–2441, 2012.

[12] J. Pineau, M. Montemerlo, M. Pollack, N. Roy, and S. Thrun, "Towards robotic assistants in nursing homes: Challenges and results," *Robotics and Autonomous Systems*, vol. 42, no. 3, pp. 271–281, 2003.

[13] S. Haddadin, M. Suppa, S. Fuchs, T. Bodenmüller, A. Albu-Schäffer, and G. Hirzinger, "Towards the robotic co-worker," in *Robotics Research*, ser. Springer Tracts in Advanced Robotics, C. Pradalier, R. Siegwart, and G. Hirzinger, Eds. Springer Berlin Heidelberg, 2011, vol. 70, pp. 261–282.

[14] C. Liu and M. Tomizuka, "Control in a safe set: Addressing safety in human robot interactions," in *Dynamic Systems and Control Conference (DSCC)*. ASME, 2014, p. V003T42A003.

[15] ——, "Safe exploration: Addressing various uncertainty levels in human robot interactions," in *American Control Conference (ACC)*, 2015, pp. 465 – 470.

[16] N. E. Du Toit and J. W. Burdick, "Robot motion planning in dynamic, uncertain environments," *Robotics, IEEE Transactions on*, vol. 28, no. 1, pp. 101–115, 2012.

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

[18] E. A. Sisbot, L. F. Marin-Urias, X. Broquere, D. Sidobre, and R. Alami, "Synthesizing robot motions adapted to human presence," *International Journal of Social Robotics*, vol. 2, no. 3, pp. 329–343, 2010.

[19] J. J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Proceedings of Robotics and Automation (ICRA), 2000 IEEE International Conference on*, vol. 2. IEEE, 2000, pp. 995–1001.

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

[21] L. Gracia, F. Garelli, and A. Sala, "Reactive sliding-mode algorithm for collision avoidance in robotic systems," *Control Systems Technology, IEEE Transactions on*, vol. 21, no. 6, pp. 2391–2399, 2013.

[22] L. A. Schwarz, A. Mkhitaryan, D. Mateus, and N. Navab, "Human skeleton tracking from depth data using geodesic distances and optical flow," *Image and Vision Computing*, vol. 30, no. 3, pp. 217–226, 2012.

[23] C. Liu and M. Tomizuka, "Modeling and controller design of cooperative robots in workspace sharing human-robot assembly teams," in *Proceedings of Intelligent Robots and Systems (IROS), 2014 IEEE/RSJ International Conference on*. IEEE, 2014, pp. 1386–1391.

[24] W. Zhang, X. Chen, J. Bae, and M. Tomizuka, "Real-time kinematic modeling and prediction of human joint motion in a networked rehabilitation system," in *American Control Conference (ACC)*, 2015, pp. 5800–5805.

[25] C.-S. Tsai, J.-S. Hu, and M. Tomizuka, "Ensuring safety in human-robot coexistence environment," in *Proceedings of Intelligent Robots and Systems (IROS), 2014 IEEE/RSJ International Conference on*. IEEE, 2014, pp. 4191–4196.

[26] V. Macagon and B. Wünsche, "Efficient collision detection for skeletally animated models in interactive environments," in *Proceedings of IVCNZ*, vol. 3. Citeseer, 2003, pp. 378–383.

[27] H. Cheng, "Vision and inertial sensor based drive trains control," Ph.D. dissertation, University of California at Berkeley, 2010.

[28] D. Eberly, *Robust Computation of Distance Between Line Segments*, Geometric Tools, LLC, 2015.

[29] R. F. Hartl, S. P. Sethi, and R. G. Vickson, "A survey of the maximum principles for optimal control problems with state constraints," *SIAM review*, vol. 37, no. 2, pp. 181–218, 1995.