Real-Time Safety for Human – Robot Interaction

Dana Kulić and Elizabeth A. Croft

Abstract— **This paper presents a strategy for ensuring safety during human-robot interaction in real time. A measure of danger during the interaction is explicitly computed, based on factors affecting the impact force during a potential collision between the human and the robot. This danger index is then used as an input to real-time trajectory generation when the index exceeds a predefined threshold. The danger index is formulated to produce stable motion in the presence of multiple surrounding obstacles. A motion strategy to minimize the danger index is developed for articulated multi degree of freedom robots. Simulations and experiments demonstrate the efficacy of this approach.**

Index Terms—**human-robot interaction, safety, danger index**

I. INTRODUCTION

As robots move from the industrial to home and office and other public environments, safety of humans interacting with robots becomes a key issue [1, 2]. The robot system must provide a mechanism to ensure human safety, given the uncertain environment and untrained users.

To ensure the safety and intuitiveness of the interaction, the complete system must incorporate (i) safe mechanical design, (ii) human friendly interfaces such as natural language interaction and (iii) safe planning and control strategies. Our work focuses on this third item. In particular, the goal is to develop strategies to ensure that unsafe contact does not occur between any point on an articulated robot and a human in the robot's workspace. This paper focuses specifically on the real time safety during human-robot interaction.

A. Related Work

In industrial applications, the safety of human–robot interaction is effected by isolating the robot from the human [1, 3, 4]. In effect there is no interaction. As robots move from isolated industrial environments to interactive environments, this approach is no longer tenable [1]. Three main approaches can be used to mitigate the risk during human-robot interaction: (i) redesign the system to eliminate the hazard, (ii) control the hazard through electronic or physical safeguards, and, (iii) warn the operator/user, either during operation or by training [3]. While the third, warn/train, option has been used in industry, it has not been deemed effective in that setting [3], and is even less suitable for robot interaction with untrained users. Examples of the first option, namely, redesign, include using a whole-body robot visco-elastic covering [5], and the use of spherical and compliant joints [6, 7], and the use of distributed actuation [8, 9].

In unstructured environments, mechanical design alone is not adequate to ensure safe and human friendly interaction. While mechanical measures can be used to decrease the force experienced upon impact, planning and control measures can also be used to avoid impact. Several approaches have been proposed for ensuring safety through control. They focus on either slowing down or stopping when a hazardous situation is identified [7, 10, 11], moving to evade contact [12], or trying to minimize the impact force if contact occurs [13]. A key problem for all of these control methods is to identify when safety is threatened. One approach is to use tactile sensors and force/torque sensors to identify a hazard when unplanned contact occurs [5]. Another approach is to consider every object in the environment, including any humans, as an obstacle, and use a real-time obstacle avoidance strategy, such as [14, 15]. These strategies are all based on considering the distance between the robot and the obstacle as the primary decision factor.

Ikuta et al. [16] developed a danger evaluation method using the potential impact force as an evaluation measure. In their work, the danger index is defined as a product of factors which affect the potential impact force between the robot endpoint and the human, such as relative distance, relative velocity, robot inertia and robot stiffness. The danger index is then proposed as a factor for improved mechanical design and control [16] and end-effector motion planning [17]; although, a real-time control based implementation of the danger index was not presented in those works. The work described herein develops a danger evaluation method, specifically for real-time control application of the entire robot-arm configuration. The danger index is estimated within the control cycle, and is used to generate a real-time motion trajectory to move the robot to a safe location during a potential collision event. Simulations and experiments show the feasibility of our approach.

II. PROPOSED ALGORITHM

The proposed safety system is responsible for moving the robot to a safe location, if a real-time change in the environment threatens the safety of the human in the interaction. The system is responsible for reacting to sudden changes in the environment, not anticipated during the planning stage. The inputs into the safety system consist of the proposed next configuration of the robot provided by the trajectory planner, including the velocity and acceleration information and the current user configuration. Based on this information, the safety system evaluates the current level of danger of the interaction. If the level of danger is low, the proposed plan can proceed, otherwise, a corrective decision is made and an alternate trajectory is generated and passed to the low-level controller.

The alternate trajectory generated by the safety system is designed to lower the danger present in the interaction. Our approach is most similar to the impedance type strategies presented in [14, 15]. A virtual force is calculated by the safety system, pushing the robot away from the person or obstacle. The safety system performs as a one-step-ahead local planner in real-time, moving the robot to a location of lowest danger. The virtual force generated is based on a set of factors affecting the potential impact force during a collision, such as the relative distance and velocity between the robot and the obstacle [15] and the effective impedance at the impact point [14]. However, unlike [14, 15], the algorithm executes in joint space, and is therefore able to perform safely and accurately at all points in the workspace, including any robot singularities.

A. Danger Index Formulation

The key element of the safety system is the estimation of the level of danger, i.e. the *danger index*. The danger index is constructed from measures that have an effect on the potential impact force during a collision. As suggested by Ikuta et al. [12], a "danger measure" should include the distance between the robot and the human, the relative velocity between them, as well as the inertia and stiffness of the robot. However, unlike the method in [12], since the robot is an articulated linkage of bodies, for the real-time control application in this work it is not adequate to consider only the end effector. Instead, the entire robot body must be considered as a potential source of impact. For each link, the point closest to the person is considered, this point is called a *critical point*. The danger index is estimated for each critical point. The factors included are the distance between the robot and the person at the critical point being considered, their relative velocity, and the effective inertia at the critical point. The stiffness can be more effectively lowered through mechanical design [16, 18].

The distance factor f_D is given in equation (1), where, d is the distance from the critical point to the nearest point on the person. The scaling constant k_D is used to scale the distance factor function such that the value of the function is zero when the distance between the human and the robot is large (i.e. larger than D_{max}), and is one when the distance between the human and the robot is the minimum allowable distance (D_{min}).

$$f_D(d) = \begin{cases} k_D\left(\frac{1}{d} - \frac{1}{D_{\max}}\right)^2 & : \; d \le D_{\max} \\ 0 & : \; d > D_{\max} \end{cases}, \tag{1}$$

$$k_D = \left(\frac{D_{\min} \cdot D_{\max}}{D_{\min} - D_{\max}}\right)^2 \tag{2}$$

Values of the distance factor above one indicate an unsafe distance.

The velocity factor f_V is based on the velocity v between the critical point and the nearest point on the person, along the distance vector. The velocity v is considered positive when the robot and the human are moving towards each other.

$$f_V(v) = \begin{cases} k_V\left(v - V_{\min}\right)^2 & : \; v \ge V_{\min} \\ 0 & : \; v < V_{\min} \end{cases}. \tag{3}$$

The scaling constant k_V is used to scale the velocity factor function such that the value of the function is zero when the velocity is lower than V_{min} and one when the velocity is V_{max}. V_{min} is usually set to a negative value (i.e. the robot is moving away from the person).

$$k_V = \left(\frac{1}{V_{\max} - V_{\min}}\right)^2. \tag{4}$$

Values of the velocity factor above one indicate an unsafe velocity.

The inertia factor is defined as:

$$f_I(I_{CP}) = \frac{I_{CP}}{I_{\max}}, \tag{5}$$

where, I_{CP} is the effective inertia at the critical point, and $I_{\max}$ is the maximum safe value of the robot inertia.

The danger index is then the product of the distance, velocity and inertia factors.

$$DI = f_D \cdot f_V \cdot f_I \tag{6}$$

The danger index is formulated as a combination of these dependent factors, similar to [19]. The result is that the danger index is zero through most of the workspace, and is only non-zero when all the conditions for a potential impact are present: small distance, positive velocity towards the person and high robot inertia. This formulation avoids false positives that would be present if a sum of factors was used. Since the danger index is used to activate evasive action, it is important to avoid false positives, which would induce unnecessary evasive action.

Once the danger index is calculated, it is used to generate the virtual force to push the robot away from the human.

B. A one-dimensional example

To visualize the action of the danger index, it is helpful to first consider the one-dimensional case. Consider a point robot moving along a line. Three scenarios are possible: (1) there are no (human) obstacles on either side the robot, (2) there is an obstacle on one side of the robot, or (3) there is an obstacle on both sides of the robot, as shown in Figure 1.

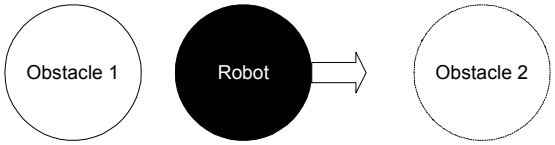

Figure 1 - Point Robot moving between obstacles (one-dimensional case)

Since the safety system always acts to decrease the danger at the highest critical point, only the nearest obstacles are considered; additional obstacles further away from the robot do not affect system behavior. If there are no obstacles on either side of the robot, the robot proceeds as planned, as the danger index is zero. If the obstacle is on one side of the robot only, and the danger index is non-zero, a virtual force (F_{SE}) pushing the robot in the direction away from the obstacle is generated, proportional to the danger index.

$$F_{SE} = Km \cdot DI(d, v) \tag{7}$$

This is analogous to a virtual impedance between the robot and the obstacle, similar to [15]. However, unlike [15], the impedance is non-linear. This is because stronger evasive action is desirable as the danger increases. Figure 2 shows a comparison between the linear impedance (LI) and the proposed danger index (DI) method for the single degree of freedom system. The robot initial configuration is at 0.2m away from the obstacle, with a velocity of 1m/s towards the obstacle. Close to the obstacle, when the danger index is high, the non-linear impedance results in a faster reaction to move away from the obstacle.

If obstacles are present on both sides of the robot, two virtual impedances are present, one between each obstacle and the robot. In this case, the resultant force (F_{BE}) is the difference between the two impedances, based on the danger index calculated with respect to each obstacle.

$$F_{BE} = Km \cdot \left[DI(d_1, v_1) - DI(d_2, v_2) \right]. \tag{8}$$

To illustrate the behavior of the two-obstacle case, consider the simple case when both obstacles are stationary and the robot is moving between them. In this case, $v_1 = -v_2$. The system is then characterized by a second order differential equation:

$$a = \frac{1}{m} \cdot F_{BE}(x, v), \tag{9}$$

where a is the acceleration and m is the mass of the point robot. Figure 3 shows the phase portrait of the system, and Figure 4 shows a sample time trajectory. The system is stable about the equilibrium point d = 0.5, v = 0, at the midpoint between the two obstacles. However, care must be chosen when selecting the danger index parameters, to avoid oscillatory behavior about the equilibrium point. Parameter selection is discussed in Section II.D.

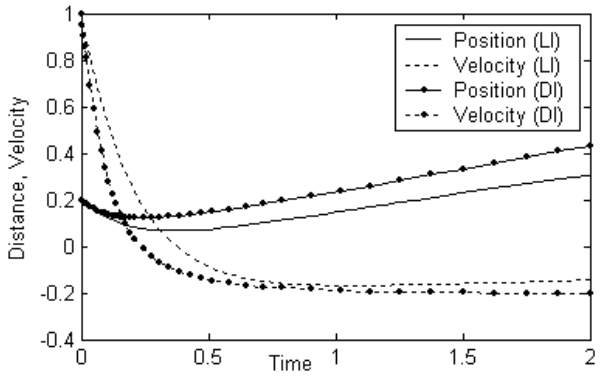

Figure 2 – Comparison between linear impedance and the danger index for a 1DoF system

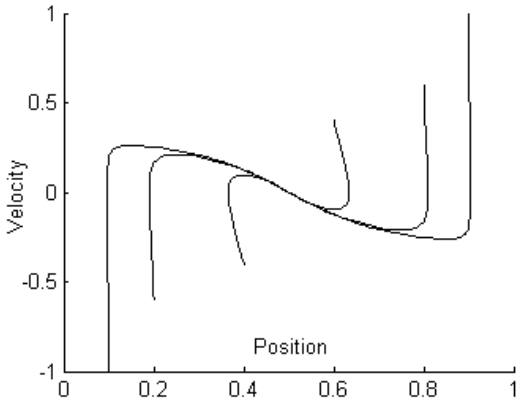

Figure 3 – Phase portrait of the two obstacle case (1DoF)

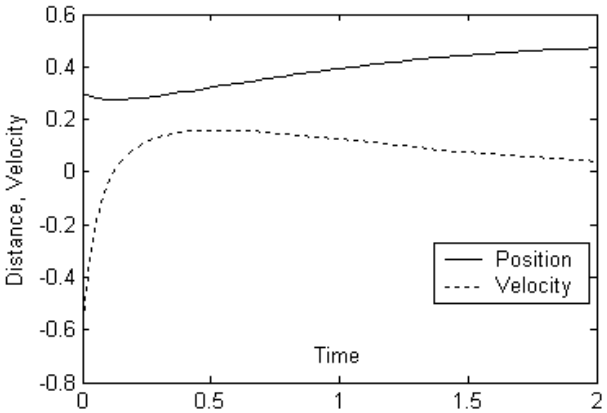

Figure 4 – Sample Time Trajectory for the two obstacle case (1DoF)

C. Full Algorithm

When the algorithm is extended to an articulated robot operating in three-dimensional task-space, each robot joint is evaluated sequentially as a one degree of freedom system. This results in a local sequential planner, where backtracking motion is not considered [20]. Backtracking

motion requires global planning, which cannot be executed in real time. The goal of the safety system is to generate a plan to move the robot to the safest possible location in real time, and then issue a request to the planner to generate a global plan.

The one-dimensional algorithm must be modified when applied to a multi-link manipulator, because the position and velocity of each link is affected by motion of any proximal links. Proximal joints must consider not only critical points at the corresponding link, but all critical points at links distal to the joint. The algorithm proceeds in two steps. First, the danger index at each potential collision locations is calculated. All non-zero locations are stored as critical points. If the danger index at any critical point is above a defined threshold, DI_t, then the planned trajectory is discarded and a new trajectory is generated by the safety system. The desired safe motion for each joint is calculated, starting with the base joint. For each joint, all critical points distal to the joint are considered (i.e. not just critical points at that joint). Analogous to the one dimensional case, three scenarios are possible: (1) there are no critical points distal to this joint, (2) all critical points generate virtual forces in a single direction, or (3) critical points generate forces in opposing directions. If no critical points are present, a virtual damping force (F_d) is applied to stop any motion of that joint (that is, the prior motion assigned by the planned trajectory).

$$F_d = -B\dot{q}, \tag{10}$$

where B is a damping constant and $\dot{q}$ is the measured joint velocity. If all critical points generate virtual forces in a single direction, F_{SE} is applied as specified in Equation (7) above. If opposing critical points are present, F_{BE} is applied as specified in equation (8). For each joint, a new trajectory is then generated based on the desired acceleration and the current position and velocity.

This algorithm is applicable to both non-redundant and redundant manipulator, and scales linearly with the number of robot links.

To calculate the distances and velocities between each robot link and the human (or other obstacles) a *set of spheres* representation is used. This representation is based on the approach described in [21], and generates very fast and conservative distance estimates for 3D distances between the robot and the human.

D. Parameter Selection

Selecting the parameters for the distance and velocity factors will determine the onset and magnitude of the control action in the event that a hazard is detected. In particular, the minimum safe distance (D_{min}) and the maximum safe velocity (V_{max}) determine when the danger index climbs above 1 and control action strongly increases. These should be based on the physical characteristics and capabilities of the robot. D_{min} can be estimated based on the maximum robot deceleration and velocity, while V_{max} can be estimated based on indices of injury severity, such as the Gadd Severity Index or the Head Injury Criterion [18, 22]. The largest distance (D_{max}) and lowest velocity (V_{min}) at which the safety system begins to consider a potential hazard are then selected. The distance D_{max} should be based on the physical size of the robot, and the geometry of the workspace. V_{min} must be smaller than zero to ensure stability. It is desirable to set the ranges between D_{min} and D_{max}, and V_{min} and V_{max} large, so that there is time to react before the danger is imminent. However, making these ranges excessively large prevents the robot from operating in more cluttered environments. Too large a range between V_{min} and V_{max} also reduces the effective damping, which can result in oscillations in the 2nd order system at each joint. False positive reactions of the safety system can be significantly reduced by setting the reaction threshold DI_t above zero.

III. SIMULATIONS

To illustrate the behavior of the safety system, simulations on a 3-DoF planar robot were generated. Each of the robot links is 0.4m long. The obstacles have a radius of 0.2m. Table 1 shows the settings for all the algorithm parameters. In each case, the initial robot trajectory was from [0,0,0] (horizontally stretched out) to [pi/2;0;0] (upright). The robot is animated using the Robotics Toolbox for Matlab [23].

TABLE 1
PARAMETER VALUES FOR SIMULATIONS

Parameter	Value
Dmin	0.4
Dmax	0.8
Vmin	-0.2
Vmax	1
DI_t	0.3

Figure 5 shows the behavior of the safety system in a sample simulation. The higher obstacle is stationary, while the lower obstacle moves 0.5m vertically up starting from rest at the start of the simulation and stopping halfway through. In this case, the safety system is able to generate a trajectory to clear the robot from the obstacles. As can be seen in the figure, the robot stays further away from the lower obstacle, since the lower obstacle is moving towards the robot, therefore the velocity factor at the critical point between that obstacle and the robot will be higher than the velocity factor at the upper obstacle.

Figure 6 shows frames of a case when the safety system is unable to generate an escape trajectory. The upper obstacle blocks the possible escape of joint 1, while the lower obstacle moves upwards. The robot moves between the obstacles, equidistant between the two obstacles in terms of the danger index. While the lower obstacle is moving towards the robot, the robot is closer to the upper obstacle. Once the lower obstacle stops, the robot moves to the middle, between the two obstacles.

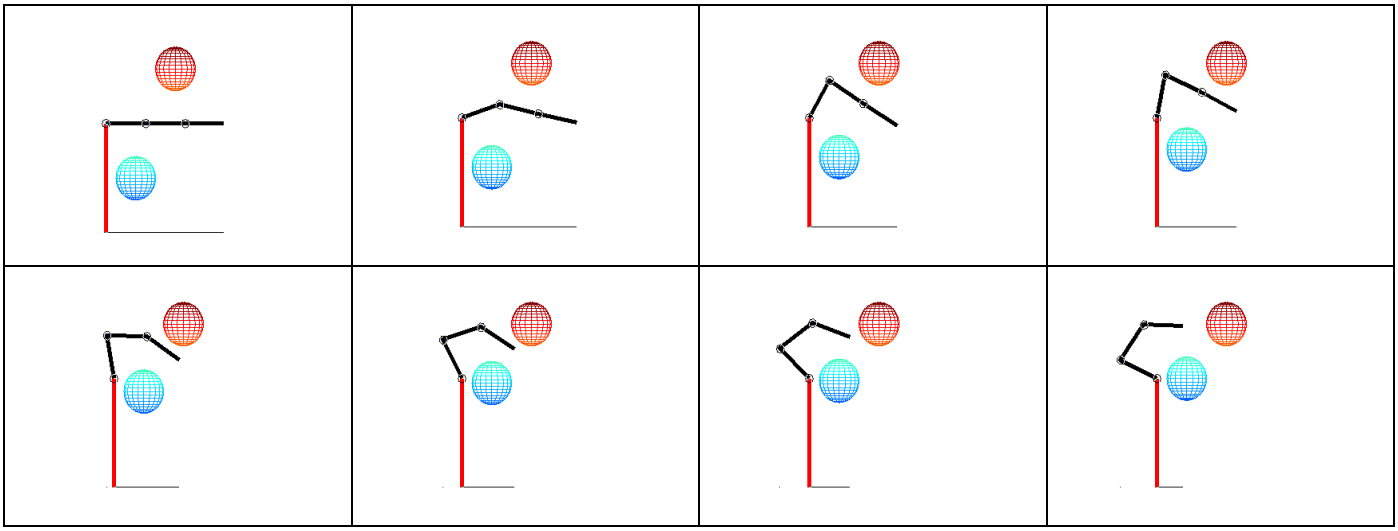

Figure 5 – Planar Robot Simulation (robot clears obstacles)

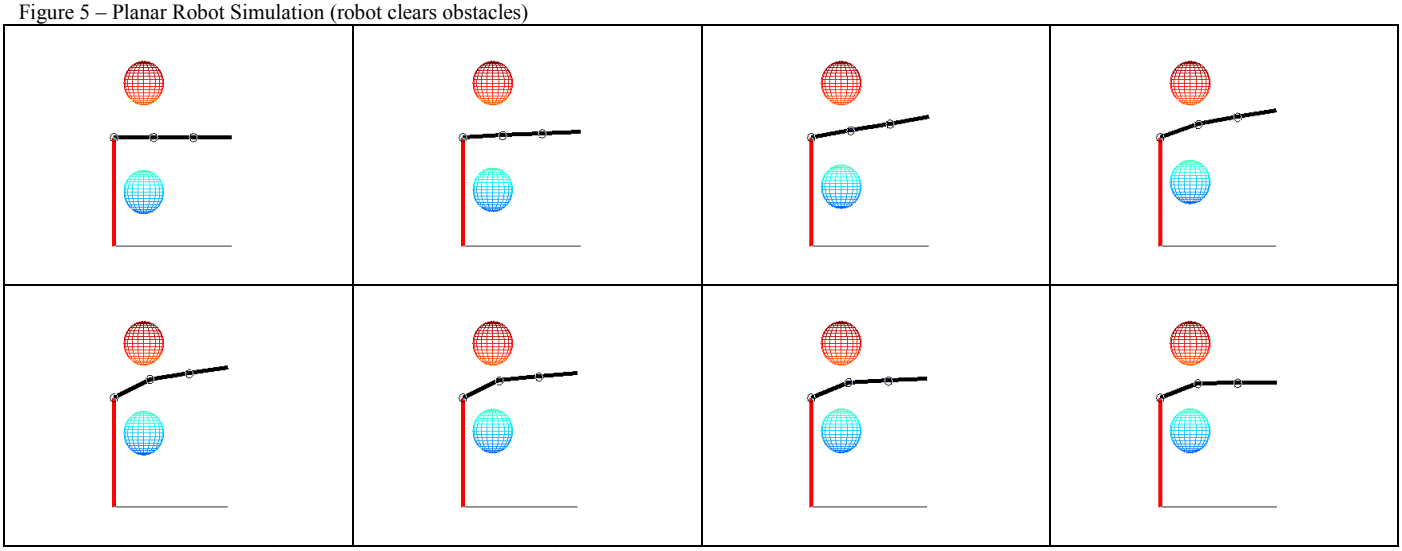

Figure 6 – Planar Robot Simulation (robot cannot clear obstacles)

In this case the safety module generates a 'safest' stable (non-oscillating) real time trajectory, giving time for higher level systems to determine the next most appropriate action.

IV. EXPERIMENTS

The safety system was tested with the CRS A460 6-DoF manipulator. The robot was controlled by an in-house open architecture controller, based on the Quanser MultiQ PCI card and WinCon software [24]. The controller was implemented on a Pentium 4 2.8GHz computer. The low level control was PID. Both the safety system and the low level controller executed at a frequency of 1kHz.

Information about the human location was obtained from a stereo camera mounted in front of the robot base and facing the approximate human location. The camera algorithms ran on a second Pentium 4 2.8GHz computer, linked to the controller via a serial connection. The camera stereo routines were used to extract a depth map of the environment. The depth map was then used to generate the set of spheres which envelop the human location. The set of spheres representation results in a conservative estimate of the space occupied by the human, which means that highly accurate position sensing is not required.

The safety system was tested as a standalone unit, without an on-line planner. A default trajectory was issued to simulate a planned trajectory. The person would then move to block the robot's path, and the safety system would engage to move the robot to a safe location. Table 2 shows the values of the parameters used during the experiments.

TABLE 2
PARAMETER VALUES FOR EXPERIMENTS

Parameter	Value
Dmin	0.6
Dmax	1.0
Vmin	-0.5
Vmax	0.5
DI_t	0.3

Figure 7 shows video frames from a sample experiment. In this case, the initial trajectory moves the robot from the upright position towards the table in front of the human. As the human raises his hands towards the robot, the safety system activates and the robot moves upwards and away from the human.

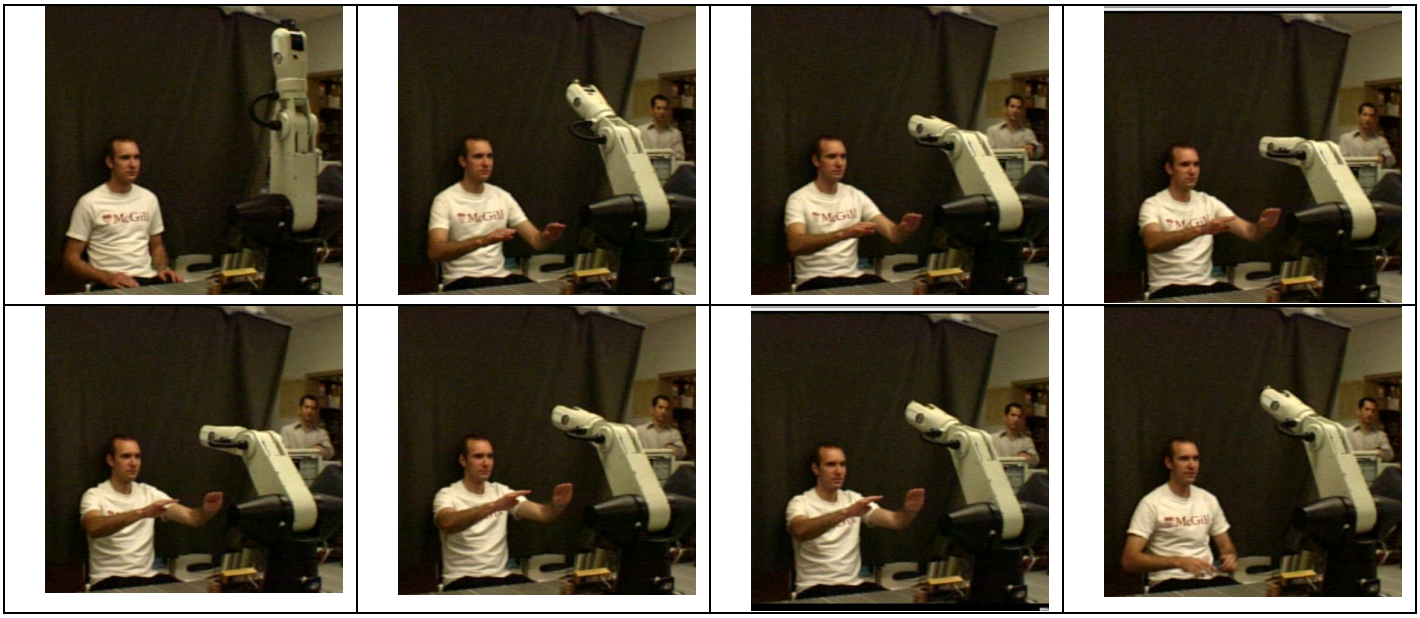

Figure 7 – CRS Robot experiment video frames.

V. CONCLUSIONS

This paper presents a methodology for ensuring human safety during a human-robot interaction in real time. The level of danger in the interaction due to a potential collision is explicitly defined as the danger index. A sequential one-step ahead trajectory planner (the safety system) is presented which generates robot motion by minimizing the danger index. The algorithm can be used for redundant or non-redundant manipulators, and operates correctly at all robot configurations, including singularities.

REFERENCES

[1] P. I. Corke, "Safety of advanced robots in human environments," Discussion Paper for IARP, 1999.

[2] C. W. Lee, Z. Bien, G. Giralt, Peter I. Corke, and M. Kim, "Report on the First IART/IEEE-RAS Joint Workshop: Technical Challenge for Dependable Robots in Human Environments," IART/IEEE-RAS, 2001.

[3] "RIA/ANSI R15.06 - 1999 American National Standard for Industrial Robots and Robot Systems - Safety Requirements." New York: American National Standards Institute, 1999.

[4] S. P. Gaskill and S. R. G. Went, "Safety Issues in Modern Applications of Robots," *Reliability Engineering and System Safety*, vol. 52, pp. 301-307, 1996.

[5] Y. Yamada, Y. Hirawawa, S. Huang, Y. Umetani, and K. Suita, "Human - Robot Contact in the Safeguarding Space," *IEEE/ASME Transactions on Mechatronics*, vol. 2, pp. 230-236, 1997.

[6] Y. Yamada, T. Yamamoto, T. Morizono, and Y. Umetani, "FTA-Based Issues on Securing Human Safety in a Human/Robot Coexistance System," presented at IEEE Systems, Man and Cybernetics SMC'99, pp. 1068-1063, 1999.

[7] Y. Yamada, Y. Hirawawa, S. Huang, Y. Umetani, and K. Suita, "Human - Robot Contact in the Safeguarding Space," *IEEE/ASME Transactions on Mechatronics*, vol. 2, pp. 230-236, 1997.

[8] M. Zinn, O. Khatib, B. Roth, and J. K. Salisbury, "Towards a Human-Centered Intrinsically Safe Robotic Manipulator," IARP-IEEE/RAS Joint Workshop on Technical Challenges for Dependable Robots in Human Environments, Toulouse, France, 2002.

[9] M. Zinn, O. Khatib, and B. Roth, "A new actuation approach for human friendly robot design," presented at IEEE Int. Conf. on Robotics and Automation, New Orleans, USA, pp. 249-254, 2004.

[10] A. J. Bearveldt, "Cooperation between Man and Robot: Interface and Safety," presented at IEEE Int. Workshop on Robot Human Communication, pp. 183-187, 1993.

[11] J. Zurada, A. L. Wright, and J. H. Graham, "A Neuro-Fuzzy Approach for Robot System Safety," *IEEE Transactions on Systems, Man and Cybernetics - Part C: Applications and Reviews*, vol. 31, pp. 49-64, 2001.

[12] V. J. Traver, A. P. del Pobil, and M. Perez-Francisco, "Making Service Robots Human-Safe," presented at IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2000), pp. 696-701, 2000.

[13] J. Y. Lew, Y. T. Jou, and H. Pasic, "Interactive Control of Human/Robot Sharing Same Workspace," presented at IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 535-539, 2000.

[14] O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," *The Int. Journal of Robotics Research*, vol. 5, pp. 90-98, 1986.

[15] T. Tsuji and M. Kaneko, "Noncontact Impedance Control for Redundant Manipulators," *IEEE Transactions on Systems, Man and Cybernetics - A: Systems and Humans*, vol. 29, pp. 184-193, 1999.

[16] K. Ikuta and M. Nokata, "Safety Evaluation Method of Design and Control for Human-Care Robots," *The Int. Journal of Robotics Research*, vol. 22, pp. 281-297, 2003.

[17] M. Nokata, K. Ikuta, and H. Ishii, "Safety-optimizing Method of Human-care Robot Design and Control," presented at Proceedings of the 2002 IEEE Int. Conf. on Robotics and Automation, Washington, DC, pp. 1991-1996, 2002.

[18] A. Bicchi, S. L. Rizzini, and G. Tonietti, "Compliant design for intrinsic safety: General Issues and Preliminary Design," presented at IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 1864-1869, 2001.

[19] D. Kulic and E. Croft, "Safe Planning for Human-Robot Interaction," *Journal of Robotic Systems*, In Press, 2005.

[20] K. K. Gupta and Z. Guo, "Motion Planning for Many Degrees of Freedom: Sequential Search with Backtracking," *IEEe Transactions on Robotics and Automation*, vol. 11, pp. 897-906, 1995.

[21] B. Martinez-Salvador, A. P. del Pobil, and M. Perez-Francisco, "A Hierarchy of Detail for Fast Collision Detection," presented at IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 745-750, 2000.

[22] A. Bicchi and G. Tonietti, "Fast and "Soft-Arm" Tactics," *IEEE Robotics and Automation Magazine*, vol. 11, pp. 22-33, 2004.

[23] P. I. Corke, "A Robotics Toolbox for Matlab," *IEEE Robotics and Automation Magazine*, vol. 3, pp. 24-32, 1996.

[24] http://www.quanser.com/english/html/solutions/fs_Q8.html