IEEE Int. Conf. on Robotics and Automation ICRA2005, Barcelona, Spain, April 2005

Calibration and Synchronization of a Robot-Mounted Camera for Fast Sensor-Based Robot Motion

Friedrich Lange and Gerd Hirzinger
Deutsches Zentrum für Luft- und Raumfahrt e. V. (DLR)
Oberpfaffenhofen, D-82234 Weßling, Germany
http://www.robotic.dlr.de/Friedrich.Lange/

***Abstract*— For precise control of robots along paths which are sensed online it is of fundamental importance to have a calibrated system. In addition to the identification of the sensor parameters - in our case the camera calibration - we focus on the adaptation of parameters that characterize the integration of the sensor into the control system or the application. The most important of such parameters are identified best when evaluating an application task, after a short pre-calibration phase. The method is demonstrated in experiments in which a robot arm follows a curved line at high speed.**

***Index Terms*— visual servoing, adaptation, calibration, self calibration, synchronization**

I. Introduction

We discuss the task of an off-the-shelf industrial robot manipulator which has to be servoed along unknown curved lines as e.g. cables or edges of workpieces (see Fig. 1). This sample task stands for a variety of applications as the spraying of the glue for a sealing or the cutting of parts using a laser. Such visual servoing task have to be realized at high speed. Therefore in addition to an appropriate camera we need a special control approach.

In former papers, [1], [2], the authors proposed a universal sensor control architecture (see Fig. 3). This approach consists of two levels: In the lower level a predictive positional control is implemented. This control completes the control setup provided by the robot manufacturer by adding an adaptive feedforward controller. The upper level defines the desired trajectory using image data. The interface between the two levels is not only represented by the current desired and actual pose of the tool center point (tcp). In addition the part of the desired trajectory which has to be executed in the next sampling steps is provided. This enables the feedforward controller to realize an *ideal robot* which accurately follows the desired path without delay.

The desired trajectory is determined by a camera which is mounted near the tcp. Since the positions of sensed object points $\mathbf{x}_s$ are acquired with respect to the camera pose, the components of a desired position of the tcp $\mathbf{x}_{d(k+i)}$ result from the actual position of the camera $\mathbf{x}_{c(k)}$ at the time instant k of the exposure and from image points which can be matched to object points corresponding to sampling step $(k+i)$ (see Fig. 2). Assuming that the camera is always aligned with the world system this yields

$$\mathbf{x}_{s(k+i)} = \mathbf{x}_{c(k)} + {}^{c(k)}\mathbf{x}_{s(k+i)}, \tag{1}$$

where an upper left index denotes the system to which the point is related. This means that ${}^{c(k)}\mathbf{x}_{s(k+i)}$ is the difference between the point $\mathbf{x}_{s(k+i)}$ and the camera. ${}^{c(k)}\mathbf{x}_{s(k+i)}$ is computed from the coordinates of image points measured without error, and from the given distance of the object points in time step $(k+i)$. The desired position $\mathbf{x}_{d(k+i)}$ is then computed from $\mathbf{x}_{s(k+i)}$ by adding the desired distance.

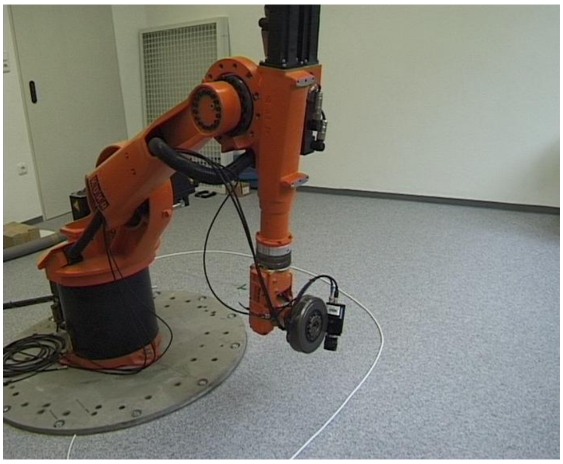

Fig. 1. The task: Refinement of the robot path so that the cable remains in the image center

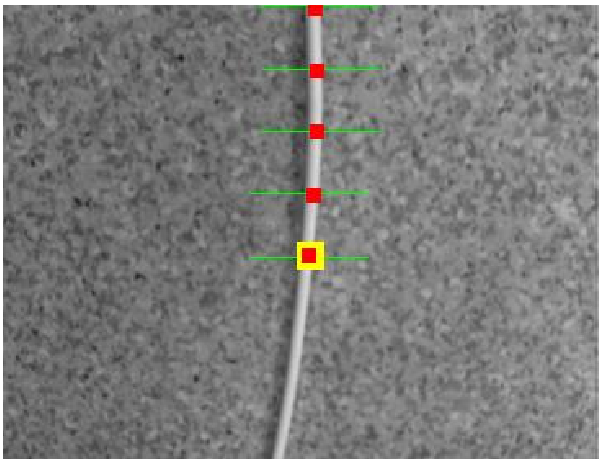

Fig. 2. View of the robot mounted camera: The line is visible for the current time instant (image center) and for future sampling steps

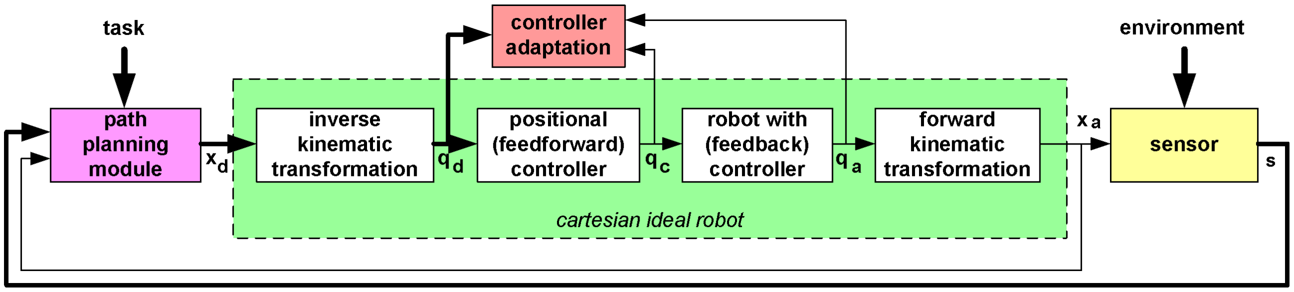
task
controller adaptation
environment
path planning module
x_d
inverse kinematic transformation
q_d
positional (feedforward) controller
q_c
robot with (feedback) controller
q_a
forward kinematic transformation
x_a
sensor
s
cartesian ideal robot

Fig. 3. Control architecture with positional control as lower level (green) and the determination of the desired path as upper level. (Bold face lines include predicted future values.)

Degrees of freedom of $\mathbf{x}_d$ which cannot be determined by image information are given in terms of a reference path.

Adaptation of the feedforward controller of the lower level is possible using predefined paths, i.e. without using the vision system. For this task the goal is to minimize the difference between the actual robot joint angles which can be internally measured and the desired joint values given by the reference path [3], [4]. Unfortunately, adaptation of the upper level und thus optimization of the total system cannot be easily reached.

According to Gangloff and de Mathelin [5] the reachable path accuracy when tracking planar lines by a robot mounted camera is limited substantially by the accuracy of camera calibration. Therefore in the past we used a thoroughly calibrated camera, but we could not reach path errors less than 1 mm [2]. So in this paper besides the camera calibration we concentrate on the adaptation of the parameters that represent the interaction between camera and control. We do so both to minimize the path errors caused by calibration errors and to reach high accuracy without time consuming manual calibration.

Since we want to minimize path errors of typical application tasks, for parameter adaptation we particularly consider those path errors which occur during normal use. In contrast to other self-calibration approaches, for our method a pre-calibration phase is advantageous.

Previous work exists mainly with respect to the calibration of only the camera parameters (see Sect. II-A). Other approaches focus on self-calibration (see e.g. [6]) or on temporal problems (see e.g. [7]) but ignore unknown camera parameters. We account for all these considerations and apply them to our architecture.

We begin with the identification of the unknown parameters (Sect. II). Then the adaptation of the total system is possible. Sect. III demonstrates the improved accuracy reached in experiments with the robot.

II. Identification of the unknown parameters

For the scope of this paper there are no external measurements available. So we have to restrict to tcp poses computed by joint angles and to images. This suggests a model-based approach.

The determination of the parameters is facilitated if a pre-calibration is executed before real application task. Such pre-calibrations reduce the number of unknown parameters for the main adaptation. Strictly speaking we distinguish 3 phases:

1) Identification of the camera parameters by static images
2) Identification of further parameters using a special test motion of the robot
3) Identification of the remaining relevant parameters using the application task

In the first phase we compute the camera parameters using a special setup. There are several images taken of the calibration object seen from different camera poses (see Sect. II-A). The parameters include the pose of the camera with respect to the tcp as well as lens distortion.

We assume that after compensation of these camera calibration errors there are still errors which have not been significant during the first calibration phase but that may have a crucial effect on the application task. Random errors as in [8] play a minor role in our scenario. Instead we consider

- a temporal difference between the capture of the images and the measurement of the joint angles by the robot (Because of an automatic shutter speed this time offset depends on the lighting),
- errors in the orientation of the camera ${}^{a}\mathbf{T}_c$ with respect to the tcp (arm pose), and
- errors in the scale of the mapping, caused by an erroneously estimated distance ${}^{c}z_s$ between camera and sensed line.

Similar to the methods using self-calibration (see e.g. [9]) these parameters are estimated by comparing different images when following the line at high speed. We use that the reconstruction of line points has to yield identical points for images taken from different camera poses. According to (1) the line points are computed by the location of the points in the image and by the camera pose which on its part is calculated by the forward kinematics of the robot and the measured joint angles.

Initially we command a test motion perpendicular to the line. This is the second of the three phases and will be outlined in Sect. II-B. Then the real application task of line following (phase 3, Sect.II-C and II-D) can be executed. This splitting is favorable since for single tasks the parameters are correlated. Robust identification

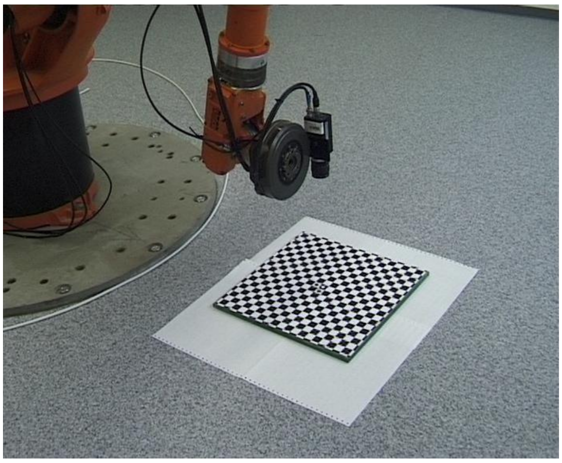

Fig. 4. Setup for camera calibration

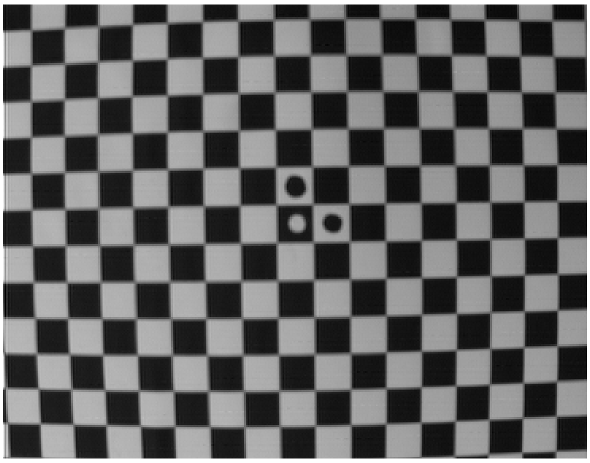

Fig. 5. Distorted view of the calibration object

therefore requires different motions.

A. Static calibration of the camera

The pose of the camera with respect to the robot system is referred to hand-eye calibration. It is given by the *external camera parameters*. The pose of the camera is defined by the optical axis, the robot system is given by the tcp pose. The *internal camera parameters* represent the pose of the ccd-chip with respect to the camera system as well as scale and lens distortion.

The internal parameters w_{ij} represent radial and not radial distortion according to a third order setup proposed by Weng et al. [10].

$$\begin{aligned} u &= w_{10} + w_{11} \cdot \xi + w_{12} \cdot \eta + w_{13} \cdot \xi^2 \\ &\quad + w_{14} \cdot \xi \cdot \eta + w_{15} \cdot \eta^2 + w_{16} \cdot \xi^3 \qquad (2) \\ &\quad + w_{17} \cdot \xi^2 \cdot \eta + w_{18} \cdot \xi \cdot \eta^2 + w_{19} \cdot \eta^3 \\ v &= w_{20} + w_{21} \cdot \xi + w_{22} \cdot \eta + w_{23} \cdot \xi^2 \\ &\quad + w_{24} \cdot \xi \cdot \eta + w_{25} \cdot \eta^2 + w_{26} \cdot \xi^3 \qquad (3) \\ &\quad + w_{27} \cdot \xi^2 \cdot \eta + w_{28} \cdot \xi \cdot \eta^2 + w_{29} \cdot \eta^3 \end{aligned}$$

u und v are the image coordinates of the original (distorted) image, ξ and η denote the values of the undistorted image, used in the sequel of this paper.

A toolbox [11] previously developed at the institute allows to identify the so defined external and internal camera parameters using different images of a calibration pattern (Fig. 5). These images are taken in each case from a stopped robot (Fig. 4). The poses of the tcp $\mathbf{T}_a$ when taking the individual images are assumed to be correctly known. The parameters w_{ij} are then computed by a least-square parameter fit using the Levenberg-Marquardt algorithm [12].

The resulting parameters ${}^c\mathbf{T}_a$ are used when computing the kinematics. The internal camera parameters contribute when the undistorted coordinates ξ and η of the detected line points are computed from the image.

The determination of the camera parameters cannot be done during the application task since only a calibration object with features distributed over the whole image allows a consistent parameter estimation. Otherwise a camera model would be built which reproduces the line points correctly only for the test motion. So if no camera calibration as described above is possible a simpler camera model has to be used, e.g. a radial model according to Tsai [13] which as well can represent the barrel distortion of Fig. 5. Svoboda [14] describes a method to determine such a model of lens distortion without using a calibration object. Instead of different images of a single camera, what we would suggest, he uses multiple cameras.

B. Execution of a test motion perpendicular to the line

After the very camera calibration a purely horizontal translation perpendicular to the line of Fig. 1 is executed. In contrast to the experiment of the next section here we do not use any visual feedback. In this experiment a fixed line point $\mathbf{x}_s$ is considered for all images. This especially allows to estimate the temporal offset $\Delta k = \hat{k} - k$ (in sampling steps) between the time instant of the exposure $\hat{k}$ and the internal positional measurement at the robot sampling step k. In addition, scaling errors Δz are identified.

For the chosen motion the camera orientation is constant and aligned with the world system. Then we have

$$\mathbf{x}_s = \mathbf{x}_{c(k)} + {}^{c(\hat{k})}\hat{\mathbf{x}}_s + \Delta k \cdot \left(\mathbf{x}_{c(k+1)} - \mathbf{x}_{c(k)}\right) + \Delta z \cdot {}^{c(\hat{k})}\hat{\mathbf{x}}_s \quad \forall k. \tag{4}$$

In contrast to the ideal value ${}^{c(k)}\mathbf{x}_s$ of (1) ${}^{c(\hat{k})}\hat{\mathbf{x}}_s$ is the sensed position which is computed from the image by the scalar equations of projection

$${}^{c(\hat{k})}\hat{x}_{s(k+i)} = \xi(\hat{k}, i) \cdot {}^{c(\hat{k})}z_{s(k+i)} \tag{5}$$

and

$${}^{c(\hat{k})}\hat{y}_{s(k+i)} = \eta(\hat{k}, i) \cdot {}^{c(\hat{k})}z_{s(k+i)} \tag{6}$$

using a given distance ${}^{c(\hat{k})}z_{s(k+i)}$. i is an index of a line point which will be explained in Sect. II-C. Here we only consider $i = 0$. This is the line point in the center image row of Fig. 2 .

If the lines are approximately parallel to the y-axis of the camera system we can replace the vectorial equation

(4) by the scalar equation of the x-component.

$$x_s = x_{c(k)} + {}^{c(\hat{k})}\hat{x}_s + \Delta k \cdot (x_{c(k+1)} - x_{c(k)}) + \Delta z \cdot {}^{c(\hat{k})}\hat{x}_s \quad \forall k, \tag{7}$$

or

$$\begin{aligned} x_s = {} & x_{c(k)} + {}^{c(\hat{k})}z_s \cdot \xi(\hat{k},0) \\ & + \Delta k \cdot {}^{c(k)}x_{c(k+1)} + \Delta z \cdot {}^{c(\hat{k})}z_s \cdot \xi(\hat{k},0) \quad \forall k. \end{aligned} \tag{8}$$

The parameters Δk and Δz are determined by parameter estimation. The unknown line position is not required if the equation with $k = 0$ is subtracted in each case. With $x_{c(k)} - x_{c(0)} = {}^{c(0)}x_{c(k)}$ this yields

$$\begin{aligned} 0 = {} & {}^{c(0)}x_{c(k)} + {}^{c(\hat{k})}z_s \cdot \xi(\hat{k},0) - {}^{c(\hat{0})}z_s \cdot \xi(\hat{0},0) \\ & + \Delta k \cdot ({}^{c(k)}x_{c(k+1)} - {}^{c(0)}x_{c(1)}) \\ & + \Delta z \cdot ({}^{c(\hat{k})}z_s \cdot \xi(\hat{k},0) - {}^{c(\hat{0})}z_s \cdot \xi(\hat{0},0)) \quad \forall k, \end{aligned} \tag{9}$$

where $\hat{0}$ denotes the time instant of the exposure corresponding to time step 0 of the robot.

C. Execution of the application task

The third experiment is the application task. This is a feedback controlled following of the online sensed line. In this case a line point $\mathbf{x}_{s(l)}$ corresponding to a sampling step $l = k + i$ is visible in the images of multiple time steps k. Parameter estimation is done for each line point by comparing all positions ${}^{c(\hat{k})}\hat{\mathbf{x}}_{s(l)}$ computed from the images of all time instants $\hat{k}$ in which the point is visible.

With constant orientation of the camera, according to (7) this yields the x-component of the camera system

$$\begin{aligned} x_{s(l)} = {} & x_{c(k)} + {}^{c(\hat{k})}\hat{x}_{s(l)} + \Delta k \cdot {}^{c(k)}x_{c(k+1)} \\ & + \Delta z \cdot {}^{c(\hat{k})}\hat{x}_{s(l)} - \Delta\alpha \cdot {}^{c(\hat{k})}\hat{y}_{s(l)} \quad \forall k. \end{aligned} \tag{10}$$

$\Delta\alpha$ is the error in the camera orientation around ${}^{c(\hat{k})}z$. ${}^{c(\hat{k})}\hat{x}_{s(l)}$ and ${}^{c(\hat{k})}\hat{y}_{s(l)}$ are computed using (5) and (6) where $\xi(\hat{k},i)$ and $\eta(\hat{k},i)$ denote the horizontal and the vertical projective image coordinate of a line point i of the image taken at time instant $\hat{k}$. The matching between image points and line points of a particular sampling step is done using the y-component of the reference trajectory expressed in camera coordinates (see [2]).

Equations (10) of different time instants $\hat{k}$ are compared similar to (9). Like this here in particular the error in the orientation $\Delta\alpha$ around the optical axis of the camera system is identified. This is extremely important since for the compensation of time-delays caused by the robot dynamics especially image points at the image border are used for feedforward control. In contrast, for a well suited control, scale errors have only little influence since the line position in the image is almost constant.

D. Application tasks with varying camera orientation

The preceding equations are not valid if the orientation of the camera is not constant and aligned with the world system. In that case we have to use transformation matrices $\mathbf{T}$ in homogeneous coordinates instead of the vectorial representations $\mathbf{x}$. Thus (10) is replaced by

$$\begin{aligned} \begin{pmatrix} \mathbf{x}_{s(l)} \\ 1 \end{pmatrix} & = \mathbf{T}_{c(\hat{k})} \cdot \mathbf{T}_{\alpha,z} \cdot \begin{pmatrix} {}^{c(\hat{k})}\hat{\mathbf{x}}_{s(l)} \\ 1 \end{pmatrix} \\ & = \mathbf{T}_{c(k)} \cdot {}^{c(k)}\mathbf{T}_{c(\hat{k})} \cdot \mathbf{T}_{\alpha,z} \cdot \begin{pmatrix} {}^{c(\hat{k})}\hat{\mathbf{x}}_{s(l)} \\ 1 \end{pmatrix} \quad \forall k. \end{aligned} \tag{11}$$

The interpolation for the time instant is

$${}^{c(k)}\mathbf{T}_{c(\hat{k})} = \mathbf{I} + \Delta k \cdot ({}^{c(k)}\mathbf{T}_{c(k+1)} - \mathbf{I}). \tag{12}$$

The matrix $\mathbf{T}_{\alpha,z}$ of the other parameters is set to

$$\mathbf{T}_{\alpha,z} = \begin{pmatrix} \mathbf{R}_\alpha \cdot (1 + \Delta z) & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \tag{13}$$

using

$$\mathbf{R}_\alpha = \begin{bmatrix} 1 & -\Delta\alpha & 0 \\ \Delta\alpha & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{14}$$

This yields

$$\begin{aligned} \begin{pmatrix} \mathbf{x}_{s(l)} \\ 1 \end{pmatrix} = {} & \mathbf{T}_{c(k)} \cdot [\mathbf{I} + \Delta k \cdot ({}^{c(k)}\mathbf{T}_{c(k+1)} - \mathbf{I})] \\ & \cdot \begin{pmatrix} \mathbf{R}_\alpha \cdot (1 + \Delta z) & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \cdot \begin{pmatrix} {}^{c(\hat{k})}\hat{\mathbf{x}}_{s(l)} \\ 1 \end{pmatrix}. \end{aligned} \tag{15}$$

Splitting the transformation matrices $\mathbf{T} = \begin{pmatrix} \mathbf{R} & \mathbf{x} \\ \mathbf{0}^T & 1 \end{pmatrix}$ into the rotation matrix $\mathbf{R}$ and the translation vector $\mathbf{x}$ this yields

$$\begin{aligned} \begin{pmatrix} \mathbf{x}_{s(l)} \\ 1 \end{pmatrix} = {} & \begin{pmatrix} \mathbf{R}_{c(k)} & \mathbf{x}_{c(k)} \\ \mathbf{0}^T & 1 \end{pmatrix} \\ & \cdot \left[\mathbf{I} + \Delta k \cdot \begin{pmatrix} ({}^{c(k)}\mathbf{R}_{c(k+1)} - \mathbf{I}) & {}^{c(k)}\mathbf{x}_{c(k+1)} \\ \mathbf{0}^T & 0 \end{pmatrix} \right] \\ & \cdot \begin{pmatrix} \mathbf{R}_\alpha \cdot (1 + \Delta z) & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \cdot \begin{pmatrix} {}^{c(\hat{k})}\hat{\mathbf{x}}_{s(l)} \\ 1 \end{pmatrix} \end{aligned} \tag{16}$$

and finally

$$\begin{aligned} \mathbf{x}_{s(l)} = {} & \mathbf{x}_{c(k)} + \mathbf{R}_{c(k)} \cdot \mathbf{R}_\alpha \cdot (1 + \Delta z) \cdot {}^{c(\hat{k})}\hat{\mathbf{x}}_{s(l)} \\ & + \mathbf{R}_{c(k)} \cdot \Delta k \cdot [{}^{c(k)}\mathbf{x}_{c(k+1)} + ({}^{c(k)}\mathbf{R}_{c(k+1)} - \mathbf{I}) \\ & \cdot \mathbf{R}_\alpha \cdot (1 + \Delta z) \cdot {}^{c(\hat{k})}\hat{\mathbf{x}}_{s(l)}] \quad \forall k, \end{aligned} \tag{17}$$

where for convenience, we do not use homogeneous coordinates any more.

Neglecting products of the small quantities $\Delta k, \Delta z$ and $\Delta\alpha$ yields

$$\begin{aligned} \mathbf{x}_{s(l)} = {} & \mathbf{x}_{c(k)} + \mathbf{R}_{c(k)} \cdot \mathbf{R}_\alpha \cdot {}^{c(\hat{k})}\hat{\mathbf{x}}_{s(l)} \\ & + \mathbf{R}_{c(k)} \cdot \Delta z \cdot {}^{c(\hat{k})}\hat{\mathbf{x}}_{s(l)} + \mathbf{R}_{c(k)} \cdot \Delta k \cdot [{}^{c(k)}\mathbf{x}_{c(k+1)} \\ & + ({}^{c(k)}\mathbf{R}_{c(k+1)} - \mathbf{I}) \cdot {}^{c(\hat{k})}\hat{\mathbf{x}}_{s(l)}] \quad \forall k. \end{aligned} \tag{18}$$

Neglecting changes in the orientation within a single sampling step $\left({}^{c(k)}\mathbf{R}_{c(k+1)} - \mathbf{I}\right)$ and inserting (14) yields the x-component

$$
\begin{aligned}
x_{s(l)} \;=\;& x_{c(k)} \\
+\;& t_{c(k),00} \cdot \big({}^{c(\hat{k})}\hat{x}_{s(l)} - \Delta\alpha \cdot {}^{c(\hat{k})}\hat{y}_{s(l)} \\
& \qquad + \Delta z \cdot {}^{c(\hat{k})}\hat{x}_{s(l)} + \Delta k \cdot {}^{c(k)}x_{c(k+1)}\big) \\
+\;& t_{c(k),01} \cdot \big({}^{c(\hat{k})}\hat{y}_{s(l)} + \Delta\alpha \cdot {}^{c(\hat{k})}\hat{x}_{s(l)} \\
& \qquad + \Delta z \cdot {}^{c(\hat{k})}\hat{y}_{s(l)} + \Delta k \cdot {}^{c(k)}y_{c(k+1)}\big) \\
+\;& t_{c(k),02} \cdot \big({}^{c(\hat{k})}\hat{z}_{s(l)} \\
& \qquad + \Delta z \cdot {}^{c(\hat{k})}\hat{z}_{s(l)} + \Delta k \cdot {}^{c(k)}z_{c(k+1)}\big) \quad \forall k,
\end{aligned}
\tag{19}
$$

where $t_{c(k),i_1 i_2}$ denotes the element i_1, i_2 of the transformation matrix $\mathbf{T}_{c(k)}$. For $\mathbf{R}_{c(k)} = \mathbf{I}$, equation (19) is identical to (10).

Now we express (19) in the camera system of time step l. Inserting (5) and (6) and subtracting the equation of $k = l$, i.e. the equation with $i = 0$ and $\hat{k} = \hat{l}$, yields

$$
\begin{aligned}
& {}^{c(l)}x_{c(k)} \\
& + {}^{c(l)}t_{c(k),00} \cdot \Big[\big((1+\Delta z)\cdot\xi(\hat{k},i) - \Delta\alpha\cdot\eta(\hat{k},i)\big)\cdot{}^{c(\hat{k})}z_{s(l)} \\
& \qquad + \Delta k \cdot {}^{c(k)}x_{c(k+1)}\Big] \\
& + {}^{c(l)}t_{c(k),01} \cdot \Big[\big((1+\Delta z)\cdot\eta(\hat{k},i) + \Delta\alpha\cdot\xi(\hat{k},i)\big)\cdot{}^{c(\hat{k})}z_{s(l)} \\
& \qquad + \Delta k \cdot {}^{c(k)}y_{c(k+1)}\Big] \\
& + {}^{c(l)}t_{c(k),02} \cdot \big[(1+\Delta z)\cdot{}^{c(\hat{k})}z_{s(l)} + \Delta k \cdot {}^{c(k)}z_{c(k+1)}\big] \\
& = \big((1+\Delta z)\cdot\xi(\hat{l},0) - \Delta\alpha\cdot\eta(\hat{l},0)\big)\cdot{}^{c(\hat{l})}z_{s(l)} \\
& \qquad + \Delta k \cdot {}^{c(l)}x_{c(l+1)} \quad \forall k.
\end{aligned}
\tag{20}
$$

We propose to determine Δk and Δz according to Sect. II-B since at least Δz cannot be determined with a small control error. Then, here we can restrict on the estimation of $\Delta\alpha$. After convergence we try to improve the result by refining Δk as well.

The estimated parameters can be applied in the camera module for the interpolation of the time instant of the exposure and for the mapping. Then here in the estimation module the remaining calibration errors will be $\Delta z = 0$ und $\Delta k = 0$. In this case we get $\Delta\alpha$ without the above mentioned approximations using

$$
\begin{aligned}
& {}^{c(l)}x_{c(k)} + {}^{c(l)}t_{c(k),00}\cdot\big(\xi(\hat{k},i) - \Delta\alpha\cdot\eta(\hat{k},i)\big)\cdot{}^{c(\hat{k})}z_{s(l)} \\
& \quad + {}^{c(l)}t_{c(k),01}\cdot\big(\eta(\hat{k},i) + \Delta\alpha\cdot\xi(\hat{k},i)\big)\cdot{}^{c(\hat{k})}z_{s(l)} \\
& \quad + {}^{c(l)}t_{c(k),02}\cdot{}^{c(\hat{k})}z_{s(l)} \\
& \quad = \big(\xi(\hat{l},0) - \Delta\alpha\cdot\eta(\hat{l},0)\big)\cdot{}^{c(\hat{l})}z_{s(l)} \quad \forall k.
\end{aligned}
\tag{21}
$$

Since ${}^{c(k)}y_{s(l)}$ and ${}^{c(k)}z_{s(l)}$ are given for each time step l by the reference trajectory, we can interpolate ${}^{c(\hat{k})}y_{s(l)}$ and ${}^{c(\hat{k})}z_{s(l)}$. Equation (6) then allows to compute $\eta(\hat{k},i)$ which has to be given to get $\xi(\hat{k},i)$ from the image. Thus each equation of a line point which is visible at time $\hat{k}$ and $\hat{l}$ yields a value for $\Delta\alpha$. These values have to be filtered to get the result.

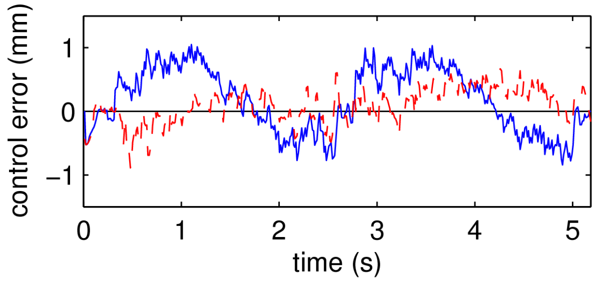

Fig. 6. Improvement by camera calibration when following a curved line: The solid blue line corresponds to #1 of table I, the dashed red line to #2

III. Experimental Results

The method is demonstrated using the cable tracking experiment of Sect. I. By compensation of the errors in the way explained in Sect. II the system is improved: With a well calibrated camera the mean path error is about 50 % of that of the experiment without calibration. Once again a reduction of almost 50 % is reached when adapting the other parameters by some runs using (9) and further runs with (21) thereafter. The results are listed in Table I and Figures 6 and 7, including experiments with a reduced number of adapted parameters. Run #7 would yield similar results with (20) and without (9), but then the adaptation would need about 10 runs to converge.

Table I demonstrates that for our scenario the estimation of the scale Δz and of the external camera parameters ${}^{c}\mathbf{T}_a$ may be omitted. The importance of the scale depends on the desired distance of the line from the image center. Important errors in the external camera parameters are compensated by other parameters. In contrast, without knowing the internal camera parameters w_{ij} no satisfying control is possible.

The parameter errors detected in the experiment are 0.010 s time offset, 0.005 rad orientational error, and

TABLE I

Remaining mean path deviations when following a curved line. (The errors are computed from the center image row. The maximum observed errors during the experiment are displayed in brackets. The lower three rows correspond to a changed layout of the line using the previously adapted parameters)

	compensation of errors in					image error	path error
	w_{ij}	${}^{c}\mathbf{T}_a$	Δk	$\Delta\alpha$	Δz	(pixel)	(mm)
1	-	-	-	-	-	2.31 (4.2)	0.99 (1.8)
2	+	+	-	-	-	1.05 (2.8)	0.53 (1.5)
3	-	-	+	+	+	2.20 (4.8)	0.95 (2.1)
4	+	-	+	+	+	0.52 (1.5)	0.20 (0.5)
5	+	+	-	+	+	0.96 (2.0)	0.48 (1.2)
6	+	+	+	-	+	0.67 (2.5)	0.39 (1.4)
7	+	+	+	+	-	0.53 (1.8)	0.33 (1.1)
8	+	+	+	+	+	0.51 (1.9)	0.32 (1.1)
11	-	-	-	-	-	1.95 (4.1)	0.84 (1.8)
12	+	+	-	-	-	1.40 (3.6)	0.64 (2.0)
13	+	+	+	+	+	0.47 (1.3)	0.26 (0.7)

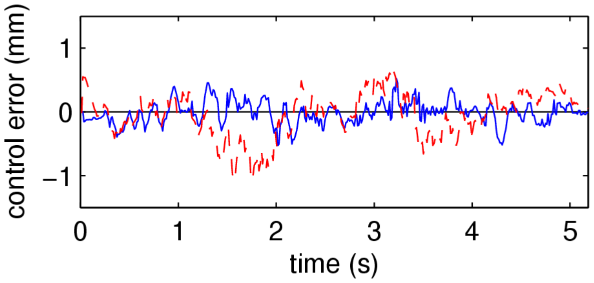

Fig. 7. Improvement by adaptation of the other parameters: The dashed red line corresponds to #12 of table I, the solid blue line to #13

10 mm distance error. These values are then used with a totally different layout of the line. This proves the portability of the adapted parameters (see lower part of Table I and Figures 7 and 8).

IV. Conclusion

We have developed a method that enables a robot manipulator to follow lines that are visible by a robot mounted camera. For high speed motion the parameters of the camera and of the interaction between robot and camera have been identified.

In contrast to the tracking of moving objects as described by [15], [16] the accuracy can be very high when following motionless edges since in this case there are no future accelerations of the target that could affect the performance. High performance, however, can only be reached if the vision system and the controller are tuned and synchronized. This has be done using the application task, thus yielding unbiased parameters.

The experiments documented in Sect. III do not show the calibration error directly. Instead they are the result from the cooperation of camera, vision algorithms, control, and the robot dynamics when executing a fast motion (0.7 m/s speed along the path) with huge a priori uncertainties (see Fig. 8). To the knowledge of the authors no other approach reaches such an accuracy at this speed. The achieved mean pixel error of half a pixel will certainly only be outperformed with high costs.

For further increase of the accuracy a high resolution camera will be required that combines precision in the horizontal image component (line position) with a wide range in the vertical component (prediction of the line points of future time steps). Besides, the robot hardware has to be improved. Alternatively an external measuring device might be used to survey the camera position. Otherwise joint elasticity which is present also with industrial robots, will limit the accuracy, as can be seen by (1).

Previous approaches [17] for compensation of joint elasticities when using the architecture of Fig. 3 propose external measuring devices for both adaptation of the ideal robot and determination of the real arm trajectory.

In contrast static deviations of the robot position caused by elasticity only play a minor role for fast sensor based motion. The same is true if the kinematic parameters are only roughly known.

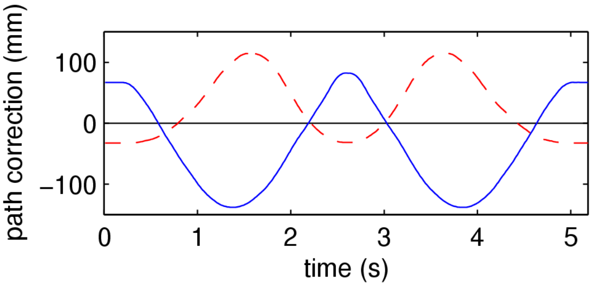

Fig. 8. Sensed path correction with respect to the reference path (The dashed red line corresponds to #13 of table I, the solid blue line to #8

References

[1] F. Lange and G. Hirzinger. A universal sensor control architecture considering robot dynamics. In *Proc. Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems*, pages 277–282, Baden-Baden, Germany, August 2001.

[2] F. Lange and G. Hirzinger. Spatial control of high speed robot arms using a tilted camera. In *Int. Symposium on Robotics (ISR 2004)*, Paris, France, March 2004.

[3] F. Lange and G. Hirzinger. Learning of a controller for non-recurring fast movements. *Advanced Robotics*, 10(2):229–244, April 1996.

[4] M. Grotjahn and B. Heimann. Model-based feedforward control in industrial robotics. *The International Journal on Robotics Research*, 21(1):45–60, January 2002.

[5] J. A. Gangloff and M. F. de Mathelin. Visual servoing of a 6-dof manipulator for unknown 3-d profile following. *IEEE Trans. on Robotics and Automation*, 18(4):511–520, August 2002.

[6] D. Caltabiano, G. Muscato, and F. Russo. Localization and self calibration of a robot for volcano exploration. In *Proc. 2004 IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 586–591, New Orleans, LA, April 2004.

[7] Y. Liu et al. A new generic model for vision based tracking in robotic systems. In *Proc. 2003 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 248–253, Las Vegas, Nevada, Oct. 2003.

[8] V. Kyrki, D. Kragic, and H. I. Christensen. Measurement errors in visual servoing. In *Proc. 2004 IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1861–1867, New Orleans, LA, April 2004.

[9] J.-M. Frahm and R. Koch. Robust camera calibration from images and rotation data. In *25th Pattern Recognition Symposium, DAGM'03*, Magdeburg, Germany, Sept. 2003. Springer-Verlag, Heidelberg.

[10] J. Weng, P. Cohen, and M. Herniou. Camera calibration with distortion models and accuracy analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(10):965–980, Oct. 1992.

[11] http://www.robotic.dlr.de/vision/projects/Calibration/, 2004.

[12] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C.* Cambridge University Press, 1988.

[13] R. Y. Tsai and R. K. Lenz. A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. *IEEE Trans. Robotics and Automation*, 5(3):345–358, 1989.

[14] Tomáš Svoboda. Quick guide to multi-camera self-calibration. Technical Report BiWi-TR-263, Computer Vision Lab, Swiss Federal Institute of Technology, Zürich, Switzerland, August 2003. http://cmp.felk.cvut.cz/˜svoboda/SelfCal/Publ/ selfcal.pdf.

[15] S. Hutchinson, G. D. Hager, and P. I. Corke. A tutorial on visual servo control. *IEEE Trans. on Robotics and Automation*, 12(5):651–670, Oct. 1996.

[16] A. Namiki, K. Hashimoto, and M. Ishikawa. A hierarchical control architecture for high-speed visual servoing. *The International Journal on Robotics Research*, 22(10-11):873–888, Oct-Nov 2003.

[17] F. Lange and G. Hirzinger. Learning accurate path control of industrial robots with joint elasticity. In *Proc. IEEE Int. Conference on Robotics and Automation*, pages 2084–2089, Detroit, Michigan, May 1999.