

Research Article

Advanced Methods for Robot-Environment Interaction towards an Industrial Robot Aware of Its Volume

Fabrizio Romanelli

Comau Robotics S.p.A., Via Rivalta 30, 10095 Turin, Italy

Correspondence should be addressed to Fabrizio Romanelli, fabrizio.romanelli@comau.com

Received 27 January 2011; Accepted 11 July 2011

Academic Editor: Gordon R. Pennock

Copyright © 2011 Fabrizio Romanelli. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A fundamental aspect of robot-environment interaction in industrial environments is given by the capability of the control system to model the structured and unstructured environment features. Industrial robots have to perform complex tasks at high speeds and have to satisfy hard cycle times while maintaining the operations extremely precise. The capability of the robot to perceive the presence of environmental objects is something still missing in the real industrial context. Although anthropomorphic robot producers have faced problems related to the interaction between robot and its environment, there is not an exhaustive study on the capabilities of the robot being aware of its volume and on the tools eventually mounted on its flange. In this paper, a solution to model the environment of the robot in order to make it capable of perceiving and avoiding collisions with the objects in its surroundings is shown. Furthermore, the model will be extended to take also into account the volume of the robot tool in order to extend the perception capabilities of the entire system. Testing results will be showed in order to validate the method, proving that the system is able to cope with complex real surroundings.

1. Introduction

Robots are meant to become part of everyday life, as assistants at home, as our appliances, and in the industrial environments; they can take part to the work chain as coworkers, completing hard and demanding jobs. In this context, creating autonomous robots that can learn to act in unpredictable environments has been a long-standing goal of robotics, artificial intelligence, and cognitive sciences. An important step towards the autonomy of robots is the need to provide them with a certain level of independence and dynamic behaviour in order to face quick changes in the environment surrounding them; to get robots operating outside rigidly structured environments, such as research centres or universities facilities and beyond the supervision of engineers or experts, it is necessary to face different technological challenges, amongst them, the development of strategies that allow robots to learn from their own experiences and interaction with the environment. The interaction is the generic relation between two (or more) subjects where each subject modifies repeatedly its behaviours in relation to the behaviours of other subjects. In order to extend this

definition to robotics, it is important to understand the basis of human interaction with the environment in order to make the robot acting in a similar manner to what people do while moving and interacting across their space. The ways the robot interacts with the environment are several: it acquires data from the surroundings through its sensors to provide the necessary input signals to the controller and it performs actions in order to achieve desired tasks. The entire robot-environment interaction can be described completely using two models, under the assumption that the robot controller is reactive where the output of the controller does not depend on its internal states, but only on the current input signals provided to it; moreover, another assumption must be that the robot works and operates in a controlled environment with no other external factors which influence its surroundings (i.e., the change in robot perception is only dependent on the actions of the robot). Given these assumptions, the robot-environment interaction model is described both by the robot controller model, which computes the desired motor responses of the manipulator according to its perception, and the perception model, which emulates how the perception of the robot is affected by its own actions.

We will focus on the robot controller model in order to describe the robot-environment interaction, starting from the modelling of the static environment through the use of elementary geometrical regions as in [1]. In regard to the nature of the interaction between a robot and its environment, robotic applications can be categorized in two classes. The first class is referred to noncontact tasks (unconstrained motion in a free space, without any environmental influence on the robot). In these tasks, the robot dynamics is the most important aspect as regards its performance: several industrial applications such as pick-and-place, spray painting, gluing, and arc or spot welding belong to this category. In contrast to these tasks, many complex advanced robotic applications (packaging, assembling, or machining) require the manipulator to be coupled with other objects which can move. The contact tasks can be furthermore divided into two subclasses: essential force tasks and compliant motion tasks, as described in [2]. The first subclass requires the end-effector to establish a physical contact with the objects in the environment and exert a process-specific force. In these tasks, a synergy between the control of the end-effector position and interaction forces is required; some examples of this kind of tasks are deburring, roughing, bending, polishing, and so forth. In these tasks, the force has to be controlled in relation to the particular process in order to prevent overloading or damaging the tool or the objects to be manufactured. In the second subclass, the tasks focus on the end-effector motion, which has to be realized close to the constrained surfaces, and it must be compliant (i.e., capable of reacting to the interaction forces). In this subclass, the problem of controlling the robot is joined to the problem of accurate positioning (as in part-mating process). These processes are often characterized by the presence and occurrence of contact with constrained surfaces, and then the control must cope with reaction forces. The measurement of interaction force here provides information for error detection and for the identification of the parameters to modify the prescribed robot motion. In the future of robotics, the interaction with the environment is fundamental and more and more tasks will include and require interaction. In this paper, we will focus on the noncontact task class; in this context, a lot of considerations can be done, taking into account the feasibility of these applications in a real industrial environment.

As a first step, we have developed a system which can manage a set of different geometrical shapes (spheres, cylinders, and parallelepipeds) which define regions where movement and access are forbidden or allowed; besides this feature, a warning zone is defined as a thickness from the region of interference, useful in order to control the general speed override of the robot Tool Centre Point (TCP for now on) in fact, in these areas the speed override is controlled according to a control law which changes in function of the robot TCP position, adjusting the speed proportionally to the distance between the TCP and the avoided zone. At this stage, the algorithm checks only the TCP position and the spatial checks are performed on it (and not on other parts of the robot). The following step will take into account the modelling of volumes interactions in order to extend the method and the control system to include collision detection

paradigms between volumes. This will allow the definition of an effective model of the static features inside a real industrial cell, which must be checked against the volume of part of the robot. At first, an overview of the theory of collision between solids has to be analyzed in order to introduce in the interference regions that model the possibility to check minimum distances between volumes and to prevent collision between the same volumes. Subsequently, the critical points which have to be passed to the interference regions control have to be detected by the engine. The entire volumetric collision detection engine will be engineered in order to take into account the bulk of the tool mounted on the robot, without limiting the capabilities of the interference regions control system of testing the only tool centre point against the elementary geometrical regions. The theory of collisions between volumes is here studied, in order to detect if a volume is in contact with others; furthermore, in order to correctly implement the methods of volumetric interference regions, a further module of computation of minimum distance between volumes has to be taken into account. The study of such architecture is a step towards the realization of a more intelligent robot that is aware of its volume, starting from the tool mounted on its flange and giving the opportunity to extend this model to the whole robot arm. The final step is to create a general architecture where volume collision detection and closest point detection are integrated together with the interference regions control system. Test results will be shown to prove the effectiveness of the presented architecture. These tests refer to the robot while it is moving at its maximum speed and while performing technological tasks in a real industrial cell. The algorithm that has been tested here is the interference regions control system without the management of volumetric control paradigm. We are confident that these results can be easily obtained with the presented volumetric method, since the following tests refer to the collision avoidance core which takes as input a set of tool centre point positions which are the same output of the closest point detection block previously described; therefore, we think that these results can be generalized for the presented solution.

A side aspect of the robot-environment interaction is the interaction of the robot with human operators; this is an important social aspect that has to be taken into account when developing new robotics theories; in the last century, the growth in automation inside the factories and industries was exponential and this has led to a rapid change in the conditions of human operators. In particular for muscular fatigue, technology has substituted tension and mental effort; for the more advanced automated plants, the transformation of physical energy into technical and mental skills is even emphasized [3]. Another product of the increase in automation is the sense of alienation which has to be faced by the human operators who have to work in tight connection with robots or other mechanical instruments, being actors of repetitive tasks without complex interactions with their "mechanical coworkers." The interaction between robot and environment has been studied in this paper in order to also take into account the role of the human operator in the programming and maneuvering of robots, making his role

closer to the productive process and less alienating; this is possible, considering the overall increase in the interaction between robot and environment which is the basic idea of this paper.

2. Model of Static Environment

In this section, a model of the static environment will be described in order to define a correct and effective model of static features inside a real industrial robotized cell. The first step consists of the definition of a general model that can be analyzed in order to cover all the possible scenarios inside the industrial environment; keeping this in mind, it is necessary to synthesize a control system capable of taking into account the elementary geometrical regions in order to provide the robot with the fundamental tools to face a basic level of interaction. The study of this subject is then a good start to realize a more intelligent integrated robotized cell, where the strong interaction between robots and humans becomes closer and closer.

2.1. State of the Art. In this paragraph, the modelling of static environment will be shown; this topic is still at a basis level, and there are several starting points of study in particular in the industrial automation background. PILZ developed a new control system for areas crowded with robots, moving machines, and human operators [4]; with this option, it is possible to broadcast a safe output in order to immediately halt the machines operating in those areas, avoiding potentially harmful situations. With ABB world zone software option [5], it is possible to define zones where the robot tool centre point cannot enter. If the robot end effector ends in an off-limits area, previously defined by the programmer, the control cuts the power and quickly stops the robot. This system lets the user program world zone software which is especially useful when two robots work close in order to prevent collision and establish working protocols and policy. However, this solution is quite limited as the model of the static environment is made taking into account the robot tool centre point rather than the complex structure of the robot arm; anyway, this solution represents a valid starting point to go through the development of a wider and more complex control paradigm to take into account volume versus volume collision, as will be shown later on. In the literature, there are several different approaches to the study of space occupancy and cooperation and to cope with collision avoidance problem. In particular, path planning is strongly associated with the problem of forbidden zones [6–8]. The management of operative space is a matter of study and development in the field of telemanipulation and robot assisted tasks [9], where security of avoiding forbidden zones is the main objective of the work. Amongst the previous studies in collision detection, there are some works to be mentioned as the one [10] where given two general polyhedra of complexity n , where one is moving while translating or rotating about a fixed axis, determine the first collision, if any, between the two objects. Another

important aspect of collision detection and control of forbidden areas is presented in several works [11, 12], where the dynamics of complex bodies is simulated over a system equipped with a collision detection algorithm. In the industrial field, other approaches aimed to reach a greater level of automation in robot-environment interaction. Fuzzy logic allows controlling a system in order to avoid access to dangerous areas [13]. There is a further approach that will be presented later on this paper, where a system uses Bayesian occupancy grid and a fuzzy logic controller in order to avoid the collision between robot and other objects or humans moving around the cell [14]. In the following paragraphs, a new method to synthesize a control system capable of managing a set of predefined geometrical areas will be shown; with this paradigm, the advantages of taking into account a space model will be shown as well. This method will be the basis to extend the control from point versus volumes, to volume versus volumes collision detection as described later on in the paper.

2.2. Modelling Static Environment with Elementary Geometrical Volumes. The modelling of robot surroundings is a crucial problem in order to cope with management of robot working space, especially when this space is shared with other robots or machinery. The basis of the robot-environment interaction is the capability of the control system to define primitive geometrical areas in order to cover all the possible configuration of the objects inside a real industrial cell. The elementary objects are defined as parallelepipeds, cylinders, spheres, and planes. With this simple modelling, the system can be provided with the capability of defining multiple geometrical areas of these types in order to cover almost every object inside the cell (such as working tables, machinery, or moving objects such as rails and conveyors). Starting from this definition of elementary geometrical area, in the following paragraph, a control system will be shown. This control system allows the user to move the robot around the operating area with the certainty that it will not collide with the forbidden regions; these regions must be previously defined, declared, and activated in order to work correctly. On the other hand, the system could manage another type of regions (called monitored) linked to a digital output which can be raised when the robot TCP enters that region.

2.2.1. Control Paradigm. The developed system allows the programmer to define multiple elementary zones which can be integrated inside the standard robot control and which represent the database of spatial forbidden areas that are used in order to control both the position and speed of the robot end effector. The areas can be easily defined by the programmer, considering several simple geometrical reasoning in order to define a parallelepiped, it is sufficient to take two points (the lower left corner at the base of the shape and the upper right corner); with these two points declared, the first shape can be integrated into the forbidden zone database. The cylinder on the other hand is univocally defined considering the centre of the base circumference, the radius, and its height with this convention, the base of the

cylinder is parallel to the XY plane of the world frame reference. The sphere is instead defined considering its centre and radius. Planes can also be defined into the system, declaring three points as the origin, X - and Y -axes and considering the Z -axis as perpendicular to the plane. Thanks to the possibility of defining several zones in the same operating area, obviously a lot of industrial applications can be covered by using this control paradigm. An important feature of this system is the possibility to consider the areas with two different features, concerning their *life*; they can be either constant or temporary. The first typology is programmable and modifiable from a particular class of users, and they cannot be ignored or modified by user programs these zones are active during all the cycle of the robot. These can be used, for instance, in order to define zones that cannot be covered by the robot end effector as they are occupied by fixed structures, such as pillars or other irremovable facilities. The second typology is temporary as it can be activated or deactivated from each user program, and it is a very useful function in order to manage interlocks for exchange zones between robots when a robot is inside an elementary zone, it is compulsory that the other robot is avoided to access the zone. This feature can be extended to those systems including a network of robot controllers where the information about the elementary areas in the robots cell is shared. In this case, the control system of each robot can be supervised by another controller in order to update the information of the position of the robots as regards the position of multiple elementary areas defined on the cell. Another important feature of the presented system is that it allows the user to define an area bigger than the elementary region, called warning zone in this zone, the robot can keep working but with a safe control system that checks the distance between the surface of the elementary area and the robot end effector, forcing the robot speed override to a value proportional to that distance. With this method, the robot end effector speed is reduced when the control system realizes that the robot violates the warning zone; activating this, it is also possible to avoid mechanical solicitations due to hard brakes and to allow human operator to better perceive the enabled elementary zone around the robot. This control law is applied to each geometrical area and the resulting speed overrides (one for each declared elementary area) are computed in order to find the minimal value of them and to apply it to the robot. Another important innovation of the presented method is that concerning the implementation of dynamic management of elementary areas; in particular, it is possible to program areas which can change their positions in time. This allows the elementary zones to be linked to moving objects (such as moving machines or to end effector of other robots inside the cell); in order to use this position, it is necessary that the current position of the objects has to be known since the robot controller must know this information, or it must similarly share it with other cooperative robots inside the cell. In order to fully describe this feature, a space where several robots operate can be considered in this configuration, it would be useful to define a dynamic zone on each robot end effector (linked to it). In this condition, each robot

knows exactly and instantly the position of other robots end effector: if a robot draws too much to another one, the presented control is able to prevent damage between them. This allows the robot programmer to be released from the need of defining interlocks with some useless waits, while with this management it is possible to define digital outputs when a robot accesses a specified zone (not when specified in a user program); this control is parallel and acts in real-time, despite the classic management of interlocks. Thus, it is very important during the productive cycle when robot programmer has to develop applications where more robots and machines share the same working space, and the presented method helps the user to exactly bound the working areas. The definition of a shared information on the state of the elementary zones (if they are occupied by a robot or not) is very useful and innovative for what concerns monitoring a dynamical geometrical area (e.g., with conveyors). With this system, it is possible to link a dynamic zone to a moving object and this allows the definition of dynamical interlocks, which can be shared through a network between robots, giving a global visibility in the whole cell. The kinematics information of the first three robot joints and the tool allows the definition of useful information avoiding collision between robots cooperating in the same cell; this system, on the other hand, is not safe for the interaction between robots and human operators, but it is thorough in order to protect and prevent damage between robots and facilities without the need of further devices.

2.2.2. Control System Integrated in the Robot. In this section, an accurate description of the architecture of the control system will be shown. With the proposed solution, each controlled elementary zone can be programmed and defined using both the programming language (PDL2 for Comau robots), describing the complete geometry, shape, thickness of the warning zone, and its static or dynamic typology. Using this method, it is possible to program elementary areas to be controlled, in a precise way, and this solution can be very suited for off-line programming. A second method to perform the definition of an elementary zone is to use the robot in order to teach the distinctive points of a geometrical area (as shown in the previous section). With this method, the user will be asked to move the robot around the working area while locating the characteristic points of the geometrical shapes to be defined teaching these points will bring the advantage of having a direct comparison with the taught elementary-controlled zone and the real obstacle inside the working area. These methods provide the user with simple tools in order to create the database of the controlled elementary zones which allow the control system to perform complex operations of the spatial checks on the robot working area. With this solution, it is also possible to program and define for each declared zone, *channels* of shared information which can be activated automatically whenever the robot end effector enters a controlled zone or a warning zone; this also allows to have a quantization of the working area. The operator who uses the proposed solution has the possibility of tuning a set of parameters which makes

the system extremely flexible and modular. It is also possible for example to define a dynamic controlled zone, linked to the end effector of another robot, in order to check the possible collisions between the robots. The control scheme is depicted in Figure 1.

As shown in the scheme the geometrical control algorithm checks if the robot end effector is inside a controlled zone or, analogously, a warning zone. This check is done on the basis of the database of geometrical areas, previously defined by the user; in this context, the dynamic objects position provides the control system with the possibility to link the geometrical areas to arbitrary moving points (as conveyors or rails), which can be read from external sensors like encoders. The speed control is performed by the geometrical area control block which detects the typology of the shape and selects the correct control law to be applied in order to modify the robot override, preventing collisions with the user-defined zones. The speed override is changed smoothly when the robot end effector comes up against a spherical elementary zone, according to the following control law:

$$\begin{aligned} v &= v_o \cdot \frac{d-r}{\delta}, & r \leq d \leq r + \delta, \\ v &= v_o, & d > r + \delta, \\ v &= 0, & d < r, \end{aligned} \quad (1)$$

where v is the actual speed override of the robot end effector, v_o is the old override, d is the distance between the robot end-effector and the centre of the elementary spherical area, δ is the thickness of the warning zone and r is the radius of the sphere (the area is depicted in Figure 2(a)). When the robot encounters a cylindrical elementary zone its speed override is subject to the following control law:

$$\begin{aligned} v &= v_o \cdot \frac{d_1 - r}{R - r}, & 0 \leq z \leq h, \\ v &= v_o \cdot \frac{d_2}{\delta}, & h < z \leq h + \delta, \quad -\delta \leq z < 0, \quad p \in \text{cyl}, \\ v &= v_o \cdot \frac{d_3}{\delta}, & h < z \leq h + \delta, \quad -\delta \leq z < 0, \quad p \notin \text{cyl}, \end{aligned} \quad (2)$$

where h is the height of the cylinder p the position of the robot end effector. The distances d_i represent the distance between the centre of the cylinder and the robot position (d_1), the distance between the cylinder top or bottom base and the robot position, when it belongs to the top/bottom cylinder with thickness δ (d_2), and the minimal distance between the robot position and the points on the circumference of the top/bottom cylinder base (d_3). The robot speed override coincides to the old speed override when the robot end effector is outside the warning zone; the cylindrical elementary area is depicted in Figure 2(b) with its warning zone.

Another geometrical shape is the plane depicted in Figure 2(c). The control law for the plane is the following:

$$v = v_o \cdot \frac{d}{\delta}, \quad (3)$$

where d is the distance between the plane surface and the robot position when it is inside the warning region with thickness δ . The last modelled elementary geometrical volume is represented by the parallelepiped in Figure 2(d); its control law is quite complex as its warning zone is composed by 8 half-lunes, 12 quarters of cylinder, and 6 planes. Given that the mathematical treatment of the control law for the parallelepiped warning area is not reported here, it is enough to consider that this control law smoothly covers the whole warning zone area, with an appropriate speed override for each sector of it. Each elementary zone declared inside the working space has its own control law, also depending on the thickness of the warning zone; it is fundamental when the control system has to fix the controlled speed override that the correct value will be chosen in an efficient way. It is chosen according to the following:

$$\min(v_1, v_2, \dots, v_M), \quad (4)$$

where M is the number of elementary geometrical volumes defined inside the robot working space. This solution, notwithstanding its easiness, assures that the selected controlled speed override v_i follows a smooth trend when several different elementary zones are defined in the robot environment, even if they overlap. The last important feature of the presented paradigm is the possibility to manage dynamic geometrical areas, linking the position of moving objects to distinctive points belonging to the previously defined elementary shapes for the spherical area, this point is characterized by the centre of the sphere. The cylinder will have its characteristic point on the centre of its bottom base; the plane will have its characteristic point on the origin; eventually, the bottom base centre of the parallelepiped will represent its characteristic point.

3. Volume Environment Modelling

In this section, a model of the collision between volumes will be described; this will allow the definition of an effective model of the static features inside a real industrial cell, which must be checked against a part of the robot volume. First of all, an overview of the theory of collision between solids has to be analyzed in order to introduce in the presented interference regions model the possibility to check minimum distances between volumes and to prevent collision between the same volumes. Subsequently, the engine has to detect the critical points that have to be passed to the interference regions control in order to manage in the usual way the override speed of the robot. The entire volumetric collision detection engine will be engineered in order to take into account the bulk of the tool mounted on the robot, with no limit on the capabilities of the interference regions control system to test the tool centre point against the elementary geometrical regions. The theory of collisions

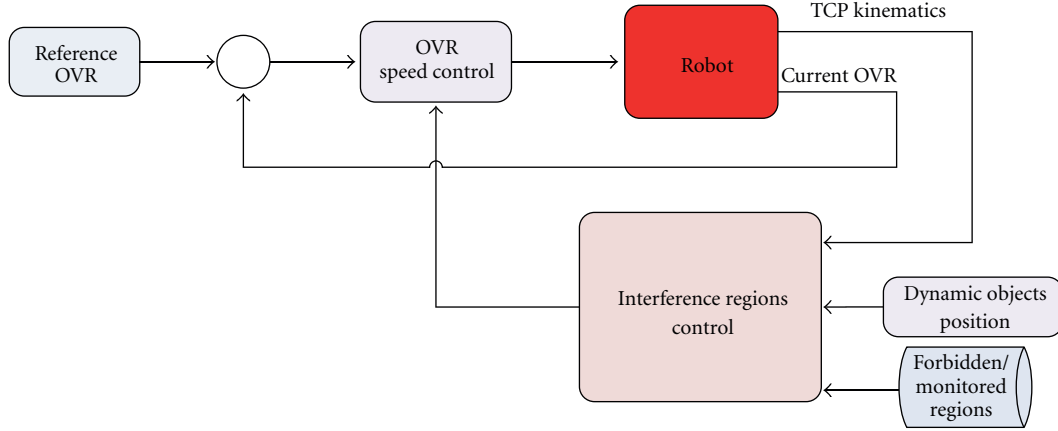


FIGURE 1: Interference Regions Control Scheme.

between volumes is here studied, in order to detect if a volume is in contact with others; furthermore, in order to correctly implement the methods of volumetric interference regions, a further module to compute the minimum distance between volumes has to be taken into account this study is strictly connected to the one related to collision detection and it will not be described here. For a deeper treatment of this topic, the reader can refer to [15]. The study of such architecture is a step towards the realization of a more intelligent robot and aware of its volume, starting from the tool mounted on its flange and giving the opportunity to extend this model to the whole robot arm.

3.1. Objects Intersection. In order to begin the analysis and study of collision detection for volumes, it is necessary to introduce some theory from computer graphics; at first we approach the method of detecting intersection between two convex shapes in two-dimensional space [16], after we will extend these results to 3D [17]. For generic geometrical objects lying in 2D space the separating axis theorem, a special case of the separating hyperplane theorem [18] states that given two convex shapes, there exists a line onto which their projections will separate if and only if they are not intersecting (i.e., colliding); a line for which the objects have disjoint projections is called a separating axis.

Another way of stating the theorem is to say that two convex shapes in the plane are not intersecting if and only if a line can be placed with one shape to one side of the line and the other shape to the other side such a separating line will be perpendicular to a separating axis (see Figure 3). In the specific case of 3D objects, the separating axis theorem can be easily extended in order to detect intersections between three-dimensional convex shapes. The separating plane theorem thus states that for any arbitrary convex disjoint polyhedra there exists at least one separating axis, where the projections of the polyhedra, which form intervals on the axis, are also disjoint.

The separating axis in 3D can be obtained considering that where a plane can be inserted between two three-dimensional objects, that plane's normal defines a separating

axis (see Figure 4). This theory is the basis to describe more complex interactions between volumes.

3.2. Volume versus Volume Collision Detection. Starting from separating axis/plane theorem, we want to model the robot geometry with convex disjoint polyhedra, in order to adapt the presented theory to robotics. Initially, we want to prevent collisions between the robot tool and the elementary geometrical regions, as defined in the previous chapter. The basic idea is to include the tool mounted on the robot flange inside a bounding box, which can be both oriented and moving rigid with the robot flange. The bounding box should be analyzed to fully describe its minimum set of variables which can be used in the computation of interactions and collisions.

The OBB (oriented bounding box) \mathbf{B} is depicted in Figure 5 and it can be described by the following set of variables:

$$\mathbf{B} = (\mathbf{b}^c, \mathbf{b}^u, \mathbf{b}^v, \mathbf{b}^w, h^u, h^v, h^w). \quad (5)$$

It is true that for each robot tool can be found an OBB which can contain it: this is an initial step towards the definition of a higher level of collision detection between solids. The tool centre point check of the interference regions control system is then substituted by an OBB check against interference regions. In order to proceed in this direction, it is necessary to analyze singularly the possible intersections between solids, in particular we have to describe intersections between

- (i) OBB versus *plane*;
- (ii) OBB versus *sphere*;
- (iii) OBB versus *cylinder*;
- (iv) OBB versus OBB.

3.2.1. OBB versus Plane Intersection. With reference to Figure 6, the analytical method to determine whether an OBB intersects a plane $\pi : \mathbf{n} \cdot \mathbf{x} + d = 0$ is to put the vertices of the OBB into the plane equation. If both a positive and

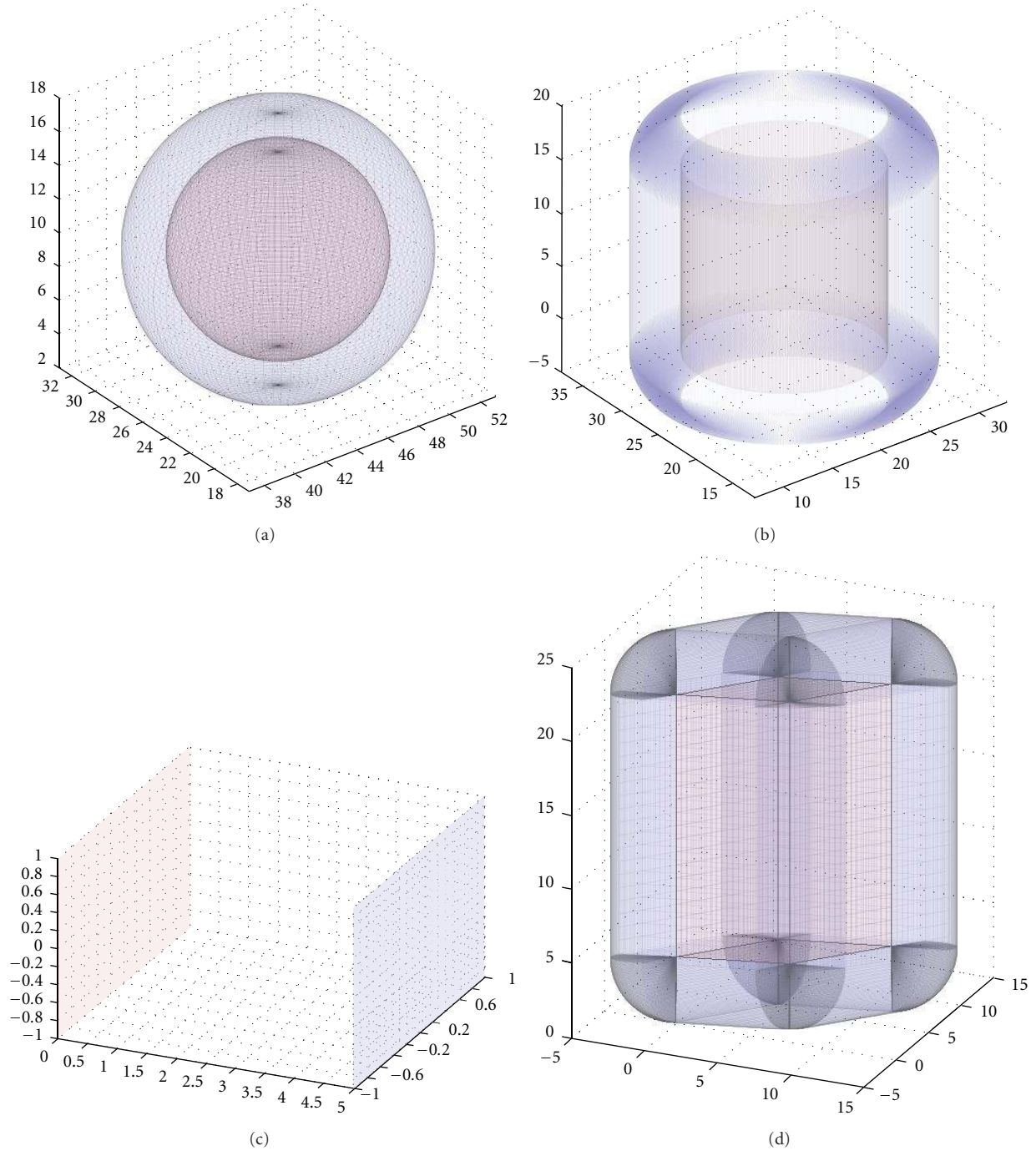


FIGURE 2: Elementary shapes (a) sphere, (b) cylinder, (c) plane, (d) parallelepiped (red), and their warning zones (blue).

a negative results (or a zero) are obtained, the vertices are located on both sides of (or on) the plane, and therefore, an intersection is detected. There are other smarter methods where only two points have to be inserted into the plane equation; for an OBB, there are two diagonally opposite corners on the box that are the maximum distance apart, when measured along the plane's normal. Every box has four diagonals formed by its corners. Taking the dot product of each diagonal's direction with the plane's normal, the largest

value identifies the diagonal with these two furthest points. By testing these two points, the box as a whole is tested against a plane. So we can assume that we have an OBB defined by a centre point, \mathbf{c} and a positive half diagonal vector, \mathbf{h} . The first step is to compute \mathbf{c} and \mathbf{h} as follows:

$$\begin{aligned} \mathbf{c} &= \frac{\mathbf{b}^{\max} + \mathbf{b}^{\min}}{2}, \\ \mathbf{h} &= \frac{\mathbf{b}^{\max} - \mathbf{b}^{\min}}{2}, \end{aligned} \quad (6)$$

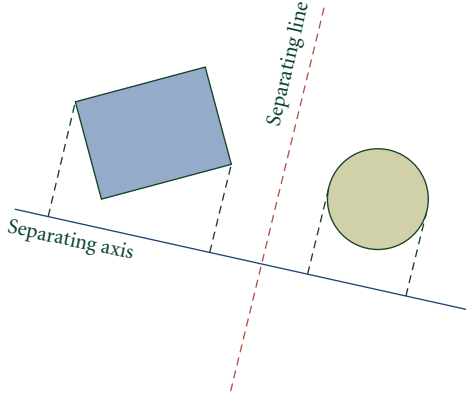


FIGURE 3: Illustration of the separating axis theorem.

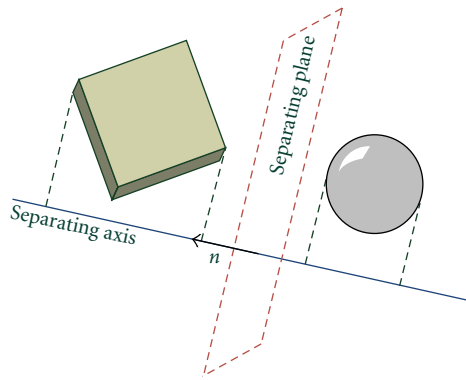


FIGURE 4: Illustration of the separating plane theorem.

where \mathbf{b}^{\min} and \mathbf{b}^{\max} are the minimum and maximum corners of OBB. Now we have to test our OBB against the plane π . In order to do that in a fast way, we can consider the computation of the extent, denoted as e , of the box when projected onto the plane normal, \mathbf{n} . The equation of the extent is the following:

$$e = h_u |\mathbf{n} \cdot \mathbf{b}^u| + h_v |\mathbf{n} \cdot \mathbf{b}^v| + h_w |\mathbf{n} \cdot \mathbf{b}^w|. \quad (7)$$

Next, we compute the signed distance, s , from the centre point, \mathbf{c} , to the plane. This can be achieved by computing $s = \mathbf{c} \cdot \mathbf{n} + d$. With the signed distance and the extent of the box, the following will state if box is inside/outside the plane region:

$$\begin{aligned} \text{if } (s - e) > 0 & \text{ then outside,} \\ \text{if } (s + e) < 0 & \text{ then inside,} \end{aligned} \quad (8)$$

This simplification is explained in [18].

3.2.2. OBB versus Sphere Intersection. The principle to detect an intersection between an OBB and a sphere is depicted in Figure 7 and it consists on finding the point of the OBB that is closest to the sphere's centre \mathbf{c} . The tests to be used are one-dimensional one for each axis of the OBB. The sphere's centre coordinate for an axis is tested against the bounds of OBB: if

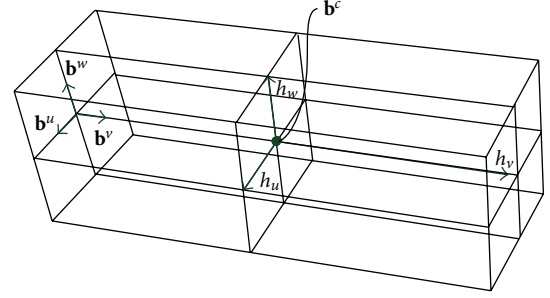
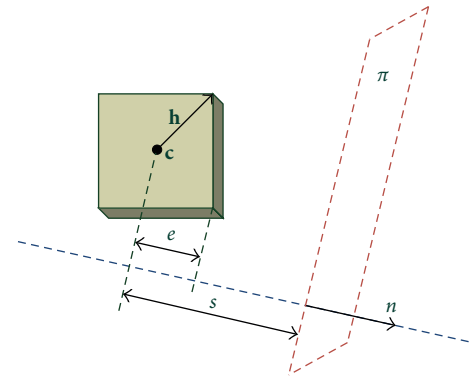
FIGURE 5: Oriented 3D bounding box: \mathbf{b}^c is the centre point, together with normalized and positive-oriented side vectors \mathbf{b}^u , \mathbf{b}^v , and \mathbf{b}^w . The vectors h_u , h_v , and h_w are the distances from the centre of the box to the centre of its faces.

FIGURE 6: Principles of the intersection test between an OBB and a plane.

it is outside the bounds, the distance between the sphere's centre and the box along this axis is computed and squared. After having executed the same calculus for each of the three axes, the sum of these squared distances is compared to the squared radius, r^2 , of the sphere. If the sum is less than the squared radius, the closest point is inside the sphere, and the box overlaps. However, this algorithm is only applicable if the OBB reference frame is parallel to the sphere's frame reference, and in order to use the presented method, it is compulsory for an oriented bounding box to first transform the sphere's centre into the OBB's space; this means that the OBB's normalized axes as the basis for transforming the sphere's centre have to be used.

The algorithm can be resumed as shown in Algorithm 1.

The tests ($e < -r$) and ($e > r$) are used here to quickly reject the possibility of any intersection between the volumes.

3.2.3. OBB versus Cylinder Intersection. With reference to Figure 8, the principle to detect an intersection between an OBB and a cylinder is quite complex and here it will be analyzed a possible implementation. First of all, we have to transform cylinder into the OBB space in order to simplify the computation. After that, the closest points, P_{OBB} and P_{CYL} , between OBB and cylinder have to be detected; a first simplification can be done, stating that if the distance

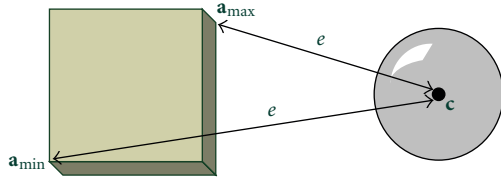


FIGURE 7: Principles of the intersection test between an OBB and a sphere.

```

d = 0
for each i ∈ {x, y, z}
  if ((e = ci - ai,min) < θ)
    if (e < -r) return (DISJOINT);
    d = d + e2;
  else if ((e = ci - ai,max) > θ)
    if (e > r) return (DISJOINT);
    d = d + e2;
if (d > r2) return (DISJOINT);
return (OVERLAP);

```

ALGORITHM 1:

between P_{OBB} and P_{CYL} is greater than r (the radius of the cylinder), then the two volumes do not intersect.

Otherwise, the algorithm should go forward computing:

$$C_{\text{cyl}} = \frac{C_0 - C_1}{2}, \quad (9)$$

Then

if $(\text{Length}(OP_{\text{OBB-per}}) > r)$ then outside,

if $(\text{Length}(OP_{\text{OBB-per}}) < \text{dist}(C_{\text{cyl}}, C_0))$ then inside. (10)

The algorithm continues stating that

if $(\text{Length}(OP_{\text{OBB-para}}) > \text{dist}(C_{\text{cyl}}, C_0))$. (11)

Then if the red vector cyl_{ray} intersects the plane and the point of intersection lies in the OBB's boundaries, then there is an intersection.

3.2.4. OBB versus OBB Intersection. The algorithm to detect intersections between two OBBs uses the separating axis theorem; the test is done in the coordinate system formed by A 's centre and axes, as depicted in Figure 9. The origin then is $\mathbf{a}^c = (0, 0, 0)$ and the main axes are $\mathbf{a}^u = (1, 0, 0)$, $\mathbf{a}^v = (0, 1, 0)$, and $\mathbf{a}^w = (0, 0, 1)$. The OBB B is assumed to be located relative to A , with a translation t and a rotation matrix \mathbf{R} .

Based on the separating axis theorem, it is sufficient to find one axis that separates A and B to be sure that they are disjoint and so do not overlap. In order to complete the test, fifteen axes have to be tested: 3 from the faces of A , 3 from the faces of B , and $3 \cdot 3 = 9$ from the combinations of edges

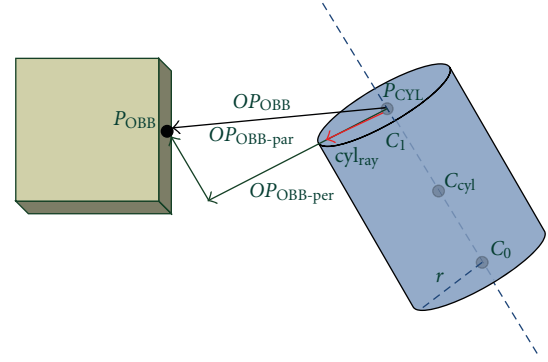


FIGURE 8: Principles of the intersection test between an OBB and a cylinder.

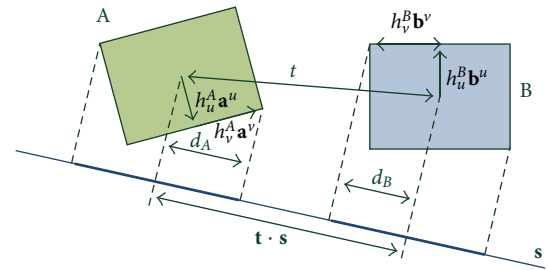


FIGURE 9: Principles of the intersection test between two OBBs.

from A and B . Assuming that a potential separating axis is \mathbf{s} , the radii, d_A and d_B , of the OBBs on the axis, \mathbf{s} , are obtained by simple projections as stated in the following

$$d_A = \sum_{i \in \{u,v,w\}} h_i^A |\mathbf{a}^i \cdot \mathbf{s}|, \quad (12)$$

$$d_B = \sum_{i \in \{u,v,w\}} h_i^B |\mathbf{b}^i \cdot \mathbf{s}|.$$

If and only if \mathbf{s} is a separating axis, then the intervals on the axis should be disjoint and the following disjoint is valid:

$$|\mathbf{t} \cdot \mathbf{s}| > d_A + d_B. \quad (13)$$

Summarizing, if any of these fifteen tests is positive, the OBBs are disjoint; otherwise, if all the tests are negative, the OBBs intersect. For an extensive and more detailed theoretical treatment of this subject refer to [19].

4. Volumetric Interference Regions

The control systems' features, for the management of robot behaviour in static environments, have been described as the capability to control the robot speed override as different changes in the configuration of the robot arise. The architectural scheme of the proposed volumetric interference regions control system is depicted in Figure 10. From this scheme, it is possible to locate the blocks where the volume collision detection and closest point detection are taken; in particular, the volume collision detection (VCD) block takes as input the tool volume kinematics and the database of user-defined monitored/forbidden regions.

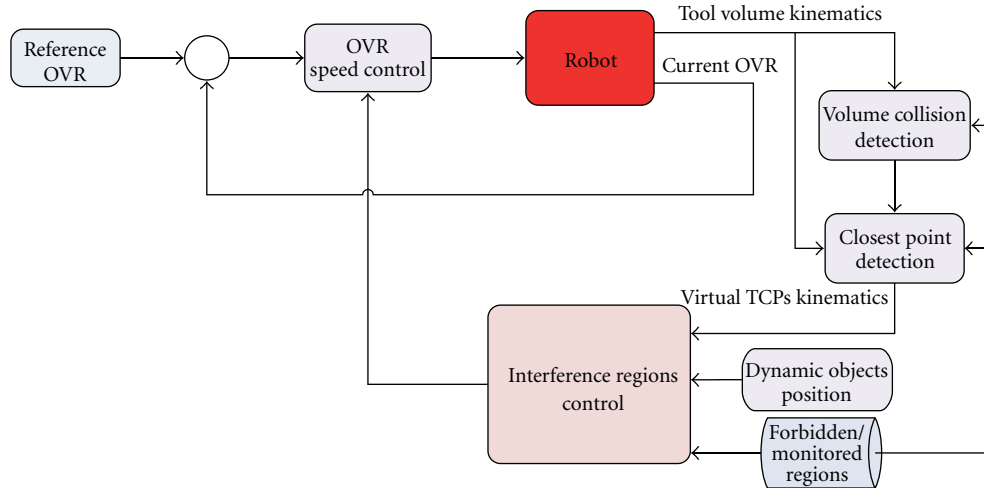


FIGURE 10: Volumetric interference regions control scheme.

The robot tool must be bounded inside an OBB; in this way, all the forbidden and monitored regions, previously defined by the user, can be successfully checked for collision. The information coming from the robot is then a complete kinematics set of the tool's OBB (i.e., its position, speed, and acceleration of its centre point); this information, together with some previous computation from the VCD and the information about the forbidden and monitored regions, are then used inside the closest point detection (CPD) block in order to detect for each region in the robot's workspace the closest point. This set of points is the output of the CPD which gives is an array of virtual TCPs kinematics (i.e., positions, speeds, and accelerations of the detected closest points). After the VCD and CPD block, the integrated solution showed in Figure 1 depicting the interference regions control scheme can be fully adopted in order to manage collisions between volumes and output a controlled speed override applied to the robot. This solution assures that the results obtained with the previous control architecture, concerning performances and effectiveness, are still valid, as showed in the next section.

5. Test Results

In the present section an overview of the tests executed on a real industrial robot will be shown. These tests refer to the robot while it is moving at its maximum speed and while performing technological tasks in a real industrial cell. The interference regions control system has been analyzed during this test session, without taking into account the management of the volumetric control paradigm. We are confident that these results can be easily obtained with the presented volumetric method since the following tests refer to the collision avoidance core which takes as input a set of tool centre point positions which are the same output of the closest point detection block previously described. Therefore, we do think that these results can be generalized for the presented solution. The robot arm used in the test is



FIGURE 11: The SMART NS16 industrial robot manipulator.

a SMART NS16 manipulator (shown in Figure 11) which is a 6-axis industrial manipulator with a maximum load at wrist of 16 kilograms and a high repeatability of 0.05 mm. During this test phase, the SMART NS16 has been programmed, as usual in industrial robotics, as if it was moving over a working path (technological move of welding process) with one active cylindrical-controlled area.

In Figure 12(a), the robot end-effector position (X,Y) is plotted for two different movements: the first one where the control is not active (the blue pattern) and the second one where the control algorithm is enabled (the red pattern). The manipulator moves from the starting position (*Start*), up to a first point A. The next movement is a Cartesian linear move towards point B, parallel to Y -axis. The two circles depicted in the figure, highlight in yellow the warning zone and the forbidden zone in red. On the other hand, when the forbidden zone is not active, the end effector moves towards the start position after reaching point B (blue pattern in

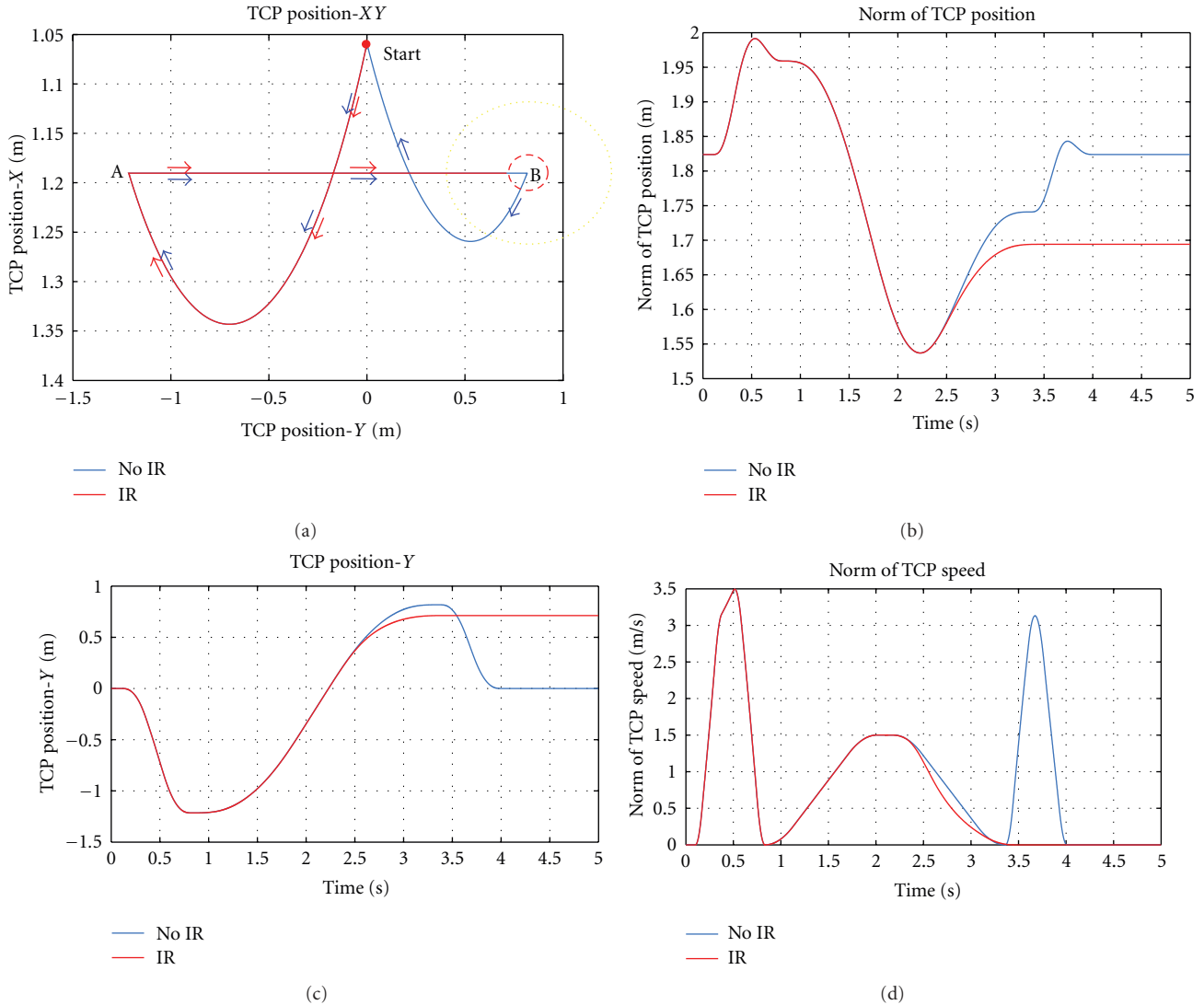


FIGURE 12: (a) Robot position (X,Y) , with enabled/disabled control, (b) norm of the end-effector position over time, (c) trend of the Y end-effector position over time, and (d) norm of the robot speed over time.

the figure). The second movement is depicted in red, and represents the behaviour of the robot end effector when the forbidden zone is enabled; the robot end effector stops inside the warning zone at the boundaries of the forbidden zone when the control is enabled.

The position trend over time is depicted in Figures 12(b) and 12(c). From Figure 12(b) it is possible to see the different trends of the norm position of the end effector when the control is enabled (in red) and when it is deactivated (in blue). The robot begins moving along the trajectory parallel to Y at 1.5 seconds, and after almost one second it enters the warning zone. In Figure 12(c), it is depicted how, without enabled warning zone control algorithm, the trend along the Y -axis continues as far as the next movement (at about 3.5 seconds); when the control algorithm is enabled, it is shown that the Y position of the robot end effector becomes constant when the end effector encounters the forbidden zone. In Figure 12(d), is then depicted how the

end-effector speed changes when it comes in contact with the warning zone at about 2.5 seconds. From these results, it comes out clear that the speed reduction is in inverse relation to the distance between the end effector and the geometrical elementary region surface. These results can be easily extended to a more complex environment where several different interference regions are present and where the control algorithm has to manage the override control for each area, even if they overlap. As stated before, the algorithm is still valid if the input of the interference regions control system is a vector of virtual tool centre points, coming from the closest point detection and from collision avoidance blocks.

6. Conclusion

In this paper, we presented a new method to extend the perceptive capabilities of an industrial robot. Although anthropomorphic robot producers have faced problems related to

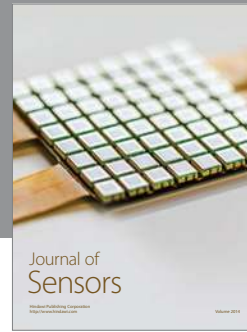
the interaction between robot and its surroundings, there is not an exhaustive study on the capabilities of a robot to be aware of its volume (in particular on the tools eventually mounted on its flange). This paper presents methods to model the space around the robot in a manner that the robot could be capable of interacting with some particular regions, here called interference regions, in its workspace. The added value of this paper is that the model of the robot surroundings is designed in order to make the robot capable of perceiving the volume of the surrounding objects through the volume of the bounding box containing its tool. The test results have been shown considering the original interference regions control architecture, and the possibility to extend them to the new architecture are remarked and expounded in order to validate the new control paradigm. The proposed solution then prove that the system is able to cope with complex real surroundings where the interactions to be checked are between volumes in order to get closer to a real industrial environment.

Acknowledgments

This paper has been supported by Comau Robotics S.p.A.; the author would like to thank the company and the control engineering team for the support and help on the experimental work and on the state of the art research.

References

- [1] F. Romanelli and F. Tampalini, "A control algorithm for the management of multiple dynamical geometrical areas for industrial manipulators," in *Proceedings of the 17th International Workshop on Robotics in Alpe-Adria-Danube Region (RAAD '08)*, Ancona, Italy, September 2008.
- [2] M. Vukobratović, D. Surdilović, and Y. Ekalo, *Dynamics and Robust Control of Robot-Environment Interaction*, World Scientific, River Edge, NJ, USA, 2008.
- [3] H. Marcuse, *One-Dimensional Man: Studies in the Ideology of Advanced Industrial Society*, Routledge, 1964.
- [4] B. Schulz, "The gift of sight," *Australian Manufacturing Technology*, pp. 86–89, 2007.
- [5] B. Rooks, "Precision robot welding for world's largest particle physics apparatus," *Industrial Robot*, vol. 32, no. 4, 2005.
- [6] S. Chunxue, B. Yingyong, and L. Jianghui, "Mobile robot path planning in three-dimensional environment based on ACO-PSO hybrid algorithm," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM '08)*, pp. 252–256, August 2008.
- [7] J. Wolfe, B. Marthi, and S. Russell, "Combined task and motion planning for mobile manipulation," in *the 20th International Conference on Automated Planning and Scheduling (ICAPS '10)*, pp. 254–257, May 2010.
- [8] M. H. Kazemi, S. Ahangarian, and A. Hashempour, "Fuzzy path tracking control of a 5-DOF experimental robot," in *IEEE International Conference on Information and Automation (ICIA '08)*, pp. 436–440, June 2008.
- [9] M. Matinfar, C. Baird, A. Batouli, R. Clatterbuck, and P. Kazanzides, "Robot-assisted skull base surgery," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '07)*, pp. 865–870, November 2007.
- [10] G. van den Bergen, *Collision Detection in Interactive 3D Environments*, Morgan Kaufmann, 2003.
- [11] E. Drumwright, "A fast and stable penalty method for rigid body simulation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 1, pp. 231–240, 2008.
- [12] J. U. Kuehnle, M. Danzer, A. Verl, and R. Bischoff, "Real-time 3D environment model for obstacle detection and collision avoidance with a mobile service robot," in *Intelligent Robots and Computer Vision XXVII: Algorithms and Techniques*, vol. 7539 of *Proceedings of SPIE*, January 2010.
- [13] M. Shahrokhi and A. Bernard, "A fuzzy approach for definition of dangerous zone in industrial systems," in *IEEE International Conference on Systems, Man and Cybernetics (SMC '04)*, pp. 6318–6324, October 2004.
- [14] F. Romanelli and F. Tampalini, "BOG and fuzzy filters based multimodal collision avoidance for industrial manipulators," in *Proceedings of the IEEE Conference on Robotics, Automation and Mechatronics*, pp. 779–784, Chengdu, China, September 2008.
- [15] M. A. Abam, M. de Berg, S.-H. Poon, and B. Speckmann, "Kinetic collision detection for convex fat objects," in *the European Symposium on Algorithms*, 2006.
- [16] P. K. Agarwal and N. H. Mustafa, "Independent set of intersection graphs of convex objects in 2D," in *Proceedings of the 9th Scandinavian Workshop on Algorithm Theory (SWAT '04)*, T. Hagerup and J. Katajainen, Eds., Lecture Notes in Computer Science, pp. 127–137, 2004.
- [17] H. A. Sulaiman and A. Bade, "Balanced hierarchical method of collision detection in virtual environment," in *the 2nd International Conference on Software Engineering and Computer Systems (ICSECS '11)*, vol. 181, pp. 493–501, 2011.
- [18] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.
- [19] T. Akenine-Möller, E. Haines, and N. Hoffman, *Real-Time Rendering*, A K Peters, Wellesley, Mass, USA, 2008.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

