Force Control and Visual Servoing Using Planar Surface Identification

Olsson, Tomas; Bengtsson, Johan; Johansson, Rolf; Malm, Henrik

Published in:
Proceedings. ICRA '02.

DOI:
10.1109/ROBOT.2002.1014414

2002

Proceedings of the 2002 IEEE
International Conference on Robotics & Automation
Washington, DC • May 2002

Force Control and Visual Servoing Using Planar Surface Identification

Tomas Olsson[+], Johan Bengtsson[+], Rolf Johansson[+], Henrik Malm[×]

[+]Department of Automatic Control, Lund Institute of Technology, Lund University, SE-221 00 Lund Sweden;
E-mail: Tomas.Olsson@control.lth.se, Johan.Bengtsson@control.lth.se, Rolf.Johansson@control.lth.se

[×]Centre for Mathematical Sciences, Lund Institute of Technology, Lund University, SE-221 00 Lund Sweden;
E-mail: henrik.malm@math.lth.se

Abstract

When designing flexible multi-sensor based robot systems, one important problem is how to combine the measurements from disparate sensors such as cameras and force sensors. In this paper, we present a method for combining direct force control and visual servoing in the presence of unknown planar surfaces. The control algorithm involves a force feedback control loop and a vision based reference trajectory as a feed-forward signal. The vision system is based on a constrained image-based visual servoing algorithm designed for surface following, where the location and orientation of the planar constraint surface is estimated online using position-, force- and visual data. We show how data from a simple and efficient camera calibration method can be used in combination with force and position data to improve the estimation and reference trajectories.

The method is validated through experiments involving force controlled drawing on an unknown surface. The robot will grasp a pen and use it to draw lines between a number of markers drawn on a white-board, while the contact force is kept constant. Despite its simplicity, the performance of the method is satisfactory.

1 Introduction

Many applications of robotics require the robots to operate in environments that are complex, and where the geometry is at least partially unknown. Designing a flexible robot system, capable of operating in such environments, requires the integration of a number of different sensors into the robot system.

It has been realized for many years that force sensing capabilities are crucial when robots are required to interact with their environment. Over the last decade there has also been a growing interest in vision based control, since a lot of information can be obtained from visual data. A background on visual servoing is given in [1].

The nature and limited accuracy of vision based control makes it less suitable for controlling interaction with objects in the environment. An interesting and obvious solution is to combine force control and visual servoing in a multi-sensor control system. Recently, some research on this subject has been presented [2, 3, 4, 5, 6].

Perhaps the most obvious approach to the problem is to combine the measurements from the cameras and force sensor using multi-sensor fusion methods. However, as several researchers have pointed out [3], the force- and visual sensors are fundamentally different in that they measure very different physical phenomena, while the goal of most multi-sensor fusion methods is to obtain a single piece of information from the sensor data. This makes such an approach less suitable in this case.

In this work another approach is taken, based on the following observations. Far away from any constraints, the robot can be controlled by unconstrained visual servoing only. When close to the constraint surface, one or several degrees of freedom should become force controlled in order to accurately control the interaction with the constraint surface. The remaining degrees of freedom should then be controlled by a constrained visual servoing algorithm. The unknown constraint equations can be estimated recursively from the available sensor data. The estimated constraint can also be used to form a feed-forward signal in order to improve the behavior in the force controlled directions.

This paper shows how these ideas can be used to implement a system, where an industrial robot interacts with planar objects in its environment through a combination of force- and vision based control. Further, we show that data from the sensors can be used to accurately measure the position and orientation of the planar object, and show how this new information can be used to improve the control.

0-7803-7272-7/02/$17.00 © 2002 IEEE

We also present a simple but efficient way to calibrate the camera system, and discuss how the results from this calibration can be used for improving the control and the identification of the parameters describing the plane.

In section 3 we present some experimental results for the described system.

2 Methods

2.1 Calibration of the camera system

Many methods for camera calibration use a number of images of a planar calibration object to estimate the intrinsic and extrinsic camera parameters [7]. The images are taken from several different positions and orientations.

In our case we use cameras that are fixed in the workspace, and we attach the calibration object to the robot end-effector itself. From the kinematics of the robot we can get accurate information on the movement of the end-effector between the images. Therefore, the extrinsic parameters are partially known, and this is used in the algorithm in order to increase the accuracy and decrease the number of parameters that need to be estimated.

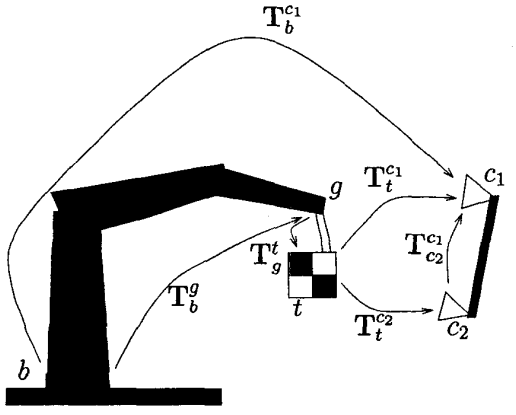

Figure 1: The most important frames and transformations.

The setup of the camera/robot system is shown in Fig. 1. The two fixed cameras are observing a calibration object that is rigidly attached to the end-effector. The relative position of the calibration object with respect to the end-effector frame does not have to be known. The goal of the calibration is to be able to find the intrinsic parameters of the two cameras, as well as the rigid transformations $\mathbf{T}_{c_2}^{c_1}$ and $\mathbf{T}_b^{c_1}$ in Fig. 1.

The calibration algorithm works in three steps:

1. Estimation of intrinsic and extrinsic camera parameters according to [7].
2. Calculation of $\mathbf{T}_g^t$ by hand-target calibration according to [8].
3. The results can be improved by simultaneous calculation of all system parameters using nonlinear least squares optimization. This step is not necessary, but we have found that it will improve the accuracy of the calibration.

If we choose to perform the optimization in step 3 above, steps 1 and 2 will be used mainly for obtaining suitable starting values.

In step 2, a hand-eye calibration method [8] can be used if we note that for two different end-effector positions, denoted by numbers p and q, and camera c_k, there holds a relationship

$$\mathbf{T}_t^{c_k}(q)^{-1}\,\mathbf{T}_t^{c_k}(p)\,\mathbf{T}_g^t = \mathbf{T}_g^t\,\mathbf{T}_b^g(q)\,\mathbf{T}_b^g(p)^{-1} \qquad (1)$$

which can be solved for the unknown constant $\mathbf{T}_g^t$.

The final optimization is used to minimize, for both cameras, the errors between the m measured and reprojected image points in each of the n images

$$\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{2}\left(\mathbf{x}_{ijk} - \hat{\mathbf{x}}_{ijk}(\mathbf{K}_1, \mathbf{K}_2, \mathbf{T}_g^t, \mathbf{T}_{c_2}^{c_1}, \mathbf{T}_b^{c_1})\right)^2$$

where k is the camera number, and $\hat{\mathbf{x}}_{ijk}$ is given by the projection equations

$$\begin{aligned}\lambda_1\hat{\mathbf{x}}_{ij,1} &= \mathbf{K}_1\mathbf{T}_b^{c_1}\mathbf{T}_b^g(i)^{-1}(\mathbf{T}_g^t)^{-1}\mathbf{X}_j \\ \lambda_2\hat{\mathbf{x}}_{ij,2} &= \mathbf{K}_2(\mathbf{T}_{c_2}^{c_1})^{-1}\mathbf{T}_b^{c_1}\mathbf{T}_b^g(i)^{-1}(\mathbf{T}_g^t)^{-1}\mathbf{X}_j \qquad (2)\end{aligned}$$

where $\mathbf{X}_j$ are the model points of the calibration object. $\mathbf{K}_1$ and $\mathbf{K}_2$ are matrices of intrinsic camera parameters, see [7]. The minimization is performed with respect to $\mathbf{K}_1$, $\mathbf{K}_2$, $\mathbf{T}_b^{c_1}$, $\mathbf{T}_{c_2}^{c_1}$ and $\mathbf{T}_g^t$. Multiplying Eq. (2) with the inverses of $\mathbf{K}_1$ and $\mathbf{K}_2$ we get projection equations on the form

$$\begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{Z}\begin{pmatrix} X \\ Y \end{pmatrix}, \qquad (3)$$

for the point $(X\ \ Y\ \ Z)^T$ in camera coordinates.

2.2 Visual servoing

Visual servoing methods can roughly be divided into two main groups [1]. In *position-based* visual servoing, the measured output to be controlled is the Cartesian position and orientation of an object. In the fixed-camera case, we control the pose of the robot end-effector. The pose has to be reconstructed from the available image data. The other group of methods is *image-based* visual servoing, where the quantities to be controlled are defined directly as features in image space. Because of this, it is not necessary to perform a complete 3D-reconstruction of the scene. There also exist methods that combine elements from image-based and position-based servoing, often called 2 1/2 D visual servoing [9].

Any of these methods could have been used in this work, but the image-based servoing method was chosen since it is known to be very robust to calibration errors [1].

In an image-based visual servo, the control error $\mathbf{e}$ is defined as

$$\mathbf{e} = \mathbf{y}_r - \mathbf{y}$$

where $\mathbf{y}_r$ and $\mathbf{y}$ are vectors of image space feature parameters, $\mathbf{y}_r$ is the desired (end) position of the feature locations, and $\mathbf{y}$ is the measured position. A simple control law that would drive the error $\mathbf{e}$ to zero is

$$\dot{\mathbf{r}} = k_v \left[\mathbf{J}_v^{l,r}(\mathbf{r})\right]^+ \mathbf{e} = k_v \left[\mathbf{J}_v^{l,r}(\mathbf{r})\right]^+ (\mathbf{y}_r - \mathbf{y}) \qquad (4)$$

where $\mathbf{r}$ are the Cartesian coordinates and $\dot{\mathbf{r}}$ is the velocity screw of the end-effector, measured in the robot base frame, k_v is a constant gain, and $\mathbf{J}_v^{l,r}(\mathbf{r})$ is the so called image Jacobian, which relates image space velocities $\dot{\mathbf{f}}$ of the features to the corresponding end-effector velocity in Cartesian space by

$$\dot{\mathbf{f}} = \mathbf{J}_v^{l,r}(\mathbf{r})\dot{\mathbf{r}} \qquad (5)$$

Note that the image Jacobian is in general a function of the Cartesian coordinates $\mathbf{r}$, which means that we need some Cartesian information to calculate it exactly, usually the depth of the imaged points in the cameras. In this work, approximate depth information is calculated by using the data from the calibration of the camera system and the robot kinematics, but the depth information could also be obtained from the stereo images.

When using stereo cameras, the correct combined Jacobian for the stereo system is obtained by stacking the Jacobians for the individual cameras [10]

$$\mathbf{J}_v^{l,r}(\mathbf{r}) = \begin{pmatrix} \mathbf{J}_v^l(\mathbf{r})\mathbf{M}_b^l \\ \mathbf{J}_v^r(\mathbf{r})\mathbf{M}_b^r \end{pmatrix} \qquad (6)$$

where $\mathbf{M}_b^l$ and $\mathbf{M}_b^r$ are the transformation matrices for the screw, from the robot base frame to the left and right cameras respectively, and $\mathbf{J}_v^l(\mathbf{r})$ and $\mathbf{J}_v^r(\mathbf{r})$ are the Jacobians for the left and right cameras. The exact form of these Jacobians for point features can be found for instance in [1]. The screw transformation matrix for the left camera is given by

$$\mathbf{M}_b^l = \begin{pmatrix} \mathbf{R}_b^l & [\mathbf{t}_b^l\times]\,\mathbf{R}_b^l \\ \mathbf{0}_{3\times3} & \mathbf{R}_b^l \end{pmatrix} \qquad (7)$$

where $\mathbf{R}_b^l$ and $\mathbf{t}_b^l$ describe the relative position of the left camera with respect to the robot base frame. Equivalent expressions hold for the right camera.

2.3 Constrained motion

The constraint on the reference trajectories is that the motion should be in the plane $\hat{\mathbf{p}}$ in Cartesian space defined by

$$\hat{\mathbf{p}}^T\hat{\mathbf{r}} = 0 \qquad (8)$$

where $\hat{\mathbf{p}} = (p_1 \;\; p_2 \;\; -1 \;\; p_4)^T$ and $\hat{\mathbf{r}} = (X \;\; Y \;\; Z \;\; 1)^T$. Differentiating this expression leads to an equation

$$\mathbf{p}^T\dot{\mathbf{r}} = 0 \qquad (9)$$

for the constrained velocity $\dot{\mathbf{r}} = \left(\dot{X} \;\; \dot{Y} \;\; \dot{Z}\right)^T$ of the end-effector, with $\mathbf{p} = (p_1 \;\; p_2 \;\; -1)^T$.

The motion on the surface of the plane is now given by

$$\begin{aligned} \dot{\mathbf{f}} &= \mathbf{J}_v^{l,r}(\mathbf{r}) \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ p_1 & p_2 \end{pmatrix} \begin{pmatrix} \dot{X} \\ \dot{Y} \end{pmatrix} = \\ &= \mathbf{J}_{v,c}^{l,r}(\mathbf{r}) \begin{pmatrix} \dot{X} \\ \dot{Y} \end{pmatrix} \end{aligned} \qquad (10)$$

It is obvious that if $\mathbf{J}_v^{l,r}(\mathbf{r})$ has full rank, so does the reduced image Jacobian $\mathbf{J}_{v,c}^{l,r}(\mathbf{r})$. The constrained vision based control law becomes simply

$$\dot{\mathbf{r}}_c = k_v \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ p_1 & p_2 \end{pmatrix} \left[\mathbf{J}_{v,c}^{l,r}(\mathbf{r})\right]^+ (\mathbf{y}_r - \mathbf{y}) \qquad (11)$$

As the constraint itself, described by Eq. (8), is in general unknown, a method for its determination is required. A method based on local estimation of the constraint using measurements of forces and torques is suggested in [2]. In our case the low signal-to noise ratio of the force sensor, in combination with large friction forces, makes this method less suitable. Instead we estimate the parameter vector $\hat{\mathbf{p}}$ using a recursive least-squares method and the equations

$$\begin{pmatrix} X_m & Y_m & (Z_m - F_z/k_z) & 1 \end{pmatrix} \hat{\mathbf{p}} = 0 \qquad (12)$$

$$(p_1,\; p_2,\; -1) \left(\left[\mathbf{J}_v^{l,r}(\mathbf{r})\right]^+ (\mathbf{y}_r - \mathbf{y})\right)^T = 0 \qquad (13)$$

where X_m, Y_m and Z_m are the measured Cartesian coordinates for the end-effector obtained from the robot kinematics. Eq. (12) is derived from Eq. (8), where the z-coordinate is calculated from the measured force F_z and the stiffness k_z of the spring-mounted tool in the z-direction, the direction in which the tool is pointing. Eq. (13) is derived from Eq. (9) and the fact that in an accurately calibrated stereo system, the unconstrained Cartesian velocities $\dot{\mathbf{r}}$ obtained from the control law will produce trajectories that are straight lines. If the system is moving between two points that both lie in the constraint plane, then all the velocity vectors $\dot{\mathbf{r}}(t)$ will be approximately parallel to the plane. The purpose of Eq. (13) is to use the predictive capability of the visual information in the estimation of the slope, which can be expected to speed up the convergence. Note that Eq. (13) requires the calculation of the pseudo-inverse of $\mathbf{J}_v^{l,r}(\mathbf{r})$. In our case, this is a 3×3 matrix, and the time to compute the inverse can be considered to be negligible.

Note that in Eq. (12) it is required that the stiffness k_z is known. However, it is possible to use just a rough estimation, since the force control will keep F_z approximately

constant. Because of this the slope of the plane should still be estimated correctly.

2.4 Hybrid force/vision control

We use a proportional motion rate controller

$$\dot{\mathbf{r}}_F = \begin{pmatrix} \mathbf{0}_{2\times 1} \\ k_F(F_r - F) \end{pmatrix} \tag{14}$$

where F_r is the reference for the force F in the z-direction. If we combine this with the vision based reference trajectory of Eq. (11), the hybrid control law becomes

$$\begin{aligned} \dot{\mathbf{r}}_H &= \dot{\mathbf{r}}_F + \dot{\mathbf{r}}_c = \begin{pmatrix} \mathbf{0}_{2\times 1} \\ k_F(F_r - F) \end{pmatrix} + \\ &+ k_v \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ p_1 & p_2 \end{pmatrix} \left[\mathbf{J}^{l,r}_{v,c}(\mathbf{r})\right]^+ (\mathbf{y}_r - \mathbf{y}) \end{aligned} \tag{15}$$

3 Experiments

3.1 Experimental setup

We have set up an experimental system in the Robotics Lab at the department of Automatic Control at Lund Institute of Technology. The system consists of a 6-degree-of-freedom ABB Industrial Robot 2000 equipped with a 6DOF JR3 force/torque sensor, and two Sony DFW-V300 digital cameras working at a frame rate of 30 images/second. The cameras send image data through a 400 Mbps IEEE-1394 connection to a standard 450 Mhz WindowsNT PC where the image processing is performed. The extracted feature point locations are then sent to a Sun Ultra60 computer running a Matlab/Simulink version of the vision/force controller. The sampling period of the controller is 67 ms. The low level joint position control is performed by an open robot control system developed at our department [11].

In the experiments we assume that the relative positions between the cameras and the robot are unknown, but that both cameras are viewing the robot. Further we assume that the intrinsic camera parameters are completely unknown.

A simple and illustrative task is chosen to test the performance of the presented methods. Two objects are placed in the view of both cameras, a pen and a white-board. The exact position and orientation of the objects is unknown, but the pen is standing in the vertical position. On the white-board, a number of dots are drawn in random positions. The system should be able to align the end-effector with the pen and grasp it, and use it to connect the dots on the white-board with lines. The control should keep the contact force constant during the drawing phase. The exact location of the board should be estimated accurately using available sensor data.

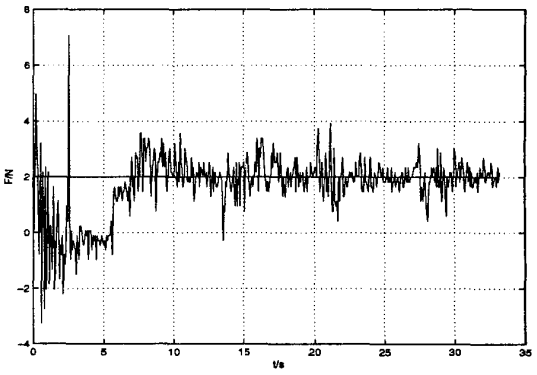

Figure 2: Measured force F and reference $F_r = 2$ N.

3.2 Experimental methodology

The experiment is divided into an off-line and an online phase. Off-line, the robot with the attached calibration object moves into 7 different positions, the locations of 48 coplanar features on the object are measured in each image and the camera system is calibrated. Experiments with real and simulated data show that $\mathbf{T}_b^{c_1}$ and $\mathbf{T}_{c_2}^{c_1}$ can be estimated with an error in translation of approximately 2 cm, and an orientation error of 1° in the Euler angles, using only a small (10cm×15cm) calibration object. Then, the positions in the cameras of the pen and the dots on the white-board are found and stored.

Online, the robot end-effector is aligned with the stored pen position using visual servoing. Once the pen is grasped, we use visual servoing to guide the pen to the board, and once contact with the board is established, the force/vision control makes the robot connect the dots.

3.3 Results

In Fig. 2 we see the measured force in the force controlled direction. The force control is switched on at $t = 5.3$ s and contact is achieved at $t = 5.5$ s. Note the large initial influence of the inertial forces during the acceleration at $t = 0$.

The final estimation of the plane parameters is $\hat{\mathbf{p}} = (-0.0461 \;\; 0.0128 \;\; -1 \;\; 1.235)^T$, compared to the true values $\hat{\mathbf{p}}_r = (-0.0478 \;\; 0.0155 \;\; -1 \;\; 1.237)^T$ obtained from an accurate measurement using the robot. The recursively estimated parameters can be seen in Fig. 3. The estimation is started at $t = 6.7$ s. In Fig. 4 a), c) and d) we see the residuals for Eq. (12), their autocorrelation, and the histogram of the residuals. The residuals for Eq. (13) can be seen in Fig. 4 b). Note that $t = 0$ in Fig. 4 a) and b) corresponds to $t = 6.7$ s in Fig. 2.

The trajectory of the tip of the pen in Cartesian space can be seen in Fig. 5, and the corresponding image-space trajectories can be seen in Fig. 6–7.

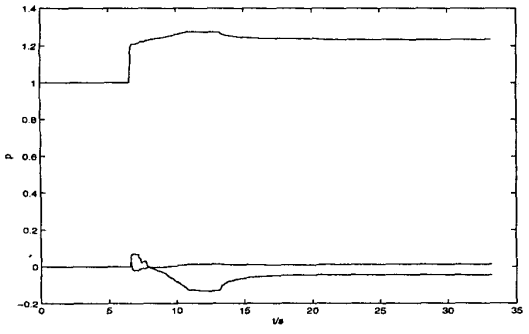

Figure 3: Plane parameters p_1, p_2 and p_4.

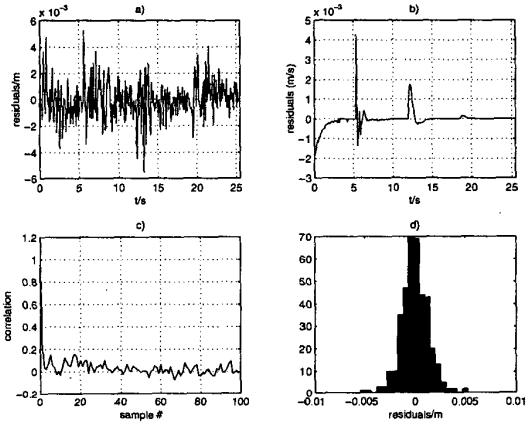

Figure 4: a) Residuals in Eq. (12). b) Residuals in Eq. (13). c) Autocorrelation of the residuals in Eq. (12). d) Histogram of the residuals in Eq. (12).

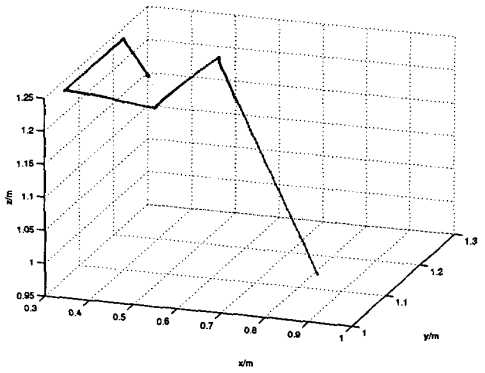

Figure 5: Trajectory of the pen, Cartesian space.

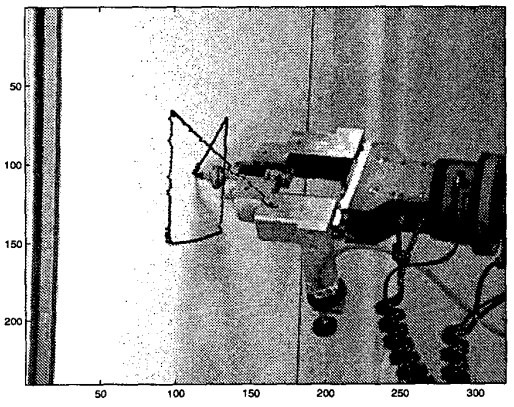

Figure 6: Pen trajectory, camera 1.

4 Discussion

We see from Fig. 2 that the force overshoots slightly at the beginning of the first line at $t \approx 7$s. The reason is that the estimate of the plane parameters $\hat{\mathbf{p}}$ has not yet converged, and the accuracy of the reference trajectories from the vision system is therefore limited by the relatively low accuracy of the cameras. The combined stiffness of the environment and the spring-mounted pen is estimated to 400 N/m, which means that the overshoot corresponds to an error of approximately 1.5 mm in the reference trajectory. Fig. 2 also shows the stick-slip effect due to friction at $t \approx 16$ s.

Another source of error is the noise resulting from errors in the image feature extraction, most clearly seen in Fig. 6. This will result in noise in the reference trajectories and the resulting contact forces, see Fig. 2.

The estimation of the plane parameters changes stepwise, with fast convergence to the final value at time $t = 13.3$ s, the start time for the drawing of the second line. The estimated values at $t < 13.3$ s reflects the slope of the plane along the first line. The small error in the estimation is caused by the noisy data from the force sensor, errors in the estimation of the stiffness of the spring, and calibration errors. Another reason is that the board is flexible, and is therefore deformed slightly by the contact forces.

In Fig. 4 c) and d) wee see that the residuals of Eq. (12) are approximately uncorrelated, Gaussian white noise. The residuals of Eq. (13) seen in Fig. 4 b) are of course not white noise, since they are affected not only by random image-space errors, but also by the constant errors in the Cartesian data estimated in the calibration. In general, the maximum error in $\dot{z}$ decrease to a value around $2 \cdot 10^{-3}$ m/s at a velocity in the x-y plane of 0.15 m/s, roughly corresponding to an error in the slope of the plane of around 1°. This agrees well with the estimated cali-

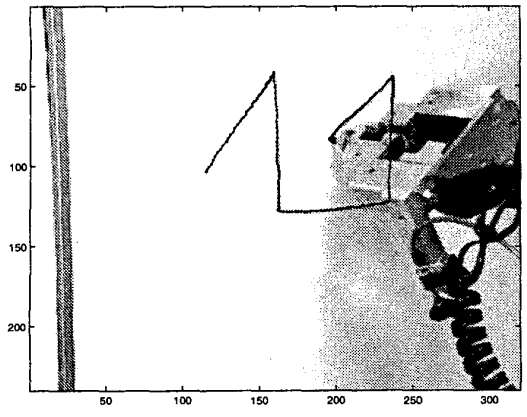

Figure 7: Pen trajectory, camera 2.

bration accuracy.

We have used hybrid force/position control but impedance control would apply in the same way. Elaborate specification of robot skills will require hybrid dynamic systems descriptions both for analysis and design.

5 Conclusion

In this paper, we have developed a method for combining visual servoing methods with force control, based on explicit estimation of the position of an unknown planar constraint surface. The method differs from previous work [2] in that it does not rely on assumptions of negligible friction, or the possibility to recover the normal of the plane from accurate measurements of contact forces and torques. Instead, we use data from a calibrated robot and camera system to estimate the constraint location. The main drawbacks of this approach are that it requires the constraints to be (piecewise) planar surfaces, and that calibration is required.

Our approach involves an image based visual servoing system with an explicit planar constraint on the possible velocity screws $\dot{\mathbf{r}}_c$, used to generate reference trajectories for the force control. The method is shown to work in practise in experiments involving force controlled, vision guided drawing on a planar surface. The system is run at a sampling frequency of 15 Hz, which could easily be increased to the the video rate of 30 Hz.

References

[1] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," in *IEEE Transactions on Robotics and Automation*, October 1996, vol. 12, pp. 651–670.

[2] D. Xiao, B. Ghosh, N. Xi, and T.J. Tarn, "Intelligent robotic manipulation with hybrid position/force control in an uncalibrated workspace," in *Proceedings IEEE International Conference on Robotics and Automation*, Leuven, Belgium, May 1998, pp. 1671–1676.

[3] G. Morel, E. Malis, and S. Boudet, "Impedance based combination of visual and force control," in *Proceedings IEEE International Conference on Robotics and Automation*, Leuven, Belgium, May 1998, pp. 1743–1748.

[4] J. Baeten, H. Bruyninckx, and J. De Schutter, "Combining eye-in-hand visual servoing and force control in robotic tasks using the task frame," in *Proceedings IEEE International Conference on Multisensor Fusion*, Taipei, Taiwan, August 1999, pp. 141–146.

[5] A. Pichler and M. Jägersand, "Uncalibrated hybrid force-vision manipulation," in *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2000, vol. 3, pp. 1866–1871.

[6] K. Hosoda, K. Igarashi, and M. Asada, "Adaptive hybrid visual servoing/force control in unknown environment," in *Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1996, vol. 3, pp. 1097–1103.

[7] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1999, vol. 1, pp. 666–673.

[8] R. Tsai and R. Lenz, "A new technique for fully autonomous and efficient 3d robotics hand/eye calibration," in *IEEE Transactions on Robotics and Automation*, vol. 5, pp. 345–358. June 1989.

[9] F. Chaumette and E. Malis, "2 1/2 d visual servoing: a possible solution to improve image-based and position-based visual servoings," in *Proceedings IEEE International Conference on Robotics and Automation*, San Francisco, CA, April 2000, pp. 630–635.

[10] P. Martinet and E. Cervera, "Stacking jacobians properly in stereo visual servoing," in *Proceedings IEEE International Conference on Robotics and Automation*, Seoul, Korea, May 2001, pp. 717–722.

[11] K. Nilsson, *Industrial Robot Programming*, Ph.D. thesis, Lund Institute of Technology, 1996.