Path Planning for 3D Visual Servoing: for a Wheeled mobile Robot

Hassen Mekki[1,2] ,Manel Letaief[1]
[1] National School of Engineering of Sousse, University of Sousse,Tunisia
[2] Intelligent Control design and Optimization of complex Systems, University of Sfax, Tunisia
mekki.hassen@gmail.com, Letaiefmanel.ia@gmail.com

Abstract—**In this paper, we are interested in 3D visual servoïng path planning and path tracking .In fact, in the 3D visual servoïng task, there is no control in the image space and the object may get out of the camera field of view during servoing. To solve this problem, we have used a new approach based on a flatness concept. The 3D visual servoïng suffer from another major problem, is to determine the relative pose of the camera and the object. Generally, the pose estimation is made by correspondences between points of one image and points of the space that is the 2D-3D correspondence. In our work we have used a 3D visual sensor called Kinect. To show the efficiency of the proposed algorithm, we have implemented it on a wheeled Koala robot.**

Keywords— 3D Visual servoïng; Path planning; Path tracking; Mobile robot; Kinect camera.

I. INTRODUCTION

The visual servoing is a technique which uses feedback information extracted from a vision sensor incorporated into or outside the system to control the motion of a robot.

There are two main approaches in visual servoïng. The first approach is called Image-Based Control (IBC) or 2D visual servoing [4], [5]. In the IBC, feedback is directly defined in the image. IBC techniques are more popular due to their local stability and convergence in the presence of camera modeling and calibration errors. However, the IBC techniques suffer from stability and convergence problems, in particular when the initial and desired camera poses are distant, and from the object being able to get out of the camera view field. The second approach is called Position-Based Control (PBC) or 3D visual servoïng [2], [3]. In the PBC the control error function is computed in the Cartesian space. Image features are extracted from the image and a perfect model of the target is used to determine its position with respect to the camera frame. In the literature, there are many methods which allow determining the situation (position and orientation) of the camera [2], [3]. These methods are generally based on the knowledge of the (3D or 2D) object model and the camera intrinsic parameters. These methods use visual information of different nature, such as points [7], lines [9]…However, there are some works which use the 3D reconstruction by dynamic vision techniques, allowing estimating the object model, or localizing the camera from the 2D or 3D movement measures [16], [18]. In our work we have used a 3D Kinect camera to get the robot position.

The main advantage of the PBC approach is that it controls the camera trajectory directly in the Cartesian space. However, there are some problems; namely, there is no control in the image space, so the object may get out of the camera field of view during servoïng. In our recent work [17] we have envisaged a blind movement of the robot thanks to an open loop control, given by the flatness concept. To show the efficiency of the proposed algorithm, we have implemented it on a wheeled robot.

This paper is organized as follow: the based 3D visual serving problems are given in section 2. Section 3 describes the experimental platform. We present the used method in section 5.Section 6 shows the experimental results.

II. FORMULATION PROBLEM

In the PBC, the control error function is computed in the Cartesian space. The situation (position and orientation) of the camera is extracted from the image. The task of the 3D visual servoing consists in joining a reference situation r* as shown in Fig.1.

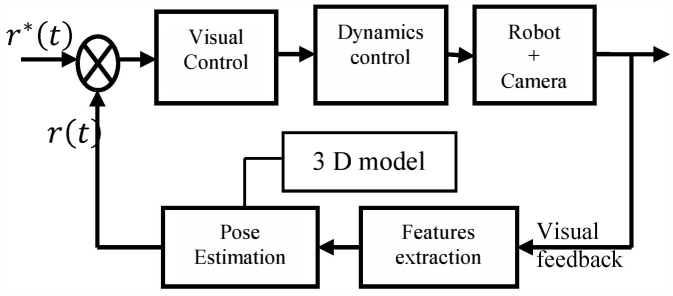


Fig.1:Based visual servoïng loop (3D visual servoïng)

The inconvenience using this approach is that there is no control in the image space and the object may get out of the camera view field during servoïng. Moreover, an uncertainty is considered, caused by the pose estimation algorithm. We are

978-1-4799-2813-2/13/$31.00 ©2013 IEEE

interested, in this paper particularly, in the path planning and path tracking problems using the flatness concept by testing the proposed algorithm on a wheeled Koala robot. To solve the features extraction and pose estimation problems, we have used a 3D Kinect camera.

III. PLATFORM DESCRIPTION

Visual servoing usually involves the implementation of three subsystems: a visual sensor, a processing system and a powered mechanical device. As shown in the Fig.2, The Kinect camera is used as a visual sensor, a computer as a processing system, and a mobile Koala robot as a mechanical device in our study.

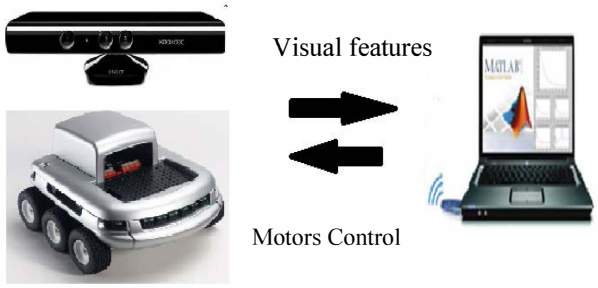


Fig.2:communication between the robot and The processing device

A. THE ROBOT MODEL:

We have worked with a mid-sized robot named Koala, designed for real-world applications, and developed by K-Team. It is a mobile robot with six wheels, 20 cm height and 4 kg weight including the battery, 30x30 cm volume. It can reach a maximum speed of 0.6 m / s directly or a speed of 0.38 m / s using Proportional Integral Differentiator (PID) controller speed, and for a maximum acceleration it can reach up to 0.7m / s, also using the PID controller. The Koala robot can be connected to a computer that transmits movement commands using an RS232 serial link cable or wireless [14].

The wireless communication range is 100 meters and the transmission speed reaches up to 115.2 Kb / s with a frequency of 2.45 GHz.

Fig.3: Koala Robot

The considered mobile robot model is shown in Fig.4.[13]

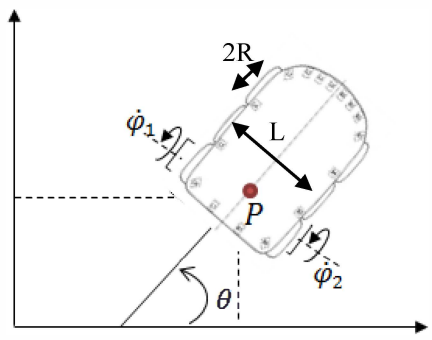


Fig. 4: The Robot model

L: Length of the axis; $L = 138mm$

R : radius of each wheel;$R = 42mm$

$\dot{\varphi}_1, \dot{\varphi}_2$: The velocities of the points of contact of the wheels in respect to the ground

(x, z): The position variables

θ: The orientation angle of the robot

The robot kinematic model can be written as follows (without considering slip phenomenon):

$$\begin{cases} \dot{x} = u_1 \cos(\theta) \\ \dot{z} = u_1 \sin(\theta) \\ \dot{\theta} = u_2 \end{cases} \quad (1)$$

$$\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} \frac{R}{2} & \frac{R}{2} \\ \frac{-R}{L} & \frac{R}{L} \end{pmatrix} \begin{pmatrix} \omega_1 \\ \omega_2 \end{pmatrix} \quad \dot{\varphi}_1 = \omega_1 R \;\; \dot{\varphi}_2 = \omega_2 R$$

B. THE KINECT

In our work we have used Kinect for the Xbox 360 which is an active 3D camera launched by Microsoft in 2010. As shown in Fig.5, it consists of two parts:

- 3D depth sensor (IR projector, IR CMOS sensor)
- RGB camera (color CMOS sensor)

The Kinect sensor works on the principle of the transmission to the scene of the IR light by an infrared projector, and the acquisition of the reflected signal by an infrared (IR) CMOS camera, with a 320x240 resolution, which allows us to get the depth information.

For each pixel in the image, it ultimately provides 4 pieces of information: the three components of color, and the depth (except at the edges of the image).

From a technical standpoint, the Kinect produces images with a resolution of 640 × 480 pixels at 30 frames per second. The opening of the 3D Kinect camera's angle is about 58 ° and is smaller than that of the color Kinect camera.

We can know the distance to any object between 1.5 to 2 meters away from the camera to about 4-5 meters deep. Beyond 5 meters, the reflected IR radiation is too low to be measured accurately.[15]

Fig. 5: KINECT camera

IV. THE USED METHOD

A. Controller design

The control objective is to perform an asymptotic tracking of a desired trajectory $(x^*, z^*)$and can reject some additive disturbances.

Using (1) we can show easily that the coordinates of $P(x(t), z(t))$ are the flat output of the system. In fact:

$$\theta = \arctan\left(\frac{\dot{z}}{\dot{x}}\right) \tag{2}$$

$$u_1 = \sqrt{\dot{x}^2 + \dot{z}^2} \tag{3}$$

$$u_2 = \frac{\ddot{z}\dot{x} - \dot{z}\ddot{x}}{\dot{x}^2 + \dot{z}^2} \tag{4}$$

We can notice here that the relation between the inputs and the flat outputs is not invertible. We introduce, as an auxiliary control input, the time derivative of u_1 , so we have:

$$\dot{u}_1 = \frac{\dot{x}\ddot{x} + \dot{z}\ddot{z}}{\sqrt{\dot{x}^2 + \dot{z}^2}} \tag{5}$$

This control input extension yields now an invertible control input to flat output relation of the form:

$$\begin{pmatrix} \dot{u}_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} \frac{\dot{x}}{\sqrt{\dot{x}^2+\dot{z}^2}} & \frac{\dot{z}}{\sqrt{\dot{x}^2+\dot{z}^2}} \\ \frac{-\dot{z}}{\dot{x}^2+\dot{z}^2} & \frac{\dot{x}}{\dot{x}^2+\dot{z}^2} \end{pmatrix} \begin{pmatrix} \ddot{x} \\ \ddot{z} \end{pmatrix} \tag{6}$$

In this case we use an exact feed forward linearization based on the differential flatness proposed by Hagenmeyer-Delaleau [13] which consists in replacing $\dot{x}$ and $\dot{z}$ by its desired values given by $\dot{x}^*$ and $\dot{z}^*$ respectively. The linearization process leads to the following expression:

$$\begin{pmatrix} \dot{u}_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} \frac{\dot{x}^*}{\sqrt{\dot{x}^2+\dot{z}^2}} & \frac{\dot{z}^*}{\sqrt{\dot{x}^2+\dot{z}^2}} \\ \frac{-\dot{z}^*}{\dot{x}^2+\dot{z}^2} & \frac{\dot{x}^*}{\dot{x}^2+\dot{z}^2} \end{pmatrix} \begin{pmatrix} \ddot{x} \\ \ddot{z} \end{pmatrix} \tag{7}$$

Then, the non- interacting control of the system is solved and moreover, the resulting system is equivalent to the control of two chains of integrators of the form:

$$\begin{cases} \ddot{x} = \vartheta_x \\ \ddot{z} = \vartheta_z \end{cases} \tag{8}$$

where ϑ_x and ϑ_y are auxiliary control inputs to be specified in accordance with the trajectory tracking control objectives.

B. Flatness generation of trajectories

We suppose that we want to move the robot from an initial position (x_i^*, z_i^*) at the instant t_i, , to a final position (x_f^*, z_f^*) at the instant t_f, and to make it pass, for example by the point denoting by $\left(\frac{x_f^*+x_i^*}{2}, 2z_f^* - z_i^*\right)$ which must be the maximum of this curve between z_i^* and z_f^*. In order to avoid an obstacle as shown by Fig.6, this trajectory $z^*(x^*)$ must verify the four conditions denoted by:

$$z^*(x_i^*) = z_i^*,$$

$$z^*(x_f^*) = z_f^*,$$

$$z^*\left(\frac{x_f^*+x_i^*}{2}\right) = 2z_f^* - z_i^*,$$

$$\frac{dz^*}{dx^*}\left(\frac{x_f^*+x_i^*}{2}\right) = 0\ ;$$

With the constraint in $\frac{d^2z^*}{d^2x^*}\left(\frac{x_f^*+x_i^*}{2}\right) < 0$, in order to have a local maximum in this point.

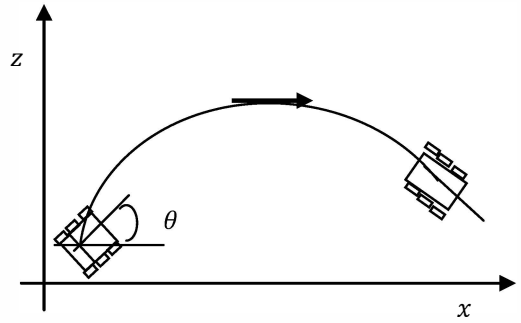


Fig. 6:Path planning in the Cartesian space

We can choose the polynomial of a third degree in x^*denoted by:

$$z^*(x^*) = z_i^* + (z_f^* - z_i^*)\left(\frac{x^* - x_i^*}{x_f^* - x_i^*}\right)\left(9 - 12\left(\frac{x^* - x_i^*}{x_f^* - x_i^*}\right) + 4\left(\frac{x^* - x_i^*}{x_f^* - x_i^*}\right)^2\right) \tag{9}$$

Which has satisfied the last conditions.

It remains to construct the trajectory of $x^*(t)$which verify:

$$x^*(t_i) = x_i^*, \dot{x}^*(t_i) = 0, \cdots, x^{*(5)}(t_i) = 0$$

$$x^*(t_f) = x_f^*, \dot{x}^*(t_f) = 0, \cdots, x^{*(5)}(t_f) = 0$$

Then, we will get the polynomial of degree 11 denoted by:

$$x^*(t) = x_i^* + (x_f^* - x_i^*)\sigma^6(t)\big(462 - 1980\sigma(t) + 3465\sigma 2t - 3080\sigma 3t + 1386\sigma 4t - 252\sigma 5tt \quad (10)$$

$$\text{With } \sigma(t) = \frac{t - t_i}{t_f - t_i}$$

C. Flatness and tracking of trajectory

The dynamic endogenous feedback calculated in subsection (1) permits choosing the new control as:

$$\vartheta_x = -\left(\frac{k_2^x s^2 + k_1^x s + k_0^x}{s(s + k_3^x)}\right)(x - x^*) + \ddot{x}^* \quad (11)$$

$$\vartheta_z = -\left(\frac{k_2^z s^2 + k_1^z s + k_0^z}{s(s + k_3^z)}\right)(z - z^*) + \ddot{z}^* \quad (12)$$

Where $k_i^x, k_i^z \in \mathbb{R}, i = 0, 1, 2$ are chosen so that the polynomials:

$$P_x(s) = s^4 + k_3^x s^3 + k_2^x s^2 + k_1^x s + k_0^x = 0 \quad (13)$$

$$P_z(s) = s^4 + k_3^z s^3 + k_2^z s^2 + k_1^z s + k_0^z = 0 \quad (14)$$

present a Hurwitz polynomial. So we have an asymptotic tracking of the trajectory (x^*, z^*).

In what precedes, we suppose that we can get, at any time, the coordinates x and z of the point P. Two cases are possible: In the first case, the object does not leave the camera field of view during servoïng. In this case, the control law scheme is given by Fig.7. The second case is that the object can get out of the camera field of view during servoïng. In this case, we will envisage a blind movement of the robot thanks to an open loop control, given by Fig.8.

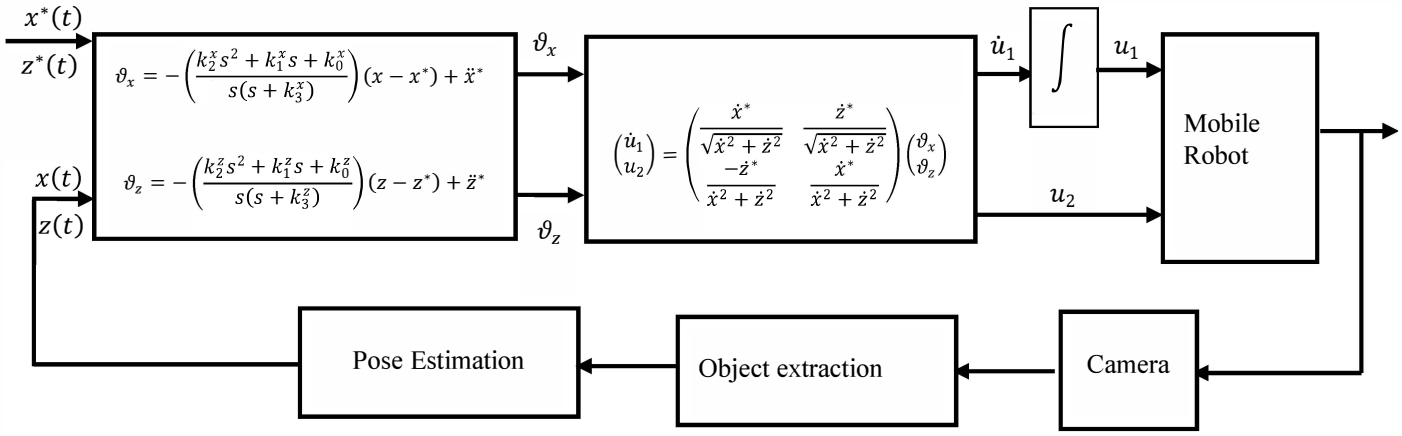


Fig.7: Case 1: closed loop : If the object is in the field of vision.

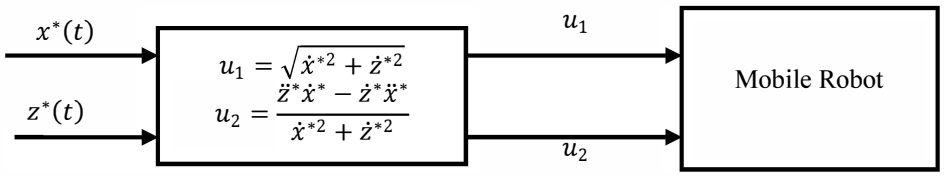
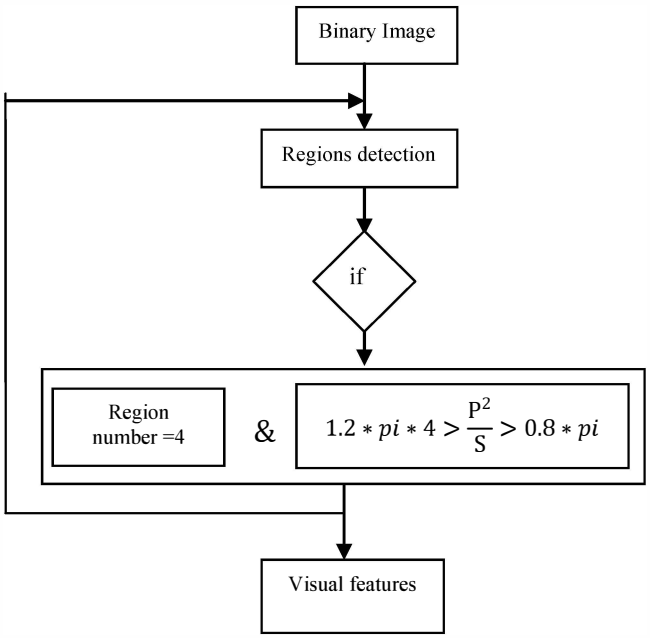


Fig. 8: Case 2: open loop: If the object is not in the field of vision (Blind movement

V. EXPERIMENTAL RESULTS:

To show the efficiency of the proposed algorithm, they have been implemented on a wheeled Koala robot. The target is a planar object with four circles. We have extracted the center of each circle by following the steps of this flow diagram.



First we have transformed the RGB image to a binary image, then we have detected the different regions (perimeter, surface, center) after that we have made conditions to detect a circle, so if the condition of the relationship between the perimeter and the square surface is checked then it is a circle, if not it is unnecessary to hold the found coordinates.

In this work we are interested only in the case where the object does not leave the camera view field.

The experimental parameters:

The initial situation xi=0mm; zi=0mm;

the final situation xf=750mm;zf=100mm;

the sample time is Te=0,33s.

Fig.9 shows the evolution of the trajectory of the robot in the Cartesian space. (1) describes the robot path .(2) describes the desired path. To stay in the field of view the object have got to be far from the camera, but seen the Kinect' limits of detection(less than 5 meters), that was not allowed. In fact the fluctuation's amplitude shown in Fig.9 is in millimeters. So it is clear that the problem is further solved through the proposed algorithm and the error in millimeters is acceptable. The robot is capable of joining its desired trajectory.

To enhance the result, we have tried to minimize the sample time, but due to the performance of the computer and the matlab, this conditions have not helped us to get better results.

Fig. 10 and 11 respectively show the variations of the two commands applied to the robot. It seems that the two commands are the same and that because we have chosen a straight line trajectory.

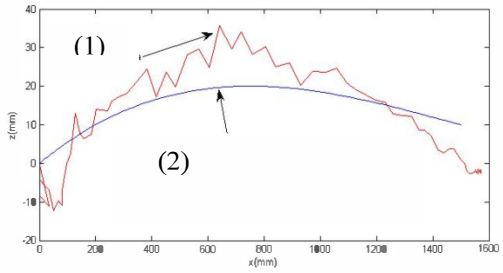


Fig.9: the position responses in the Cartesian

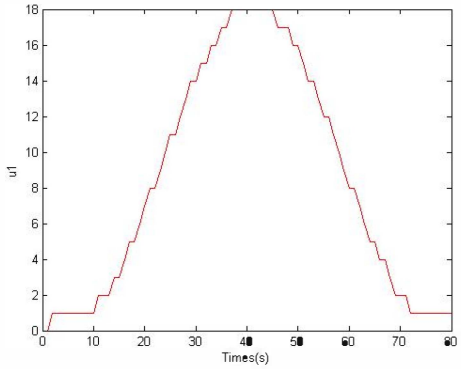


Fig.10: The control variable

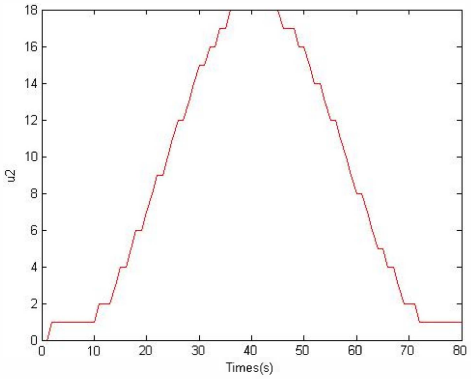


Fig.11: The control variable

VI. CONCLUSION:

In this paper we have used a flatness based path planning for visual robust-3D servoîng in order to overcome the major problem of the 3D servoing .To follow the exact path planning, an open loop control has been used, given by the flatness concept. Seen that the pose estimation generally puts some errors in determining the exact situation of the robot, we have solved it with a 3D Kinect camera. To show the efficiency of the proposed algorithm, we have used a wheeled Koala robot. To get better result it is required to implement this algorithm in a more robust platform such as the FPGA.

REFERENCES

[1] A. Chelouah, E. Delaleau, P. Martin and P. Rouchon, Differential flatness and control of induction motors, symposition on contol, optimization and surpervision; computational engineering in system application, IMACS multiconference, pp. 80-85, Lille, 9-12 July 1997.

[2] B. Allotta, D. Fioravanti,"3D motion planning for imagebased visual servoîng tasks", Proc. IEEE Int. Conf. Robot,Autom., pp 2173-8, 2005.

[3] F. Chaumette, S. Hutchinson, "Visual servo control Part I: Basic approaches", IEEE Robot Autom Mag. Vol.13, no.4, pp.82-90, 2006.

[4] F. Chaumette, S. Hutchinson, "Visual servo control Part II: Advanced approaches", IEEE Robot Autom Mag vol. 14, no. 1, pp.109-18, 2007.

[5] F. Chaumette and S. Hutchinson, "Visual servo control, i. basic approaches", IEEE Robot Autom Mag., vol. 13, no. 4, pp. 82-90, 2006.

[6] F. Chaumette, "Potential problems of stability and convergence in image-based and position-based visual servoîng, the confluence of vision and control", LNCIS Series, Springer Verlag, 237, pp. 66-78,, 1998.

[7] G. Chesi and Y. Hung, "Global path planning for constrained and optimal visual servoîng", IEEE Trans. On Robot., vol. 23, no. 5, pp. 1050-1060, 2007.

[8] G. Chesi, "Desigining image trajectories in the presence of uncertain data for robust visual servoîng path planning", Proc. IEEE Int. Conf. Robot. Autom, pp 1492-7, 2009.

[9] G. Chesi, D. Prattichizzo,A. Vicino, "Straight line path planning in visual servoîng", Trans. ASME, J Dyn Syst Meas Control vol. 129, no.4, pp. 541-3, 2007.

[10] G. Lopez-Nicholas., S. Bhattacharya., and al., "Switched homography-based visual control of differential drive vehicles with field of view constraint". Proc. IEEE Int. Conf. Robot. Autom. pp. 4238-44, 2007.

[11] J. Lévine, P. Rouchon, G. Yuan, al., "On the control of US navry cranes", European control conference, ECC'97, Brussels, July 1997.

[12] J. Lévine, "Analysis and control of nonlinear systems, A flatness-based approach", Springer, New York, 2009.

[13] J. Cortés-Romero, A. Luviano-Juarez, al., Flatness- based robust control of a mobile robot », CLCA/CAC, Merida, Venezuela, 25-28 Nov., 2008

[14] K-Team S.A. Koala user manual. [en ligne], 23 Février 1999.

[15] LEJEUNE Antoine ,Piérard Sébastien ,Droogenbroeck Marc Van ,Verly Jacques. Utilisation de la Kinect . . [en ligne], Juillet 2012

[16] M. Fliess, J. Lévine. Ph, Martin and P. Rouchon, "On differentially flat nonlinear systems", IFAC-Synopsium, NOLCOS'92 pp. 408-4012, Borddeaux, 1992.

[17] Mekki, H, Flatness-based path planning for robust 3D visual servoïng, 10th IEEE International Multi-Conference on Systems, Signals and Devices, Mars 2013, Hammamet –Tunisia

[18] R. Rothfuss, J. Rudolph and M. Zeitz, "Flatness based control of chemical reactor model", European control conference, pp. 637-642, Rome, Septembre 1995.

[19] Y. Mezouar and F. Chaumette, "Path planning for robust image-based control", IEEE Trans. Robot. Autom., vol. 18, no. 4, pp. 534-549, 2002.