The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Safety provisions for Human/Robot Interactions using Stochastic Discrete Abstractions

Ruslan Asaula and Daniele Fontanelli and Luigi Palopoli

Abstract— **We consider the problem of predicting the probability of an accident in working environments where human operators and robotic manipulators co-operate. We show how, starting from a stochastic discrete time system describing human motion, it is possible to construct a discrete abstraction of the system (a discrete time Markov Chain) to predict the possible trajectories starting from an initial point. The DTMC is used to predict the future evolution for the system, for a fixed horizon, pinpointing the states that, at each step, can be marked as dangerous. This way, the system estimates the probability of an accident and stops the robot when the result is greater than a threshold.**

I. Introduction

The new frontier of industrial robotics is to create production lines where human workers and robotics cells cooperate. One of the main challenges along this way is how to reconcile safety in human/robot interaction with productivity.

The issue has been considered in the past years, both from the perspective of the labor regulation bodies and of scientific research [1], [2]. The strategy typically adopted to avoid accidents is to prevent *any* interaction by physically restricting the access of human operators to the working area of the robots. This extreme measure is frequently required by the current regulations [3], but it severely hinders any practical cooperation between robots and human workers. A different alternative is the massive introduction of sensors on the robot whose purpose is to halt the system as soon as a potential danger is detected. This can be done by proximity sensors [4] or a special sensorized skin that detects dangerous force patterns on the surface of the robot arm [5]. While this approach can be a valid replacement for fencing off the working space of the robot, its application has still a big impact on the productivity of the site. Another possibility is to re-design the structure of the robots to reduce the injuries deriving from an impact with the robot arm [6], [7], but these specially designed machines are not available as yet.

A different approach is to anticipate a a potential danger and take corrective actions as required [8]. This paper takes a step in this direction proposing *an effective way for predicting the probability of an accident*. This challenging goal is attained by the coordinated action of two systems. The first one detects the presence of human targets and tracks their motion across the workspace of the robot. The second system is a real-time prediction engine that computes the probability of an accident in a time horizon, given the current detected position and velocity of the target and the plan of the robot. In this paper we focus on the design of this component.

The starting point of our approach is a dynamic stochastic model that describes the motion of a human target. The model was inspired by the work of Bouthemy [9] and Singer [10] and it is parametric with respect to the probability distributions of the possible human accelerations. In our approach, these distribution are identified on statistical data of the trajectories of the human operators in the considered working environment. To carry out the prediction, the model is approximated by a discrete–time Markov Chain (DTMC). The use of discrete abstraction for dynamic systems has become quite popular in the last few years [11], as an effective means to carry out control synthesis and verification of complex properties. The particular technique that we use in this paper is related to probabilistic model checking [12], which has been effectively applied to the design of fault–tolerant control systems [13] and to similar application. Our work lies in the track opened by Prandini et al. [14], where the authors show how to construct a DTMC to predict the probability of collision between two aircraft flying at the same altitude. This collision detection mechanism has been extended to mobile robot applications [15]. In this work, we extend this approach to the prediction of collisions between human operators and manipulators, showing a very efficient algorithm, which can be executed in real-time.

II. Problem Description and solution overview

The specific setup considered in this paper is described in Figure 1. A robot arm executes working tasks moving along predefined trajectories with a very high precision. This is not actually a loss of generality, since our model is able to consider imprecise motion as long as the deviation can be characterized stochastically. From the robot encoder readings and the a-priori notion of the robot trajectories, our algorithm reconstructs the posture on the plane of the factory floor.

A stereo vision system overlooking the robotic cell is used to measure positions, velocities and accelerations of human workers w.r.t. the robot base frame. The information on the current position and velocity of the target is fed into the predictor that uses a DTMC based abstraction to predict the probability of a collision in a time horizon. If such probability is above a predefined safety threshold, the robot is halted. Clearly, the result is affected by the current posture of the robot and by the future plan of its motion. Therefore, the accuracy of this information is crucial for the precision of

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7 under grant agreement n° IST-2008-224428 "CHAT - Control of Heterogeneous Automation Systems"

Authors are with Department of Information Engineering and Computer Science (DISI), University of Trento, Via Sommarive 14, Trento, Italy {palopoli,fontanelli}@disi.unitn.it

978-1-4244-6676-4/10/$25.00 ©2010 IEEE

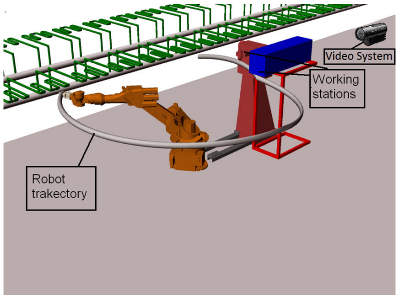


Fig. 1. The robot cell with the vision system and the robot end–effector trajectory.

the computation. To increase the robustness of the prediction w.r.t. unpredicted events (e.g., working time may vary due to unpredictable flaws on the material to be handled, or the pick position can change in time due to a different spacing of the materials on the conveyor), we use the vision system to improve the evaluation of the posture of the robot by fusing visual information with data from the robot encoders.

This scenario has been motivational for the development of this work, but we believe that the methodology can be applied to generic robots and automation tasks.

III. Human Motion Model

The motion of workers in a factory usually follows predefined patterns determined by the working activity. In this paper, we use a kinematic model for the prediction inspired by previous work [9], [10]. The parameters of the model are estimated from a data set collected by the camera in the particular context of the application.

a) Kinematics of human motion: The measured positions of the human worker $s_p(\Delta t) = [x(\Delta t)\, y(\Delta t)]^T$ are referred to a fixed Cartesian reference frame (the coordinates of a point on the plane of motion). Since the vision system is not able to detect the orientation of the human body, we choose to model the human as a moving point. In the recent literature [16] the human body has been described as a nonholonomic platform in which orientation plays a role, but our solution is more general in that it permits sideways motions of the body (incompatible with a nonholonomic constraint). More in depth, denoting with t_0 the time instant in which human positions are firstly measured, velocities $s_v(\Delta t) = [v_x(\Delta t)\, v_y(\Delta t)]^T$ and accelerations $s_a(\Delta t) = [a_x(\Delta t)\, a_y(\Delta t)]^T$ are derived as follows

$$\begin{cases} s_v(t_0 + k\Delta t) = \frac{s_p(t_0+k\Delta t) - s_p(t_0+(k-1)\Delta t)}{\Delta t} \\ s_a(t_0 + k\Delta t) = \frac{s_v(t_0+k\Delta t) - s_v(t_0+(k-1)\Delta t)}{\Delta t} \end{cases}, \qquad (1)$$

where $k \in \mathbb{N}$. For notational simplicity, the explicit reference to the sampling time Δt will henceforth be dropped.

As customary in the literature [17] on human kinematics, the accelerations are referred to a frame attached to the human body. Therefore, $a_=$ represent the tangential acceleration, while $a_\perp$ is the orthogonal acceleration ($\overline{s}_a(k) = [a_=(k)\, a_\perp(k)]^T$). Figure 2 depicts the described quantities in terms of the fixed and moving reference frames. Recalling the material point approximation of the human, the orientation

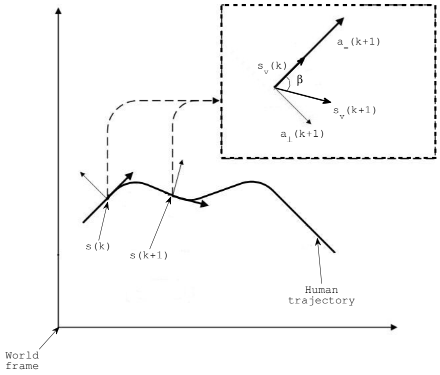
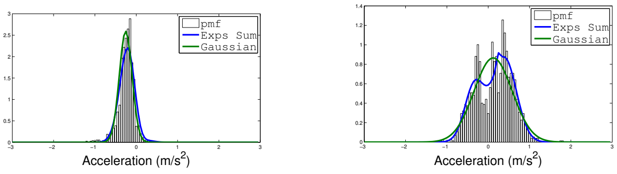


Fig. 2. Fixed frame and moving frames along the human trajectory.



(a)

(b)

Fig. 3. Experimental pmfs with the Gaussian and the sum of exponentials approximations for a particular dataset.

of the moving frame is assumed given by the current velocity direction. Hence $\overline{s}_a(k+1) = R(k)^T s_a(k+1)$, where

$$R(k) = \frac{1}{\|s_v(k)\|} \begin{bmatrix} v_x(k) & -v_y(k) \\ v_y(k) & v_x(k) \end{bmatrix} \qquad (2)$$

is a rotation matrix.

b) Stochastic Description of Accelerations: To predict the possible configuration of the human, a stochastic model for the accelerations is needed. We reasonably assume that accelerations at a certain time step $k+1$ are a function of the current velocity and acceleration, i.e., $\overline{s}_a(k+1) = f(\|s_v(k)\|, \overline{s}_a(k))$. For the purposes of prediction, $\overline{s}_a(k)$ are random variables associated with the stochastic process $\overline{\mathcal{S}} = \{\overline{s}_a(k) : k \in \mathbb{N}\}$.

The stochastic description of $\overline{\mathcal{S}}$ is derived starting from a set of data. In particular, for all the measured pairs $(\|s_v(k)\|, \overline{s}_a(k))$, we construct the histograms for the tangential and orthogonal accelerations $\overline{s}_a(k+1)$ and, hence, a Probability Mass Function (pmf) (see Figure 3).

In order to filter out noise and to smooth out the histograms derived from raw data, the pmfs are approximated by continuous Probability Density Functions (pdf). This step can be done in different ways. A first possibility is to use a single Gaussian for pdf approximations, changing the mean and the standard deviation as a function of $(\|s_v(k)\|, \overline{s}_a(k))$ [9], [10]. Although the result can be quite accurate in some cases (see Figure 3-(a)), the visual inspection of Figure 3-(b) evidently shows that such an approximation may also be very rough when the experimental distribution is multi-modal and/or asymmetric. To address this problem, a possible approach is the approximation by a sum of three exponential

Error	Figure 3-(a)	Figure 3-(b)
Single Gaussian		
Kolmogorov-Smirnov	0.0678	0.4811
Kullback-Leibler	1.2943	0.8758
Sum of Exponentials		
Kolmogorov-Smirnov	0.5596	1.1404
Kullback-Leibler	0.5024	0.2878

TABLE I
COMPARISON OF PDFS ERRORS.

terms, similar to Gaussian kernels. More precisely, each term is of the form $e^{-(\frac{x-m}{c})^2}$, where the values of m and c depend on $(\|s_v(k)\|, \overline{s}_a(k))$ (see [18]). The derived pdfs (for orthogonal and tangential accelerations) are given by

$$\frac{\sum_i W_\star^i(\|s_v(k)\|, \overline{s}_a(k)) B_\star^i(\|s_v(k)\|, \overline{s}_a(k))}{\sum_i W_\star^i(\|s_v(k)\|, \overline{s}_a(k)) b_\star^i(\|s_v(k)\|, \overline{s}_a(k))},$$

where $W_\star^i(\cdot)$ are weight functions (specialized for tangential and orthogonal accelerations), $B_\star^i(\cdot)$ are the exponential terms, and $b_\star^i(\cdot)$ are normalization functions, i.e., integrations of $B_\star^i(\cdot)$ on the region of interest. In Figure 3-(b) we observe a much closer resemblance of the pdf with the data set.

Generally speaking, pdfs with multiple exponential kernels are able to describe multimodal and asymmetric densities. To make a numerical comparison with the approximation with a single Gaussian, we adopt the metric defined by the Kolmogorov-Smirnov hypothesis testing [19]. In particular, we compute the distance according to the Kolmogorov-Smirnov metric between the estimated pdfs and the original collected data (the larger is the value, the better is the approximation). In both cases reported in Figure 3, the estimates with the three kernels produce better metric values. An additional evidence is offered by the computation of the Kullback-Leibler divergence [20], for which the larger is the divergence, the worse is the approximation. Results are summarized in Table I.

c) The Complete Model: Because the acceleration at step $k+1$ depends on the acceleration at step k, we need to include the acceleration in the system state. Therefore, the system state comprises six variables, $s \in \mathbb{R}^6$, $s = [s_p\, s_v\, s_a]^T$. Assuming that the inputs to the motion model are given by $\overline{s}_a(k)$, for which a stochastic description is given, the system equations derived from (1) and (2) are given by

$$s(k+1) = As(k) + B(s_v(k))U(k), \tag{3}$$

where

$$A = \begin{bmatrix} I_2 & \Delta t I_2 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \text{ and } B(s_v(k)) = \begin{bmatrix} \Delta t^2 R(k) \\ \Delta t R(k) \\ R(k) \end{bmatrix}. \tag{4}$$

where $U(k)$ is a two dimensional random variable distributed according to pdfs $g_=(\alpha_=, \|s_v(k)\|, \overline{s}_a(k))$ and $g_\perp(\alpha_\perp, \|s_v(k)\|, \overline{s}_a(k))$, computed using the previously described approximation. I_2 is the 2×2 identity matrix. It has to be noted that the accelerations $s_a(k)$ are needed in (4) for determining the input value $U(k)$. Finally, notice that the model here derived is discrete time in nature, since it is related to the sampling time of the available tracking system.

IV. DISCRETE STOCHASTIC ABSTRACTION

In order to efficiently compute the probability of a collision between a human target (that moves according to a stochastic model), we use a technique called stochastic abstraction [21], [22]. We first discretize the state space reachable by the human choosing a grid of suitable size. Then we construct a DTMC in which each state is associated with the presence of the target in a cell of the grid. The rationale behind this approach is to predict (discretized) future positions, velocities and accelerations of the human worker, up to a certain confidence, using the model in Equation (3).

a) Gridding the Space: Denoting with Δx and Δy the grid position distances and with $\Delta v_x = \Delta x/\Delta t$ ($\Delta v_y = \Delta y/\Delta t$) and $\Delta a_x = \Delta x/\Delta t^2$ ($\Delta a_y = \Delta y/\Delta t^2$) the velocity and acceleration distances between grid intersections respectively, the discrete state vector $S(k) = [S_p(k)^T, S_v(k)^T, S_a(k)^T]^T$ is defined, where $S_p(k) = [X(k), Y(k)]^T$, $S_v(k) = [V_x(k), V_y(k)]^T$ and $S_a(k) = [A_x(k), A_y(k)]^T$. The discrete position $S_p(k)$ is associated to the range of coordinates:

$$x(k) \in \left(X(k) \pm \frac{\Delta x}{2}\right),\; y(k) \in \left(Y(k) \pm \frac{\Delta y}{2}\right). \tag{5}$$

We proceed in a similar way for velocity and acceleration.

b) Transition Probability Matrix: The main problem here is to derive a stochastic model that, given current human configuration discretized to the state $S(k)$, is able to predict $S(k+1)$ according to the stochastic model in (3). The discretized space reduces the number of possible future positions. More precisely, let $\mathcal{S}_p = \{S_{p_i} | i \in \{1, \ldots, n_p\}\}$ be the set of all the positions reachable in the environment of interest (usually the sensing range of the sensor). Similarly, let $n_v = \#\mathcal{S}_v$ and $n_a = \#\mathcal{S}_a$ be respectively the total number of possible velocities S_v and accelerations S_a between grid intersections. The possible configurations of the human are then given by the set $\mathcal{S} = \mathcal{S}_p \times \mathcal{S}_v \times \mathcal{S}_a$.

It is now possible to describe the human dynamic considering each state of $\mathcal{S}$ as a state of a DTMC and encoding its stochastic variation in the transition probabilities. More precisely, let $\sigma(k)$ be a finite-state discrete-time Markov chain taking values in the finite state space $\mathcal{S}$, with transition probability matrix $P = (p_{ij})_{n \times n}$, $p_{ij} \triangleq \Pr\{\sigma(k+1) = S^j \mid \sigma(k) = S^i\}$ (S^i and S^j are two configurations of the human in $\mathcal{S}$), and with initial probability measure $\pi_\sigma(0)$. The evolution of the probability distribution $\pi_\sigma(k)$ of the process σ at time k is given by

$$\pi_\sigma(k+1) = \pi_\sigma(k)P. \tag{6}$$

The probability to be in the state $S^i \in \mathcal{S}$ at a certain time instant k is then given by the i–th entry of $\pi_\sigma(k)$.

The transition probability matrix P is derived by the stochastic description given previously. To this end, let $S^i = [S_p^i\, S_v^i\, S_a^i]^T$ and $S^j = [S_p^j\, S_v^j\, S_a^j]^T$ be the configurations for two states of the DTMC. Notice that, the transition between S^i and S^j ($S^i \rightarrow S^j$) imposes six constraints on the continuous random variables $a_=$ and $a_\perp$. Let $\mathcal{P}_{i,j}$

represent the set of orthogonal and normal accelerations that drive the position from the grid cell S_p^i to the grid cell S_p^i. Likewise, let $\mathcal{V}_{i,j}$ and $\mathcal{A}_{i,j}$ respectively denote the sets that drive the velocity from the grid cell S_v^i to the cell S_v^j and the acceleration from the grid cell S_a^i to the cell S_a^j. In view of model (3), such sets are given by:

$$\begin{aligned}
\mathcal{P}_{i,j} &= \left\{ \overline{s}_a^j = R(S_v^i)^T \left(\frac{s_p^{j\star} - S_p^i}{\Delta t^2} - \frac{S_v^i}{\Delta t} \right) \right\} \\
\mathcal{V}_{i,j} &= \left\{ \overline{s}_a^j = R(S_v^i)^T \left(\frac{s_v^{j\star} - S_v^i}{\Delta t} \right) \right\} \\
\mathcal{A}_{i,j} &= \left\{ \overline{s}_a^j = R(S_v^i)^T s_a^{j\star} \right\}
\end{aligned}$$

where $s_p^{j\star}$ ranges in the two–dimensional region defined in continuous space $\left\{ (X(k) \pm \frac{\Delta x}{2}), (Y(k) \pm \frac{\Delta y}{2}) \right\}$. In a similar way, $s_v^{j\star}$ and $s_a^{j\star}$ are two two–dimensional regions for velocity and acceleration respectively. Therefore, the transition $S^i \to S^j$ requires the accelerations to be in the set $\mathcal{I} = \mathcal{P}_{i,j} \cap \mathcal{V}_{i,j} \cap \mathcal{A}_{i,j}$. By $\mathcal{I}^{=}$ we denote the projection of $\mathcal{I}$ on the space of tangential acceleration and by $\mathcal{I}^{\perp}$ the projection on the set of normal accelerations. Using this notation, the transition probability $p_{i,j}$ associated to $S^i \to S^j$ is then computed as:

$$\begin{aligned}
& p_{ij} = \Pr\{\sigma(k+1) = S^j \mid \sigma(k) = S^i\} = \\
& \int_{\mathcal{I}^{=}} g_{=}(\alpha_{=}, \|S_v^i\|, R(S_v^i)^T S_a^i) d\alpha_{=} \\
& \int_{\mathcal{I}^{\perp}} g_{\perp}(\alpha_{\perp}, \|S_v^i\|, R(S_v^i)^T S_a^i) d\alpha_{\perp}.
\end{aligned}$$

c) Accuracy of the DTMC abstraction: The grid choice has a strong impact on the accuracy of the DTMC abstraction (i.e., on the convergence in probability of the discrete model to the continuous one). In this paper, we have followed the approach suggested by Prandini et al. [14] and come up with a numerically evaluated grid choice of $\Delta x = \Delta y = 1$ cm, given $\Delta t = 0.33$ s.

A validation of this choice was carried out by comparing the probability computed using the DTMC with that resulting from a Montecarlo simulations. In particular, we considered as a benchmark the probability of reaching a state from another after Δt. The Montecarlo simulation required 10^8 iterations to stabilize the result, which was the same to an error of 10^{-5}. Interestingly, the time required to carry out the computation in the two cases was different by an order of magnitude, with the DTMC being the faster.

V. Algorithm

After constructing a DTMC that abstracts the system model, we can use it to determine the probability of an accident in a given time horizon. A possible way to do this is by using a model checker for stochastic system identifying by a logic formula the states to be marked as "bad", i.e., *bad states*. This method allows us to specify a wide range of error conditions. In this paper, we restrict a simple verification task (collision), for which we developed an efficient checker that can be used in real-time. A *collision* is (for the purposes of this paper) associated with an impact between the robot and a target. An impact occurs if: 1) the position of the target (current or predicted) lies in a space occupied by the robot, 2) the relative velocity between the two is above a threshold (contacts with moderate relative velocities can be permitted).

In the discrete abstraction, detecting a collision amounts to finding intersection between the grid cell occupied by the human and the grid cells occupied by the robot. Hence, the impact condition can be dynamically associated with some of states of the DTMC, which encode position, velocity and acceleration of the human. Since a human is here modelled as a single grid cell, the actual shape of the person is modelled by increasing the dimension of the robot. The determination of the grid cells occupied by the robot is done adapting the Bresenham algorithm [23], commonly used to draw a straight line using "quantized" pixel.

a) Computing the probability of collision: Within the framework of DTMC, each state has an associated probability at time step k expressed in $\pi_\sigma(k)$, which evolves according to (6). The collision condition is described with a bad state, in which the DTMC is forced to switch whenever a collision is detected. In order to keep track of a collision, the bad state is an "absorbing" one, i.e., it has a self transition marked with probability 1. In such a way, the bad state collects the probability of collision in time. To increase the performance of the proposed safety system, we also define a similar absorbing state, a *safe* state, that collects the states with zero probability of collision during the prediction horizon. This way, the prediction of these states is no more propagated. Hence, $\pi_\sigma^\star(k) = [\pi_\sigma(k)^T,\ \pi_{\text{bad}}(k),\ \pi_{\text{safe}}(k)]^T$, with associated transition probability matrix $P^\star$, that is the matrix P of (6) "hemmed" with two additional columns and rows of zeros, plus an identity matrix, i.e.,

$$P^\star = \begin{bmatrix} P & 0 \\ 0 & I_2 \end{bmatrix}.$$

The additional entries correspond to the absorbing bad and safe states. Even though bad states and the associated predicates change in time, $P^\star$ is time invariant by construction.

Let S_i be the initial state of the prediction at a certain time $k\Delta t$ and consider a prediction horizon of m steps. In practice, $m = \left\lceil \frac{T_s}{\Delta t} \right\rceil$, where T_s is the time needed to stop the robot in the worst case. If the robot motion does not intersect the region spanned by all the possible reachable states from S_i within the time horizon $m\Delta t$ (this region is computed offline for each S_i) , then $\pi_\sigma(k\Delta t) = 0$, $\pi_{\text{bad}}(k\Delta t) = 0$ and $\pi_{\text{safe}}(k\Delta t) = 1$. Conversely, if S_i is already on the position occupied by the robot at time $k\Delta t$, $\pi_\sigma(k\Delta t) = 0$, $\pi_{\text{bad}}(k\Delta t) = 1$ and $\pi_{\text{safe}}(k\Delta t) = 0$.

In all the other cases, the probability $\pi_\sigma^\star(k)$ have to be propagated for m steps. Denoting $\pi_\sigma^i(k)$ as the "i–th entry of the vector $\pi_\sigma(k)$", $\pi_\sigma^\star(k)$ is iteratively computed over the chosen horizon by the following steps:

1) $\pi_\sigma^\star(k+1)^T = \pi_\sigma^\star(k)^T P^\star$;
2) For each $i = 1$ to n:
    a) if $\pi_\sigma^i(k+1) > 0$ and if S_i is a bad state for time instant $(k+1)\Delta t$ then $\pi_{\text{bad}}(k+1) = \pi_{\text{bad}}(k) +$

$\pi^i_\sigma(k+1)$ and $\pi^i_\sigma(k+1) = 0$.

b) if $\pi^i_\sigma(k+1) > 0$ and from S_i any of the reachable states is not a bad state for the m steps horizon, then $\pi_{\text{safe}}(k+1) = \pi_{\text{safe}}(k) + \pi^i_\sigma(k+1)$ and $\pi^i_\sigma(k+1) = 0$.

Since our goal is to stop the robot whenever the probability is above a threshold, we can interrupt the computation whenever the accumulated probability overcomes the threshold.

VI. Simulation results

We developed an ad-hoc software tool in C++ to implement the prediction algorithm, using additional libraries for sparse matrices handling. The software development has been optimized exploiting the information on the known finite time horizon of the prediction and on the human motion model. The predictor was executed on a standard 2.5 GHz processor with 2 GB of memory. The dynamic model for human motion was identified by collecting data with a visual based tracker using a stereo camera. A complete experimental setup is still under construction. For the purposes of this paper, the prediction was executed using data from previously acquired trajectories, while the motion of the robot was simulated considering realistic trajectories, as resulting from the scenario illustrated below. The sampling time that we considered was 0.33s and the time horizon for the prediction is of three steps. Therefore, we predict the events for a second from the current time (which is reasonable for the typical dynamics of a human).

a) The considered Scenario: We considered the following scenario: a six degrees of freedom anthropomorphic arm is used to pick objects (partially crafted shoes) from a conveyor, take it in front of a laser tool used to rough the sole, then take it to a gluing machine and finally take it back to the conveyor. This scenario takes the inspiration from a true application of a shoe manufacturing, for which a close interaction between the robotic manipulator and human workers is essential.

Since we are interested on motions constrained on a plane in order to conservatively consider all the possible impacts with the same level of dangerousness, the plane projection approximation of the six d.o.f. manipulator turns to be a two d.o.f. planar robot, with a revolute joint (the base joint) and a prismatic joint. Indeed, the joint of the torso corresponds to the base revolute joint of the 2 d.o.f. approximation, the shoulder and elbow revolute joints, having the two axis of rotations that are both parallel to plane of motion maps to the prismatic joint of the 2 d.o.f. approximation. Finally, the additional three d.o.f. of the wrist are neglected since only the posture, and not the dexterity, are meaningful for impacts.

The robot task is subdivided in steps, executed at different velocities. For instance, in the first phase the robot picks a shoe from the conveyor. This phase has a random duration (since the distance between the shoes can change) and the motion is slow. In the second phase, the robot moves to the first working station and the motion is fast. The program continues with an alternation of fast and slow motions. Without loss of generality, simulation results assume collisions only during fast motion phases. Due to the robot dynamic, we assumed that the robot needs at most one second to stop.

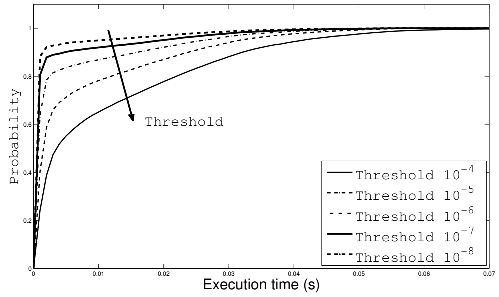


Fig. 4. Cumulative distribution function for the prediction algorithm computation times given different safety thresholds.

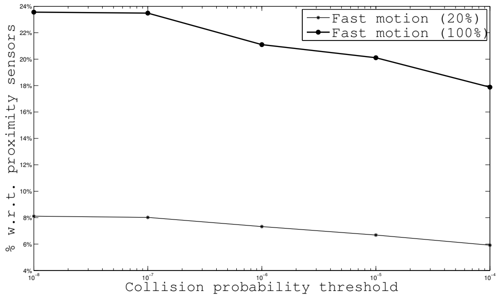


Fig. 5. Desired probability of collision against ratio between the amount of time the robot is halted by proximity sensors and by the prediction algorithm. Multiple curves, related to the percentage of processing time (slow motion) w.r.t the manipulator motion time (fast motion), are shown.

b) Computation time: We collected statistics of the execution times of the predictor (with various input data). Since the computation terminates when the threshold is reached, the computation time depends on the chosen threshold. Therefore, the statistics for different thresholds have been computed and depicted in Figure 4. For reasonable values of the threshold, we find an average computation time in the order of a few millisecond, which is compatible with a real-time execution.

c) Simulations and Comparisons: We simulated a typical working day of the robot. Human operators enter the scene randomly in time and randomly select among a set of pre-computed trajectories (that are obviously unknown to the collision detection algorithm). Each trajectory is assumed to be close to the robot and sometimes to enter into the working area of the robot. We evaluated the frequency the machine is halted, depending on different choices for the probability threshold for which the halting command is emitted. To make a comparison with traditional approaches, we also considered the application of proximity sensor, whose range was chosen equal to the maximum distance travelled by a human in one second (i.e., the time needed by the robot to stop). Figure 5 shows the dependency between the collision probability threshold and the ratio between the amount of time the robot is halted by proximity sensors. When the threshold decreases (i.e., the safety increases) the robot is stopped for a larger amount of time. Furthermore, if the portion of time

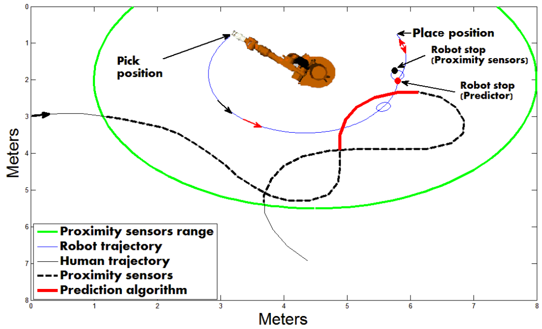


Fig. 6. Example of the manufacturing process. Different thickness and colors of the human trajectory are used for the stop times given by the collision detection algorithm and the proximity sensor. Corresponding stop points on the robot trajectory are also shown. One stop is induced by the prediction algorithm.

in which the robot moves faster (i.e., when it moves between two working stations) increases with respect to the time in which the robot moves slowly (i.e., when it is almost still to carry out the working operations), the advantages of the presented solution are reduced. Nevertheless, the algorithm displays a very good performance with respect to proximity sensors: the efficiency improvement is at least 80%. Again, this number decreases if the robot moves with a higher speed. In this evaluation, we also display how we can strike different tradeoff between risk and productivity by simply tuning the threshold in the probability (the abscissa of Figure 5).

As an example, Figure 6 reports a sample human trajectory chosen among the ones used to derive the curves of Figure 5. The figure depicts the view of the working area from the roof. Circles on the robot end–effector trajectory represent the manufacturing places. Travelled human path is represented with a black thin line. The path is superimposed with a thicker dashed black line for all the points in which the passive safety system is active (hence the robot would be stopped with the standard policy). Thick red and blue lines represent instead the points in which the collision detection algorithm stops the robot. The manipulator moves with high velocity only 20% of the overall cycle time. Robot is uselessly stopped by proximity sensors while it performs the second manufacturing process, while the collision detection algorithm stops the robot only during the returning path from the placing position to the picking one, hence a considerable gain in the robot operational time.

VII. Conclusions

In this paper, we have proposed a methodology for active recognition of dangerous situations in working environment where robotic manipulators and human workers cooperate. The procedure is based on the computation of a stochastic abstraction of human motion (a DTMC), which is used in an algorithm to predict the probability of an accident. Our simulation results, obtained for a 6 d.o.f. manipulator inspired to a real manufacturing scenario, show that for typical tasks the advantage of such a system in terms of productivity could be significant.

As a future work, we plan to study more deeply the theoretical aspects underlying the Markov approximation of the stochastic differential equation describing the human motion and to test our technique in a real robotic system.

References

[1] J. Versace, "A review of the severity index," in *Proc. 15th Stapp Car Crash Conference*, 1971, pp. 771–796.

[2] A. D. Santis, B. Siciliano, A. Luca, and A. Bicchi., "An atlas of physical human-robot interaction," *Mechanism and Machine Theory*, vol. 43, no. 3, pp. 253–270, 2008.

[3] D. of Labour, "Robot safety," 1987.

[4] E. Cheung and V. Lumelsky, "Proximity sensing in robot manipulator motion planning: system and implementation issues," *IEEE transactions on Robotics and Automation*, vol. 5, no. 6, pp. 740–751, 1989.

[5] M. Frigola, A. Casals, and J. Amat, "Human-robot interaction based on a sensitive bumper skin," in *Proc. IEEE/RSJ Intel. Conf. on Intelligent Robots and Systems*, Beijing, China, October 2006, pp. 283–287.

[6] A. Bicchi and G. Tonietti, "Fast and soft arm tactics: Dealing with the safety-performance trade-off in robot arms design and control," *IEEE Robotics and Automation Magazine*, vol. 11, no. 2, pp. 22–33, 2004.

[7] S. Wolf and G. Hirzinger, "A new variable stiffness design: Matching requirements of the next robot generation," in *Proc. IEEE Intl. Conf. on Robotics and Automation*, May 2008, pp. 1741–1746.

[8] D. Kulić and E. Croft, "Pre-collision safety strategies for human-robot interaction," *Autonomous Robots*, vol. 22, no. 2, pp. 149–164, 2007.

[9] Y. Ricquebourg and P. Bouthemy, "Real-time tracking of moving persons by exploiting spatio-temporal image slices," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 797–808, 2000.

[10] R. Singer, "Estimating optimal tracking filter performance for manned maneuvering targets," *IEEE Trans. Aerospace and Electronic Systems*, vol. 6, no. 4, pp. 473–483, July 1970.

[11] G. Pola, A. Girard, and P. Tabuada, "Approximately bisimilar symbolic models for nonlinear control systems," *Automatica*, vol. 44, no. 10, pp. 2508–2516, 2008.

[12] C. Baier, B. Haverkort, H. Hermanns, and J. Katoen, "Model-checking algorithms for continuous-time Markov chains," *IEEE Transactions on Software Engineering*, vol. 29, no. 6, pp. 524–541, June 2003.

[13] M. Kwiatkowska, G. Norman, and D. Parker, "Controller dependability analysis by probabilistic model checking," *Control Engineering Practice*, vol. 15, no. 11, pp. 1427 – 1434, 2007, special Issue on Manufacturing Plant Control: Challenges and Issues - INCOM 2004, 11th IFAC INCOM'04 Symposium on Information Control Problems in Manufacturing.

[14] J. Hu, M. Prandini, and S. Sastry, "Aircraft conflict prediction in the presence of a spatially correlated wind field," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 3, pp. 326–340, 2005.

[15] R. Asaula, D. Fontanelli, and L. Palopoli, "A Probabilistic Methodology for Predicting Injuries to Human Operators in Automated Production lines," in *Proc. IEEE Int. Conf. on Emerging Technologies and Factory Automation (ETFA)*, Mallorca, Spain, 22-26 September 2009, pp. 1–8.

[16] G. Arechavaleta, J. Laumond, H. Hicheur, and A. Berthoz, "On the nonholonomic nature of human locomotion," *Autonomous Robots*, vol. 25, no. 1, pp. 25–35, 2008.

[17] X. Li and V. Jilkov, "Survey of maneuvering target tracking. Part I: Dynamic models," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1333— 1364, 2003.

[18] H. Wang and H. Yue, "A rational spline model approximation and control of output probability density functions for dynamic stochastic systems," *Trans. of the Institute of Measurements and Control*, August 2003.

[19] E. Lehmann and J. Romano, *Testing statistical hypotheses*. Springer Verlag, 2005.

[20] Y. Bar-Shalom, X. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation*. Wiley-Interscience, 2001.

[21] H. Kushner and P. Dupuis, *Numerical Methods for Stochastic Control Problems in Continuous Time*. Sringer Verlag, 2001.

[22] R. Durrett, *Stochastic Calculus*. CRC Press, 1996.

[23] J. E. Bresenham, "Algorithm for computer control of a digital plotter," *IBM Journal of Research and Development*, vol. 4, no. 1, p. 25, 1965.