A COMPARISON BETWEEN A SIMULATED AND A REAL MOBILE ROBOT PATH TRACKING APPLICATION USING V-REP


*Electrical Engineering Department - University of São Paulo at São Carlos
Avenida Trabalhador São-Carlense, 400, São Carlos, SP, Brazil. CEP 13566-590

†Mechanical Engineering Department - University of São Paulo at São Carlos
Avenida Trabalhador São-Carlense, 400, São Carlos, SP, Brazil. CEP 13566-590

Emails: kenny.caldas@usp.br, junior.anderson@usp.br, maurinho707@usp.br, vgrassi@usp.br, terra@sc.usp.br, siqueira@sc.usp.br

Abstract—The focus of this work is the validation of the Virtual Robot Experimentation Platform, V-REP, as a reliable and resourceful simulator for robotics applications. For this purpose, a simulated and a real mobile robot path tracking application are compared under the same conditions. A genetic path planning algorithm is used for generating the reference path for the mobile robot. A kinematic controller was implemented using Kanayama’s method. Simulated and experimental results are shown using a mobile robot e-Puck and the high precision Vicon camera system.

Keywords—Mobile robots, Robots control, Path planning, Simulation.

1 Introduction

Study of the robotic systems has become a versatile and open field in all education levels, from elementary school (Bers et al., 2002), (Church et al., 2010) to postgraduate courses (Mirheydar et al., 2009). In large part, this is due to advances in control systems, computer science, and communications systems, which have made robotics more accessible. Nowadays, there is a variety of robot shapes, sizes, prices, and functionalities that allow for a broad array of individuals, not necessarily researchers, to become interested in robotics. However, depending on the cost and complexity of the project, it is convenient to make use of simulation software/environment beforehand.

Simulators play an important role in most research fields, especially in robotic systems. According to Torres-Torriti et al. (2016), they became increasingly popular due to the progress in computer technology, which allowed the development of tools to improve some pioneer simulators, such as those developed by Corke (1996), and by Marhefka and Orin (1996). Currently, there are simulators widely used for different specific areas of robotics, such as for manipulators (Ferraguti et al., 2013) or mobile robotics (Michel, 2004). A few years ago, The Coppelia Robotics developed the Virtual Robot Experimentation Platform (V-REP®) (Rohmer et al., 2013b), (Peralta et al., 2016) . The simulator has a wide range of commercially known robots (manipulators and mobile), including the possibility of constructing any other robot, as well as the simulation environment. Additionally, it allows the versatility of communication with other softwares, making it possible to command and receive information externally of the robotic project.

This paper focuses on the comparison of a robotic application in a simulated environment in V-REP with a real world scenario. To this purpose, a kinematic control (Kanayama et al., 1990) of a mobile robot with differential traction is considered. Initially, the control problem is reproduced in simulated environment through the combination of V-REP® and MATLAB®. Afterwards, the same problem is applied in real time experiment using an e-Puck robot and a high precision localization system composed by four Vicon cameras. In both scenarios the reference path for the mobile robot is generated by a genetic algorithm approach, according to Kala (2014). This way, the main contribution of this paper is the validation of the results obtained in V-REP as a reliable and resourceful software for fast prototyping of robotics systems.

The sections of this paper are organized in the following way: in Section 2, the implementation of the genetic algorithm for path planning problems is discussed; in Section 3, the kinematic model of a differential mobile robot and the proposed controller are detailed. Section 6 covers the main results obtained in simulations and experiments and Section 7 approaches the discussion and conclusion of this work.

2 Path planning with Genetic Algorithm

The path planning method with obstacle avoidance uses a camera positioned above the path region where the mobile robot will move, in a way that it will cover all possible free and occupied spaces, as shown in Figure 1. Based on this configuration, it is possible to implement a genetic algorithm to find a collision-free path from a starting point to a target position.
The first step is to take a picture of the map with the positioned camera. This image is converted to a binary matrix by thresholding using the function `im2bw` from MATLAB®, where each pixel will have a value assigned: 0 for detected obstacles (black) and 1 for free spaces (white).

\[
\text{fitness} = d_0 \ast p + d_1 \ast p + \ldots + d_n \ast p, \quad (3)
\]

where \( p \) is a penalty multiplier for paths resulting in collisions and \( d_n \) is the distance between two set of points and is calculated by

\[
d = \sqrt{(x_{n+1} - x_n)^2 + (y_{n+1} - y_n)^2}. \quad (4)
\]

The next population generation is created by using two functions: selection and crossover. The selection function chooses the parents of the new population based on their fitness value. The crossover function is responsible to determine how the genetic algorithm creates the new individuals, called children, for the next generation. The functions used in this works are the stochastic uniform method and scattered crossover, respectively. A detailed description of these functions can be seen in the documentation of MATLAB® optimization toolbox 2015.

The stopping criteria adopted is a predefined number of generations, in order to reduce the computational time required for the algorithm genetic to run. In case none solution is found under a set of parameters, an output error message will be shown.

\section{Mobile robot kinematic}

A differential mobile robot is used in this work, meaning that each motor of the actuated wheels has an independent actuation. The geometry analysis of the robot is based on Figure 3. The axes \((X, Y)\) represent the inertial coordinates system; \((X_0, Y_0)\) the local coordinates system; \(d\) the distance between the center of the wheels axis \(P_0\).
and the center of mass \( P_c \); \( b \) the distance between the actuated wheel and the symmetry axis of the robot; \( r \) the wheel radius; \( \alpha \) the clockwise angle between the inertial reference axis \( x \) and the symmetry axis of the robot and \( \theta_r \) and \( \theta_l \) are the angular displacement for the right and left wheels.

![Diagram of Unicycle Model](attachment:image.png)

**Figure 3:** Unicycle model.

The kinematic controller proposed has to consider the nonholonomic and holonomic constraints. For a differential mobile robot, there are three kinematic constraints that must be observed during the development of the controller (Coelho and Nunes, 2003). The first constraint is given by

\[
\dot{y}_e \cos \alpha - \dot{x}_e \sin \alpha - d\dot{\alpha} = 0, \quad (5)
\]

where \((x_e, y_e)\) represents the center of mass \( P_c \) position in the inertial coordinates system.

The other two constraints are present on the rotation of the actuated wheels, since they can not slide

\[
\begin{align*}
\dot{x}_e \cos \alpha + \dot{y}_e \sin \alpha + b\dot{\alpha} - r\dot{\theta}_r &= 0 \quad (6a) \\
\dot{x}_e \cos \alpha + \dot{y}_e \sin \alpha - b\dot{\alpha} - r\dot{\theta}_l &= 0. \quad (6b)
\end{align*}
\]

Based on these equations, it is possible to rewrite the three kinematic constraints of the differential mobile robot model in the matrix form by using \( q_1 = [x_e \ y_e \ \alpha \ \theta_r \ \theta_l]^T \) as the generalized coordinate of the mobile robot

\[
R(q_1) = \begin{bmatrix}
-\sin \alpha & \cos \alpha & -d & 0 & 0 \\
-\cos \alpha & -\sin \alpha & -b & r & 0 \\
-\cos \alpha & -\sin \alpha & b & 0 & r
\end{bmatrix}. \quad (7)
\]

The matrix \( R(q_1) \) is full rank and can be expressed as \([R(q_1)] R_2\), to ensure that \( R(q_1) \) avoids singularity and that \( S(q_1) = [(R_1^{-1} q_1 R_2)^T \ I_3]^T \) is in the null space of \( R(q_1) \) (Siqueira et al., 2010). This way, \( S(q_1)^T \) becomes

\[
S(q_1)^T = \begin{bmatrix}
c(b \cos \alpha - d \sin \alpha) & c(b \cos \alpha + d \sin \alpha) \\
c(b \cos \alpha + d \sin \alpha) & c(b \cos \alpha - d \sin \alpha) \\
c & -c \\
1 & 0 \\
0 & 1
\end{bmatrix}, \quad (8)
\]

where \( c = \frac{r}{2b} \). The kinematic model can be expressed by

\[
\dot{q}_1(t) = S(q_1) \dot{q}_2(t) \quad (9)
\]

or

\[
\begin{align*}
\dot{x}_c &= c(b \cos \alpha - d \sin \alpha) \theta_d + c(b \cos \alpha + d \sin \alpha) \theta_e \\
\dot{y}_c &= c(b \cos \alpha + d \sin \alpha) \theta_d + c(b \cos \alpha - d \sin \alpha) \theta_e \\
\dot{\alpha} &= c(\theta_r - \theta_l). \quad (10)
\end{align*}
\]

The position vectors of the right and left wheels are defined as \( q_2 = [\theta_r \ \theta_l]^T \) and the angular velocities vectors as \( \dot{q}_2 = [\dot{\theta}_r \ \dot{\theta}_l]^T \).

The proposed kinematic controller is based on Kanayama et al. (1990). The controller uses the kinematic model of the mobile robot to generate the desired angular velocities for the right and left wheels, in order to follow the collision-free path references found by the genetic algorithm.

The mobile robot reference and current pose are defined by \( q_r = [x_r \ y_r \ \alpha_r]^T \) and \( q_c = [x_c \ y_c \ \alpha_c]^T \), respectively, where \( \alpha_r \) is given by

\[
\alpha_r = \tan^{-1}(y_r/x_r). \quad (11)
\]

This way, it is possible to define the error between the two states as \( q_e = [x_e \ y_e \ \alpha_e]^T \), where

\[
\begin{align*}
x_e &= \cos \alpha(x_r - x_c) + \sin \alpha(y_r - y_c) \\
y_e &= -\sin \alpha(x_r - x_c) + \cos \alpha(y_r - y_c) \quad (12)
\end{align*}
\]

\( \alpha_e = \alpha_r - \alpha_c \).

The desired linear and angular velocities, \( \nu^d \) and \( \omega^d \), are calculated by the kinematic controller by using the error between the current and desired pose based on the given references by

\[
\begin{align*}
\nu_r &= \sqrt{(x_r)^2 + (y_r)^2} \\
\omega_r &= \alpha_r. \quad (13)
\end{align*}
\]

Based on these equations, it is possible to define the kinematic controller by

\[
\begin{align*}
\nu_d &= \nu_r \cos(\alpha_e) + K_x x_e \\
w_d &= w_r + \nu_r (K_\nu y_e + K_\alpha \sin(\alpha_e)), \quad (14)
\end{align*}
\]

where \( K_x, K_\nu \) e \( K_\alpha \) are the controller gains.

In order to correctly update the desired values for the left and right wheels, the angular velocities generated by the controller must consider the radius and distance between the wheels. Therefore,
the relation between the angular velocities and the robot physical parameters is given by

$$\dot{q}_d^l = \begin{bmatrix} \dot{\theta}_d^r \\ \dot{\theta}_d^l \end{bmatrix} = \begin{bmatrix} 1/r \\ 1/r \end{bmatrix} \begin{bmatrix} b/r \\ -b/r \end{bmatrix} \begin{bmatrix} v_d \\ \omega_d \end{bmatrix}, \tag{15}$$

where $\dot{\theta}_d^r$ e $\dot{\theta}_d^l$ are the angular velocities for the right and left wheel that will be updated on the robot.

4 Simulation setup

The implementation of the genetic path planning algorithm and the kinematic controller were made using the V-REP® simulator. V-REP® is a platform for virtual experiments with a wide variety of robots in its library, including the e-Puck, the differential mobile robot used in this work. This simulation framework allows a rapid algorithm implementation, system verification and quick prototyping, which can greatly reduce the deployment complexity of robotics systems (Rohmer et al., 2013a).

The first step is to design the simulation environment where the mobile robot will be moving. The map is represented in Figure 4. It has an area of $2 \times 2m^2$ divided in four rooms and a main corridor. Once the map is defined, the collision-free path is obtained by implementing the genetic algorithm. For the simulation, the origin coordinates points were chosen as $x_0 = 0.20m, y_0 = 0.25m$ and goal as $x_g = 1.50m, y_g = 1.80m$.

Figure 4: Simulation environment in V-REP®.

The path generated from the genetic algorithm implementation can be seen in Figure 5. The function $ga$ of MATLAB® was used for this application. The resulting set of points of the path were filtered, in order to eliminate the exceeding point and obtain a smooth path. In order to consider the robot dimension into the path planning problem, the dimensions of the walls on the map were incremented by the radius of the robot body. This way, even if the generated path was very close to one of the walls, the mobile robot would be in a safe distance.

5 Experiment setup

The experiment performed to compare the results obtained from the simulation in V-REP® uses the e-Puck in an environment monitored by high precision Vicon system.

The Vicon cameras are positioned on the superior corners of the experiment environment. The tracking of the object inside a workspace is obtained by reflective markers placed on the mobile robot, and its identification is done through the distance between them. Figure 6 shows the Vicon system and the reflective marker used in this work. The kinematic controller in MATLAB® updates the control inputs and transmits them to the e-Puck through Bluetooth. The main characteristics of the Vicon cameras are presented in Table 1 and a diagram of the experiment setup can be seen in Figure 7. Although the Vicon cameras have a high capture frame rate, the Bluetooth communication between the computer and the e-Puck limits the sampling period to 0.14 seconds.

Figure 5: Collision-free path generated by the genetic algorithm.

Figure 6: (a) Vicon camera model T-series T40S. (b) Reflective marker.

Once the experiment setup was defined, the same map designed in the simulation on V-REP®
Table 1: Vicon system technical characteristics

<table>
<thead>
<tr>
<th>Camera</th>
<th>T40S</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resolution</td>
<td>4Mp (2336 × 1728)</td>
</tr>
<tr>
<td>Camera frame rates</td>
<td>30 – 2000</td>
</tr>
<tr>
<td>Sensor</td>
<td>Vicon Vegas S-4</td>
</tr>
</tbody>
</table>

Figure 7: Experiment setup diagram.

was built, using the same dimensions values. Figure 8 shows the map used in the experiment.

Figure 8: Real map used in the experiment.

6 Results and discussion

The results obtained from the simulation and the experiment are shown in Figure 9. The blue and red curve represent the simulated and real trajectories, respectively. The circles are the reference path generated by the genetic algorithm. It is possible to see that both results are very similar, which shows that using the V-REP® as a prototyping tool is a good solution, since it is very accurate with the real system. The gains used in the kinematic controller are: $k_x = 0.1$, $k_y = 1000$ and $k_\alpha = 10$, based on Equation (14).

Figure 10 shows the orientation angle and Figure 11 the lateral error for simulated and real scenarios. The distance traveled by the e-Puck is used as abscissa for these graphs. Both results showed that under the same conditions, V-REP® could reproduce very closely the e-Puck behavior during its movement.

7 Conclusion

This paper presented a comparison between a simulated and a real robotic system. A genetic algorithm was used to perform a collision-free path planning of a mobile robot in a structure environment. This was done by positioning an overhead camera that could capture a picture of the whole area where the robot will move. In order to do this, a map in V-REP® was designed to replicate a real world scenario and the genetic algorithm was able to find a feasible path.

The control strategy was based on the kinematic model of the e-Puck, as shown in the work of Kanayama et al. (1990), which generates the desired angular velocities required to keep the mobile robot in the path by using a set of references for the control inputs calculation.

The results were obtained initially in the simulation environment in V-REP® and later in a real world experiment, by using the Vicon cameras for localization. The comparison of the results obtained was satisfactory, since the simulation could reproduce very accurately the behavior of the e-Puck in the real system, which validates the use
of V-REP® as a resourceful tool for quick prototyping of robotics systems.

Future works involve the comparison of simulations in different environments and robotics systems, as an alternative to reduce the time taken to develop real world robotics applications.

8 Acknowledgments

This work is supported by National Council for Scientific and Technological Development (CNPq) under grants 142080/2016-0 and by Coordination for the Improvement of Higher Education Personnel (CAPES) under grants 1584685.

References


