FAULT-TOLERANT SAFETY CONCEPT FOR HUMAN ROBOT INTERACTION

Akos CSISZAR#1, Cornel BRISAN#2

#Department of Mechatronics and Machine Dynamics, Technical University of Cluj-Napoca
Str. Memorandumului nr. 28, 400114, Cluj-Napoca, Romania

1 akos.csizsar@mdm.utcluj.ro
2 cornel.brisan@mdm.utcluj.ro

ABSTRACT

An important aspect of pHRI (physical Human Robot Interaction) is to assure the safety of the human participant in the interaction. One of the approaches to assure safety is to plan the paths executed by the robot based on a certain safety criterion. This acts as a constraint for decision making of intelligent systems. This paper introduces the Fault-Tolerant Distance as a novel safety criterion. It assures an injury free interaction between the robot and a human even in the case of a malfunction in the structure of the robot hardware or software.

Keywords: Human Robot Interaction, HRI, safety criterion,

1. Introduction

Robotics is a truly interdisciplinary domain. It unites mechanical systems, electrical machines, control systems and also computer science. Physical Human Robot Interactions (pHRI) receive more and more attention, because of its various benefits. In factories robots collaborating with humans can increase the productivity. Robotic assistants cohabitating with humans can improve the quality of life of the elderly or persons with disabilities. The most important aspect of pHRI is to assure the safety of the human participant in the interaction. Considering the safety of a human while interacting with a robot expands further the interdisciplinary character with safety technologies and in some cases even psychology.

The goal of this paper is to introduce a novel safety criterion, described at a conceptual level, which can assure the safety of the human inside the workspace of the robot, even if the robot malfunctions.

2. State of the art

An important aspect of pHRI is to assure the safety of a human inside the workspace of a robot. The approaches to assuring safety can be categorized by different criteria.

Pre-collision strategies aim to reduce injuries before the collision occurs, post-collision strategies aim to reduce injuries after a collision occurred. A similar categorization can be also found in [1-3]. In order to better understand the difference between the two approaches [4] an analogy from the automotive world is presented. The aim of the pre-collision strategies is similar to the aim of the automotive ABS (Anti-Blocking System) systems, while post-collision strategies can be compared to the automotive airbags.

By the nature of the implementation of these strategies they can be further categorized as design strategies and control strategies.

Planning the path of the robot in order to avoid collisions is a pre-collision safety strategy. The safety criterion is the condition that assures that the robot will not harm the human participant during the interaction. Path planning with obstacle avoidance is carried out to prevent collisions between the human obstacle and the robot. It is important to consider the human as a special type of obstacle, protected by a safety criterion. While collisions with different objects in the workspace can only cause material damage, collision with a human can potentially cause injury or even death.

In the simplest case this safety criterion is a safety clearance, a minimal distance around the human body that the robot cannot violate. A safety clearance of 2cm is suggested in [5]. This 2cm threshold is a rather empirical limit. The MAROCO framework for human robot cooperation, presented in [6], also suggests Euclidean distance as safety criterion, but no threshold value is specified. A general framework for path planning is presented in [7]. The elastic strip framework can be used for motion planning in the case of human robot interaction, but safety criterion other than distance is mentioned in the article. Safety clearance based criteria has the advantage of being computationally simple, but it assures that there will be no collision between the robot and the human only when everything is working properly. However, in
the case of a software or hardware fault the collision avoidance cannot be assured anymore.

A more complex safety criterion can be found in [8]. The Danger Index (DI) suggested here is defined as the ratio between the current TCP force and minimum impact force that would cause harm to the human body upon impact. The DI is more complex than safety clearance, because it quantifies the threat the robot induces upon the human, but unlike the case where safety clearance is used, the collision of the robot is possible if the impact force does not harm the human. However, a collision is not preferred, even if no injury occurs during the collision.

Similarly to the DI, the Kineto-Static Danger Field (KSDF), presented in [9], is also based on velocity. The KSDF has a potential field like formulation. It characterizes the energy accumulated in the robot, but no objective threshold of the index is specified what would guarantee safety.

In [10] a predictive version of the DI is suggested, called Cumulative Predictive Danger Index (CPDI). A neuronal network based estimator estimates the movement of the human. The scenario, where the movement of the human is not akin to the prediction, is handled by the non-predictive DI, presented above. The case where the robot does not follow its trajectory (i.e. it malfunctions) is not handled.

The Head-Injury Criterion (HIC) was developed for crash testing applications in the automotive industry [11]. It assesses the severity of a crash, similarly to other criteria used in the automotive industry (e.g. 3ms-Criterion [12]). Although these are an effective measure of car crashes robotic path planning aims to avoid collisions.

Some of the above cited papers use complex safety criteria for path planning in human environments, but all of them consider that the robot will function properly. A malfunction of the robot can cause serious injuries also when the presented safety criterions are used. In order to assure safety for every scenario, the worst case scenario has to be considered.

3. Fault tolerant safety criterion

The nature of the possible faults is various (e.g. software faults, cable faults, etc.). The key to assuring the safety of the human participant in the interaction is to consider the worst case scenario, where a fault causes robot malfunction. If using such an approach is used, a novel safety criterion can be formulated that assesses “worst case” situation instead of “best case” or “common case” situation.

The emergency brakes are the only part of the robot which can be trusted in the case of a malfunction. This is due to the reduction of the risk of brake malfunction, sometimes even by using redundant braking systems (even in this case it can be considered that only one brake functionality can be trusted). This fault tolerant property of the braking system, and the fact that the brakes are (or can be) activated when a malfunction occurs can be the starting point of assuring human safety in a "worst case" scenario, in the case when robot malfunctions.

A novel safety criterion can be formulated, based on the fact that for the current robot configuration the emergency brakes have to able to stop the robot before colliding with the human.

The key difference between the state of the art and the proposed approach is the ability of guaranteeing human safety also in the case of a malfunction, a requirement set by safety standards.

In order to implement such a criterion has to be formulated mathematically.

In [7] the elastic tunnel framework is presented. In this framework it is suggested that the path executed by the robot, should not be described simply as a curve, but as a volume of the operational space which is occupied by the robot during the execution of a path. Application of this concept to the braking path has not been found.

In the case of braking the braking path of the robot is of interest. A workspace volume can be attached to this path which corresponds to the volume occupied by the robot during the execution of the braking maneuver.

Although the execution of this braking trajectory is unlikely, it represents the worst-case situation.

Let the volume occupied by the robot during the execution of the braking maneuver be called braking volume, \( V_{braking} \). If the braking volume does not intersect any obstacles it can be assured, that in a case of an emergency braking maneuver no collisions would take place. This can be translated to a safety criterion. Traditional distance criteria consider the minimum distance between the robot and the human.

A novel, fault-tolerant distance criterion can be defined.

The fault-tolerant distance criterion considers the distance between the braking volume \( V_{braking} \) and the workspace volume occupied by the human, \( V_{human} \). This concept, compared to a traditional distance criterion, is presented in figure 1.

The volume occupied by the robot is presented as the robot convex hull \( H_{robot} \), \( \phi_{task} \). represents the path executed by the robot. \( \phi_{stop} \) represents the braking trajectory that would be described by the TCP, if the brakes were engaged at \( X_i \).

During the execution of the braking maneuver the robot would occupy described by the braking convex hull \( H_{braking} \). The volume occupied by the human is described by the convex hull \( H_{human} \). The convex hulls presented in the figure represent a simple case, the volume description can be further refined.

The safety criterion becomes a binary condition, a collision detection between the braking volume, and the workspace volume occupied by the human. A safety observer also can be implemented based on this criterion. This is the virtual replacement of the safety fence which is currently used to separate the human worker from the robot in factories.
The disadvantage of the approach lies in the difficulty of real-time operation. It is not yet possible to generate braking trajectories in real-time.

A workaround for this disadvantage is based on the fact that the braking trajectories are not affected by the environment, so they can be pre-computed offline. As this would be a very large lookup table its size can be reduced by considering that small changes in the robot state will cause small changes in the braking volume. In such cases the one slightly larger braking volume can be used to all braking volumes that are contained in that slightly larger volume.

Another workaround which could be used is controlled braking, where the braking trajectory is controlled, instead of emergency braking, this way the real-time computation, of offline lookup table of emergency braking trajectories is not necessary. The safety standards (ISO 10218) permit controlled braking, with the condition, that if it is not working, the emergency brakes must be engaged.

An important characteristic of this novel criterion is that since it characterizes the current state of the robot. This way it is well suited for local planning algorithms, but not only.

Convex hulls are one way of mathematically describing volumes. Another popular method is the usage of bounding boxes (oriented, axis aligned, etc.).

For the different volume description types, different collision detection and distance calculations are recommended. Many of these are real-time capable.

For the path planning the integration of the distance between the two volumes in the path planning algorithm would be beneficial. This way the path planner can plan paths, based on the fault-tolerant distance, which are considered safe. This functionality avoids the collisions between the two mentioned volumes. Separating the two functionalities has the advantage that special safety certified hardware and software is only required by the safety observer. The separated components, however, must use the same safety criterion.

### 4. Computation the Braking Volume

In order to compute the braking volume of the robot, the braking trajectory has to be known. Calculating the braking trajectory is not the topic of this paper. The theoretical aspects behind obtaining the braking path can be found in [13, 14]. The calculations of the braking trajectory cannot be done in real-time, but since they are not affected by the obstacles in the workspace, they can be simulated a priori, defining this way a braking volume database, which can act as a lookup table for the different robot states. Obviously size reduction of this database must be considered.

Figure 2 presents visually how to obtain the braking volume, once the braking trajectory is known.

In order to compute the braking volume, these following steps have to be completed:

1. The CAD model of the robot is required as input to the process. Most commercially available robots have their CAD model available freely. Custom made robots are usually built after their CAD design. This requirement does not limit the applicability of the method. Although there are many CAD model formats, they all have the same aim, describing the volume required by the robot accurately (up to micrometers). Most CAD formats can be converted to almost all other formats, using the correct software tools.

2. CAD models are well suited for manufacturing processes, but they are less suitable for mathematical models. In order to include CAD models in mathematical processes it is a common practice to convert them, losing this way the precision of CAD models, but gaining simplified mathematical description, and so lowering the needed computational power. There are more ways that volumes can be described mathematically, all having their advantages and disadvantages. Their principle is to approximate the volume described, with mathematical entity that is less precise than the CAD model, but has the advantages, that is require less computing power when included in a mathematical model. The simplified volume, having a regular shape must always fully incorporate the precisely described volume. Hence the name of such volumes is bounding..
volume. It is important to parameterize this volume model. Every linkage of the robot should be described separately. The parameterization of this volume model is based on the kinematic constraints of the robot, and these constraints link together the bounding volumes, surrounding the linkages.

3. The current state of the robot is an input to the process. The state of the robot must include all parameters which are relevant from the point of view of braking simulation. As mentioned before this aspect is not the topic of this research.

4. The braking path can be generated based on the state of the robot. The path describes the coordinates of the TCP while the robot executes a braking maneuver from the state described earlier.

5. The braking volume is based on the braking path and the parameterized volume model. It represents the volume in Cartesian space occupied by the robot during the execution of the braking maneuver. It can be obtained as the Minowski sum of all the instances in discreet time of the parameterized volume model while executing the braking path. The execution of the braking trajectory represents the worst case scenario which is unlikely to happen. The braking volume quantifies the worst case scenario, and it is the basis of the fault-tolerant distance criterion.

The Fault-Tolerant Safety Criterion is defined as a binary condition. This eliminates the empirical factor of defining different safety thresholds for other published safety criteria. It expresses the overlapping of the volume occupied by the human inside the workspace of the robot, and the braking volume. As such, it can be formulated as a collision detection problem (also called intersection check). If the two volumes do not overlap, the criterion is satisfied.

\[ V_{\text{Braking}} \cap V_{\text{Human}} = \emptyset \]  (1)

Where \( V_{\text{Braking}} \) represents the braking volume, \( V_{\text{Human}} \) represents the volume occupied by the human, \( \cap \) represents the intersection of the two volumes, and \( \emptyset \) represent an empty set.

The determination of \( V_{\text{Human}} \) can be done by modeling a human body as a volume, but the position of the human body in the workspace must be detected by a complex sensorial system (laser scanners, stereo cameras, etc.). Furthermore the sensor system must be safety certified or human robot interaction applications.

The violation of this condition represents the case when the two volumes overlap (e.g. a collision would occur if the robot would brake).

Collision detection problems are in many cases reformulated and solved as distance calculation problem. If the minimum distance between two volumes is greater than zero they cannot overlap. The minimum distance between the braking volume and the volume occupied by the human should be greater than zero, this way the equation (1) will be satisfied.

Converting the Fault-Tolerant Safety Criterion from a collision detection problem to a distance calculation problem leads to the definition of the Fault-Tolerant Distance.

\[ d(V_{\text{Braking}} \cap V_{\text{Human}}) \geq 0 \]  (2)

Where \( d \) represents the Cartesian distance between the braking volume, \( d(V_{\text{Braking}}) \), and the volume occupied by the human, \( V_{\text{Human}} \).

The method of calculation of the distance between two volumes is strongly influenced on how the volumes are described mathematically. The important aspect in this matter is that this distance calculation, unlike the braking volume, is dependent on the state of the environment of the robot, not just the state of the robot. As such it has to be carried out in real-time.

Different algorithms for most volume description methodologies can be found in the state of the art, that can compute the distance in real-time, having as input the two volumes. Furthermore, trading of the precision of the volume description speeds up calculations significantly, this way real-time performance can be assured.

A fault in the hardware or software structure of the robot provokes the activation of the brakes. This assures that even if a fault occurs, the robot will not collide with the human.
The reduced of chance of collision assures the physical integrity of the human, psychological factors (e.g. fear, panic, stress) are not considered.

6. Computation of Braking Trajectory

A linearized braking model was considered to generate the braking volume of the robot. It is considered that the amplitude of the braking acceleration is linearly decreasing with time.

\[ \ddot{q}_i = \alpha t + \beta \]  (3)

Where \( \ddot{q}_i \) represents the acceleration of a joint, \( t \) represents time and \( \alpha \) and \( \beta \) are parameters of the braking.

This way the velocity of the joint \( \dot{q}_i \) becomes:

\[ \dot{q}_i = \int_{t_{start}}^{t_{stop}} \ddot{q}_i \, dt \]  (4)

Where \( t_{start} \) represents the moment in time when the braking maneuver starts, and \( t_{stop} \) represents the moment in time when the braking maneuver ends.

Thus the joint angle can be written as:

\[ q_i = \int_{t_{start}}^{t_{stop}} \dot{q}_i \, dt \]  (5)

Where \( q_i \) represents the joint angle.

The obtained path in joint space can be converted to Cartesian space by discretizing the obtained curve in joints applying the forward kinematics function to the discrete points.

\[ X = f(Q) \]  (6)

Where \( X \) represents the pose of the robot in Cartesian space in vector form, \( Q \) represents the pose of the robot in joint space in vector form and \( f \) represents the forward kinematic function.

7. Results

The above described theory was applied to the theoretical model (CAD and kinematics) of a 6dof Kuka KR-500 robot.

Figure 3 shows the braking trajectory of the \( q_4 \)-axis in a certain scenario. The scenario is presented in figure 4. The robot executes the trajectory marked with the green line, and at a certain point it executes the braking trajectory marked by the red line. The braking trajectory was obtained using equations (3-6).

Having the braking trajectory the braking volume can be generated, based on the algorithm described earlier. The obtained braking volume for the presented scenario is shown in figure 5.

A point to point motion of the robot and a human obstacle is considered for the validation scenario. A path planning algorithm has been run two times, with and without the fault tolerant safety criterion. The results are shown in figure 6.

The red trajectory represents the path planned without using the safety criterion. This trajectory violates the Fault-Tolerant Safety criterion. If a fault
should occur during the execution of the portions of the path which are close to the human obstacle, the robot executing a braking maneuver will collide with the human obstacle. The blue trajectory has been planned by the APF method with integrated Fault-Tolerant Safety criterion. No other parameter of the path planning was changed. The qualitative difference can be seen on the figure. The generated path respects the Fault-Tolerant Safety criterion. This way, a fault should occur at any time during the execution of the path, the robot executing a braking maneuver will not collide with the human obstacle.

7. Conclusions
A novel approach to assuring safety of the human participant in pHRI has been presented. This novel criterion can guarantee the safety of the human even in scenarios where the robot malfunctions. The criterion characterizes the current state of the robot, relative to the current state of the obstacle. This way it is well suited both for local and for global planning methods. It characterizes a state as safe or unsafe, without empirical thresholds.

The method requires as input the braking trajectory of the robot. Unfortunately it is not yet possible to generate the braking trajectory of a robot in real-time. However, since the braking trajectory does not depend on the environment, it is possible to generate the braking trajectories offline, and use large lookup tables, or databases to have access to this information in real-time.

Acknowledgement
This work was partially supported by the DAAD (German Academic Exchange Service) and by the project "Doctoral studies in engineering sciences for developing the knowledge based society-SIDOC" contract no. POSDRU/88/1.5/S/60078, project co-funded from European Social Fund through Sectorial Operational Program Human Resources 2007-2013.

References