

VISUAL SERVO CONTROL OF A MOBILE ROBOT USING OMNIDIRECTIONAL VISION

Jun Okamoto Jr.[†] and Valdir Grassi Jr.[‡]

[†] University of São Paulo, Escola Politécnica
Dept. of Mechatronics and Mechanical Systems Engineering,
Av. Prof. Mello Moraes, 2231,
05508-900 São Paulo, SP, Brasil
email: jokamoto@usp.br

[‡] University of São Paulo, Escola Politécnica
Dept. of Mechatronics and Mechanical Systems Engineering,
Av. Prof. Mello Moraes, 2231,
05508-900 São Paulo, SP, Brasil
email: vgrassi@usp.br

Abstract

To achieve a fully autonomous robot several individual sensory based tasks can be composed to accomplish a complete autonomous mission. These sensory based tasks can be distance detection with ultrasonic sensors to avoiding collision or stereo vision acquisition and processing for environment mapping. In this context, the sensory based task that is subject of this paper is a moving target following in real time based with an omnidirectional vision system supplying data for a visual servo controlled mobile robot. This paper shows some background information on omnidirectional vision system, object tracking and visual servo control. Then the implemented solution for performing this specific task is presented and the experimental results are shown and discussed.

1 Introduction

Fully autonomous robots should be able to complete complex missions without the interference, or decisions, from human operators. To achieve this goal, many subtasks are necessary to be completed autonomously mainly based on sensory information and with specific decision making processes within this subtask. The composition of the mission would be an assembly of all these subtasks. Many sensory based subtasks are related to local goals within a complete mission. For instance, detecting distances from a mobile robot to nearby objects with ultrasonic sensors can aid the planning of the robot's movement avoiding collisions. In this context, the subtask of interest here is the tracking of a moving object with a mobile robot based on an omnidirectional vision system feedback. This task is accomplished with real time data acquisition from the omnidirectional vision system and the control of the mobile robot in a visual servo model.

The use of conventional vision system for mobile robot control requires the camera to be positioned towards a centre of attention in order to keep a subject in the field of view so that the robot can be visually servo controlled. This could be done by a two servo mechanisms to move the camera up and down and sideways independent of the robot's movement or the robot could move the camera towards the subject. In any of these cases some effort should be expended to the position of the camera besides the movement of the robot to accomplish its task.

By the other hand, a vision system that can provide a 360° field of view can avoid camera movements and keep a subject on the field of view all the times independent of the subject's position around the robot. This type of vision system is known as omnidirectional vision system. This system can be achieved by the use of a fish eye lens and a CCD camera or by the use of a convex mirror and a conventional lens and CCD camera. The second type is used in this work to control a mobile robot in a closed loop visual guided control system to accomplish the task of tracking a select target in real-time.

The task programmed into the robot is to keep the distance to the target constant. The robot can manoeuvre in flat surfaces only and the position of the robot is determined by dead reckoning only. Although it is not necessary to keep a record of the distances moved by the robot in this application.

This paper shows the results for the implementation of this system and for an experiment that shows the results obtained with the developed system and the application of the object tracking technique called SSD inside of a servo visual control loop that controls the movement of a mobile robot in real time.

2 Omnidirectional Vision

Omnidirectional vision systems can provide an image with 360° field of view of the robot environment. There are different approaches for obtaining omnidirectional images at video rate as can be seen in (Yagi 1999). A commonly used approach is to combine a convex mirror and a conventional lens CCD camera. Mirror shapes can be spherical, conic, parabolic, hyperbolic, or specifically designed shapes to achieve images with a desired propriety (Yagi et al 1994), (Nayar 1997), (Svoboda et al 1997), (Chahl and Srinivasan 1997), (Hicks and Bajcsy 1999).

We use a hyperbolic mirror in front of a conventional lens CCD camera to provide images to visual servo control the mobile robot. The hyperbolic mirror has the single projection centre propriety. For this reason, it is possible to generate from the acquired image a perspective or a panoramic image free of distortion (Peri and Nayar 1997), (Baker and Nayar 1998).

The mirror shape was designed using the methodology proposed in (Svoboda et al. 1997). Considering a coordinate system centred in the focus F' of the hyperbole, as shown on Figure 1, the mirror shape can be defined in terms of the mirror parameters a and b by the following equation:

$$y = \sqrt{a^2 \cdot \left(1 + \frac{x^2}{b^2}\right)} - \sqrt{a^2 + b^2} \quad (1)$$

In Figure 1, $e = \sqrt{a^2 + b^2}$ represents the mirror eccentricity, h is the distance between the mirror top and the camera centre, α is the vertical angle of view, r_{top} is the x coordinate on the mirror top, and $y_{top} = h - 2e$ is the y coordinate of the mirror top.

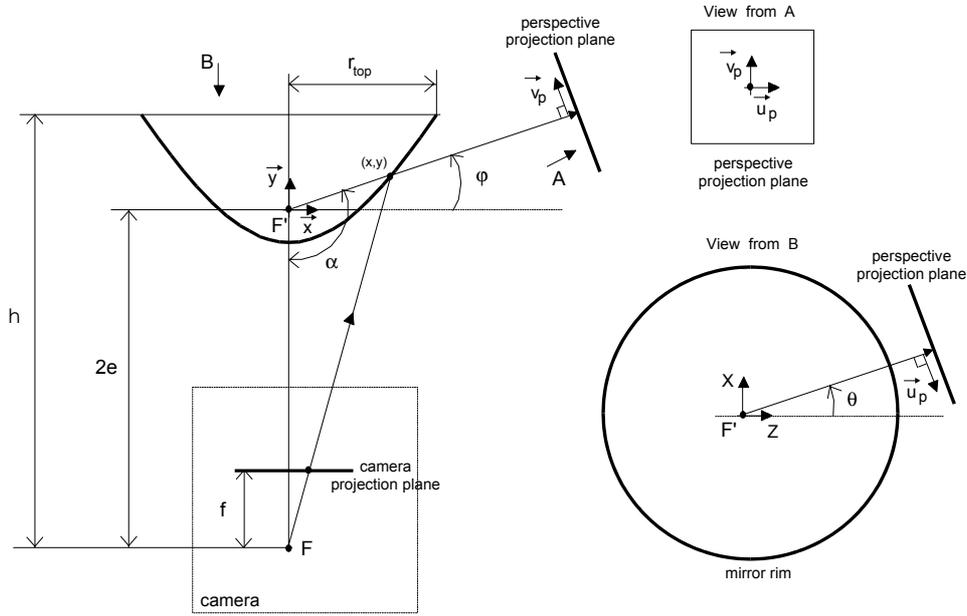


Figure 1. Hyperbolic mirror geometry

The mirror was designed considering a camera with focal distance $f = 12\text{mm}$, approximated pixel size in the CDD of $t_{pixel} = 0.01\text{mm}$, and acquired images of 640×480 pixels. The rim of the mirror is projected on the image in a circle of radius $r_{pixel} = 240\text{pixels}$. Choosing $r_{top} = 20\text{mm}$ and the relation $a = 2b$, the resulting parameters of the mirror were $h = 100\text{mm}$, $a = 39.29$, $b = 19.65$, $2e = 87.86$, with the maximum vertical angle of view of $\alpha = 121.3^\circ$. Consequently, the resulting equation that describes the hyperbolic mirror is the following:

$$y = 39,3 \sqrt{\left(1 + \frac{x^2}{186}\right)} - 43,9 \quad (2)$$

The mirror was then machined in an ultra-precision lathe from a cylinder of aluminium. There was no need to polish the surface after the machining since the ultra-precision lathe gave to the aluminium surface the mirror properties desired.

In the omnidirectional system built the camera was held vertically with its optical axis aligned to the centre of the mirror. Also, the mirror was mounted on top of a transparent acrylic cylinder so that no support structure for the mirror obstructs the image.

In order to create a perspective image from the image sensed by the omnidirectional system, the propriety of single centre of projection is used. In a system with single centre of projection all the light rays responsible for the formation of the image in the camera

when reflected by the mirror intersect at a single point. For the hyperbolic mirror this point is the focus of the hyperbole F' shown in the Figure 1.

A perspective projection plane can be defined by $(f_p, \theta_0, \varphi_0)$, with f_p being a distance from the focus of the hyperbole F' and the defined plane, θ_0 being the azimuth angle, and φ_0 being the zenith angle measured as shown in Figure 1. The pixels coordinate on the perspective projection plane (u_p, v_p) can be related to the direction (θ, φ) of a light ray that forms the image using the following two equations:

$$\tan \varphi = \frac{f_p \sin \varphi_0 + v_p \cos \varphi_0}{f_p \cos \varphi_0} \quad (3)$$

$$\tan \theta = \frac{(f_p \cos \varphi_0 - v_p \sin \varphi_0) \cdot \sin \theta_0 - u_p \cos \theta_0}{(f_p \cos \varphi_0 - v_p \sin \varphi_0) \cdot \cos \theta_0 + u_p \sin \theta_0} \quad (4)$$

Given a determined angle φ , it is possible to find the point (x, y) in the mirror surface where the light ray is reflected. This is done solving the mirror equation (2) for $y = x \cdot \tan \varphi$. Now the pixel (u, v) in the image acquired by the camera that corresponds to the light ray defined by (θ, φ) can be found using the following relations:

$$u = \frac{x \cdot (2e + y_{top}) \cdot r_{pixel}}{(x \cdot \tan \varphi + 2e)} \cos \theta \quad (5)$$

$$v = \frac{x \cdot (2e + y_{top}) \cdot r_{pixel}}{(x \cdot \tan \varphi + 2e)} \sin \theta \quad (6)$$

Despite the point (x, y) is found using an iteration method which is time consuming, it is yet possible to create perspective images in real time using lookup tables. So it is created a lookup table relating all the pixels in the image acquired by the camera with the corresponding angles (θ, φ) . So defined a perspective plane $(f_p, \theta_0, \varphi_0)$ each pixel (u_p, v_p) in the created perspective image can be related at real time to the correspond pixel (u, v) in the image acquired using the lookup table and the equations (5) and (6). In those equations f_p is a measure given in pixels.

3 Visual Tracking

The image obtained from the vision system must be processed in order to extracting target feature parameters used to calculate the error vector in the visual servo control loop. However, processing the entire image is time consuming. As only the small pixel region that corresponds to the target is useful to compute the feature parameters, a visual tracking of the target can be used. So the real time visual tracking of the target continually updates the coordinates of the target and make the computation of the target features parameters possible to be done in real time by limiting the processing region (Hager 1995).

The target used in this work consists of a white panel with an "x" drawn on it. Since the target is known the image is processed using an algorithm that operates in two steps.

Firstly, the target region is located in the entire image with a segmentation and region-based algorithm. This first step is not done in real time and does not need to be repeated once the target was found in the image. The coordinates of the target are then passed to the algorithm's second step that performs the visual tracking in real time.

We choose a sum-of-squared differences (SSD) region-based tracking algorithm as described in (Hager and Toyama 1998) to do the real time visual tracking of the target. This algorithm assumes that the deformation suffered by the target image region between the time t and $t + \Delta t$ can be modelled by an affine transformation composed of a matrix A and a translation d . The transformation is found minimizing the following function where X represent a pixel coordinate of the image region and $I(X, t)$ is the intensity of the pixel:

$$O(A, d) = \sum_X (I(AX + d, t + \Delta t) - I(X, t))^2 \quad (7)$$

The affine transformation model does not hold for the omnidirectional image acquired directly by the vision system. For this reason this tracking algorithm must be applied to a perspective image defined by the parameters $(f_p, \theta_0, \varphi_0)$ as shown in the Section 2.

As the target moves, the target translation computed by the tracking algorithm can be used to change the viewing direction of the perspective plane. This is done in order to prevent that the target get out of the field of view of the created perspective plane resulting in track lost. The new direction (θ_n, φ_n) can be computed by $\varphi_n = \varphi + \arctan(dv_p / f)$ and $\theta_n = \theta + \arctan(du_p / f)$ where dv_p and du_p are respectively the vertical and horizontal translation of the target centre. Updating continuously the viewing direction makes the target be always in the centre of the perspective image. The effect is the same as if we had a virtual perspective camera in a pan/tilt mechanism but with its focus always located in the focus of the hyperbole. This approach was also used by (Chang and Hebert 1998).

4 Visual Servoing

The visual servoing task implemented in this work consist of control a mobile robot to track a select target in real-time. For this purpose it was used the image-based visual servoing approach using an interaction matrix as described in (Chaumette et al. 1991), (Hutchinson et al 1996) and (Carvalho et al. 2000). The interaction matrix L^T relates the motion of a 3D frame attached to a perspective camera with the motion of the target features parameters vector.

$$\dot{s} = L^T \cdot T_c \quad (8)$$

The target feature parameters s chose are the coordinates in the perspective image of the four corners that defines the target region being tracked by the visual tracking algorithm, $s = [u_{p1}, u_{p2}, u_{p3}, u_{p4}, v_{p1}, v_{p2}, v_{p3}, v_{p4}]^T$, where (u_{p1}, v_{p1}) is the coordinate in pixels of the first point and so on. The motion of the perspective camera frame can be represented as $T_c = [{}^c\dot{x}_c, {}^c\dot{y}_c, {}^c\dot{z}_c, {}^c\dot{w}_x, {}^c\dot{w}_y, {}^c\dot{w}_z]^T$.

The visual tracking algorithm is initialised with a square target region with width dimension of $2a$ positioned in the centre of the image. As the robot must follow the target, the desired reference features can be represented by $s_r = [-a, a, a, -a, -a, a, a, -a]^T$, where a is measured in pixels. The interaction matrix computed for s_r is given by (Chaumette et al. 1991).

$$L^T|_{s=s_r} = \begin{bmatrix} -f_p/cz_t & 0 & -a/cz_t & -a^2/f_p & (-f_p^2 - a^2)/f_p & a \\ -f_p/cz_t & 0 & a/cz_t & a^2/f_p & (-f_p^2 - a^2)/f_p & a \\ -f_p/cz_t & 0 & a/cz_t & -a^2/f_p & (-f_p^2 - a^2)/f_p & -a \\ -f_p/cz_t & 0 & -a/cz_t & a^2/f_p & (-f_p^2 - a^2)/f_p & -a \\ 0 & -f_p/cz_t & a/cz_t & (f_p^2 + a^2)/f_p & a^2/f_p & a \\ 0 & -f_p/cz_t & a/cz_t & (f_p^2 + a^2)/f_p & -a^2/f_p & -a \\ 0 & -f_p/cz_t & -a/cz_t & (f_p^2 + a^2)/f_p & a^2/f_p & -a \\ 0 & -f_p/cz_t & -a/cz_t & (f_p^2 + a^2)/f_p & -a^2/f_p & a \end{bmatrix} \quad (9)$$

In the matrix above, f_p is equivalent to a focal distance but given in pixels and used to create the perspective image plane, cz_t is the distance of the target from the camera. It is possible to use an approximated value for cz_t or if it is unknown, it can be set equal to one and the visual servo control will still converge (Chaumette et al. 1991).

The control law computed in terms of the camera motion T_c can be expressed by $T_c = -\lambda \cdot e$ with $\lambda > 0$. The error is computed by $e = C \cdot (s - s_r)$ where C is the pseudo-inverse of the interaction matrix, s is the current visual feature parameters, and s_r is the reference visual feature parameters. The current visual feature parameters are given as a result of each interaction of the visual tracking algorithm as seen in Section 3.

Since the camera is mounted on the robot, it is possible to relate the camera motion vector T_c with the control parameters of the robot finding a control law equation adequate to our application. We use a syncro-drive mobile robot for which the control parameters are $U = [\dot{z}_b, \dot{\theta}_b]^T$, where \dot{z}_b is the translation speed, and $\dot{\theta}_b$ is the steering speed of the robot.

The component ${}^c\dot{y}_c$ of the camera motion frame is zero since the camera and the robot moves only in the horizontal plane. To simplify the implemented visual servoing task it was considered also that ${}^c\omega_x$ and ${}^c\omega_z$ are also zero. This implies that ϕ_0 , which is one of the angles used to define the image plane, remains constant and also that the target does not suffers rotation. The rotation speed ${}^c\omega_y$ of the perspective camera is software controlled by the visual tracking algorithm. So the remaining components of the camera frame motion $[{}^c\dot{x}_c, {}^c\dot{z}_c]^T$ are related to the control parameters $U = [\dot{z}_b, \dot{\theta}_b]^T$ using the Jacobian matrix J (Carvalho et al. 2000).

$$\begin{bmatrix} {}^c\dot{x}_c \\ {}^c\dot{z}_c \end{bmatrix} = J * U, \text{ with } J = \begin{bmatrix} -\sin\theta_r & -cz_t \\ -\cos\theta_r & cx_t \end{bmatrix} \quad (10)$$

In the equation above θ_r is the difference between the direction of the movement θ_b and the direction of the perspective camera θ_c , $\theta_r = \theta_b - \theta_c$. The distance ${}^c x_t$ is the distance of the target from the z -axis of the camera, and can be estimated using the value of ${}^c z_t$ and the focal distance. Although to our application as the visual tracking keeps the target centred in the image, the distance ${}^c x_t$ is estimated as being zero.

The inverse of the Jacobian J is given by:

$$J^{-1} = \begin{bmatrix} J_1 & J_2 \\ J_3 & J_4 \end{bmatrix} = \frac{1}{{}^c x_t \cdot \sin \theta_r - {}^c z_t \cdot \cos \theta_r} \begin{bmatrix} {}^c x_t & -{}^c z_t \\ -\cos \theta_r & -\sin \theta_r \end{bmatrix} \quad (11)$$

Finally, the control law can be written as the following:

$$U = -\lambda \cdot N \cdot C \cdot (s - s_r), \text{ where } N = \begin{bmatrix} J_1 & 0 & J_2 & 0 & 0 & 0 \\ J_3 & 0 & J_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (12)$$

As the control of the camera direction was implemented to keep the target always centred in the image as described in Section 3, we chose to control the steering angle θ_b in a manner that the value of θ_r remains always zero. Consequently, the forward speed is controlled by the scaling information provided by the visual tracking algorithm. Although another approach could be used, or another visual servoing task could be defined, for example, docking.

When following a target using a proportional control law as described above, while the target is moving with a constant velocity, there will be a constant error $s - s_r$ which will cease when the target stop. This error can be suppressed if a proportional derivative control where used (Chaumette et al. 1999). So the control law in terms of the camera motion would be $T_c = -\lambda \cdot e - \lambda_2 \partial e / \partial t$.

5 Experimental Results

The hardware set-up used to implement the visual servoing task is basically a syncro-drive mobile robot with an omnidirectional vision system, and a Pentium 4 1.4GHz PC workstation running linux. The mobile robot has on-board a low level controller connected to a PC104 computer by RS232c, a video radio transmitter and an Ethernet radio.

Figure 2 shows the mobile robot with the omnidirectional vision system mounted on top of it, and Figure 3 shows an omnidirectional image acquired with the system. Figure 4 shows a diagram of the used hardware set-up.

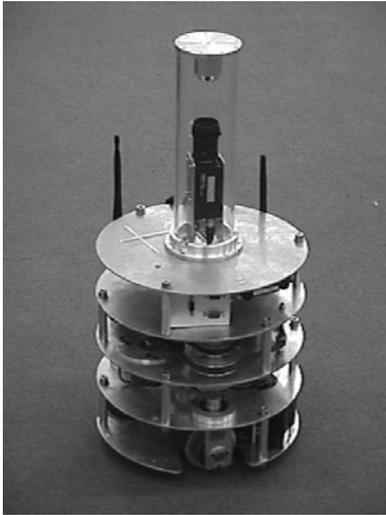


Figure 2. Mobile robot with omnidirectional vision system.

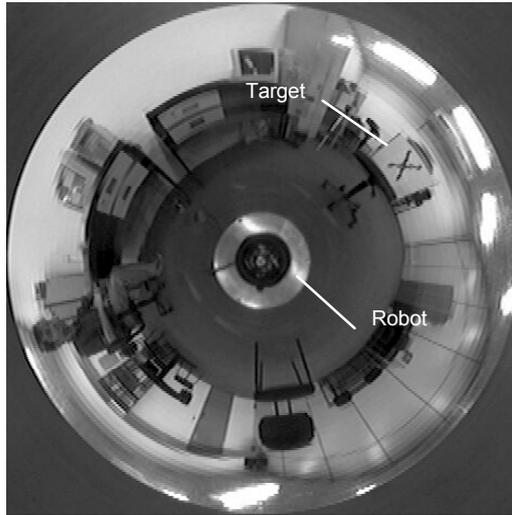


Figure 3. Ominidirectional image acquired with the system.

The image acquired by the CCD camera is transmitted by a radio link to the linux workstation that locates the pre-defined target and executes the visual servo control loop. Then the calculated control parameters of the robot are sent by an Ethernet link to the robot's PC104 computer that sends the commands to the low level controller.

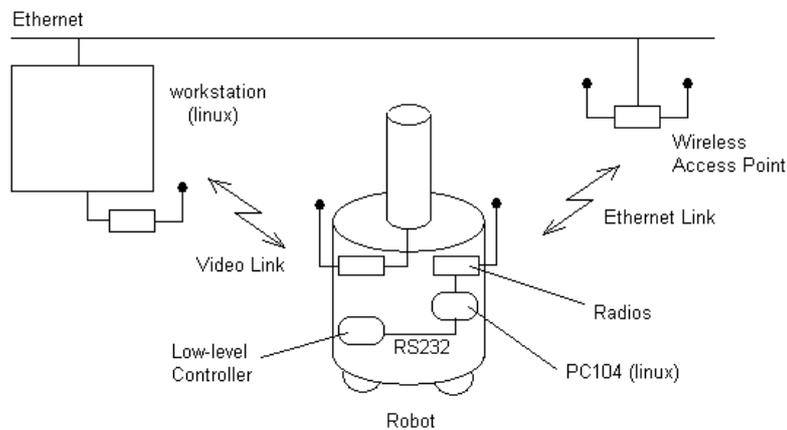


Figure 4. Hardware set-up

With this hardware configuration it was possible run the visual servo control loop at a rate of 33 ms, without any frame loss, with the low level controller of the robot working at a rate of 7.5 ms. The experiment was done by first initialising the control with the target at the distance of one meter from the robot, as shown in Figure 5. The target was then moved 60 cm away from the robot in a straight line and then stopped.

Figure 6 shows the results for the error signal and the forward speed control signal for this experiment. The graph on the left shows a small error signal while the robot is stopped. The error signal increases when the target start to move and get smaller as the

robot approaches the target. When the target stop moving the error signal reduces to approximately ± 2 pixels. This is about the resolution for the velocity control internal loop. The graph on the right shows the profile of the velocity of the robot. The behaviour of this graph follows the error signal and shows a maximum speed of 170 mm/s. As the maximum velocity that could be achieved by the robot control algorithm is approximately 350 mm/s the target could have moved faster than the tracking algorithm and the robot could still follow it in real time.

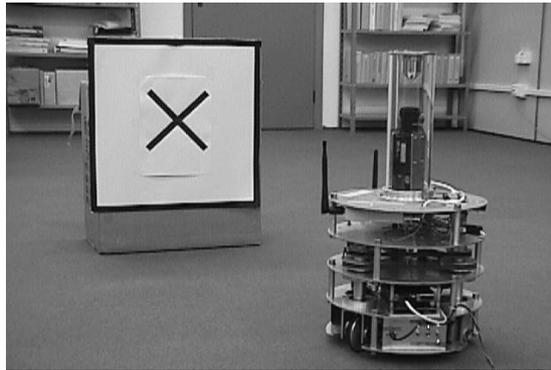


Figure 5. Target to be followed by the mobile robot

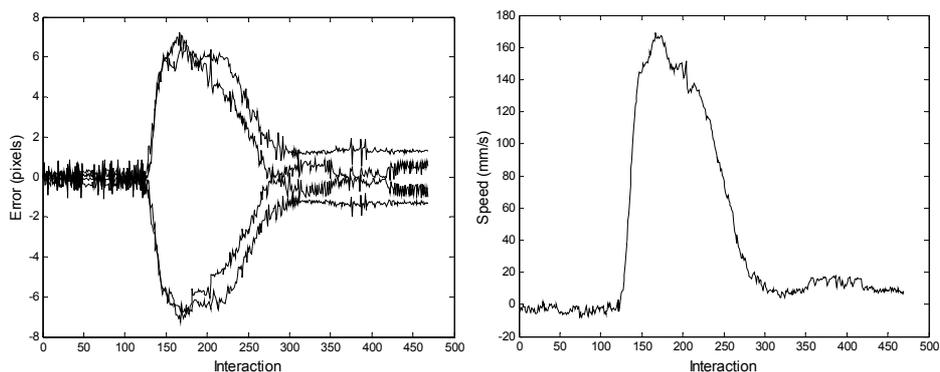


Figure 6. Error signal (left) and forward speed control signal (right) for the test done

6 Conclusion

This paper presented the development of a mobile robot system guided by an omnidirectional vision system to achieve a target following task with a visual servoing control structure. The developed omnidirectional vision system was presented showing its appropriate characteristics for the task. Then a background information on visual tracking and visual servoing was presented, showing the changes in the conventional algorithms that were necessary for the use of the omnidirectional vision system. The experimental results were presented to show that the task could be performed with the omnidirectional vision system in a visual servo control loop in real time. A change in the controller is planned soon to minimize the tracking error as treated by (Chaumette et al. 1999).

7 Acknowledgments

The authors would like to acknowledge the support of FAPESP, under grant # 1998/9096-9 for the construction of the infrastructure necessary to develop this research project, under grant # 1999/12610-9, for the financial support of a researcher; and to the RECOPE/MANET/FINEP project, under grant # 7797093500, for some important infrastructure and material used in this research project. The authors also would like to thank Prof. Arthur Vieira Porto from the Nucleus of Advanced Manufacture (NUMA) in the Dept. of Mechanical Engineering of the São Carlos School of Engineering of USP for helping machining the hyperbolic mirror for the omnidirectional vision system. And would like to thank the undergraduate students Fabricio M. Cunha and Giovana Breitschaft for the great help on the project.

References

- Baker, S., Nayar, S. K. (1998), *A Theory of Catadioptric Image Formation*, Proceedings of the 6th International Conference on Computer Vision, Bombay, India, pp. 35-42.
- Carvalho, J. R. H.; Rives, P.; Santa-Bárbara, A.; Bueno, S. S. (2000), *Visual servo control of a class of mobile robot*. Proc. of IEEE International Conference on Control Applications, Anchorage, Alaska, USA, pp.1-6.
- Chahl, J.S., Srinivasan, M.V. (1997), *Reflective Surfaces for Panoramic Imaging*, Applied Optics, **36(31)**, pp. 8275-8285.
- Chang, P., Hebert, M. (1998), *Omni-directional Visual Servoing for Human-Robot interaction*, Proc. of IEEE/RSJ International Conference on Intelligent Robotic Systems (IROS), Victoria, B.C., Canada.
- Chaumette, F., Rives, P., Espiau, B. (1991), *Positioning of a Robot to an Object, Tracking it and Estimating its Velocity by Visual Servoing*, Proc. of IEEE International Conference on Robotics and Automation (ICRA), Sacramento, California, pp.2248-2253.
- Hager, G. & Toyama, K. (1998), *The 'Xvision' system: A general purpose substrate for real-time vision applications*. Computer Vision and Image Understanding, **69(1)**, pp.23-27.
- Hicks, A., Bajcsy, R. (1999), *Reflective Surfaces as Computational Sensors*, Workshop on Perception for Mobile Agents at CVPR'99, Fort Collins, Colorado, USA.
- Hutchinson, R., Hager G. and Corke, P. (1996), *A tutorial on visual servoing*, IEEE Trans. Robotics and Automation **12(5)**, pp. 651-670.
- Nayar, S.K. (1997), *Catadioptric Omnidirectional Camera*, Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 482-488.
- Peri, V.N., Nayar, S.K., 1997, *Generation of Perspective and Panoramic Video from Omnidirectional Video*, Proc. of DARPA Image Understanding Workshop, pp. 243-245.
- Svoboda, T., Pajdla, T., Hlavac, V. (1997), *Central Panoramic Cameras: Geometry and Design*, Research Report K335/97/147, Czech Technical University, Faculty of Electrical Engineering, Center for Machine Perception, available at <ftp://cmp.felk.cvut.cz/pub/cmp/articles/svoboda/TR-K335-97-147.ps.gz>
- Yagi, Y. (1999), *Omnidirectional Sensing and Its Applications*, IEICE Transactions on Information and Systems, **E82-D(3)**, pp. 568-579.
- Yagi, Y., Kawato, S. & Tsuji, S. (1994), *Real-Time Omnidirectional Image Sensor (COPIS) for Vision-Guided Navigation*, IEEE Transactions on Robotics and Automation, **10(1)**, pp. 11-22.