

Safe Planning for Human–Robot Interaction

Dana Kulić and Elizabeth A. Croft

*Department of Mechanical Engineering
University of British Columbia
Vancouver, Canada
e-mail: dana@mech.ubc.ca*

Received 1 April 2004; accepted 7 February 2005

This paper presents a strategy for improving the safety of human–robot interaction by minimizing a danger criterion during the planning stage. This strategy is one part of the overall methodology for safe planning and control in human–robot interaction. The focus application is a hand-off task between an articulated robot and an inexperienced human user. Two formulations of the danger criterion are proposed: a criterion assuming independent safety-related factors, and a criterion assuming mutually dependent factors. Simulations of the proposed planning strategy are presented for both 2D and 3D robots. The results indicate that a criterion based on scaled mutually dependent factors such as the robot inertia and the human robot distance generates safe, feasible paths for interaction.

© 2005 Wiley Periodicals, Inc.

1. INTRODUCTION

Robots have been successfully employed in industrial settings to improve productivity and perform dangerous or monotonous tasks. More recently, attention has turned to the use of robots to aid humans outside the industrial environment, in places such as the home or office. For example, as the population in the developed world ages, robots that can interact with humans in a safe and friendly manner while performing necessary homeware/daily living tasks would allow more seniors to maintain their independence. Such devices could alleviate some of the non-medical workload from health-care professionals, and reduce

growing healthcare costs. To achieve such objectives, robotic devices must become more safe and user friendly. Untrained users need to feel comfortable and confident when interacting with a device that, unlike most passive household appliances, is potentially active in its interaction with the user.

Two key issues hampering the entry of robots into unstructured environments populated by humans are safety and dependability.^{1,2} 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.

The design of safe planning and control strategies can be divided into three key components: safe planning, human interaction monitoring, and safe control.³ The human monitoring component of our method deals with monitoring the human participant of the interaction, and estimating the level of attention and approval the human exhibits towards the robot's actions. This aggregate measure of the user's response is the *intent* estimate. Discussion of the human monitoring and control system components can be found in Kulic and Croft.^{3,4} This paper focuses specifically on the planning aspects of the system.

1.1. Related Work

In industrial applications, the safety of human-robot interaction is effected by isolating the robot from the human.^{1,5,6} 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.⁵ While the warn/train option has been used in industry, it had not been deemed effective in that setting,⁵ and is even less suitable for robot interaction with untrained users. Examples of redesign include using a wholebody robot visco-elastic covering, and the use of spherical and compliant joints.^{7,8}

In unstructured environments, mechanical design alone is not adequate to ensure safe and human friendly interaction. Additional safety measures, utilizing system control and planning, are necessary. Several approaches have been proposed for ensuring safety through control. They focus on either slowing down or stopping when a hazardous situation is identified,⁸⁻¹⁰ moving to evade contact,¹¹ or trying to minimize the impact force if contact occurs.¹² 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.¹³ Recently, Ikuta et al.¹⁴ 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 and the human, such as relative distance, relative velocity, robot inertia and robot stiffness.

Motion planning and the a priori identification of

potentially hazardous situations as a means of reducing potential robot-safety hazards has received less attention than control-based (reactive) techniques. However, safe planning is important for any interaction that involves motion in a human environment, especially those that may contain additional obstacles. Application examples include service scenarios such as a dish clearing robot,⁹ services for the disabled, such as approaching the human for a feeding task,^{15,16} and pick and place tasks for picking up and delivering common objects.¹⁷

Several authors consider an *a priori* evaluation of the workspace to determine motion parameters within the various zones of the workspace.^{9,13} Nokata et al.¹⁸ use a danger index based on the distance and velocity between the human and the manipulator end effector to plan a safe path for a planar manipulator. Chen and Zalzal¹⁹ use the distance between the robot and any obstacles as a measure of "safeness" in the cost function for path planning for mobile manipulators. Brock and Khatib²⁰ describe the elastic strips framework for motion planning for highly articulated robots moving in a human environment. Although their paper does not deal explicitly with ensuring safety, they introduce a posture potential to the planner cost function; this additional potential can be used to formulate safety-based constraints.

Including safety criteria at the planning stage can place the robot in a better position to respond to unanticipated safety events. Planning is thus used to improve the control outcome, similar to using smooth trajectory design to improve tracking.^{21,22} Herein, a similar approach to Nokata et al.¹⁸ is considered. However, in order to address safety in unstructured environments, the whole arm configuration of the manipulator, rather than only the end-effector state, is considered in the planning stage. Within this context, potential danger criteria are formulated and evaluated, using a motion planning framework similar to Brock and Khatib.²⁰ Each proposed criterion explicitly considers the manipulator inertia and center of mass location with respect to the human to evaluate danger. A two stage planning approach is proposed to address issues of potentially conflicting planning criteria. The proposed approach is evaluated in simulation to compare the criteria and to demonstrate their efficacy in an example handoff task.

1.2. System Overview

The proposed system architecture assumes a user-directed robot system. The user must initiate each interaction, but the robot has sufficient autonomy to

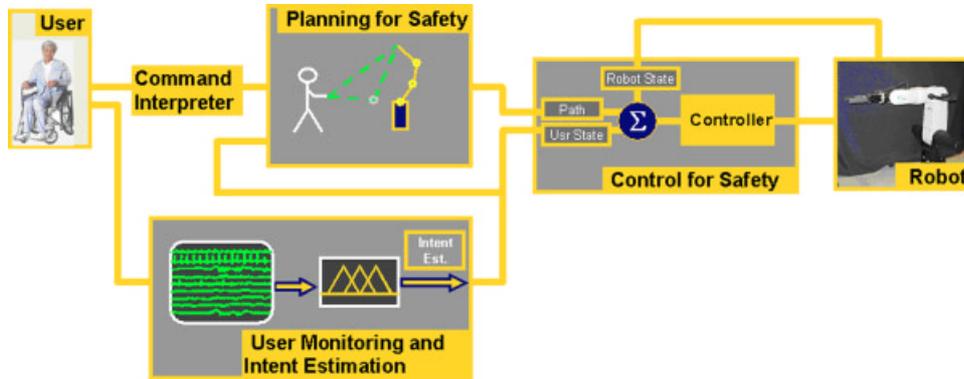


Figure 1. System overview.

perform commanded actions without detailed instructions from the user. An overview of the system is presented in Figure 1. The user issues a command to the robot to initiate the interaction. The command interpreter translates the natural language command (e.g., pick up the red cup) into a set of target locations and actions (e.g., execute a grip maneuver at coordinates $[x, y, z]$). The planning module is divided into two parts: the global path planner and the local trajectory planner. The global planner module begins planning a geometric path for the robot over large segments of the task, utilizing the safety strategy described herein. Segment end points are defined by locations where the robot must stop and execute a grip or release maneuver. For example, one path segment is defined from the initial position of the robot to the object to be picked up. The local planner generates the trajectory along the globally planned path based on real-time information obtained during task execution. The local planner generates the required control signal at each control point. Because the local planner utilizes real-time information, it generates the trajectory in short segments. The algorithm of MacFarlane and Croft²¹ is used to implement the local planner. During the interaction, the user is monitored to assess the user's level of approval of robot actions. The local planner uses this information to modify the velocity of the robot along the planned path. The safety control module evaluates the safety of the plan generated by the trajectory planner at each control step. The safety control module initiates a deviation from the planned path if a change in the environment is detected that threatens the safety of the interaction. This deviation will move the robot to a safer location. Meanwhile, the recovery evaluator will initiate a re-assessment of the plan and initiate replanning if nec-

essary. This paper describes the planning strategy used by the global planner; other components of the system are described in refs. 3 and 4.

2. APPROACH

A hazard requiring a change in robot behavior can be defined by a minimum distance between the robot and the person,^{9,11} or by using a threshold level of the danger index based on impact force.^{14,18} In this work, an index similar to the danger index of Nokata et al.^{14,18} is proposed, and applied to configuration space planning of the robot motion. By selecting safer configurations at the planning stage, potential hazards can be avoided, and the computation load for hazard response during real time control can be reduced; cf. Figure 2. For this reason, safe planning is an

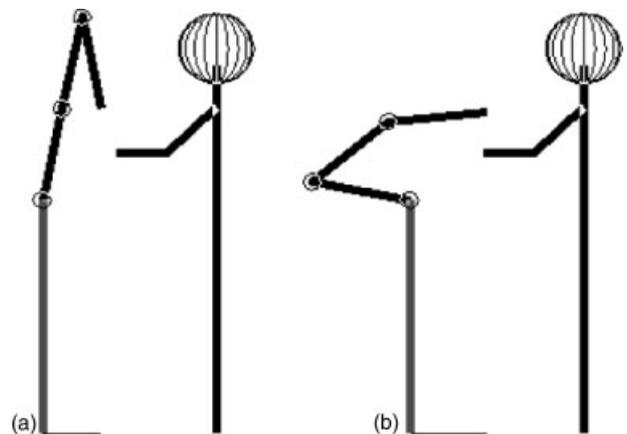


Figure 2. Planning a safe interaction. Posture (b) has minimized potential hazard to human.

important component of the safety strategy. For example, if the path to be followed is planned with a general path planning method, the robot may spend the majority of the path in high inertia configurations. If the human suddenly moves closer to the robot, the potential collision impact force will be much higher than if the robot had been in a low inertia configuration, regardless of the real-time controller used to deal with potential collision events.

When selecting a path planning strategy, there is a tradeoff between fast local methods that may fail to find the goal, and slow global methods.²³ To exploit advantages of both methods, recent path planning algorithms have used a hybrid approach, where global path planning is used to find a coarse region through which the robot should pass, and local methods are used to find the exact path through the region.²⁰

Similarly, in this approach, the global planner generates a safe contiguous region in space through which the robot can move to complete the given task. This region in space is described by a set of contiguous configurations, which represent the path. It is then left to the trajectory planner and the safety module to generate the exact path in the region, and the trajectory along that path. This trajectory is again evaluated and corrected at every control step by the safety control system to handle the real-time aspects of the interaction.

Since the task planning is done following a user request, the global planner must execute within several seconds at most, to avoid a significant delay between a user request and robot response. To ease the computational load on the global planner, the task is separated into segments. Natural segment separation points occur when the robot is required to be at a particular location, for example at each grasp or release point. Only the first segment must be planned before the planned path can be passed on to the local planner and the robot can begin executing the task. In this way, global planning of the next segment can continue in parallel with execution of the current segment.

2.1. Danger Criterion

The planning module uses the best first planning approach.²³ In cases when the number of degrees of freedom (DOF) of the robot affecting gross end effector motion are small (less than 5), the best first planning approach provides a fast and reliable solution.²³ For highly redundant robots, a different search strategy can be employed, such as randomized planning.²⁴ However, the search criteria presented

herein remain identical regardless of the search strategy used. The safest path is found by searching for contiguous regions that (i) remain free of obstacles, (ii) lead to the goal, and, (iii) minimize a measure of danger (a danger criterion). The planning algorithm seeks a path that minimizes a cost function consisting of a quadratic goal seeking function, a quadratic obstacle avoidance function, and the danger criterion (DC).

The central contribution of the planner cost function is the danger criterion. Since path planning (as opposed to trajectory planning) does not consider robot velocities, a configuration-based (quasistatic) danger criterion is required. The danger criterion should be constructed from measures that contribute to reducing the impact force in the case of unexpected human-robot impact, as well as reducing of the likelihood of impact. These can include the relative distance between the robot and the human, the robot stiffness, the robot inertia, end effector movement between contiguous configurations, or some combination of these measures, similar to those proposed by Ikuta and Nokata.¹⁴ Nokata et al.¹⁸ use the danger index to find an optimum safe path, however, only the end effector point trajectory with respect to the human is considered. Herein, a safe path for the entire robot structure is planned, explicitly planning the robot posture. However, since some of the factors affecting danger can conflict (e.g., a low stiffness configuration can also be high inertia configuration) it is important to formulate the danger criterion so that conflicting factors do not act to cancel each other out. Herein, the robot inertia and the relative distance between the robot and the human center of mass are used. The stiffness can be more effectively lowered through mechanical design.^{14,25} Dynamic factors such as the relative velocity and acceleration between the robot and the human are handled by the trajectory planner and the safety controller.³

Since the inertia of a general articulated body is a 3×3 tensor, a scalar value for the robot representing the effective robot inertia at each configuration must be computed. For a general robot architecture, where the robots inertia may be distributed in more than one plane, the highest eigenvalue of the inertia tensor may be used as the scalar measure. For robots with a single sagittal plane (e.g., anthropomorphic, SCARA), the scalar inertial value is extracted by calculating the robot inertia about an axis originating at the robot base and normal to the robot's sagittal plane

$$I_s = \bar{v}^T I \bar{v}. \quad (1)$$

The sagittal plane is the vertical plane (plane of symmetry) passing through the center of the outstretched robot arm. Here, I_s is the inertia about the v axis, v is the unit vector normal to the robot sagittal plane, and I is the 3×3 robot inertia tensor about the base.

For each danger criterion factor, a potential field function is formulated as a quadratic function. The quadratic potential function is most commonly used in general potential field planners. It has good stabilization characteristics, since the gradient converges linearly towards zero as the robot's configuration approaches the minimum.^{23,26}

2.1.1. Sum-Based Criterion

Two danger criterion formulations are proposed: a sum-based and a product-based criterion. For the sum-based danger criterion, the inertia factor is

$$f_{I_{sum}}(I_s) = \frac{I_s}{m} \quad (2)$$

where m is the total mass of the robot. This function can be interpreted as a quadratic attractive function, attracting each link towards the robot base.

The relative distance factor for the sum-based danger criterion is implemented by a repulsive function between the human and the robot center of mass (c.m.). The center of mass distance is used (instead of the closest point distance) to allow the robot end effector to contact the human during interaction tasks, while maximizing the distance between the human and the bulk of the robot. The potential field is described by Eq. (3) below:

$$D_{CMO} = |D_{CM} - D_{\min}|, \quad (3)$$

$$f_{CM_{sum}}(D_{CM}) = \begin{cases} \frac{1}{2\epsilon}: & D_{CMO} \leq \epsilon, \\ \frac{1}{2} \left(\frac{1}{D_{CMO}} - \frac{1}{D_{\max}} \right)^2: & \epsilon < D_{CMO} < D_{\max}, \\ 0: & D_{CMO} \geq D_{\max}. \end{cases}$$

D_{CM} is the distance between the robot center of mass and the human, D_{\min} is the minimum allowable distance between the robot center of mass and the human, D_{CMO} is the distance over the minimum, and D_{\max} is the distance at which this factor no longer contributes to the cost function (for example, if no hu-

man is visible in the environment). ϵ is a small number used to limit the function for D_{CM} near D_{\min} . This potential field is analogous to an obstacle potential field acting between the human's and the robot's center of mass points. As the distance between the robot and the human approaches the minimum allowable distance (D_{\min}), the potential field approaches infinity.

The sum based danger criterion is comprised of the inertia factor and the center of mass distance factor described above, as follows:

$$DC_{sum} = W_i \cdot f_{I_{sum}}(I_s) + W_d \cdot f_{CM_{sum}}(D_{CM}). \quad (4)$$

Here, W_i and W_d are weights of the inertia and distance term, scaled such that $W_i + W_d = 1$. The weights W_i and W_d are tuned based on the particular robot structure. For low inertia robots, and when the robot is close to the person, the distance factor will dominate the danger criterion, because the distance factor approaches infinity as the robot approaches the person. If inertia reducing behavior is desired for the path in these cases, W_i should be greater than W_d .

2.1.2. Product Based Criterion

For the product-based danger criterion, the criteria are scaled such that for each potential function, the level of danger is indicated within the range 0–1. Values greater than one indicate an unsafe configuration.

The product based inertia criterion is defined as

$$f_{I_{prod}}(I_s) = \frac{I_s}{I_{\max}}, \quad (5)$$

where I_{\max} is the maximum safe value of the robot inertia. In the simulations described in Section 5, the maximum robot inertia is used; however, a lower value can be used for high-inertia manipulators. In this case, the maximum safe value can be established based on the largest force magnitude that does not cause pain²⁷ and the maximum robot acceleration.

For the product based distance criterion, similar to the sum based distance criterion, the center of mass distance between the robot and the human is used. The relative distance criterion for the product-based danger criterion is

$$f_{CM_{prod}}(D_{CM}) = \begin{cases} k \left(\frac{1}{D_{CM}} - \frac{1}{D_{\max}} \right)^2: & D_{CM} \leq D_{\max}, \\ 0: & D_{CM} > D_{\max}. \end{cases} \quad (6)$$

The scaling constant k is used to scale the potential function such that the value of the potential function is zero when the distance between the human and the robot is large enough (larger than D_{\max}), and is one when the distance between the human and the robot is the minimum allowable distance (D_{\min})

$$k = \left(\frac{D_{\min} \cdot D_{\max}}{D_{\min} - D_{\max}} \right)^2 \quad (7)$$

Values of the product-based distance criterion above one indicate an unsafe distance.

The product-based danger criterion is then computed as a product of these contributing factors

$$DC_{prod} = f_{I_{prod}}(I_s) \cdot f_{CM_{prod}}(D_{CM}) \quad (8)$$

2.2. Goal and Obstacle Potential Fields

For the goal seeking and obstacle avoidance functions, the customary quadratic potential field functions are used.^{20,23,26} The goal seeking function f_G is defined as

$$f_G(D_G) = \frac{1}{2} D_G^2 \quad (9)$$

where D_G is the distance between the end effector and the goal.

The obstacle avoidance function f_O is defined as

$$f_O(D_O) = \begin{cases} \frac{1}{2} \left(\frac{1}{D_O} - \frac{1}{D_{O_{\min}}} \right)^2 & D_O \leq D_{O_{\min}} \\ 0 & D_O > D_{O_{\min}} \end{cases} \quad (10)$$

where D_O is the distance between the robot and the nearest obstacle, $D_{O_{\min}}$ is the distance from the obstacle at which the obstacle begins to repel the robot (the influence distance). For the obstacle avoidance function, the distance between the robot and the nearest obstacle is taken as the distance between the closest point on the robot and the closest point on the obstacle. The distance between the robot and the nearest obstacle, as well as the distance between the robot and the noninteracting parts of the human are estimated using the hierarchy of spheres representation, illustrated in Figures 3 and 4. This representation is based on the approach described in Martinez-Salvador et al.²⁸ In this approach, the robot and the



Figure 3. Human, robot representation in a noninteractive task.

obstacles in the environment are described as a set of enveloping spheres. Initially, a small set of large enveloping spheres is used for each object. If no intersecting spheres are found, the distance between the two closest sphere centers is returned as the distance between the robot and the nearest obstacle or human. If two intersecting spheres are found, the robot and the obstacles are decomposed into a set of smaller enveloping spheres. The process is repeated until a non-intersecting set of spheres is found, or until the maximum level of decomposition is reached, in which case the algorithm reports that a collision has been detected. The level of decomposition required to find a collision free set of spheres is also used to determine the size of the region within which local trajectory planning may be executed, similar to Brock and Khatib.²⁰

When defining the enveloping spheres for the human, the current robot task also becomes important. If the goal of the interaction is for the robot to ap-



Figure 4. Human, robot representation in an interactive task.

proach and/or contact the human, then it is not appropriate to represent the human simply as an obstacle, as was done in Traver et al.²⁹ In this work, during preplanning, each segment of the path is classified as interactive or noninteractive. If the segment is classified as noninteractive, the entire region of space occupied by the human is treated as an obstacle. If the segment is classified as interactive, a smaller set of spheres is used, such that the target area of the human (for example, the hand) is excluded from the obstacle area. Figure 3 shows the robot and the human represented with enveloping spheres in a noninteractive task segment. Figure 4 shows the representation during an interactive task segment.

2.3. The Overall Cost Function

The planning cost function is generated by combining the goal seeking, obstacle avoidance, and danger criteria. The planned path is generated by searching for a set of configurations that minimize the cost function:

$$J = W_G \cdot f_G(D_G) + W_O \cdot f_O(D_O) + W_D \cdot K \cdot DC. \quad (11)$$

Here, W_G is the weighting of the goal seeking criterion, W_O is the weighting of the obstacle avoidance criterion, W_D is the weighting of the danger criterion, and K is a scaling factor. The selection of the weight levels is discussed in the following section.

3. IMPLEMENTATION

Using the above cost function, it is likely that the danger criterion will conflict with the goal seeking criteria during the search, leading to local minima and long search times. To avoid this problem, a two-stage search is proposed. In the first stage, maximum priority is placed on minimizing the danger criterion. A threshold is established for determining when an acceptable maximum level of danger is achieved. Once a path is found that places the robot below this threshold, the second stage of the search is initiated. In this stage, maximum priority is placed on the goal-seeking criterion. In the resulting overall path, the robot will spend most of its time in low danger regions. One can note that, this approach will not result in the shortest distance path. The tradeoff between increased safety and reduced distance can be controlled by modifying the threshold where switching from the first stage to the second stage occurs. The two stages are implemented by modifying the weighting factors.

In the first (danger minimization) stage, W_D is greater than W_G , while in the second stage, W_G is greater than W_D . As long as the relative weights are set in this manner, the algorithm does not require tuning of the weight levels when using the product based DC. For the sum based DC, if the robot is approaching the person, W_D must be small (0.1 or less) in the second stage to avoid interference with the goal attraction criterion.

Even when the proposed two stage planning approach is used to minimize the conflict between the danger and goal seeking criteria, it is still possible for the goal seeking and the obstacle avoidance to conflict in a cluttered environment, or when joint limits are encountered during the search. The search time is also extended if the robot needs to reverse configurations during the path (for example, from an elbow down starting configuration to an elbow up final configuration). In these cases, a circuitous path is often generated, requiring some postprocess smoothing.²³ In particular, if there are several obstacles positioned close to the robot, it may not be possible to complete the stage 1 search within the given threshold. In this case, the user should be notified that a safe path cannot be found in the current environment. The problem of long search times can also be addressed by taking advantage of particular robot geometry, and searching only through joints that affect the end effector position. For example, although the PUMA560 is a 6 DOF robot, only the first three joints contribute to the gross end effector movement. After the position path is generated, the remaining three joints can be used to maintain a desired end effector orientation, as required by the payload.

4. SEARCH STRATEGY IMPROVEMENTS (BACKWARDS SEARCH)

The global planning strategy presented above is generally valid for nonredundant as well as redundant robots, as well as robots with either prismatic or articulated joints, or mobile robots. This is because the search is conducted forwards from an initial configuration, the search steps are generated from that initial configuration, and therefore only forward kinematics are required to calculate the workspace potential field functions. If an inverse kinematics routine is available for the robot, the algorithm search time can be improved by adding a backwards search stage. This addition is useful in those cases when the robot goal is in a crowded area, for example when the robot's goal is the user's hand. In this case, to get to the goal, the

robot must go into an area of higher potential field, since the goal is surrounded by obstacles generating a repulsive field. Therefore, the algorithm must perform “well-filling” to find the path, which will increase the search time, and may also result in a convoluted final path. On the other hand, if the search could be performed backwards, gradient descent can be used to find the lowest potential path to the goal, reducing the search time. In general, it is always more efficient to search from the cluttered end of the path, reducing the search time. The inverse kinematics routine is used to generate the goal configuration, given the goal workspace position and the desired end effector orientation at the goal. The search is then initiated backwards from the goal configuration towards the start configuration. Once obstacle influence is minimal, the backwards search stops, and the forward search (as described in Section 2) is initiated, with the last configuration of the backwards generated path as the goal. If there are multiple solutions to the inverse kinematics problem, the danger criterion at each solution is evaluated, and the solution with the lowest danger criterion is selected. The location at which the backwards search stops, and the new goal location for the forwards search is named the intermediate location (IL).

For continuity, the algorithm must also ensure that the forwards and backwards generated paths meet at the same point in configuration space. Since the goal and obstacle potential fields are defined in the workspace, it is possible for an articulated robot to reach the starting point of the backwards path in an incorrect posture (e.g., elbow up vs elbow down). In this case, the two paths cannot be joined by simply generating a spline between the two postures. This could cause the robot to move into obstacles or move through a more dangerous configuration. Instead, during the initial stage of the forward search, an additional “posture” potential function²⁰ is added, that favors the starting posture of the backwards path. The posture function is defined as

$$f_{pos}(q) = \begin{cases} \frac{1}{2}(q - quad_{min})^2: & q < quad_{min} \\ 0: & quad_{min} \leq q \leq quad_{max} \\ \frac{1}{2}(quad_{max} - q)^2: & q > quad_{max} \end{cases} \quad (12)$$

Here, q is the search configuration joint angle, and $quad_{min}$ and $quad_{max}$ are the quadrant bound-

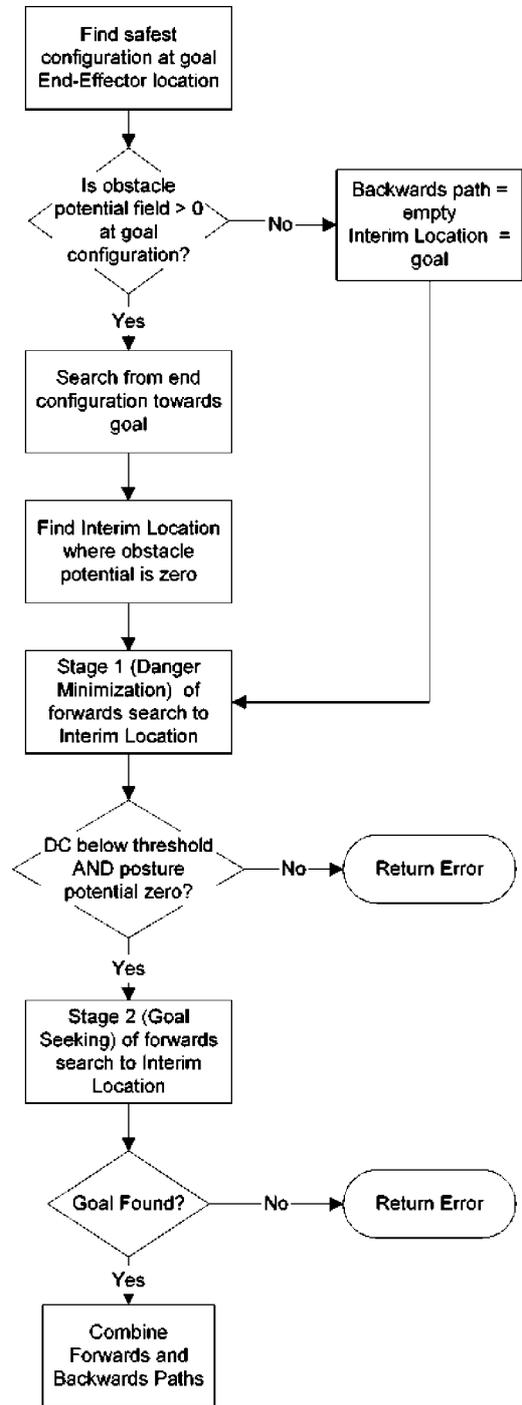


Figure 5. Combined backwards-forwards search algorithm flowchart.

aries centered around the joint angle found in the final configuration of the backwards search.

The posture function is calculated for each joint; the total posture function is the sum over all the

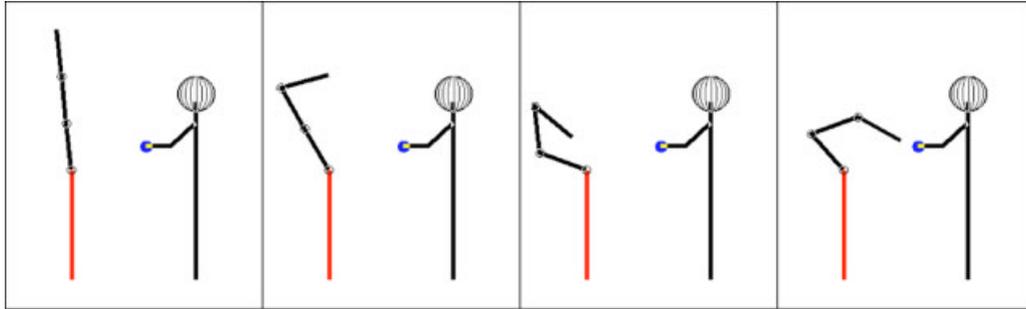


Figure 6. Planned path with sum-based danger criterion.

joints. The posture potential is only active while the robot is in the incorrect posture. When the robot reaches the correct posture, the posture potential becomes inactive. To ensure that the correct posture is reached prior to merging with the backwards planned path, an additional condition is added to the switchover from the first to the second stage of the planning. Namely, in addition to the low danger index requirement, the posture potential must also be zero. A flow chart for the complete algorithm, including the backwards search, is shown in Figure 5.

5. SIMULATIONS AND DISCUSSION

A simulation environment was developed to test the planning algorithms with various robot architectures. A handover task is used as the sample task in the simulations. This type of planned motion is applicable to a wide range of service robot scenarios, including picking up and delivering objects to the user as a personal assistant¹⁷ or positioning the feeding utensil during a feeding task.^{15,16} The robots are modeled using the Robotics Toolbox.³² Figure 6 shows the planned motion of a 3 link planar robot using the ba-

sic algorithm (i.e., without the backwards search), with the sum-based danger criterion. The robot's goal is to pick up the object being held by the human. The same task is shown as planned by the product-based danger criterion in Figure 7. In both cases, to better illustrate the effect of the danger criterion, only the goal and danger criterion cost functions are included. The cost function weights used for both plans are given in Table I. Figure 8 shows a comparison between the human-robot center of mass distance and the robot inertia for the sum-based and the product-based danger criteria.

Figures 6 and 7 illustrate the differences between the two danger criterion formulations. The sum-based danger criterion implies that the factors affecting the danger are mutually independent. One advantage of the sum-based criterion is that the formulation is similar to other quadratic cost functions normally used in the potential field approach, and is distance based. Therefore, the sum-based criterion does not need to be scaled when combined with the other criteria (i.e., $K=1$). The center of mass distance factor is a repulsive potential field, and can, therefore, become infinite in magnitude when the center of mass distance between the robot and the hu-

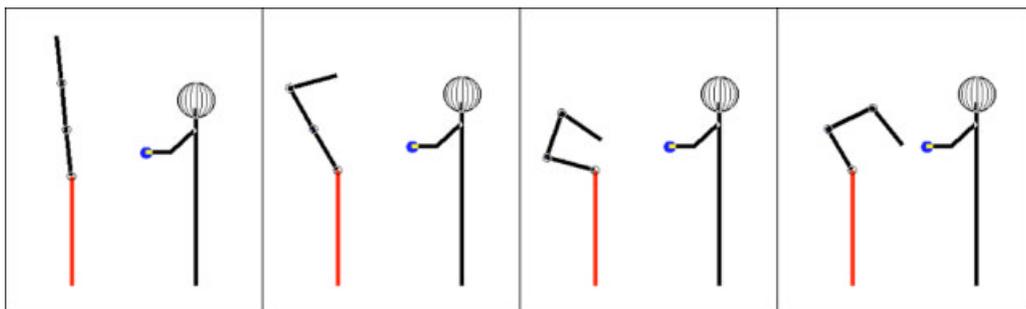


Figure 7. Planned path with product-based danger criterion.

Table I. Planar robot simulations weights.

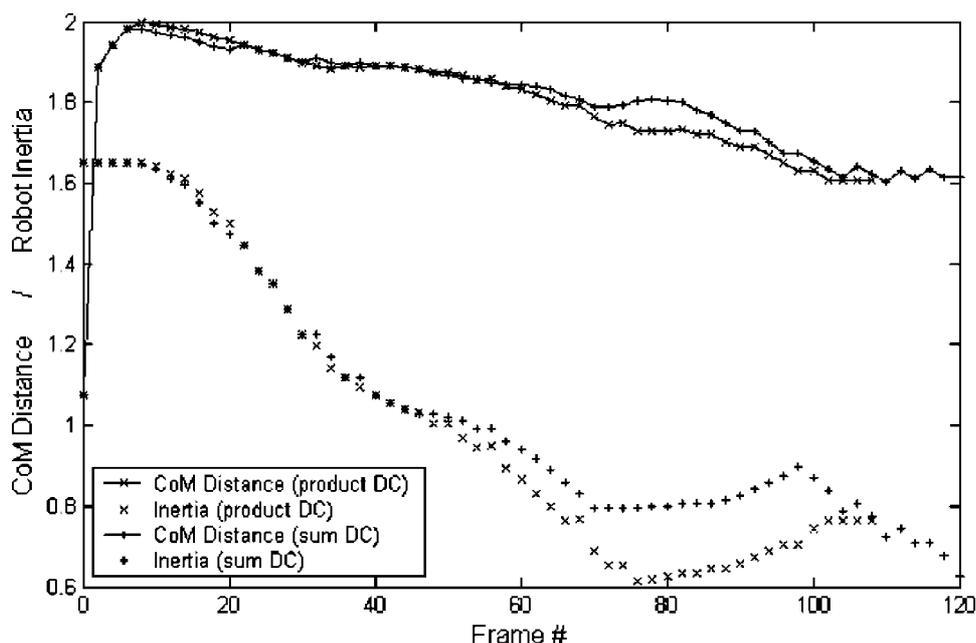
	W_G	W_O	W_D
Stage 1	0.2	0	0.8
Stage 2	0.9	0	0.1

man (D_{CM}) is close to the minimum safe distance (D_{min}). Thus, when the robot and the human are close together, the distance factor will dominate over the inertia factor. This effect is illustrated in the last frame of the Figure 6 sequence. As a result, near the point of interaction, the sum based criterion results in a higher inertia, as can be seen in Figure 8. In general, the disadvantage of such a sum based formulation is that one of the factors always tends to dominate the others. Furthermore, for the sum-based criterion it is difficult to define the threshold at which one should switch from the danger minimization stage to the goal seeking stage, since the danger criterion is a combination of the robot link distances from the robot base and the distance from the robot to the person.

The product-based danger criterion implies that the factors affecting the danger criterion are dependent. For example, if the distance between the robot and the person is large, the other contributing factors will not be minimized either. In Figure 7, since the distance between the robot and the person is small,

both the distance factor and the inertia factor are minimized. In addition, when both factors have significant magnitude, the danger criterion gradient will be steepest, ensuring that the danger criterion is prioritized over the other criteria. Because the two factors are dependent, both are minimized to achieve the required safety level. Another advantage of the product-based criterion is that the criterion represents a clear indication of the level of danger, ranging from 0 to 1 [values greater than 1 are possible when the distance between the robot and the human (D_{CM}) is smaller than the minimum safe distance ($D_{CM\ min}$)]. Therefore, it is easy to specify the switch threshold as the desired level of danger. However, for the product-based criterion, a scaling factor (K) must be chosen so that the danger criterion is on the same scale as the goal and obstacle criteria.

In the majority of cases, the product-based danger criterion is more suitable. The product-based criterion is more suitable for redundant robots, where both the inertia and c.m. distance factors can be minimized, regardless of the c.m. distance. When the robot is close to the person, the product-based danger criterion will decrease inertia and increase c.m. distance. On the other hand, close to the person, the sum-based danger criterion becomes dominated by the distance factor, so inertia is not reduced as significantly. The sum-based danger criterion may be more suitable with large, underarticulated robots. In this

**Figure 8.** Comparison between the sum-based and product-based danger criteria.

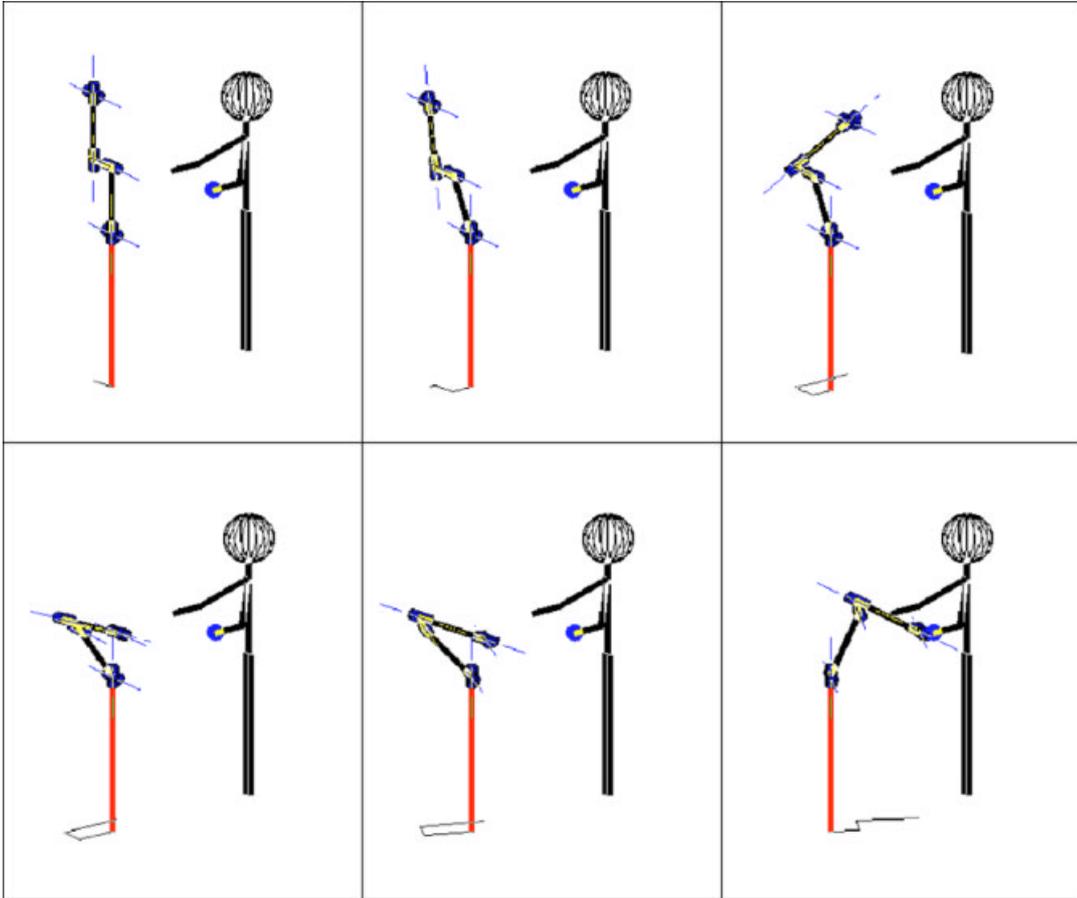


Figure 9. Planned sequence for a PUMA560 robot (product danger criterion).

case, the difference between the maximum and minimum robot inertia may not be very significant, whereas the strong c.m. distance action will ensure that the robot does not get too close to the user.

Figure 9 shows a planned motion sequence with a PUMA 560 robot, using the basic algorithm with the goal, obstacle and danger criteria. The product based danger criterion is used. Table II gives the weights used in the search. For comparison, a path was generated using the best-first planner without any danger criterion. As illustrated in Figure 10, the danger criterion pushes the c.m. of the robot away from the person along the majority of the path, as well as significantly reducing the inertia.

Figure 11 shows a planned sequence using the modified algorithm, with the backwards search added. In this case, the initial robot pose is the reverse of the required final pose generated by the backwards plan. The same weights were used as for the basic algorithm, as specified in Table II.

Initially, while the danger is low, the posture function dominates the potential field, and the robot moves first to move to the correct posture. As the robot comes closer to the person, the danger criterion

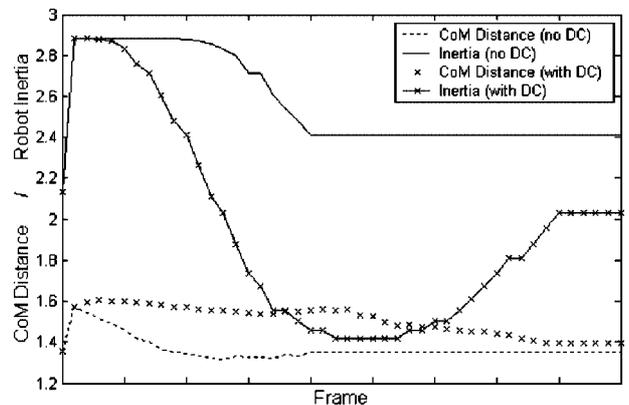


Figure 10. Effect of the danger criterion search on the danger factors (product danger criterion).

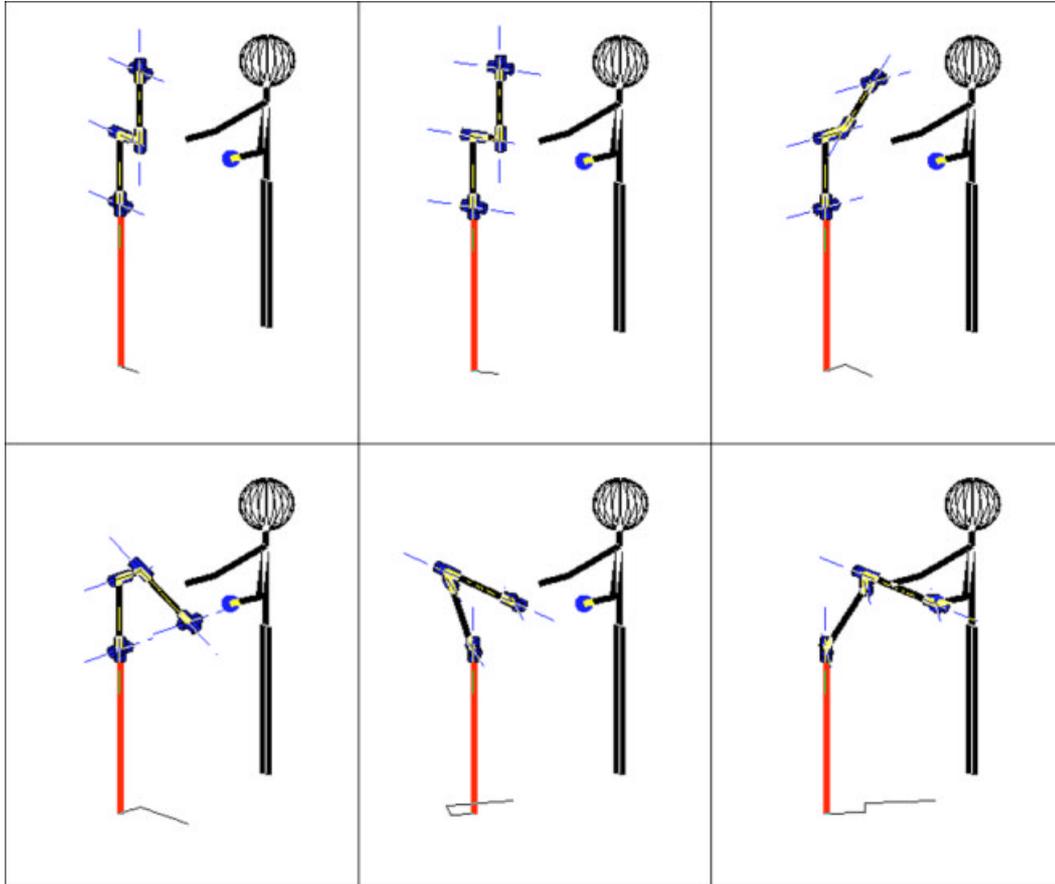


Figure 11. Planned sequence for a PUMA560 robot with backwards search (product danger criterion).

begins to dominate the potential field, and the robot inertia is reduced. Once the danger index has been reduced, the robot moves towards the goal. Posture correction is performed during low danger sections of the path. As shown in Figure 5, the backwards search is only performed when the goal location is within the influence distance of obstacles. In this case, the basic planner must find the entrance into an obstacle region. Using the backwards search, finding a path from the obstacle enclosed goal location to a free region is much easier.³¹ Once a configuration free from obstacle influence is found through the backwards search, the forwards search, incorporating the danger

criterion, is initiated to this configuration. The posture potential must then be added to the forwards search cost function to ensure that the forwards and backwards paths join at the same robot configuration. This allows the planner goal and obstacle fields to be defined in the workspace, while still ensuring a contiguous path in the joint space.

6. CONCLUSIONS

The proposed safe planner reduces the factors affecting danger along the path. Using the two-stage planning approach reduces the depth of local minima in the cost function, allowing the planner to execute quickly. Minimizing the danger criterion during the planning stage ensures that the robot is in a low inertia configuration in the case of an unanticipated collision, as well as reducing the chance of a collision by distancing the robot center of mass from the human. This advance-planning approach puts the robot in a better position to deal with real-time safety hazards.

Table II. PUMA560 simulations weights.

	W_G	W_O	W_D
Stage 1	0.1	0.2	0.7
Stage 2	0.7	0.2	0.1

When an inverse kinematics routine is available for an articulated robot, the performance of the planner can be further improved by adding a backwards search. That is, the path is generated backwards from the goal when the goal location is in an area crowded by obstacles. To ensure that the forwards and backwards generated paths meet, a posture potential is added to the total cost function. By including the posture potential directly into the cost function, rather than splining the two paths after they are generated, the algorithm ensures that posture correction occurs during low-danger sections of the path.

ACKNOWLEDGMENTS

This work is supported by the National Science and Research Council of Canada (NSERC). This work was presented in part at the IEEE International Conference on Robotics and Automation, New Orleans, LA, April 2004.

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, P.I. Corke, and M. Kim, Report on the First IART/IEEE-RAS Joint Workshop: Technical Challenge for Dependable Robots in Human Environments, 2001.
3. D. Kulić and E. Croft, Strategies for safety in human robot interaction, IEEE International Conference on Advanced Robotics, Coimbra, Portugal, 2003, pp. 644–649.
4. D. Kulić and E. Croft, Estimating intent for human-robot interaction, IEEE International Conference on Advanced Robotics, Coimbra, Portugal, 2003, pp. 810–815.
5. RIA/ANSI R15.06—1999 American National Standard for Industrial Robots and Robot Systems—Safety Requirements, American National Standards Institute, 1999.
6. S.P. Gaskell and S.R.G. Went, Safety issues in modern applications of robots, Reliability Engineering and System Safety 52 (1996), 301–307.
7. Y. Yamada, T. Yamamoto, T. Morizono, and Y. Umetani, FTA-based issues on securing human safety in a human/robot coexistence system, IEEE Systems, Man and Cybernetics SMC'99, 1999, pp. 1068–1063.
8. Y. Yamada, Y. Hirawawa, S. Huang, Y. Umetani, and K. Suita, Human-robot contact in the safeguarding space, IEEE/ASME Transactions on Mechatronics 2:(4) (1997), 230–236.
9. A.J. Bearveldt, Cooperation between man and robot: interface and safety, IEEE International Workshop on Robot Human Communication, 1993, pp. 183–187.
10. 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 31:(1) (2001), 49–64.
11. V.J. Traver, A.P. del Pobil, and M. Perez-Francisco, Making service robots human-safe, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000), 2000, pp. 696–701.
12. J.Y. Lew, Y.T. Jou, and H. Pasic, Interactive control of human/robot sharing same workspace, IEEE/RSJ International Conference on Intelligent Robots and Systems, 2000, pp. 535–539.
13. Y. Yamada, Y. Hirawawa, S. Huang, Y. Umetani, and K. Suita, Human-robot contact in the safeguarding space, IEEE/ASME Transactions on Mechatronics 2:(4) (1997), 230–236.
14. K. Ikuta and M. Nokata, Safety evaluation method of design and control for human-care robots, The International Journal of Robotics Research 22:(5) (2003), 281–297.
15. E. Guglielmelli, P. Dario, C. Laschi, R. Fontanelli, M. Susani, P. Verbeeck, and J.C. Gabus, Humans and technologies at home: from friendly appliances to robotic interfaces, IEEE International Workshop on Robot and Human Communication, 1996, pp. 71–79.
16. K. Kawamura, S. Bagchi, M. Iskarous, and M. Bishay, Intelligent robotic systems in the service of the disabled, IEEE Transactions on Rehabilitation Engineering 3:(1) (1995), 14–21.
17. R. Bischoff and V. Graefe, HERMES—A versatile personal robotic assistant, Proceedings of the IEEE 92:(11) (2004), 1759–1779.
18. M. Nokata, K. Ikuta, and H. Ishii, Safety-optimizing method of human-care robot-design and control, Proceedings of the 2002 IEEE International Conference on Robotics and Automation, Washington, DC, 2002, pp. 1991–1996.
19. M. Chen and A.M.S. Zalzal, A genetic approach to motion planning of redundant mobile manipulator systems considering safety and configuration, Journal of Robotic Systems 14:(7) (1997), 529–544.
20. O. Brock and O. Khatib, Elastic strips: a framework for motion generation in human environments, The International Journal of Robotics Research 21:(12) (2002), 1031–1053.
21. S. Macfarlane and E. Croft, Jerk-bounded robot trajectory planning—design for real-time applications, IEEE Transactions on Robotics and Automation 19:(1) (2003), 42–52.
22. K. Erkorkmaz and Y. Altintas, High speed CNC system design: Part I—Jerk limited trajectory generation and quintic spline interpolation, International Journal of machine tools and manufacture 41:(9) (2001), 1323–1345.
23. J.-C. Latombe, Robot motion planning, Kluwer Academic Publishers, Boston, MA, 1991.
24. J. Barraquand and J.-C. Latombe, Robot motion planning: A distributed representation approach, The International Journal of Robotics Research 10:(6) (1991), 628–649.
25. A. Bicchi, S.L. Rizzini, and G. Tonietti, Compliant design for intrinsic safety: General issues and preliminary design, IEEE/RSJ International Conference on Intelligent Robots and Systems, 2001, pp. 1864–1869.
26. O. Khatib, Real-time obstacle avoidance for manipula-

- tors and mobile robots, *The International Journal of Robotics Research* 5:(1) (1986), 90–98.
27. Y. Yamada, K. Suita, K. Imai, H. Ikeda, and N. Sugimoto, A failure-to-safety robot system for human-robot coexistence, *Journal of Robotics and Autonomous Systems* 18 (1996), 283–291.
 28. B. Martinez-Salvador, A.P. del Pobil, and M. Perez-Francisco, A Hierarchy of detail for fast collision detection, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)*, 2000, pp. 745–750.
 29. V.J. Traver, A.P. del Pobil, and M. Perez-Francisco, Making service robots human-safe, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)*, 2000, pp. 696–701.
 30. Y.K. Hwang and N. Ahuja, Gross motion planning—A survey, *ACM Computing Surveys* 24:(3) (1992), 219–291.
 31. K. Kondo, Motion planning with six degrees of freedom by multistrategic, bidirectional heuristic free space enumeration, *IEEE Transactions on Robotics and Automation* 7:(3) (1991), 267–277.
 32. P.I. Corke, A robotics toolbox for Matlab, *IEEE Robotics and Automation Magazine* 3:(1) (1996), 24–32.