TRAJECTORY PLANNING AND STABILIZING USING VISUAL FEEDBACK AND TRANSPUTER NETWORK *

BOGDAN KWOLEK†, MARIAN WYSOCKI†

† Rzeszów University of Technology, Department of Control and Computer Engineering, W. Pola 2, PL-35 959 Rzeszów, Poland, bkwolek@prz.rzeszow.pl

Abstract. The paper describes development and experimental implementation of a visual control for the miniature laboratory robot Khepera. The robot and its stationary environment are observed by a static CCD camera, a collision free path in the two dimensional workspace is determined by artificial potential field methods and it is realized using real-time visual feedback. Computations are performed on T9000 new generation transputers. The system calibration, motion prediction and tracking, and trajectory stabilization for nonholonomic vehicle are considered.

Key Words. Visual feedback, motion tracking and prediction, trajectory stabilizing, real-time, transputer networks

1. INTRODUCTION

Computer vision plays important role in control of robots and mobile robots. Two general solutions are used in practical applications. Traditionally visual sensing and manipulation are combined in an open loop fashion look-than-move. As alternative is a visual feedback control loop which will increase the accuracy of the system. Usually, this approach is a fusion of high speed image processing, kinematics, dynamics, control theory and real-time computing. These aspects are considered in the paper, which describes development and experimental implementation of a visual control for the miniature laboratory mobile robot Khepera [9]. Determining a collision free path and its realization using real-time visual feedback are considered. Images obtained from a static CCD camera are used as visual input. Real-time computations are performed on a transputer network with T9000 new generation processors.

The organization of the paper is as follows. The system is described and main problems are characterized in the section 2. Solutions of the problems and experimental results are presented in the successive sections 3-7, which discuss camera calibration, collision-free path planning, motion tracking and prediction, and trajectory stabilization followed by the conclusions.

*) This research was supported by the Polish Commitee for Scientific Research (KBN) within the project PB/180/T11/97/12

2. SYSTEM DESCRIPTION AND PROBLEM FORMULATION

The system (see Fig. 1.) consists of the miniature laboratory robot Khepera [9] and a transputer network based on the PARSYS SN 94000 system with T9000 transputers.

The robot has a cylindrical shape with a diameter of 55 mm and a high of 30 mm. It uses two wheels for locomotion. Every wheel is driven by a DC motor controlled by the on board 68331 controller of Motorola, which communicates with external computers through an RS-232 serial line (we used the IrDA standard). The robot is observed by a static CCD camera connected to the transputer system through the home made frame-grabber based on the T425 transputer [6]. Registration of 25 frames 256x256 per second is possible.

Main computations are performed on two T9000 transputers [8]. Additional T800 transputer is reasponsible for communication with the PC computer. We prepared software for on line image display and for communication with MATLAB which makes the analysis of results easier.

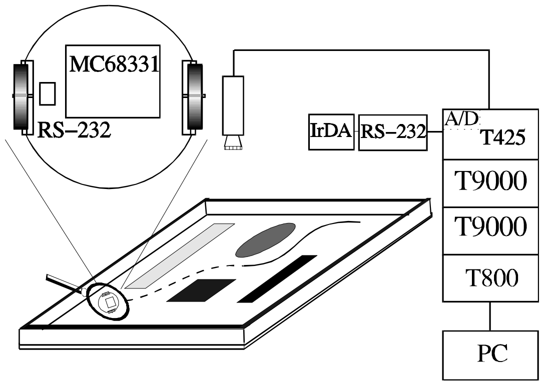

Fig.1. The system scheme.

Two main problems are considered:

- determining a collision free path of the robot in the two dimensional static workspace using visual observation
- stabilizing the robot on the given feasible trajectory using real-time visual feedback.

Important tasks related to these problems and considered in the paper are camera calibration and motion tracking and prediction. The first task is discussed in the next section.

3. CAMERA CALIBRATION

Camera calibration is the process of determining the internal geometric and optical characteristics of the camera (intrinsic parameters) and the three dimensional position and orientation of the camera relative to chosen world coordinate system (extrinsic parameters) We used the technique proposed by Tsai [4][7]. In the original method a calibration chart containing a lattice of identical black squares is placed in the view of the camera. The parameters of camera model are identified by least squares method on the basis of calibration points at the corners of the squares. Since the visual measurements of the Khepera robot play most important part in our problem of trajectory stabilization, we used the robot placed at 35 known places of the plane, instead of the calibration chart. It is explained in the Fig. 2a containing composed pictures of the robot. Positions of the robot center were used as calibration points. Determining of the robot center on the camera picture (done using moments) was preceded by Gaussian filtering, thresholding, morphological closing [5] (to remove white cracks on the robot picture (see Fig. 2a)) and segmentation. Extrinsic and intrinsic parameters were obtained by two stage linear regression (first for the translation and rotation parameters, then for lens distortion and the focal length) which prepared starting points for the Levenberg-Marquardt optimisation [10]. Distribution of the calibration errors is shown in the Fig. 2b. The mean of the absolute value of the error between the real and best fit location is 1.27 pixels with standard deviation of 0.78 pixels (0.122 cm, 0.076 cm, respectively).

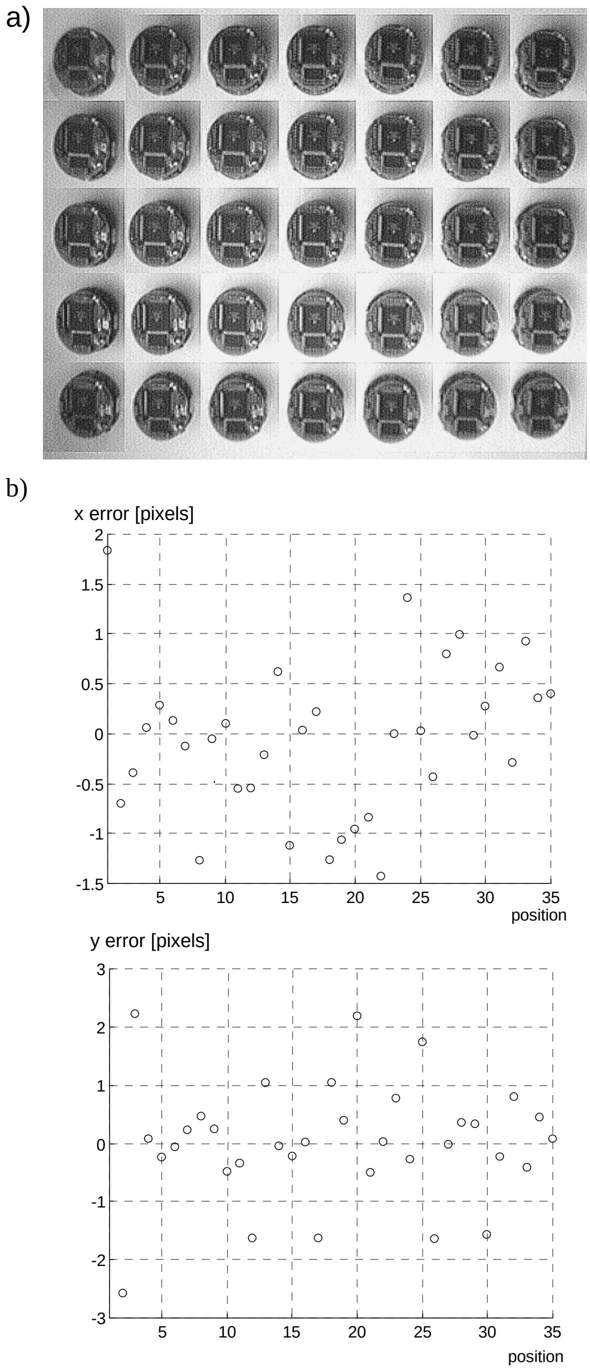

Fig. 2. Camera Calibration. a) composed picture of the robot at 35 known positions b) distribution of the errors.

4. TRAJECTORY PLANNING

Our trajectory planner uses the picture obtained from the static CCD camera. After Gaussian filtering, binarization and segmentation of the picture the obstacles are detected and artificially enhanced. This expansion is based upon some constant safety factor which represents the robot radius end enables the robot to be shrunk to a point in the two dimensional workspace. A collision-free path is computed using potential field methods [3].

Two algorithms have been implemented on the transputer network, i.e. NF1 and NF2 [3]. The first one computes the potential function U as the L^1 distance to the goal point in the discretized free subspace of R^2. The collision-free path follows the steepest descent from the initial point. This path is a minimum length path for the L^1 distance between the initial and goal point, but it in general grazes the enhanced obstacles what should be taken into account in the expansion procedure.

The NF2 method computes the skeleton of the free subspace, and then the potential U in the skeleton and in the rest of the free space. Wavefront axpansion algorithms are used. The path is computed by the best first method, which consist of iteratively constructing a tree whose nodes are points in the discretized free space. At every iteration the algorithm examines the neighbours of the leaf of the tree that has the smallest potential.

Fig. 3. shows the sample picture of the robot and its environment, equipotential lines and collision free-paths obtained by NF1 and NF2 methods.

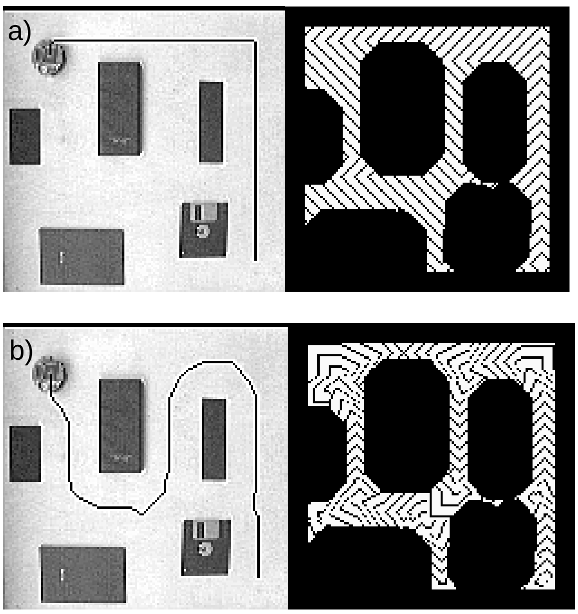

Fig. 3. Sample collision free paths and related equipotential lines, a) NF1, b) NF2.

Computational times obtained on two T9000 transputers for the 128x128 grid are 0.256 s (NF1) and 3.96 s (NF2).

Most of the computing time (about 75%) is occupied by finding best elements of various lists. This task can be distributed among processors that analyse shorter lists and compare results. Because of communication and synchronization, the speedup of that part of computations depends on the lengths of the shorter lists.

5. TRACKING OF THE ROBOT WITH KALMAN FILTER

To determine the position and orientation of the robot in real-time we cannot use the whole camera picture 256x256, but rather its small windows, say 50x50, covering the robot. The position of the center of the windows has to be predicted on the basis of robot positions identified in the past windows. Techniques using the kinematic model of the robot and extended Kalman filter (EKF) can be applied. A drawback of the EKF is that one has to compute its gains on line. To avoid that and to make us independent of the exact model of the moving object to be traced by visual feedback we used the following approximate model (see [2]),

$$\begin{aligned} \xi_k &= A\xi_{k-1} + w_k \\ \eta_k &= C\xi_k + v_k \end{aligned} \tag{1}$$

where k denotes the sample time ($t_{k+1} = t_k + T$; T is the sample period), $\xi_k = \left[X_k, \dot{X}_k, Y_k, \dot{Y}_k\right]^T$ is the system state $\eta_k = \left[X_k, Y_k\right]^T$ is the measurement, X_k and Y_k are the state coordinates of the robot center transformed to the camera picture, $\dot{X}_k, \dot{Y}_k$ are the velocities, w_k and v_k are disturbance noises assumed to be described by zero mean, Gaussian mutually independent noises with covariances Q and R, respectively.

The matrices A and C are

$$A = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2}$$

The recursive equation for the prediction of the windows centre is the following

$$\hat{\xi}_{k/k-1} = A\hat{\xi}_{k-1/k-1} \qquad (3)$$

where the estimates $\hat{\xi}$ are defined by the Kalman Filter algorithm

$$\hat{\xi}_{k/k} = \hat{\xi}_{k/k-1} + K_F\left(\eta_k - C\hat{\xi}_{k/k-1}\right) \qquad (4)$$

where the Kalman gain K_F can be computed off-line. The most important is proper selection of the input dynamic disturbance noise covariance matrix Q and the measurement noise covariance matrix R. R can be determined based on the analysis of the measurement noise. Selection of Q is more difficult because the robot motion is assumed arbitrary in this approach. We chose on the basis of practical experiments Q = diag(15.0, 15.0, 15.0, 15.0), R = diag(0.5, 0.5).

The Fig. 4. shows Kalman prediction errors in sample experiments. In the first case (Fig. 4a) the robot realised a circular path with a constant velocity 50mm/s and stopped, in the second one the path was a straight line segment and the velocity of the robot changed from 16 mm/s to 256 mm/s and from 256 mm/s to 0. The sampling period was 0.13 s.

a)

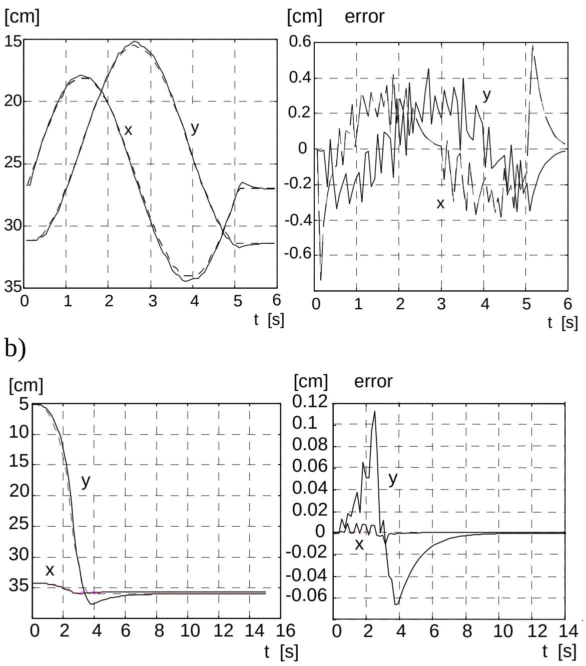

Fig. 4. Kalman prediction errors a) circular path, constant velocity b) linear path variable velocity.

6. TRAJECTORY STABILIZATION

6.1. Kinematics of the Khepera robot

The kinematic equations of the robot are given as

$$\dot{x} = u_1\cos\varphi,\ \dot{y} = u_1\sin\varphi,\ \dot{\varphi} = u_2 \qquad (5)$$

where the state of the system (5) $q = [x, y, \varphi]^T$ is the position of the wheel axis center (x, y) and the robot orientation φ with respect to the x-axis.

The control variables u_1 and u_2 are respectively, the tangent and angular velocities, and are related to the wheel velocities in the following manner

$$u_1 = \frac{1}{2}(u_R + u_L),\ u_2 = \frac{1}{2\Delta}(u_R - u_L).$$

The velocity u_R (u_L) is the tangent velocity of the right (left) wheel at its center of rotation (i.e. motor velocity times wheel radius). The distance between the point (x, y) and each of the wheel is Δ.

6.2. Reference trajectory and control problem

The reference trajectory is given by the functions $x_d(t)$ and $y_d(t)$ describing desired positions of the wheel axis center. We will assume that x_d and y_d are smooth and that the reference trajectory is feasible and can be realised with nominal controls

$$u_{1d} = \dot{x}_d\cos\varphi_d + \dot{y}_d\sin\varphi_d,\ u_{2d} = \dot{\varphi}_d \qquad (6)$$

where $\varphi_d = \arctan\left(\frac{\dot{y}_d}{\dot{x}_d}\right)$

The control problem consist in finding a control law, which will stabilize the system (5) to reference trajectory. It is well known, that the system (5) is nonholonomic (i.e. with nonitegrable velocity constraints). Literature results suggest that for nonholonomic systems, stabilizing about a trajectory is a better problem to consider than stabilizing to a point. In the paper [1] the linear time varying feedback law

$$u = u_d + K(t)e(t) \qquad (7)$$

has been proposed, where

$$u = [u_1, u_2]^T,\ u_d = [u_{1d}, u_{2d}]^T$$

$$e = [e_x, e_y, e_\varphi]^T$$

$$e_x = x_d - x,\ e_y = y_d - y,\ e_\varphi = \varphi_d - \varphi$$

and the gain matrix $K(t)$ is calculated on the basis of the solution of a Riccati type differential equation. The solution can be computed off line using time varying linear model obtained after linearization of

(5) about the reference trajectory. The control law (7) has been shown to locally exponentially stabilize the system to the desired trajectory [1]. A drawback of this approach is that each new reference trajectory requires time expensive computations for the gain *K(t)*. So we propose the control rule

$$u_1 = u_{1d} + k_{11}\left(e_x \cos\varphi + e_y \sin\varphi\right) +$$

$$u_{2d}\left(-e_x \sin\varphi + e_y \cos\varphi\right) \qquad (8)$$

$$u_2 = u_{2d} + k_{22}\left(-e_x \sin\varphi + e_y \cos\varphi\right) + k_{23} e_\varphi$$

which gave satisfactory results, both in simulations and in experiments with the real robot. Local asymptotic stability of the control system and selection of the gains k_{11}, k_{22}, k_{23} will be discussed in separate paper. We only remark that results known for linear time varying systems can be applied.

6.3. Experimental results

Estimates $\hat{x}_{k+1/k}$, $\hat{y}_{k+1/k}$ obtained on the basis of visual measurements, as described in the section 5, have been applied in the control laws (7) or (8). To predict the robot orientation φ we used the following algorithm

$$\hat{\varphi}_{k/k-1} = \alpha * \arctan\left(\frac{\hat{y}_{k/k-1} - \hat{y}_{k-1/k-1}}{\hat{x}_{k/k-1} - \hat{x}_{k-1/k-1}}\right) + (1-\alpha)\hat{\varphi}_{k-1/k-1}$$

$$\qquad (9)$$

$$\hat{\varphi}_{k/k} = \arctan\left(\frac{\hat{y}_{k/k} - \hat{y}_{k-1/k-1}}{\hat{x}_{k/k} - \hat{x}_{k-1/k-1}}\right)$$

where the parameter α of the smoothing filter has been selected experimentally. Sample results related to the reference paths shown in the Fig. 3 are presented in the Fig. 5. As we can see the controller stabilizes the robot on the desired trajectory. Results of experiments with other trajectories are similar to those in Fig. 5. We also observed that both controllers (7) and (8), under appropriate selection of their parameters, worked similarly.

The transputer system and our software prepared in occam enabled realization of the visual control with the sample rate 0.13 s.

The following software elements related to the image processing have been prepared:

- Gaussian filtering (GF)
- Thresholding (B)
- Morphological closing (MC)
- Image segmentation, robot recognition and position determination (SPD)

The table 1. gives typical execution times with 1 or 2 T9000 transputers for images 256x256 and 50x50. The execution times include two way communication overhead.

Table 1. Average processing time in [s] related to image 256x256 from Fig. 3. and to 50x50 ones covering the robot.

Task	256x256		50x50	
	T9	2xT9	T9	2xT9
GF	0.4639	0.2484	0.0173	0.0096
B	0.0462	0.0342	0.0019	0.0017
MC	0.2048	0.1185	0.0075	0.0048
SPD	0.5326	0.4100	0.0229	0.0180
GF+B+MC+SPD	1.2022	0.6343	0.0494	0.0293

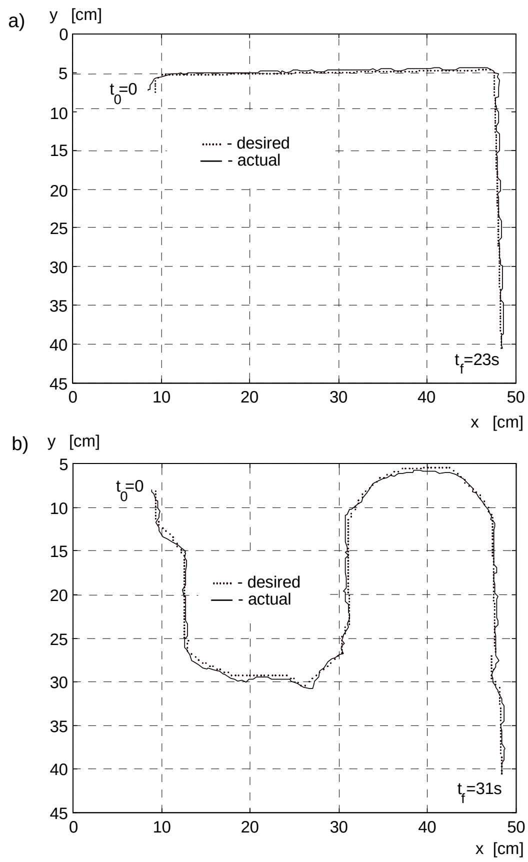

Fig. 5. Visual feedback trajectory stabilization for the robot Khepera on trajectory obtained by a) NF1, b) NF2.

7. CONCLUSIONS

The main problems considered in the paper are related to vision based path planning with artificial potentials methods and to trajectory stabilization using real time visual feedback.

To solve these computational expensive tasks a transputer network with T9000 processors has been used.

The results are promising. The warranted sampling period in the trajectory stabilizing problem is 0.13 s. Future work includes modification of the used methods and algorithms to accelerate the computations. We plan to integrate the on line trajectory planning (for dynamic environments) into the hierarchical control system and to apply the on board camera as additional visual input.

8. REFERENCES

1. Walsh G.: et al.: Stabilization of trajectories for systems with nonholonomic constraints, Proc. of the IEEE Conference on Robotics and Automation, 1992
2. Hashimoto K. (ed.): Visual servoing, World Scientific, London, 1993
3. Latombe J. C.: Robot motion planning, Kluwer Acad. Publ., London, 1991
4. Marshall A. D., Martin R. R.: Computer vision models and inspection. World Scientific, London, 1993
5. Sonka M., Hlavac V., Boyle R.: Image processing, analysis and machine vision, Chapman & Hall, London, 1994
6. Kwolek B., Wysocki M.: Vision feedback based multiprocessor control system, Proc. of the Fifth National Conference on Robotics, Wroc³aw, 1996, vol. 1, pp. 362-369 (in Polish)
7. Lenz R. K., Tsai R. Y.: Techniques for calibration of the scale factor and image center for high accuracy 3-D machine vision metrology, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 10, no. 5, 1988
8. INMOS, T9000 transputer hardware reference manual, 1993
9. Khepera, User manual, version 3.0, Laboratoire de microinformatique, Swiss Federal Institute of Technology, Lausanne, 1994
10. Grace A.: Optimization toolbox for use with MATLAB, The Math Works, Inc., 1994