A Kalman-Filter Approach for the Multirate-Control Problem in Visual Servoing Systems

Wolfgang Höllmüller
Technische Universität Wien
w.hoellmueller@gmx.net

Norbert Härle
Queensland University of Technology, EESE, Brisbane
n.harle@qut.edu.au

Abstract

We propose a control approach for feature-based visual servoing applied to eye-in-hand systems. We address a Kalman filter based method to increase the dynamics of a visual sensor in control loops and so avoid the multi-rate problem. We show the capability of the proposed method with experimental results embedded in a hierarchical servoing structure.

1 Introduction

Visual feedback in a control loop can either be *position based* or *feature based.* This classification was proposed by Weiss in 1980 [Weiss and Sanderson, 1980]. According to this classification, feature based visual servoing approach is used in this paper and is shown in Figure 1.

The position based method estimates global positions using features and runs in a Cartesian control space.

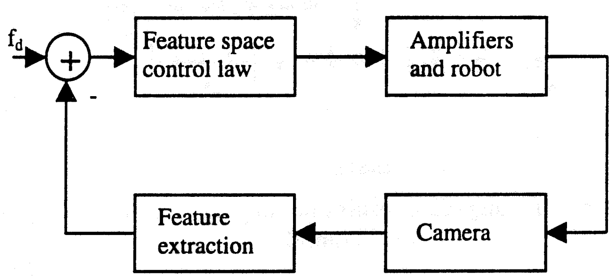

Figure 1: Feature based visual servoing structure

In contrast to that, the feature-based method feeds extracted image features directly back into the control loop, so the controller operates in the image space (feature space) and therefore the following criteria are desirable:

- Feature extraction period synchronised with robot control
- Low noise output from feature extraction
- Linearity of sensor

Frame rates of standard cameras (CCIR, RS140) lie between 25 and 30 Hz respectively and feature extraction generally decreases the image rate even further, which is generally too small to be used for control purposes. In addition to that, the extracted feature vector and the controller are not synchronised, which introduces a multi-rate problem to the control loop. The effect of multi-rate sampling is difficult to analyse and is generally not fully covered in textbooks. Corke [Corke, 1996] uses a pragmatic approach in modelling the multi-rate sampler as a single-rate sampler with a time delay. Nemani [Nemani *et al.*, 1994] addresses the problem by modelling the camera as a discrete time delayed sensor and analyses the system through the lifting technique, which converts the periodic time-varying multi-rate system to a time-invariant one.

Feddema [Feddema *et al.*, 1991] uses a trajectory generator operating in feature space instead of using the features directly for feedback. Online image Jacobian estimation has been investigated by Jägersand [Jagersand *et al.*, 1996] and Corke [Corke, 1996] uses a Kalman filter to estimate target positions.

In this paper the multi-rate problem is avoided by introducing a Kalman filter into the control loop, which is synchronised with the frame rate, but samples at a rate suitable for the controller. The filter estimates and predicts feature positions for control. Extracting features from images is computationally time consuming, so one approach to overcome this problem is to use specialised and therefore fast hardware, but it is highly inflexible and expensive. Apart from that, using more complex feature extraction techniques brings up the problem of a slow sensor in a control loop and therefore estimation and prediction of features using software is an approach worth investigating. Especially now, that "off the shelf" computers have quasi parallel processing capability and can process large amount of data in "reasonable" time[1]. This parallel processing capabilities gives the opportunity to process images in different depths, from fast and simple to slow and sophisticated algorithms. Here, we also want to discuss a two stage visual servoing software structure.

1 The expression "real-time" is left out deliberately, because no timing description is given.

2 Software Structure

In this section we briefly introduce the software structure placed around the control system as shown in Figure 2. It consists of two main parts, which process the image at different rates. Besides that, a database for different objects consisting of feature, control and general hardware information is provided.

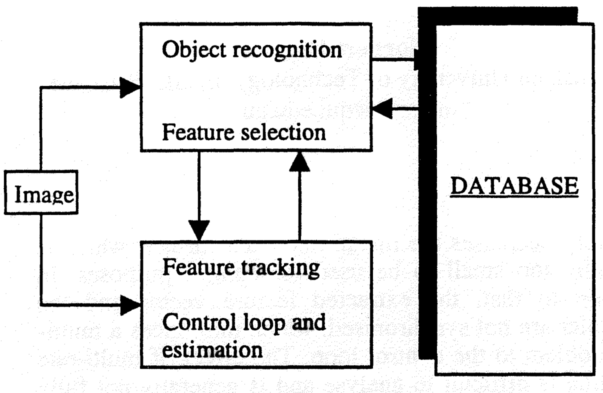

Figure 2: Two stage visual servoing structure

2.1 Feature tracking and control loop

This is the task we concentrate on, its aim is to track the object using simple features like edges, centrepoints of areas or circles. This unit could be split up into several parallel tracking algorithms depending on the number of degrees of freedom (DOF) available with the robot. Some parameters have to be supplied from the object recognition unit:

- The kind of features to be tracked
- Initial position of features
- Quality measures of features
- Control parameters

The feature tracker itself gives position information back to the logically higher object recognition part.

2.2 Object recognition

This task uses the feature information supplied by the database and finds the initial positions of the features to be tracked or grabbed. After this initialisation step it checks the validity of the positions found by the tracker and supplies quality measures of features. Such measures are:

- Slope of edges in grey scale images
- Length of a line (edge)
- Measures of ellipsoids
- Smoothness of feature movement

3 Control Structure and Kalman Filter

3.1 General

Compared with the feature based visual servoing structure defined by Weiss (see Introduction), a Kalman Filter [Grewal and Andrews, 1993] is added, see Figure 3.

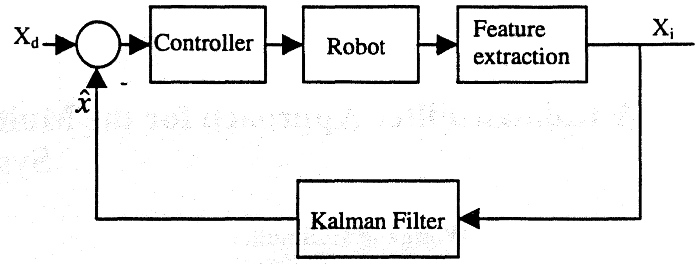

Figure 3: Control structure

Typically the control sampling rate is much bigger than the image rate, so the Kalman filter is the link between these two rates (Figure 4) and is used to predict the position ($\hat{x}$) of the feature for the following image, knowing all the prior positions. In this structure, feature extraction doesn't need to be at frame rate, because the Kalman filter can predict more than one step, but precision decreases with increasing prediction intervals, because of the increased measurement lag. Also the input period of the filter doesn't need to be constant for the same reason. It is sufficient that the filter runs at a constant frequency.

The controller uses the recent and the predicted position of the filter and interpolates additional points, depending on the sampling rate of the Kalman filter, in order to achieve the necessary control rate (Figure 4). In the end the frame rate is not visible for the controller any more and so we have a single rate control loop.

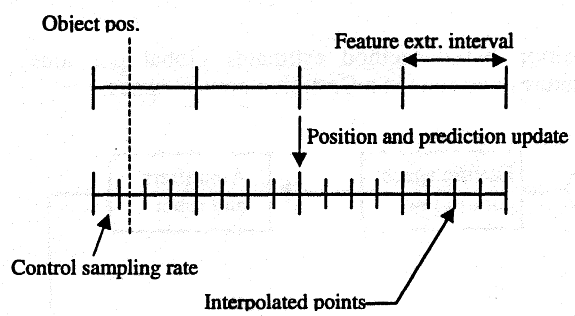

Figure 4: Timing relationships in image processing and robot control

Looking at timing relationships between image processing and robot control, Figure 4 shows, that less than two intervals after an object position changes, the position and the filter prediction parameters are updated, so the delay is kept as small as possible. The error made with the prediction is mainly due to changing object movement and not due to robot motion, provided that the robot model is accurate enough.

3.2 Kalman Filter

The filter has to estimate positions according to robot movement and object movement. The image only supplies the position of the object relative to the camera, so the filtering parameters for object movement have to be extracted out of the measured object position and the estimated position according to robot movement only. Therefore it is crucial to use an accurate model of the robot (especially the camera lens).

A third order model of the target motion can be formulated as follows:

$$\underline{x}_k = [x_k \quad \dot{x}_k \quad \ddot{x}_k]^T, \tag{1}$$

$$\underline{x}_{k+1} = \mathbf{A}\underline{x}_k, \tag{2}$$

$$y_k = \mathbf{C}\underline{x}_k, \tag{3}$$

$$\mathbf{A} = \begin{bmatrix} 1 & T & T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}, \tag{4}$$

$$C = [1 \quad 0 \quad 0]. \tag{5}$$

T is the sampling period of the filter.

The general Kalman filter equations following Balakrishnan's notation [Balakrishnan, 1987] are:
(between observations)

$$\hat{\underline{x}}_{k+1|k} = \mathbf{A_k}\hat{\underline{x}}_{k|k} + \mathbf{B_k}\underline{u}_k, \tag{6}$$

$$\mathbf{P_{k+1|k}} = \mathbf{A_k P_{k|k} A_k^T} + \mathbf{Q_k}, \tag{7}$$

(at observation)

$$\underline{K}_{k+1} = \mathbf{P_{k+1|k} C_{k+1}^T} \left[\mathbf{C_{k+1} P_{k+1|k} C_{k+1}^T} + R_{k+1}\right]^{-1}, \tag{8}$$

$$\hat{\underline{x}}_{k+1|k+1} = \hat{\underline{x}}_{k+1|k} + \underline{K}_{k+1}(z_{k+1} - \mathbf{C_{k+1}}\hat{\underline{x}}_{k+1|k}), \tag{9}$$

$$\mathbf{P_{k+1|k+1}} = \mathbf{P_{k+1|k}} - \underline{K}_{k+1}\mathbf{C_{k+1}P_{k+1|k}}. \tag{10}$$

The model for the robot motion will be illustrated at the experimental results because it is application dependent.
We have divided the filtering task into two parts acquiring the filter input from different sources at different rates (image rate, control rate). Therefore each filter should use it's own input rate and in the end be synchronised to the control sampling time. In that case, only the output from the filter simulating object movements needs to be interpolated, because the robot movement based filter is at control rate already.

It was stated above that the filter is capable of predicting more than one step. This is done by adjustment of the process and observation noise covariances online according to the step. In the case of a prediction step the observation noise covariance R_k must be big and Q_k small to dismiss the measurement. The opposite applies in a measurement step. A problem arises, when the prediction interval increases because the measurement information is not up to date any more. This problem is only significant when the object moves.

4 Experimental results

In this chapter we give a brief description of the used robot system and present the preliminary experimental results.

4.1 Hardware

The described ideas have been implemented on an Andros 4x4 mobile robot with a manipulator arm and a RS170 standard camera mounted at the end of it. The arm itself has 3 DOF, but the experiments have been carried out on only one DOF so far. Control commands are transferred to the robot via a serial interface (Figure 5) with a sampling period of 50 ms, which is much bigger than the one of industrial robots used in other papers (Puma) and so comparison of results is difficult due to differences in dynamics. In this configuration no interpolation between estimated feature positions is needed, because the control period is bigger than the frame rate.

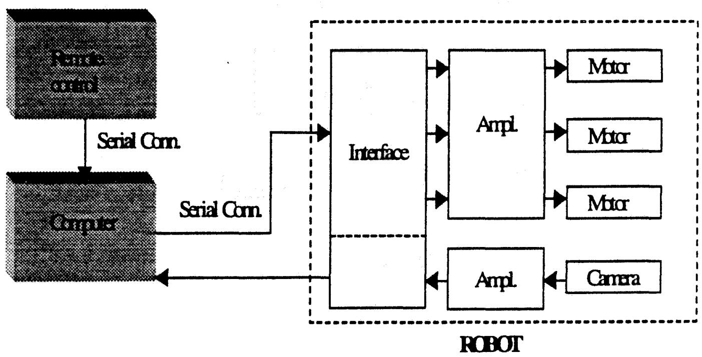

Figure 5: Description of hardware

For image processing a XPG1000 (Dipix) frame grabber card is used. The signal processor on this grabber card is programmed directly in order to maintain portability of the algorithm and for stability reasons.

4.2 Plant Identification

As mentioned above, the accuracy of the Kalman filter depends highly on the model of the robot, especially the lens of the camera. The gain of the camera (Figure 6) depends on the distance of the camera from the rotation axis r, on the target angle α and on the target distance d besides the focal length of the lens f:

$$K_{lens} = f\frac{\tan(\alpha) + \frac{2r}{l}\tan(\frac{\alpha}{2})}{1 - \frac{2r}{l}\tan(\frac{\alpha}{2})\tan(\alpha)} \qquad \text{Eq. 11}$$

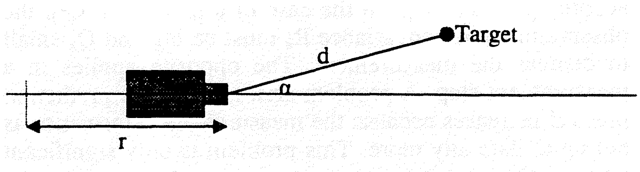

Figure 6: Geometrical relations of camera

We modelled the object motion depending on the speed of the robot arm, in particular the velocity of one joint. The model is of 6^{th} order, because the mechanics of the arm produce a delay time of approximately 6 times the image frame rate. Because of the fact, that the model for this particular application is processed at this rate (Kalman filter), the delay time is set to 6. The state-vector represents the position x of the feature at decreasing times:

$$\underline{X}_k = [x_{k-1} \ \cdots \ x_{k-6}]^T \qquad (12)$$

The state matrices are:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad (13)$$

$$\mathbf{B} = [0 \ \ 0 \ \ 0 \ \ 0 \ \ 0 \ \ K_{lens}]^T \qquad (14)$$

$$\mathbf{C} = [1 \ \ 0 \ \ 0 \ \ 0 \ \ 0 \ \ 0] \qquad (15)$$

4.3 Results

The results presented here assume that the object (feature) is not moving. The target object is a white circle on a black background and the position of the centre is used as the feature to track. A sinusoidal set point with a frequency of 0.4 Hz and an amplitude of 200 pixel is applied to the controller. The controller has a PDT_1 characterisic with a small derivative part.

In Figure 7 the experimental tracking performance with a measurement interval of 30ms (frame interval) is shown. The solid line is the set point and the dotted curve represents the output of the Kalman filter. The lag characteristic is evident and in this setup the filter only compensates the noise from measurements. The error is shown in Figure 8 and has a maximal value of approximately 180 pixel. This error was expected for a controller without an integral part, which can also be evaluated by simulation. The error is primarily a question of the controller and is not investigated here, but a smaller error can be achieved through the use of feed forward control as shown by Corke [Corke, 1996].

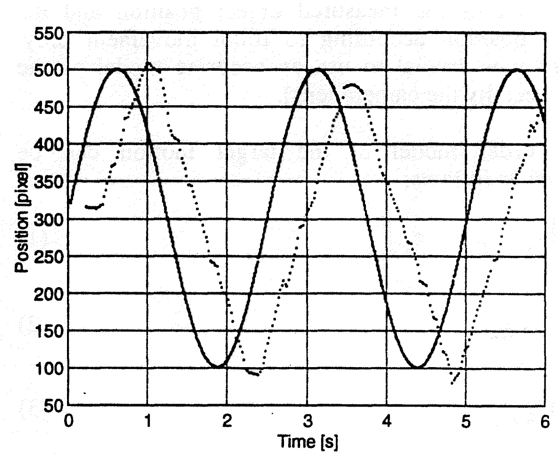

Figure 7: Experimental tracking performance with a measurement interval of 30ms

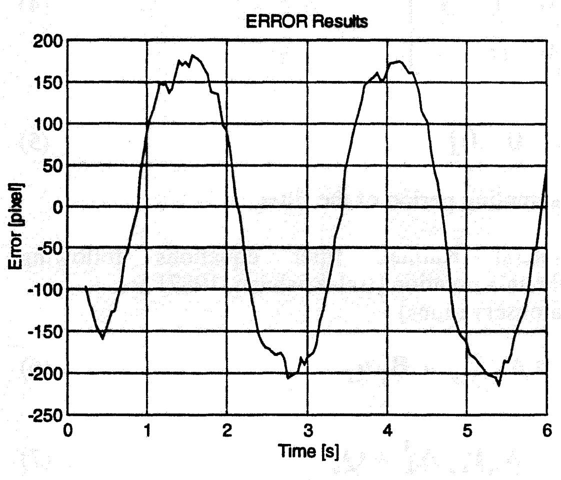

Figure 8: Control error from Figure 7

The same experiment with a measurement interval of 300ms is shown in Figure 9 and Figure 10. Comparing these results with the first experiment shows, that the difference is insignificant. In particular the error is almost the same as in the first experiment although only every 10^{th} measurement was taken into account. This is an indication for the appropriate model of the robot.

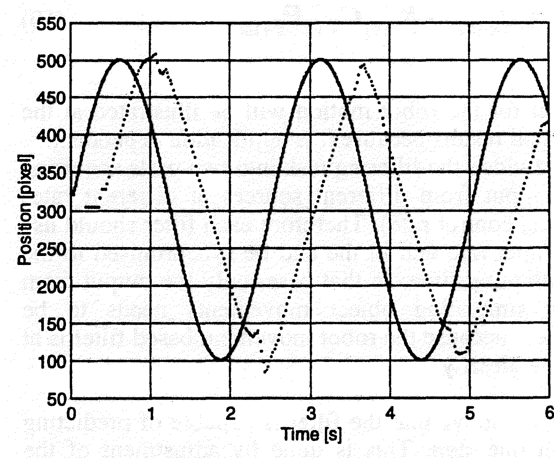

Figure 9: Exper. tracking perform. (300ms)

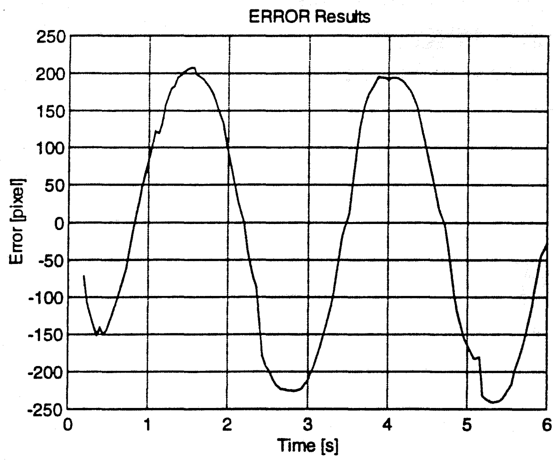

Figure 10: Control error from Figure 9

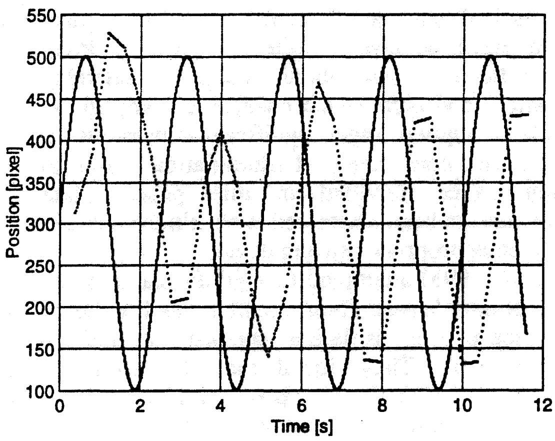

Figure 11: Tracking performance without Filter and a measurement interval of 300ms

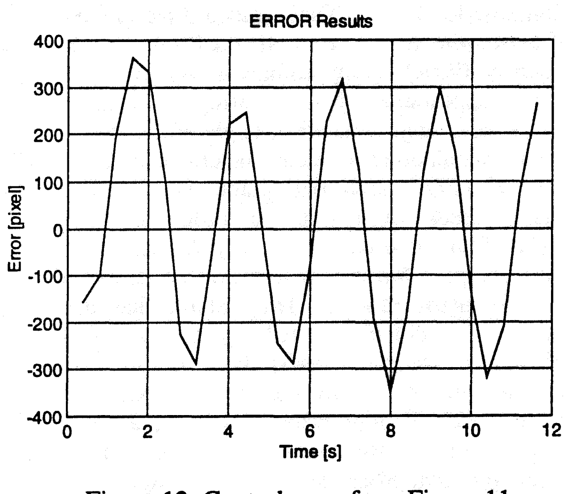

Figure 12: Control error from Figure 11

In the last experiment a controller with a measurement interval of also 300ms, but without a Kalman filter was used and the results are illustrated in Figure 11 and Figure 12. The robot arm can only roughly follow the set point and the controller operates only at positive and negative saturation. Therefore the error is increased as well. In Figure 11 and Figure 12 values were interpolated between the actual measurement points for illustration purposes only.

5 Conclusion

The low sampling rate of visual sensors due to a huge amount of data in the sensor itself and its time to be processed is one of the biggest problems in visual servoing and makes dynamic control difficult. The speed of such sensors depends highly on the image processing time, which is very likely to vary from one frame to the other. Separating the sensor from the control loop and using prediction and smoothing techniques is a possible way to overcome the lack of speed of image sensors, the multi rate problem associated with it and noise. In this paper an approach using a Kalman filter was introduced and the results so far are very promising, because the introduced system is insensitive to low and varying measurement rates. It suggests the use of more sophisticated image processing techniques even for tracking. However, any behaviour of the robot different from the model used for the Kalman filter, e.g. unexpected disturbances, inaccuracies, etc. can only be corrected at image rate. In that case, additional sensor data is needed to guarantee an improved reaction time. Moving objects have not been taken into account yet, which will restrict the performance of the system to a certain degree. The very briefly introduced software structure is a basis for visual servoing tasks and can be extended to match many different tasks.

References

[Balakrishnan, 1987] Balakrishnan A. V. Kalman Filtering Theory, 1987

[Corke, 1996] Corke Peter I. Visual Control of Robots, 1996 , John Wiley & Sons Inc. Robotics and Mechatronics Series, Prof.J.Billingsley

[Feddema et al., 1991] Feddema J.T., Lee C.S.G., and Mitchell O.R. Weighted Selection of Image Features for Resolved Rate Visual Feedback Control, *IEEE Transactions on Robot. and Automation*, 7(1): 31-47, 1991

[Grewal and Andrews, 1993] Grewal M.S. and Andrews A.P.Kalman Filtering - Theory and Practice, 1993

[Jagersand et al., 1996] Jagersand M., Fuentes O., and Nelson R., Experimantal evaluation of uncalibrated visual servoing for precision manipulation, *IEEE International Conference on Robotics and Automation*, 1996

[Nemani et al., 1994] Nemani M., Tsao T.-C., and Hutchinson S., Multi-Rate Analysis and Design of Visual Feedback Digital Servo-Control System, *Journal of Dynamic Systems, Measurement and Control*, 116 45-55, 1994

[Weiss and Sanderson, 1980] Sanderson A.C. and Weiss L.E., Image based visual servo control using relational graph error signals., *Proc. IEEE*, 1074-1077, 1980