Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Precise indoor localization of multiple mobile robots with adaptive sensor fusion using odometry and vision data

**Ganzorig Baatar, Mike Eichhorn,
Christoph Ament**

Institute for Automation and System Engineering
Technische Universität Ilmenau
98693 Ilmenau, Germany
e-mail: {ganzorig.baatar, mike.eichhorn, christoph.ament}@tu-ilmenau.de

Abstract: An accurate, reliable and a cost effective localization is the key feature of self-navigation of autonomous mobile robots. The position and orientation, together known as pose, of the mobile robot can be determined by using certain localization systems. In this work we use the mobile robot system Robotino – a practice-orientated educational and training system offered by Festo Didactic GmbH. For the Robotinos, position determination can be provided by laser scanner for Robotino or Robotino® Northstar System (Festo Didactic GmbH). These existing systems are quite expensive and localization accuracy in certain field dimensions is quite low. In this paper we provide a relatively inexpensive and a more accurate localization system, which combines the strengths of odometry and vision-based localization. The fusion of odometry and vision-based localization data is accomplished with the use of the Extended Kalman Filter (EKF). The resulting localization system delivers better accuracy and more frequent pose information of the mobile robots on the test field.

Keywords: Vision-based localization, Robotino, Mobile robots, (Adaptive) Sensor fusion, Extended Kalman Filter

1. INTRODUCTION

Autonomous navigation and interaction of mobile robots demand exact and reliable localization. For the absolute localization of outdoor autonomous mobile robots Global Positioning System (GPS) is used. Much research has been carried out due to the inaccuracy of GPS localization. Combining the odometry of the mobile robot, gyroscope and GPS signals guaranteed for example self-driving and interaction of multiple autonomous mobile robots [1]. For indoor usage, mobile robots may utilize landmarks by vision based self-localization, which allows the robots to maneuver when a GPS signal is unavailable [2]. The same approach can be applied by capturing a special marker system positioned on the floor, using an on-board camera [3].

Festo Didactic GmbH offers the Robotino® Northstar System for the global localization of mobile robots, especially Robotinos. The Robotino® Northstar System consists of an infrared projector and its corresponding receiver-sensor. The infrared projector projects two infrared light spots onto the ceiling, which means that the operating room has to have a flat ceiling. This system is able to detect and track mobile robots tagged with these receiver-sensors with an error of 50 mm within a 4 m×4 m operating area [4]. In our project we could not use this system because the error is too severe, e. g. for a docking maneuver, and a projection area was not available on the ceiling in our test hall.

For our research we used a camera to capture the test field in order to localize multiple mobile robots (vision-based localization). This allows the determination of the absolute positions and orientation of each robot relating to a predefined coordinate system (world frame).

Vision based localization in our system delivers new position and orientation information for every 80 ms with a positioning inaccuracy of a maximum of 12 mm. In other words, we can reach a tracking frequency of 12 Hz by using only the vision-based localization. In contrast to this relatively low tracking frequency vision-based localization provides very accurate position and orientation determination compared to the other localization systems. In order to achieve a higher tracking frequency at the same level of accuracy or better we use local localization techniques for each robot, e. g. odometry, to combine it with vision-based localization. Due to the nonlinearity of position estimation using wheel encoder signals we utilize the Extended Kalman Filter (EKF).

The remainder of this paper is structured as follows. In section 2 we provide the kinematics model of Robotino and the mathematical model of odometry calculation. Section 3 gives an overview about the vision-based localization technique and the special marker system. In section 4, we explain our approach for the sensor fusion method using EKF with adaptive correction factors for odometry. We evaluate our localization system in section 5, followed by a conclusion in the last section.

2. KINEMATICS MODEL OF ROBOTINO

In this work we use the three wheeled, omnidirectional mobile robot system Robotino [5], whose wheels are mounted symmetrically with 120 degrees from each other. Figure 1 shows a schematic diagram of Robotino. The kinematic equation of the robot is based on [6]:

$$\begin{pmatrix}\dot{x}\\ \dot{y}\\ \dot{\varphi}\end{pmatrix}=\begin{pmatrix}\frac{2}{3}\cos(\varphi+\delta) & -\frac{2}{3}\cos(\varphi-\delta) & \frac{2}{3}\sin(\varphi)\\ \frac{2}{3}\sin(\varphi+\delta) & -\frac{2}{3}\sin(\varphi-\delta) & -\frac{2}{3}\cos(\varphi)\\ \frac{1}{3R} & \frac{1}{3R} & \frac{1}{3R}\end{pmatrix}\begin{pmatrix}\dot{q}_1\\ \dot{q}_2\\ \dot{q}_3\end{pmatrix}, \quad (1)$$

where $(\dot{x},\dot{y})$ is the velocity relating to the world frame, $\dot{\varphi}$ is the heading rate and R is the distance between wheel and center of Robotino. The variables $\dot{q}_1$, $\dot{q}_2$ and $\dot{q}_3$ are the wheel velocities or rolling velocity expressed in body frame. The parameter δ is the wheel orientation with respect to the robot direction. In case of Robotino this parameter equates to 30°. The wheel encoders of Robotino deliver rotation speed of the motors (n_1, n_2, n_3) in **r**evolutions **p**er **m**inute (**rpm**). This rotation speed information will be converted into rolling speed $\dot{q}_1, \dot{q}_2$ and $\dot{q}_3$ using the conversion factor p, which allows to calculate the robot translation $(\dot{x},\dot{y})$ and the rotation velocity ($\dot{\varphi}$) using the Equation (1) directly in m/s (meters per second) and rad/s (radians per second):

$$\begin{pmatrix}\dot{q}_1 & \dot{q}_2 & \dot{q}_3\end{pmatrix}^T=\begin{pmatrix}n_1 & n_2 & n_3\end{pmatrix}^T p, \quad (2)$$

$$p=\frac{2\pi r_w}{g\cdot 60}, \quad (3)$$

where the parameter r_w is the wheel radius and g is the gear ratio of Robotino [5].

It should be remarked that the input order of the wheel encoder signal has to have the same order as shown in Figure 1 (wheel encoder orders differ in the actual version 2.1.1 of MATLAB/Simulink library for Robotino [7]).

3. VISION-BASED LOCALIZATION

Localization using camera or vision systems allows very accurate and reliable results. In this work we use an industrial camera (IDS-Imaging UI-6250SE-C 1/1.8“ CCD, 2 MP, 20 fps, GigE Camera [8]) to capture the operational field. The operational field in our case is a 4.8 m×3.5 m free space in an experimental hall and the camera is placed on the ceiling of the hall. Using the images captured by the camera we realize the global localization of all mobile robots on the field. To differentiate the robots, we identify each mobile robot with a special marker. As mentioned before, the vision-based localization allows only low frequency tracking. Another drawback of this method is the localization failure in certain situations, for instance bad illumination, markers on the robots cannot be detected and localized properly.

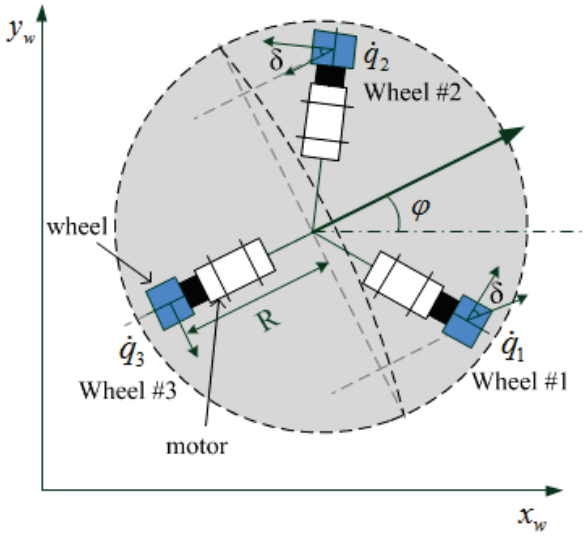


Fig. 1. Kinematics model of Robotino in a world coordinate system

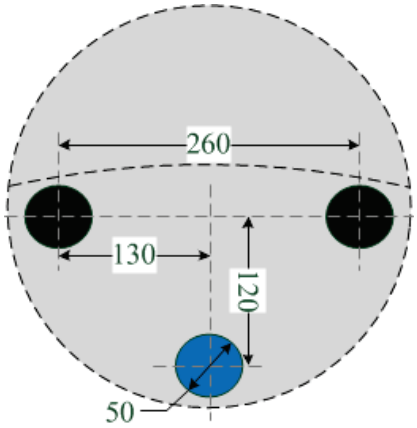


Fig.2. A marker system for Robotino

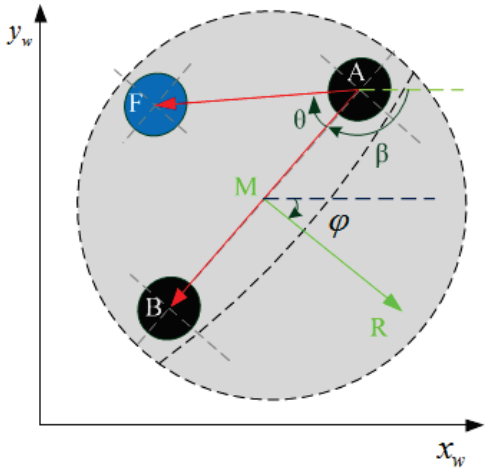
3.1 Marker system for mobile robots

For proper localization and differentiation of mobile robots we use the so called 3-point-Marker. This marker employs two black circles and one colored circle with a certain radius (see Figure 2). The colored circle is used to determine the orientation of robots and to differentiate them. Using a monochrome camera, differentiation can be carried out using certain geometrical shapes, i. e. a triangle, or any other shape instead of a colored circle. The distances between the circles are affected by the geometry of the Robotino.

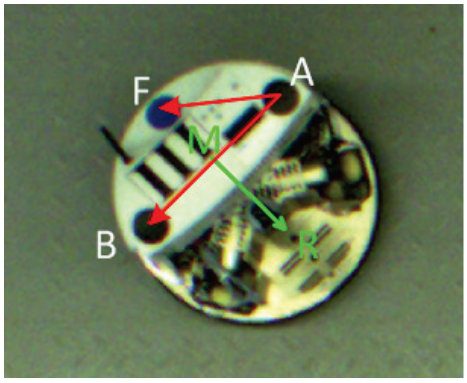
3.2 Localization

We developed a localization algorithm for each robot marked with the marker system. As a result, all three circles of the marker are localized and clustered together.

Using the position information of these circles we can use our image processing algorithm for orientation and position determination for the Robotino.



(a)



(b)

Fig. 3. Determination of orientation using circle positions; (a) Illustration of angles; (b) Illustration of the vectors on real Robotino image

For the determination of the orientation (see Figure 3) first we measure the angles β (angle between the x-axis of world frame and the vector AB in Figure 3a) and θ (angle between the vectors AB and AF in Figure 3a). Using these angles we define the current orientation φ:

$$\varphi = \begin{cases} \beta - 90^{\circ}, \theta > 0 \\ \beta + 90^{\circ}, \theta < 0. \end{cases} \quad (4)$$

Since the positions of the black circles of the marker on the robot and the robot orientation are known, the center position of the Robotino can be determined. Due to the geometric form of Robotino, the black circles may not be in the middle axis of the robot. The midpoint between the black circles (M) is 20 mm behind the robots central position. Hence, after the defining position of M, this must be moved forward by 20 mm using orientation information to get the absolute center point of Robotino.

In order to realize the vision-based localization some preparations of the image have to be applied in a short time period. As can be seen in Figure 4 the captured image contains non-interesting regions for the localization of the robots. Therefore it is necessary to eliminate the non-interesting regions. For this we used a "grid-search-algorithm" so that we get only regions the robots are located at (see Figure 5). The search of the approximate position of the robots is not necessarily precise, so that somewhat non-interesting regions are still seen on the reduced image.

The image processing program for the vision-based localization was realized using MVTec's machine vision software HALCON version 11.0.2 [9]. In our case the localization of three robots in the test field takes about 80 ms, which is not fast enough to guarantee proper control of three Robotinos. Image distortion caused by the short focal length of the lens (focal length = 5 mm) and imaging noise can increase the inaccuracy of the localization algorithm up to 12 mm. In addition, if the image processing algorithm fails to detect the corresponding marker circles, the pose determination of the concerning Robotino will not occur. It makes the single use of vision-based localization unreliable. To ensure better reliability and accuracy, we use EKF for sensor fusion, which is described in the next section.

HALCON allows to export the image processing program code as C++ code in order to establish a C++ application. In our case we realized an image processing application for the vision-based localization with an inter-process TCP/IP Socket, which allowed sending the determined pose information to MATLAB/Simulink, where the sensor fusion was realized.

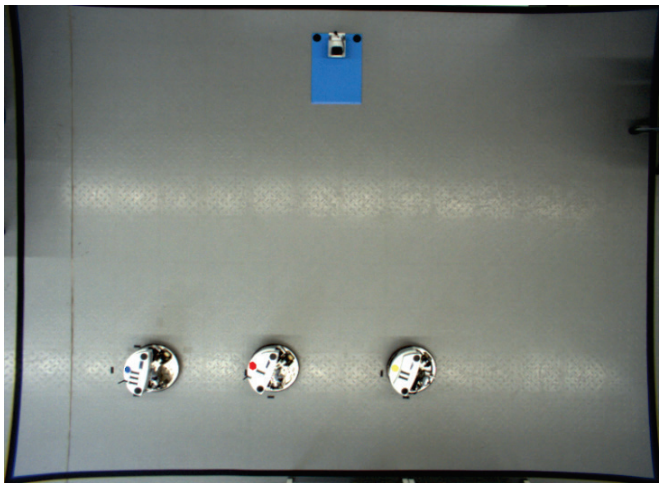


Fig. 4. Raw image from camera: Robotinos; Docking-Station

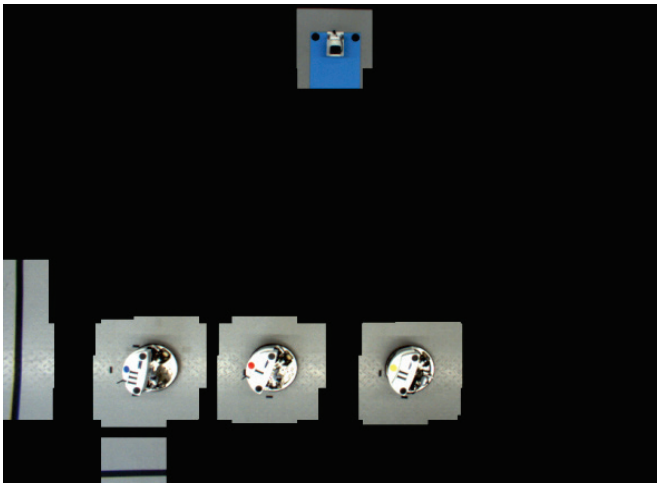


Fig. 5. Reduced image used for precise localization

4. SENSOR FUSION USING EXTENDED KALMAN FILTER

Kalman filter is one of the most widely used algorithms for sensor fusion. In this paper, we employ the well-known Extended-Kalman-Filter (EKF) [10] for the state estimation of the nonlinear model (1) for odometry. Pose changes of the mobile robot can be determined using wheel encoder signals at a very high frequency. However, this allows only relative localization and delivers very inaccurate pose information during a long time maneuver. Using odometry in short time intervals though, it provides very reasonable pose information. In contrast, vision-based localization delivers much more accurate pose information at a low frequency.

We have implemented an EKF, which utilizes the odometry localization as its state prediction method. It is reasonable to use the pose information from the vision-based localization for the measurement update (a posteriori update) of EKF. The measurement update is only executed, when new position and orientation information is provided from the vision-based localization.

In order to use the odometry in the prediction phase of the EKF, we utilize the Equation (1) as state update calculation and set the wheel encoder signal as part of the input of the EKF.

Extended Kalman Filter has the following system and measurement equations [10]:

$$x_k = f\left(x_{k-1}, u_k, w_k\right), \tag{5}$$

$$y_k = g(x_k, v_k), \tag{6}$$

where x_k is the state vector of the system and u_k is the input vector of the filter, w_k and v_k are Gaussian noises, $f\left(x_{k-1}, u_k, w_k\right)$ is the nonlinear function representing the model of the system and $g(x_k, v_k)$ is the output function. The linearization of the nonlinear state and measurement equations is made by using [11]:

$$x_k = f\left(\hat{x}_{k-1}^{+}, u_{k-1}, w_{k-1}\right) + A_{k-1}\left(x_{k-1} - \hat{x}_{k-1}^{+}\right) + F_{k-1}(w_k - w_{k-1}), \tag{7}$$

$$y_k = g(\hat{x}_k^{-}, v_k) + C_k\left(x_k - \hat{x}_k^{-}\right) + G_k v_k, \tag{8}$$

where $\hat{x}_{k-1}^{+}$ is the previous a priori state, $\hat{x}_k^{-}$ is the current a posteriori state, A, F, C and G are the linearized system-matrices or Jacobi matrices (see Appendix A). These matrices are defined by:

$$A_{k-1} = \left.\frac{\partial f\left(x_{k-1}, u_{k-1}, w_{k-1}\right)}{\partial x}\right|_{x=\hat{x}_{k-1}^{+}}; F_{k-1} = \left.\frac{\partial f\left(\hat{x}_{k+1}^{+}, u_{k-1}, w\right)}{\partial w}\right|_{w=w_k};$$
$$C_k = \left.\frac{\partial g\left(x_k, v_k\right)}{\partial x}\right|_{x=\hat{x}_k^{-}}; \quad G_k = \left.\frac{\partial g\left(\hat{x}_k^{-}, v\right)}{\partial v}\right|_{v=v_k}. \tag{9}$$

As an input of EKF u_k we use three wheel encoder signals and pose information from vision-based localization. For further information about the EKF we recommend corresponding literature and [10].

At first, we have implemented a 6-state EKF with the following states

$$x_k = \left(dx, dy, d\varphi, x, y, \varphi\right) \in \mathbb{R}^6, \tag{10}$$

where $dx, dy, d\varphi$ are the predicted speed of Robotino and x, y, φ are the predicted pose information. In this case we calculate the Jacobi matrix A as follows:

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & A_{1,6} \\ 0 & 0 & 0 & 0 & 0 & A_{2,6} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \Delta t & 0 & 0 & 0 & 0 & 0 \\ 0 & \Delta t & 0 & 0 & 0 & 0 \\ 0 & 0 & \Delta t & 0 & 0 & 0 \end{pmatrix}, \tag{11}$$

with the sampling time Δt of EKF,

$$A_{1,6} = -\frac{2}{3}\cdot\sin\left(x_6+\delta\right)\cdot\dot{q}_1 + \frac{2}{3}\cdot\sin\left(x_6-\delta\right)\cdot\dot{q}_2 + \frac{2}{3}\cdot\cos\left(x_6\right)\cdot\dot{q}_3$$

and

$$A_{2,6} = \frac{2}{3}\cdot\cos\left(x_6+\delta\right)\cdot\dot{q}_1 - \frac{2}{3}\cdot\cos\left(x_6-\delta\right)\cdot\dot{q}_2 + \frac{2}{3}\cdot\sin\left(x_6\right)\cdot\dot{q}_3.$$

The output matrix C selects the pose information from the states of EKF and has a dimension of $\mathbb{R}^{3\times 6}$:

$$C = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{12}$$

During the evaluation of this implemented 6-state EKF, we have recognized a drawback in position accuracy, which is described in the evaluation section later in this paper. In order to compensate the drawback of 6-state EKF, we implemented 9-state EKF for the identification of three additional correction factors k_{rw1}, k_{rw2} and k_{rw3}. The state vector of this 9-state EKF is given by:

$$x_k = \left(dx, dy, d\varphi, x, y, \varphi, k_{rw1}, k_{rw2}, k_{rw3}\right) \in \mathbb{R}^9. \tag{13}$$

The correction factors are deployed to compensate faulty estimated wheel rolling speeds caused by uneven floor and the mounting position of the front wheels in a body frame. This allows better speed prediction of the Robotino between the update periods. Hence, the odometry equation (1) is manipulated with the correction factors and was set in EKF as follows:

Table 1. Pseudo code for EKF [11]

EKF Algorithm

{Initialization}

$x_0 = \{0\}$

$P_0^+ = \{0\}$

$Q = \{\}, R\{\}$

loop

{prediction update - a priori update}

linearization(A,F)

$\hat{x}_k^- = f(x_{k-1}^+, u_k, w_k)$

$P_k^- = A_{k-1} P_{k-1}^+ A_{k-1}^T + F_{k-1} Q_{k-1} F_{k-1}^T$

{measurement update – a posteriori update}

if *new pose information from vision-based localization*

linearization(C, G)

$K = P_k^- C_k^T / (C_k P_k^- C_k^T + G_k R G_k^T)$

$\hat{x}_k^+ = \hat{x}_k^- + K(y_k - g(\hat{x}_k^-, 0))$

$P_k^+ = (I - KC) P_k^-$

else – *no measurement (a posteriori= a priori)*

$P_k^+ = P_k^-$

$x_k^+ = x_k^-$

end if

{calculate the output}

$y_k = C_k x_k + v_k$

end loop

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{pmatrix}_{EKF} = \begin{pmatrix} \frac{2}{3}\cos(\varphi+\delta) & -\frac{2}{3}\cos(\varphi-\delta) & \frac{2}{3}\sin(\varphi) \\ \frac{2}{3}\sin(\varphi+\delta) & -\frac{2}{3}\sin(\varphi-\delta) & -\frac{2}{3}\cos(\varphi) \\ \frac{1}{3R} & \frac{1}{3R} & \frac{1}{3R} \end{pmatrix} \cdot \begin{pmatrix} \dot{q}_1 \cdot k_{rw1} \\ \dot{q}_2 \cdot k_{rw2} \\ \dot{q}_3 \cdot k_{rw3} \end{pmatrix}. \quad (14)$$

The Equation (14) provides in 9-state EKF the state update calculation and the correction factors, which compensate the faulty rolling speed estimation by acting for example as a variable wheel radius (see Equation (2) and (3)) caused by an uneven floor. The correction factors are adaptive and have the same effect as the Kalman gain and it is to be expected that these factors correlate to a certain value.

A faulty estimation of the rolling speed of the wheels can be caused by the slip, nubby floor in the experimental hall and special form of omni-wheels [12] and its mounting positions on the Robotino body frame [5]. The error propagation of odometry and longtime gap between update phases caused the increasing inaccuracy of pose estimation in the 6-state EKF. The correction factors in 9-state EKF however allowed compensating the faulty pose estimation of odometry. Consequently we are able to make long period update cycles in order to get more reliable pose information from vision-based localization. Due to possible extremely faulty pose information (very big value difference to last position) from vision-based localization, we limit the value range of the correction factors from 0.7 to 1.3. Otherwise, the extreme value changes may lead to uncertain Kalman gain and correction factors $(k_{rw1}, k_{rw2}, k_{rw3})$, which make the EKF prediction false.

The system noise covariance matrix Q is determined empirically, and during experimental tests we have defined the noise of wheel encoders with 10 rpm:

$$Q = 10^2 \cdot I_3 . \quad (15)$$

Measurement noise covariance matrix R is described by the inaccuracy of vision-based localization. Inaccuracy of position determination, in our case, equates to 12 mm and orientation inaccuracy to 0.029 rad or 1.7°:

$$R = \begin{pmatrix} 0.012^2 & 0 & 0 \\ 0 & 0.012^2 & 0 \\ 0 & 0 & 0.029^2 \end{pmatrix}. \quad (16)$$

The sampling time of EKF defines the speed of the localization system. In our case we chose the sampling time with 40 ms. This sampling time equals to the sampling time of the control system for mobile robots in MATLAB/Simulink. Otherwise, faster sampling causes a lack in wheel encoder signal information, which makes the calculations of odometry incorrect and the use of EKF ineffective. The implemented EKF is shown in Table 1.

5. EVALUATION

For the evaluation of tracking we compare different trajectories of the robot(s) by driving them using pure odometry, vision and two different EKF pose information.

As we know, localization by using odometry cannot deliver very good results. Therefore, we do not evaluate the odometry localization in this case. For the evaluation of a vision-based localization method, we utilized simple measurements on the ground and compared it with the pose information from the vision. As a result we define the standard deviation of position determination using vision-based localization in x and y with 12 mm and the standard deviation of orientation determination is set to 1.7°.The reason of these inaccuracies are to be explained by the image distortion caused by short focal length of the used camera system. This inaccuracy information, as mentioned in the last section, is used to describe the noise covariance matrix. In certain situations, vision-based localization fails for the fast moving robots because of blurred marker points. After setting and using the EKF, localization failures at visual level can be compensated for, making the localization system more reliable.

We have implemented two different demonstrations using Robotinos. The first demonstration (Figure 6) is a docking maneuver of a single Robotino into its docking-station. Robotino is driven by using pose information from:

- Odometry localization,
- Pure vision-based localization,
- 6-state EKF localization and
- 9-state EKF localization.

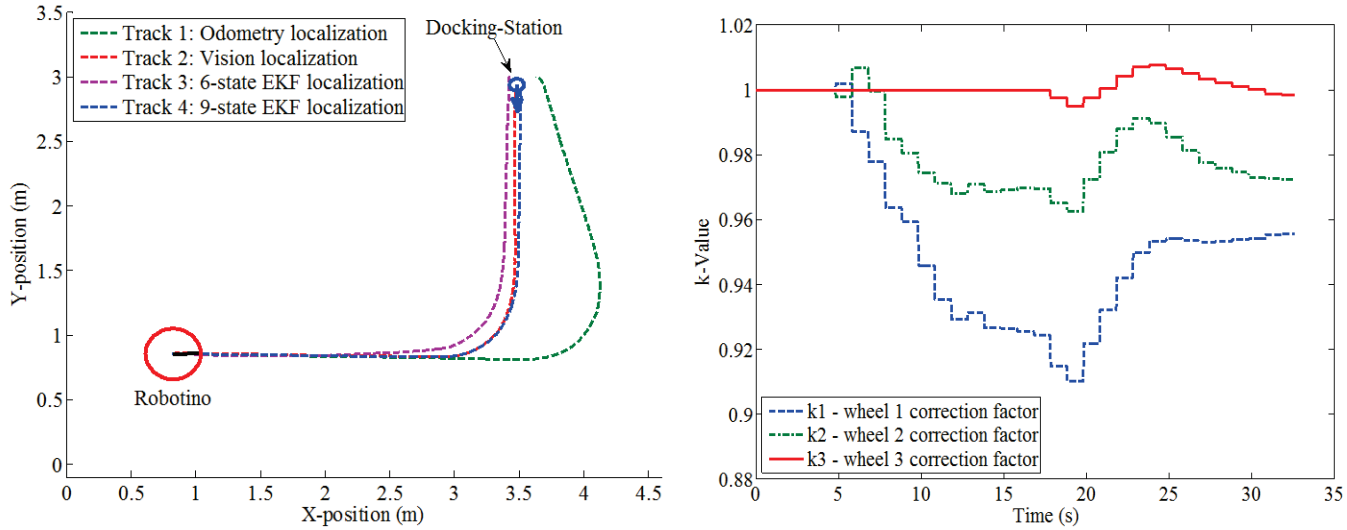


Fig. 6. Docking drive of Robotino; (a) Different driving trajectories caused by different localization methods; (b) Signal pattern of correction factors of 9-state EKF

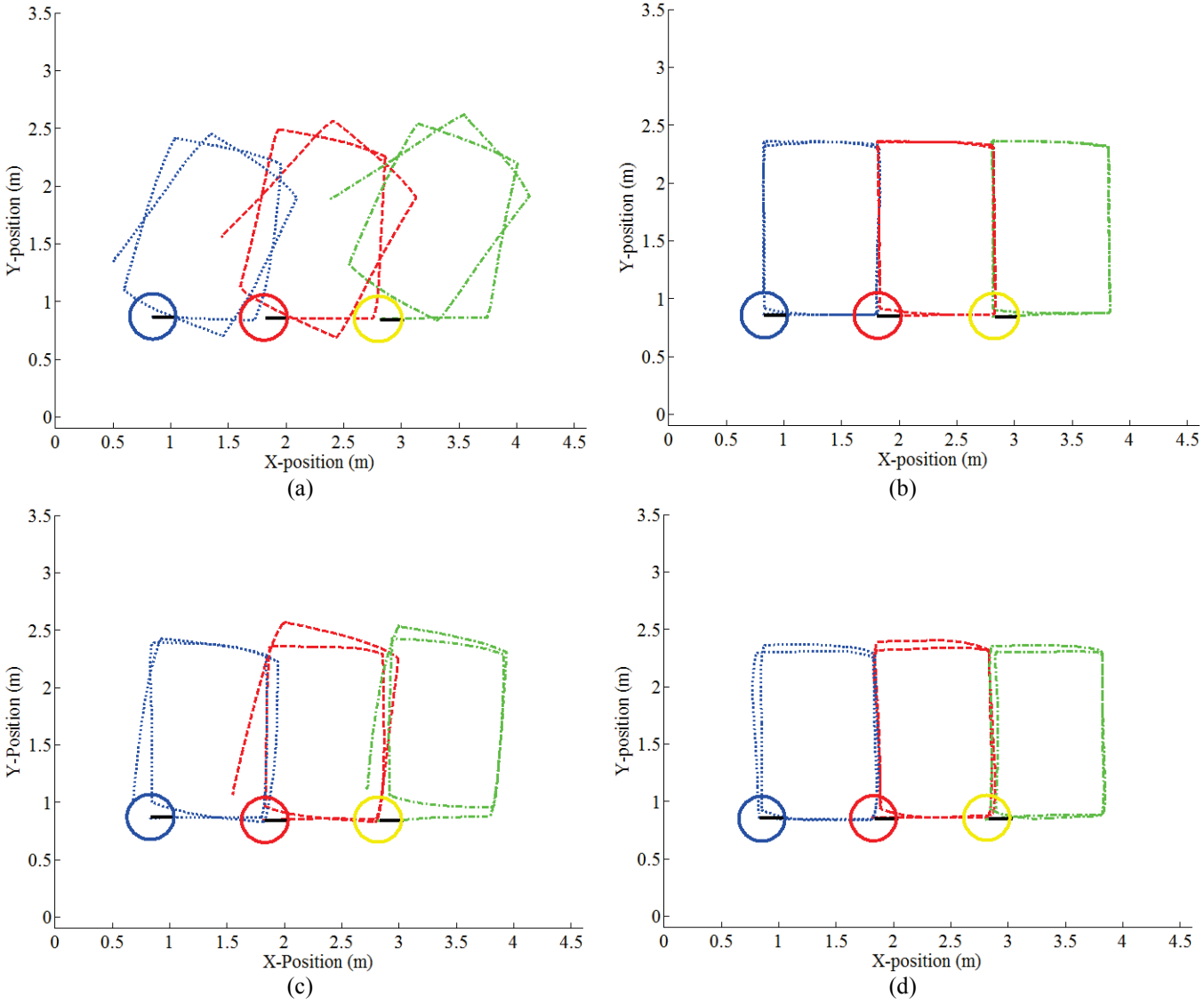


Fig. 7. Synchronous drive of the Robotinos; Robotino trajectories driven by: (a) odometry localization; (b) vision localization; (c) 6-state EKF; (d) 9-state EKF localization

As seen in Figure 6a Robotino driven by odometry pose information (green line) fails to hit its mark with a 200 mm deviation. Using pure vision-based localization (red line), we got very reasonable results and it was possible to dock the Robotino into the docking station. However, we have noticed bucking movements of Robotino due to the low tracking frequency of vision-based localization. Trajectories caused by using two different EKF pose information are also visualized in Figure 6a.

Robotino driven by 6-state EKF pose information (magenta line) missed its mark by 20 – 30 mm while using 9-state EKF pose information allowed perfect docking of Robotino. These results show the advantages of correction factors in 9-state EKF (blue line). Figure 6b shows the signal pattern of the correction factors in 9-state EKF. During the extensive tests we have seen that the rolling speed correction factors of the two front wheels (wheel #1 and #2) correlated into 0.95 and the correction factor of the back wheel (wheel #3) remains at the value of 1. These values result from the special form of the omni-wheel and their orientation in Robotino frame. In order to achieve good results using 9-state EKF, it is reasonable to define the correlating values of each correction factor detected during test drives and set these as the initial state values in 9-state EKF.

In a second demonstration, three Robotinos drive a given rectangle in formation (synchronously). Figure 7a shows the trajectory of the Robotinos driven by using odometry localization. As can be seen, error propagation is the main problem when using odometry for localization. Controlling the Robotinos using the vision-based localization method (Figure 7b.) allows for very good compliance of formation, which results in the Robotinos driving a nearly perfect rectangle track. However, the bucking movements of Robotinos were still observed. The same accuracy results could not be repeated by using 6-state EKF (Figure 7c). In this case Robotinos cannot follow the expected trajectories and have the same behavior as the driving by odometry localization. Because of the missing correction factors, error propagation from the odometry could not be corrected properly in 6-state EKF. Robotino trajectories driven by 9-State EKF are shown in Figure 7d. The first round had some deviations from the expected rectangle track. This can be explained with the learning phase of the EKF or the lack of compensation for correction factors of 9-state EKF. Additional rounds allowed Robotinos to follow the predetermined tracks with more accuracy. Upon further evaluations, we considered implementing more complex maneuvers for Robotinos. For measuring the ground truth, we need to use an independent localization system with corresponding accuracy to draw some comparable conclusions.

6. CONCLUSIONS

In this paper we presented a reliable and accurate localization method for the indoor mobile robots. We have shown that a well-combined use of odometry and vision-based localization methods provide very reasonable results. For this combination (sensor fusion) we employed the Extended Kalman Filter, which allows direct use of the nonlinear translation equations of odometry. Vision-based localization using a single camera system provided global pose information of all marked mobile robots, in this case Robotinos.

The localization approach presented here delivers position and orientation information of each robot at a 25 Hz tracking frequency with a maximum inaccuracy of 12 mm (position) and 1.7° (orientation) in a 4.8 m × 3.5 m field dimension. Using additional sensors, for example a gyroscope, the localization accuracy can be improved. In future work, we will also investigate possibilities to employ onboard cameras of Robotinos for so called “vision odometry”, for example to properly maneuver the Robotino into a docking station beyond a capture field of the global vision-based localization. We will also investigate the possibilities to use our localization system for a precision positioning of objects or mobile robots to certain points.

REFERENCES

[1] P. Goel, S. I. Roumeliotis und G. S. Sukhatme, *Robust Localization Using Relative and Absolute Position Estimates* in *IEEE/RSJ international Conerence on Intelligent Robots and Systems*, Los Angeles, CA, USA, 1999,

[2] P. F. Alcantrilla, S. M. Oh, G. L. Mariottini, L. M. Bergasa, F. Dellaert, *Learning Visibility of Landmarks for Vision Based Localization* in *IEEE International Conference on Robotics and Automation*, 2010,

[3] B. Bischoff, D. Nguyen-Tuong, F. Streichert, M. Ewert, A. Knoll, *Fusing Vision and Odometry for Accurate Indoor Robot Localization* in *12th International Conference on Control, Automation, Robotics & Vision*, Guangzhou, China, 2012, pp. 347-352.

[4] Festo Didactic GmbH, “Robotino Northstar Handbook“.

[5] Festo Didactic GmbH, “Robotino Handbook”. http://www.robotino.de

[6] X. Li, M. Wang, A. Zell, *Dribbling Control of Omnidirectional Soccer Robot*“ in *IEEE International Conference on Robotics and Automation*, Roma, Italy, 2007, pp. 2623-2628

[7] Openrobotino - Robotino Wiki, *Robotino Matlab programming*, http://wiki.openrobotino.org/index.php?title=Matlab,

[8] IDS-Imaging GmbH, *UI-6250SE-C Camera*, http://de.ids-imaging.com/store/ui-6250se.html,

[9] MVTec Software GmbH, “*HALCON 11*”, http://www.mvtec.com/halcon

[10] Dan Simon, *Optimal State Estimation,* Hobokoen: Wiley Interscience, 2006,

[11] M. Norgaard, *State Estimation for Nonlinear Systems – KALMTOOL,*Technical University of Danemark, Kongens, Lyngby, Denmark, 2002

[12] RoboterNETZ, *Omni-Wheels*, http://www.rn-wissen.de/index.php/OmniWheels

Appendix A. Linearized System matrices of 9-State EKF

$$A=\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & A_{1,6} & A_{1,7} & A_{1,8} & A_{1,9} \\ 0 & 0 & 0 & 0 & 0 & A_{2,6} & A_{2,7} & A_{2,8} & A_{2,9} \\ 0 & 0 & 0 & 0 & 0 & 0 & A_{3,7} & A_{3,8} & A_{3,9} \\ A_{4,1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & A_{5,2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & A_{6,3} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$A_{1,6}=-\frac{2}{3}\cdot\sin\left(x_6+\delta\right)\cdot\dot{q}_1\cdot x_7+\frac{2}{3}\cdot\sin\left(x_6-\delta\right)\cdot\dot{q}_2\cdot x_8+\frac{2}{3}\cdot\cos\left(x_6\right)\cdot\dot{q}_3\cdot x_9$$

$$A_{1,7}=\frac{2}{3}\cdot\cos\left(x_6+\delta\right)\cdot\dot{q}_1;\ A_{1,8}=-\frac{2}{3}\cdot\cos\left(x_6-\delta\right)\cdot\dot{q}_2;\ A_{1,9}=\frac{2}{3}\cdot\sin\left(x_6\right)\cdot\dot{q}_3;$$

$$A_{2,6}=\frac{2}{3}\cdot\cos\left(x_6+\delta\right)\cdot\dot{q}_1\cdot x_7-\frac{2}{3}\cdot\cos\left(x_6-\delta\right)\cdot\dot{q}_2\cdot x_8+\frac{2}{3}\cdot\sin\left(x_6\right)\cdot\dot{q}_3\cdot x_9;$$

$$A_{2,7}=\frac{2}{3}\cdot\sin\left(x_6+\delta\right)\cdot\dot{q}_1;\ A_{2,8}=-\frac{2}{3}\cdot\sin\left(x_6-\delta\right)\cdot\dot{q}_2;\ A_{2,9}=-\frac{2}{3}\cdot\cos\left(x_6\right)\cdot\dot{q}_3;$$

$$A_{3,7}=\frac{1}{3\cdot R}\cdot\dot{q}_1;\ A_{3,8}=\frac{1}{3\cdot R}\cdot\dot{q}_2;\ A_{3,9}=\frac{1}{3\cdot R}\cdot\dot{q}_3;\ A_{4,1}=\Delta t;\ A_{5,2}=\Delta t;\ A_{6,3}=\Delta t\ ;$$

$$F=\begin{pmatrix} F_{1,1} & F_{1,2} & F_{1,3} \\ F_{2,1} & F_{2,2} & F_{2,3} \\ F_{3,1} & F_{3,2} & F_{3,3} \end{pmatrix}$$

$$F_{1,1}=\frac{2}{3}\cdot\cos\left(x_6+\delta\right)\cdot p\cdot x_7;\ F_{1,2}=-\frac{2}{3}\cdot\cos\left(x_6-\delta\right)\cdot p\cdot x_8;\qquad F_{1,3}=\frac{2}{3}\cdot\sin\left(x_6\right)\cdot p\cdot x_9;$$

$$F_{2,1}=\frac{2}{3}\cdot\sin\left(x_6+\delta\right)\cdot p\cdot x_7;\ F_{2,2}=-\frac{2}{3}\cdot\sin\left(x_6-\delta\right)\cdot p\cdot x_8;\qquad F_{3,3}=-\frac{2}{3}\cdot\cos\left(x_6\right)\cdot p\cdot x_9;$$

$$F_{3,1}=\frac{1}{3\cdot R}\cdot p\cdot x_7;\ F_{3,2}=\frac{1}{3\cdot R}\cdot p\cdot x_8;\ F_{3,3}=\frac{1}{3\cdot R}\cdot p\cdot x_9;$$

$$C=\begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix};\qquad G=0$$