We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



29

Hybrid Control Techniques for Static and Dynamic Environments: a Step towards Robot-Environment Interaction

Fabrizio Romanelli
Comau Robotics S.p.A.
Italy

1. Introduction

Nowadays industrial robots have to perform complex tasks at high speeds and have to be capable of carrying out extremely precise and repeatable operations in an industrial environment; however, robot manipulators can perform just a small set of interactions with structures in their surrounding environment and with human operators which can be eventually present in the working area. The great gap between the capability of performing tasks with a high precision and speed, and the ability to perceive the environment and to act with it, claims the need of a smart and versatile control system which must guarantee a high degree of interaction between the robot manipulator and its world, whilst assuring the same performances required by the industry processes. In this context the ability to perceive its environment forms a crucial characteristic of such control system, which has to cope with problems of incompleteness of data and uncertainty. In order to achieve a high level of interaction, the control has to provide the system with the capability of robustly perceive the robot surroundings and to promptly react to any change in the state of its environment. A fundamental aspect of the robot-environment interaction is related to the capability of the control paradigm to model the structured and unstructured environment, such as its static and dynamic features. In this chapter, the main effort is that of describing new control architectures capable of taking into account both the static and dynamic characteristics of the robot world. In order to correctly introduce the problem, a particular focus has to be done on the static and dynamic aspects of the environment, considering at a first glance the two sides as different and separated and then, the last effort has to be done to merge the control techniques together in order to give a general solution to the modelling of environment and control of the robot. As an introduction for the first aspect, and as stated before, one of the most innovative and important problem in the nowadays industrial and service robotics is that one of completely controlling, not just the robot itself with its kinematics and dynamics, but even its interaction with the space where it works (Van Wichert & Lawitzky, 2001; Kraiss, 2006). It happens very often that the manipulators have to work in spaces shared with human operators or with other robots (eventually in some interchange zones) or they have to move and operate in places with static and/or dynamic facilities (Barraquand et al., 1989). Under this point of view, the use of a system that can

control speed in a safe way (Winkler, 2007), allows to control operative areas, assuring a greater safety to human operators and between robots and machines, and it increases the efficiency of the used space as it is possible to concentrate more industrial devices in the same space with a great economical and cycle-time savings. The robots indeed can interact between them covering shorter paths, with the certainty that they will not collide, as far as the algorithm is active; at the same time, the presented paradigm (Romanelli & Tampalini, 2008a) allows the interaction of robots with other moving machines sharing the same spaces, increasing the efficiency of the used space. In addition to this, the system is capable of identifying the presence of the robot end effector inside the controlled zone or inside a larger zone called warning zone, using configurable outputs in order to communicate to the other devices inside the robot cell or to communicate to the other robots as well, if a warning or controlled zone is violated. This first paradigm gives a reference for the modelling of static industrial environment (such as interlock areas, machinery and other industrial devices) and for the control of the robot to interact with these predefined spaces. In order to model the robot surroundings, a system has been developed to manage a set of different geometrical shapes which define zones where the movement and access is forbidden or allowed; for the control system, the warning zone has been introduced. It is defined as a thickness from the controlled zone which is the core of the control system as it intrinsically defines the control for the general speed override of the robot end effector: an opportune control law has been studied in order to cope with particular geometrical conditions where more than one different shapes have to be controlled simultaneously. The proposed model and control has been applied and extended to dynamic objects moving in the robot working area (such as conveyors and rails which have a known trajectory) as well: the spatial checks and control in this case are performed on one or more moving geometrical zones. This aspect links the static model of the environment to the dynamic one, and puts the basis for a more general control paradigm. The dynamic and unstructured aspect of the problem instead has been analyzed in a slightly different way, taking into account the behaviour of the obstacles moving inside the robot working area without any prior knowledge, such as human operators working in tight connection to the robot; occupancy grids (Moravec, 1988; Elfes, 1989) have been taken into account in order to face this problem. They are used in order to tessellate the space (i.e. the operating area of the robot) in regular cells, and to store in each cell fine grained, quantitative information. With Bayesian occupancy grids (Collins et al., 2007), the idea is to extend the meaning of the value contained in each cell to the probability of that cell being occupied by an object. The nature of the decomposed space may be Euclidean space or a higher dimension state-space which could take into account velocities, accelerations and orientations as well. Such maps are extremely useful for robotic applications, such as obstacle/collision avoidance. In this kind of applications, the problem of the uncertainty of the information given by the sensors (proprioceptive or exteroceptive) is one of the biggest in this field. Such paradigm, using the Bayesian occupancy grids, face the problem in a very efficient way, as it models the unreliability of the measurements with probability. Another advantage of the use of occupancy grids is that they allow sensor fusion to be performed in a flexible way even if the system presents different typologies of sensors (even with very heterogeneous sensor models). Fuzzy logic (Zadeh, 1965, Klir & Yuan, 1965; Mamdani & Assilian, 1993) control has been chosen in order to take into account the behavioural aspect of the interaction between robots and human operators. Fuzzy logic controllers of particular interest are those used in Antilock Braking Systems (Mauer, 1995),

in camera applications and where robots or automatic systems have to carry out behavioural tasks, such as collision avoidance and path planning (Kim et al., 1999). The paradigm concerning the dynamic aspect of the environment is based on the decisional and behavioural component on a fully reactive system based on Fuzzy logic controllers. The information about obstacle position around the working area related to the robot end effector, are computed in order to establish which kind of behaviour has to be taken and how it has to be applied. It is therefore possible after a proper adjustment of the control, to synthesize a system capable of acting with complex strategies, based on a simple set of behaviours; the result of this paradigm is a control which expresses precisely qualitative concepts, defined formally in terms of mathematical functions, called membership functions. Late in this chapter a new approach to deal with collision avoidance in dynamic environments is proposed. In the industry sphere the problem of collision and obstacle avoidance is relevant as the interactions between humans and machines are closer and closer. This is an important aspect which is matter of studies in the field of robotics and automation. In this context the basis idea of this chapter is to give a first step towards integration between the work of humans and robots; this integration can't be set aside of security which is the most relevant aspect of the problem and which has been taken into account during this study as a first requirement. The obstacle avoidance algorithm proposed (Romanelli & Tampalini, 2008b) is based both on a probabilistic framework, such to make the connection between the sensorial perception and the control of the robot, and on a polyvalent logic framework. There are no particular restrictions to the exteroceptive sensorial input model to the system, as the uncertainty of position of the obstacles given from the sensors can coexist in the same system, as the probabilistic framework also gives a good instrument to obtain sensor fusion. This method, which is efficient for medium-low distances obstacles, was combined with a fuzzy logic engine, which is very efficient for a medium-high distances obstacles and it is well adaptable to define politics to decide the reference override speed in function of heading. The advantages of utilizing a combination of the two approaches is that the robot override speed can be controlled, acting with both the controls in a continuous and smooth way. This control law takes into account both the trajectory of the obstacles moving around the robot area and the behaviour of the moving objects. The advantages of the aforementioned algorithms for the management of both static and dynamic environments have been merged in a hybrid control system, to make it capable of interacting with static objects (such as industrial facilities), objects with structured dynamics (i.e. objects bound to move on predefined paths, such as rails) and objects with unstructured dynamics (such as humans moving around the robot area). The results of the proposed control technique have been tested over a simulated system, for both the static and dynamic aspects; experiments on a real system have also been carried out, limited to the interaction between robot and structures, due to security reasons. Late in the chapter it will be showed how the presented approach can be extended in order to take into account other cognitive features.

2. Robot-Environment Interaction

Creating autonomous robots that can learn to act in unpredictable environments has been a long standing goal of robotics, artificial intelligence, and cognitive sciences. Robots are meant to become part of everyday life, as our appliances, assistants at home, and in

particular in industrial environments, co-workers at the workplace. Nevertheless, to get robots operating outside research centres or universities 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. This would provide robots with certain level of independence and dynamic behaviour. As a first step to face the problem of robot interaction with the environment it is important to understand and observe the interaction of humans with the environment in order to make the robot acting in a similar manner to what people do while moving in the same environment. A fundamental step to make the robot acting as a human is to define the typology of sensors it needs to perform a complex task, with enough accuracy and with robustness to unpredictability. The robot interacts with the environment in different ways: it acquires information from the environment through its sensors to provide the necessary input signals to the controller and it performs actions in its surroundings in order to achieve the desired tasks. The fundamental of interaction here is that sensing and acting are coupled dynamically and can not be analyzed independently since the perception of the robot influences the actions of the robot and the actions of the robot influence how the robot perceives the environment. This interaction exhibits complex and unpredictable characteristics and it is very difficult to identify the whole system using a generic method. In order to give the basis for a simplified theory, the analysis can start with the following assumptions:

- The robot controller is reactive where the output of the controller does not depend on the internal states of the controller, but only on the current input signals provided to the controller.
- The robot works and operates in a controlled environment with no other externals factors which influence the environment. Therefore it can be assumed that, when the robot performs actions in the environment, the change in the perception of the robot will be only dependent on the actions of the robot.

The entire robot-environment interaction can be described in a complete form using two models, under the previous assumptions: 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. In this chapter the attention will be focused on the robot controller model. There is also an important social aspect that has to be taken into account when developing new theories for automation and robotics in the industrial society; 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 (Marcuse, 1964). 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 co-workers”. In this work, the interaction between robot and environment has been studied in order to also take taking into account the role of the human operator in the manoeuvring of robots, making his role more integrated to the

productive process; this is possible, considering the overall increase in the interaction between robot and environment which is the basis idea of the present chapter. Furthermore the operation of a robot, as robot-environment interaction, is governed by three major components: the *robot* itself, its sensors, actuators and general hardware morphology, the *environment*, its perceptual properties and environmental conditions, and the *task*, the control program being executed on the robot. Given this, the behaviour of the robot emerges through the interaction of these three aspects. Any theory of robot-environment interaction will be dependent upon quantitative descriptions of the robot's behaviour. In order to assert that the robot-environment interaction is more influenced by the control program (the task) than the environment, here the chaos theory will be applied to describe the robot's behaviour quantitatively based on the considerations and results made in (Nehmzow & Walker, 2005). As a first approximation, it can be assumed that the robot's trajectory encapsulates the important aspects of the robot's behaviour; so this theory focuses the attention on the application of dynamical systems theory to the analysis of robot trajectories. One of the most distinctive characteristics of a chaotic system is its sensitivity to a variation in the system's variables, so that if two trajectories that started close each other will diverge from one another as time progresses, the more chaotic system, the greater the divergence. The Lyapunov exponent λ in (1) represents a measurement of chaos, so the larger the positive Lyapunov exponent, the quicker knowledge about the system is lost.

$$\lambda = \lim_{n\to\infty} \lim_{E_0\to 0} \frac{1}{n} \sum_{k=1}^{n} \log \left| \frac{E_k}{E_{k-1}} \right| \tag{1}$$

Where E_0 is the initial error, and E_k, E_{k-1} is respectively the error at time *k* and *k-1*. From the experiments conducted by Nehmzow and Walker, the robot was firstly asked to conduct two different tasks in the same environment, and the computed Lyapunov exponent differed between the two tasks, signifying that the overall behaviour of the robot differed between the two experiments, and that this must have been due to the changes in control program. On the contrary the other experiment was conducted varying the environment and keeping the same task active. In this condition the Lyapunov exponent was the same in any of the changed environments, showing that the robot-environment interaction is far more influenced by the control program than the environment. This is an important result for the study of robot-environment interaction in the context of industrial robotics where the tasks are very often repetitive and where the environmental conditions can vary over time (as the presence of activated or deactivated controlled zones or the presence of human operators in the cell): this means that the obtained results greatly depend on the particular task the robot has to accomplish more than by the changes in its environment (e.g. changes in temperature which can affect motor dynamics, etc.).

3. Static environment modelling

In order to define a correct and effective model of the static features inside an industrial environment it is very important to define the possible set of structured objects inside a robot cell (which can be both static and dynamic). After this first step, where a general model can be proposed and analyzed in order to cover all the possible scenarios inside a real

environment, it is necessary to synthesize a control system capable of taking into account these predefined geometrical areas, which can be both forbidden and allowed, in order to provide the robot with the fundamental tools to face an entry 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 (taking into account this first modelling together with the dynamic environment modelling). In literature this topic is still at a basis level, and there are a lot of starting points of study: in the industrial automation background, PILZ developed a brand new control system for areas crowded with robots, machines and humans (Schulz, 2007). With this system, made of a safety camera capable of identifying access to forbidden areas, it is possible to send an output to safety devices in order to immediately stop the machines operating in those areas, avoiding harmful situations. Another application of this feature is the one made by ABB which implements a world zone management system made as follows: it is possible to define volumes where the robot presence is avoided. If the robot end effector is inside the allowed working space, the robot keeps working; if the robot end effector ends in an off-limits area, previously defined by the programmer, the control cuts the power and immediately stops the robot. This system lets the user program world zone software which is especially useful when two robots are working in close proximity to prevent collision and establish working protocols (Rooks, 2005). In 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 to the problem of forbidden zones (Red et al., 1987; Brooks, 2003; Roy & Pratihar, 2003). The management of operative space is a matter of study and development in the field of telemanipulation and robot assisted tasks (Matinfar et al., 2007), 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 (Canny, 1984) where given two general polyhedra of complexity n, one of which 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 (Barak & Witkin, 1992; Bouma & Vanecek, 1992), 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 (Shahrokhi & Bernard, 2004). There is a further approach that will be presented later on this chapter, 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 (Romanelli & Tampalini, 2008b). In the following paragraphs, a new method to synthesize a control system capable of managing a set of predefined geometrical areas will be showed; with this paradigm, the advantages of taking into account a space model will be showed as well.

3.1 Management and control of multiple geometrical areas

The main problem for managing the robot working space, in particular when this space is shared with other robots or machinery, is the modelling of the robot surroundings. The first step to give the basis for a system capable of managing and control the interaction with the environment has to be the definition of primitive geometrical areas in order to cover all the

possible configuration of the objects inside an industrial cell. The elementary objects are defined as parallelepipeds, cylinders and spheres (as depicted in Figure 1).

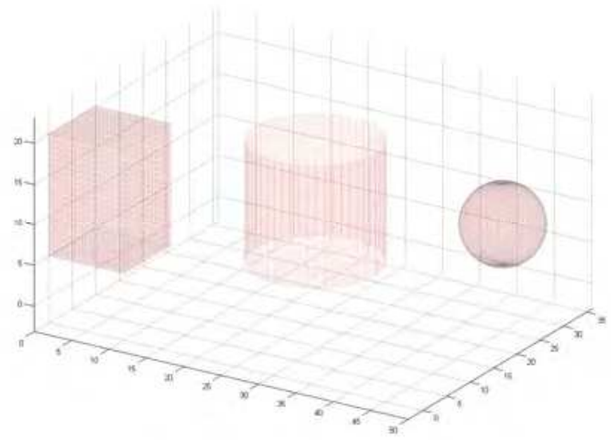


Fig. 1. Elementary geometrical shapes to model robot environment

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 showed. When this control system is active, it is possible to move the robot around the operating area with the certainty that, if the forbidden areas were previously defined and activated, if the robot end effector enters that zone, it will be immediately stopped (or similarly a digital output can be raised).

3.1.1 Control system

Starting from the elementary geometrical areas previously defined, the system allows the programmer to define multiple elementary zones which can be integrated inside the system and which represent the database of spatial forbidden areas which 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 that for the parallelepiped it is sufficient to define 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 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. Thanks to the possibility of defining several zones in the same operating area, it is clear that a great part of the industrial applications can be covered by the use of 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 can not 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 can not 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 where a network of robot controllers is present and where the information about the elementary areas present 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 in respect to the position of the multiple elementary areas defined on the cell. Another important feature of the presented system is that one which allows the user to define a further area bigger than the elementary one, 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; with this control active, 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, with this system is possible to program areas which can change their position over 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, to which the dynamic zone has to be attached, has to be known as 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 of the classic management of interlocks. Thus 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 defining dynamical interlocks, which can be shared through a network between robots, giving a global visibility in the whole cell. The kinematics information about the 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.

3.1.2 Proposed integrated solution

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. With 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 and to locate the distinctive points of the geometrical shapes which have 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 allows the control system to perform complex operations of the spatial checks on the robot working area. With the proposed 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 2.

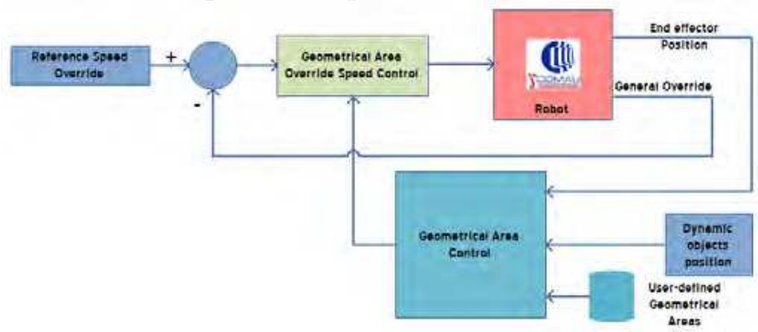


Fig. 2. Architecture of the multiple geometrical areas control system

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

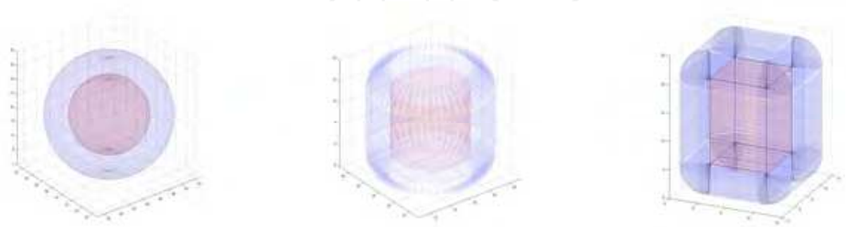
be applied in order to modify the robot override, preventing collisions with the user-defined zones. The speed override is changed smoothly when the robot end-effector comes up against a spherical elementary zone, according to the following control law:

$$\begin{cases} v = v_o \cdot \dfrac{d-r}{\delta}, & r \le d \le r+\delta \\ v = v_o, & d > r+\delta \\ v = 0, & d < r \end{cases} \tag{2}$$

Where v is the actual speed override of the robot end-effector, v_o is the past 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 3.a). When the robot encounters a cylindrical elementary zone its speed override is subject to the following control law:

$$\begin{cases} v = v_o \cdot \dfrac{d_1 - r}{R-r}, & 0 \le z \le h \\ v = v_o \cdot \dfrac{d_2}{\delta}, & h < z \le h+\delta, -\delta \le z < 0, p \in cyl \\ v = v_o \cdot \dfrac{d_3}{\delta}, & h < z \le h+\delta, -\delta \le z < 0, p \notin cyl \end{cases} \tag{3}$$

Where h is the height of the cylinder, p the position of the robot end-effector and R is r+δ. 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 3.b with its warning zone.



3.(a) 3.(b) 3.(c)

Fig. 3. Elementary shapes a) sphere, b) cylinder, c) parallelepiped (red) and their warning zones (blue)

The last modelled elementary geometrical area is represented by the parallelepiped; 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, but it is enough to consider that this control law aims at smoothly covering 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:

$$\min(v_1, v_2, \ldots, v_M) \tag{4}$$

Where M is the number of elementary geometrical areas 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; finally the bottom base centre of the parallelepiped will represent its characteristic point.

4. Dynamic environment modelling

In this section an effective and robust method to model the dynamic features in the industrial environment is described. As a first assumption, this model needs as an input the position over time or the trajectory of the objects which move around the robot working area. These can be identified as the inputs coming from different sensors (laser or camera scan and so on) or coming from more sophisticated devices like trackers (Harville & Li, 2004). The position of the multiple tracked objects is passed over time to the control algorithm which computes the correct speed override according to the combination of the Bayesian occupancy grid controller (Moravec & Elfes, 1985; Fulgenzi et al., 2007; Vasquez et al., 2006) and the fuzzy logic filter (Dong et al., 2005; Yen & Pfluger, 1995; Malhotra & Sarkar, 2005). In order to model the uncertainty coming from the sensors, a valid framework has to be taken into account; the Bayesian framework has suited models to cope with the uncertainty on the position of the obstacles but it also has an intrinsic capability to perform sensor fusion. In this context, a powerful instrument to face the problem of dynamic modelling of the space surrounding the robot is the Bayesian occupancy grid, a tessellated 2D grid in which each cell stores its probability of occupation. The behavioural side of the approach, which is necessary in order to model complex and unpredictable trajectories (like that of humans), is given by the fuzzy logic control which is very efficient for obstacles at medium-high distances and it is well adaptable to define politics to decide the reference speed override in function of heading. In this control system, sensor observations are processed from both the Bayesian occupancy grid algorithm and the fuzzy filter and the results of the computation are given as input to the collision avoidance algorithm. The

combination of the two methods is effective as the robot speed override can be changed acting with both the controls in a continuous and smooth way; this control law in fact takes into account both the trajectory of the obstacles moving around the robot area and the behaviour of the obstacles. In the following paragraphs the two frameworks will be shown, examining the characteristics of both methods and comparing the advantages of the integrated solution in respect of the single solutions.

4.1 Bayesian control technique

The occupancy grid framework is based on the division of space (both Cartesian and multidimensional, taking for instance into account speed, acceleration and orientation as well) into cells. The probabilistic approach applied to the occupancy grid paradigm gives the possibility to extend the concept of cell value: if applied to obstacle or collision avoidance this value can fit well with the probability that the cell of the grid is occupied by an obstacle. Given as input for the algorithm the position $X = [x,y]^T$ of each obstacle, or likewise (ρ,θ), Bayes' theorem states:

$$P_c(Occ \mid X) \propto P_c(X \mid Occ) \cdot \hat{P}(Occ) \tag{5}$$

Where $P_c(Occ \mid X)$ is the probability that the cell of the grid is occupied by an obstacle, given the measurement, and the right side member of (5) is a distribution of probability, and it is shaped as Gaussian multimodal distribution as shown in the following:

$$P_c(X \mid Occ) \cdot \hat{P}(Occ) \propto \aleph(\mu, \Sigma) \tag{6}$$

Where $\mu = [\mu_1,\ldots, \mu_N]^T$ and Σ is the covariance matrix (positive-definite real N by N matrix). The probability density function is defined as follows:

$$f_X(x_1,\ldots,x_N) = \frac{1}{(2\pi)^{N/2}\left|\Sigma\right|^{1/2}} \cdot e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)} \tag{7}$$

The formula in (7) describes the probability density of an obstacle, in each point of the space $\mathbb{R}^N$ where (N = 2). In order to extend Bayes' theorem to more than one obstacle, assuming that all the events (obstacles) are independents, the generalized union probability theorem can be used:

$$\begin{aligned} P\left(\bigcup_{i=1}^{n} A_i\right) = \quad & \sum_i P(A_i) - \sum_{ij}{}' P(A_i \cap A_j) + \\ & + \sum_{ijk}{}'' P(A_i \cap A_j \cap A_k) - \ldots + \\ & + (-1)^{n-1} P\left(\bigcap_{i=1}^{n} A_i\right) \end{aligned} \tag{8}$$

This theorem states that if the probability that an obstacle is occupying a cell is independent from the others (which is reasonable for the problem), the union probability can be expressed in a closed form. Under this point of view, this method is a good approach to the obstacle avoidance problem since, besides the possibility to solve the problem of modelling multi-object space occupancy, it also faces the problem of sensor fusion, as the structure of Bayesian occupancy grid is well suited for the integration of different typologies of sensor measurements (Stepan et al., 2005). The algorithm is structured as follows:

1. At the beginning, the occupancy grid is initialized with a 0.5 probability of occupation (when no information is yet available from the field);
2. As a new measurement is available, the grid is updated following the Bayes' rule described in (5);
3. The grid is further updated using the generalized union probability theorem, in order to merge together all the obstacles in the robot area;
4. Back to step 2.

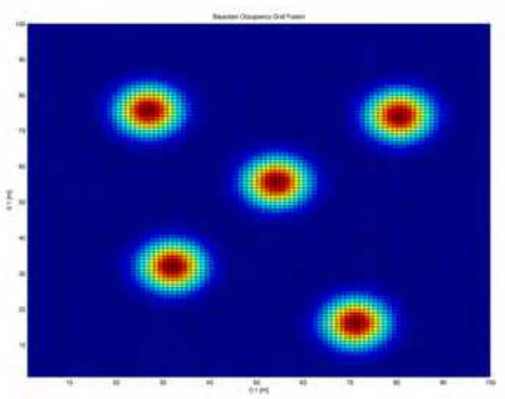


Fig. 4. Bayesian Occupancy Grid: the probability of a cell occupation grows from blue to red

Figure 4 shows an example taken from a simulation of a Bayesian occupancy grid, where 5 obstacles are moving around the robot area.

4.2 Fuzzy logic control technique

In the context of collision avoidance problem and robot-environment interaction the pure reactive systems could be a good solution to face these problems, even because they require few computational resources. Other advantages of purely reactive systems are the following:

- Emphasis on the importance of a tight relationship between perception and action;
- Absence of abstract knowledge and symbolic reasoning;
- Vertical decomposition of the problem into sub-problems to be executed in parallel;
- Modularity of the software;
- Architectures are often inspired by theory from several disciplines.

Following the classis outline (Brooks, 1986), the presented fuzzy logic controller is composed of two components: the functional module and the behavioural module. The

functional component, depicted in Figure 5, acquires information which will be then used as an input to the engine.

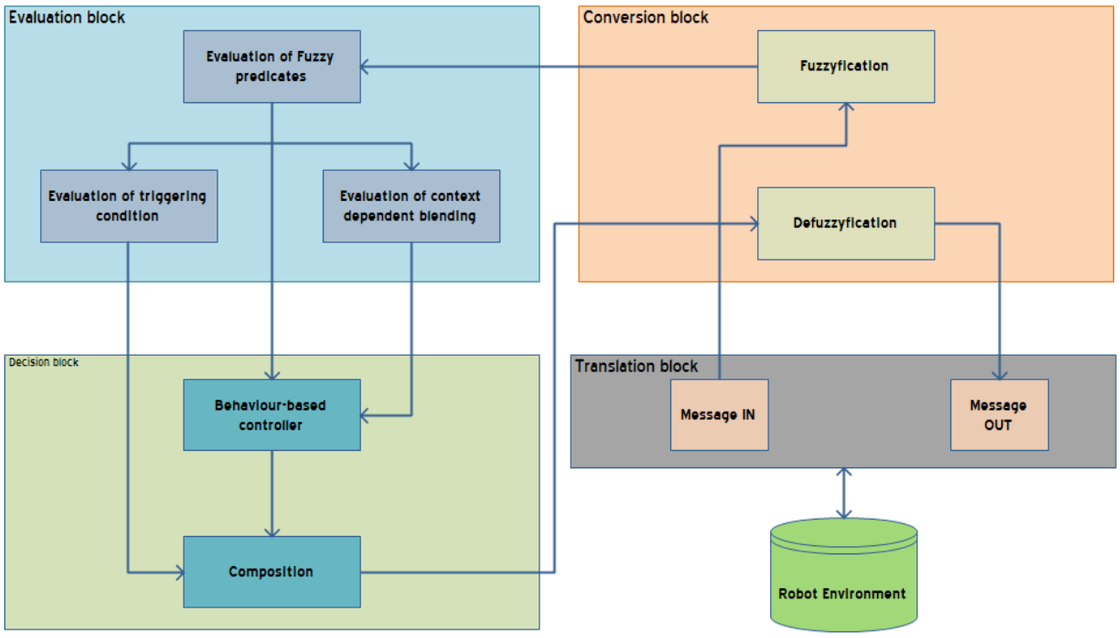


Fig. 5. Functional view of the Fuzzy logic controller

As these data have been computed, the functional component blends resulting actions and transmits values to the actuators and is composed of:

- Translational block: it is an interface between the environment surrounding the robot and data processed inside the engine. It transforms all the information from world frame to robot frame coordinates;
- Conversion layer: information acquired from translation layer is here transformed into fuzzy values (fuzzyfication process) and, after computation, output fuzzy values are transformed again into crisp values (defuzzyfication process);
- Evaluation layer: it is composed by three sub-modules, each one managing a sub-tree. Predicates sub-tree, behaviour triggering conditions sub-tree and behaviour evaluation sub-tree;
- Decision layer: decides actions to be carried out on the basis of environment information that is provided by previous layer. Behaviours are entities totally independent from each other and from the environment, describing activities to be carried out.

In respect of the classical implementation of the Fuzzy logic controller, the proposed solution does not have the purpose of determining type and position of the obstacles in the environment. The information about distance [*mm*] and heading [°] should be generated by the Bayesian occupancy grid controller. Fuzzy rules are the basis where the operative knowledge of the robot can be built from a human heuristic knowledge. A template of a Fuzzy rule can be the following: *IF* <*antecedent*> *THEN* <*consequent*>. Where the antecedent could consist of an arbitrary large number of precondition combined through logic operators AND, OR and NOT. So for instance, the following rule: *IF ((obstacle ∉ North) AND*

(obstacle ∈ Far)) THEN (speed ∈ Fast). This Fuzzy rule states that if an obstacle is present in the working area and if it is far and north of the robot, then the robot must advance pretty fast. Obviously, while in the antecedent all the aforementioned logic operation can be used, in the consequent only the AND operator is acceptable. Moreover, credibility value (i.e. the membership degree of a variable to the membership function) range between 0 and 1, both included. This implements that *T-norm* and *T-conorm* are the AND and OR operators of classic logic, where *T-norm* is:

$$\begin{matrix} \min(x, y) \\ x \cdot y \\ \max(x + y - 1, 0) \end{matrix} \tag{9}$$

And where *T-conorm* is:

$$\begin{matrix} \max(x, y) \\ x + y - x \cdot y \\ \min(x + y, 1) \end{matrix} \tag{10}$$

The fuzzyfication, blending and defuzzyfication blocks in the functional engine scheme are depicted in Figure 6. The effective engine component is labelled *Inference*. This is the scheme chosen for the Fuzzy logic controller component, in which the block that evaluates the triggering condition is canned before the behavioural sub-tree (in order to avoid wasting computational resources) and there is a blending block for each behaviour. The purpose of activation threshold is to state the effective possibility that the robot behaviour is not going to change. In this way the computational load is decreased because the engine is not forced to scan all the rules. Another important component is the blending block that fuses the outputs of the basic behaviours: this block allows the coexistence of behaviours even if there are conflicting tasks to be performed.

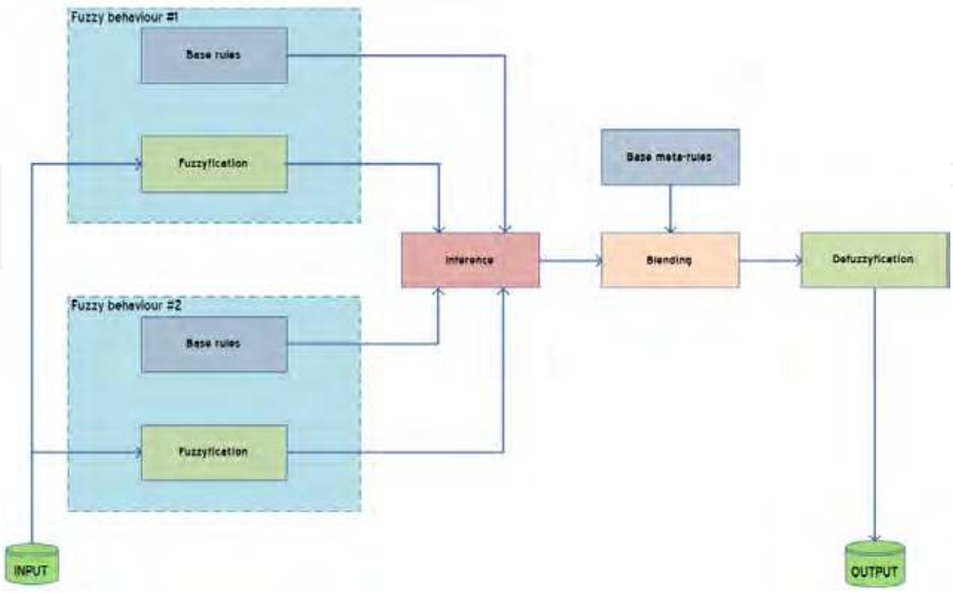


Fig. 6. Fuzzy inferential engine scheme

In order to implement the arbitration two different strategies can be chosen: a strategy in which there is for every fuzzy rule a blending block or the strategy that has a blending block for each basic behaviour.
Blending rules are contained in the component labelled meta-rules base in Figure 6. Formally:

$$\begin{aligned} Des_{B_i}\left(\underline{x},\underline{c}\right) &= \max_k \left\{ \left\{ \min \begin{bmatrix} d_1^k \cdot \mu_{A_1}^k(x_1), d_2^k \cdot \mu_{A_2}^k(x_2), \dots \\ \dots, d_N^k \cdot \mu_{A_N}^k(x_N) \end{bmatrix} \right\}, \mu_C^k(\underline{c}) \right\} \\ Des_{B_i}\left(\underline{x},\underline{c}\right) &= \max_k \left\{ \min \left\{ \min \begin{bmatrix} \mu_{A_1}^k(x_1), \mu_{A_2}^k(x_2), \dots \\ \dots, \mu_{A_N}^k(x_N) \end{bmatrix} \right\}, d^k \cdot \mu_C^k(\underline{c}) \right\} \end{aligned} \tag{11}$$

Where Des_{Bi} is the deliverability function (Ruspini, 1991) and where $d_1^k, d_2^k, \dots, d_N^k$ is the relative weight of the k-th meta-rule.

4.3 The integrated solution

In this paragraph the developed algorithm and the proposed solution to interact with dynamic unstructured environments will be showed. The Bayesian occupancy grid is a probabilistic method that models the space occupied by obstacles on an environment. The input of the occupancy grid is the position of each object (given either as a Cartesian position or as a ρ, θ representation); this position is considered as two-dimensional, in order to better fit with the readings from the sensors (such as camera trackers or laser scanners). The grid is relative to the space surrounding the robot, as the industrial robot manipulator is supposed to be always in a fixed position; a simple extension of the proposed control paradigm can be also taken into account, considering a non-fixed position of the robot. The grid is then mapped on the real working area of the robot, and it is possible to choose a different resolution of thickness of the grid, in order to achieve more accuracy on the possibility to have an obstacle in a cell. As it can be supposed that the obstacles are humans (i.e. workers that can interact with the manipulator around its working area) and that they can be seen as input from cameras, as an (x, y) position on a plane, or from laser scanners, as distances and angles, to the Bayesian occupancy grid algorithm, the two-dimensional occupancy grid has to be extended to a three-dimensional occupancy grid, since the robot end-effector is given each instant as a position and orientation in the space. In order to extend the 2D Bayesian occupancy grid to 3D, each obstacle can be modelled as a cylinder of probabilities, where the centre is given by the mean value of the Gaussian distribution of the obstacle. At the same time, the Bayesian occupancy grid override speed is computed by the algorithm: this is a quite simple operation, as the Bayesian occupancy grid gives a probability framework which is well suited for the problem of controlling override speed of the robot end-effector.

$$O_{BOG} = 1 - P_c(occ \mid X) \cdot K_s \tag{12}$$

The Bayesian occupancy grid override speed O_{BOG} is computed as the inverse of the probability that a cell is occupied given the measurement, multiplied by a constant K_s, which is the weight given to the probability that an obstacle is occupying the cell. The algorithm also gives as input for the Fuzzy logic controller, the distance and angle of the nearest obstacle to the robot. The Fuzzy logic controller takes the distance and angle computed from the Bayesian occupancy grid algorithm and, taking into account the behaviours of the nearest obstacle, computes the override speed according to the defuzzyfication process. The O_{FLC} is computed taking advantage of the behavioural nature of fuzzy logic filter. This override speed is then used by the control in order to compute the override value which has to be assigned to the robot. This value is computed considering the general override at time $t-1$, which is the value used to weight the Bayesian occupancy grid and Fuzzy logic controllers overrides.

$$\begin{aligned} O_{gen}(t) \quad = \quad & O_{gen}(t-1)\cdot O_{FLC}(t) + \\ & +(1-O_{gen}(t-1))\cdot O_{BOG}(t) \end{aligned} \tag{13}$$

The meaning of using the past general override $O_{gen}(t-1)$, as shown in (13), as a weight is that the control law gives more importance to the Fuzzy logic control algorithm when the general override speed is high (i.e. the obstacle is far from the robot), and more importance to the Bayesian occupancy grid algorithm when the general override speed is low (i.e. the obstacle is near). This typology of control has two main behaviours: when the obstacle is far, and the general speed override is high, the Fuzzy logic controller acts as the main control, since it allows behaviour based on the heading of the robot end-effector towards the obstacle. In this state the decisional policy is submitted to the Fuzzy logic controller as the distance from the obstacle is high: for instance while the robot end-effector is moving and an obstacle is moving away, behind the heading of the end-effector, the Fuzzy logic override speed will be fixed to the maximum speed, if there is no obstacle in front of the heading of the robot end-effector. On the other hand when the obstacle is near, and the general speed override is low, the Bayesian occupancy grid algorithm acts as the main control, as it allows a behaviour based on the probability of encountering an obstacle in a portion of space, then adjusting the Bayesian occupancy grid override with an appropriate control law. In this state the decisional policy is submitted to the Bayesian algorithm as the distance from the obstacle is low, and a good accuracy in the choice of the correct speed override is needed by the control constraints. It is then also important to emphasize the way the control behaves, as the switch between the two typologies of control techniques is entirely smooth. This means that both the Bayesian occupancy grid and the Fuzzy logic controllers act simultaneously in every condition; the advantages of this technique is that it merges together the advantages of each control law and that, taking advantage of feedback on general speed override, the control is fast, robust and ready.

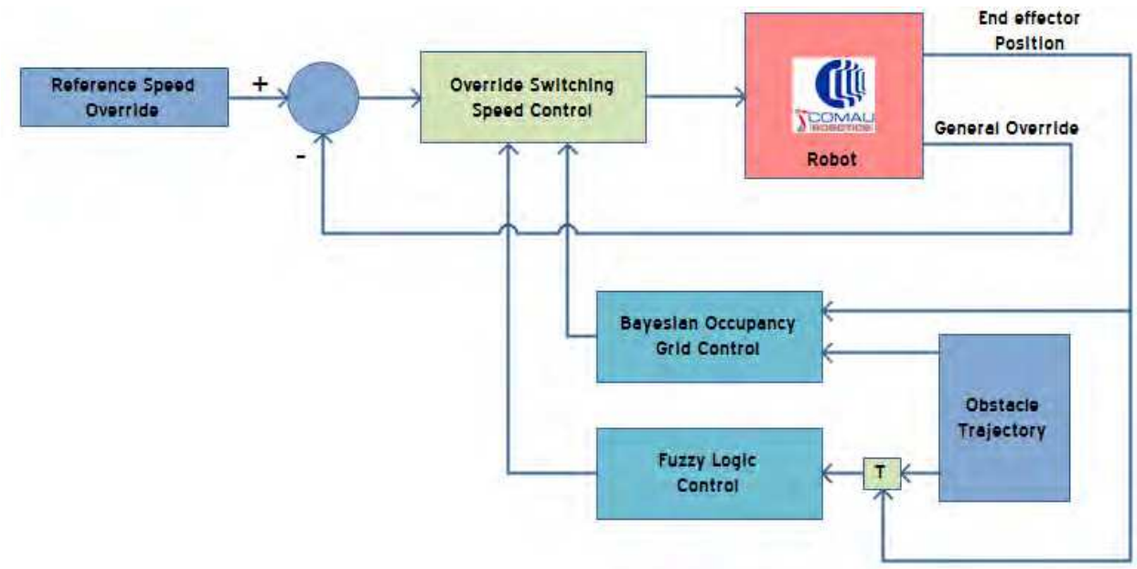


Fig. 7. Architecture of the control system for dynamic environments: robot is controlled by the feedback controller from the output (general speed override)

The proposed algorithm acts as follows (Figure 7):

1. At the beginning, the Bayesian occupancy grid and Fuzzy logic controller variables are initialized;
2. As a new measurement is available, the occupancy grid is updated;
3. The grid is further updated using the generalized union probability theorem, in order to merge together all the obstacles in the robot area;
4. The Bayesian occupancy grid override, the distance and angle of the nearest obstacle is then computed and passed to the Fuzzy logic controller;
5. The Fuzzy logic controller computes the speed override according to the defuzzyfication process;
6. The Bayesian occupancy grid and Fuzzy logic controller partial speed overrides are used to compute the general speed override according to (13);
7. Back to step 2.

5. Hybrid control paradigm

The features of the control systems, for the management of robot behaviour in static and dynamic environments, have been described as the capability to control the robot speed override as different changes in the environment arise. The architectural scheme of the proposed hybrid control system is depicted in Figure 8.
From the scheme it is possible to locate the three blocks where the decisions about the overrides related to static and dynamic models are taken; in particular the geometrical area control block takes as input the position of the robot end-effector, the dynamic objects (to be eventually connected to the elementary geometrical areas) and the database of user-defined geometrical areas.

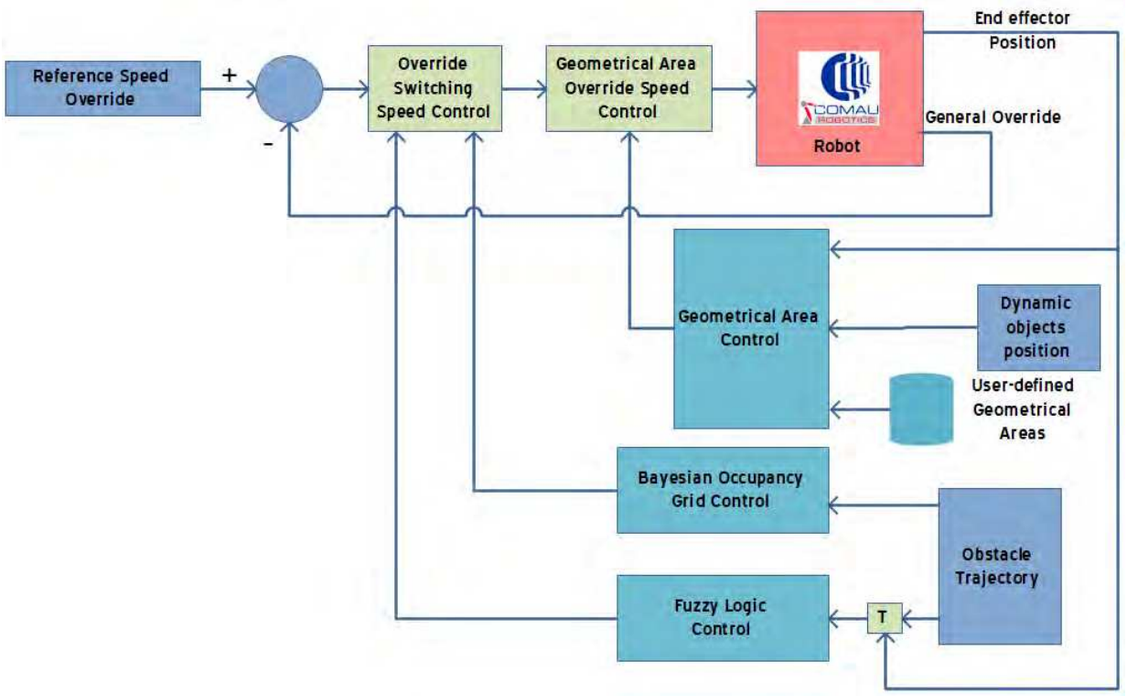


Fig. 8. Architecture of the hybrid control system

With this approach an override is computed for the static model of the system and then used by the geometrical area override speed control in order to select the best value to be taken into account for the regulation together with the values coming from the override switching speed control. The dynamic model of the objects are taken into account within the Bayesian occupancy grid and fuzzy logic control blocks which take as input the trajectory of the obstacles and the robot end-effector trajectory as well. The presented hybrid control gives the robot the possibility to react to changes both in the dynamic and static environment, preserving the characteristics of both controls; in the following sections the hybrid control will be tested in a simulation environment and in a real industrial robotized cell in order to prove the effectiveness of the studied control methods.

6. Simulation results

In this section a set of simulation results will be showed for both the static and dynamic control algorithms which have been analysed and developed in the previous sections. In particular, the simulation environment developed in Matlab and VRML will be showed, in order to prove the effectiveness of the presented paradigms and the hybrid system, and to validate the results before the application of the control methods on a real industrial robotic system. Given this, the simulation 3D environment has been developed with the aid of VRML in order to give a better representation of the virtual robot, while it is moving around its working area.
It is important to note that the control algorithms have been implemented into the real robot control, with the generation of real targets for the motors, but where the motors has not been attached to the controller unit.

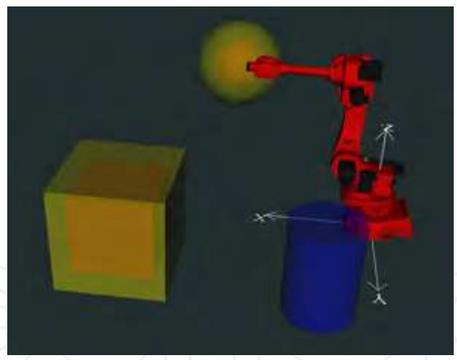


Fig. 9. 3D environment - The dynamic obstacle is depicted in blue as a cylinder; the controlled geometrical areas are depicted with their warning zones in yellow and red

Therefore the targets have been processed by the robot controller and then transferred via TCP/IP to the Matlab simulation environment through a client/server connection; in this typology of simulation, the motors target is updated real-time as in the real robot, and the VRML model of the manipulator acts in the same way as the robot should do in reality. Moreover a framework of the elementary geometrical areas and the dynamic obstacles moving around the robot area has been integrated in the simulation in order to better perceive the effectiveness of the hybrid control system while the robot is moving, changing its speed according to the control laws, when the manipulator end-effector encounters an obstacle in its trajectory or proximity or, in general, when it has to interact with its structured or unstructured environment. The obstacle avoidance control algorithm is capable of prevent impacts on the objects moving around the robot area; in this test, five obstacles moving around the robot surroundings were taken into account. The control system takes as input the override reference signal (which is considered constant), and it produces a controlled speed override value in order to control robot movements. The robot end-effector and obstacles are represented as points in space and the relative trajectories are defined a priori inside the simulator. The obstacles speeds are constant while the end-effector speed value is modified by the feedback control system output.

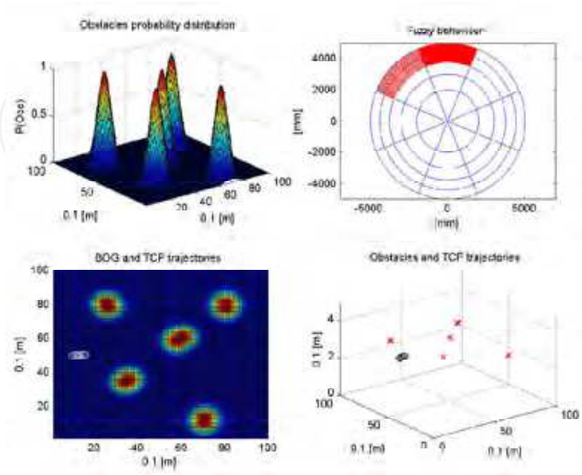


Fig. 10. General overview of simulation results for five obstacles moving around the robot surroundings

A general overview of the computed outputs is depicted in Figure 10 for five obstacles moving in the robot area; from this view it is possible to see the distribution of probabilities of the obstacles produced by the Bayesian occupancy grid filter in the upper left side. In the bottom left side the trajectories of the robot end-effector (represented as white circles) in 2D representation of the Bayesian occupancy grid are depicted. In upper right side, the Fuzzy logic controller behaviour is depicted for the closest obstacle. Figure 11 shows the behaviour of control system for five obstacles.

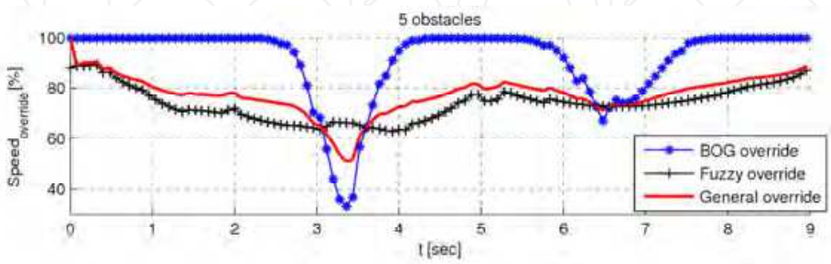


Fig. 11. General overview of simulation results for five obstacles moving around the robot surroundings

The trend shows that the Fuzzy logic controller reacts taking into account all the obstacles in the robot environment, and this can be seen considering the slow dynamics in the Figure. On the other hand Bayesian occupancy grid controller is more sensible to close obstacles and influences the general override in a significant way, when general override is low.

7. Experimental results

The second session of tests has been executed on the real Comau C4G robot controller and SMART NS16 manipulator (Figure 12); the SMART NS16 is a 6-axis industrial manipulator with a maximum load at wrist of 16 kilograms and a high repeatability of 0.05 mm.



Fig. 12. The SMART NS16 industrial robot manipulator

It is worth noting that for security reason, the algorithm to avoid collisions with humans has not been tested on the real robot as this experiment should include an appropriate hardware module to make the system absolutely safe, with redundancy controls on the real position of

the robot end-effector. In this section will be then showed the results of the control algorithm, while the robot is moving at its maximum speed and performing technological tasks in a real industrial cell. 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 13.a the robot end-effector position (X,Y) is plotted for two 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, where it is calibrated at home position, to a first point A. The following move is a Cartesian linear move, parallel to Y axis as far as the second point B. In the figure, two circles are depicted, highlighting the warning zone in yellow and the forbidden zone in red. When the forbidden zone is deactivated, on the blue trajectory, the end-effector moves towards the start position after reaching point B; with the red trajectory, the robot end-effector stops inside the warning zone at the boundaries of the forbidden zone when the control is enabled.

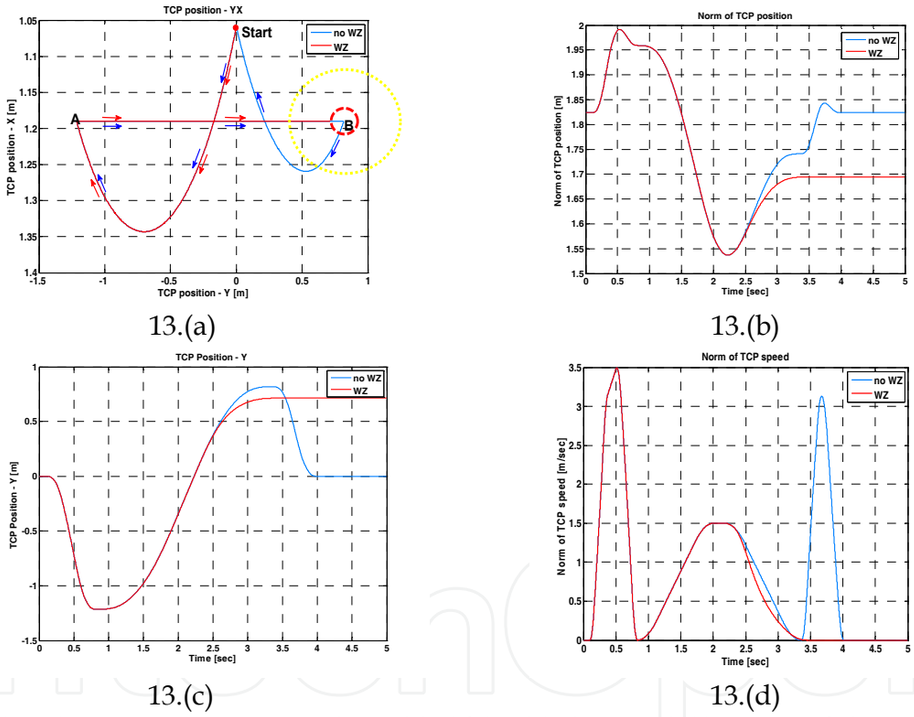


Fig. 13. a) Robot position (X,Y), with enabled/disabled control, b) Norm of the end-effector position over time, c) Trend of the Y end-effector position over time, d) Norm of the robot speed over time

The position trend over time is depicted in Figure 13.b and 13.c. From Figure 13.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 starts to move along the trajectory parallel to Y at 1.5 seconds and after almost one second it enters the warning zone. In Figure 13.c it is possible to see 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 13.d is then possible to see how the end-effector speed changes when it comes in contact with the warning zone at about 2.5 seconds. From a first analysis of these results it is clear how the reduction of speed is in inverse relation to the distance between the end-effector and the geometrical elementary zone surface. A similar analysis can be done concerning the acceleration trend shown in Figure 14.

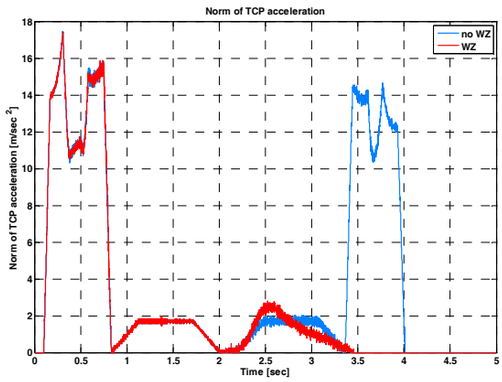


Fig. 14. Norm of the robot acceleration over time

These results can be easily extended to a more complex environment where several different geometrical areas are present and where the control algorithm has to manage the override control for each area, even if they overlap. Moreover the effectiveness of this method in presence of dynamic geometrical areas can be proved as well, when the system can be provided with sensors like external encoders which can be linked to a geometrical area, as stated in Section 3.1.2.

8. Conclusion

In this chapter new hybrid control techniques for the modelling of static and dynamic environments have been analyzed and developed, in order to make the designed controller capable to cope with both static and dynamic features; from the dynamic side of the problem, a framework which models both behavioural and probabilistic characteristics of the world surrounding the robot has been taken into account. A further control paradigm has been presented in order to interact with static environment, developing a solution with which it is possible to define several areas around the robot where its movements are allowed or likewise forbidden. It has also been showed as the presented hybrid control paradigm can be used as the basis for an overall increase of perception and interaction between the robot and its surroundings, distinguished by both environmental structures and human operators. The hybrid control system has been implemented both in simulation and on a real system, providing proofs of its feasibility, robustness and effective increase in robot-environment interaction. Moreover the modularity of the system allows extending its characteristics also taking into account other cognitive features.

9. Acknowledgments

This work 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.

10. References

Barak, D. & Witkin, A. (1992). Dynamic simulation of non-penetrating flexible bodies, *Computer Graphics*, vol. 26, no. 2, 1992, pp. 303-308

Barraquand, J.; Langlois, B. & Latombe, J.-C. (1989). Robot motion planning with many degrees of freedom and dynamic constraints, *Proceedings of the 5th international symposium on Robotics research,* pp. 435-444, 0-262-13253-2, 1991, MIT Press, Cambridge, MA, USA

Bouma, W. & Vanecek, J.G. (1992). Collision detection and analysis in a physically based simulation, *Proceedings of the 2nd Eurographics Workshop on Animation and Simulation*, 1992, Vienna, Austria

Brooks, R. A. (1986). A robust layered control system for a mobile robot, *IEEE Journal of Robotics and Automation*, vol. 2, no. 1, March 1986, pp. 14-23

Brooks, R. (2003). Planning collision-free motions for pick-and-place operations, *International Journal of Robotics Research*, vol. 2, no. 4, 2003, pp. 19-44

Canny, J. (1984). Collision detection for moving polyhedra, 1984, Cambridge, MA, USA

Collins, T.; Collins, J. & Ryan, C. (2007). Occupancy grid mapping: an empirical evaluation, *Proceedings of the 15th Mediterranean Conference on Control and Automation*, Athens, Greece, July, 2007

Dong, T.; Liao, X.; Zhang, R.; Sun, Z. & Song, Y. (2005). Path Tracking and Obstacles Avoidance of UAVs - Fuzzy Logic Approach, *Proceedings of the 14th IEEE International Conference on Fuzzy Systems. FUZZ apos; 05*, pp. 43-48, 0-7803-9159-4, May, 2005

Elfes, A. (1989). Using occupancy grids for mobile robot perception and navigation, *Computer*, vol. 22, June 1989, pp. 46-57

Fulgenzi, C.; Spalanzani, A. & Laugier, C. (2007). Dynamic obstacle avoidance in uncertain environment combining PVOs and occupancy grid, *Proceedings of the IEEE International Conference on Robotics and Automation*, Rome, Italy, 2007

Harville, M. & Li, D. (2004). Fast, Integrated Person Tracking and Activity Recognition with Plan-View Templates from a Single Stereo Camera, *Proceedings of IEEE Conference on Vision and Pattern Recognition*, pp. 398-405, 2004

Kim, B.; Kwon, O.; Kim, K.; Lee, E. & Hong, S. (1999). A study on path planning for mobile robot based on fuzzy logic controller, *Proceedings of IEEE TENCON'99*, pp. 1002-1005, December, 1999

Klir, G. & Yuan B. (1995). *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, Upper Saddle River, NJ, Prentice Hall

Kraiss, K.-F. (2006). *Advanced Man-Machine Interaction: Fundamentals and Implementation,* Springer

Malhotra, R. & Sarkar, A. (2005). Development of a fuzzy logic based mobile robot for dynamic obstacle avoidance and goal acquisition in an unstructured environment, *Proceedings of Advanced Intelligent Mechatronics 2005 IEEE/ASME*, pp. 1198-1203, 0-7803-9047-4, 2005

Mamdami, E.H. & Assilian, S. (1993). An experiment in linguistic synthesis with a fuzzy logic controller, *in Readings in Fuzzy Sets for Intelligent Systems*, pp. 283-289, 1993, B. Dubois, H. Prade, and R.R. Yager Eds., San Mateo, CA

Marcuse, H. (1964). One-Dimensional Man, *Studies in the ideology of advanced industrial society*, 1964.

Matinfar, M.; Baird, C.; Batouli, A.; Clatterbuck, R. & Kazanzides, P. (2007). Robot-assisted skull base surgery, *Intelligent Robots and Systems - IROS 2007 IEEE/RSJ International Conference*, 2007, pp. 865-870

Mauer, G.F. (1995). A fuzzy logic controller for an abs braking system, *IEEE FS*, vol. 3, November, 1995, pp. 381-388

Moravec, H. & Elfes, A. (1985). High resolution maps from wide angle sonar, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 116-121, 1985, St. Louis, Missouri: IEEE Computer Society Press

Moravec, H. (1988). Sensor fusion in certainty grids for mobile robots, *AI Magazine*, vol. 9, no. 2, July August 1988, pp. 61-74

Nehmzow, U. & Walker, K. (2005). Quantitative description of robot-environment interaction using chaos theory, *Robotics and Autonomous Systems*, vol. 53, Issues 3-4, 31 December 2005, pp. 177-193

Red, W.; Troung, C. & Kim, K. (1987). Robot path planning in three-dimensions using direct subspaces, *ASME Journal of Dynamics Systems, Measurement and Control*, vol. 109, 1987, pp. 238-244

Romanelli, F. & Tampalini, F. (2008a). A control algorithm for the management of multiple dynamical geometrical areas for industrial manipulators, *Proceedings of the RAAD 2008 17th International Workshop on Robotics in Alpe-Adria-Danube Region,* Ancona, Italy, September 15-17, 2008

Romanelli, F. & Tampalini, F. (2008b). BOG and fuzzy filters based multimodal collision avoidance for industrial manipulators, *Proceedings of the IEEE Conference on Robotics, Automation and Mechatronics*, pp. 779-784, 978-1-4244-1675-2, Chengdu, China, 21-24 September, 2008

Rooks, B. (2005). Precision robot welding for world's largest particle physics apparatus, *Industrial Robot: An International Journal*, vol. 32, no. 4, 2005

Roy, S. & Pratihar, D. (2003). A genetic-fuzzy approach for optimal path-planning of a robotic manipulator among static obstacles, *IE(I) Journal.CP*, vol. 84, 2003, pp. 86-89

Ruspini, E. H. (1991). On the semantics of fuzzy logic, *International Journal of Approximate Reasoning*, vol. 5, no. 1, January 1991, pp. 45-88

Schulz, B. (2007). The gift of sight, *Austrlian Manufacturing Technology*, 2007, pp. 86-89

Shahrokhi, M. & Bernard, A. (2004). A fuzzy approach for definition of dangerous zone in industrial systems, *SMC(7)*, 2004, pp. 6318-6324

Stepan, P.; Kulich, M. & Preucil, L. (2005). Robust data fusion with occupancy grid, *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, vol. 35, Feb 2005, pp. 160-115

Van Wichert, G. & Lawitzky, G. (2001). Man-machine interaction for robot applications in everyday environments, *Proceedings of the 10th IEEE International Workshop on Robot and Human Interactive Communication*, pp. 343-346, 2001

Vasquez, D.; Romanelli, F.; Fraichard, T. & Laugier, C. (2006). Fast object extraction from Bayesian occupancy grids using self organizing networks, *Proceedings of the 9th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pp. 1-6, 2006

Winkler, B. (2007). Safe space sharing human-robot cooperation using a 3D time-of-flight camera, *Techical Conference - Robots and Vision Show,* 2007

Yen, J. & Pfluger, N. (1995). A fuzzy logic based extension to Payton and Rosenblatt's command fusion method for mobile robot navigation, *IEEE Transactions on Systems, Man and Cybernetics*, vol. 25, no. 6, 1995, pp. 971-978

Zadeh, L. (1965). Fuzzy sets, *Information and Control*, vol.8, June 1965, pp. 338-353



Robot Manipulators New Achievements

Edited by Aleksandar Lazinica and Hiroyuki Kawai

ISBN 978-953-307-090-2
Hard cover, 718 pages
Publisher InTech
Published online 01, April, 2010
Published in print edition April, 2010

Robot manipulators are developing more in the direction of industrial robots than of human workers. Recently, the applications of robot manipulators are spreading their focus, for example Da Vinci as a medical robot, ASIMO as a humanoid robot and so on. There are many research topics within the field of robot manipulators, e.g. motion planning, cooperation with a human, and fusion with external sensors like vision, haptic and force, etc. Moreover, these include both technical problems in the industry and theoretical problems in the academic fields. This book is a collection of papers presenting the latest research issues from around the world.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Fabrizio Romanelli (2010). Hybrid Control Techniques for Static and Dynamic Environments: a Step towards Robot-Environment Interaction, Robot Manipulators New Achievements, Aleksandar Lazinica and Hiroyuki Kawai (Ed.), ISBN: 978-953-307-090-2, InTech, Available from: http://www.intechopen.com/books/robot-manipulators-new-achievements/hybrid-control-techniques-for-static-and-dynamic-environments-a-step-towards-robot-environment-inter

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License, which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen