Stereovision Based Leader/Follower Tracking control for Nonholonomic Mobile Robots

Zeghlache SAMIR and Mohand Saïd DJOUADI

Laboratoire robotique & productique,

zeghlache_samir@yahoo.fr, mxdjouadi@gmail.com

Abstract—This work discusses a control strategies to solve the target tracking problem for nonholonomic mobile robots, in our experiments we have considered two robots, a leader and a follower. The follower robot tracks a target (leader) moving along an unknown trajectory. It uses only stereovision to establish its both relative position and orientation to the target, so as to maintain a specified and a given distance between the two robots. Experimental results are included and discussed to show the performances of the proposed vision-based tracking control system.

Keywords—mobile robot, stereovision, vision based control, Lyapunov theory

I. INTRODUCTION

In the recent years, efforts have been made to give autonomy to mobile robots by using different sensors to collect information from the surroundings and react to the changes of its immediate environment. Computer vision is one of the most popular perception sensors employed for autonomous robots. In many task such as surveillance and grasping, visual path tracking, visual tracking of a moving object. The application of vision based tracking control design for robotic application has been an active area of robotic research, the papers [1,2] present study of visual servoing approaches for tracking control of non-holonomic mobile robots, in [3] the authors present study of visual navigation mobile robots, this work subdivided into the visual navigation in indoor environment and the visual navigation in outdoor environment. A method and illustration of vision data extraction is also presented, in order to allow a robot to operate in an unknown and dynamic environment. In [4] the authors have presented a vision-based scheme for driving a non-holonomic mobile robot to intercept a moving target, on the lower level, the pan-tilt platform which carries the onboard camera is controlled so as to keep the target at the centre of the image plane; on the higher level, the robot operates under the assumption that the camera system achieves perfect tracking. In particular, the relative position of the ball is retrieved from the pan-tilt angles through simple geometry, and used to compute a control law driving the robot to the target. Various possible choices are discussed for the high-level robot controller. In [5] a robust visual tracking controller is proposed for tracking control of a mobile robot in image plan, this work based on the proposed error-state model, the visual tracking control problem is transformed into the stability problem. The robust control law is then proposed to guarantee that the visual tracking system satisfies the necessary stability condition based on LYAPUNOV theory. [6] Describes the position control of autonomous mobile robot using combination of kalman filter and fuzzy logic technique. Both techniques have been used to fuse information from internal and external sensors to navigate the mobile robot in unknown environment. An obstacle avoidance algorithm using stereovision technique has been implemented for obstacle detection. [7] Presents a strategy for a non-holonomic mobile robot to autonomously follow a target based on vision information from an onboard pan camera unit. Homography based techniques are used to obtain relative position and orientation information from the monocular camera images. The proposed kinematic controller, based on the Lyapunov method, achieves uniform ultimately bounded tracking. In [8] and [9] the authors have proposed the state feedback control using the fuzzy logic controller to realize a leader and follower mobile robot based on laser and infrared sensors.

The monocular vision based tracking control suffers to obtain the 3D target position. However, in our work the stereovision system has been used to solve this problem, we can at each instant have a pair of images, from two geometrical defined cameras, that allow us to have four image coordinates. The triangulation equation is used to estimate the relative 3D position (tracker-target). In this paper we aim to ensure an accurate tracking in case of a leader/follower mobile robot scheme, by experimenting different stereovision based control strategies. First we will describe the kinematical model of a mobile robot (Section II), and the camera model (section III), section IV present the image processing, in section V we present a method to detect a target using the Hough transform, in section VI we present the depth estimation using the triangulation equations and the parameters of the cameras, section VI presents the mobile robot’s visual control development. Finally we arrive to the conclusion of the whole work.

II. MOBILE ROBOT MODEL

In this work is considered the unicycle mobile robot, the navigation is controlled by the speed on either side of the robot. This kind of robot has non-holonomic constraints, which should be considered during path planning. The kinematical scheme of a mobile robot can be depicted as in Fig. 1, where \( v \) is the velocity of the robot, \( v_l \) is the velocity of the left wheel, \( v_r \) is the velocity of the right wheel, \( r \) is the
radius of each wheel, \( l \) is the distance between the first two wheels, \( x \) and \( y \) are the position of the mobile robot, and \( \phi \) is the orientation of the robot.

This type of robot can be described by the following kinematics equations:

\[
\begin{align*}
\dot{x} &= v \cos \phi \\
\dot{y} &= v \sin \phi \\
\dot{\phi} &= \omega
\end{align*}
\]  
(1)

The non-holonomic restriction for model (1) is

\[
\dot{y} \cos \phi - \dot{x} \sin \phi = 0
\]  
(2)

According to the motion principle of rigid body kinematics, the motion of a mobile robot can be described using equations (1) and (2), where \( \omega_l \) and \( \omega_r \) are the angular velocities of the left and right wheels respectively, and \( \omega \) is the angular velocity.

The left and a right velocity of robot:

\[
\begin{align*}
v_x &= r \omega_l \\
v_y &= r \omega_r \\
\omega &= \frac{v_y - v_x}{l} \\
v &= \frac{v_x + v_y}{2}
\end{align*}
\]  
(3)

Combining (2) with (3) we can obtain:

\[
\begin{align*}
\omega &= \frac{r}{l} (\omega_r - \omega_l) \\
v &= \frac{r}{2} (\omega_r - \omega_l)
\end{align*}
\]  
(5)

### III. CAMERA MODEL

Calibration is a heavily worked on area in vision because it is necessary to estimate 3D distance information contained in an image. It allows to model mathematically the relationship between the 3D coordinates of an object in a scene and its 2D coordinates in the image [8].

The parameters of the camera are classified in two categories, **internal parameters** which define the properties of the geometrical optics and the **external parameters** which define position and orientation of the camera. More specifically, the camera calibration consists of determining the intrinsic parameters and the extrinsic parameters [9]. The model of the camera is presented in fig. 2.

**A. Intrinsic parameters**

Intrinsic parameters of the camera define the scale factors and the image centre.

\[
I_c = \begin{pmatrix}
\alpha_x & 0 & u_0 & 0 \\
0 & \alpha_y & v_0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & u_0
\end{pmatrix}
\]  
(6)

\( K_x, K_y \) represent the horizontal and vertical scale factor, \( f \) represent the focal length and \( u_x, v_0 \) represent the image centre.

**B. Extrinsic parameters**

Which define the transformation from the world to the camera frame given by the matrix \( A \).

\[
A = \begin{pmatrix}
r_{11} & r_{12} & r_{13} & t_x \\
r_{21} & r_{22} & r_{23} & t_y \\
r_{31} & r_{32} & r_{33} & t_z \\
0 & 0 & 0 & 1
\end{pmatrix} = \begin{pmatrix}
R & T
\end{pmatrix}
\]  
(7)

The matrix \( A \) is a combination of rotation matrix \( R \) and translation matrix \( T \) from to the world frame to the camera frame.

The transformation from the world to the image frame is given by the matrix \( M \).

\[
M = I_c A
\]  
(8)

We can write:

\[
\begin{pmatrix}
su \\
sv \\
s
\end{pmatrix} = \begin{pmatrix}
m_{11} & m_{12} & m_{13} & m_{14} \\
m_{21} & m_{22} & m_{23} & m_{24} \\
m_{31} & m_{32} & m_{33} & m_{34}
\end{pmatrix} \begin{pmatrix}
X \\
Y \\
Z
\end{pmatrix}
\]  
(9)
In this equation, X, Y, Z are the coordinates of a point B in the world frame.

IV. IMAGE PROCESSING

There exist a lot of image-processing algorithms extracting a map of the environment from the data provided by a camera. But the implementation of such sophisticated algorithms is quite complex [10].

In our application, the bumblebee 2 camera is used for perception. There resolution is 320x240 pixels, in this work the image processing procedure comprises four steps:

- **Step1**: acquisition of the two images.
- **Step2**: YCbCr image transform, when the YCbCr is a family of color spaces used in video systems. Y is the luminance component and Cb and Cr the chrominance components. YCbCr is sometimes abbreviated to YCC. YCbCr signals are created from the corresponding gamma adjusted RGB source using two defined constants \( K_b \) and \( K_r \) as follows [11]:

\[
Y = k_r \cdot R' + (1 - k_r - k_b) \cdot G' + k_b \cdot B' \\
Cb = \frac{0.5 \cdot (B' - Y)}{(1 - k_b)} \\
Cr = \frac{0.5 \cdot (R' - Y)}{(1 - k_r)}
\]

Where \( k_b \) and \( k_r \) are derived from the definition of the RGB space. \( R' \), \( G' \) and \( B' \) are assumed to be nonlinear and to nominally range from 0 to 1, with 0 representing the minimum intensity and 1 the maximum.

Mathematical morphology is a useful tool for image segmentation and processing. Due to the complexity of color-scale morphology, we transform color image into binary image by subsampling techniques and threshold-based segmentation. Then operators in the area of binary morphology including dilation, erosion, closing and opening are used for data pre-processing [12]. Then step 3 and 4 given by follow:

Dilation operation is defined as:

\[
X \oplus S = \{x / S \cap x \neq \emptyset \} \text{ with } X = \{x / x = 1\}
\]

Erosion operation is defined as:

\[
X \ominus S = \{x / S \cap x \neq \emptyset \} \text{ with } X' = \{x / x = 0\}
\]

The closing is a dilatation followed by erosion using the same structuring element for both operations.

Dilation operation is defined as:

\[
Y \circ S = (Y \ominus S) \oplus S
\]

An opening operation is defined as

And the opening is erosion followed by dilatation

- **Step3**: Cr image plan.
- **Step4**: binairisation and morphology operators [12].
V. DETECTION THE TARGET USING HOUGH TRANSFORM

The first step correspondence algorithm is to find the circles in both images. These circles are extracted from binary edges images using the hough transform. It is a well-know method for detection of parametric curves in binary images and it was recognized as an important means of searching for objects features in binary images.

The hough transform is a robust and effective method for finding circles a set of 2D points. It is based on a transformation from the \((x, y)\) plane (called Cartesian plane) to the \((r, \theta, a)\) plane (called houghf plane). A circle is defined in the Cartesian plan by the following equation:

\[
(x-a)^2 + (y-b)^2 = r^2
\]

\((x, y)\): The coordinates of the circles point in the Cartesian frame.
\((r)\): The radius points.
\((a, b)\): The coordinate of the centre circle.

![Fig.8 The circle polar parameter](image)

The HT uses the polar parameters of the circle (Fig. 8). A circle is defined by the following equation:

\[
\begin{cases}
    a = \rho \cos(\theta) \\
    b = \rho \sin(\theta) \\
    r = \sqrt{(x - \rho \cos(\theta))^2 + (y - \rho \sin(\theta))^2}
\end{cases}
\]

Where \(\rho\) is the distance between the \(xy\) plane origin and the centre circle and \(\theta\) is the angle between the \(x\) axis and \(\rho\). To detect a circle, we must determine three parameters \((a, b, r)\), which complicate to use the hough transform in 2D. To solve this problem, we should be pass by two step, first is to find all centre circle with different radius, than is to find the radius of the circle to be detect.

We consider the hough space \(H(a, b, r)\) who is in this case a space in 3D, each point of coordinate \((x, y, \theta)\) in the \(xy\) plane corresponds the cone in the space \(H(a, b, r)\) (fig. 9), and for a fixed radius it corresponds a circle in the \((a, b)\) plane. the idea of the algorithm is to compute \(a\) and \(b\) (the coordinate of the centre circle) for each radius \(r\), then we trace this circle in the hough space corresponding to the point of coordinate \((x, y, \theta)\) in the \(xy\) plane. When all the circles are cut in the same point, then the good radius and the coordinate \((a, b)\) of the point which correspond to the circle was found.

![Fig.9 Hough space \(H(a, b, r)\)](image)

Let \(\theta_i\) be the quantification step of \(\theta\) dimension and \(\rho_j\) the quantification step of \(\rho\) dimension, we compute for every discrete value of \(i\) \(\theta\) and \(j\) \(\rho\) its corresponding points \((a_i, b_j, r)\), the quantification of the plan \((\theta, \rho, r)\) returns to quantifier the interval \(0 < \theta < \frac{\pi}{2}\) for the dimension of \(\theta\), and the interval \(0 < \rho < \rho_{\text{max}}\) for the dimension of \(\rho\) whit \(\rho_{\text{max}}\) is the diagonal of the image, and the interval \(0 < r < r_{\text{max}}\) for the dimension of \(r\) with \(r_{\text{max}} = \frac{\rho_{\text{max}}}{10}\), the discrete value of \(\theta, \rho,\) and \(r\) are given by:

\[
\begin{align*}
    \theta_i &= t_i \theta, \quad 0 < t_i < \eta_\theta \\
    \rho_j &= t_j \rho, \quad 0 < t_j < \eta_\rho \\
    r_k &= t_k r, \quad 0 < t_k < \eta_r
\end{align*}
\]

With \(\eta_\theta, \eta_\rho, \eta_r\) are the numbers of the discrete values in the intervals \(\rho, \theta\) and \(r\).

\[
\eta_\theta = \frac{\pi}{2 \theta}, \quad \eta_\rho = \frac{\rho_{\text{max}}}{\rho_\theta}, \quad \eta_r = \frac{r_{\text{max}}}{\rho_\rho}
\]

For every point of coordinates \((x, y)\) in the image, following the equation:

\[
\begin{align*}
    a_i &= \rho \cos(\theta) \\
    b_i &= \rho \sin(\theta) \\
    r_i &= \sqrt{(x - \rho \cos(\theta))^2 + (y - \rho \sin(\theta))^2}
\end{align*}
\]

Fig.7 Application the opening followed the closing operation

Right image

Left image
Then, we increase by value 1, the whole cell \((a_j, b_j, r_j)\) of the accumulator table. This cell will be increased every time an edge point lies on the circle whose the polar parameters are \((a_j, b_j, r_j)\). Initially, all the cells values of the accumulator table are set to zero.

The choice of the \((a, b, r)\) parameters space quantification must carry the three essential goals that are:

- good detection and precision;
- less memory storage of accumulators;
- fast implementation.

Then, we seeks the maximum of the accumulator which correspond to the circle in the edge image. By using the parameters corresponding to the maximum of the accumulator, and we compute the radius and the coordinates of the circle centre.

Let us note, that the hough transform is robust for the parametric form detection such as the circles, the lines ……, the fig.12 and fig.13 show that in spite of degradation the edge of the target and the presence of the noises, we have got a good detection of the target.

\[
\begin{align*}
    x_i &= m_{i_2}X + m_{i_3}Y + m_{i_5}Z + m_{i_6} \\
    y_i &= m_{i_2}X + m_{i_3}Y + m_{i_5}Z + m_{i_6} \\
    x_i &= m_{i_2}X + m_{i_3}Y + m_{i_5}Z + m_{i_6} \\
    y_i &= m_{i_2}X + m_{i_3}Y + m_{i_5}Z + m_{i_6}
\end{align*}
\]

The points \(P\) and \(\tilde{P}\) are the coordinates of the target centre calculated by hough transform.

VI. DEPTH ESTIMATION

Depth calculates from a pair of stereoscopic images suppose the matched corresponding points between the left and right images.

If we places in the case of the three dimensional rebuilding, and if the point \(P(x, y)\) of the left image at summer put in correspondence with the point \(\tilde{P}(x, y)\) of the right image, using the equation (8) we have:

\[
\begin{align*}
    x_i &= m_{i_2}X + m_{i_3}Y + m_{i_5}Z + m_{i_6} \\
    y_i &= m_{i_2}X + m_{i_3}Y + m_{i_5}Z + m_{i_6} \\
    x_i &= m_{i_2}X + m_{i_3}Y + m_{i_5}Z + m_{i_6} \\
    y_i &= m_{i_2}X + m_{i_3}Y + m_{i_5}Z + m_{i_6}
\end{align*}
\]

Fig. 12. Robustness of the hough transform in spite the degradation of the edge

Fig. 13. Robustness of the hough transform in the presence of noise
Fig.14. geometry of the stereoscopic camera

The coordinate X, Y and Z of the point P rebuilt in the calibration frame, are calculated by solving a linear system of four equations. The equation (20) can be written by:

\[
\begin{pmatrix}
m_1 - x_l & m_2 - x_l & m_3 - x_l & m_4 - x_l \\
m_1 - y_l & m_2 - y_l & m_3 - y_l & m_4 - y_l \\
m_1 - x_r & m_2 - x_r & m_3 - x_r & m_4 - x_r \\
m_1 - y_r & m_2 - y_r & m_3 - y_r & m_4 - y_r
\end{pmatrix}
\begin{pmatrix}
x \\
y \\
z
\end{pmatrix}
= \begin{pmatrix}
-m_1 - x_l \\
-m_1 - y_l \\
-m_1 - x_r \\
-m_1 - y_r
\end{pmatrix}
\]

The system of equation (21) can be rewritten in the form:

\[ E \, P = W \]  

We can solve the equation (22) using the least squares method:

\[ P = (E^T \, E)^{-1} \, E^T \, W \]  

We can also rebuild the point P in the left camera frame using the intrinsic parameters matrix of the left and right camera \( I_{l}, I_{r} \). The coordinates \( X_{l}, Y_{l} \) and \( Z_{l} \) of the point P are given by [13]:

\[
\begin{align*}
Z_{l} &= \frac{b}{y_{1} - y_{2}} \\
X_{l} &= x_{1} \, Z_{l} \\
Y_{l} &= y_{1} \, Z_{l}
\end{align*}
\]

With:

\[
\begin{pmatrix}
x_{1} \\
y_{1}
\end{pmatrix} = I_{r}^{-1} \begin{pmatrix}
x_{1} \\
y_{1}
\end{pmatrix} \quad \text{and} \quad \begin{pmatrix}
x_{1} \\
y_{1}
\end{pmatrix} = I_{l} \begin{pmatrix}
x_{1} \\
y_{1}
\end{pmatrix}
\]

By making a translation along the Y by \( \frac{b}{2} \), the 3D coordinate of the target centre can be calculated in the frame located between the two camera left and right as is shown in fig.14. Thus the equation (25) becomes:

\[
\begin{align*}
Z &= \frac{b}{y_{1} - y_{2}} \\
X &= x_{1} \, Z \\
Y &= y_{1} \, Z - \frac{b}{2}
\end{align*}
\]

Thus we can calculate the distance which separates the mobile robot and the target, which are given by the following equation:

\[ d = \sqrt{Y^{2} + Z^{2}} \]  

The angle of deviation is given by:

\[ \varphi = \tan^{-1} \left( \frac{y}{Z} \right) \]

To evaluate the performances of the depth estimation using our algorithm, we took a series of measurement at different distance from the target, the results are presented by the following figures:

VII. VISUAL CONTROL OF THE MOBILE ROBOT

In this section we describe the two different robot controllers, that have been integrated in our tracking control, the following figure illustrate the case of our application settled leader follower [17], the goal is the follower tacked the leader with conservation distance of security.
A. Complete system design and hardware

Two Pioneer wheeled robots were used in our experiment, one as the pursuer and the other as the target. On the pursuer robot, we mounted a Bumblebee2 stereoscopic camera, which was operated at a resolution of $320 \times 240$ pixels, and it contains embedded computer with a CPU of 1.6 GHZ, the system was implemented in C++ with the openCV library of image processing [18] and the ARIA software development environment [19] running in windows XP operating system. It operated in real-time, with a calculation period of 0.3s.

a. First Controller: Non linear controller based Lyapunov method

The control objective is defined as follows:
Assuming a case where a leading robot moves along an unknown trajectory, make the follower robot keep a desired distance $d$ to the leader and pointing to it (that $\phi = 0$), using only visual information, the control objective can be expressed as:

\[
\lim_{t \to \infty} e_1(t) = \lim (d_1 - d) = 0
\]
\[
\lim_{t \to \infty} e_2(t) = \lim (\phi - \phi) = 0
\]

(29)

According to the fig.35 we have
\[
\phi = \pi + \phi - \Psi
\]
\[
d = \sqrt{x^2 + y^2}
\]
\[
\dot{d} = \frac{x\dot{x} + y\dot{y}}{\sqrt{x^2 + y^2}}
\]
\[
\begin{bmatrix}
    x \\
    y \\
\end{bmatrix}
= d \cos (\psi) \\
\begin{bmatrix}
    x \\
    y \\
\end{bmatrix}
= d \sin (\psi)
\]

(30) (31) (32) (33)

Replacing (1) and (33) in (32) we obtain:
\[
\dot{d} = -v \cos (\phi)
\]
\[
\phi = \dot{\phi} - \psi
\]
\[
\tan(\psi) = \frac{y}{x}
\]
\[
\dot{\psi} = \frac{y\dot{x} + x\dot{y}}{x^2 + y^2}
\]

(34) (35) (36) (37)

Replacing (1) and (33) in (37) we obtain:
\[
\psi = -\frac{v}{d} \sin(\phi)
\]
\[
\dot{\phi} = \omega + \frac{v}{d} \sin(\phi)
\]

(38) (39)

If we arrange the expression of $\dot{\phi}$ and $\dot{d}$ into matrix form we get:
\[
\begin{bmatrix}
    \dot{d} \\
    \dot{\phi} \\
\end{bmatrix}
= \begin{bmatrix}
    -\cos(\phi) & 0 \\
    \frac{1}{d} \sin(\phi) & 1
\end{bmatrix}
\begin{bmatrix}
    v \\
    \omega
\end{bmatrix}
\]

(40)

The evolution of the posture of the follower robot relative to the leader will be stated by the time derivative of the two error variables, the variation of distance is given by:
\[
\dot{e}_d = \dot{d}_1 - \dot{d} = -d
\]
\[
\dot{e}_\phi = v \cos(\phi)
\]

(41) (42)

Likewise the variation of the angle error is given by:
\[
\dot{e}_\phi = \dot{\phi} - \dot{\phi} = -\phi
\]
\[
\dot{e}_\omega = \left(\omega + \frac{v}{d}\right) \sin(\phi)
\]

(43) (44)

Posing:
\[
\begin{bmatrix}
    \dot{e}_d \\
    \dot{e}_\phi \\
\end{bmatrix}
= - K_1 (e_1)
\]
\[
\begin{bmatrix}
    \dot{e}_d \\
    \dot{e}_\phi \\
\end{bmatrix}
= - K_2 (e_2)
\]

(45)

The non linear controller is calculated by:
\[
\begin{aligned}
    v &= \frac{-1}{\cos(\phi)} f_{sa}(e_v) \\
    \omega &= f_{sa}(e_v) - \frac{v}{d} \sin(\phi)
\end{aligned}
\]  

(46)

\( f_{sa}(e_v), f_{sa}(e_a) \in \Gamma \) with \( \Gamma \) the set function that meet the following definition:

\[
\Gamma = \{ f : \mathbb{R} \to \mathbb{R} / f(0) = 0 \text{ and } x f(x) > 0 \ \forall \ x \in \mathbb{R} \}
\]

We chose the function \( f_{sa}(e_v), f_{sa}(e_a) \) used in [20]:

\[
\begin{aligned}
    f_{sa}(e_v) &= k_v \tanh(\lambda_v e_v) \\
    f_{sa}(e_a) &= k_a \tanh(\lambda_a e_a)
\end{aligned}
\]

(47)

With: \( k_v, \lambda_v, k_a, \lambda_a \) is positive gain.

The variable by this controller \( (\phi, d) \) as given by the equation (27) and (28) are calculated by the stereo vision system.

• Stability analysis

Let us consider Lyapunov function as follows:

\[
V = \frac{1}{2} (e_v^2 + e_a^2) > 0
\]

(48)

The derivative of (48) is given by:

\[
\dot{V} = e_v \dot{e}_v + e_a \dot{e}_a
\]

(49)

\[
\dot{V} = -e_v f_{sa}(e_v) - e_a f_{sa}(e_a) < 0
\]

(50)

It then concludes the asymptotic stability is verified.

We generate a circular and sinusoidal trajectory at the leader mobile robot, the parameters of the leader robot (angular and linear velocity) and the controller is shown in the following table.

Table.3 Parameters the leader robot and the controller

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>( v_v(t) ) [mm/s]</td>
<td>200</td>
</tr>
<tr>
<td>( \omega_v(t) ) [deg/s]</td>
<td>8</td>
</tr>
<tr>
<td>( d_v ) [mm]</td>
<td>500</td>
</tr>
<tr>
<td>( k_v )</td>
<td>360</td>
</tr>
<tr>
<td>( \lambda_v )</td>
<td>0.005</td>
</tr>
<tr>
<td>( k_a )</td>
<td>10</td>
</tr>
<tr>
<td>( \lambda_a )</td>
<td>0.1</td>
</tr>
</tbody>
</table>

To illustrate the efficiency of the proposed non linear controller, the experimental results is shown in the follower figures, in fig.20 and fig.24 we not that the relative distance between the leader and the follower robot is maintained constant with the desired distance, the fig.21 and fig.25 show the evolution of angle \( \phi \), the fig.22 and fig.26 show the control actions are calculated and sent to the follower robot, and the trajectory carried by the two robots is shown in fig.23 and fig.27.
Fig. 22. Desired control action

Fig. 23. Trajectories followed by the leader and the follower robots

Fig. 24. Evolution of the distance and error

Fig. 25. Evolution of the angle $\phi$

Fig. 26. Desired control action
The fuzzy logic is presented by the philosopher Max block since 1937, the concept of fuzzy logic was really introduced in 1965 by L. zadeh [21]. It is based on the mathematical theory of the fuzzy subsets, the aptitude of fuzzy logic reside in the handling of the imprecise measurements used by the human language. The number of application based on this logic increased in last years considerably, this is due to the fact that fuzzy logic is usually expressed by linguistic rules (if-then). However, this theory is still too recent to achieve the unanimity within the scientific community.

In this section, the vision based tracking control of the follower mobile robot using the proposed fuzzy logic controller is presented. For translation velocity and angular velocity controls, two controllers FLC1 and FLC2 are adopted, respectively. Let the input variables of the FLC1 and FLC2 be defined as:

\[ e_s = d - d_s \]  
\[ e_{\omega} = \phi_s - \phi \]  

We want to design the two fuzzy logic controllers FLC1 and FLC2 for the follower mobile robot such that it can pursue the target mobile robot within a constant distance. The fuzzy control rules can be represented as a mapping from input linguistic variables \( e_s(k) \), \( \Delta e_{\omega}(k) \) to output linguistic variables \( v \) and \( \omega \) as follows:

\[ v = \text{FLC1}\left(e_s\right) \]  
\[ \omega = \text{FLC2}\left(e_{\omega}\right) \]  

The membership functions of input linguistic variables \( e_s(k) \), \( e_{\omega}(k) \) and the membership functions of output linguistic variables \( v \) and \( \omega \) are shown in fig.28.
The input of the FLC1 \( e_c \) is decomposed into eight fuzzy partitions, such as: negative small (NS), very negative small (VNS), zero (ZE), very positive small (VPS), positive small (PS), positive medium (PM), positive big (PB) and very positive big (VPB). The input of the FLC2 \( e_o \) is decomposed into seven fuzzy partitions, such as: negative big (NB), negative medium (NM), negative small (NS), zero (ZE), positive small (PS), positive medium (PM), and positive big (PB).

The input \( e_c \) is divided into eight fuzzy sets and \( e_o \) is divided into seven fuzzy sets, 8 fuzzy rules for linear velocity control and 7 fuzzy rules for angular velocity control. The defuzzification strategy is implemented by the weighted average method.

\[
u = \frac{\sum_{j=1}^{N} \mu_j(u_j)u_j}{\sum_{j=1}^{N} \mu_j(u_j)}
\]

(55)

\( N \) is the number of the rules.

Where \( \mu \) may be the linear velocity or angular velocity command of the tracker mobile robot, \( u_j \) is the support of each fuzzy set \( j \), and \( u \) is the membership function value of each rule.

We generate circular and sinusoidal trajectories for the leader mobile robot, the parameters of the leader robot (angular and linear velocity) are shown in the following table.

<table>
<thead>
<tr>
<th>Table.1 Parameters the leader and the controller</th>
</tr>
</thead>
<tbody>
<tr>
<td>( v_c(t) ) [mm/s]</td>
</tr>
<tr>
<td>( \omega_c(t) ) [deg/s]</td>
</tr>
<tr>
<td>( d_c ) [mm]</td>
</tr>
</tbody>
</table>

The experimental results are presented in the following figures, in fig.29 and fig.33 we not that the fuzzy controller given a satisfied performance and a good tracking by the follower robot by maintaining the desired distance without static error, the fig.30 and fig.34 show the evolution of angle \( \phi \), the fig.31 and fig.35 show the control actions are calculated and sent to the follower robot, finally the trajectory of the leader and follower robot is depicted in fig.32 and fig.36.
Fig. 31. Desired control action

Fig. 32. Trajectories followed by the leader and the follower robots

Fig. 33. Evolution of the distance and error

Fig. 34. Evolution of the angle $\phi$

Fig. 35. Desired control action
VIII. CONCLUSION

This paper proposes two vision-based tracking control approaches for nonholonomic robot, for objective is the tracker robot follows a target vehicle by maintaining a desired distance. In spite of its implementation simplicity, but it seems clearly the Lyapunov controller is more preferment than the fuzzy logic controller specially on terms of tracking accuracy. Finally, we can say that the different experimental results verified availability of our proposed approaches.

REFERENCES


[19] Active media.


Fig.36. Trajectories followed by the leader and the follower robots