The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

# Image Based Visual Servoing Using Algebraic Curves Applied to Shape Alignment

Ahmet Yasin Yazicioglu, Berk Calli and Mustafa Unel

***Abstract*** **Visual servoing schemes generally employ various image features (points, lines, moments etc.) in their control formulation. This paper presents a novel method for using boundary information in visual servoing. Object boundaries are modeled by algebraic equations and decomposed as a unique sum of product of lines. We propose that these lines can be used to extract useful features for visual servoing purposes. In this paper, intersection of these lines are used as point features in visual servoing. Simulations are performed with a 6 DOF Puma 560 robot using Matlab Robotics Toolbox for the alignment of a free-form object. Also, experiments are realized with a 2 DOF SCARA direct drive robot. Both simulation and experimental results are quite promising and show potential of our new method.**

## I. INTRODUCTION

Vision based control of robotic systems has been a steadily improving research area recently. Commercially available cameras provide a cheap and powerful tool for many complex robotic tasks in dynamic environments. One particular problem in this domain is object alignment. In visual servoing applications, most of the current alignment systems are based on objects with known 3D models such as industrial parts or objects which have good features due to their geometry or texture. Mostly, features which are feasible to extract and track in real time are used in these approaches [1], [2]. Many works are reported in the literature on alignment using points, lines, ellipses, image moments, etc. [1]-[4]. On the contrary, visually guided alignment of smooth free-form planar objects presents a challenge since these objects may not provide necessary amount of such features. One method to tackle this difficulty is to use polar descriptions of object contours [5]. Also, correlation between reference and current object images can be calculated and used for visual servoing purposes [6]. Alternatively, curves can be fitted to these free-form objects [7]. However obtaining features from these curves for visual servoing algorithms is not a trivial task.

In this paper we propose to use implicit polynomial representation in aligning planar closed curves by employing calibrated image based visual servoing [1]. With the proposed method, an implicit polynomial representation of target object boundary is obtained by a curve fitting algorithm. Acquired polynomial is then decomposed as a unique sum of product of line factors [8]. The intersection points of these lines are then used as point features in visual servoing.

A. Y. Yazicioglu, B. Calli and M. Unel are with Faculty of Engineering and Natural Sciences, Sabanci University, Istanbul 34956, Turkey {ahmetyasin,berkc}@su.sabanciuniv.edu, munel@sabanciuniv.edu

The remainder of this paper is organized as follows: Section 2 presents implicit polynomial representation of curves and how to decompose them into sum of product of line factors. Section 3 reviews image based visual servoing for a calibrated camera. Simulation results are presented in Section 4. Section 5 is on experimental results for curve alignment and discussions. finally, Section 6 concludes the paper with some remarks.

## II. IMPLICIT POLYNOMIAL REPRESENTATION OF PLANAR CURVES

Algebraic curves and surfaces have been used in various branches of engineering for a long time, but in the past two decades they have proven very useful in many model-based applications. Various algebraic and geometric invariants obtained from implicit models of curves and surfaces have been studied rather extensively in computer vision, especially for single computation pose estimation, shape tracking, 3D surface estimation from multiple images and efficient geometric indexing of large pictorial databases [8]-[13]. Algebraic curves are defined by implicit equations of the form $f(x,y)=0$, where $f(x,y)$ is a polynomial with real coefficients in the variables $x$ and $y$. In general an algebraic curve of degree $n$ can be defined by the *implicit polynomial* equation as [13], [14]:

$$\begin{aligned} f_n(x,y) &= \underbrace{a_{00}}_{h_0} + \underbrace{a_{10}x + a_{01}y}_{h_1(x,y)} + \dots \\ &+ \underbrace{a_{n0}x^n + a_{n-1,1}x^{n-1}y + \dots + a_{0n}y^n}_{h_n(x,y)} \\ &= \sum_{r=0}^{n} h_r(x,y) = 0, \end{aligned} \tag{1}$$

where each $h_r(x,y)$ is a homogeneous polynomial of degree $r$ in the variables $x$ and $y$ and $h_n(x,y)$ is called the *leading form*. Since this equation can always be multiplied by a non-zero constant without changing its zero set, it can always be made monic ($a_{n0}=1$) and we will consider the monic curves in this study.

Among the implicit polynomials, odd degree ($n=2k+1$) curves have at least one real asymptote and therefore they are inherently open. On the other hand, even degree ($n=2k$) curves can be either closed or open, depending on the existence of complex or real asymptotes which is determined by the *leading form*. Consequently, closed bounded object contours can only be represented by even degree implicit

Fig. 1. Some sample objects and their outline curves obtained by regularized 3L fitting algorithm.

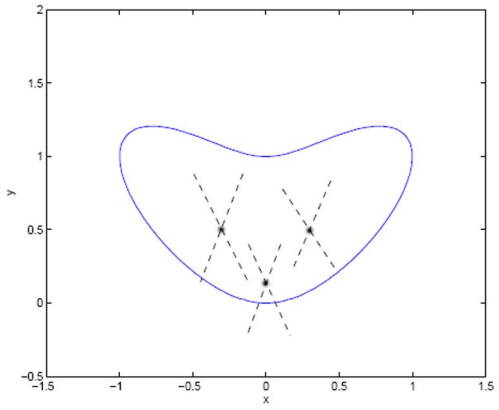


Fig. 2. Three pairs of complex-conjugate lines obtained from the decomposition of a boundary curve.

polynomials. An even degree polynomial can be obtained through fitting algorithms. Some results obtained by using regularized 3L algorithm [7] are shown in Figure 1.

Once the implicit polynomial coefficients are obtained, this polynomial is decomposed as sum of product of line factors [8] to obtain the features we use in visual servoing.

*Theorem 1:* Any non-degenerate monic polinomial $f_n(x,y)$ can be uniquely decomposed into sum of product of complex or real line factors in the following way [8], [13]:

$$f_n(x,y) = \Pi_n(x,y) + \gamma_{n-2}[\Pi_{n-2}(x,y) + \gamma_{n-4}[\Pi_{n-4}(x,y) + ...]] \tag{2}$$

In this equation, $\gamma_j$ denotes constants of the decomposition and $\Pi_j(x,y)$ is the product of $j$ line factors in the following way:

$$\Pi_j(x,y) = \prod_{i=1}^{j}[x + l_{j,i}y + k_{j,i}] \tag{3}$$

For example, by using the proposed decomposition quadratic, cubic and quartic curves can be represented as follows

$$f_2(x,y) = L_1(x,y)L_2(x,y) + \gamma_0 = 0$$
$$f_3(x,y) = L_1(x,y)L_2(x,y)L_3(x,y) + \gamma_1 L_4(x,y) = 0$$
$$\begin{aligned} f_4(x,y) =& L_1(x,y)L_2(x,y)L_3(x,y)L_4(x,y) + \\ & \gamma_2 L_5(x,y)L_6(x,y) + \gamma_2\gamma_0 = 0 \end{aligned} \tag{4}$$

This decomposition is unique for a non-degenerate implicit polynomial in variables $x$ and $y$. For example, $f_4(x,y)$ is completely described by the six line factors $L_i(x,y)$, $i = 1,2,...,6$ and two scalar parameters $\gamma_2$ and $\gamma_0$. Therefore, this decomposition can be used to extract certain robust features that represent the curve and we propose that such features can be used for visual servoing purposes. As the coefficients of the first $n$ lines are complex conjugate pairs for this unique decomposition of a closed curve, these lines give rise to $n/2$ real intersection points on the image plane. Hence, one possible application of this method to visual servoing can be using these pairwise intersection points as image features. An example is shown in Figure 2 where an implicit polynomial of degree four is fitted on the target boundary. Pairwise intersection points of six complex lines are also shown in this figure. In this paper we focus on these point features and point out some alternatives in future works.

In 6 DOF motion, reference and current boundary data are related by a perspective transformation. As we treat the extracted points as point features, they should correspond to the same points with respect to the curve under perspective transformations. Such a correspondence depends on the invariance of curve fitting. Algebraic curve fitting method used in this work is Euclidean invariant and we achieve affine invariance through the whitening normalization [15] of boundary data. If two boundary curves are affine equivalent, their whitening normalization provides rotationally equivalent curves [12]. Consequently with our method correspondence of extracted features under affine transformations is achieved. As long as the average depth of the object from the camera is large, or rotations about the $X$ and $Y$ axis of the camera are small object boundary in two different images will be related by an affine transformation and our method would work properly. However, even in the deviations from affine relation the closed loop control helps in handling this problem. As the closed loop control forces the end effector to the reference pose it also forces the relation between the current and reference boundary data to be affine. In the 6 DOF simulations our results support this claim, however in very large deviations from the affine model this method may not be applicable.

## III. Image Based Visual Servoing

Let $s \in \Re^k$ and $r \in \Re^6$, denote the vectors of image features obtained from visual system and the pose of the end

effector of the robot, respectively. The vector $s$ is a function of $r$, and their time derivatives are related with the image Jacobian $J_I(r) = \partial s/\partial r \in \Re^{kx6}$ as,

$$\dot{s} = J_I(r)\dot{r} \tag{5}$$

For eye-in-hand configuration the image Jacobian corresponding to a single point feature vector $s = [x, y]^T$ is given by:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \underbrace{\begin{bmatrix} -1/Z & 0 & x/Z & xy & -(1+x^2) & y \\ 0 & -1/Z & y/Z & 1+y^2 & -xy & -x \end{bmatrix}}_{J_{xy}} V_c \tag{6}$$

where

$$x = \frac{x_p - x_c}{f_x},\ y = \frac{y_p - y_c}{f_y} \tag{7}$$

and $(x_p, y_p)$ are pixel coordinates of the image point, $(x_c, y_c)$ are the coordinates of the principle point, and $(f_x, f_y)$ are effective focal lengths of the camera. By rearranging and differentiating (7), and writing in matrix form, the following expression can be obtained.

$$\begin{bmatrix} \dot{x_c} \\ \dot{y_c} \end{bmatrix} = \begin{bmatrix} f_x & 0 \\ 0 & f_y \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \underbrace{\begin{bmatrix} f_x & 0 \\ 0 & f_y \end{bmatrix} J_{xy}}_{J_I} V_c \tag{8}$$

where $J_I$ is the pixel-image Jacobian.

In (5), $\dot{r} = V_c$ is also called the end effector velocity screw in eye to hand configuration. This velocity screw is defined in the camera frame, and should be mapped onto the robot control frame. Denoting $V_R$ the end effector velocity skew in robot base frame the mapping can be written as,

$$V_c = TV_R \tag{9}$$

The robot-to-camera velocity transformation matrix $T \in \Re^{6x6}$ is defined as below

$$T = \begin{bmatrix} R & [t]_x R \\ 0_3 & R \end{bmatrix} \tag{10}$$

where $[R, t]$ are being rotational matrix and the translation vector that map camera frame onto robot control frame and $[t]_x$ is the skew symmetric matrix associated with vector $t$.

In light of equation (10), (5) can be rewritten as,

$$\dot{s} = \underbrace{J_I T}_{\triangleq \bar{J}_I} V_R = \bar{J}_I V_R \tag{11}$$

The new image Jacobian matrix $\bar{J}_I$ defines the relation between the changes of image features and end effector velocity in robot base frame. Considering $p$ point features e.g. $s = [x_1, y_1, ..., x_p, y_p]^T$, the Jacobian matrices corresponding to each point should be stacked as

$$\bar{J}_I = \begin{bmatrix} \bar{J}_I^1 \\ . \\ . \\ . \\ \bar{J}_I^p \end{bmatrix} \tag{12}$$

Let $s^*$ be the constant reference feature vector and $e = s - s^*$ define the error. The visual servoing problem is designing an end-effector velocity screw $V_R$ in such a way that the error decays to zero, i.e. $e \to 0$.

By imposing $\dot{e} = -\Lambda e$, where $\Lambda$ is a positive definite gain matrix, an exponential decrease of the error function is realized. Consequently, the velocity screw is derived as:

$$V_R = -\bar{J}_I{}^{\dagger} \Lambda (s - s^*) \tag{13}$$

where $\bar{J}_I{}^{\dagger}$ is the pseudo-inverse of the image Jacobian and $V_R = [V_x, V_y, V_z, \omega_x, \omega_y, \omega_z]^T$.

## IV. Simuation Results

The proposed method is simulated on a 6 DOF Puma 560 robot in eye-in-hand configuration as shown in Figure 4. In simulations, Matlab Robotics Toolbox [16] is used. A planar object is initialized in the field of view of the camera. To evaluate the performance of the method in applications that require 6 DOF motion, a combination of translations and rotations in $x$, $y$ and $z$ directions are introduced between reference and initial positions. Reference and initial position of the object boundary in image and the trajectories of the points which are extracted from the decomposition are given in Figure 3. Control signals and feature errors are presented in Figures 5 and 6 respectively. As it can be seen from these results, the performance of the method is quite promising in this alignment task.

## V. Experimental Results

Some experimental results are presented in this section. Experiments are conducted with a 2 DOF direct drive SCARA robot and a fire-i400 digital camera in an eye-to-hand configuration. A planar free-form object is placed on the tool tip of the robot and camera is fixed above the robot as it can be seen in Figure 7. The robot is controlled with

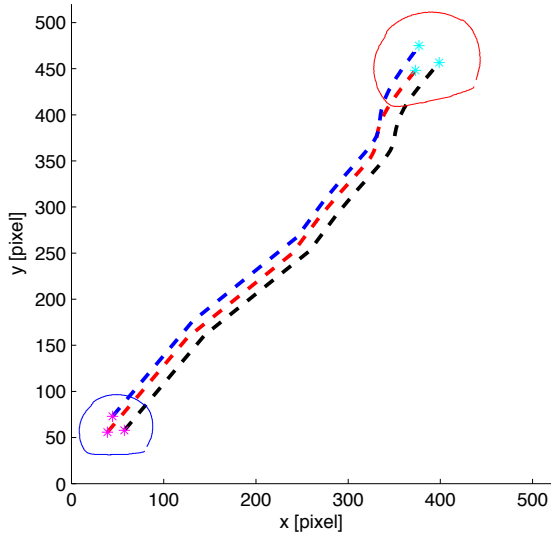


Fig. 3. Trajectory of the target object in image space

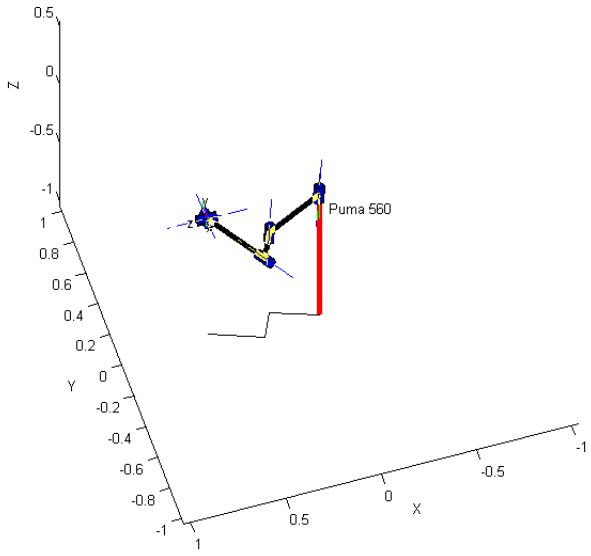


Fig. 4. 6 DOF Puma 560 robot in Matlab Robotics Toolbox

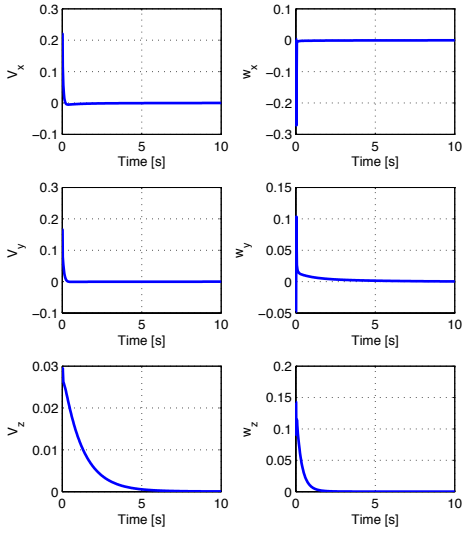


Fig. 5. Control efforts

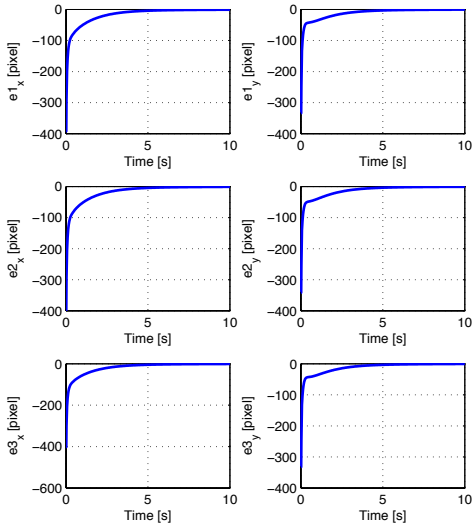


Fig. 6. Pixel errors

Fig. 7. Experimental Setup

a dSPACE 1102 controller card. The programming language of the card is Visual C.

In the experiments, object boundary is extracted by using Canny edge detection algorithm [17]. From these edges, we obtain a fourth degree implicit polynomial by using the regularized 3L fitting algorithm [7]. The implicit curve is then decomposed as explained in Section II. Two point features are obtained from the intersection of the first 4 complex-conjugate lines. These points are then used as point features in visual servoing.

The control loop is made up of one inner and one outer loops. The outer loop is run via vision system. It uses the point features to generate velocity references to the inner loop by using pixel errors of these points. These references are used by the inner loop to position the robot. Sampling time of the inner control loop is 1 ms. The frame rate of the camera is 30 fps.

A diagonal gain matrix of

$$\Lambda = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \tag{14}$$

is used in computing the velocity screw of the end effector. According to calibration results, effective focal lengths of the camera in x and y directions are measured as $f_x = f_y = 970$, and image center coordinates $(x_c, y_c) = (160, 120)$.

Two experiments are presented in this section. In the first experiment object plane is parallel to the image plane and

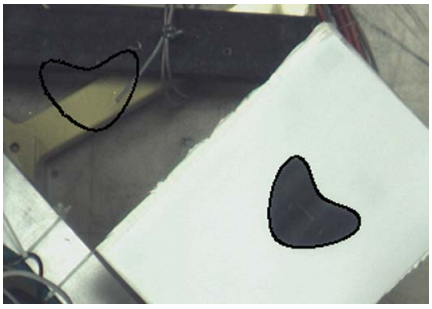

Fig. 8. Reference and initial poses

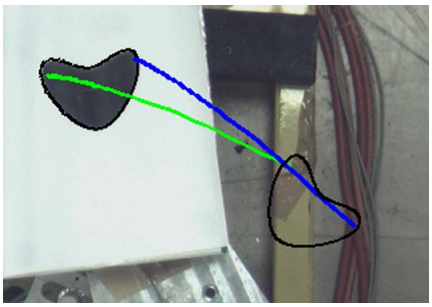

Fig. 9. Trajectory of point features

motion of the end effector induces rigid body motion for the object boundary. Significant rotation and translation exist between the reference and initial poses. The reference and initial positions are as in Figure 8. Trajectories of the point features can be seen in Figure 9.

The error plots are given in Figures 10 and 11. Control signals are presented in Figure 12. Less than 2 pixel errors is observed in steady state.

In the second experiment, the case where the image plane is not parallel to the object plane is examined. In this case motion of the end effector induces affine motion on the object boundary data. Significant translation and rotation are introduced between reference and initial pose. Reference and initial poses can be seen in Figures 13 and 14 respectively. Pixel errors in $x$ direction, pixel errors in y direction and control efforts are depicted in Figures 15, 16 and 17.

## VI. CONCLUSIONS AND FUTURE WORKS

### A. Conclusions

In this paper, a novel method for using implicit curves as image features for vision based robot control is presented. Implicit polynomials of degree $n$, where $n$ is an even number are fitted to the object boundary and decomposed into line factors. Intersection of the complex conjugate lines are real points and they are used as image features in visual servoing. Results of simulations with a 6 DOF Puma 560 and experiments conducted on a 2 DOF SCARA robot are quite promising.

### B. Future Works

The method presented in this paper is one of the many possible choices for using implicit form of closed curves in visual servoing. Instead of using the intersection of complex conjugate line factors, one could use the parameters of those lines as features. This can be achieved by deriving the analytical image Jacobian corresponding to the parameters of the extracted line factors. In our future research we are planing to extend our work along these lines.

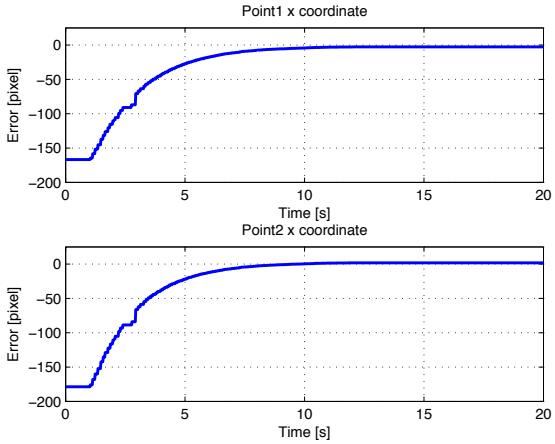


Fig. 10. Pixel errors in x direction of the image plane

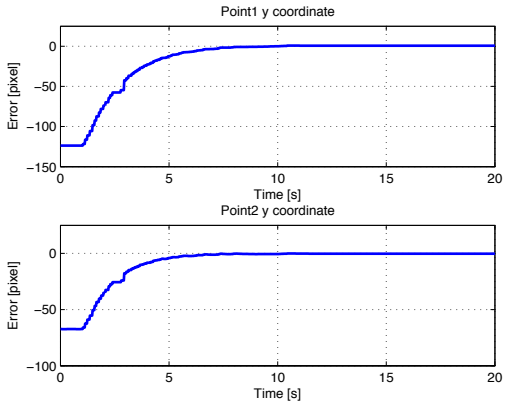


Fig. 11. Pixel errors in y direction of the image plane

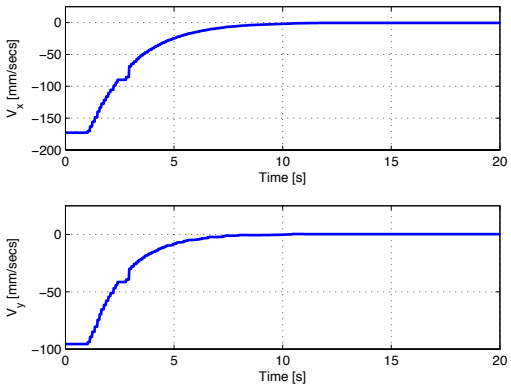


Fig. 12. Control efforts

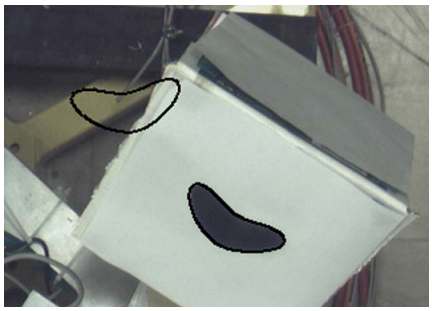

Fig. 13. Reference and initial poses

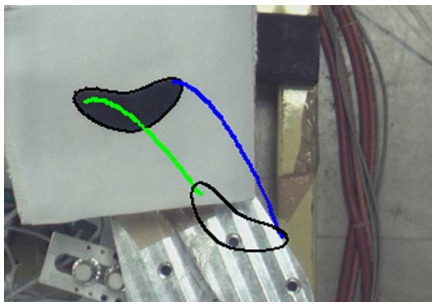

Fig. 14. Trajectories of point features

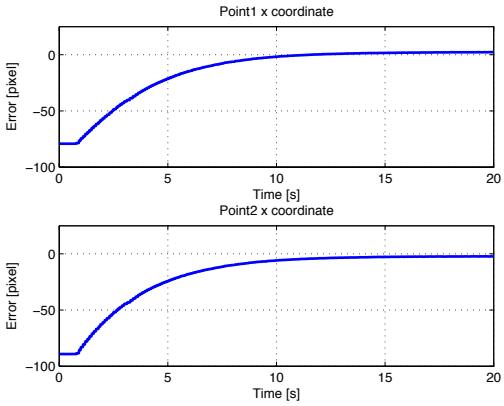


Fig. 15. Pixel errors in x direction of the image plane

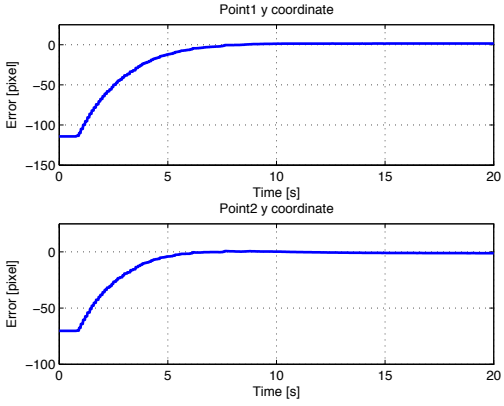


Fig. 16. Pixel errors in y direction of the image plane

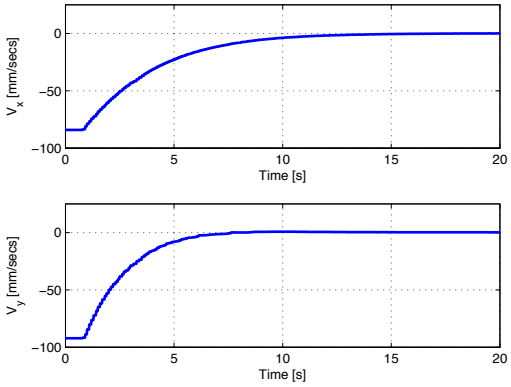


Fig. 17. Control efforts

## REFERENCES


[1] S. Hutchinson, G. D. Hager and P. I. Corke, "A tutorial on visual servo control", *IEEE Trans. on Robotics and Automation*, Vol. 12, No. 5, pp. 651-670, 1996.

[2] F. Chaumette, S. Hutchinson, "Visual Servo Control, Part I: Basic Approaches and Part II: Advanced Approaches", *IEEE Robotics and Automation Magazine*, Vol. 13, No. 4, pp. 82-90, 2006.

[3] B. Espiau, F. Chaumette, P. Rives, "A New Approach to Visual Servoing in Robotics", *IEEE Trans. on Robotics and Automation*, Vol. 8, No. 3, pp. 313-326, 1992.

[4] O. Tahri, F. Chaumette, "Point Based and Region Based Image Moments for Visual Servoing of Planar Objects", *IEEE Trans. on Robotics and Automation*, Vol. 21, No. 6, pp. 1116-1127, 2005.

[5] C. Collewet, F. Chaumette, "A contour approach for image-based control on objects with complex shape," IEEE/RSJ International Conference on Intelligent Robots and Systems, Vol. 1, pp. 751-756, 2000.

[6] E. Malis, G. Chesi, R. Cipolla, "21/2D Visual Servoing with Respect to Planar Contours having Complex and Unknown Shapes" The International Journal of Robotics Research, Vol. 22, No.10-11, pp. 841-853, 2003.

[7] T. Sahin, M. Unel, "Globally Stabilized 3L Curve fitting", *Lecture Notes in Computer Science (LNCS-3211)*, pp. 495-502, Springer-Verlag, 2004.

[8] M. Unel, W. A. Woovich, "On the construction of complete sets of geometric invariants for algebraic curves," Advances in Applied Mathematics, Vol.24, No.1, pp. 65-87, 2000.

[9] G. Taubin, D. B. Cooper, "2D and 3D object recognition and positioning with algebraic invariants and covariants," Chapter 6 of Symbolic and Numerical Computation for Articial Intelligence, Academic Press, 1992.

[10] J. Bloomenthal, "Introduction to implicit surfaces," Kaufmann, Los Altos, CA, 1997.

[11] D. Keren, C. Gotsman, "Fitting curves and surfaces to data using constrained implicit polynomials," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, No. 1, January 1999.

[12] W. A. Wolovich, M. Unel, "The Determination of Implicit Polynomial Canonical Curves," IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 20, No. 10, pp. 1080-1089, 1998.

[13] M. Unel, "Polynomial decompositions for shape modeling, object recognition and alignment," Ph.D. Thesis, Brown University, Providence, 1999.

[14] C.G. Gibson, "Elementary geometry of algebraic curves", Cambridge University Press, Cambridge, UK, 1998.

[15] K. Fukunaga, "Introduction to Statistical Pattern Recognition," 2nd ed. New York: Academic Press, 1990.

[16] P. I. Corke, "A Robotics Toolbox for MATLAB," IEEE Robotics and Automation Magazine, Vol.3, No.1, pp. 24-32, 1996.

[17] J. F. Canny, "A computational approach to edge detection," IEEE Transactions on Pattern Analysis & Machine Intelligence, Vol. 8, No.6, pp. 679-698, 1986.