

# Visual servo control of mobile robots

Thomas Blanken    0638483

DC 2010.061

**Bachelor Final Project**

Dynamics and Control  
Department of Mechanical Engineering  
Eindhoven University of Technology

Supervisor:  
dr. Dragan Kostić

October 28, 2010

## ABSTRACT

This project addresses the problem of homography-based visual servo control for mobile robots with nonholonomic motion constraints. The sensor system is a vision system consisting of a fixed camera mounted to the robot. The target is defined by an image taken at the desired position prior to mobile robot navigation. The control law drives the robot from the initial position to the desired one by processing image information extracted from the current and target images. The output of the system consists of homography elements, which are estimated using matched feature points.

Camera calibration is incorporated to estimate the epipolar geometry of the scene in order to define the trajectories.

The definition of the reference trajectories proved to have disadvantages regarding the inherent undefined value of epipoles when camera positions are identical. Experimental results are presented and discussed with simulation results to show the performance of the approach.

# CONTENTS

<b>1 - INTRODUCTION .....</b>	<b>3</b>
<b>2 - PROJECT DESCRIPTION.....</b>	<b>4</b>
<b>3 - VISION ALGORITHMS FOR HOMOGRAPHY BASED VISUAL SERVO CONTROL .....</b>	<b>5</b>
HOMOGRAPHY ESTIMATION .....	5
CAMERA CALIBRATION .....	6
<b>4 - VISUAL SERVO CONTROL OF A NONHOLONOMIC MOBILE ROBOT .....</b>	<b>9</b>
MOTION MODEL OF THE MOBILE ROBOT.....	9
INPUT-OUTPUT LINEARIZATION.....	10
CONTROL LAW.....	11
REFERENCE SIGNAL TRAJECTORIES .....	11
<b>5 - SIMULATIONS AND EXPERIMENTS.....</b>	<b>14</b>
SIMULATION RESULTS .....	14
SETUP AND CONTROL IMPLEMENTATION .....	15
EXPERIMENTAL RESULTS.....	16
<b>6 - CONCLUSIONS.....</b>	<b>19</b>
<b>7 - APPENDIX A .....</b>	<b>20</b>
<b>8 - REFERENCES .....</b>	<b>26</b>

# 1 - INTRODUCTION

Nowadays, most robots operate in work environments that can be adapted to suit the robot. Robots have had less impact in applications where the environments can not be controlled. This limitation is for most due to the lack of sensory capabilities in commercial available robots. Vision is an attractive sensor modality because it provides a lot of information about the environment. It also does not need a controlled work environment for the robot to operate in. However it comes with larger than average complexity and computational cost.

Visual servo control is the process of controlling a system in closed loop by using information captured by camera systems. It uses techniques from areas as image processing, computer vision, real-time computing, dynamics and control theory. The camera system can be mounted to the robot, or it can be fixed to the environment and having a view of the robot.

A current application of visual servo control is a parking assist for cars, in which a camera system is able to estimate the position of obstacles around the car and aid the driver in steering the car in a parking space. Another application is manufacturing robots, with tasks such as grasping and mating parts.

## 2 - PROJECT DESCRIPTION

The objectives of the project are to gain competences with computer programs for visual servo control and to achieve mobile robot navigation in experiments. The work of [1] and [2] is used as a guideline, adopting the control law and reference trajectories. The method for visual servo control is adopted from chapter 12 of [8].

The mobile robot used is an E-puck [7], on which a fixed monocular camera is attached.

The control task uses the idea of homing: at every iteration, an image is taken by the camera and is compared to the target image. The target image is taken prior to robot navigation and its camera pose represents the desired final position of the robot. The homography between target and current image is estimated using Direct Linear Transform of matched feature points. This homography estimation does not need any depth estimation or measurements of the scene. The control law leads the robot from the initial pose to the target pose by means of reference trajectories in terms of homography. The algorithms and control law are implemented in MATLAB.

### 3 - VISION ALGORITHMS FOR HOMOGRAPHY BASED VISUAL SERVO CONTROL

#### Homography estimation

Homography is a planar projective transformation between two images. A mapping from  $P^2 \rightarrow P^2$  is a homography if and only if there exists a nonsingular 3-by-3 matrix  $\mathbf{H}$  such that, for any point in  $P^2$  represented by a column of coordinates  $\mathbf{x}$ , its mapped point is equal to  $\mathbf{H}\mathbf{x}$ . This is illustrated in figure 3.1. Hence, the homography of two identical images is the identity matrix. The homography can be scaled with any nonzero number without changing the projective transformation. The matrix is therefore a homogeneous matrix because only the ratios of the elements are important: homography has eight degrees of freedom. Note that the third element of any image point  $\mathbf{x}$  equals 1 because they all lie on the plane of the image.

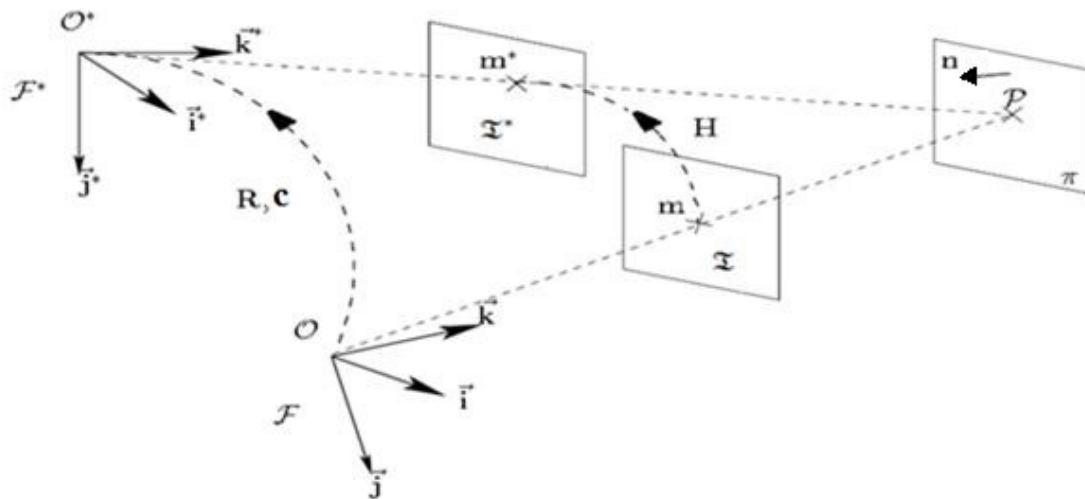


Figure 3.1. The configuration and homography relation between two images of a planar scene.

A homography between two images can be expressed in the internal camera matrix and the relative configuration between the images as:

$$\mathbf{H} = \mathbf{K}\mathbf{R}\left(\mathbf{I} + \mathbf{c}\frac{\mathbf{n}^T}{d}\right)\mathbf{K}^{-1} \quad (3.1)$$

where  $\mathbf{c}$  is the column of translation vector coordinates between the two camera frames,  $\mathbf{n}$  is the column of vector coordinates normal to the plane of the scene and  $d$  is the distance between the plane of the scene and the reference frame.  $\mathbf{K}$  is the internal camera matrix and  $\mathbf{R}$  is the rotation matrix describing the rotation from the first to the second camera coordinate frame:

$$\mathbf{K} = \begin{bmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{R} = \begin{bmatrix} \cos(\phi) & 0 & \sin(\phi) \\ 0 & 1 & 0 \\ -\sin(\phi) & 0 & \cos(\phi) \end{bmatrix} \quad (3.2)$$

where  $\alpha_x$  and  $\alpha_y$  are the focal lengths in terms of pixel dimensions,  $x_0$  and  $y_0$  are the principal point coordinates of the image in pixel dimensions and  $s$  is a skew coefficient. It describes the angle between sides of a pixel and it equals zero for orthogonal pixels, which is usually the case.  $\phi$  is the rotation angle between the two camera frames. In [6] and chapter 11 of [8] more can be found about homography and its relation to internal camera matrix and relative configuration.

Because the relative configuration and internal camera matrix are not known, (that is, without camera calibration) the homography has to be extracted from two images directly. How do we estimate homography from two images?

First, feature points are extracted from each image using an algorithm called Scale Invariant Feature Transform (sift). These feature points in both images are then matched. In [9] and [10] more detailed information about sift and matching feature points is provided.

After matched pairs of feature points are found, the homography is computed using Direct Linear Transform. The expressions of the Direct Linear Transform can be found in [3] and/or chapter 4.1 of [6]. At least four pairs of matched points are needed since the homography has eight degrees of freedom and a matched pair of feature points constraints two degrees of freedom. Any larger amount of pairs generally requires a singular value decomposition to compute the homography.

An accurate homography can only be found between images of a planar scene, because homography preserves collinearity of points in the scene. For robust homography estimation, a planar scene with sufficient feature points is obtained by attaching newspaper to a wall. Images are acquired from the camera using the MATLAB Image Acquisition Toolbox. This toolbox is not available for MATLAB under Linux.

## Camera calibration

The necessity for camera calibration is discussed in this section. The next chapter mentions that epipoles are needed to distinguish between two cases of reference trajectories. The target epipole is also used to compute a certain configuration angle, which is used to determine the reference trajectories. In Appendix A of [2] more information is provided about epipolar geometry.

The fundamental matrix, describing the epipolar geometry, can be computed by solving a linear set of equations, as seen in [6] and Appendix B of [2]. The fundamental matrix satisfies the following condition for each corresponding point  $\mathbf{x} \leftrightarrow \mathbf{x}'$  in two images:

$$\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0 \quad (3.4)$$

This set of linear equations can be found from all pairs of matched points. The fundamental matrix can then be computed using singular value decomposition. The current and target epipoles, denoted by  $\mathbf{e}_C$  and  $\mathbf{e}_T$ , are the left and right null space of the fundamental matrix respectively [6]:

$$\mathbf{F}^T \mathbf{e}_C = \mathbf{0} \quad (3.5)$$

$$\mathbf{F} \mathbf{e}_T = \mathbf{0} \quad (3.6)$$

These epipoles consist of a column of pixel coordinates. These coordinates indicate the position of the epipoles on the target and current image in terms of their pixel column, row and depth. The third coordinate (depth) should equal 1 because the epipoles lie on the image plane. To ensure this the epipoles are scaled by dividing by the third element.

This approach of computing epipoles does not require depth estimation or any measure of the scene.

During tests it has been seen that the epipoles computed in this way are not correct. They have the right magnitude, but turn out to have signs opposite to what they should be. This only occurs for configuration that should yield  $e_{CX} * e_{TX} > 0$ , where  $e_{CX}$  and  $e_{TX}$  are the coordinates of the pixel column of the epipoles. Clearly, the sign of the coordinates of the epipoles can not be estimated well.

Another way of extracting the fundamental matrix is via camera calibration. Calibration lets the system determine the relation between what appears on the image and where it is located in the 3D environment using a set of images. It is the process of extracting the cameras internal and external matrices using some information about the environment, like the size of a printed pattern visible to the camera. In chapter 4 of [2] it is assumed that the visual servo control problem does not need depth estimation or a 3D measure of the scene. With this approach only the expression for the trajectories will use a 3D measure of the scene.

The camera internal matrix is defined in (3.2) and external matrix is defined as  $[\mathbf{R} | \mathbf{t}]$  where  $\mathbf{R}$  and  $\mathbf{t}$  are the rotation matrix and translation column of vector coordinates between the camera coordinate frame and the world coordinate frame. The perspective projection matrix  $\mathbf{P}$  is defined as  $\mathbf{P} = \mathbf{K}[\mathbf{R} | \mathbf{t}]$ . The fundamental matrix of two images can be extracted using the perspective projection matrices of both images [5] [6].

A calibration toolbox for MATLAB is used [4]. A drawback of camera calibration is the large computational cost involved. For this matter, calibration is performed every fourth iteration, starting on the first iteration. The other iterations use the last computed angle  $\psi$ , thus relying on past information in order to have a higher sample frequency.

For our visual servo problem, a printed checkerboard pattern is added to the background scene as the calibration rig, the piece of information about the environment. The size of the squares is known and the toolbox functions are used



to compute the fundamental matrix between the target and the current image. The toolbox is adapted to estimate the checkerboard corners using the homography estimation of the current and target image. The corner locations on the target image are manually extracted prior to the robot navigation. The exact corner location is then calculated using a corner finder.

Because the repetitive nature of the checkerboard pattern causes the feature point matching algorithm to produce false matches, all feature points of the target image that lie on the checkerboard pattern are removed prior to robot navigation. This ensures all matching uses feature points from the environment excluding the checkerboard. In figure 3.2, feature points are extracted and the feature points on the calibration rig are excluded from matching.

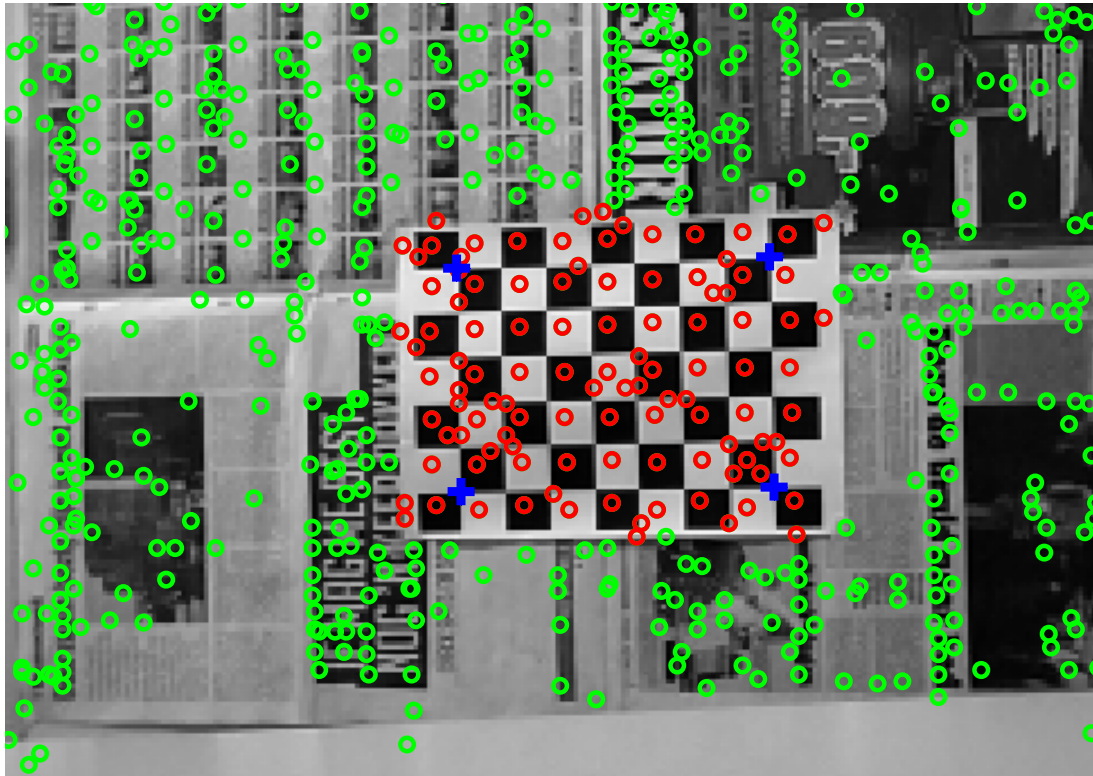


Figure 3.2. The target image including feature points suitable for matching (green) and not suitable for matching (red) because they are on the calibration rig. The estimated corner locations are shown as blue crosses.

## 4 - VISUAL SERVO CONTROL OF A NONHOLONOMIC MOBILE ROBOT

To have smooth and converging mobile robot navigation, the control law must be derived and reference trajectories must be chosen. To derive a suitable control law, the input-output behavior of the system must be modeled and a suitable error system must be defined.

### Motion model of the mobile robot

The state and output equation of a motion system can be expressed as:

$$\frac{d\mathbf{x}}{dt} = f(\mathbf{x}, u) \quad (4.1)$$

$$\mathbf{y} = h(\mathbf{x}) \quad (4.2)$$

The robot configuration space is the x-z plane as can be seen in figure 4.1. The state column is therefore defined as  $\mathbf{x} = [x \ z \ \phi]^T$ . These coordinates are defined with respect to the world coordinate frame. The robot coordinate frame at the target configuration is chosen to be coincident with the world coordinate frame, without loss of generality.

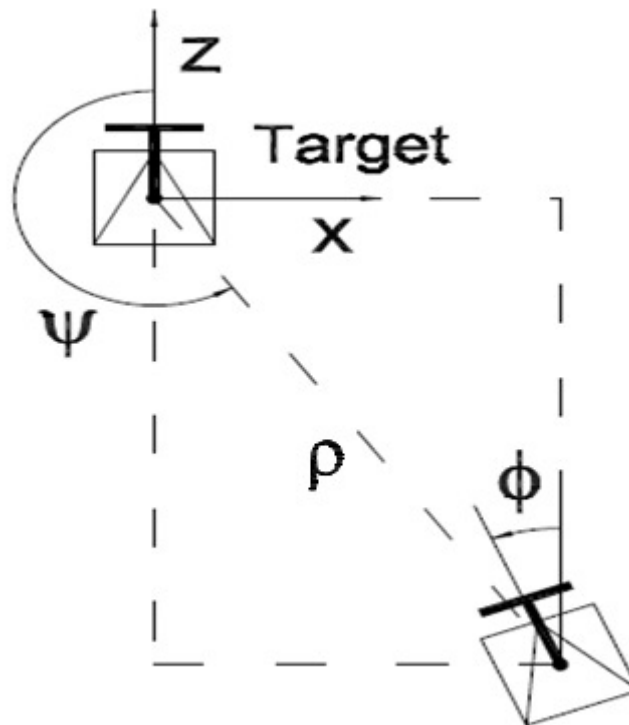


Figure 4.1. The coordinate system.

The mobile robot has a nonholonomic constraint due to the presence of its two wheels. This constraint means the robot can reach every configuration in its

configuration space, but it cannot move arbitrarily through its configuration space. Nonholonomic constraints are dependent on the time derivatives of the generalized coordinates. In this case, the lateral velocity of the robot equals zero. For the mobile robot used in this project the inputs are linear velocity  $v$  and angular velocity  $w$ . The state equation is written as:

$$\frac{d\mathbf{x}}{dt} = \begin{bmatrix} \dot{x} \\ \dot{z} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} -\sin(\phi) \\ \cos(\phi) \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} w \quad (4.3)$$

Next is to choose the output variables of the system. These must be homography elements in case of homography based control. Planar scenes with dominant  $n_z$  (planar scenes perpendicular to the camera axis) are detected more easily. Therefore  $h_{13}$  and  $h_{33}$  are chosen to form the output column. The output column is defined as:

$$\mathbf{y} = \begin{bmatrix} h_{13} \\ h_{33} \end{bmatrix} = \begin{bmatrix} \alpha_x \left( \sin(\phi) + (x \cos(\phi) + z \sin(\phi)) \frac{n_z}{d_\pi} \right) \\ \cos(\phi) + (-x \sin(\phi) + z \cos(\phi)) \frac{n_z}{d_\pi} \end{bmatrix} \quad (4.4)$$

The expressions for these homography elements are obtained by evaluating (3.1). The homography elements are then scaled by dividing by  $h_{22}$ . This is a reasonable scaling because the configuration space consists of the x-z plane so the y coordinate of the robot does not change. Therefore  $h_{22}$  is expected to be constant throughout robot navigation. Scaling the elements is necessary because the homography matrix lacks one degree of freedom to be fully constrained.

## Input-output linearization

Because the output of the system consists of homography elements, the problem of visual servo control becomes a tracking problem: the homography elements should follow reference trajectories in terms of homography. Because the model (4.3) and (4.4) gives a nonlinear relation between the inputs and the outputs, the homography elements (4.4) are linearized by differentiating the homography and are combined with the state equation (4.3) to obtain a linear relationship between inputs and outputs. This relationship can be expressed in matrix notation using a decoupling matrix  $\mathbf{L}$ :

$$\begin{bmatrix} \dot{h}_{13} \\ \dot{h}_{33} \end{bmatrix} = \begin{bmatrix} 0 & \alpha_x h_{33} \\ \frac{n_z}{d_\pi} & -\frac{h_{13}}{\alpha_x} \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} = \mathbf{L} \begin{bmatrix} v \\ w \end{bmatrix} \quad (4.5)$$

## Control law

The error and time derivative of the error are defined as:

$$\mathbf{e} = \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} h_{13}^R - h_{13} \\ h_{33}^R - h_{33} \end{bmatrix}, \quad \dot{\mathbf{e}} = \begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \end{bmatrix} = \begin{bmatrix} \dot{h}_{13}^R - \dot{h}_{13} \\ \dot{h}_{33}^R - \dot{h}_{33} \end{bmatrix} \quad (4.6)$$

Superscript 'R' stands for reference.

The input-output relation (4.5) is combined with the error definition (4.6) and rewritten to obtain the control law:

$$\begin{bmatrix} v \\ w \end{bmatrix} = \mathbf{L}^{-1} \begin{bmatrix} \dot{h}_{13}^R + k_{13}(h_{13}^R - h_{13}) \\ \dot{h}_{33}^R + k_{33}(h_{33}^R - h_{33}) \end{bmatrix} = \begin{bmatrix} \frac{h_{13}d_\pi}{\alpha_X^2 h_{33}n_Z} & \frac{d_\pi}{n_Z} \\ \frac{1}{\alpha_X h_{33}} & 0 \end{bmatrix} \begin{bmatrix} \dot{h}_{13}^R + k_{13}(h_{13}^R - h_{13}) \\ \dot{h}_{33}^R + k_{33}(h_{33}^R - h_{33}) \end{bmatrix} \quad (4.7)$$

Here,  $k_{13}$  and  $k_{33}$  are the control gains.

In section 5.3.2 of [2] it has been proven that the decoupling matrix is nonsingular in the physically feasible configuration space. This ensures a nonsingular control signal.

The error system differential equation for this control law is:

$$\dot{\mathbf{e}} + \mathbf{k}\mathbf{e} = \mathbf{0} \quad (4.8)$$

Where the control gain matrix is defined as:

$$\mathbf{k} = \begin{bmatrix} k_{13} & 0 \\ 0 & k_{33} \end{bmatrix} \quad (4.9)$$

For a positive definite control gain matrix, in this case positive  $k_{13}$  and  $k_{33}$ , both the error and the time derivative of the error converge to zero.

## Reference signal trajectories

The reference trajectories proposed by [1] are used in this project. The control law ensures the output column tracks the reference trajectories, providing convergence to the target configuration. The homography should resemble the identity matrix when the robot has reached the target configuration, hence the output column should resemble  $\mathbf{y} = [0 \ 1]^T$ . The reference trajectory reaches this value of the output column at  $t = T_2$ .

## Reference trajectory of $h_{13}$

First, a condition regarding the initial configurations has to be checked to choose one of the cases below. This condition is expressed in the target and current epipole. See Appendix B of [2] for a detailed description of epipolar geometry.

### Case 1

If  $e_{CX} * e_{TX} \leq 0$  the reference trajectory can be expressed in two steps.

$$1) \quad h_{13}^R(0 \leq t \leq T_2) = h_{13}(0) \frac{\psi(t)}{\psi(0)}$$

$$2) \quad h_{13}^R(T_2 < t < \infty) = 0$$

### Case 2

If  $e_{CX} * e_{TX} > 0$  the reference trajectory can be expressed in three steps.

$$1) \quad h_{13}^R(0 \leq t \leq T_1) = \frac{h_{13}(0) + h_{13}^R(T_1)}{2} + \frac{h_{13}(0) - h_{13}^R(T_1)}{2} \cos\left(\frac{\pi t}{T_1}\right)$$

$$2) \quad h_{13}^R(T_1 < t \leq T_2) = h_{13}(T_1) \frac{\psi(t)}{\psi(T_1)}$$

$$3) \quad h_{13}^R(T_2 < t < \infty) = 0$$

where  $h_{13}^R(T_1) = -\frac{2}{3}h_{13}(0)$  and  $0 < T_1 < T_2$ . Recall that  $e_{CX}$  and  $e_{TX}$  are defined in the paragraph ‘Camera calibration’ in chapter 3.  $\psi(t)$  is defined as the angle between the camera axis in the target coordinate frame and the line connecting the current and target camera poses, as seen in figure 4.I.

It is extracted from the epipolar geometry using  $\psi(t) = -\arctan\left(\frac{e_{TX}(t)}{\alpha_X}\right)$ .

The first step of case 2 is meant to turn the robot to the opposite side of the epipolar ray in  $T_1$ . From that point on, it is meant to provide a smooth motion towards the target configuration.

## Reference trajectory of $h_{33}$

The reference trajectory is expressed in two steps.

$$\begin{aligned} 1) \quad h_{33}^R(0 \leq t \leq T_2) &= \frac{h_{33}(0)+1}{2} + \frac{h_{33}(0)-1}{2} \cos\left(\frac{\pi t}{T_2}\right) \\ 2) \quad h_{33}^R(T_2 < t < \infty) &= 1 \end{aligned}$$

It has been discovered that, for certain initial configurations, the trajectories do not guarantee convergence of the robot to the target configuration. The right choice for the intermediate point  $h_{13}^R(T_1)$  is essential because it should provide a smooth further motion towards the target configuration, which is the objective of this step in the reference trajectory. The choice for  $h_{13}^R(T_1) = -\frac{2}{3}h_{13}(0)$  should lead the robot to turn to the opposite side of the epipolar ray. This objective can be denoted as  $\text{sgn}(e_{CX}(T_1)) = -\text{sgn}(e_{CX}(0))$ . However, this choice for the intermediate point is only valid under the assumption that  $h_{13} = 0$  for configurations that yield  $e_{CX} = 0$ . An expression for configurations that result in  $h_{13} = 0$  is:

$$\sin(\phi) + (x \cos(\phi) + z \sin(\phi)) \frac{n_Z}{d_\pi} = 0 \quad (4.10)$$

On the other hand, an expression for configurations that result in  $e_{CX} = 0$  is:

$$\phi = \psi - 180 = \arctan\left(\frac{x}{-z}\right) \quad (4.11)$$

By combining and rewriting (4.10) and (4.11) configurations that satisfy both  $e_{CX} = 0$  and  $h_{13} = 0$  are  $\mathbf{x} = [0 \quad z \quad 0]^T$ . This is generally not the case, so  $h_{13}^R(T_1) = -\frac{2}{3}h_{13}(0)$  is not a good choice.

This problem can be avoided by choosing a large (absolute) initial value of  $\phi$ , and choosing a larger  $h_{13}^R(T_1)$ , so that no initial configuration leading to the divergence of the robot to the target is chosen. In practice, this large value for  $\phi$  points out to be too large for the camera to have the scene and the calibration rig in sight, due to the limited view angle. This prevents the robot to navigate along this trajectory and no convergence to the target configuration was achieved.

Better trajectories should be chosen, possibly not depending on  $\psi$  and having a better choice for  $h_{13}^R(T_1)$ . This excludes the need for calibration during navigation. However this lies outside the scope of the project and the time that is available.

## 5 - SIMULATIONS AND EXPERIMENTS

In [1] and [2] simulations have shown the validity of the control law and trajectories for most initial configurations. Simulation results of [2] are shown in the next paragraph.

Experiments are required to show the performance and validity of both homography estimation and camera calibration. The setup, implementation and results are shown in the subsequent paragraphs.

### Simulation results

Simulation results for initial configuration  $[x \ z \ \phi]^T = [12 \ -40 \ 20]^T$

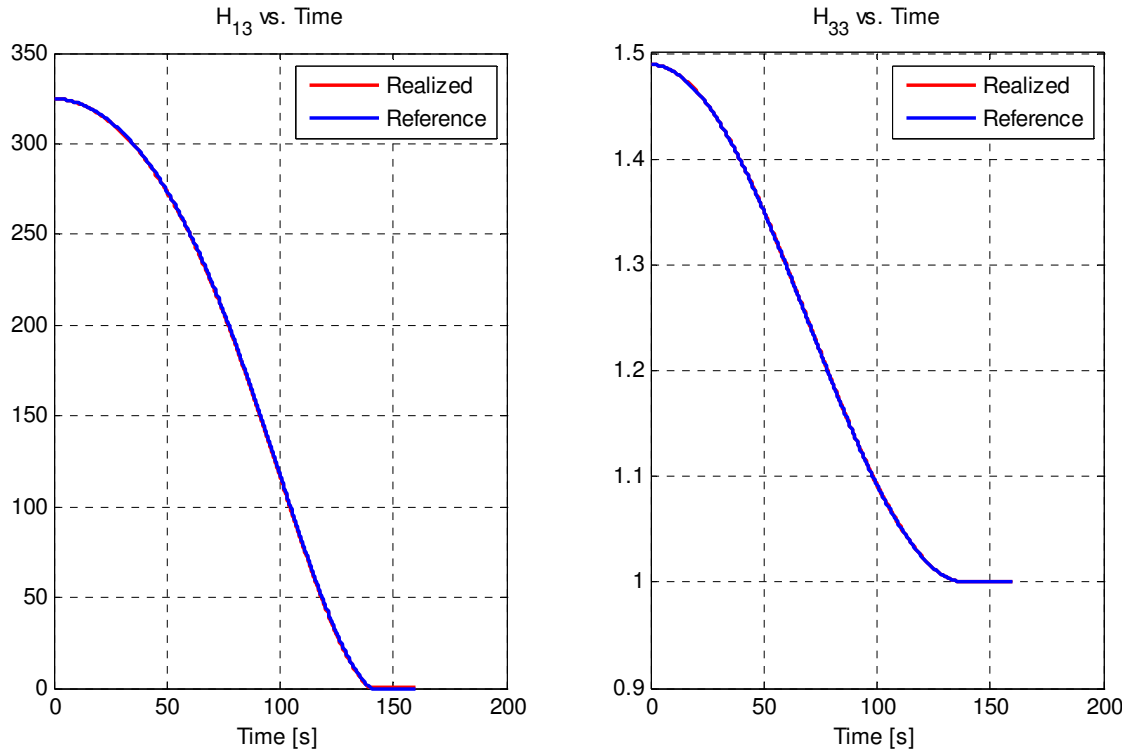


Figure 5.1. Simulated homography elements  $h_{13}$  and  $h_{33}$  as function of time. The realized homography elements and the reference trajectory nicely overlap.

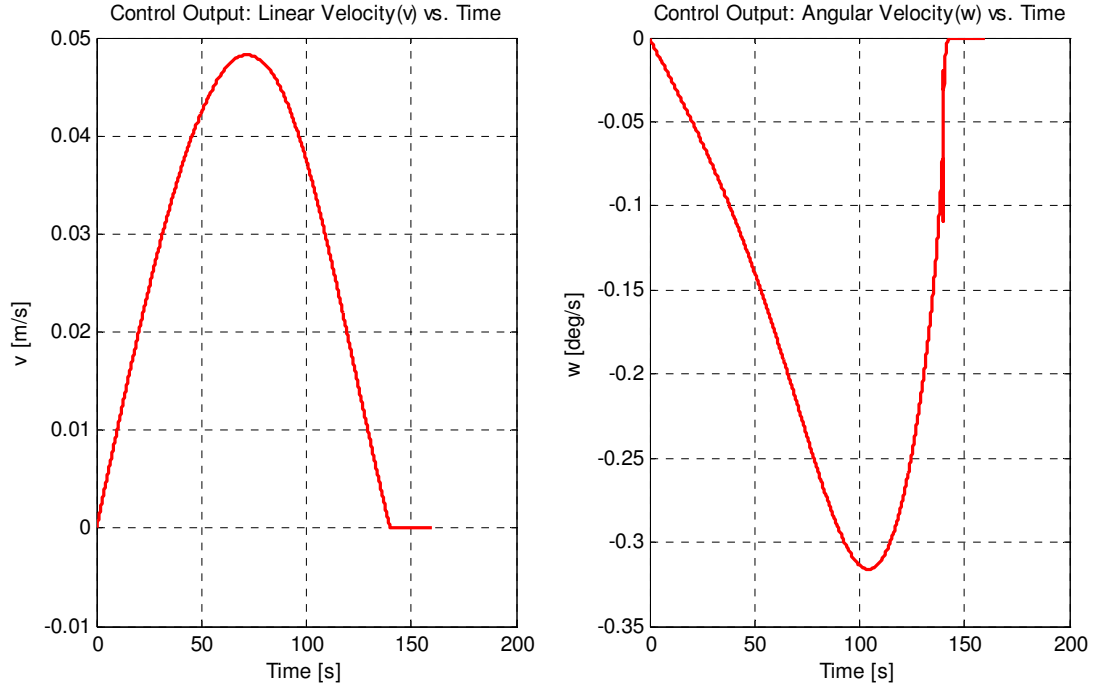


Figure 5.2. Simulated control output (plant input) as function of time.

## Setup and control implementation

Figure 5.3 shows the control scheme. The plant model has a grey rectangle around it to emphasize on its inputs, outputs and components.

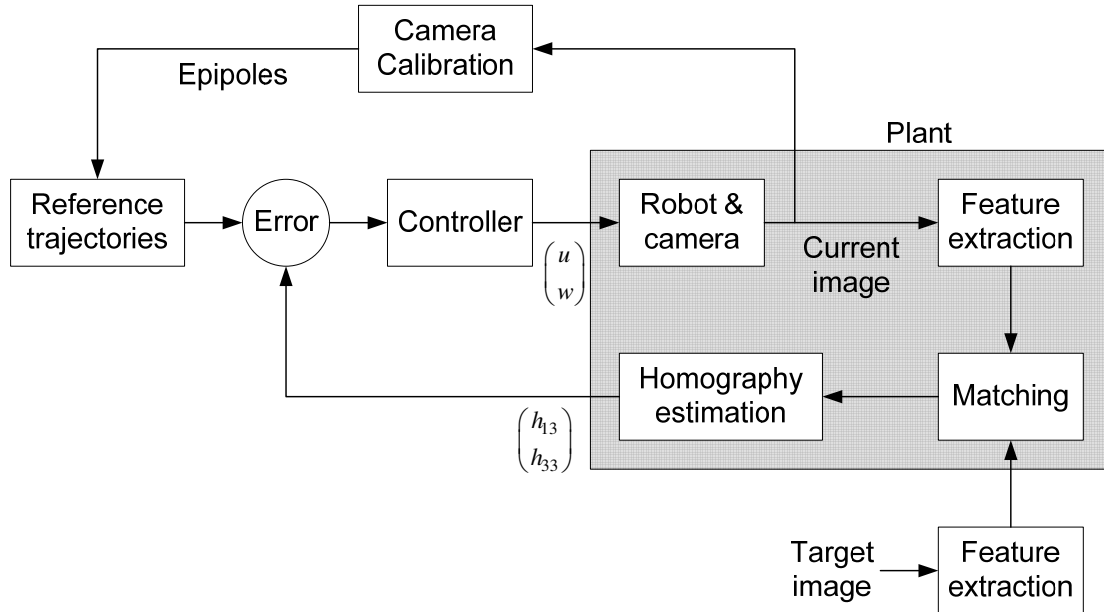


Figure 5.3. The homography based visual servo control scheme.

The experiments are carried out for different initial configurations. The distance of the planar scene to the target position is always 50 cm in positive z-direction. Recall that the target configuration is chosen to always be



$[x \ z \ \phi]^T = [0 \ 0 \ 0]^T$ . The control gains  $k_{13}$  and  $k_{33}$  are chosen to be 4 and 2 respectively by trial and error.

As can be seen in the next section, homography estimation comes with noise. Therefore the control replaces spikes of the homography matrix larger than a threshold value with the homography matrix of the previous iteration, yielding smoother navigation.

The reference trajectory of  $h_{13}$  becomes ill-conditioned when the robot configuration is close to the target configuration. This is caused by the fundamental matrix being nearly singular and resulting in epipoles moving to infinity. The robot is therefore stopped whenever the output column comes within a certain threshold value of the final reference signal for at least three consecutive iterations. The resulting final pose error regarding the minimum threshold value is within 5 cm in x and z direction and within  $2^\circ$  in  $\phi$ . Any smaller threshold value would cause the robot to diverge from the target configuration because the reference trajectory would diverge from the identity matrix. The cause of this problem was discussed in the previous chapter.

## Experimental results

In this section the results of physical robot navigation are shown.

Results for initial configuration  $[x \ z \ \phi]^T = [5 \ -45 \ 10]^T$  (measured in cm and degree respectively)

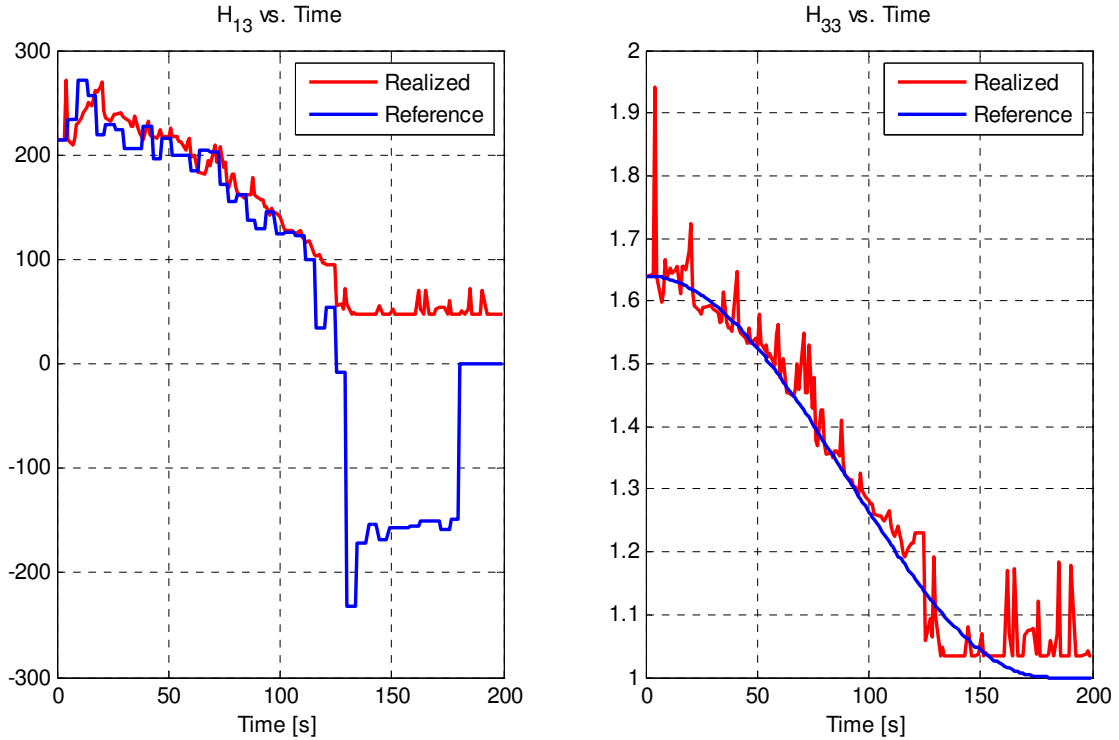


Figure 5.4. Homography elements  $h_{13}$  and  $h_{33}$  as function of time.

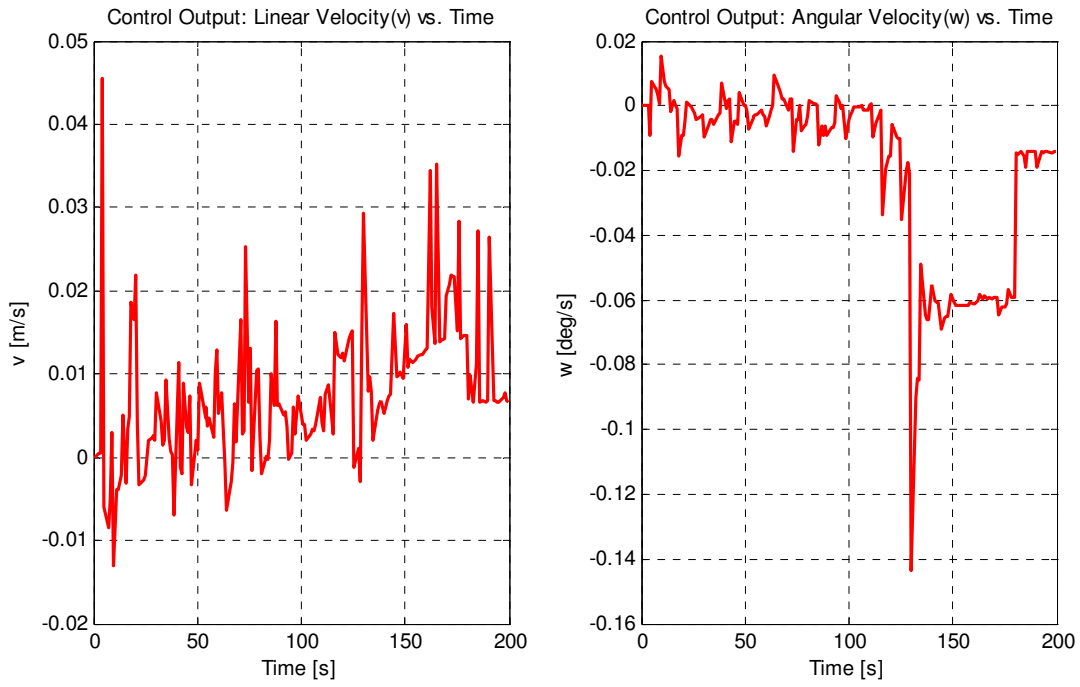


Figure 5.5. Control output (plant input) as function of time.

Results for initial configuration  $[x \ z \ \phi]^T = [5 \ -30 \ 20]^T$

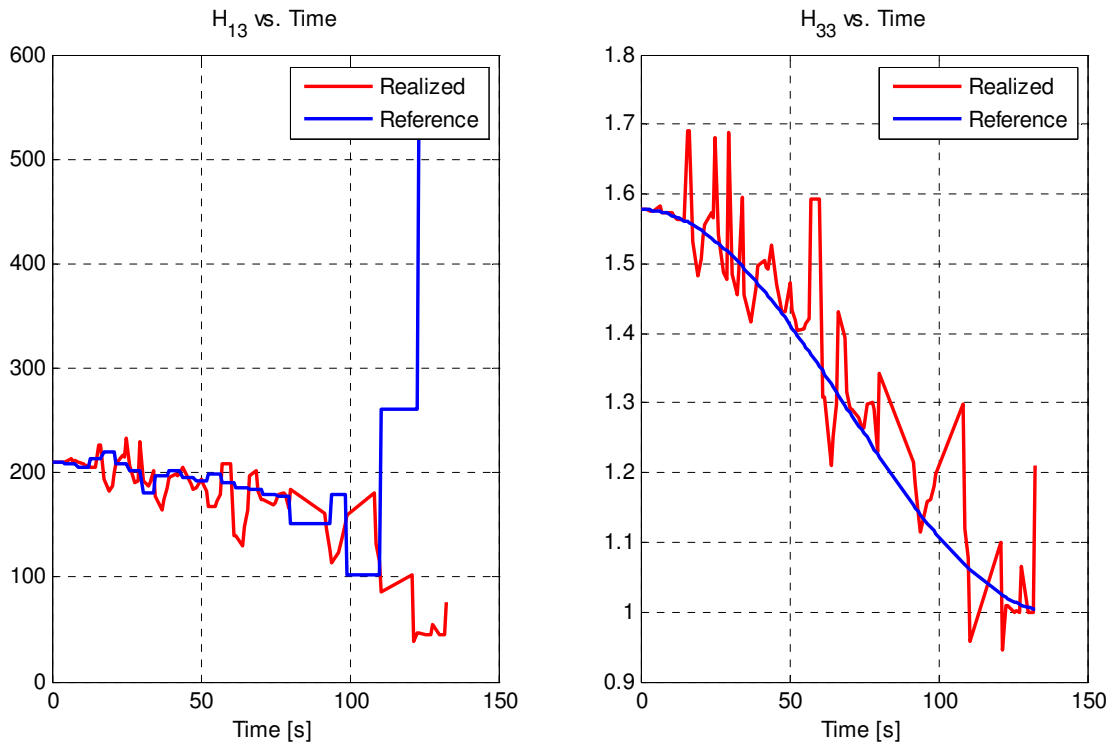


Figure 5.6. Homography elements  $h_{13}$  and  $h_{33}$  as function of time.

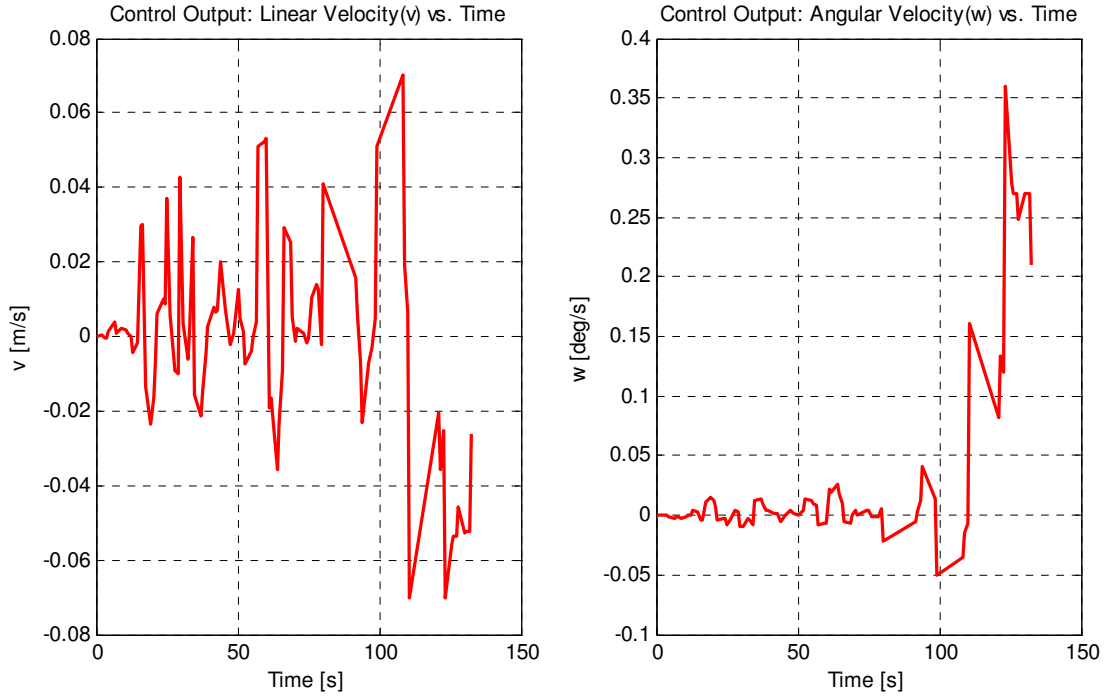


Figure 5.7. Control output (plant input) as function of time.

More experimental results can be found in Appendix A.

When comparing above experimental results with simulation results from [2], as seen in figures 5.1 and 5.2, it is obvious that the error signal is much higher in practice than it is in simulations. The fact that the homography elements have different initial values can be addressed to several factors. The arbitrary homography scaling, the estimated distance to target  $d$  and the vector normal to the scene plane  $\mathbf{n}$  might not be similar in reality and in simulations. However, estimations for these parameters do not influence the convergence to the target.

The average sample frequency is 1 Hz and the delay of the control system equals the time period of an iteration. This is caused by the high computational cost of calibration, feature point extraction and feature point matching. The result is that navigation to the target is achieved in about 200 iterations. The control output in experiments varies quite substantially, causing the robot to perform shaking motion while driving to the target pose. This is caused by noise in the homography estimation, noise in the epipole estimation, a low sample frequency and a high control delay.

## 6 - CONCLUSIONS

This goal of the project is to achieve mobile robot navigation using visual servo control methods, based on the work of [1] and [2]. A homography-based visual servo method is applied to a nonholonomic mobile robot. The control law is constructed upon the input-output linearization of the system. The outputs of the system are chosen among the homography elements and a set of reference trajectories for these controlled outputs are defined. The visual servo control problem is therefore converted into a tracking problem. The epipolar geometry is used to distinguish between certain initial configurations of the robot in defining the applied reference trajectory for that initial configuration. The target epipole is also used to express the trajectory of one of the output elements.

Experiments show that the control algorithm is robust and the convergence of the system is achieved up to a distance from the target. However there is substantial noise on homography estimation and calibration.

The visual control method used necessitates neither depth estimation nor any 3D measure of the scene. However, to compute the proposed reference trajectories, the computation of epipolar geometry is done via camera calibration, requiring a 3D measure of the scene. A drawback of calibration is the large computational cost involved. Epipole computation also has an inherent disadvantage that when the camera positions of both images get nearly identical, the epipoles move to infinity and this causes divergence of the trajectories and therefore the robot. The robot can therefore not drive to the target pose resulting in zero error.

The performance of the system is dependent on the reference trajectories of the homography elements since the problem is a tracking problem. The trajectories proposed by [1] are used in the project. The set of reference trajectories makes the robot converge towards the target in a smooth manner. However, the mobile robot can not always converge to the target pose.

It is analytically shown that there exist initial configurations for which the reference trajectories do not lead the robot to the target configuration. This is caused by a false assumption meant to define an intermediate point in the trajectory.

The reference trajectories can be improved in future works. One criterion could be the elimination of camera calibration, which implies that the trajectories must not be dependent on epipoles. As a result, no calibration rig image is needed in the scene and the computational cost of the control task is reduced. Second, the trajectories could be redesigned to function properly for every possible initial configuration. This means the trajectory must include valid definitions for intermediate trajectory points. Third, the control system could be implemented in C to function under Linux or the functionalities of the Image Acquisition Toolbox could be reproduced to work in MATLAB under Linux. This would speed up the computations and lead to faster mobile robot navigation.

## 7 - APPENDIX A

In this appendix additional experimental results are shown.

Results for initial configuration  $[x \ z \ \phi]^T = [5 \ -45 \ 10]^T$

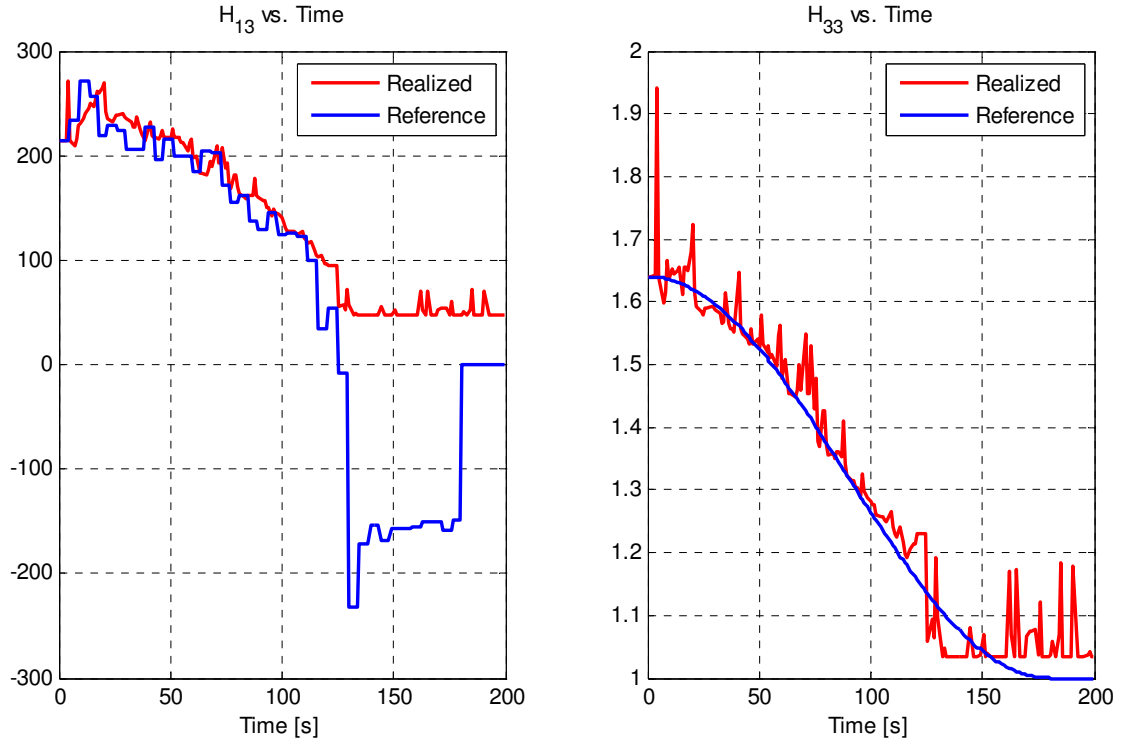


Figure 7.1. Homography elements  $h_{13}$  and  $h_{33}$  as function of time.

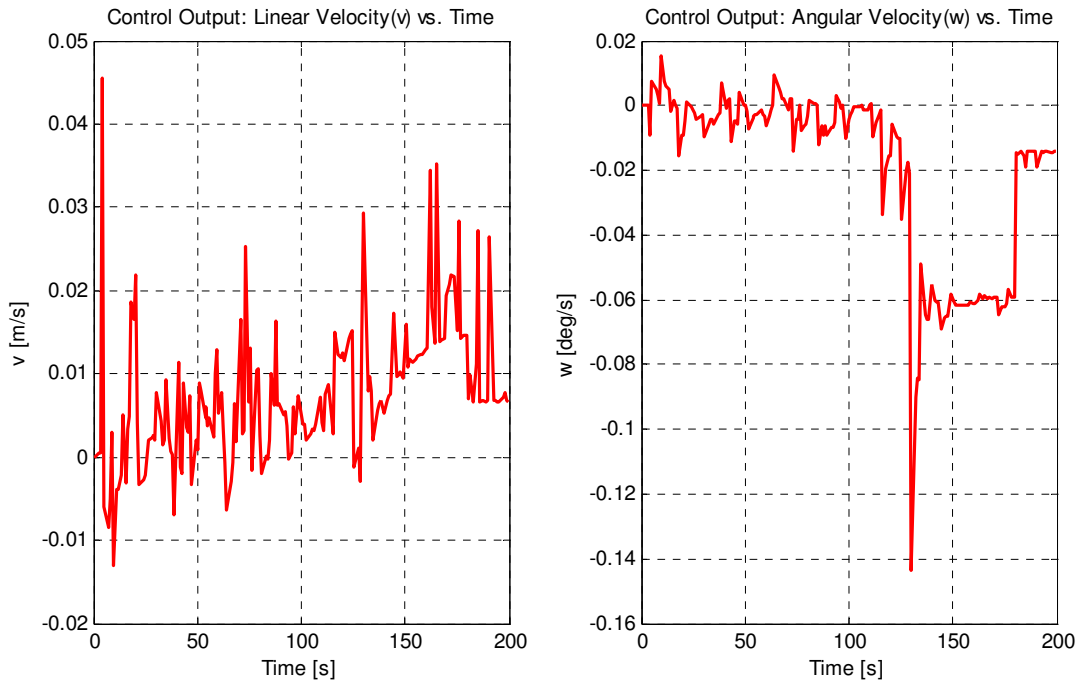


Figure 7.2. Control output (plant input) as function of time.

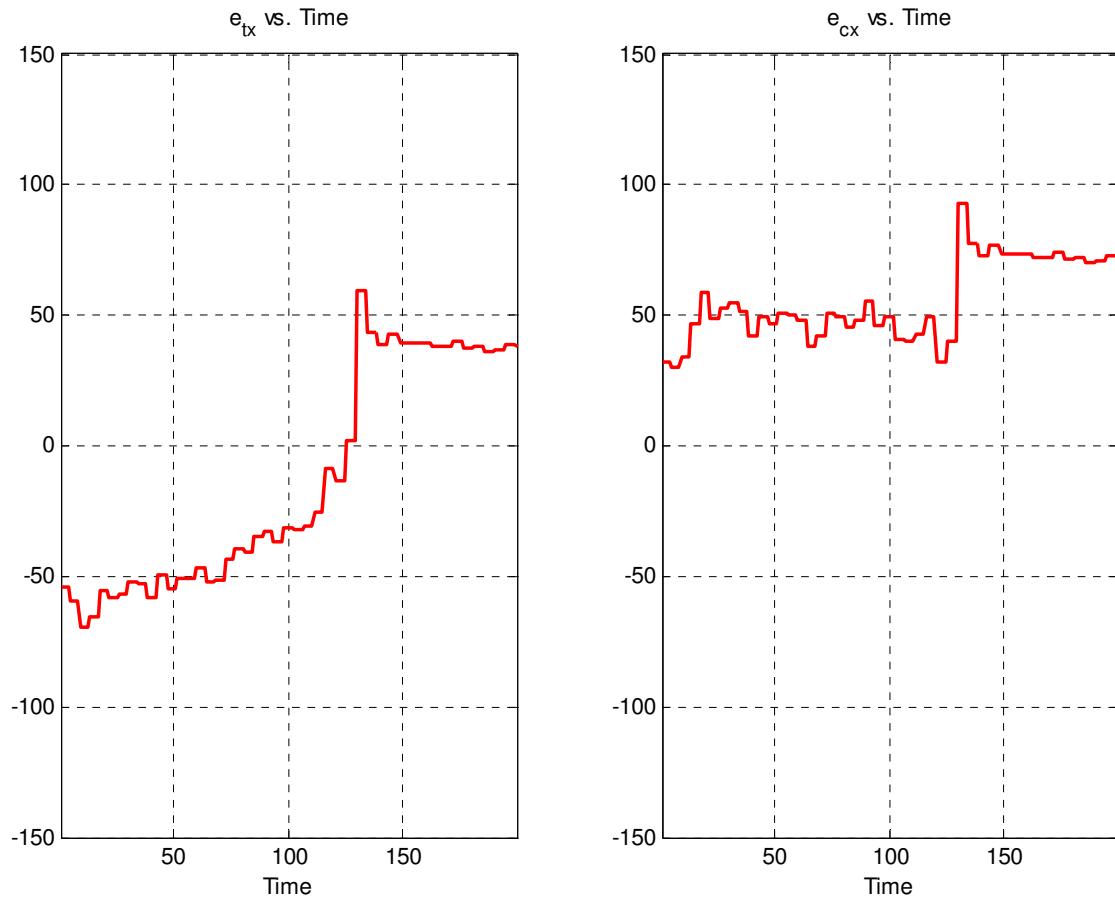


Figure 7.3. X-coordinates of target and current epipole as function of time.

Results for initial configuration  $[x \ z \ \phi]^T = [5 \ -50 \ 5]^T$

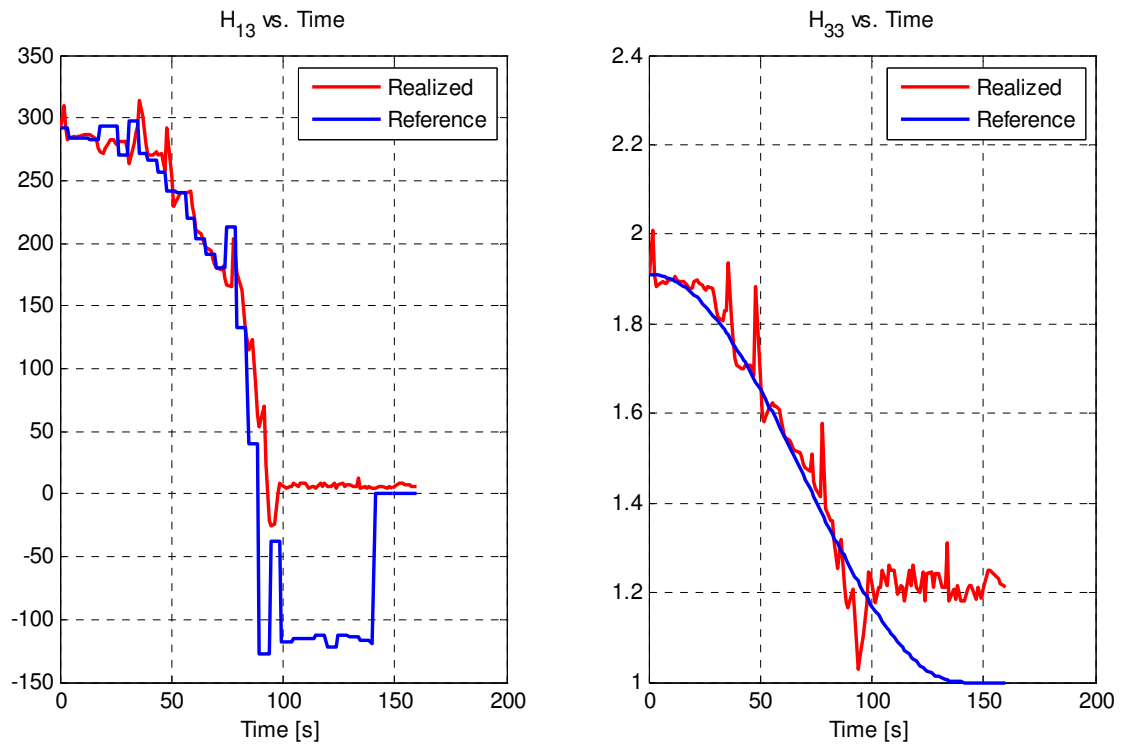


Figure 7.4. Homography elements  $h_{13}$  and  $h_{33}$  as function of time.

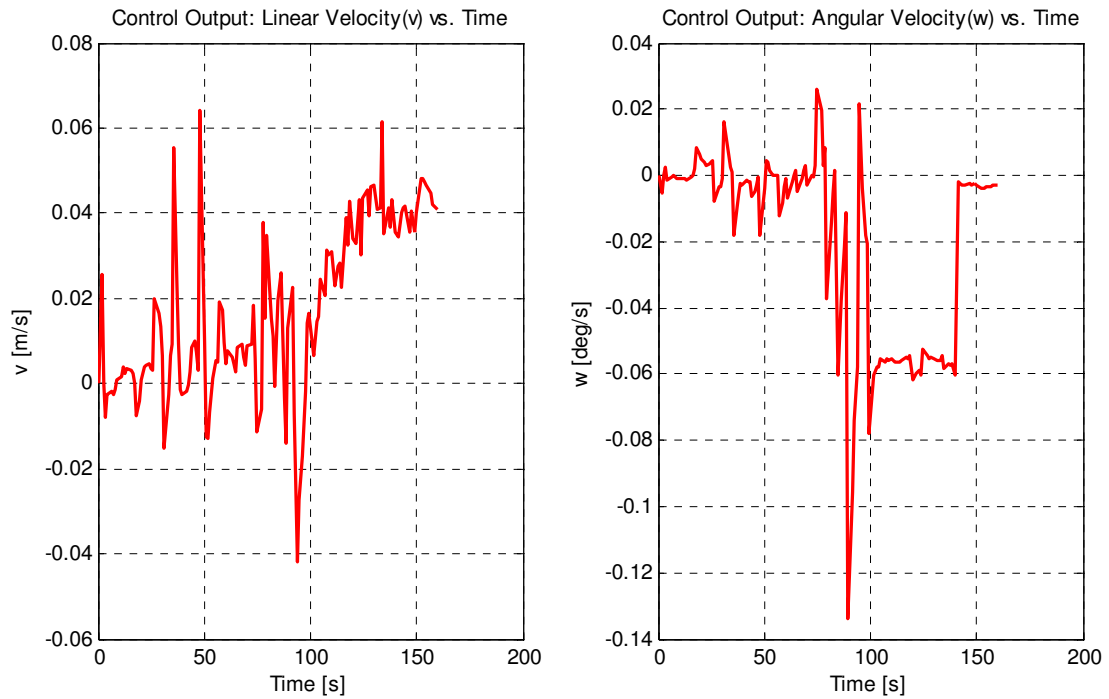


Figure 7.5. Control output (plant input) as function of time.

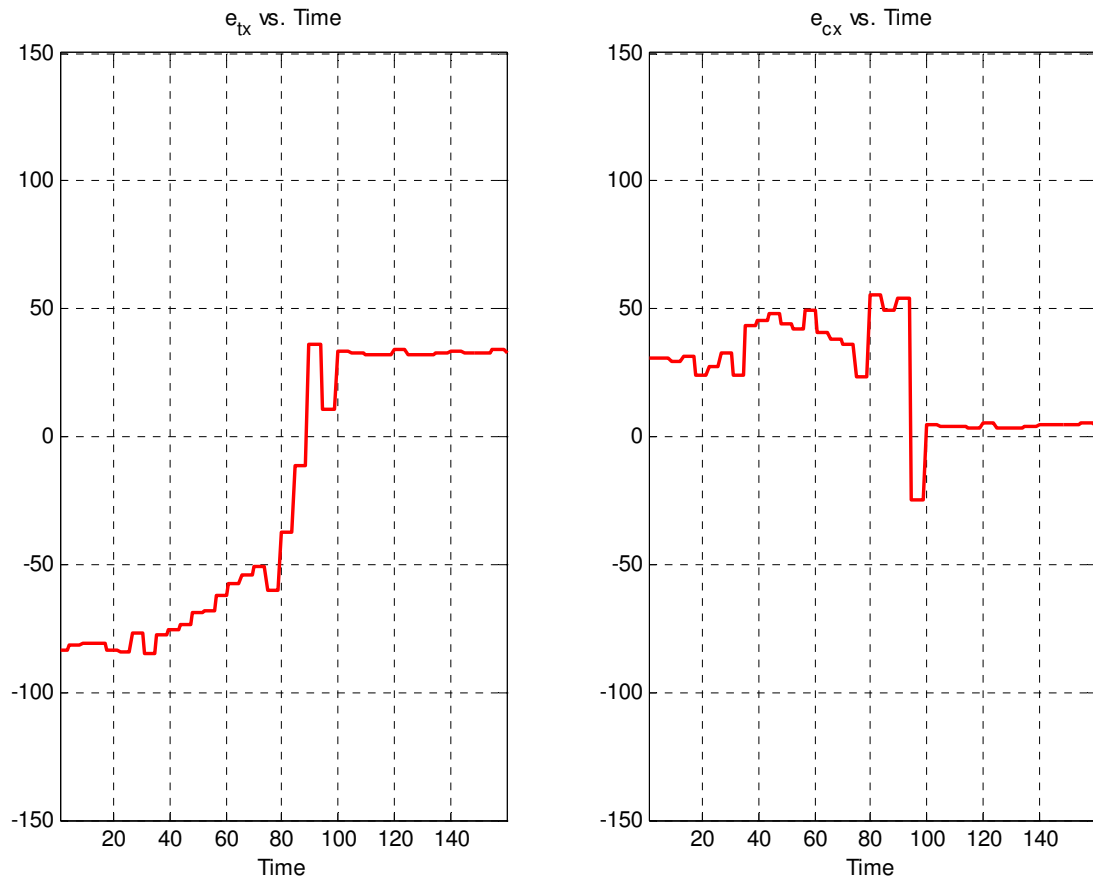


Figure 7.6. X-coordinates of target and current epipole as function of time.



Results for initial configuration  $[x \ z \ \phi]^T = [5 \ -30 \ 20]^T$

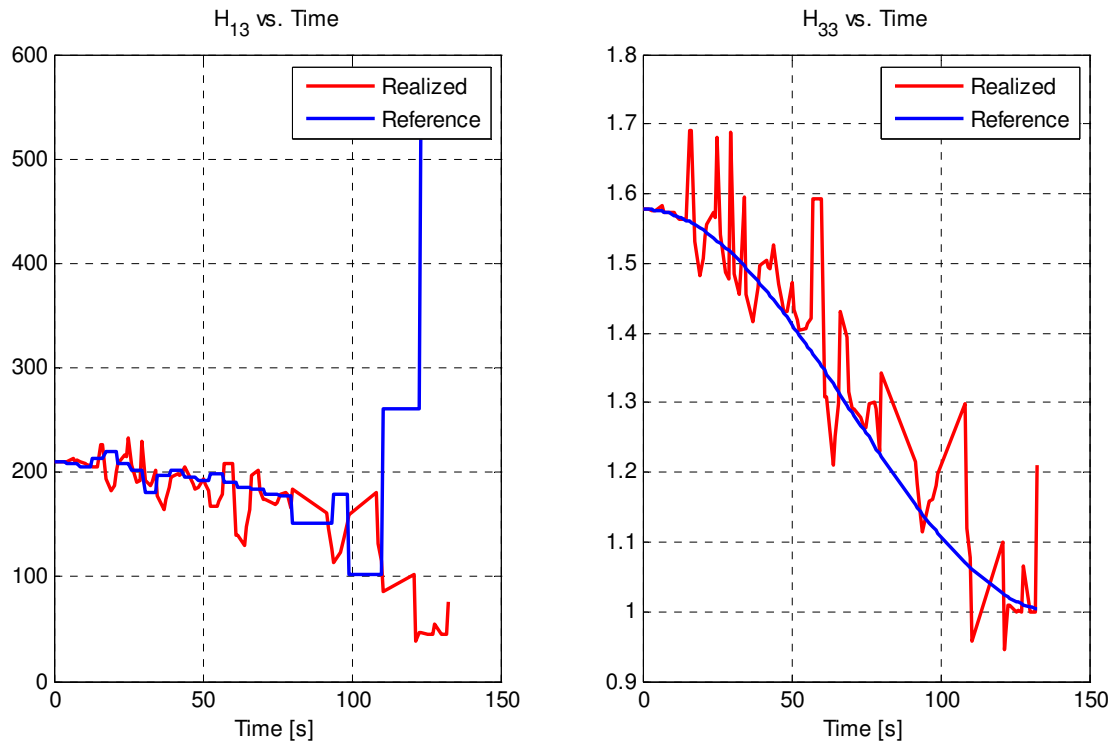


Figure 7.7. Homography elements  $h_{13}$  and  $h_{33}$  as function of time.

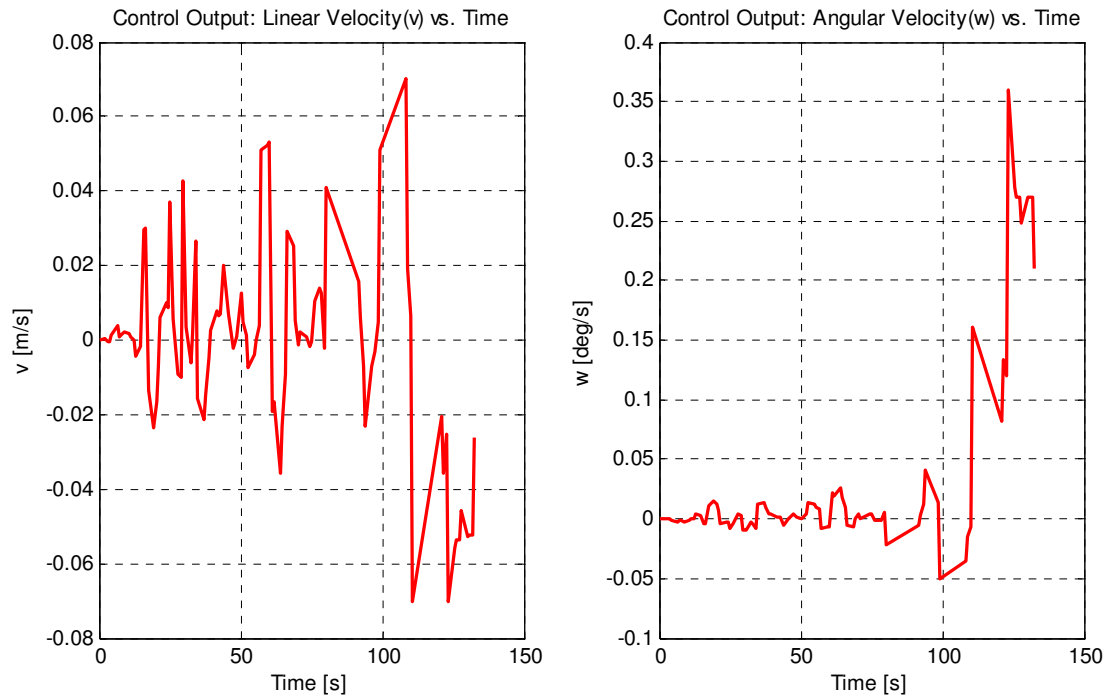


Figure 7.8. Control output (plant input) as function of time.

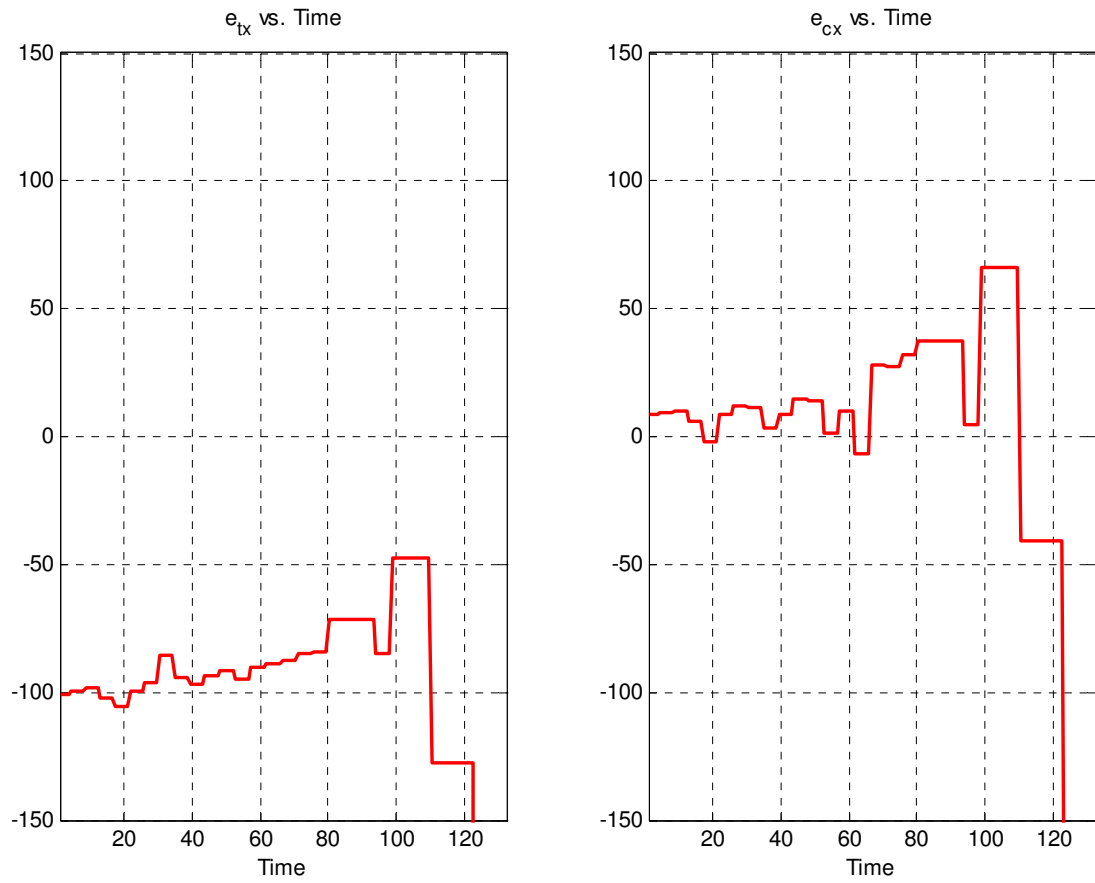


Figure 7.9. X-coordinates of target and current epipole as function of time.

## 8 - REFERENCES

- [1] Visual Control of Vehicles Using Two View Geometry, C. Sagues, G. Lopez-Nicolas, J.J. Guerrero.
- [2] Mobile Robot Navigation Using Visual Servoing, T. Tepe. Coaches: prof.dr. H. Nijmeijer, dr. D. Kotic. MSc Internship TU/e, DC 2010.018
- [3] Homography based visual control of nonholonomic vehicles, C.Sagues, G. Lopez-Nicolas, J.J.Guerrero, IEEE Int. Conference on Robotics and Automation, pages 1703-1708, Rome- Italy, April 2007
- [4] Camera Calibration Toolbox for MATLAB, Jean-Yves Bouguet, [http://www.vision.caltech.edu/bouguetj/calib\\_doc/](http://www.vision.caltech.edu/bouguetj/calib_doc/)
- [5] MATLAB Functions for Multiple View Geometry, D. Capel, A. Fitzgibbon, P. Kovesi, T. Werner, Y. Wexler, and A. Zisserman, <http://www.robots.ox.ac.uk/~vgg/hzbook/code/>, vgg\_F\_from\_P.m
- [6] Multiple View Geometry in Computer Vision, Cambridge University Press: Cambridge, UK, R. Hartley, A. Zisserman.
- [7] E-puck mobile robot, <http://www.e-puck.org/>
- [8] Robot Modeling and Control, John Wiley & Sons, Inc., Mark W. Spong, Seth Hutchinson, M.Vidyasagar.
- [9] Distinctive Image Features from Scale-Invariant Keypoints, January 2004, David G. Lowe.
- [10] Object Recognition from Local Scale-Invariant Features, David G. Lowe.