Reactive Mobile Robots Based on a Visual Servoing Approach

Patrick Rives

INRIA Sophia Antipolis
2004 Route des Lucioles
06565 Valbonne Cedex - France
rives@sophia.inria.fr

Roger Pissard-Gibollet

ISIA-Ecole des Mines de Paris
Rue Claude Daunesse
06565 Valbonne Cedex - France
pissard@isia.cma.fr

Abstract

This paper deals with vision based control applied to autonomous mobile robots. The general approach in robot vision separates vision from control (static look and move). An alternative way consists to specify the problem in terms of control directly into the sensor frame. This approach seems to be a fruitful way for implementing robotics tasks based on reactivity concept. We introduce a low level task specification by the notion of virtual linkage. *This* virtual linkage *can be expressed by a set of constraints on the motion of a frame linked to the robot with respect to a certain frame associated to the environnment.*
It seems interesting to apply such a vision based control to a mobile robot. The robot considered here is a two wheels driven nonholonomic cart with a camera mounted on 3 d.o.f manipulator. In this study we consider the whole mechanical system constituted by the cart and its manipulator like a single kinematic chain. So in adding the degrees of freedoom of the arm to the cart, it becomes possible to fully control the trajectory of the effector.
In order to validate this approach, a simulation tool has been developped in Mathematica. *A first application concerns automatic exploration and cartography of an unknowm room by wall following techniques.*
Key words : *vision based control or visual servoing, virtual linkage, nonholonomy, mobile robot control, kinematic model, vision oriented task.*

1 Introduction

Recent improvements in the field of vision sensors and image processing hardware allow to hope that the use of vision data directly into the control loop of a robot is no more an utopic way. Commonly, the general approach in robot vision is the following: processing vision data into the frame linked to the sensor, converting data into the frame linked to the scene by means of inverse calibration matrix and computing the control vector of the robot into the frame linked to the scene. This scheme works in open loop with respect to vision data and cannot take into account errors in the measurement or in the estimation of the robot position. Such an approach needs to perfectly overcome the constraints of the problem: the geometry of the sensor (by example, in a stereovision method), the model of the environment and the model of the robot. In some cases, this approach is the only one possible but, in many cases, an alternative way consists to specify the problem in terms of control directly into the sensor frame. This approach is often referred as visual servoing [WEI 84] or sensor based control [ESP 87]; in this case, a closed loop can be really performed with respect to the environment and allows to compensate the perturbations by a robust control scheme. Moreover, this approach seems to be a fruitful way for implementing robotics tasks based on *reactivity concept* [BRO 86]. The work described in this paper deals with such an approach applied to autonomous mobile robots using vision sensor.

2 Task Planning

It is usual to distinguish several levels in the problem of task planning. At the high level, problems of path planning and off line programming have often been addressed. Sophisticated algorithms using Computational Geometry results allow us to compute efficiently free trajectories in the configuration space taking into account geometric and some kinematic constraints [LAU 86], [BAR 89]. Unfortunately, in the major cases, from the paths computed by these methods, it is not obvious to deduce an efficient control scheme performing the task with good properties of robustness and accuracy with regard to the perturbations due to the imperfect modelling of the robot and its environment and to the errors in the measurements. For completing correctly the task, we need to translate the path provided by the high level in a specification at the control level in terms of interactions between the robot and its environment. For doing that, we introduce a low level task specification which can be expressed by a set of constraints on the motion of a frame linked to the robot with respect to a certain frame associated to the environment. This set of constraints is called a linkage by analogy to the Mechanism Theory. The class of the linkage is the dimension of the subspace of compatible velocities which are let unconstrained by the linkage. This subspace of compatible velocities can be also viewed as a

---

[0] This work is supported by firm ECA (BP 242, 83078 Toulon France) under a CIFRE contract

screw space and it fully characterizes the interactions between the robot and its environment. This concept of linkage plays an essential role in the synthesis of control laws using contact sensors for assembly applications. It can be easily extended to the case of no contact sensors like proximity or vision sensors. So, we can define the notion of *virtual linkage* by replacing the mechanical constraints by, for example, optical constraints like bringing optical axis of a camera into alignment with a certain axis of a frame attached to the scene.

3 Vision Oriented Tasks

Now, we apply the different concepts above described to the particular case of mobile robots using vision sensors. We only give the main characteristics of this approach, more inquires can be found in [CHA 90], [CHA 91] and [SAM 90].

3.1 Modelling the interaction camera-environment

A camera (C) is linked to a robot (R) with related frame F_C. (C) interacts with an environment at which is associated a frame F_T. The position (location and attitude) of (R) is an element $\bar{r}$ of SE_3 which is an six dimensional differential manifold. The sensor (C) is linked to a mechanical system with several degrees of freedom what means that the joint coordinates $\bar{q}$ constitutes a local chart of SE_3.

Let us assume that the image given by (C) is fully characterized by the relative position of (C) with respect to (T) (we don't consider image changes dues to lightning variations).Moreover, let us consider that the information into the image may be modelled as a set of *elementary signals* which result from the projection into the image of the 3D objects belonging to the scene. Then, these elementary signals constitute a vector $s(\bar{q}, t)$ which is fully defined by the position of the robot $\bar{q}$ and the time t representing the own motion of the object into the scene. Deriving this vector, we obtain :

$$\dot{s} = H \bullet T_{CT}, \tag{1}$$

where
H is called the *interaction screw*, T_{CT} is the velocity screw of the camera and $\bullet$ represents the screw product.

Considering the bilinear form L^T of the interaction screw H expressed in the camera frame F_C, and with an obvious breach of notation, we obtain :

$$\dot{s} = L^T.T_{CT}, \tag{2}$$

Interaction matrix can be derived for many exteroceptive sensors. In the case of vision sensor, elementary signals can be chosen among a set of geometrical primitives like points, lines, circle... An analytical expression for the interaction matrix when the image features are general algebraic curves can be also obtained (for more details, see [CHA 90]).

3.2 Relating Interaction Matrix and Virtual Linkage

Let us consider a complex robotic task (ie, an obstacle avoidance task), as a succession of elementary subtasks which can be expressed in terms of a succession of virtual linkages which have to be realized between the robot and its environment. Let us now consider the case where visual signal output is invariant ($\dot{s} = 0$) with respect to a set of camera motions defined by T^*_{CT} :

$$\dot{s} = L^T.T^*_{CT} = 0 \tag{3}$$

It can be easily seen that :

$$T^*_{CT} = Ker(L^T) \tag{4}$$

where $Ker(L^T)$ is the kernel of interaction matrix.

Now, coming back to the definition of the virtual linkage, we constate that the subspace spanned by T^*_{CT} is precisely the subspace of compatible velocities which leads no change in the image during the motion of the camera. According to this fact, it becomes possible to define a virtual linkage or more generally a set of constraints between the camera frame and a frame associated to a part of the environment, by a set of elementary visual signals in the image invariant with respect to a motion belonging to the subspace of compatible velocities spanned by the virtual linkage. To perform a specific virtual linkage, many different features in the image can be used like points, lines, area, inertial moments and so on... In practice, their choice is guided by the application : a priori knowledge about the scene, low cost image processing, robustness with respect to the measurement noise and, as we shall see later, good properties with respect to the control scheme.

By this formalism, this approach gives a fruitful way for specifying a low level robotic task in terms of an *image target* to be reached when the task is well performed. It appears us it could be a straightforward manner for implementing reactivity approaches using vision sensors. In the next section, we shall show how such an approach can be embedded in a robust closed loop control scheme using the *task function formalism* [SAM 90].

3.3 Sensor Based Task Function

A now classical approach in robotics it to consider the process of achieving a task such as tracking or positioning as a problem of regulation of a certain function: *the task function.* For example, if the imposed task is to follow a given trajectory in cartesian space, the task function approach version of this problem would be: keep $\|e(t)\|$ minimum through time. Where *e the task function vector* is: $e(t) = \bar{r}(q(t)) - R^*(t)$, R^* is the desired trajectory. The control vector in this problem is the set of joint torques we apply to change the joint coordinates (denoted $\underline{q}$ in our equations). The task is achieved when $\|e(t)\| = 0$ (see [SAM 90] for full details about the task function approach in robot control). The application of the task function approach to sensor based tasks is straightforward: the task is now described as a desired values for a certain set of elementary visual signals to obtain. In mathematical terms the general definition of a visual task function vector e is :

$$e\ (t) = C\ (s\ (\bar{r}(q(t))) - s^*(t)) \tag{5}$$

where

- s^* can be considered as a desired value of the vector of the elementary visual signals (reference image target) to be reached in the image.
- $s\ (\bar{r})$ is the value of the visual features currently observed by the camera, which only depends on the position between the camera and the scene.
- C is a matrix which allows, for robustness issues, to take into account more visual features than necessary. This matrix must have some properties as we will see later, for ensuring convergency at control level.
- $\underline{q}$ and $\bar{r}$ are the robot joint and cartesian coordinates.

The robot coordinates q appear in this equation since we do not have direct control on the cartesian coordinates $\bar{r}$ but rather on the joint coordinates $\underline{q}$.

Once again we should emphasize the important concept below the formalism contained in this equation: the task is not described in the cartesian or joint coordinates of a robot but in terms of image characteristics. So the vision process does not only provide a mere tool to verify that a closed loop based on robot coordinates behaves correctly and achieves the task, but rather permits us to now deal with a closed loop based on vision information.

The quantity which arises naturally in our output regulation problem is *task error* $\dot{e}$. This matrix governs the controllability, robustness and convergence of a feedback control law in visual signals space. For simplicity, we deal with a very simple example of *velocity control scheme* using visual feedback (for more advanced control schemes, see [SAM 90]). Moreover, if we assume the presence of an approriate robot's velocity regulation loop, we can use as input, the velocity T_c of the camera ($T_c = J(\underline{q}).\underline{\dot{q}_c} \simeq J(\underline{q}).\underline{\dot{q}} = T_{CT}$ where $J(\underline{q})$ is the jacobian of the robot). We thus may choose the following control law :

$$T_c = -\lambda e \tag{6}$$

with $\lambda > 0$.

Let us consider the assumption that e is only depending of $(\bar{r}, t)$, we get :

$$\dot{e} = \quad \frac{\partial e}{\partial s}.\frac{\partial s}{\partial \bar{r}}.T_c + \frac{\partial e}{\partial t} \tag{7}$$

$$\dot{e} = \quad C.L^T.T_c + \frac{\partial e}{\partial t} \tag{8}$$

where $\frac{\partial e}{\partial t}$ parametrizes an eventual motion of the target. In the case of a motionless target $\frac{\partial e}{\partial t} = 0$, and we obtain :

$$\dot{e} = \quad C.L^T.T_c = -\lambda\ C.L^T.e \tag{9}$$

An exponential convergence will thus be ensured under the sufficient condition:

$$CL^T > 0 \tag{10}$$

in the sense that a $n \times n$ matrix A is positive if $x^T A x > 0$ for any nonzero $x \in \mathbf{R}^n$.

A good and simple way to satisfy this convergence condition in the neighbourhood of the desired position is to choose for the matrix C the generalized inverse of the interaction matrix associated to s^*:

$$C = {L^T}^+_{|s=s^*} \tag{11}$$

4 Vision Based Control Applied to a Mobile Robot

Applying vision based control theory to mobile robots provides an interesting framework for implementing reactive tasks. However it is necessary to take some care due to the presence of nonholonomic constraints which implies that the dimension of the configuration space is larger than the number of degrees of freedom really controllable. Due to this fact, it is now wellknown [SAM 91] that if we considered a two-wheels driven nonholonomic cart, despite the controllability of the system, pure state feedback stabilization of the cart's configuration around a given terminal configuration is not possible. This particularity has for consequence that, for a camera rigidly attached to the mobile base, we cannot ensure a convergence of the task error by using simple gradient like methods in the image frame. One way to avoid this problem consists in adding some degrees of freedom to the camera by mounting it on the wrist of a manipulator. Then, if we consider the whole mechanical system constituted by the cart and its manipulator like a single kinematic chain, we can show that the mapping between the configuration space (d.o.f of the cart + d.o.f of the manipulator) and the position (attitude and location) of the camera is almost everywhere regular (excepted in some isolated configuration) and it becomes possible to fully control the motion of the camera without being limited by the cart nonholonomic constraints. By this way, we can execute visual servoing tasks directly defined in the camera frame in the same manner that previously, in the case of manipulator. Obviously, in such an approach, we will not be able to control explicitely the trajectory of the cart but, by using some trivial geometric considerations, we can bound the volume which will be occupied by the cart during its displacement. We apply this approach to our system constituted by a two wheels driven cart with a camera mounted on a 3 d.o.f manipulator.

4.1 Computing the kinematic model

The system (Figure 1) has 5 d.o.f parametrized by its joint coordinates $\underline{q} = (q_d, q_g, q_1, q_2, q_3)^T$, where :

- (q_d, q_g) are the joint coordinates of the cart wheelss,
- (q_1, q_2, q_3) are the joint coordinates of the arm.

In order to simplify the calculus of the Jacobian, we choose a parametrization for the orientation of the camera which directly provides us a 5×5 square matrix : we use roll (Φ), pitch (Θ), yaw (Ψ) (RPY) representation where:

$$\Phi = q_1 + \frac{\pi}{2},\ \Theta = 0,\ \Psi = \frac{\pi}{2} - q_2 - q_3. \tag{12}$$

From this representation, we express the geometric and kinematic model of the arm linking the base frame (F_N) to the camera frame (F_C) :

- **geometrical model of the arm :**

$$\bar{r}^{(F_N)} = GM(q_1, q_2, q_3)\bar{r}^{(F_C)} \tag{13}$$

- **kinematic model of the arm :**

$$\dot{\bar{r}}_{ypr}^{(F_N)} = J_{arm}.(\dot{q_1}, \dot{q_2}, \dot{q_3})^T \tag{14}$$

where J_{arm} expression is:

$$\begin{pmatrix} -s_1(l_2c_2 + l_3c_{23}) & -c_1(l_2s_2 + l_3s_{23}) & -c_1l_3s_{23} \\ c_1(l_2c_2 + l_3c_{23}) & -s_1(l_2s_2 + l_3s_{23}) & -s_1l_3s_{23} \\ 0 & l_2c_2 + l_3c_{23} & l_3c_{23} \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & -1 \end{pmatrix} \tag{15}$$

This expression is given for the simple case where the effector's point considered is on the focal center of the camera. In order to simplify notations we use : $cos(q_i) = c_i, sin(q_i + q_j) = s_{ij}$.

- **kinematic model of the cart :**

$$(V_M, \Omega_{F_M/F_O}) = J_{cart}.(\dot{q_d}, \dot{q_g})^T \tag{16}$$

where

$$J_{cart} = \begin{pmatrix} \frac{r}{2} & \frac{r}{2} \\ 0 & 0 \\ 0 & 0 \\ \frac{r}{2R} & -\frac{r}{2R} \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \tag{17}$$

with r is the radius of the wheels and $2R$ is the distance between the two rear wheels.

Using fundamental kinematic relation, we can write :

$$\begin{cases} \vec{V_C} = \vec{V_M} + \Omega_{F_M/F_O} \wedge \vec{MC} + \vec{V_{C/F_M}} \\ \Omega_{F_C/F_O} = \Omega_{F_M/F_O} + \Omega_{F_C/F_M} \end{cases} \tag{18}$$

Using J_{arm}, J_{cart} and the previous equations, we can compute the jacobian matrix of the whole mechanism system:

$$\left(\begin{array}{c|c} \Omega_{F_M/F_O} \wedge \vec{MC} + \vec{V_M} & J_{arm} \\ \Omega_{F_M/F_O} & \end{array} \right) \tag{19}$$

In this matrix, the line corresponding to the roll velocity is a null line which can be eliminated. So J becomes J_c, a 5×5 square matrix and we can write:

$$\dot{\bar{r}}_{ypr}^{(F_N)} = J.\underline{\dot{q}} \tag{20}$$

In order to relate the joints velocities to the kinematic screw expressed in the camera frame, we compute the matrix M_p which transforms the YPR representation of the velocity expressed in the reference frame of the arm $\dot{\bar{r}}_{ypr}^{(F_N)}$ in an instantaneous rotation vector expressed in the camera frame $T_C^{(F_C)}$.

$$M_p = \begin{pmatrix} -s_1 & -c_1s_{23} & c_1c_{23} & 0 & 0 & 0 \\ c_1 & -s_1s_{23} & s_1c_{23} & 0 & 0 & 0 \\ 0 & c_{23} & s_{23} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c_{23} & s_{23} \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \tag{21}$$

and finally we obtain:

$$\dot{\underline{q}} = J_c^{-1}.M_p.T_C^{(F_C)} \tag{22}$$

The computation of J_c determinant gives :

$$Det(J_c) = -\frac{dl_2r^2c_2}{2R} \tag{23}$$

Singular configurations appear only when the robot reaches its limit configurations ($q_2 = \frac{\pi}{2} + k\pi$). Physically these configurations correspond to the second segment (l_2) vertical where the camera can't go up vertically. $d \neq 0$ and $l_2 \neq 0$ assumes that cart rotation with q_1 and q_2 with q_3 are not redundants.

So we can conclude that there are no singularity coming from the nonholonomic constraints and J_c is alway a full rank matrix when the manipulator does not reach its limit configurations.

5 Simulation results

Visual servoing approach has been yet validated in simulation and real experiments in the case of manipulator (see [CHA 90], [RIV 89]). Translating this approach in the case of mobile robots needs to include new capabilities in our simulation tool. In that aim, a new software has been developped using the symbolic language *Mathematica* to aid the design and the test of vision based task by providing an useful assistance for, by example, modelling the robot and its environment or for selecting visual features to be controlled in the task. This simulation tool contains several modules wich provide four kinds of functionnalities :

- *Robot Module* : These modules allow to describe the geometry of the robot and to compute in a symbolic way, the different models (geometric and kinematic) which can be used in the simulation. Some graphic tools allow us to verify the correctness of the model.

- *Environment Module* : It provides a data structure and functions which allow to define a 3D scene using geometric primitives like points and segments (which will be soon extended to CSG models). The camera uses a simple perspective model and its intrinsic parameters like focal lenght and clipping can be chosen by the user. It is also possible to link the camera frame to any position with respect to the robot frame.

- *Visual Servoing Module* : It allows to compute, always in a symbolic way, the interactions matrix,its kernel and the camera screw from a visual feature.

- *Interfaces* :It provides functions for the user interface with menus, files, graphics windows. There is also a interface with a real mobile robot in open loop via a serial link.

For a given robotic task, the simulation needs the following different steps:

- Description by the user of the specific 3D scene which can be used in the simulation.

- Positioning the robot to its initial position with regard to the scene frame.

- At this step, the user can verify its application by moving the robot in open loop in this environnment and by visualizing images from the camera during its displacement.

- With respect to the task, the user has to select the different visual features which will be used by the robot to perform its task. In a next version, the task specification can be done directly in terms of desired virtual linkage.

- The simulator computes the interaction matrix (L^T), its kernel (T^*_{CT}) of which a base represents the camera motions leading to no change in the image.

- After having selected a particular camera motion consistent with the visual constraints, we can simulate the robot's behaviour or execute directly the computed trajectory in open loop on our mobile robot. Graphic simulation results are presented in Figure 3 and 4 and show that the robot "manoeuvres" in way to satisfy a nominal trajectory of the camera.

The first part of the software written validates the kinematic model and confirms that any camera motions are possible without being limited by the cart nonholonomic constraints. The second part of this tool is dedicated to the study of visual interactions between the mobile robot and its environment.

In order to illustrate the use of the simulator, we will give a simple example where the robot task is defined by two successive visual servoing sub-tasks. We set the robot to an initial position with respect to an environment constituted by several segments (see figure 2). Its initial arm configuration is :($q_1 = \pi, q_2 = -\frac{pi}{4}, q_3 = 0$). The robotic task consists of wall following. Knowing the size of the skirting board of the room, we can use it to compute the interaction matrix taking as features the two corresponding parallel lines. So its returns:

$$L1^T = \begin{pmatrix} -0.156323 & 0.236455 & -0.0509507 & 0.0991288 & -0.149942 & -1 \\ -0.240955 & 0.364468 & -0.0785346 & -0.861134 & -0.569307 & 0 \\ -0.158875 & 0.265158 & -0.0766261 & 0.127409 & -0.212643 & -1 \\ -0.243381 & 0.406197 & -0.117384 & -0.910519 & -0.545556 & 0 \end{pmatrix} \tag{24}$$

The matrix kernel, after reduction by taking in account that $\Theta = 0$, gives the camera trajectory which do not modify the feature in the image. This trajectory is defined by the camera screw :

$$T1^*_{CT} = (1.118667, 1, 1, 0, 0, 0) \tag{25}$$

By applying this screw during 4 secondes (see Figure3) in simulation, we illustrate that this "kernel trajectory" is a translation along the wall. We can see that the robot "manoeuvres" in order to satisfy the camera motion without using heuristic method.

The second vision based subtask consist to run around the corner of the room. For doing that , we use the vertical line formed by the intersection of the walls. After a half-turn of the arm and moving q_2 on zero, we have now a vertical segment on the image. Taking this segment as a new feature for visual servoing, the simulator can compute the interactions matrix corresponding to this new task (knowing the height of this segment). Here again the reduced kernel has only one dimension and we can deduce a camera screw :

$$T2^*_{CT} = (-1, 0.0000148184, 0.0145816, 0, 0.126032, 0) \tag{26}$$

In this second sub-task when this screw is applied on camera, it rotates around the vertical segment (see Figure 4).

More elaborated tasks would be defined by seeking other features for visual servoing.

6 Experimental results

In a parallel way to the simulation developments, we work towards a testbed to allow us to validate reactive approaches using visual servoing concepts in real experiments. This testbed uses a cart-like mobile robot carrying a 3 d.o.f arm with a hand eye camera. The computer architecture in board is built around a VME Bus which can accept until 20 VME boards. An external link is provided to a Sun4 Workstation by means of an Ethernet link. Image processings performances are a central point in a vision based control approach because the stability is very dependant of the sampling rate of the measurement. We are developping a new architecture [MAR 91] based on DSP devices which should be able to perform in real time (less than 30ms) specific processings well adapted to the visual servoing approach. This architecture is based on the concept of *active windows* and allows to track until 30 simultaneous windows per image with different sizes and can execute on each window different kinds of processing in real time.

7 Conclusion and Future Trends

The results, yet obtained in simulation, are cheerful and it seems that visual servoing approaches coud be a good mean to implement reactive tasks in mobile robotics. More efforts remain to be done to validate this approach in an effective way,(ie with a real robot in a real 3D environment). In the next future, we plan to devellop our work in the following directions :

- The work of implementation on hardware and software must be finished in order to verify experimentally mobile robot simulation results.
- To extend the functionnalities of the simulator, it seems interesting to integrate it in a more general one named SIMPARC ([AST 92]). This integration will allow to simulate hardware, algorithms for vision based control and robot dynamics. With this extension, we will dispose of a complete simulator wich allows us to write more efficiently algorithms for the real robot.
- The simulation examples given, show how to do a direct relation between a mobile robot task and the visual features. Simple robot's behaviour with respect to its environment can be specified in terms of : to follow a wall, to go in front of a door, to turn in respect of a wall angle, and so on... This work has to be pursued and completed in order to dispose of a set of elementary mobile robot tasks described by visual features to track. Combining these simple vision oriented tasks, it will be possible to realise more complex ones.

The first application will concern automatic exploration and cartography of an unknowm room. In a next future, we should hope to address the problem of programming complex robotic tasks in term of a succession of *virtual linkage*.

Figure 1 : Robot Description : Frames and Notations

Figure 2 : Initial Location of the Robot

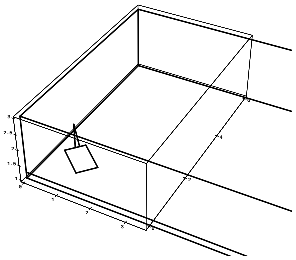


Figure 3 : The Robot Follows the Wall

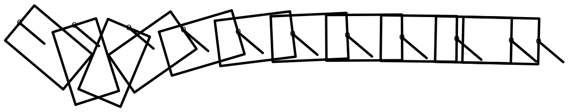
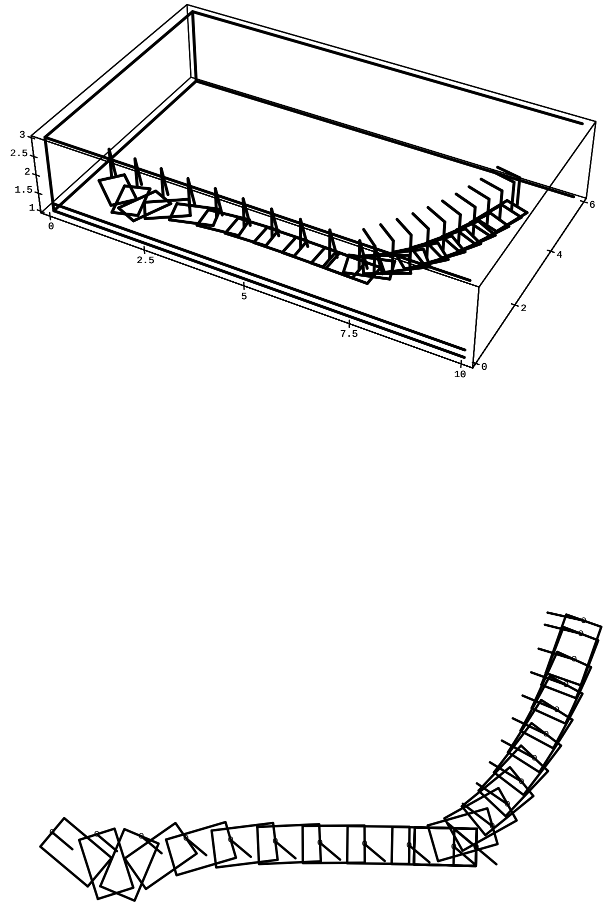


Figure 4 : The Robot Rotates Around a Vertical Segment



References

[AST 92] C. Astraudo, J. J. Borrelly :*Simulation of Multiprocessor Robot Controllers*, IEEE Int. Conf. on Robotics and Automation,Nice, May 1992.

[BAR 89] J. Barakan, J. C. Latombe :*On nonholonomic mobile robots and optimal maneuvering*, Revue d'Intelligence Artificielle, Vol 3, n2, Ed Hermes, pp. 77-103, 1989.

[BRO 86] R. Brooks : *A robust layered control system for a mobile robot*, IEEE Journal of robotics and Automation, vol 2, n1, pp 14-33, 1986.

[CHA 90] F. Chaumette : *La relation vision-commande: théorie et applications à des tâches robotiques*, Ph-D Thesis, University of Rennes I, France, July 1990.

[CHA 91] F. Chaumette, P. Rives, B. Espiau : *positioning of a robot with respect to an object, tracking it and estimating its velocity by visual servoing*, Proceedings IEEE Int. Conf. on robotics and automation, Sacramento, pp 2248-2253, April 1991.

[ESP 87] B. Espiau :*Sensory-based Control: Robustness issues and modelling techniques. Application to proximity sensing*, NATO Workshop on Kinematic and Dynamic Issues in Sensor Based Control, Italy, October 1987.

[FED 91] J. Feddema, G. Lee :*Weighted selection of image features for resolved rate visual feedback control*, IEEE transactions on robotics and automation, vol 7, n1, pp 31-47, April 1991.

[LAU 86] J. P. Laumond :*Feasible trajectories for mobile robots with kinematic and environnment constraints*, Int. Conf. on intelligent autonomous systems, Amsterdam, pp. 346-354, 1986.

[MAR 91] P. Martinet, P. Rives, P. Fickinger, J.J Borrelly : *Parallel Architecture for Visual Servoing Applications*, 91'CAMP Workshop on Computer Architecture for Machine Perception, Paris, Dec. 1991

[PIS 92] R. Pissard-Gibollet, P. Rives : *Asservissement visuel appliqué à un robot mobile: Etat de l'art et modélisation cinématique*, INRIA Report, n. 1577, 1922.

[RIV 89] P. Rives, F. Chaumette, B. Espiau: *Visual Servoing Based on a Task Function Approach*, First International Symposium on Experimental Robotics, Montréal, Canada, June 1989.

[SAM 91] C. Samson, K. Ait-Abderrahim : *Feedback control of a nonholonomic wheeled cart in cartesian space*, Proceedings IEEE Int. Conf. on robotics and automation, Sacramento, pp 1136-1141, April 1991.

[SAM 90] C. Samson, B. Espiau, M. Le Borgne : *Robot control: the task function appproach*, Oxford University Press, 1990.

[WEI 84] L. E. Weiss : *Dynamic visual servo control of robots. An adaptive image based approach.* Carnegie Mellon Technical Report, CMU-RI-TR-84-16, April 1984.