



Planning and visual-servoing for robotic manipulators in ROS

R. I. Maldonado-Valencia¹ · C. H. Rodriguez-Garavito¹ · C. A. Cruz-Perez¹ · J. S. Hernandez-Navas¹ · D. I. Zabala-Benavides¹

Received: 6 February 2022 / Accepted: 17 July 2022 / Published online: 27 August 2022
© The Author(s) 2022

Abstract

This article presents a probabilistic road map (PRM) and visual servo control (visual-servoing) based path planning strategy that allows a Motoman HP20D industrial robot to move from an initial positional to a random final position in the presence of fixed obstacles. The process begins with an application of the PRM algorithm to take the robot from an initial position to a point in space where it has a free line of sight to the target, to then apply visual servoing and end up, finally, at the desired position, where an image captured by a camera located at the robot's end effector matches a reference image, located on the upper surface of a rectangular prismatic object. Algorithms and experiments were developed in simulation, specifically, the visual servo control that includes the dynamic model of the robot and the image sensor subject to realistic lighting were developed in robot operating system (ROS) environment.

Keywords Path planning · Visual servoing · HP20D robot · ROS

1 Introduction

Automatic robot movement is a complex task requiring an integration of different areas of engineering including computer vision, dynamic systems modeling, and robot control. Autonomous robotics is developed mainly using two approaches: path planning and visual servoing, both of which allow guiding a robot through a series of positions in the joint or task space, from an initial to a desired position.

The first approach is path planning, a field of research that seeks collision-free movement for a robot without

considering dynamic model. The techniques addressed by path planning include soft computing-based algorithms (Beom and Cho 1995; Hu et al. 2004; Garcia et al. 2009); analytical path definition-based techniques such as potential fields (Koren et al. 1991; Borenstein and Koren 1991), where field lines are mathematically modeled to connect points in space and which suffer deformations as a result of objects present therein; and sample-based planning (SBP) path planning (Barraquand and Latombe 1991). This notion of sampling the configuration space gave way to the most used planning algorithms, namely probabilistic roadmap method (PRM), algorithms based on searching for paths using a roadmap (Kavraki et al. 1996); and rapidly-exploring random trees (RRT) (LaValle 1998).

The PRM algorithm is used to solve for multiple paths in an environment, whereas RRT is used for specific planning and, as such, is faster at resolving a single path. Both algorithms are probabilistically complete. These base algorithms have been modified, so they work in unstructured, dynamic environments generating smooth and optimum paths. One example of these is the smoothly RRT (S-RRT) planner (Wei and Ren 2018).

Applications that use path planning include molecular modeling and interaction (Al-Bluwi et al. 2012; Gipson et al. 2012), autonomous navigation (Yang et al. 2013), non-linear

R. I. Maldonado-Valencia and C. H. Rodriguez-Garavito contributed equally to this work.

✉ C. H. Rodriguez-Garavito
cerodriguez@unisalle.edu.co

R. I. Maldonado-Valencia
rmaldonado19@unisalle.edu.co

C. A. Cruz-Perez
ccruz54@unisalle.edu.co

J. S. Hernandez-Navas
jhoanshernandez40@unisalle.edu.co

D. I. Zabala-Benavides
dzabala13@unisalle.edu.co

¹ Automation Engineering, Universidad de La Salle, Cra. 2 # 10-70, Bogotá D.C. 111711, Colombia

dynamic systems control (Branicky et al. 2003), digital animation (Kuffner 2000), among others.

The second approach is robot control, which is where visual servoing can be found. This is a joint control technique that allows defining a robot's movement based on information captured by a visual sensor in the robot's workspace. Three types of visual servoing exist depending on how this information is used: image based visual servoing (IBVS), position based visual servoing (PBVS) and hybrid visual servoing (HVS). IBVS control (Espiau et al. 1992; Malis 2004; Marey and Chaumette 2008), uses coordinates on an image plane corresponding to observation points S that vary over time and are used to calculate an error associated to the reference, which is then used to calculate a compensation signal to guarantee its convergence to zero. PBVS control (Thuilot et al. 2002; Allen et al. 1993; Wilson et al. 1996), takes the object's orientation and positioning as parameters that are compared to the reference to calculate its error, while the HVS control method (Malis et al. 1999), is based on an integration of the above methods to formulate the control law. Visual control can also be classified according to the camera position. If the camera is static in the robot's workspace, it is called an eye-to-hand configuration. If the camera is positioned on the robot, i.e., if the robot's movement moves the camera, the configuration is called eye-in-hand.

Another consideration related to the visual controller architecture is the control loop structure. Two cases are considered here. In the first, the visual control law provides set points that are input into an "internal" controller to the robot, which controls the robot's joints. This is known as dynamic look-and-move. In the second architecture, the visual controller sends information directly to the joints. This is known as direct visual servo control.

These methodologies have been implemented in diverse areas, including medical applications (Azizian et al. 2014; Wei et al. 1997; Krupa et al. 2003), robotic positioning and manipulation tasks (Vahrenkamp 2008; Kragic 2003), object tracking (Chaumette et al. 1991; Allen et al. 1993), teleoperation tasks (Gridseth et al. 2015; Suetsugu et al. 2014), mobile robots (Fang et al. 2005), and industrial applications (Nelson et al. 1993; Nomura and Naito 2000; Zhou et al. 2006).

Some work has used both approaches together to make robotic systems more flexible, autonomous, and robust (Chesi and Hung 2007; Kazemi et al. 2010; Deng et al. 2005).

This article is structured as follows: Sect. 2 presents the base path planning and visual servoing methods and algorithms for industrial robots. It also explores the theory related to homographic decomposition, used to estimate a camera's position depending on observed visual features, and the sensor that will be used for visually controlling the

HP20D robot. Then, Sect. 3 shows the implementation of the algorithms and experiments to validate operation and performance. Finally, Sect. 4 contains a discussion of the results obtained and posits conclusions from the automatic path planning strategy for a manipulator robot.

2 Methods

2.1 Robot model environment

The proposed path planning strategy, in its obstacle avoidance, requires the HP20D robot's direct and inverse kinematic models for implementation. In other words, given the joint position of a robot Q with six joints, we require establishing the position of its end effector P , defined by its coordinates $^{base}\{x, y, z\}$ and its spatial orientation $^{base}\{\alpha, \beta, \gamma\}$, with respect to a base reference system $\{base\}$. These models, together with their mathematical deduction using the Denavit-Hartenberg approach can be found in Camacho et al. (2018).

On the other hand, the visual servoing phase is implemented based on the direct and inverse Jacobian and rigid body motion theory. The robot's physical description is a core element of this model, as its materials, inertias, and lengths intervene in the dynamic phenomena that interact between links when the robot is in movement. This dynamic model is incorporated in the ROS Gazebo simulator (Koenig and Howard 2004), as is the camera, which is used to capture images that are processed to feedback the velocities of the visual characteristics detected upon the target where we wish to take the robot's end effector. Figure 1 contains a view of the robot, its working environment, and the image sensor.

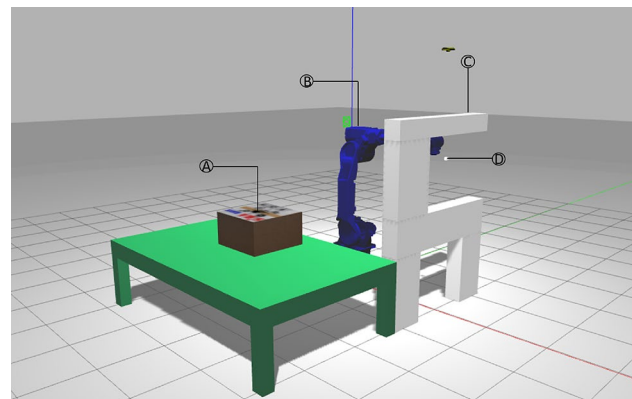


Fig. 1 View of HP20D robot (B), its workspace (A–C), and the image sensor in ROS simulator (D)

2.2 PRM-based path planning algorithm

The probabilistic roadmap path planning algorithm was initially presented in Kavraki et al. (1996). It includes two phases. The first of these includes construction of a non-directional graph $\mathcal{G} = (V, E)$ where the nodes V are parametric positions of a robot that are free of collision within the workspace, $\mathcal{Q}_{free}$; and the edges E are collision-free paths that are also in $\mathcal{Q}_{free}$ and are described using a local planner, Δ , which, in its simplest form, is a linear interpolator within a robot's joint space. The second phase, or planner query phase, consists of connecting an initial node, q_{init} , and a final node, q_{goal} , to $\mathcal{G}$, to seek the shortest possible collision free path. This query is performed by implementing an informed search algorithm as A^* , although a viable path cannot always be obtained for $\mathcal{G}$, if the roadmap does not completely cover the free space $\mathcal{Q}_{free}$. If this occurs, the graph $\mathcal{G}$ must be extended using new seed nodes placed using different techniques, for example as a function of the number of connections of the nodes that exist in $\mathcal{G}$.

A well-known variation of basic PRM is the so-called Lazy PRM, (Bohlin and Kavraki 2000). This version builds $\mathcal{G}$ including q_{init} and q_{goal} , over the entire $\mathcal{Q}$ space. It searches for the shortest path between the nodes of interest, $T_{shortest}$, and then checks the nodes and edges included in that path to see if they are free from collisions. If it detects points in the solution path that interfere with obstacles, these nodes or edges are disconnected from $\mathcal{G}$, and the process is repeated until a minimum collision free path T_{free} is found, or until seed nodes need to be introduced as it can see in Fig. 2.

In our approach, The Lazy PRM algorithm introduces some improvements over the version described by Bohlin and Kavraki, specifically in "Check path for collision" and in the node enhancement block.

The standard procedure for checking collision at every interpolated point along every segment belonging to a feasible path is to discretize each edge in $T_{shortest}$ at a step size fixed. In this work, we define a variable step size that decrements its magnitude gradually as iteration n runs according to a function

$f_{ss}(n)$. The step size is saved for each segment in E and recovered when the collision checker needs it, adjusting its value depending on the current iteration.

This procedure speeds up the algorithm and generates a feasible early path. Nevertheless, this path could be in collision because the test resolution might jump a small area of $\mathcal{Q}_{obs}$; therefore, the feasible path under test is checked again but at a minimum step size before validating $T_{shortest}$ as T_{free} .

Furthermore, in the node enhancement block, the original version establishes that when a path from q_{ini} to q_{goal} is not feasible, the Roadmap adds several nodes. Some of them are distributed uniformly in $\mathcal{Q}$ and the others around cue points called seeds, saved during the process that checks feasible path edges for collisions; the first point, where a collision occurs along a feasible path segment, is marked as seed. In this work, we set additionally as seeds every vertex in $\mathcal{Q}_{obs}$. This assumption is easy to meet because a 3D reconstruction of the robot's workspace generates a description of each obstacle as a set of vertexes.

2.3 Visual servoing

Visual control encompasses different areas, including robot modeling, control theory, and image processing. Implementation of this type of methodology can make robotic systems more flexible. This technique can be implemented using different types of cameras, whether monocular, stereo, and RGB-D. The information captured by the camera is compared to a reference to generate an error signal used as an input by the controller.

The work presented is based on an image based IBVS controller with an eye-in-hand configuration and a dynamic look-and-move architecture given their simplicity and advantages. A monocular camera is positioned at the end effector from which sufficient information is obtained for the control law. Some basic foundations of visual control theory that will be useful for developing this work are provided below.

First, a relationship between the robot's motion and changes in image features induced by an object observed in the camera's field of view is required when formulating the visual control law. Now, consider angular velocity ${}^c\Omega(t) = [\omega_x, \omega_y, \omega_z]^T$ and translational velocity ${}^cT(t) = [T_x, T_y, T_z]^T$ of the end effector transferred to a point ${}^cP = [x, y, z]^T$ attached to an object described in the camera frame $\{c\}$, in the form shown in 1 and 2.

$${}^c\dot{P} = {}^c\Omega \times {}^cP + {}^cT \quad (1)$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} z\omega_y - \frac{vz}{\lambda}\omega_z + T_x \\ \frac{uz}{\lambda} - z\omega_x + T_y \\ \frac{z}{\lambda}(v\omega_x - u\omega_y) + T_z \end{bmatrix} \quad (2)$$

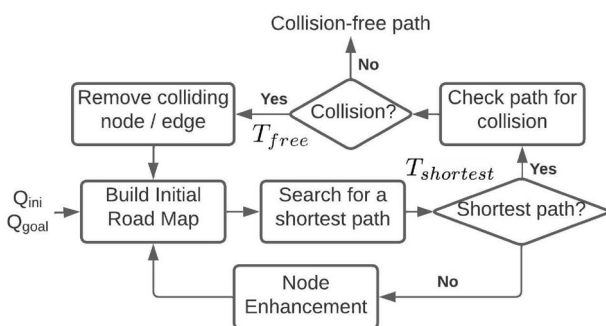


Fig. 2 Lazy PRM diagram

This point cP is projected on the image plane of the camera π with coordinates of pixel $s = [u, v]^T$, the relationship between the coordinates on image plane π and the coordinates of point cP is expressed by the pinhole camera model as:

$$\pi(x, y, z) = \begin{bmatrix} u \\ v \end{bmatrix} = \frac{\lambda}{z} \begin{bmatrix} x \\ y \end{bmatrix} \quad (3)$$

where λ is the focal length of the camera.

Velocities on the image plane can be obtained if we derive (3). If we substitute (2) in these derivatives, we obtain the following expression.

$$\underbrace{\begin{bmatrix} \dot{u}_1 \\ \dot{v}_1 \\ \vdots \\ \dot{u}_i \\ \dot{v}_i \end{bmatrix}}_S = \underbrace{\begin{bmatrix} \frac{\lambda}{z_1} & 0 & \frac{\lambda}{z_i} & 0 \\ 0 & \frac{\lambda}{z_1} & 0 & \frac{\lambda}{z_i} \\ -\frac{u_1}{z_1} & -\frac{v_1}{z_1} & \dots & -\frac{u_i}{z_i} & -\frac{v_i}{z_i} \\ \frac{u_1 v_1}{\lambda^2 + u_1^2} & \frac{-\lambda^2 - v_1^2}{\lambda} & \dots & \frac{u_i v_i}{\lambda^2 + u_i^2} & \frac{-\lambda^2 - v_i^2}{\lambda} \\ \frac{\lambda}{\lambda^2 + u_1^2} & \frac{u_1 v_1}{\lambda} & \dots & \frac{\lambda}{\lambda^2 + u_i^2} & \frac{u_i v_i}{\lambda} \\ \frac{\lambda}{\lambda} & \frac{u_1 v_1}{\lambda} & \dots & \frac{\lambda}{\lambda} & \frac{u_i v_i}{\lambda} \\ -v_1 & u_1 & \dots & -v_i & u_i \end{bmatrix}^T}_L \underbrace{\begin{bmatrix} T_x \\ T_y \\ T_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}}_V \quad (4)$$

The relationship between the movement of the features on the image plane and the speed of a point with respect to the camera is given in (4). The matrix that relates them is commonly known as a Jacobian or Interaction Matrix. To control a n DOF robot with $n = 6$, where i represent the number of feature points detected, $i \geq 3$ is required. This work used $i=4$, four pairs of visual features corresponding to the top face's corners of a box where an image was painted and used as a template to guide the motion control of the robot. The package, our object of interest, is seen by the camera held by the robot in its end effector. The four visual reference points s_{ri} used for calculating the interaction matrix are estimated using a homography, a matrix that transforms visual features from the template image to the observation image. Dozens of SIFT feature matches built this matrix with an algorithm for parametric estimation that rejects wrong matches. Hence, the quality of the interaction matrix defined by its condition number improves directly with the quality of homography estimation, which grows concerning the number of feature matches detected, producing better visual parameters for building the interaction matrix.

Thus, the reference velocity for the internal control loop can be determined. To do this, we first solve for V_c , with which we obtain:

$$V_c = L_s^+ \dot{S} \quad (5)$$

Because if $n \neq 2i$, L_s^{-1} does not exist, then, due to $2i > n$, L_s^{-1} must be approximated by $L_s^+ = (L_s^T L_s)^{-1} L_s^T$ solving a least square problem. To guarantee that $2i$ is greater than n , we

propose using the path planning and visual servoing techniques in cooperation. The robot will meet a pose at the end of path planning, where the camera captures the object of interest. At this point, the sensor will retrieve enough image feature parameters to solve L_s^+ .

Finally, the objective of the visual controller is to minimize an error $e(t)$, which can be expressed as:

$$e(t) = S_d - S(t) \quad (6)$$

Substituting (6) in (5), we obtain:

$$V_c = L_s^+ \dot{e} \quad (7)$$

To ensure an exponential decrease of the error to zero, we have.

$$V_c = K L_s^+ e \quad (8)$$

where K is an adjustable gain matrix that allows determining the convergence velocity. Given that four points of interest were used, the interaction matrix has the form $L_s = [L_{s_1}]^T [L_{s_2}]^T [L_{s_3}]^T [L_{s_4}]^T]^T$, where $[L_{s_x}]$ is a pair of rows x of the matrix L_s .

Having defined the control law and the interaction matrix, it can be seen that this matrix depends on the coordinates of the points of interest on the image plane, the camera's focal length, and z , which is a normal distance from the image plane to the feature position in $\mathbb{R}^3$. Depth z must be obtained for complete knowledge of the interaction matrix.

A homography decomposition method was used for this work, described in Malis and Vargas (2007). Analytical development of this method returns four possible solutions, two of which can be discarded. Homography decomposition methods consist of obtaining perspective transformation components between two different views from the homography information resulting from two captures in different positions, which can be described as:

$$H \Rightarrow \{R, t, n\}$$

To obtain these components $\{R, t, n\}$ we first define a symmetric matrix S in terms of the homography transformation.

$$S_r = H H^T - I = (R + t n^T)(R^T + n t^T) - I \quad (9)$$

Then the following variables are introduced:

$$x = \frac{R^T t}{\|R^T t\|} = \frac{R^T t}{\|t\|} \quad (10)$$

$$y = \|R^T t\| n = \|t\| n \quad (11)$$

These, (10) (11), are substituted in (9), by analogous terms and using an auxiliary variable $\mathbf{z}$ we obtain three second order equations that solve as:

$$z_n = \alpha_n \pm \beta_n$$

where n varies from one to three, where:

$$\alpha_1 = \frac{s_{12}}{s_{22}}; \quad \beta_1 = \frac{\sqrt{s_{12}^2 - s_{11}s_{22}}}{s_{22}} = \frac{\sqrt{M_{S_{33}}}}{s_{22}} \quad (12)$$

$$\alpha_2 = \frac{s_{13}}{s_{33}}; \quad \beta_2 = \frac{\sqrt{s_{13}^2 - s_{11}s_{22}}}{s_{33}} = \frac{\sqrt{M_{S_{22}}}}{s_{33}} \quad (13)$$

$$\alpha_3 = \frac{s_{23}}{s_{22}}; \quad \beta_3 = \frac{\sqrt{s_{23}^2 - s_{22}s_{33}}}{s_{22}} = \frac{\sqrt{M_{S_{11}}}}{s_{22}} \quad (14)$$

With the introduction of the variable $\mathbf{z}$ defined in terms of $\mathbf{y}$ we have:

$$\mathbf{y} = \begin{bmatrix} z_1 \\ 1 \\ z_3 \end{bmatrix}; \quad y_2 \pm \sqrt{\omega} \quad (15)$$

where ω is an auxiliary variable that leads to a second order equation. The value of ω can be found by solving this equation. n can be determined by analogy of this equation with the definition of the auxiliary variable y . Four solutions are arrived at from the possible solutions to the homography decomposition problem (Malis and Vargas 2007), which have the form:

$$\mathbf{n}'_e = \begin{bmatrix} s_{h_{12}} \pm \sqrt{M_{S_{h_{33}}}} \\ s_{h_{22}} \\ s_{h_{23}} \pm \epsilon_{h_{13}} \sqrt{M_{S_{h_{11}}}} \end{bmatrix} \quad (16)$$

where $s_{h_{ij}}$ and $M_{S_{h_{ij}}}$ are the components and minors of the symmetrical matrix respectively and ϵ is $\text{sign}(M_{S_{h_{13}}})$, $e = a, b$. By dividing (16) by its norm, we obtain $\mathbf{n}_e$, with a similar procedure, the solution for translation vector $\mathbf{t}$ and rotation vector $\mathbf{R}$ is obtained. The translation vector is obtained from 10 and 11 expressed in terms of the components of the symmetrical matrix.

$$\mathbf{t}_a^*(s_{33}) = t_3 \begin{bmatrix} s_{13} - t_1 \\ s_{23} - t_2 \\ s_{33} \end{bmatrix} - t_4 \begin{bmatrix} s_{13} + t_1 \\ s_{23} + t_2 \\ s_{33} \end{bmatrix} \quad (17)$$

$$\mathbf{t}_b^*(s_{33}) = t_3 \begin{bmatrix} s_{13} + t_1 \\ s_{23} + t_2 \\ s_{33} \end{bmatrix} - t_4 \begin{bmatrix} s_{13} - t_1 \\ s_{23} - t_2 \\ s_{33} \end{bmatrix} \quad (18)$$

where

$$t_1 = \epsilon_{12} \sqrt{M_{S_{22}}} \quad t_3 = \frac{\|\mathbf{n}'_a(s_{33})\|}{2s_{33}} \\ t_2 = \sqrt{M_{S_{11}}} \quad t_4 = \frac{\|\mathbf{t}_e\|^2}{2\|\mathbf{n}'_a(s_{33})\|}$$

The rotation matrix can be obtained from the definition of the homography matrix (9) and (10).

$$\mathbf{R}_e = \mathbf{H} \left(\mathbf{I} - \frac{\mathbf{1}}{\mathbf{1} + \mathbf{x}_e^T \mathbf{y}_e} \right) \quad (19)$$

3 Results

3.1 PRM

To test and improve the Lazy PRM algorithm before its implementation in the 6 Degree of Freedom (DOF) space corresponding to the HP20D robot, we first had to start with a simplified version of the environment in 2 DOF.

The 2D version seeks paths made up of collision free straight segments in a rectangular environment with the presence of convex obstacles. The 6D version extends the roadmap idea to three-dimensional space and is applied in a simulated environment to the movement of a Motoman HP20D type industrial robot to take its end effector from a start point to an target point in the presence of objects built based on cubes within its workspace.

3.1.1 PRM 2D

Figure 3a shows a synthetic environment with static polygonal obstacles, showing a start point q_{init} represented as a green node on the graph and an end point q_{goal} node in magenta. An initial graph generated randomly over the entire space can be observed that connects the start and end nodes. Links with the starting node can be seen in green, and links to the ending node are in magenta. All other links appear in red.

Figure 3 shows an iterative process that seeks a collision free path from the start point q_{init} to the end point q_{goal} in 2 DOF. The sequence of Fig. 3a, b shows the initial roadmap, a feasible collision path, and the updated map without the nodes and edges of the obtained path found in C_{obs} . Likewise, the sequence of Fig. 3b, c shows the

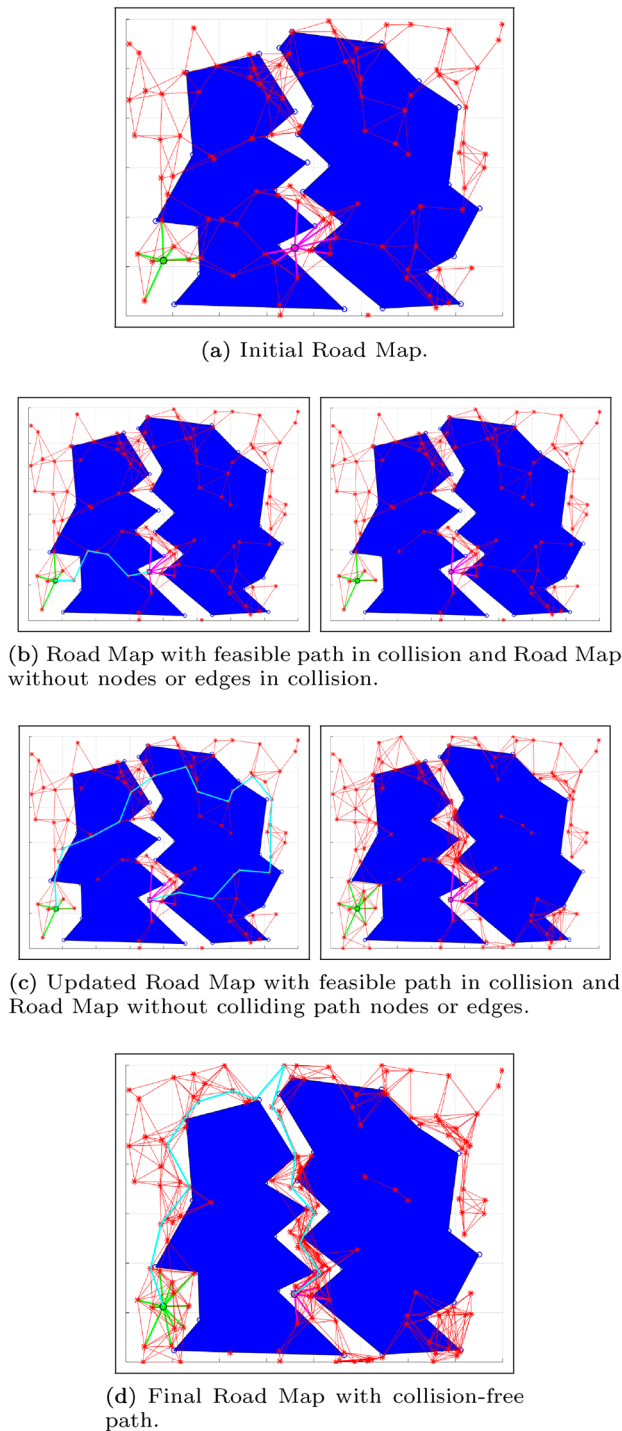


Fig. 3 2D PRM planning

refined map after the first iteration, a second path found with interference, and its removal from the roadmap, together with the addition of nodes due to the impossibility of finding a third feasible path. Finally, in Fig. 3d a new viable, collision-free path is found.

To measure the performance of the Lazy PRM 2D planner, an experiment was performed where the algorithm was run 10 times per combination of maximum number of connections [5, 10, 15] and number of start nodes [30, 50, 100, 200, 500]. The process recorded the algorithm's overall processing time and the number of nodes generated for each solution path Fig. 4. Likewise, the number of calls to the collision detector due to the nodes and edges of the feasible paths generated was recorded Fig. 5. Finally, Fig. 6 shows the time used by each of the main stages of the Lazy PRM algorithm: roadmap generation, local planner, and collision check.

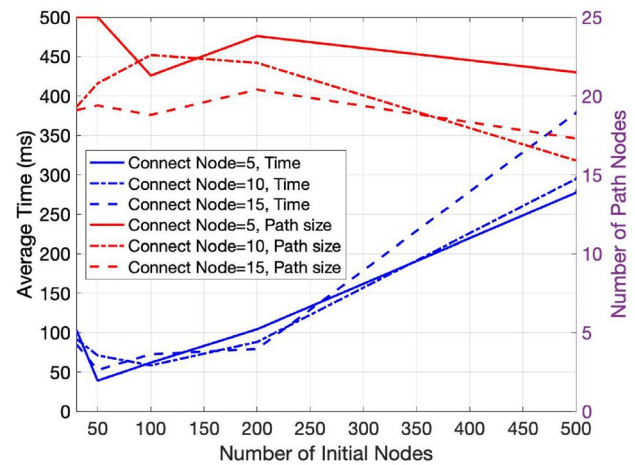


Fig. 4 Performance for Lazy PRM 2D

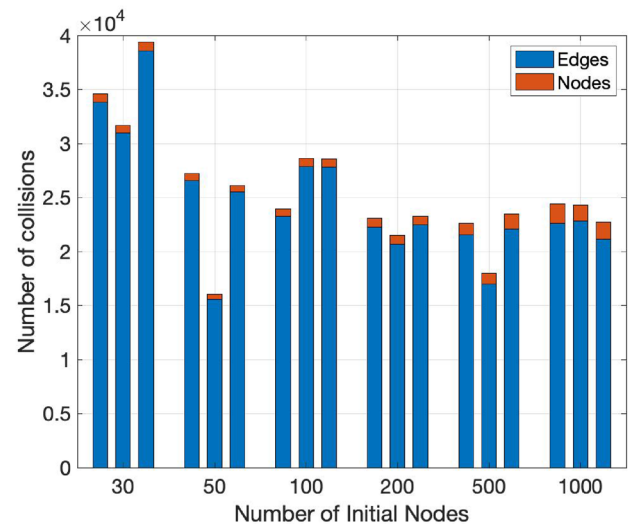


Fig. 5 Collision performance for Lazy PRM 2D

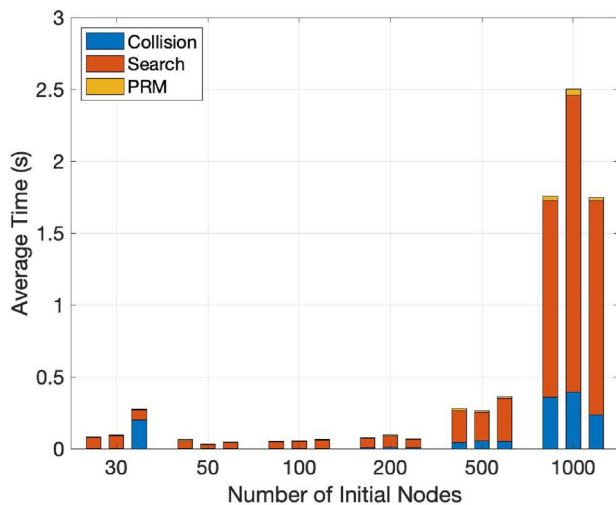


Fig. 6 Time performance for Lazy PRM 2D

3.1.2 PRM 3D

This section covers path planning for a Motoman HP20D 6 DOF industrial robot using a Lazy PRM algorithm. The robot and its workspace are simplified using prismatic links and primitives, which allow efficiently checking collision between the links of the robot themselves and the links and the objects located within the workspace (Rodriguez-Garavito et al. 2018). This object modeling is conservative and adjustable to a robot's real form. Figure 7 contains a visual description of the workspace showing the robot's starting position q_{init} and ending position q_{goal} .

Figure 8 shows the HP20D robot's path planning process for moving from q_{init} to q_{goal} . Figure 8a shows a feasible path in collision and Fig. 8b contains a representation of its associated roadmap. Figure 8c then shows a feasible, collision free path after eliminating the nodes and paths with collisions, planning new paths, and progressively refining the roadmap. The final path found is obtained using a linear

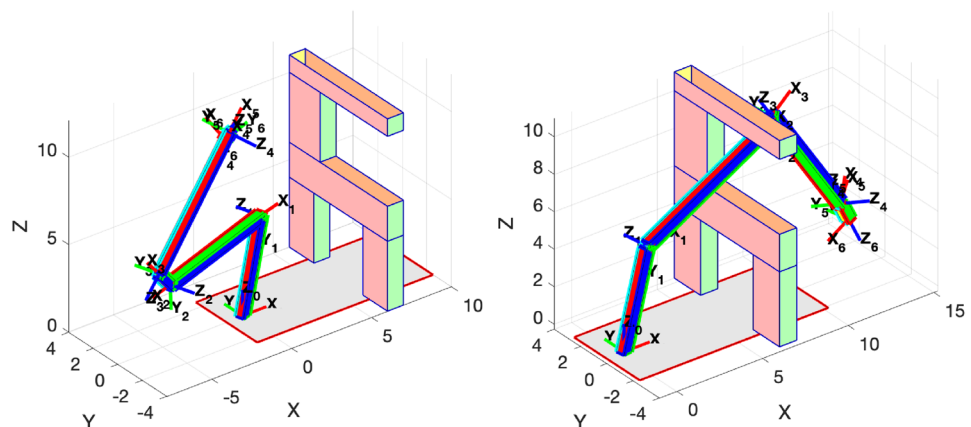
interpolator in the joint space. In other words, the path shown in Fig. 8d is the movement the robot's end effector actually follows, while Fig. 8c is an interpretation of a succession of points in the N_{XYZ} space that makes up the path found.

Finally, just like in the 2D space, the performance of the Lazy PRM algorithm is applied to a manipulator robot in a six-dimensional joint or workspace, where collision-free paths are expected to be found. The proposed experiment consists of running the Lazy PRM planner 40 times for different numbers of initial nodes [200, 300, 400, 500, 700, 1 000, 3000] on the roadmap, and then measure the average overall processing time for the algorithm and the number of nodes in the solution paths, Fig. 9. The number of calls to the collision detector to evaluate nodes and segments in feasible paths is also counted, Fig. 10. Finally, Fig. 11 records the processing time for each stage of the Lazy PRM 3D algorithm: local planner, collision detector, and manipulation of the graph representing the roadmap.

3.2 Visual control

To demonstrate the effectiveness of the visual controller technique, the ROS simulated model of the HP20D manipulator was used together with a box-type object of interest and a monocular camera with an eye-in-hand configuration. Then a reference image is taken, which is used to determine the error required to calculate the controller input. Since the control law requires an error, the position of a group of visual characteristics on the image plane at an instant i th in relation to the reference image is required. For this, the SIFT algorithm together with the RANSAC parametric estimator was used to determine a homography transformation matrix which enables relating visual characteristics between the reference image and the image captured at instant i th and thus finding these positions. The transformation matrix obtained can be used to

Fig. 7 Initial and final position of HP20D robot for path planning using Lazy PRM algorithm



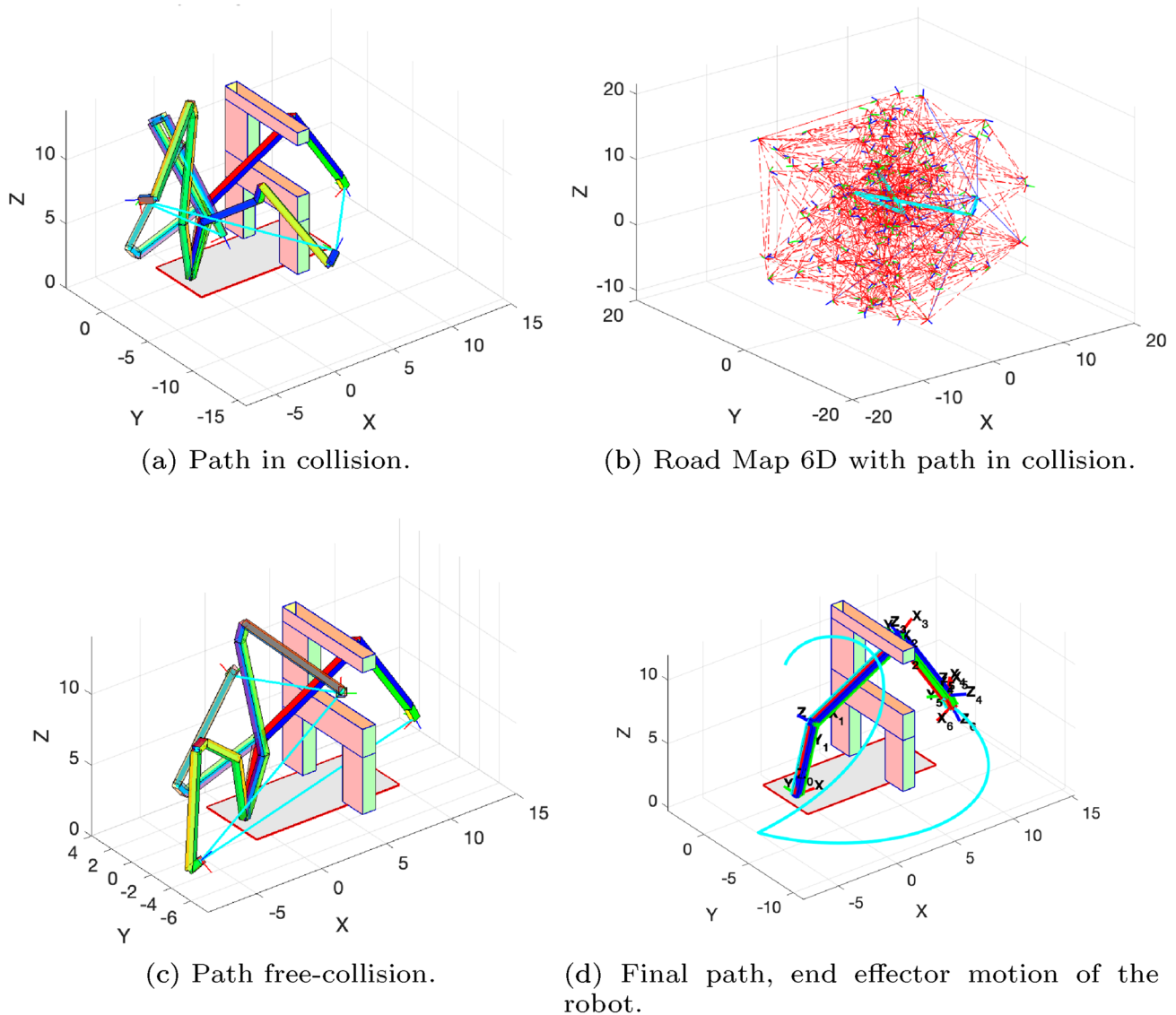


Fig. 8 3D PRM path planning

apply the homography decomposition method mentioned in Sect. 2.3 and calculate the spatial position of the visual characteristics of the object of interest.

Figure 12 shows the evolution of the robot's position and the camera's perspective in the experiment proposed at six instances of a visual control trajectory, Fig. 13, these images show a basic configuration with an object on a table, where Fig. 12a corresponds to the robot's starting position and Fig. 12f corresponds to the ending position once the system is stable.

During the execution of this experiment, velocities and errors in joints tend to zero, as we can see in Figs. 14 and 15.

4 Conclusions

This work proposes a flexible strategy to provide a robot with the ability to move freely within workspace, avoiding obstacles by mixing motion techniques like Path Planning and visual servoing. Both algorithms cooperate to guarantee that the robot, working under an eye-in-hand configuration, has the object of interest within field of view when starting its Visual Servoing phase. In this fashion, the visual features are available to calculate an accurate homography and interaction matrix with low uncertainty.

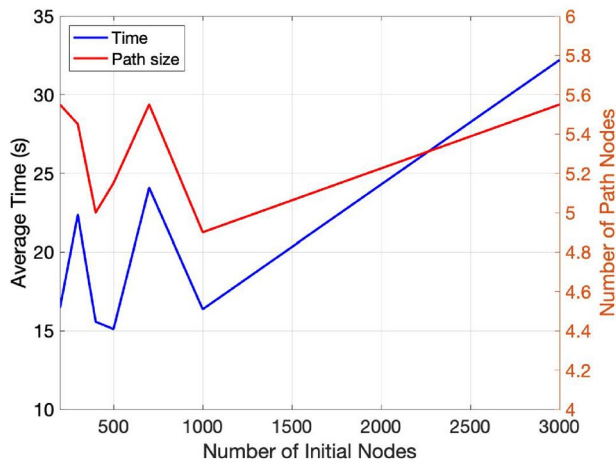


Fig. 9 Overall time and path length performance for Lazy PRM 3D

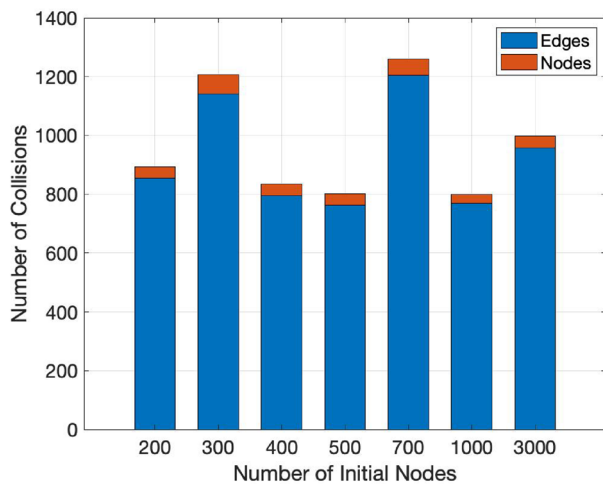


Fig. 10 Collision performance for Lazy PRM 3D

Each motion technique was probed independently. The Lazy PRM implementation produces some findings from the 2 and 6 DOF experiments: first, Figs. 4 and 9 show an elbow effect, where we could establish a suitable number of initial nodes related to the algorithm processing time around 100 for 2D planning and 500 for 3D planning. In addition, it is possible to verify that the maximum number of connections allowed for each node does not directly correlate with the processing time. Second, the Lazy PRM planner reaches a solution in a minimum of 50 ms with a path size of 23 nodes working in a 2 DOF environment, while in a 6 DOF environment it does so in 15 s with a path size of 5 nodes; what is reasonable for a planning query online in an environment subject to uncertainty. Finally, to conclude the analysis of the global Lazy PRM planner, we can observe that, depending on the application environment, algorithm

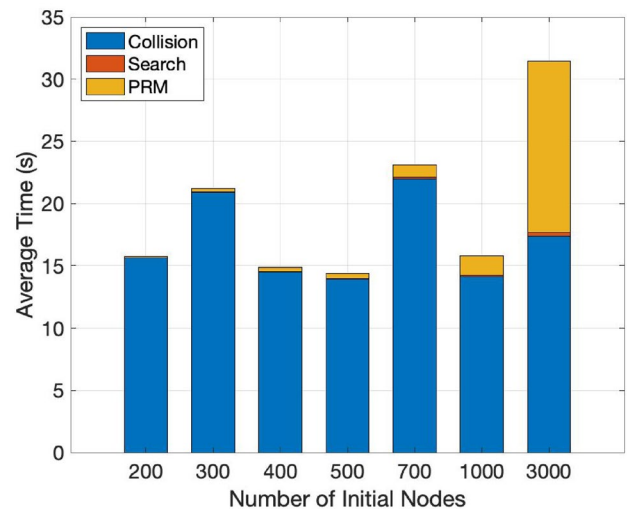


Fig. 11 Time process performance for Lazy PRM 3D

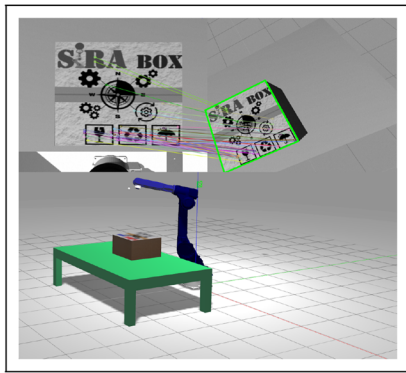
processing times change. In the 2 DOF case, if the number of nodes increases, the task that takes up the most run time is a local search for feasible paths. In contrast, in the 6 DOF, as the number of nodes on the connection graph increases, the costliest task is the collision detector. Its implementation needs to be efficient and based on a conservative simplification of the environment.

Implementing a camera, manipulator, and environment in ROS provide a suitable framework for testing cooperatively both visual servoing and planning, making possible its future migration over to a real industrial framework.

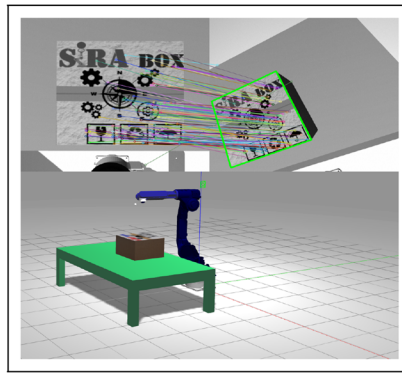
In Fig. 12 we can see the sequence of images for evolving camera captured frames and manipulator position in a visual servoing trajectory, which shows smooth robot position changes. However, the graph shown in Fig. 13 describes the trajectory of the four feature positions taken as a reference where we can see an overshoot at the end of the motion. In principle, this seems to be related to the time used by the visual control algorithm to calculate the output, but this overpass does not seem to be significant. Considering that the vision algorithm required to determine homography requires the most time for a control inference, a more efficient vision algorithm is necessary to produce smoother position and velocity changes and a shorter stabilization time.

The proposed technique allows for solving the loss of field of view (FOV) issue since the planner can run such that the object of interest will once again be visible.

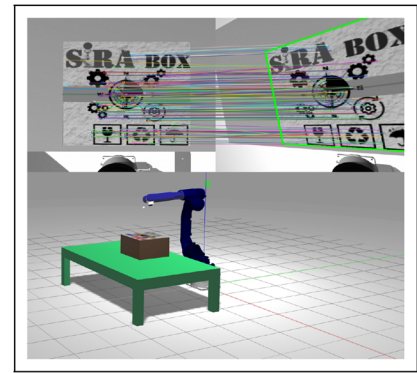
As future work, based on these motion techniques developed, visual servoing, and path planning for industrial robots, we propose transferring these algorithms to a real implementation based on an RBX1 robot integrated in ROS. In some previous works, we have developed a vision sensor for package-like objects; this will be the link to close the control loop and tackle several autonomous manipulation



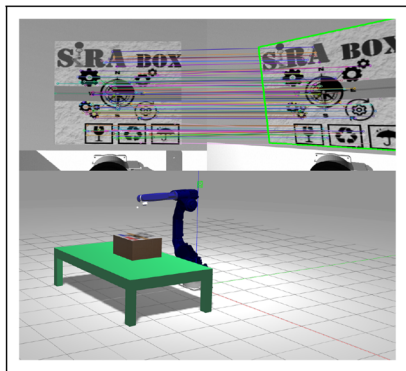
(a) Initial position of the robot. Position a in Fig. 13.



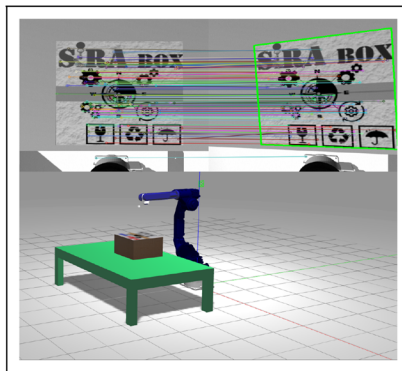
(b) Position b in Fig. 13.



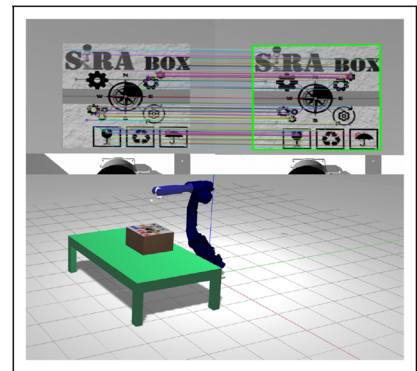
(c) Position c in Fig. 13.



(d) Position d in Fig. 13.



(e) Position e in Fig. 13.



(f) Final position of the robot. Position f in Fig. 13.

Fig. 12 Visual controller sequence. The upper images correspond to matching features at each instant. The lower images correspond to eye-in-hand configuration visual servo control

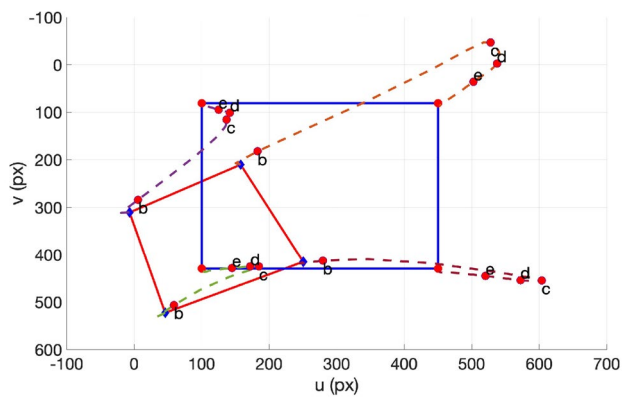


Fig. 13 Trajectory's features in different instants a–f

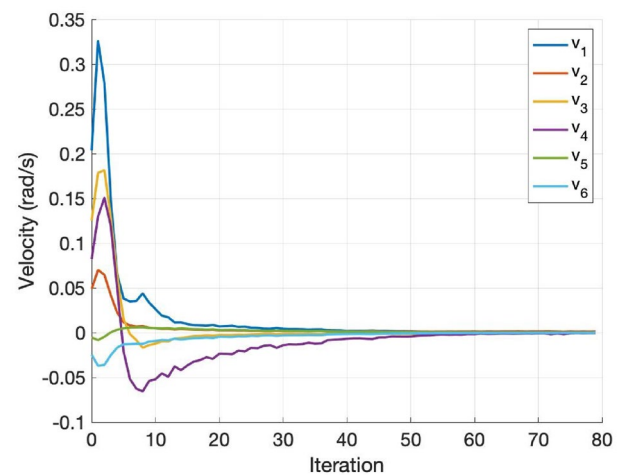


Fig. 14 Joints velocity

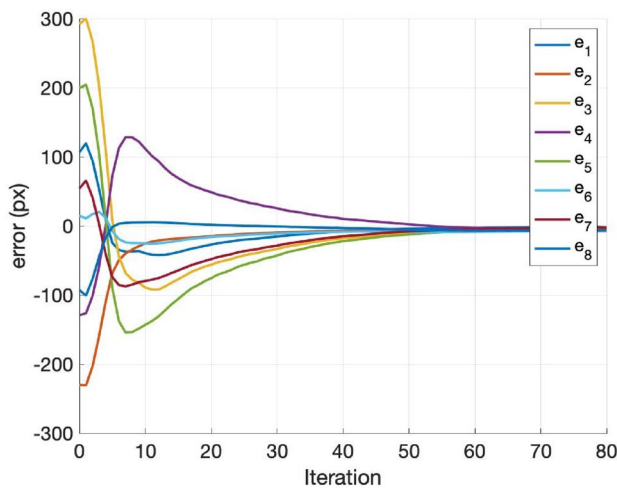


Fig. 15 Feature error

tasks on the road to robotics applications in the context of goods warehousing in the logistics sector.

Funding Open Access funding provided by Colombia Consortium.

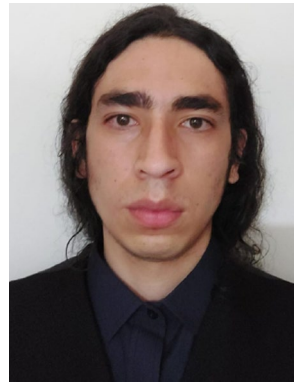
Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Al-Blawi, I., Siméon, T., Cortés, J.: Motion planning algorithms for molecular simulations: a survey. *Comput. Sci. Rev.* **6**(4), 125–143 (2012)
- Allen, P.K., Timcenko, A., Yoshimi, B., Michelman, P.: Automated tracking and grasping of a moving object with a robotic hand-eye system. *IEEE Trans. Robot. Autom.* **9**(2), 152–165 (1993)
- Azizian, M., Khoshnam, M., Najmaei, N., Patel, R.V.: Visual servoing in medical robotics: a survey. Part i: endoscopic and direct vision imaging—techniques and applications. *Int. J. Med. Robot. Comput. Assist. Surg.* **10**(3), 263–274 (2014)
- Barraquand, J., Latombe, J.-C.: Robot motion planning: a distributed representation approach. *Int. J. Robot. Res.* **10**(6), 628–649 (1991)
- Beom, H.R., Cho, H.S.: A sensor-based navigation for a mobile robot using fuzzy logic and reinforcement learning. *IEEE Trans. Syst. Man Cybern.* **25**(3), 464–477 (1995)
- Bohlin, R., Kavraki, L.E.: Path planning using lazy prm. In: *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings* (Cat. No. 00CH37065), vol. 1, pp. 521–528. IEEE (2000)
- Borenstein, J., Koren, Y., et al.: The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE Trans. Robot. Autom.* **7**(3), 278–288 (1991)
- Branicky, M.S., Curtiss, M.M., Levine, J.A., Morgan, S.B.: Rrts for nonlinear, discrete, and hybrid planning and control. In: *42nd IEEE International Conference on Decision and Control* (IEEE Cat. No. 03CH37475), vol. 1, pp. 657–663. IEEE (2003)
- Camacho M, G.A., Rodriguez G, C.H., Álvarez-Martínez, D., et al: Modelling the kinematic properties of an industrial manipulator in packing applications. In: *14th International Conference on Control and Automation (ICCA)*, pp. 1028–1033. IEEE (2018)
- Chaumette, F., Rives, P., Espiau, B.: Positioning of a robot with respect to an object, tracking it and estimating its velocity by visual servoing. In: *ICRA*, pp. 2248–2253 (1991)
- Chesi, G., Hung, Y.S.: Global path-planning for constrained and optimal visual servoing. *IEEE Trans. Robot.* **23**(5), 1050–1060 (2007)
- Deng, L., Janabi-Sharifi, F., Wilson, W.J.: Hybrid motion control and planning strategies for visual servoing. *IEEE Trans. Ind. Electron.* **52**(4), 1024–1040 (2005)
- Espiau, B., Chaumette, F., Rives, P.: A new approach to visual servoing in robotics. *IEEE Trans. Robot. Autom.* **8**(3), 313–326 (1992)
- Fang, Y., Dixon, W.E., Dawson, D.M., Chawda, P.: Homography-based visual servo regulation of mobile robots. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **35**(5), 1041–1050 (2005)
- Garcia, M.P., Montiel, O., Castillo, O., Sepulveda, R., Melin, P.: Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation. *Appl. Soft Comput.* **9**(3), 1102–1110 (2009)
- Gipson, B., Hsu, D., Kavraki, L.E., Latombe, J.-C.: Computational models of protein kinematics and dynamics: beyond simulation. *Annu. Rev. Anal. Chem.* **5**, 273–291 (2012)
- Gridseth, M., Hertkorn, K., Jagersand, M.: On visual servoing to improve performance of robotic grasping. In: *2015 12th Conference on Computer and Robot Vision*, pp. 245–252. IEEE (2015)
- Hu, Y., Yang, S.X., Xu, L.-Z., Meng, M.-H.: A knowledge based genetic algorithm for path planning in unstructured mobile robot environments. In: *2004 IEEE International Conference on Robotics and Biomimetics*, pp. 767–772. IEEE (2004)
- Kavraki, L.E., Svestka, P., Latombe, J.-C., Overmars, M.H.: Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. Autom.* **12**(4), 566–580 (1996)
- Kazemi, M., Gupta, K., Mehrandezh, M.: Path-planning for visual servoing: a review and issues. *Vis. Serv. Adv. Numer. LNCIS* **401**:189–207 (2010)
- Koenig, N., Howard, A.: Design and use paradigms for gazebo, an open-source multi-robot simulator. In: *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE Cat. No. 04CH37566), vol. 3, pp. 2149–2154. IEEE (2004)
- Koren, Y., Borenstein, J., et al.: Potential field methods and their inherent limitations for mobile robot navigation. In: *ICRA*, vol. 2, pp. 1398–1404 (1991)
- Kragic, D.: Visual servoing for manipulation: Robustness and integration issues. Ph.D. thesis, Royal Institute of Technology (2003)
- Krupa, A., Gangloff, J., Doignon, C., De Mathelin, M.F., Morel, G., Leroy, J., Soler, L., Marescaux, J.: Autonomous 3-d positioning of surgical instruments in robotized laparoscopic surgery using visual servoing. *IEEE Trans. Robot. Autom.* **19**(5), 842–853 (2003)
- Kuffner Jr, J.J.: Autonomous agents for real-time animation. Ph.D. thesis, Stanford University (2000)
- LaValle, S.: Rapidly-exploring random trees: a new tool for path planning. The annual research report (1998)

- Malis, E., Vargas, M.: Deeper understanding of the homography decomposition for vision-based control. Ph.D. thesis, INRIA (2007)
- Malis, E.: Improving vision-based control using efficient second-order minimization techniques. In: IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004, vol. 2, pp. 1843–1848. IEEE (2004)
- Malis, E., Chaumette, F., Boudet, S.: 2 1/2 d visual servoing. IEEE Trans. Robot. Autom. **15**(2), 238–250 (1999)
- Marey, M., Chaumette, F.: Analysis of classical and new visual servoing control laws. In: 2008 IEEE International Conference on Robotics and Automation, pp. 3244–3249 (2008). IEEE
- Nelson, B., Papanikolopoulos, N., Khosla, P.: Visual servoing for robotic assembly. In: Visual Servoing: Real-Time Control of Robot Manipulators Based on Visual Sensory Feedback, pp. 139–164. World Scientific (1993)
- Nomura, H., Naito, T.: Integrated visual servoing system to grasp industrial parts moving on conveyor by controlling 6dof arm. In: Smc 2000 Conference Proceedings. 2000 IEEE International Conference on Systems, Man and Cybernetics. 'cybernetics Evolving to Systems, Humans, Organizations, and Their Complex Interactions' (Cat. No. 0), vol. 3, pp. 1768–1775. IEEE (2000)
- Rodriguez-Garavito, C., Patiño-Forero, A.A., Camacho-Munoz, G.A.: Collision detector for industrial robot manipulators. In: The 13th International Conference on Soft Computing Models in Industrial and Environmental Applications, pp. 187–196. Springer (2018)
- Suetsugu, T., Matsuda, Y., Sugi, T., Goto, S., Egashira, N.: A visual supporting system for teleoperation of robot arm using visual servo control. In: 2014 Proceedings of the SICE Annual Conference (SICE), pp. 1847–1852. IEEE (2014)
- Thuijlot, B., Martinet, P., Cordesses, L., Gallice, J.: Position based visual servoing: keeping the object in the field of vision. In: Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292), vol. 2, pp. 1624–1629. IEEE (2002)
- Vahrenkamp, N., Wieland, S., Azad, P., Gonzalez, D., Asfour, T., Dillmann, R.: Visual servoing for humanoid grasping and manipulation tasks. In: Humanoids 2008-8th IEEE-RAS International Conference on Humanoid Robots, pp. 406–412. IEEE (2008)
- Wei, K., Ren, B.: A method on dynamic path planning for robotic manipulator autonomous obstacle avoidance based on an improved rrt algorithm. Sensors **18**(2), 571 (2018)
- Wei, G.-Q., Arbter, K., Hirzinger, G.: Real-time visual servoing for laparoscopic surgery. Controlling robot motion with color image segmentation. IEEE Eng. Med. Biol. Mag. **16**(1), 40–45 (1997)
- Wilson, W.J., Hulls, C.W., Bell, G.S.: Relative end-effector control using cartesian position based visual servoing. IEEE Trans. Robot. Autom. **12**(5), 684–696 (1996)
- Yang, K., Keat Gan, S., Sukkarieh, S.: A Gaussian process-based rrt planner for the exploration of an unknown and cluttered environment with a uav. Adv. Robot. **27**(6), 431–443 (2013)
- Zhou, L., Lin, T., Chen, S.-B.: Autonomous acquisition of seam coordinates for arc welding robot based on visual servoing. J. Intell. Robot. Syst. **47**(3), 239–255 (2006)

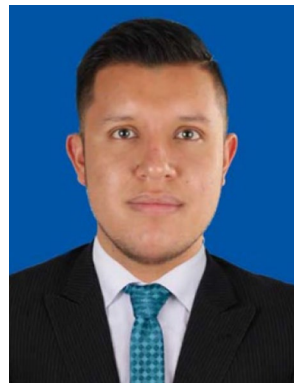
Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



R.I. Maldonado-Valencia received his B.E. degree in Automation Engineering from Universidad de La Salle in 2022. He has currently the role of software developer, using tools like java, spring, sql and server side tools. His research interests include control, soft computing, robotics and intelligent systems.



C. H. Rodriguez-Garavito is an associate professor at La Salle - Bogotá University. Electronic Engineer of Distrital University F.J.C. (2003), Master of Engineering - Industrial Automation, Nacional University of Colombia (2007), and Doctor of Electrical, Electronic and Automation Engineering from the Carlos III University of Madrid (2017). In 2018, He founded the research incubator group focused on Autonomous Robotics (SIRA) at La Salle University. His research activities include autonomous robotics, computer vision, and path planning for industrial robots.



C. A. Cruz-Perez was born in Bogotá-Colombia in the year 2022. He obtained his degree in Automation Engineering at Universidad de la Salle in Bogotá, Colombia, and was a member of the research incubator group focused on Autonomous Robotics (SIRA). He currently works on implementing WMS. His research interests are control theory applied to industry.



J. S. Hernandez-Navas was born in Bucaramanga-Colombia. He is an automation engineering student at La Salle University in Bogota-Colombia. He is currently studying his 10th semester, belongs to the research incubator group focused on Autonomous Robotics (SIRA) at La Salle University, and works in the hydrocarbon industry.



D. I. Zabala-Benavides was born in Bogotá-Colombia. He was a member of the research incubator group focused on Autonomous Robotics (SIRA). In 2021 he obtained his degree in Automation Engineering from Universidad de La Salle Bogotá, Colombia. He currently works as a Test Automation Engineer at Globant company.