Robotica (2006) volume 24, pp. 711–726. © 2006 Cambridge University Press
doi:10.1017/S0263574706002888 Printed in the United Kingdom

Comparative analysis of collision-free path-planning methods for multi-manipulator systems

J. C. Fraile*, J. Perez-Turiel, J. L. Gonzalez-Sanchez, E. Baeyens and R. Perez

Departamento de Ingeniería de Sistemas y Automática, Escuela Técnica Superior de Ingenieros Industriales, Universidad de Valladolid, Valladolid, Spain

(Received in Final Form: May 9, 2006. First published online: July 11, 2006)

SUMMARY

Motion planning for manipulators with many degrees of freedom is a complex task. The research in this area has been mostly restricted to static environments. This paper presents a comparative analysis of three reactive on-line path-planning methods for manipulators: the elastic-strip, strategy-based and potential field methods. Both the elastic-strip method [O. Brock and O. Khatib, "Elastic strips: A framework for integrated planning and execution," *Int. Symp. Exp. Robot.* 245–254 (1999)] and the potential field method [O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Robot. Res.* **5**(1), 90–98 (1986)] have been adapted by the authors to the problem at hand related to our multi-manipulator system (MMS) (three manipulators with five degrees of freedom each). Strategy-based method is an original contribution by the authors [M. Mediavilla, J. L. González, J. C. Fraile and J. R. Perán, "Reactive approach to on-line path planning for robot manipulators in dynamic environments," *Robotica* **20**, 375–384 (2002); M. Mediavilla, J. C. Fraile, T. González and I. J. Galindo, "Selection of strategies for collision-free motion in multi-manipulator systems," *J. Intell. Robot Syst* **38**, 85–104 (2003)].

The three methods facilitate on-line path planning for our MMS in dynamic environments with collision avoidance, where the three manipulators may move at the same time in their common workspace. We have defined some 'basic motion problems' for the MMS, and a series of simulations has been running that will tell us how effective each path-planning method is. The simulations have been performed and the obtained results have been analysed by using a software program developed by the authors.

The paper also presents experimental results obtained applying the path-planning methods to our MMS, that perform pick-and-place tasks sharing common working areas.

KEYWORDS: Path planning; Collision-free; Reactive; Dynamic environments; Manipulators.

* Corresponding author. E-mail: jcfraile@eis.uva.es

1. INTRODUCTION

Path planning in mobile robots (autonomous vehicles) has experienced a great advance in recent years due to the development of new techniques combining 'task execution' and 'collision avoidance' in dynamic environments.

On the other hand, planning collision-free paths for manipulators has not seen such an advance and the techniques developed are applied, in particular, to the static and structured environments, but not to the dynamic ones. Since most path planning is designed for static environments, manipulators (in industry as well as in many research projects) are restricted to precalculated paths and are subject to rigid timing constraints. These features make them too rigid and unable to adapt to changes. Increasing the reactivity of the manipulator motion would have important advantages:

– React to unexpected events such as moving obstacles or faults. Normally, industrial manipulators follow precalculated paths, and thus, any unexpected situation (fault, mobile obstacle) must be solved by human intervention. A reactive on-line path-planning algorithm would avoid human intervention to react in such situations, unless the manipulators require following paths optimized in time and/or space.
– One of the fields of service robotics that is undergoing most of the development is that of robots helping the disabled. These robots must move in unknown environments where path planning needs to have a high reactivity.

Path planning for manipulators has been studied by numerous authors: McLean and Mazon[23] perform path planning for an industrial robot in static environments, using the probabilistic roadmaps (PRM) proposed by Kavraki *et al.*[17]. Leven and Hutchinson[20] improve the efficiency of PRM by pre-building a roadmap for a manipulator in an obstacle-free Cartesian workspace. The presence of obstacles is considered by modifying only the affected portion of the roadmap. The dynamic roadmap (DRM) obtained is thus able to cope with dynamic changes in the environment, and replanning is possible in a dynamic environment. Li and Latombe[21] describe an application of on-line path planning of two Scara robots. They get on-line path planning by reducing the problem to a set of subproblems with three degrees of freedom each.

https://doi.org/10.1017/S0263574706002888 Published online by Cambridge University Press

The potential field method proposed by Khatib[18] is applied to the path planning of manipulators,[8] where the problem of local minima appears. The potential field method has more limitations: no passage between closely spaced obstacles, oscillations in narrow passages, goals nonreachable with obstacles nearby. To overcome these problems, Ge and Cui[10] create a new repulsive potential function that takes the relative distance between the robot and the goal into consideration, which ensures that the total potential has a global minimum at the goal position. Kim *et al.*[19] present a new repulsive potential field with angle distributions. This proposal uses the angle between robot-to-goal and robot-to-obstacle, as well as relative distances (robot–goal, robot–obstacle) to solve a number of local minimum and collision problems. Rimon and Koditschek[31] present a new class of artificial potential functions, called 'navigation functions', that have only one minimum, the goal configuration.

The randomized path-planner method proposed by Barraquand *et al.*[2] allows the potential techniques to be applied to robots with a high number of degrees of freedom (six), as in the case of a manipulator. Conolly[7] performs path planning using harmonic functions as potential functions, which are free from local minima. These have been applied by Souccar *et al.*[32] to manipulators with two or three degrees of freedom. Gupta[12] presents a method that takes advantage of the possibilities of the series structure of the manipulators. He breaks down the problem of collision-free path planning for a manipulator with 'n' degrees of freedom into a series of 'm' subproblems, corresponding to the path planning for a subgroup of '$m-1$' manipulator arms. Lindemann and LaValle[22] present an overview of sampling-based motion planning and conclude that the success of motion-planning algorithms are mainly due to their sampling-based nature, not due to the randomization used to generate the samples.

Path-planning approaches for manipulators can broadly be categorized into two classes: global and local methods. Gupta and Pobil[13] compile in their book, a collection of papers about path planning. Global methods[3,11,17] are computationally very expensive, and their computational cost increases rapidly as a function of the number of manipulator joints. Furthermore, they are not applicable when the obstacles are unmodeled or subject to uncertainties. Local methods,[18] on the other hand, are reactive and can be used for on-line path planning, but most of them are very limited in their capabilities and easily get trapped in local minima.

The on-line path planning requires a constant flow of information between the planner and the environment. Each point on the path is generated when the manipulator executes its movement, as an immediate response to the appearance of new events detected by the incorporated sensors. This kind of planning is essential for manipulators that move in dynamic or partially unknown environments.

Applications where the robots plan their movement on-line, are usual in the field of mobile robotics,[1] while very little work has been done on manipulators. McLean and Mazon[23] consider an on-line planning strategy for two manipulators that move in a coordinated way. They use a non-reactive method based on global planning. Hamilton and Dodds[16] propose a reactive method based on a set of behaviours. This approach has a problem similar to the local minima of potential fields. Quinlan *et al.*[29,30] look at on-line path planning by using the elastic band concept. This approach tries to bring together the local and global aspects of path planning in order to apply it in real time for robots in a changing environment. Brock and Khatib[5] propose a reactive planning method: 'the elastic-strip framework', to integrate behaviour during reactive motion execution. It allows the combination of task execution and obstacle avoidance.

For the last few years, our research group has been working on a prototype of a multi-manipulator system (MMS)[9,24–28] located in our laboratories at the University of Valladolid. The three manipulators of our MMS share a wide portion of their workspace, but the path-planning methods that we are developing enables them to operate despite the fact that they might be moving in the same area. This means that when a supply comes in, one of the manipulators starts its motion towards it, irrespective of whether the rest of the manipulators are at that time already in motion or not. This kind of behaviour needs the manipulators to move using an on-line and reactive planning method.

The paper is organized as follows. Section 2 shows an adaptation of the framework of elastic strips proposed by Brock and Khatib[5] to apply to our MMS (a 15-degree-of-freedom manipulator system). Also, we work with the concept of bubble proposed by Quinlan,[29] considering the bubbles in the manipulator's workspace. Sections 3 and 4 present the main ideas of the other two alternative path-planning methods[27,28] developed by the authors for our MMS: strategy-based path-planning and potential method. Section 5 defines the concept of 'motion problem' and the kind of experiments using MMS that will be used for a comparative analysis of the three path-planning methods presented. Section 6 analyses and compares the simulation results obtained with the three path-planning methods using a software package developed by the authors. Section 7 shows the results of the implemented path-planning methods that have been tested on the MMS. Finally, Section 8 provides our conclusions.

2. PATH PLANNING IN MMS: AN ADAPTATION OF ELASTIC-STRIPS METHOD

2.1 Preliminary concepts

Our MMS is composed of three Scorbot-ER IX units, each with five degrees of freedom (five rotational links), and several working platforms (see Fig. 1). One of the platforms is located in the centre and is shared as the assembly platform. The other three tables are located at both sides of each manipulator and are used for parts and product release and storage. Each manipulator can access two side-working tables and the central area. When carrying out a task, each of the manipulators in motion constitutes a dynamic obstacle for the others. In addition, the floor constitutes a static obstacle. The MMS has no sensors. Therefore, in order to know the position of the manipulator–obstacles at each moment, a kinematic model of the Scorbot-ER IX has been used.

The computational cost of path planning for each manipulator is determined by the need to calculate the available free workspace so that the manipulator can move

https://doi.org/10.1017/S0263574706002888 Published online by Cambridge University Press



Fig. 1. MMS in our laboratory.

without collisions. The high number of degrees of freedom (15) of the MMS generates a very high computational cost for the calculation of an explicit representation of the entire free space. In addition, the presence of dynamic obstacles would indicate the use of global methods that explore all the configuration space, because of its highly changeable nature.

The elastic-strip framework proposed by Brock and Khatib[5] defines a protective hull consisting of a set of overlapping bubbles that guarantee an obstacle-free configuration space in the manipulator workspace around. In the elastic band theory proposed by Quinlan and Khatib,[30] the obstacles produce repulsive forces that deform the elastic material, and modify the path. When the obstacle disappears, the path recuperates its initial form.

We use the concepts and theory of Brock and Quinlan, and adapt and apply them to our MMS, to obtain a path-planning method and collision-free working in the manipulators workspace.

The tool centre point (TCP) for each manipulator of our MMS is defined). These manipulators perform pick-and-place tasks; then the TCP is located at the end of the tool. When the TCP is located in a point of the workspace, we say that the manipulator reaches this point.

A path for a manipulator is defined as a set of points of its workspace that the TCP should pass in some certain instants of time. For a manipulator's configuration q_i given by $q_i = (q_{1i}, q_{2i}, q_{3i}, q_{4i}, q_{5i})$, the position of the TCP in the manipulator workspace is denoted by p_i^{TCP}. Therefore, a manipulator path will be given by a group of workspace points p_i^{TCP} $(i = 0, 1, \ldots, n)$ that should be reached by its TCP in certain instants of time.

The path-planning method that we apply to our MMS follows the work of Brock *et al.*[4,5] and Quinlan *et al.*,[29,30] and uses the concept of dynamic tunnel to calculate 'on-line' collision-free path in the workspace. A local exploration of the workspace of each manipulator is carried out 'on-line', until a volume can be found that guarantees the existence of a collision-free path between the initial and final points of the path. This free space within the workspace of the manipulator is called the 'dynamic tunnel'.

Using the parameter of 'distance', the bubble around p_i^{TCP} can be computed in the workspace. The dynamic tunnel is made up of a set of bubbles. The bubble B_i^{TCP} around p_i^{TCP} contains all points 'p' in the free workspace that are closer to p_i^{TCP} than the closest obstacle (Brock[4]), and is defined as

$$B_i^{\mathrm{TCP}} = \left\{ p : \left\| p_i^{\mathrm{TCP}} - p \right\| < \frac{D}{2} \right\}$$

where

– D is the diameter of the bubble, defined by the user. $D/2$ is the minimum distance between the obstacles and the manipulator when its TCP is on p_i^{TCP}. The TCP can move up to a distance $D/2$ in any direction, guaranteeing no collision with any obstacle.
– $\|\cdot\|$ is the distance operator between two points in the workspace.

Let $p_0^{\mathrm{TCP}}, p_1^{\mathrm{TCP}}, \ldots, p_n^{\mathrm{TCP}}$ be a sequence of $n+1$ points belonging to the path of the TCP in the workspace. Each point of the sequence has associated a bubble B_i^{TCP} $(i = 0, 1, \ldots, n)$. Every set of bubbles is called a 'dynamic tunnel' $T_{\mathrm{d}}^{\mathrm{TCP}}$

$$T_{\mathrm{d}}^{\mathrm{TCP}} = \sum_{i=0}^{n} B_i^{\mathrm{TCP}}$$

and it satisfies the condition that all the points that are inside the tunnel belong to the free workspace. The path-planning method presented here will be denominated 'dynamic tunnel method'. This method has been derived as an application to our MMS, the framework of elastic strips developed by Brock.[4]

Keeping in mind the geometric disposition of the manipulators in our MMS (located in the vertexes of an equilateral triangle of 1-m side), and the nature of the tasks that will be carried out (pick-and-place tasks), an analysis of the more common collisions among the manipulators has been performed. Figure 2 shows that the most common collisions are between the TCP of a manipulator and the articulations 3, 4 or 5 of the other manipulators. For this, it has been considered enough to take the TCP (point p_i^{TCP}) as the only point of the manipulator where the bubble is defined. Selecting an appropriate diameter for the bubble can guarantee the non-existence of the type of collisions shown in Fig. 2.

2.2. Path planning in MMS using dynamic tunnel method

The MMS path-planning problem, we wish to solve, is stated as follows:

For each manipulator, given the initial and final points in the workspace of the task to be performed—p_0 and p_n—corresponding to the initial $q_0 = (q_{10}, q_{20}, q_{30}, q_{40}, q_{50})$ and final $q_n = (q_{1n}, q_{2n}, q_{3n}, q_{4n}, q_{5n})$ configurations, respectively, we have to compute on-line the dynamic tunnel that guarantees the existence of a collision-free path between the initial and final points, within a highly dynamic environment, in which one, two or all three manipulators can move simultaneously.

2.2.1. The initial dynamic tunnel $T_{\mathrm{Do}}^{\mathrm{TCP}}$. To calculate $T_{\mathrm{Do}}^{\mathrm{TCP}}$, a candidate path between the initial point p_0^{TCP} and

https://doi.org/10.1017/S0263574706002888 Published online by Cambridge University Press

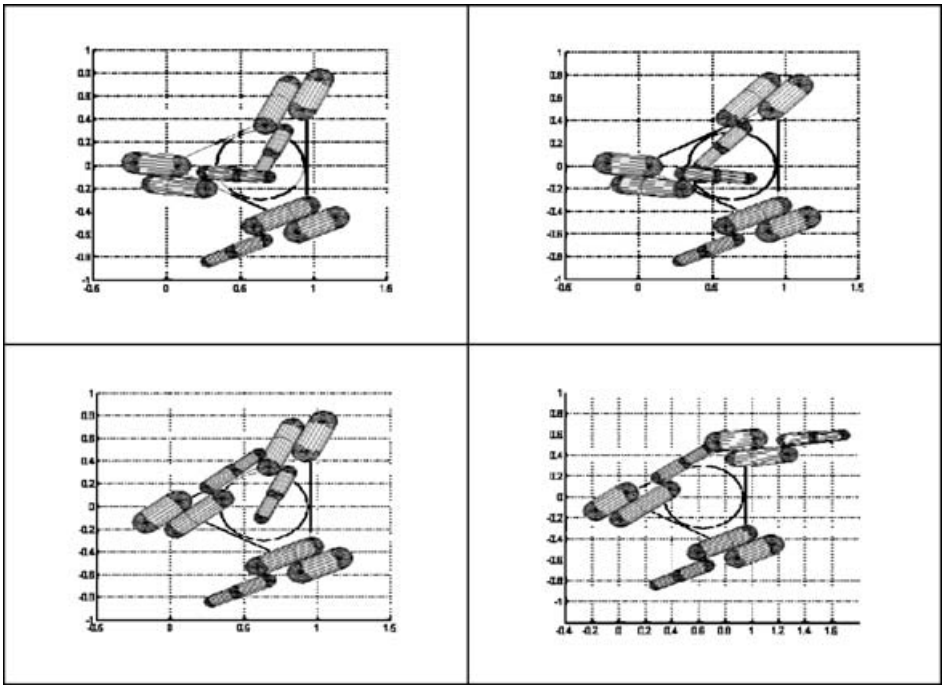


Fig. 2. Types of collisions considered among the manipulators in our MMS.

the final point p_n^{TCP} is considered. This candidate path is composed by a set of '$n+1$' points of the workspace p_0^{TCP}, $p_1^{\mathrm{TCP}}, \ldots, p_n^{\mathrm{TCP}}$, distributed in a uniform way along the straight line that joins p_0^{TCP} and p_n^{TCP}.

$T_{\mathrm{Do}}^{\mathrm{TCP}}$ is composed by a set of '$n+1$' bubbles B_i^{TCP} ($i=0, 1, \ldots, n$), centred in p_i^{TCP} ($i=0, \ldots, n$) inside the manipulator workspace. In Fig. 3, $T_{\mathrm{Do}}^{\mathrm{TCP}}$ is represented by a linear tube of constant diameter D. $T_{\mathrm{Do}}^{\mathrm{TCP}}$ is defined as

$$T_{\mathrm{Do}}^{\mathrm{TCP}} = \left\{ B_i^{\mathrm{TCP}},\ p : \left\| p_i^{\mathrm{TCP}} - p \right\| < \frac{D}{2}, \qquad i = 0, \ldots, n \right\}.$$

The value of the parameter 'n' (number of bubbles) has been experimentally obtained. A high value means a high computational cost in the calculation of the successive dynamic tunnels. A small value means that the bubbles are very separate, and some obstacles could exist among them that would not be detected and cause a collision. It has been considered from the study performed in Section 6.3 that 'n' varies between 20 and 60.

If there are no obstacles close to, or intersecting with $T_{\mathrm{Do}}^{\mathrm{TCP}}$, all their points belong to the free workspace, and the points p_i^{TCP} ($i=0, \ldots, n$) constitute the manipulator path from p_0^{TCP} to p_n^{TCP}.

2.2.2. Deformation of the dynamic tunnel: calculation of $T_{\mathrm{Di}}^{\mathrm{TCP}}$. The highly dynamic environment in which the three manipulators move produces mobile obstacles. These obstacles generate a group of forces on the bubbles that cause the deformation of $T_{\mathrm{Do}}^{\mathrm{TCP}}$, and modify its geometry, obtaining a new dynamic tunnel $T_{\mathrm{Di}}^{\mathrm{TCP}}$. The calculation of $T_{\mathrm{Di}}^{\mathrm{TCP}}$ should be updated, at least, every time the manipulator moves between two serial points, p_i^{TCP} and p_{i+1}^{TCP}, to guarantee that all the points of $T_{\mathrm{Di}}^{\mathrm{TCP}}$ are in the free workspace.

Figure 3 shows that all the points of $T_{\mathrm{Do}}^{\mathrm{TCP}}$ do not belong to the free workspace, due to the appearance of mobile obstacles close to $T_{\mathrm{Do}}^{\mathrm{TCP}}$. To assure a free collision path, the forces of the obstacles on the bubbles should displace them to a position that satisfies the condition: the distance from the centre of the bubble to a point of any obstacle of the environment has to be at least $D/2$.

The deformation of the dynamic tunnel guaranteeing that all their points belong to the free workspace is carried out by the action of a group of forces acting on each bubble. Quinlan[29] considers that the bubbles move in the configuration space, to cause the deformation of the elastic band. Three types of forces are involved in this movement: internal, external and constraint forces. In order to deform the dynamic tunnel in the workspace, these three forces have been considered acting in the manipulator workspace, instead of the configuration space. The considered forces[29] are

- internal force 'F_{i}', to remove stack in the dynamic tunnel;
- external force 'F_{e}', To move the dynamic tunnel away from the obstacles in the workspace;
- constraint force 'F_{c}' to inhibit the motion of bubbles along the dynamic tunnel.

Once deformed, $T_{\mathrm{Do}}^{\mathrm{TCP}}$ obtained the new positions of all the bubbles B_i^{TCP} ($i=0, 1, \ldots, n$), and a new deformed dynamic tunnel $T_{\mathrm{D1}}^{\mathrm{TCP}}$ is obtained. All points of this new tunnel are in the free workspace of the manipulator (Fig. 4). The centres of the bubbles are the positions of the manipulator's TCP in the workspace that define the new path free of collisions between p_0^{TCP} and p_n^{TCP}. On this new calculated path, the manipulator will carry out its first movement from p_0^{TCP} to p_1^{TCP}.

In general, after the manipulator moves from p_{i-1}^{TCP} to p_i^{TCP}, according to the path determined by the dynamic tunnel $T_{\mathrm{D}(i-1)}^{\mathrm{TCP}}$, the path-planning method must calculate a

https://doi.org/10.1017/S0263574706002888 Published online by Cambridge University Press

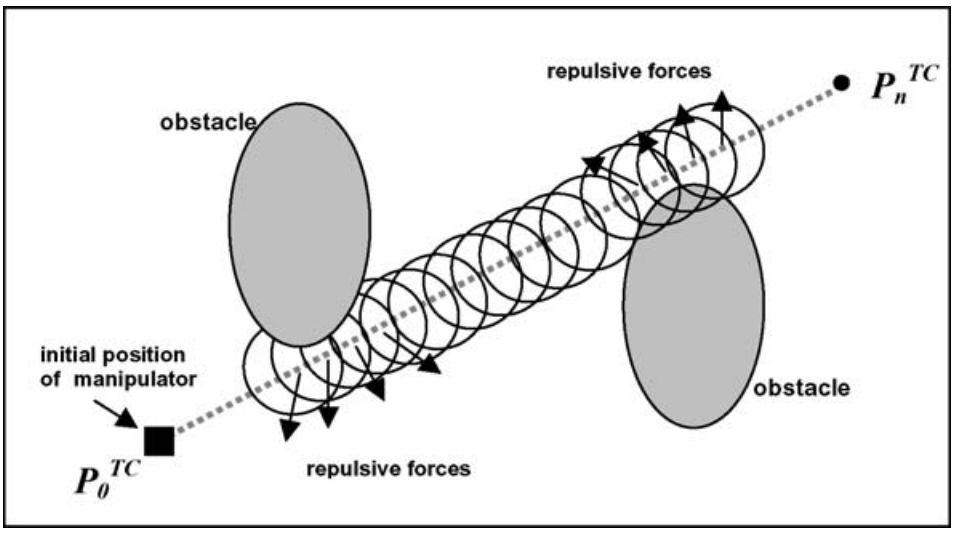


Fig. 3. Initial dynamic tunnel $T_{\mathrm{Do}}^{\mathrm{TCP}}$ repulsive forces over dynamic tunnel due to obstacles.

new deformed tunnel $T_{\mathrm{D}i}^{\mathrm{TCP}}$, since the mobile obstacles may change its position and intersect the tunnel $T_{\mathrm{D}(i-1)}^{\mathrm{TCP}}$. If the obstacles do not change, the tunnel would not be deformed and consequently $T_{\mathrm{D}i}^{\mathrm{TCP}} \equiv T_{\mathrm{D}(i-1)}^{\mathrm{TCP}}$.

$T_{\mathrm{D}i}^{\mathrm{TCP}}$ is composed by a set of '$n-i+1$' bubbles B_j^{TCP} centred on the p_j^{TCP} ($j= i, \ldots, n$) of the manipulator

$$T_{\mathrm{D}i}^{\mathrm{TCP}} = \left\{ B_j^{\mathrm{TCP}},\, p : \left\| p_j^{\mathrm{TCP}} - p \right\| < \frac{D}{2},\, j = i, \ldots, n \right\}.$$

Once $T_{\mathrm{D}i}^{\mathrm{TCP}}$ is computed, the manipulator will move inside this tunnel from position p_i^{TCP} to p_{i+1}^{TCP}. Again, the new deformed tunnel $T_{\mathrm{D}(i+1)}^{\mathrm{TCP}}$ will be computed from p_i^{TCP}. This process will be repeated several times until the manipulator reaches the final position p_n^{TCP}.

A series of deformed dynamic tunnels $T_{\mathrm{D}i}^{\mathrm{TCP}}$ is computed (see Fig. 4) to guarantee the existence of a collision-free manipulator path from the initial to the final points. Three snapshots (Fig. 4) show the process of deformation of the dynamic tunnel.

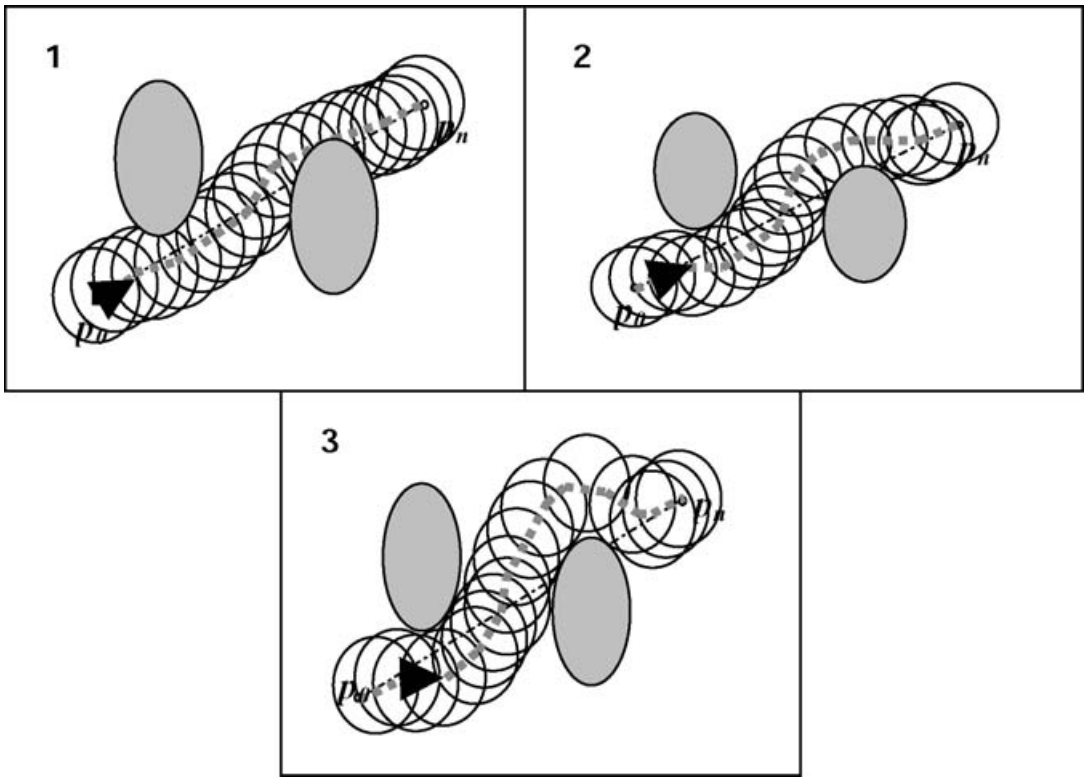


Fig. 4. Dynamic tunnel deformation $T_{\mathrm{D}i}^{\mathrm{TCP}}$. Fine dotted line: initial path. Gross dotted line: new path calculated. Arrow point: current manipulator position.

https://doi.org/10.1017/S0263574706002888 Published online by Cambridge University Press

The path-planning dynamic-tunnel method can be used both in slowly and highly dynamic environments. A parameter 'f_d', denominated the 'frequency of deformation' has been introduced. This parameter denotes the maximum number of positions that the manipulator can move to without having to recalculate a new deformed tunnel. This parameter takes values between 1 and 3. In highly dynamic environments $f_d = 1$, which means that the computation of the new deformed tunnel should be performed every time the manipulator moves a position (from p_i^{TCP} to p_{i+1}^{TCP}). If $f_d = 3$, the deformed tunnel has to be updated after three serial manipulator movements (from p_i^{TCP} to p_{i+3}^{TCP}).

2.3. MMS path generation and execution using dynamic tunnels

The dynamic tunnel method provides a tunnel T_{Di}^{TCP} and two points p_i^{TCP} and p_{i+1}^{TCP} at each step. Now, a valid path inside the tunnel T_{Di}^{TCP} from position p_i^{TCP} to p_{i+1}^{TCP} has to be computed. The path has been obtained by using a linear interpolation strategy with parabolic adjustment. The main features of this strategy are

- The manipulator performs a smooth movement, without jerking or stopping at passing points.
- The final path is a straight line between the two points, except in the proximity of the extreme points, where a second-degree polynomial allows the speed to be progressively altered.
- The computed path is almost a straight line (from one point to the other); this guarantees that it will be within the dynamic tunnel. Interpolation using larger order polynomials produces oscillatory paths that could abandon the tunnel.

2.4. Path-planning algorithm using dynamic tunnels

1. For the three manipulators of the MMS:
2. For each manipulator, input data:

   p_0^{TCP}: initial point of the path
   p_n^{TCP}: final point of the path
   D: diameter of the bubbles
   n: number of bubbles in the dynamic tunnel
   f_d: deformation frequency of the dynamic tunnel
   $\rho_{i,j}$: distance between the centres of B_i^{TCP} and B_j^{TCP}
   $\rho_{obs}(p_i)$: distance from the bubble B_i^{TCP} to the nearest obstacle.

3. Calculation of the initial dynamic tunnel T_{D0}^{TCP}:
   Calculation of p_j^{TCP} $(j = 0, \ldots, n)$
   (centres of the '$n+1$' bubbles that join p_0^{TCP} and p_n^{TCP})
4. Make $i = 0$
5. While $p_i^{TCP} \neq p_n^{TCP}$ make:
   5.1. Calculation of the dynamic tunnel p_{Di}^{TCP} (if necessary):
   Calculation of $\rho_{obs}(p_j)$ $(j = i, \ldots, n)$
   Calculation of the displacement of the bubbles B_j^{TCP} $(j = i, \ldots, n)$:
   FOR $j = i, n$
   IF $\rho_{obs}(p_j) < D/2$ then
   Calculation of the forces F_i, F_e and F_c on B_j^{TCP}
   Calculation of the new position p_{jnew} of the bubble B_j^{TCP}
   IF $[(\rho_{jnew,\,j+1} < 0.5 \cdot \rho_{obs}(p_{jnew}))$ AND
   $(\rho_{jnew,\,j-1} < 0.5 \cdot \rho_{obs}(p_{jnew}))$ AND
   $(\rho_{obs}(p_{jnew}) \geq D/2)]$ THEN
   $p_j^{TCP} = p_{jnew}^{TCP}$
   ELSE
   ERROR: Tunnel cannot be deformed. END.
   ENDIF
   ENDIF
   ENDFOR
   5.2. Move the manipulator on the path defined by T_{Di}^{TCP}
   While $p_i^{TCP} \neq p_n^{TCP}$
   move from p_i^{TCP} to $p_{1+f_d}^{TCP}$
   Make $i = i + f_d$
   5.3. Return to point 5

: END of the algorithm OK

3. STRATEGY-BASED PATH PLANNING: GENERAL DESCRIPTION

In references,[26–28] the strategy-based path-planning approach was presented. It is based on what we call *motion strategies*. Motion strategies are simple and effective ways of moving a manipulator that are designed for specific kinds of tasks. A manipulator motion strategy is implemented by restricting its motion to a subset of its configuration space, that we call *reduced subspace* (R-subspace), or C_R. We need to ensure that this restriction is correctly chosen, which is why we use an off-line stage that optimizes the choice of the reduced subspaces, and minimizes the probability of not finding a collision-free path.

Path planning based on strategies works in two stages:

- The off-line stage: This optimization stage obtains the R-subspaces (strategies) that are most suitable for the manipulators.
- The on-line stage: A fast and reactive algorithm based on local information is used to search the manipulator path inside the R-subspace obtained in the previous stage.

Mediavilla *et al.*[28] presented an in-depth strategy-based path-planning method. The method is focused on two particular motion problems, denominated problem 1 and problem 2. Motion problems are described as general types of tasks that can be solved using one motion strategy. Also, an optimization stage based on estimation of the probability of faults (EPF) was described. The result of this optimization is a set of strategies that are suitable for each motion problem. Once the strategies have been chosen, they are used for on-line motion using a local and reactive path-planning algorithm.

In reference,[26] R-subspaces are described in detail. However, a brief description of them will be given here. R-subspaces are linear and two-dimensional subspaces of the configuration space $\boldsymbol{C}$, although our method could be adapted

https://doi.org/10.1017/S0263574706002888 Published online by Cambridge University Press

to more complex subspaces. These subspaces must contain the initial q_0 and final q_n configurations of the manipulator, so that it can move from q_0 to q_n inside C_R. R-subspaces are defined by two unitary vectors $\mathbf{u_1}, \mathbf{u_2} \in \boldsymbol{C}$ such that

$$C_R = \{q \in C, \quad q = q_i + x_1 \cdot \vec{u}_1 + x_2 \cdot \vec{u}_2\}$$

where x_1 and x_2 are the independent variables of the subspace. Vector $\mathbf{u_1}$ points in the direction of the line connecting the initial and final configuration, and only depends on the task assigned to the manipulator. Vector $\mathbf{u_2}$ characterizes the particular R-subspace, and therefore, describes the motion strategy of the manipulator associated to that particular R-subspace.

4. POTENTIAL PATH PLANNING: GENERAL DESCRIPTION

A path-planning method based on potentials[18] is equivalent to an optimization algorithm searching for the minimum of a potential function. Its main drawback is the existence of local minima. Rimon and Koditschek[31] introduced an artificial potential function called a 'navigation function' that offers a formal solution to the problem of local minima in potential field motion planning. Navigation functions are smooth real-valued maps realized through suitably chosen scalar-valued cost functions. They give trajectories that guarantee collision-free motion and convergence to the destination from almost all initial conditions.

The potentials-based path-planning method has been developed and applied to our MMS. For a manipulator, we define potential $U(q_i)$ at a point $q_i \in C$ of its configuration space as the sum of the attractive $U_a(q_i)$ and repulsive $U_r(q_i)$ potentials at that point.

$$U(q_1) = U_a(q_i) + U_r(q_i).$$

The attractive potential only depends on the final configuration (final position) the manipulator must reach. The repulsive potential is associated with the obstacle, showing its influence in a region around this obstacle.

The magnitude of the force applied to the manipulator is obtained by calculating the gradient of the potential $U(q_i)$, and the direction of the force will be that of the potential's maximum variation with the opposite sign, thus moving the manipulator towards less potential areas, that is

$$\vec{F}(q_i) = -\vec{\nabla} U(q_i)$$

For a manipulator on a configuration q_i, the attractive potential $U_a(q_i)$ caused by the final configuration is given by

$$U_a(q_i) = \frac{1}{2} \times \xi \times \rho^2_{obj}(q_i)$$

where

- $\xi \in [10, 1000]$ is an attractive gain.
- ρ_{obj} is the distance between the manipulator and the final configuration.

The repulsive potential $U_r(q_i)$ is defined with the following considerations:

- It must be inversely proportional to the distance between the manipulator and the obstacle.
- The repulsive potential should decrease in line with the distance between the manipulator and the objective.

$$U_r(q_i) = \begin{cases} \frac{1}{2} \times \eta \times \left(\frac{1}{\rho_{obs}(q_i)} - \frac{1}{\rho_0} \right)^2 \times \rho^2_{obj} & si\ \rho_{obs}(q_i) \le \rho_0 \\ 0 & si\ \rho_{obs}(q_i) > \rho_0 \end{cases}$$

where

- $\eta \in [10, 1000]$ is the repulsive gain.
- ρ_{obs} is the distance between manipulator and obstacle.
- ρ_o is the distance of influence of the obstacle.

Different values for the repulsive and attractive gain have been experimentally tested in the MMS. The best results have been found for rates of attractive/repulsive gain around 10/1. If this ratio is close to 1/1, the results are not very efficient, as the length of the path becomes unnecessarily long, and thus, the time the manipulators take to reach the final point of the path also increases.

The geometric disposition of the three manipulators in the MMS produces a reduced workspace with obstacles (the manipulators themselves) in motion. Thus, the application of the potentials method to path planning in our MMS generates a large number of blockages in the movements of the manipulators due to the local minima of the function that defines the field of potentials. Several strategies have been implanted in the MMS to avoid these local minima or to abandon them once they have been reached. Several authors[6,14,15] propose strategies to avoid or to escape from the local minimum. These strategies have been adapted and applied to our MMS.

They are

- '*Wait*'. When a manipulator has reached a local minimum, it stops for an instant of time and does not plan its movements. In the following instant of time, the environment in which it has to move has changed (logically only if one of the obstacles has changed its position, that is, one of the other manipulators is in motion), and the actual position is no longer a local minimum. On carrying out the new path planning, the manipulator will automatically be released from the blockage.
- 'Random motion'. This technique generates a random and very small displacement of the manipulator that has 'fallen into' a local minimum. This is not a very efficient strategy in most motion problems because, if many random movements are made, the result could be a collision with an obstacle.
- 'Backtracking'. A strategy of backtracking is used, which aims to take out the manipulator from the local minimum. However, instead of returning along its own path, the manipulator is forced to backtrack considering only the obstacles (only the repulsive potential), ignoring the final goal.

https://doi.org/10.1017/S0263574706002888 Published online by Cambridge University Press

– 'Increase'. The repulsive potential is increased at those points of the workspace that have already been reached by the manipulator.

5. COMPARATIVE ANALYSIS OF PATH-PLANNING METHODS: BASIC MOTION PROBLEMS DEFINITION

Before implementing the three path-planning methods studied in this paper on our MMS, its expected behaviour has been simulated using the 'MMS path generator' software package. This software has been developed and programmed by our research group at the University of Valladolid using C++ and MATLAB. The simulations were carried out on a Pentium IV PC at 3.1 GHz, 1024-Mb RAM.

To develop a comparative study of the three path-planning methods, two types of experiments were defined:

– E2.X: Experiments with two manipulators in movement:

Two manipulators perform pick-and-place operations moving between tables. They might cross in common areas. Each moving manipulator is a dynamic obstacle for the other. Four kinds of path-planning experiments of increasing complexity are defined.

- E2.1: Two moving manipulators. The third manipulator is stopped far from them (it is not a static obstacle).
- E2.2: Two moving manipulators. The third is stopped, occupying the lower area of the common workspace and constitutes a static obstacle.
- E2.3: Two moving manipulators. The third is stopped, occupying the upper area of the common workspace and constitutes a static obstacle.
- E2.4: Two moving manipulators. The third is stopped, occupying the middle area of the common workspace and constitutes a static obstacle.

– E3.X: Experiments with three moving manipulators:

Three manipulators are performing pick-and-place operations between their side-tables, and between their side-tables and the central table. They might cross in common areas. Three types of path-planning experiments of increasing complexity are defined.

- E3.1: All three manipulators move simultaneously from one side-table to another, performing tasks that occupy the same area of the common workspace.
- E3.2: Two manipulators move from their side-tables to the central table, and the other is moving between its side-tables.
- E3.3: The three manipulators move from their side-tables to the central table.

These experiments do not represent the manipulators' motion from one specific configuration in one table to another configuration in another table. They represent 'any kind of task' that the manipulators might perform between working areas. The initial and final configurations of the manipulators might vary inside the initial and final areas, and the relative delay of the manipulators might vary too. Thus, the manipulators might find many different types of conflicts in the same type of experiment.

Once we have defined the experiments, we want to train our manipulators to do them. In order to do this, we perform a series of simulations or tests that will tell us how effective a path-planning method is (potential, strategy-based, dynamic tunnel) for performing these experiments. Its purpose is to look for the method that gives us the best parameters: longitude of the planned path, number of points to the critical distance, number of faults. We define 'fault' in this context as a deadlock or a livelock (an oscillation of two manipulators, for example, each one trying to avoid the other) of the manipulator system inside the workspace.

For the simulation results to be valid, the experiments must be independent.[28] Since we are dealing with experiment problems that involve several manipulators moving at the same time, it is not easy to look for a set of N independent motions. Our experiments are a continuous simulation of the operation of two or three manipulators of our system. For example, experiments E2.X simulate two manipulators going back and forth between their side-tables, the final point of one of the pick-and-place tasks is the starting point of the next. Since we are doing a continuous simulation, the tasks might be correlated to each other; therefore, this dependency must be broken. The experiments must include all possible kinds of conflicts between manipulators, no matter when the motion of the manipulators starts and no matter in which part of the working platform the manipulators move. In reference,[28] a correlation analysis of the times gave us information about how correlated these experiments are.

Experiments E2.X describe 'two manipulators moving in the cell between their side-tables'. One way of describing all the possible conflicts of this general definition is by setting two TCP positions for each of the manipulators that are located in the centre of the initial and final working areas: $p_{\mathrm{oc}}^{\mathrm{TCP}}$, $p_{\mathrm{nc}}^{\mathrm{TCP}}$, respectively, that we call central configurations. The tasks of the manipulators are running in a continuous simulation where the manipulators go back and forth between configurations that are computed by adding a random (uniform distribution) amount to these central configurations, meeting on the way, the other manipulator that is also going back and forth between configurations computed by adding a random amount to its central configuration. The resulting positions cover an area in the manipulator workspace whose maximum length is the parameter $\varepsilon_{\mathrm{alea}}$. The delay times between tasks are generated using an exponential random distribution whose parameter is μ_{exp}.

In the simulations presented in this paper, the method described[28] has been used to guarantee independent experiments. The following parameters have been used in all simulated tasks of all the experiments:

- $\varepsilon_{\mathrm{alea}} = 0.1\,\mathrm{m}$; the random change in the central configurations.
- $\mu_{\mathrm{exp}} = 10$; exponential distribution that generates the random waiting times.
- $N = 100$; number of tasks performed by the manipulator in each experiment.

Once the tasks are generated, they are simulated in Matlab and C++. The results given by the simulation are a set of parameters:

https://doi.org/10.1017/S0263574706002888 Published online by Cambridge University Press

– PL: Length (in number of points) of the planned path for each manipulator of the MMS, for a given task. We know that a task is faulty when the obtained number of points needed to perform the task is greater than a certain amount. Let us call PL_k^j the path length of the manipulator 'j' in each of the tasks 'k' ($k = 1, 2, \ldots, N$), of each simulated experiment. If $PL_k^j \geq PL_{max}$, a fault is reported in task 'k'. PL is defined as

$$PL = \frac{1}{m \cdot N} \sum_{j=1}^{m} \sum_{k=1}^{N} PL_k^j$$

where 'm' indicates the manipulators moving in the task.

– CP: Number of critical points, expressed in % of the total number of calculated points. We defined a threshold named 'critical distance' (= 100 mm, in all simulated tasks). Every point on the calculated path, whose distance to the closest obstacle (static or dynamic) is less than this critical distance, is a critical point which has a high risk of collision due to its proximity to the obstacle. Thus, this parameter allows us to analyse the proximity of the planned paths to the obstacles. Let CP_k^j be the number of critical points for the manipulator 'j' in each of the tasks 'k' ($k = 1, 2, \ldots, N$), of each simulated experiment. CP is defined as

$$CP = \frac{1}{m \cdot N} \sum_{j=1}^{m} \sum_{k=1}^{N} CP_k^j.$$

– F: Faults. This is the number of tasks with faults, expressed in % of the total number of simulated tasks. Let F^j be the number of tasks with faults (in %) of the manipulator 'j' in 'N' tasks simulated of each type of experiment. F is given by

$$F = \left(\frac{1}{m}\right) \sum_{j=1}^{m} F^j.$$

6. SIMULATION RESULTS OF MOTION PROBLEMS

This section describes the results obtained in the simulations of the MMS carried out with the software developed for the three path-planning methods described earlier: dynamic tunnel, strategy-based and potential method, to the motion problems described in Section 5. The results obtained are compared and analysed.

6.1. Strategy-based path planning: simulation results

Table I shows the results of testing a simulation of $N = 100$ tasks of each of the experiments E2.X and E3.X, using the strategy-based path-planning method. The results obtained are presented using two different strategies.

- Optimum strategy (best strategy):
  - Experiments E2.X: manipulator 1 basically follows a 'go-up' strategy, lifting up the arm, forearm and hand, while manipulator 2 'bends-up-hand-down' lifting up joint 2 and decreasing the value of links 3 and 4.
  - Experiments E3.X: manipulator 1 goes up with hand-up, manipulator 2 'folds-up-hand–down', manipulator 3 goes up with hand-down.
- Non-optimum strategy (non-best strategy).

Table I. Results of testing a simulation of experiments E2.X and E3.X using strategy-based path-planning method.

Experiment type	Strategy	Average values PL	CP (%)	Faults (%)	DC
E2.1	Optimum	49	48	0	1
	Non-optimum	54	51	0	1
E2.2	Optimum	61	80	0	4
	Non-optimum	62	80	0	2
E2.3	Optimum	48	62	0	2
	Non-optimum	55	69	0	1
E2.4	Optimum	57	88	0	1
	Non-optimum	84	88	0	7
E3.1	Optimum	48	35	0	0
	Non-optimum	60	43	0	3
E3.2	Optimum	49	63	0	1
	Non-optimum	74	44	0	2
E3.3	Optimum	51	78	0	2
	Non-optimum	157	92	0	31

For all experiment types, the number of tasks is 100. PL: path length (number of points of the planned path); CP: critical points; DC: direction change.

Figure 5 (obtained starting from Table I) shows that the paths obtained with optimum strategies are short and, approximately, of the same length (48 ÷ 61 points for experiments E2.X, and 48 ÷ 51 points for experiments E3.X). In spite of having increased the complexity of the problem, the method provides paths of almost identical length for all the experiments carried out with two or three manipulators. Even in the most complex problems (E3.1, E3.2, E3.3), the paths are slightly shorter. This is because, as all the obstacles are mobile, the optimum strategy adequately shares out the

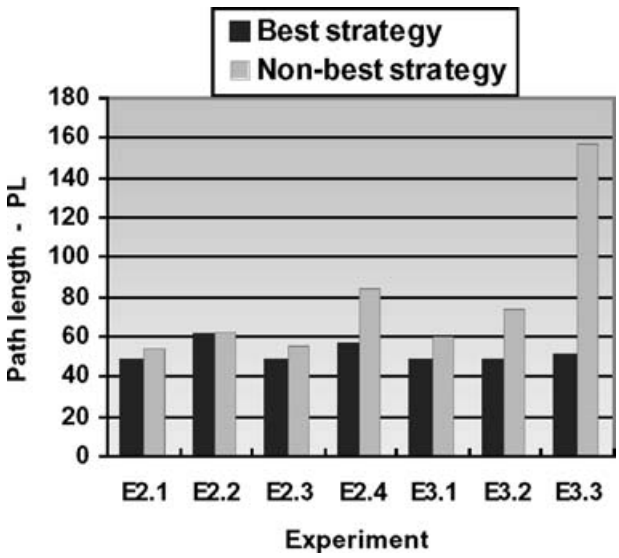


Fig. 5. Path length using strategy-based path-planning method.

https://doi.org/10.1017/S0263574706002888 Published online by Cambridge University Press

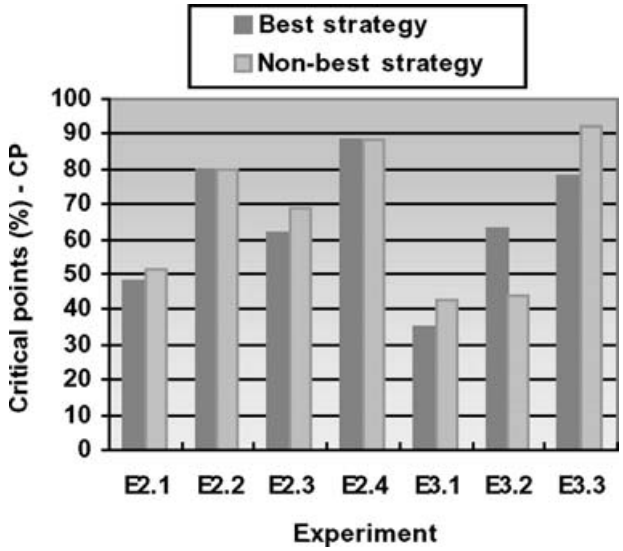


Fig. 6. Percentage of critical points using strategy-based path-planning method.

available workspace, avoiding close encounters of the manipulators. However, in the experiments E2.X, one of the manipulators is stopped, and permanently occupies a portion of the workspace that cannot be "shared out" among the moving manipulators.

The method based on strategies with optimum plan is not very sensitive to the increase in complexity of the tasks for two or three manipulators in motion. This indicates the importance of a good calculation of the optimum strategy.

With non-optimum strategies, the paths are longer and less uniform (54 ÷ 84 points for experiments E2.X, 60 ÷ 157 points for experiments E3.X). Paths of increasing length are generated and the increase is proportional to the increase in the complexity of the experiments. Thus, for example in E3.3, not using an optimum plan generates an increase in path length of approximately 300% (157 points as opposed to 51), due to the high number of backtracks and changes of direction the manipulators must carry out to complete the task.

Figure 6 (obtained starting from Table I) shows that the strategy-based path planning gives very high percentage of points at a critical distance (100 mm), between 60 and 90%, except in less complex tasks (E2.1 ≈ 50% and E3.1 ≈ 40%), and does not show any dependence on the optimum plan. This planning method generates paths very close to the obstacles.

Finally, by the analysis of Table I, it can be seen that the number of paths with faults (blockages) is zero (out of the 1.400 tests carried out), which demonstrates the high efficiency of the strategies-based method.

6.2. Potential path planning: simulation results

Table II shows the simulation results obtained in the path planning for MMS using the potentials method. Figures 7 and 8 are obtained from this table. On analysing the results of Fig. 7, it can be seen that a complex experiment (E3.3: 52 points) provides shorter paths than a simpler experiment (E2.3: 75 points). This is because in experiment E3.3, the local minima that can appear are resolved by applying back-

Table II. Results of testing a simulation of experiments E2.X and E3.X using potential path-planning method.

	Average values		
Experiment type	PL	CP (%)	Faults (%)
E2.1	44	45	0
E2.2	71	49	30
E2.3	75	36	20
E2.4	57	70	50
E3.1	56	48	0
E3.2	47	74	33
E3.3	52	69	60

For all experiment types, the number of tasks is 100. PL: path length (number of points of the planned path); CP: critical points.

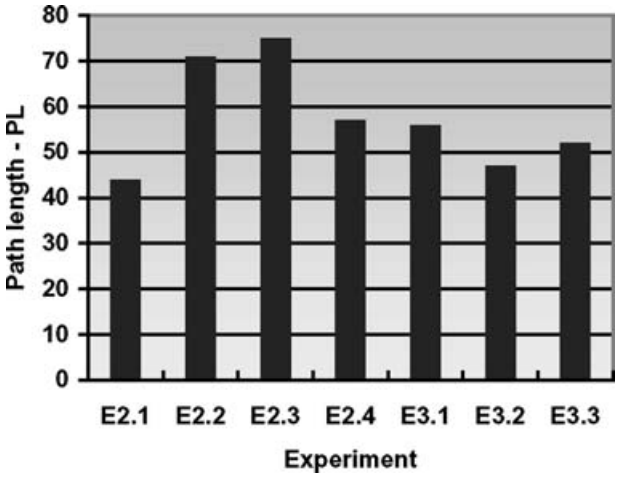


Fig. 7. Path length using potential path-planning method.

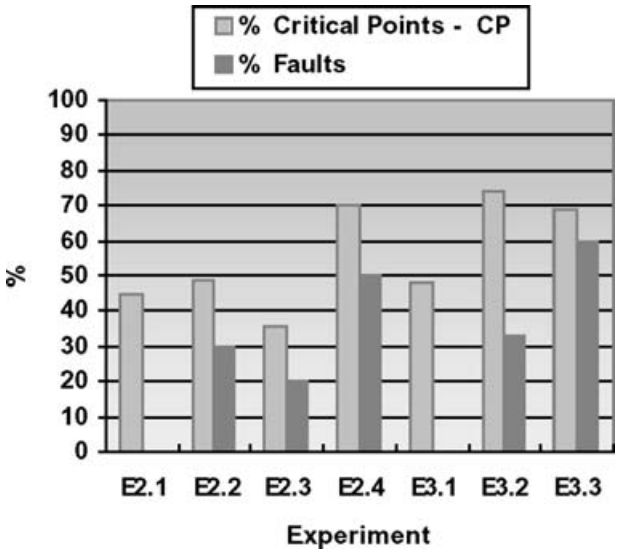


Fig. 8. Percentage of faults and critical points using potential path-planning method.

tracking and/or random motion to two of the three manipulators in motion. However, in E2.3, backtracking and/or random motion is only applied to one of the two manipulators. Thus, in E3.3, the environment changes more

https://doi.org/10.1017/S0263574706002888 Published online by Cambridge University Press

Table III. Analysis of the influence of the bubble diameter using dynamic tunnel path planning [Experiment 2.3 (E2.3)].

Bubble diameter (mm)	Number of bubbles	PL	Faults (%)
10	60	–	100
15	50	79	30
20	35	64	42
25	30	77	43
30	20	60	42
40	20	64	45
50	20	55	95

PL: path length (number of points of the planned path).

rapidly and the local minimum that may appear is more easily resolved, resulting in shorter paths.

The analysis of Fig. 8 shows that, in the experiments with complex tasks (E3.2 and E3.3), the percentage of critical points ($\approx 70\%$) is greater than in experiments with simple tasks (E2.1, E2.2, E2.3: $36\% \div 49\%$), as the three manipulators occupy an important part of the common workspace and have to move very close to each other. It can also be seen that the potentials method results in a high number of paths with blockages, $20\% \div 60\%$, according to the increase in the experiment's complexity. In fact, as the available volume of work for the three manipulators is small, it is also necessary to carry out additional movements to be able to get rid of the local minima.

6.3. Dynamic tunnel path planning: simulation results

Table III shows the result of the analysis of the influence of the bubble's diameter on path planning for the MMS. Experiment E2.3 has been used for this analysis, using bubbles with diameters varying between 10 and 50 mm (which also means that the dynamic tunnel should have between 20 and 60 bubbles, respectively, to guarantee their continuity, keeping in mind the dimensions of the manipulators' workspace). One hundred tasks have been simulated for each bubble diameter. The results (Fig. 9) show that the most adequate path (shorter and with fewer blockages) is obtained for a dynamic tunnel made up of 20 bubbles of 30 mm in diameter. These values

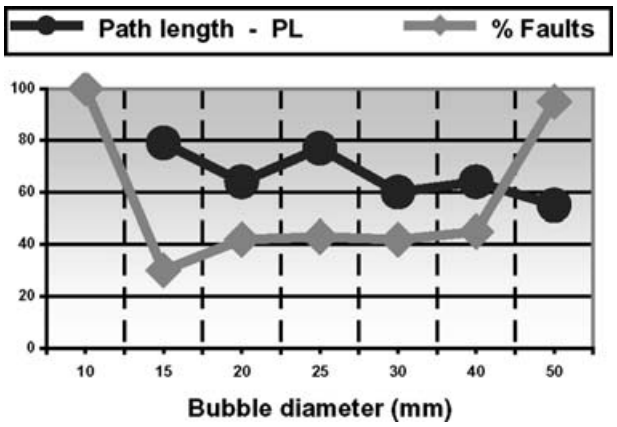


Fig. 9. Dynamic tunnel method: influence of bubble diameters and number of bubbles over path length.

Table IV. Simulation results of experiments E2.X and E3.X using dynamic tunnel path planning.

	Average values	
Experiment type	PL	Faults (%)
E2.1	49	40
E2.2	70	20
E2.3	60	42
E2.4	–	100
E3.1	64	60
E3.2	–	100
E3.3	–	100

For all experiment types, the number of tasks is 100. Bubble diameter: 30 mm. Number of bubbles: 20. Deformation frequency (f_d) = 2. PL: path length (number of points of the planned path).

have been used in the simulations of all the experiments carried out.

Table IV presents the simulation results obtained in the path planning of the MMS using the dynamic tunnel method. The results obtained are represented in Fig. 10. They show that this method is not really adequate for MMS path planning, because it produces a great number of blockages in most of the proposed tasks, reaching 100% in some experiments. This is because the common workspace for the three manipulators is already reduced (due to their geometric disposition, very close to each other), and it is difficult to calculate a dynamic tunnel, 30 mm in diameter, inside the free workspace, to guarantee a collision-free path, even when the tunnel is dynamic and is recalculated at each instant of the path.

Applying the dynamic tunnel method to E2.X experiments produces more blockages as the complexity of the task increases, and for the most complex experiment (E2.4), a collision-free path is never achieved. For tasks in which a solution is obtained, the path lengths ($49 \div 70$ points) are very similar to those obtained by the other planning methods. When there are three manipulators in motion and the task is

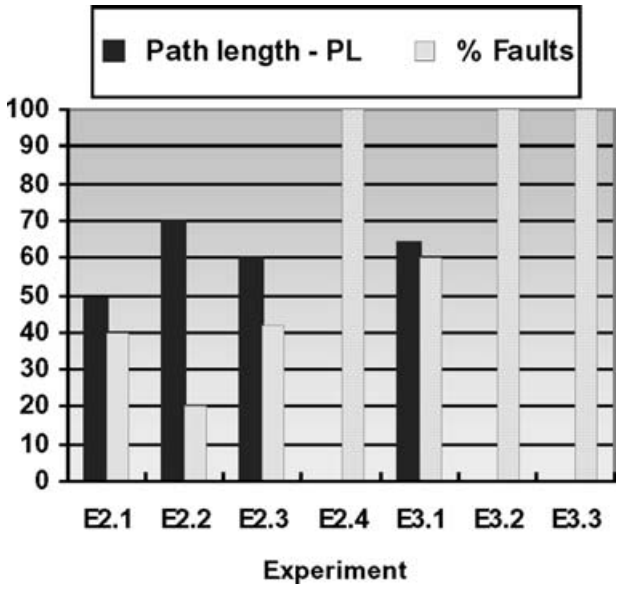


Fig. 10. Percentage of faults and critical points using dynamic tunnel path planning.

https://doi.org/10.1017/S0263574706002888 Published online by Cambridge University Press

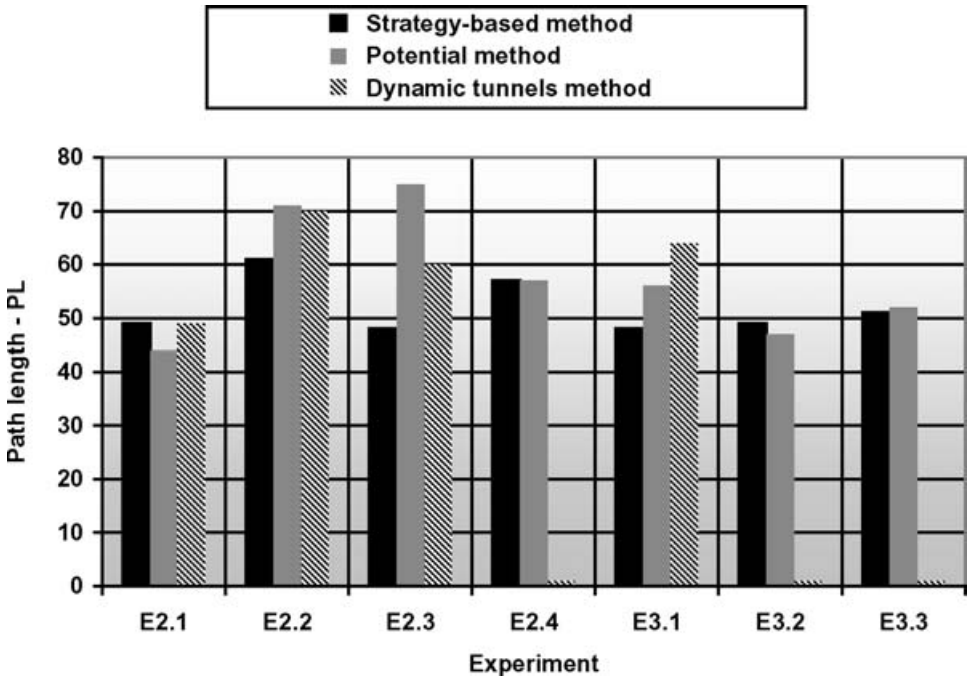


Fig. 11. Comparative analysis of path length for three path-planning methods.

simple (E3.1), the dynamic tunnel method is useful as it generates paths that are not very long (64 points), although with a high percentage of blockages (60%).

6.4. Comparative analysis

Figure 11 shows a comparative analysis of path length (PL parameter) generated by the simulation of the three path-planning methods studied in this paper, applied to MMS. The analysis of this figure provides the following conclusions:

– The "strategy-based" method generates the shortest paths when the optimum plan is chosen. An increase in the complexity of the tasks does not cause an increase in path length. However, in the experiments E2.X, as one of the manipulators is static, that portion of the workspace is always occupied and cannot be "shared out" among the manipulators in motion. This explains why, in more complex tasks with three manipulators in movement, the number of critical points is lower (see Fig. 12).
– The potentials method generates longer paths because the appearance of local minima forces new movements to be carried out in order to try to get free from the said minima, thus increasing the path length of the manipulators. The dynamic tunnel method provides path lengths very similar to those provided by other methods in tasks for which a path is obtained.

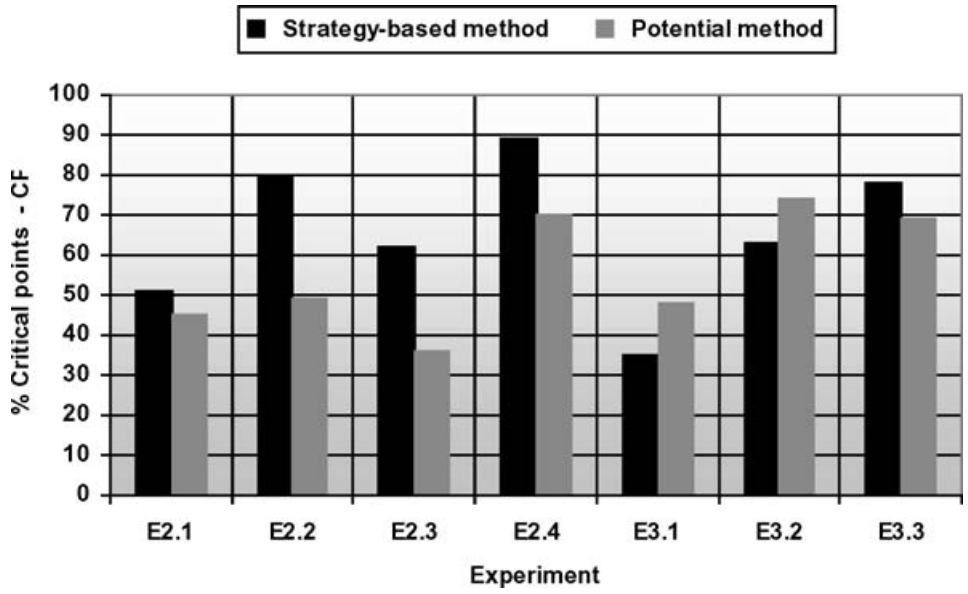


Fig. 12. Comparative analysis of path critical points for two path-planning methods.

https://doi.org/10.1017/S0263574706002888 Published online by Cambridge University Press

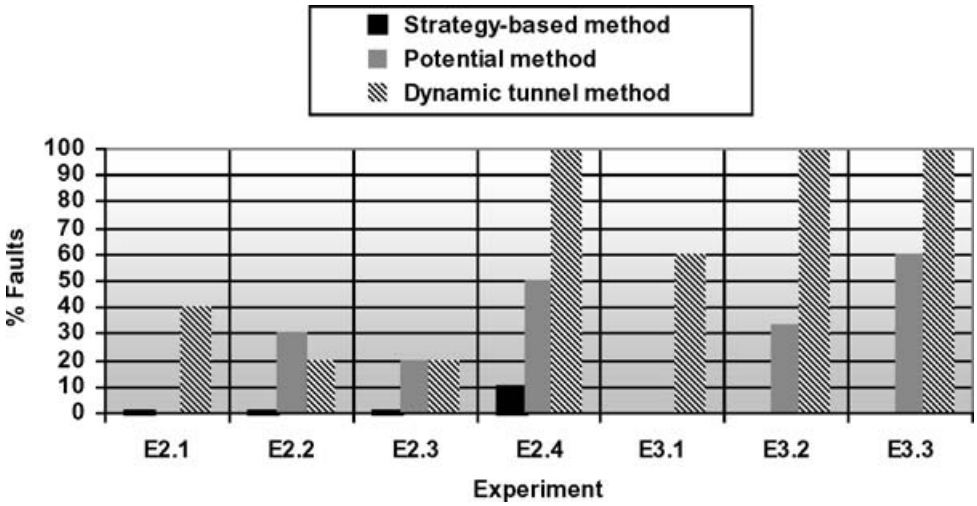


Fig. 13. Comparative analysis of faults for three path-planning methods.

Thus, strategy-based method is the most adequate path-planning method for MMS, as long as the optimum strategy is chosen using the proposal presented in reference.[28]

Figure 12 shows a comparative analysis of path critical points (CP parameter) generated by simulation of the strategy-based and potential methods. The dynamic tunnel method has not been considered due to the bubbles having a diameter of 30 mm. From the analysis of this figure, it can be seen that the strategy-based method generates the greatest number of critical points, and therefore, paths closer to the obstacles.

Figure 13 shows the percentage of faults in path planning for the three proposed methods. The strategy-based method always obtains a collision-free path (in the 1.400 tasks carried out). The potentials method, however, produces blockages in more complex experiments due to the appearance of local minima. The dynamic tunnel method results in a very high number of blockages as the reduced volume of work for the manipulator means that the tunnel cannot be in free space.

Figure 14 indicates the mean times spent doing each simulation. It can be seen that the mean time used by the potentials method to calculate paths is 5 ÷ 8 s, for experiments with two or three manipulators, respectively. The quickest method is the strategy-based method (a mean of 2 s to generate the path), and the worst, the dynamic tunnel method, which, in those cases where it is possible to find a collision-free path, needs an average of 60–145 s for the tasks with two and three manipulators, respectively.

A global analysis of the three path-planning methods discussed in this paper, running on our MMS, leads to the following conclusions:

- Strategy-based path planning always finds a solution of short length to a given motion problem. The time spent for planning is very short due to the simplicity of the calculations to be carried out, since the method is local and uses the concept of reduced subspace. The main inconvenience in this method is that the length of the paths can increase unnecessarily if the optimum strategy is not used. In fact, this planning method results in manipulators being very "short-sighted" (local planner), and when two manipulators in motion approach each other, they take avoiding action in the direction given by the chosen strategy. If this strategy is not the optimum one, both

COMPUTING TIMES SPENT IN SIMULATION (average values)			
Hardware: PC Pentium IV, 3.1 GHz, 1024 Mb RAM Software: C++, Matlab 5.3			
	Strategy-based mehod (Optimum strategy)	Potential method	Dynamic tunnel
Experiments with two manipulators - E2.X	2 sec.	5 sec.	60 sec.
Experiments with three manipulators - E3.X	2 sec.	8 sec.	145 sec.

Fig. 14. Times (maximum) spent in simulations for the three path-planning methods running for the MMS.

https://doi.org/10.1017/S0263574706002888 Published online by Cambridge University Press

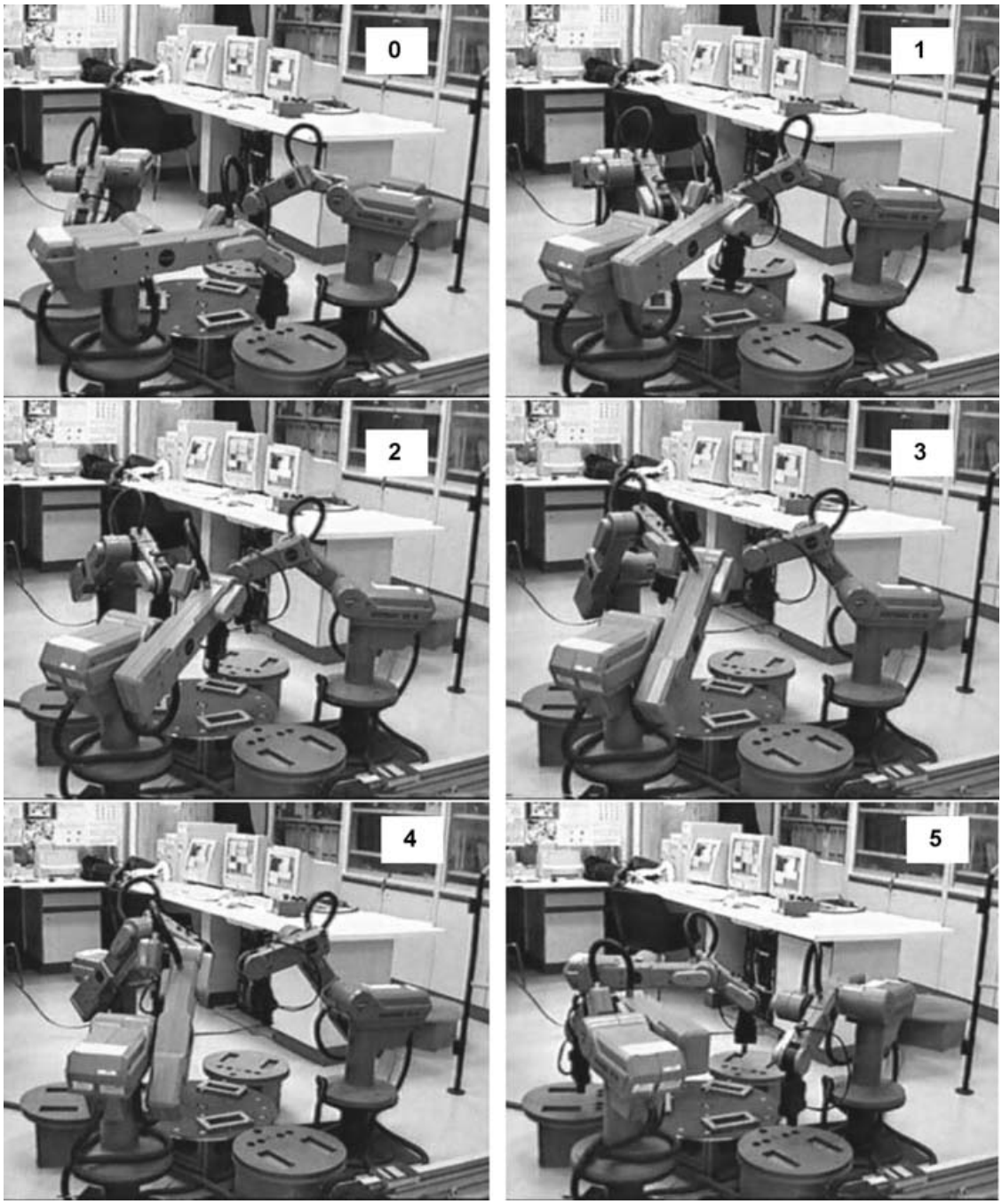


Fig. 15. Snapshots of a path where the three robots move at the same time using strategy-based path planning. The robots go from their right to their left with the optimum strategy.

manipulators may move in such a way as to obstruct the other one. The way to correct this is to always choose the optimum strategy.

- The potential method provides better results whenever the environment is more variable. Tasks with three manipulators in motion are shorter than tasks with two manipulators, due to the greater dynamic character of the environment when there are three manipulators in motion. It has the inconvenience that the local minima cannot be completely suppressed, so the system becomes blocked to a greater or lesser extent, depending on the type of experiment.
- The dynamic tunnel method becomes less efficient when the complexity of the problem increases and the environment becomes more changeable. For simple problems, it generates very smooth and elegant paths, yet cannot find solutions for complex problems (reaching 100% of faults). This method, therefore, is limited for use in problems with two manipulators in motion with not very complex tasks. Another great inconvenience is the high calculation time.

7. EXPERIMENTAL SET-UP

To analyse the behaviour of the three planning methods, they were implemented and tested on the MMS of our laboratory. It is composed of three Scorbot-ER IX manipulators with five rotating links each, and several working platforms (see

https://doi.org/10.1017/S0263574706002888 Published online by Cambridge University Press

Fig. 1). One of the platforms is located in the centre and is used by the three manipulators as the assembly platform. The other three tables are located at both sides of each manipulator and are used for parts and product release and storage. Each manipulator can access two side-working tables and the central area.

The control architecture of the MMS is implemented in a distributed way: Each manipulator has its own controller, managed by a Pentium PC. The control and the path-planning algorithms (written in C++) run simultaneously into these three Pentium PCs as independent and distributed processes. These computers are linked through a local Ethernet network and the programs are run under the Linux O. S. Each manipulator knows the actual position of the rest of the manipulators before planning its next movement, since the encoder values are transmitted, and the kinematic model of the manipulator is used to calculate its Cartesian position (TCP). Despite the fact that the encoder information is very accurate, the actual position of each manipulator is subject to some degree of uncertainty, since the communication of the data between the manipulators takes some time, and the manipulators might move in that interval. This uncertainty in the position of the obstacle–manipulators is bounded using appropriate margins of security, which can be calculated once the maximum speed of the manipulators is limited.

The tasks running on the MMS are mainly pick-and-place operations between the working platforms (experiments E2.X and E3.X). Only gross motion is considered, and it is assumed that no joint manipulation of pieces is needed.

Figure 15 shows a sequence of snapshots from the execution on the real MMS (15 degree-of-freedom) using the strategy-based path-planning method. The snapshots have been obtained while running experiment E3.1: the three manipulators are moving at the same time from the table at their right to the one at their left, using the "best strategy" described in Section 6.1.

8. CONCLUSIONS

An adaptation (that we call dynamic tunnel) of the elastic-strips framework proposed by Brock and the concept of bubble proposed by Quinlan has been developed and applied to our MMS, for path-planning in dynamic environments, in problems that involve two and three manipulators moving at the same time in common areas. This path-planning method along with two other path-planning methods was developed by the authors for the MMS: the planning method based on strategies and that based on potentials, have been presented.

These three path-planning methods have been tested in our MMS for on-line path planning of two and three manipulators in pick-and-place operations. The main contribution of this paper is the implementation of several simulations for on-line path planning in dynamic environments that gives an estimation of how suitable a path-planning method is for a certain kind of task. These simulations have been done carefully assuring independent experiments.

The results obtained from the simulations of the three planning methods have been presented and analysed. Factors such as the complexity of the task (movements from one side-table to another, movements from the side-tables to the central table) and the number of dynamic obstacles (manipulators in movement) have a major impact on the performance of the path-planning method.

The results obtained show that the strategy-based method is the best path-planning method, since it always finds a solution of short length and short time to a given motion problem, provided the optimum strategy is chosen. Path planning with potential method provides better results when the environment is more dynamic. However, it also presents local minima that cannot be avoided in spite of using "backtracking", "random motion", "wait". The dynamic tunnel is the worst path-planning method. It becomes less efficient with increasing complexity of the problem and with more dynamic environment. Another great inconvenience is the high calculation time.

Acknowledgements

The authors would like to thank the reviewers for their careful review and constructive comments on earlier versions of this paper. This work was partially supported by the Spanish research project DPI2003-09705-C02-01.

References

1. R. C. Arkin, *Behavior-Based Robotics* (MIT Press, Cambridge, MA, 1998).
2. J. Barraquand, L. Kavraki, J. C. Latombe, T.-Y. Li and P. Raghavan, "A random sampling scheme for path planning," *Int. J. Robot. Res.* **16**(6), 759–773 (1997).
3. J. Barraquand, B. Langloids and J. C. Latombe, "Numerical potential field techniques for robot path planning," *IEEE Trans. Syst. Man Cybern.* **22**(2), 224–241 (1992).
4. O. Brock, Generating Robot Motion: The Integration of Planning and Execution *Ph.D. Thesis* (Stanford, CA: Stanford University, 2000).
5. O. Brock and O. Khatib, "Elastic strips: A framework for integrated planning and execution," *Int. Symp. Exp. Robot.* 245–254 (1999).
6. H. Chang, "A new technique to handle local minimum for imperfect potential field based motion planning," *IEEE Int. Conf. Robot. Autom.* 108–112 (1996).
7. C. I. Conolly, "Harmonic functions and collision probabilities," *Int. J. Robot. Res.* **16**(4), 497–507 (1997).
8. B. Faverjon and P. Tournassoud, "A local based approach for path planning of manipulators with a high number of degrees of freedom," *IEEE Int. Conf. Robot. Autom.* 1152–1159 (1987).
9. J. C. Fraile, C. J. Paredis, C.-H. Wang and P. K. Khosla, "Agent-based planning and control of a multi-manipulator assembly system," *IEEE Int. Conf. Robot. Autom.* 1219–1225 (1999).
10. S. S. Ge and Y. J. Cui, "New potential functions for mobile robot path planning," *IEEE Trans. Robot. Autom.* **16**(5), 615–620 (2000).
11. L. Gouzénes, "Strategy for solving collision-free path problems for mobile and robotic manipulators," *Int. J. Robot. Res.* **3**(4), 51–65 (1984).
12. K. Gupta, "The sequential framework for practical motion planning for manipulator arms: Algorithm and experiments," **In:** Practical Motion Planning in Robotics (K. Gupta and A. P. del Pobil, eds.), (Wiley, New York, 1998) pp. 9–31.
13. K. Gupta and A. P. Pobil, eds., *Practical Motion Planning in Robotics: Current Approaches and Future Directions* (Wiley, New York, 1998).
14. K. Gupta and Z. Guo, "Motion planning for many degrees of freedom: Backtracking with sequential search," *IEEE Trans. Robot. Autom.* **11**(6), 897–906 (1995).

https://doi.org/10.1017/S0263574706002888 Published online by Cambridge University Press

15. K. Gupta and Z. Zhu, "Practical global motion planning for many degrees of freedom: A novel approach within sequential framework," *J. Robot. Syst.* **12**(2), 105–117 (1995).
16. K. Hamilton and G. I. Dodds, "Reactive planning of robot arms in single and cooperative tasks," *IEEE Int. Conf. Robot. Autom.* 336–340 (1998).
17. L. Kavraki, P. Svestka, J. C. Latombe and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Autom.* **12**(4), 566–580 (1996).
18. O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Robot. Res.* **5**(1), 90–98 (1986).
19. D. H. Kim *et al.*, "Local Path Planning Based on New Repulsive Potential Functions with Angle Distributions,". *Proceedings of the IEEE International Conference on Information Technology and Applications (ICITA'05)*, Sydney, Australia (2005), pp. 745–750.
20. P. Leven and S. Hutchinson, "Toward Real-Time Path Planning in Changing Environments," *Proceedings of the Workshop on Algorithmic Foundations of Robotics (WAFR)*, Hanover, NH (2000), pp. 363–376.
21. T. Y. Li and J. C. Latombe, "On-line manipulation planning for two robot arms in a dynamic environment," *IEEE Int. Conf. Robot. Autom.* 1048–1055 (1995).
22. S. R. Lindemann and S. M. LaValle, "Current Issues in Sampling-Based Motion Planning," *Proceedings of the 11th International Symposium on Robotics Research (ISRR)*, Siena, Italy (2003).
23. A. McLean and I. Mazon, "Incremental roadmaps and global path planning in evolving industrial environments," *IEEE Int. Conf. Robot. Autom.* 101–107 (1996).
24. M. Mediavilla, J. C. Fraile and G. I. Dodds, "Optimisation of collision-free paths in multi robot system," *IEEE Int. Conf. Robot. Autom.* 2910–2915 (1998).
25. M. Mediavilla, J. C. Fraile and J. R. Peran, "A multi robot system of three manipulators with on-line path planning abilities," *Proceedings of the LAAS-CNRS International Symposium on Intelligent Robotic Systems*, Toulouse, France (2001), pp. 47–58.
26. M. Mediavilla, J. L. González, J. C. Fraile and J. R. Perán, "On-line motion planning for robotic arms: New approach based on the reduction of the search space," *IEEE Int. Conf. Robot. Autom.* 3825–3830 (2002).
27. M. Mediavilla, J. L. González, J. C. Fraile and J. R. Perán, "Reactive approach to on-line path planning for robot manipulators in dynamic environments," *Robotica* **20**, 375–384 (2002).
28. M. Mediavilla, J. C. Fraile, T. González and I. J. Galindo, "Selection of strategies for collision-free motion in multi-manipulator systems," *J. Intell. Robot Syst* **38**, 85–104 (2003).
29. S. Quinlan. Real-Time Modifications of Collision-Free Paths *Ph.D. Thesis* (Stanford, CA: Stanford University, 1994.)
30. S. Quinlan and O. Khatib, "Elastic bands: Connecting path planning and control," *IEEE Int. Conf. Robot. Autom.* 802–807 (1993).
31. E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Trans. Robot. Autom.* **8**(5), 501–518 (1992).
32. K. Souccar, J. Coelho, C. Conolly and R. Gruppen, "Harmonic functions for path planning and control," **In:** Practical motion planning in robotics (K. Gupta and A. P. del Pobil, eds.), (Wiley, New York, 1998) pp. 277–299.

https://doi.org/10.1017/S0263574706002888 Published online by Cambridge University Press