# Test-Environment based on a Team of Miniature Walking Robots for Evaluation of Collaborative Control Methods

Florian Weissel, Marco F. Huber, and Uwe D. Hanebeck

***Abstract*— For the collaborative control of a team of robots, a set of well-suited high-level control algorithms, especially for path planning and measurement scheduling, is essential. The quality of these control algorithms can be significantly increased by considering uncertainties that arise, e.g. from noisy measurements or system model abstraction, by incorporating stochastic filters into the control. To develop these kinds of algorithms and to prove their effectiveness, obviously real-world experiments with real world uncertainties are mandatory. Therefore, a test-environment for evaluating algorithms for collaborative control of a team of robots is presented. This test-environment is founded on miniature walking robots with six degrees of freedom. Their novel locomotion concept not only allows them to move in a wide variety of different motion patterns far beyond the possibilities of traditionally employed wheel-based robots, but also to handle real-world conditions like uneven ground or small obstacles. These robots are embedded in a modular test-environment, comprising infrastructure and simulation modules as well as a high-level control module with submodules for pose estimation, path planning, and measurement scheduling. The interaction of the individual modules of the introduced test-environment is illustrated by an experiment from the field of cooperative localization with focus on measurement scheduling, where the robots that perform distance measurements are selected based on a novel criterion, the normalized mutual Mahalanobis distance.**

## I. Introduction

In recent years, the interest in algorithms for collaborative robots has increased steadily [1]. Here, cooperative path planning and control [2], cooperative sensor scheduling and management [3], as well as cooperative localization and mapping [4] are of special importance. Especially for cooperative localization, the explicit consideration of uncertainties, which arise from different sources, e.g. imperfect sensor measurements or system models, leads to significantly improved pose estimation. This also leads to an improved performance of any of the other components of the cooperative robot control, as they highly depend on this information. To obtain an efficient cooperative localization mechanism for autonomous robots, it is typically not sufficient to just acquire data in a brute-force manner, but it is necessary to employ elaborate measurement scheduling techniques to increase the information gain of every single measurement and thus decrease the resource, i.e., energy, consumption.

Any of the necessary control components for cooperative robot control and especially their interaction can only be developed and validated on the basis of real-world experiments.

Florian Weissel, Marco F. Huber, and Uwe D. Hanebeck are with the Intelligent Sensor-Actuator-Systems Laboratory, Institute of Computer Science and Engineering, Universität Karlsruhe (TH), Germany. {weissel|marco.huber|uwe.hanebeck}@ieee.org

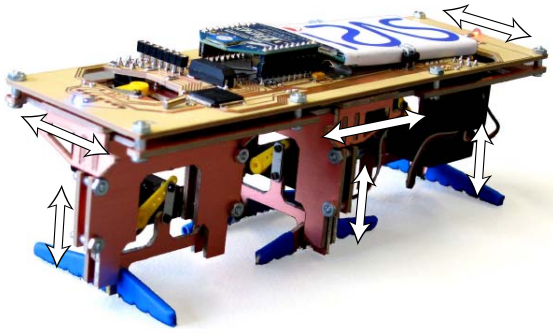

Fig. 1. One of the employed robots. The 6 linear degrees of freedom are indicated by arrows.

Consequently, for investigating the properties of these control algorithms, a variety of test-environments employing several robots of different sizes and types has been developed, where most of the employed robots are wheel based [5], [6], [7] or use differential-drive tracks [8]. These technologies, which are fairly easy to build and operate, have the disadvantage that the robots are typically not able to handle difficult real-world surface conditions and limit the experiments significantly due to the special kinematics of the different robots.

In this paper, we present a test-environment that is based on miniature walking robots with six independent linear degrees of freedom (DOF) as depicted in Fig. 1. This robot design leads to a wide variety of different possible motion patterns like going forwards, going sidewards as well as rotating. Additionally, it allows the robots to cope with real world situations like small obstacles and uneven ground.

The robots are embedded in a control-environment comprising modular components that can be easily altered or exchanged. User interaction, i.e., control of the experiments as well as feedback of the robots' status is provided by a graphical user interface that allows supervising the main module of the test-environment software, the high-level control, which comprises submodules for the robots' pose estimation, a path planning submodule as well as a submodule for measurement scheduling. The high-level control acts either on the real radio-controlled robots or it acts on a simulation engine that mimics the real robots. The robots' true poses are determined with an overhead camera system, whose data is also used for generating simulated sensor readings on which the high level control is based.

The remainder of this paper is structured as follows: In the next section, the 6-DOF miniature walking robots that lay the foundation of the presented test-environment are introduced. The test-environment and its modules are

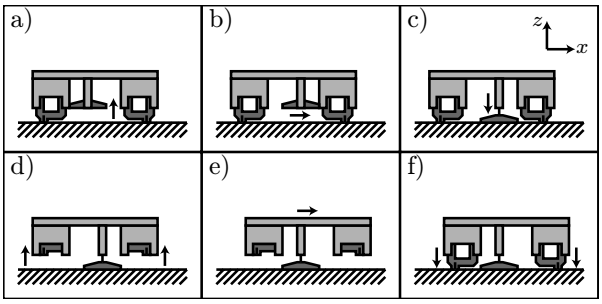


Fig. 2. Forward step of the robot (side view).

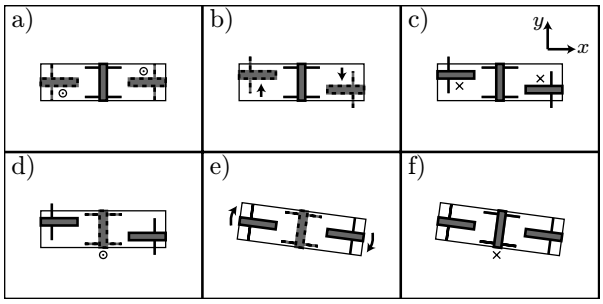


Fig. 3. Rotation of the robot (top view with transparent top segment, raised feet are indicated by dashed lines).

described in Section III. In Section IV, the high-level robot control module and its submodules for pose estimation, path planning, and measurement scheduling are highlighted, where a novel normalized mutual Mahalanobis distance for measurement scheduling is introduced. In Section V, experimental results from the field of cooperative localization with emphasis on the measurement scheduling are presented that illustrate the practical usefulness of the test-environment and the advantages that can be gained by its employment. The paper closes with conclusions and perspectives on future work.

## II. Miniature Walking Robots

The presented test-environment is based on a group of miniature walking robots. These robots are inspired by the robots presented in [9]. Even if the basic concept is similar to some degree, the overall design has significantly changed. The robots comprise four main elements, three legs and a top segment with a length of 17 cm and a width of 5 cm that forms the body (Fig. 1). Each leg has two independent linear degrees of freedom (DOF), which leads to a total of six. Each linear DOF has a range of 2 cm. The first DOF of each leg allows it to be lifted individually and the second one permits the front and back leg to move sidewards and the middle leg to move forwards and backwards. The overall height of the robot lies between 5 and 7 cm, depending on the pose of the legs. Due to the symmetric design of the mechanical structure of the robot, there is no dedicated front or back, thus if the robot changes direction, the tail becomes the head and vice versa.

### A. Motion Patterns

The design of the robots allows a variety of different basic motion patterns that all may be superimposed due to the independence of all six joints as shown in the attached video.

One important motion pattern is the forward motion as depicted in Fig. 2. Here, first the middle foot is lifted, moved forward, and then lowered again (Fig. 2 a) - c)). Then, the front and back feet are lifted simultaneously (Fig. 2 d)) and the base is moved forward relative to the middle foot on which the robots stands (Fig. 2 e)). Finally, the front and back feet are lowered again. The robot is in its initial configuration and the next step can follow directly.

Another important motion pattern is the rotation of the robot. In Fig. 3 the necessary steps for a rotation are illustrated from a top view, where the top segment is shown transparently. First, the front and back feet are lifted (Fig. 3 a)). Then, for a clockwise rotation, the front foot is moved to the right and the back foot is moved to the left (Fig. 3 b)). After that, both the front and back feet are lowered and the middle foot is raised (Fig. 3 c) - d)). For executing the actual rotation, the front and back feet are reset to their initial position, which leads to a rotation movement of the top segment, where the axis of rotation is in the middle of the robot (Fig. 3 e)). Finally, the middle foot is lowered and the robot is in its initial configuration.

If the rotation motion pattern is slightly modified, the robot moves in a sidewards motion, i.e., a crabwalk. Here, the second step of the motion, as illustrated for the rotation in Fig. 3 b), is modified such that both the front and the back feet move not in the opposite but in the same direction, which eventually leads, if the fifth step is also altered accordingly, not to a rotation but a sidewards motion.

Besides these basic motion patterns, a wide variety of other different motion patterns is possible. For example, if the rotation as described above is just executed with only the front foot, this leads to a rotation of the robot with the axis of rotation not in the middle of the robot but at its back foot. Obviously, all these motion patterns can easily be superimposed as all six DOFs of the robot are totally independent, which permits omnidirectional motion with different steps sizes including high precision small steps ($< 1$ mm). Due to the range of the linear joints, the step height can be adjusted between a few millimeters for a fast step sequence and up to 2 cm to cope with small obstacles and uneven ground. Next to these advantages compared to wheel-based systems, the robot's kinematic model is implicitly (time) discretized due to the robot's steps (Section II-B) and the simple but robust design allows for low cost in-house manufacturing (Section II-C).

### B. Kinematic Model

High-quality collaborative localization requires an adequate model of the walking robot's kinematic for employing stochastic filters. If the forward motion, the sidewards motion, and the rotation around the middle axis of the robot are superimposed, the omnidirectional motion of a single robot

on a 2D surface can be described by means of the discrete-step nonlinear kinematic model

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \Phi_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ \Phi_k \end{bmatrix} + \begin{bmatrix} \cos(\Phi_k) & -\sin(\Phi_k) & 0 \\ \sin(\Phi_k) & \cos(\Phi_k) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{s}_k^F \\ \hat{s}_k^S \\ \hat{\omega}_k \end{bmatrix} + \begin{bmatrix} w_k^x \\ w_k^y \\ w_k^\Phi \end{bmatrix} . \quad (1)$$

The state vector $\underline{x}_k = [x_k, y_k, \Phi_k]^{\mathrm{T}}$ comprises the pose at step $k$, while $\hat{s}_k^F$, $\hat{s}_k^S$ and $\hat{\omega}_k$ are the forward step width, the sideward step width, and the rotation angle of the current step, respectively. To offer robust dead-reckoning while neglecting subordinate physical effects, the high-level kinematic model (1) incorporates additive white Gaussian noise $\underline{w}_k = [w_k^x, w_k^y, w_k^\Phi]^{\mathrm{T}}$. The parameters of the noise vector were determined by means of a series of parameter identification runs.

### C. Technical Realization

In order to build robots as described above, several design challenges have to be met. While maintaining extremely low weight, the whole construction still has to be strong enough to withstand an everyday testbed situation. Additionally, the design has to be simple and cheap enough to allow volume production. Next to the desired small size, this leads to the need of very precise and automated manufacturing processes.

Each linear joint is built in a sandwich design with two connected stationary outer boards and a movable middle board, which is held by linear ball bearings from both sides. The middle board is actuated by a miniature rotatory servo motor.

The mechanical structure of the robots is built from a copper laminated composite material with a thickness of 1.5 mm. This material is also used to build printed circuit boards. Besides the high strength and the low weight of this material, it can be cut and milled very precisely with a CNC circuit board plotter, which allows for automatic production with very high precision, especially of the necessary grooves for ball bearings. The use of the copper laminated material has the additional advantage that all electrical components can be built directly onto the mechanical structure of the robot. Therefore, a high level of system integration can be achieved.

The robots are each controlled by a 16-bit Texas Instruments MSP430F1611 micro controller with a clock rate of 8 MHz, 10 KB RAM and 48 KB Flash memory. Therefore, computation and memory resources are available that allow not only on-board execution of the low-level motion control but also, if desired, of complex high-level control algorithms. Radio communication is made available through a MaxStream XBee module that allows ZigBee/IEEE 802.15.4 compliant communication at the 2.4 GHz frequency band. The necessary power is supplied by a 3.7 V, 720 mAh lithium polymer accumulator.

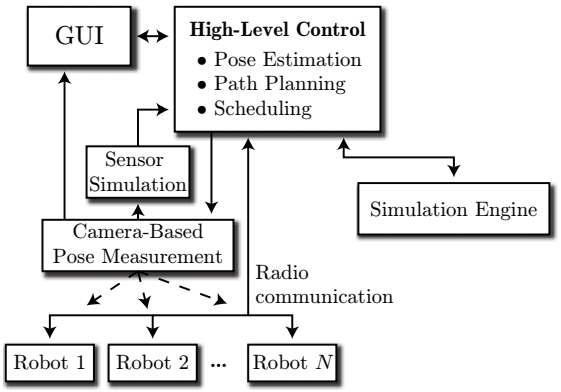


Fig. 4. System overview.

## III. Structure of the Test-Environment

The whole test-environment is designed in a modular fashion, which allows to easily exchange one of its modules. This is an important trait, since one typically wants to test a new algorithm for only a special module, e.g. the measurement scheduling, without having to alter the rest of the test-environment.

The robots, as presented in the last section, lay the foundation for the introduced test-environment. The overall structure of the test-environment is depicted in Fig. 4. All of its software components except the ones on the robot are realized using MathWorks MATLAB, which allows easy implementation and modification even of mathematically demanding modules.

### A. Graphical User Interface

Both, the experiments with real robots as well as the simulations can be controlled and monitored with a graphical user interface (GUI). Besides setting the experimental parameters, also the loading and saving of test scenarios is possible. In the center of the GUI is a display showing the true poses of the robots as well as their estimated positions, as illustrated in the plot in Fig. 6.

### B. High-Level Control

The high-level control comprises three submodules, pose estimation, path planning, and measurement scheduling, whose current instantiations are highlighted in Section IV. These modules are executed on a central control PC, which allows easy implementation and interchangeability of centralized as well as decentralized algorithms.

### C. Camera-Based Pose Measurement and Sensor Simulation

The robots' true poses are determined with an overhead color CCD camera on the basis of the position of two LEDs that are mounted on the front and back of each robot. The employed camera has a resolution of 1024 x 768 pixels and overlooks, after rectification, an area of 2.6 m x 1.8 m. The employed measurement system allows to determine the robots' poses with a resolution of well below 1 cm.

The acquired true robots' poses are used in a twofold manner. For one thing, they are used to generate various simulated sensor readings of the robots, which are used for pose estimation. It is for example possible to simulate ultrasonic distance sensors that give distance measurements between two robots, or an electronic compass. Thus, the benefits of various different sensor types and characteristics can be easily evaluated. Additionally, they are directly used in the GUI to provide user-feedback of the true robots' poses in comparison to the estimated ones that are provided by the pose estimation module (Section IV-A).

### D. Event-Driven Simulation Engine

All aspects of the real test-environment can also be simulated by an event-driven simulation engine that allows not only to evaluate algorithms prior to experimental runs with the real robots in time-lapse mode, but also to rerun experiments with modified control algorithms.

## IV. High-Level Robot Control

The main purpose of the provided test-environment is evaluating high-level algorithms for collaborative robot control under explicit consideration of uncertainties. The algorithms presented in this section are specific instantiations. The modular structure of the test-environment facilitates replacing the control submodules for convenient test, comparison, and improvement of the algorithms.

### A. Pose Estimation

In the following, dead-reckoning and distance measurements to other robots (from simulated ultrasonic transducers) are used for collaboratively estimating the pose of each robot. Due to the nonlinearity of the kinematic model (1) and the distance measurements, the well-known extended Kalman filter (EKF) [10] is utilized for an efficient estimation process. Here, a centralized EKF is employed for best estimates, even if a decentralized version is not expected to alter the results significantly.

*1) Dead-Reckoning:* While the mean pose vector $\hat{\underline{x}}_{k+1}^{(i)}$ of robot $i$ can be determined directly by means of the kinematic model (1), the covariance matrix estimation $\mathbf{C}_{k+1}^{(i)}$ requires linearization of the kinematic model at the mean vectors $\hat{\underline{x}}_k^{(i)}$ and $\hat{\underline{w}}_k$. This leads to the error propagation equation[1]

$$\begin{bmatrix} \bar{x}_{k+1} \\ \bar{y}_{k+1} \\ \bar{\Phi}_{k+1} \end{bmatrix} = \underbrace{\frac{\partial}{\partial \underline{x}_k} \underline{x}_{k+1}\Big|_{\hat{\underline{x}}_k}}_{=:\mathbf{A}_k} \cdot \bar{\underline{x}}_k + \underbrace{\frac{\partial}{\partial \underline{w}_k} \underline{x}_{k+1}\Big|_{\hat{\underline{w}}_k}}_{=:\mathbf{B}_k} \cdot \bar{\underline{w}}_k \ ,$$

with

$$\mathbf{A}_k = \begin{bmatrix} 1 & 0 & \left(\hat{s}_k^F \cdot \sin(\hat{\Phi}_k) + \hat{s}_k^S \cdot \cos(\hat{\Phi}_k)\right) \\ 0 & 1 & \left(\hat{s}_k^F \cdot \cos(\hat{\Phi}_k) - \hat{s}_k^S \cdot \sin(\hat{\Phi}_k)\right) \\ 0 & 0 & 1 \end{bmatrix} , \ \mathbf{B}_k = \mathbf{I} \ ,$$

where $\bar{\underline{x}}$ denotes the deviation of the estimate $\hat{\underline{x}}$ and the actual value $\underline{x}$. The matrices $\mathbf{A}_k$ and $\mathbf{B}_k$ are then used in the Kalman filter covariance time update step [11].

[1] Superscripts $(i)$ are omitted for improved readability.

*2) Distance Measurements:* Pose estimation can be significantly improved by incorporating additional sensor information. In the following, simulated distance measurements between the robots as introduced in Section III-C are used. For modeling the distance measurement between robot $i$ and $j$, the measurement equation (Euclidean distance)

$$d_k^{(i,j)} = \underbrace{\sqrt{\left(x_k^{(i)} - x_k^{(j)}\right)^2 + \left(y_k^{(i)} - y_k^{(j)}\right)^2}}_{=:h\left(\underline{x}_k^{(i)}, \underline{x}_k^{(j)}\right)} + v_k \quad (2)$$

is employed, where $d_k^{(i,j)}$ is the distance, $\underline{x}_k^{(i)}$ comprises the pose of robot $i$, and $v_k$ is additive white zero-mean Gaussian noise. Linearizing (2) at mean vectors $\hat{\underline{x}}_k^{(i)}$ and $\hat{\underline{x}}_k^{(j)}$ leads to

$$\bar{d}_k^{(i,j)} = \underbrace{\frac{\partial}{\partial \underline{x}_k^{(i,j)}} d_k^{(i,j)}\Big|_{\hat{\underline{x}}_k^{(i,j)}}}_{=:\mathbf{H}_k^{(i,j)}} \cdot \bar{\underline{x}}_k^{(i,j)} + \bar{v}_k \ ,$$

where

$$\underline{x}_k^{(i,j)} = \begin{bmatrix} \underline{x}_k^{(i)} \\ \underline{x}_k^{(j)} \end{bmatrix} \ , \quad \mathbf{H}_k^{(i,j)} = \begin{bmatrix} \mathbf{H}_k^{(i)} & \mathbf{H}_k^{(j)} \end{bmatrix} \ ,$$

$$\mathbf{H}_k^{(i)} = -\mathbf{H}_k^{(j)} = \frac{1}{h\left(\hat{\underline{x}}_k^{(i)}, \hat{\underline{x}}_k^{(j)}\right)} \begin{bmatrix} \hat{x}_k^{(i)} - \hat{x}_k^{(j)} \\ \hat{y}_k^{(i)} - \hat{y}_k^{(j)} \\ 0 \end{bmatrix}^{\mathrm{T}} .$$

With distance measurement $d_k^{(i,j)}$ and measurement matrix $\mathbf{H}_k^{(i,j)}$, applying the Kalman filter measurement update equations for mean vector and covariance matrix results, due to correlations, in improved pose estimates for all robots, especially robots $i$ and $j$.

### B. Measurement Scheduling

If $N$ is the number of robots, there are $\frac{1}{2} \cdot N \cdot (N-1)$ possible distance measurements between robots. Moreover, dead-reckoning offers localization over multiple steps to bridge the time between two distance measurements. To save energy and computational resources in real world scenarios, the number of performed distance measurements can be drastically reduced if a measurement selection strategy is employed. We propose a two-step scheduling strategy:

1) Robot preselection,
2) Selection of a landmark robot.

The robot preselection determines robots with the most uncertain pose estimation, while the landmark selection determines a landmark robot for each preselected robot that leads to the greatest uncertainty reduction.

*1) Robot Preselection:* As only distance measurements are performed, the uncertainty of the robots' orientations $\Phi_k^{(i)}$ could distort the preselection. To bypass this, only the submatrix $\tilde{\mathbf{C}}_k^{(i)}$ of $\mathbf{C}_k^{(i)}$ representing the position uncertainty of the robot is considered. Thus, the subvector $\tilde{\underline{x}}_k^{(i)}$ comprising the mean position from the mean pose vector $\hat{\underline{x}}_k^{(i)}$, i.e., $\tilde{\underline{x}}_k^{(i)} = [\hat{x}_k^{(i)}, \hat{y}_k^{(i)}]^{\mathrm{T}}$ is used.

As an example for preselecting robots, only the robot with the largest position uncertainty is used for performing a

distance measurement. Since the covariance matrix $\tilde{\mathbf{C}}_k^{(i)}$ represents the position uncertainty, this robot can be determined by

$$\arg\max_i \left\{ \psi(\tilde{\mathbf{C}}_k^{(i)}) \right\} \ ,$$

where the function $\psi(\cdot)$ quantifies the dimension of the covariance matrix. Typical choices for $\psi(\cdot)$ are the trace or the determinant. In the experiments, the trace is employed, as the trace of a covariance matrix is proportional to the circumference of the covariance ellipse.

*2) Selection of a Landmark Robot:* When using distance measurements, an increased uncertainty reduction and thus estimation improvement can be achieved if the distance measurement of the preselected robot $i$ is performed to a beneficial landmark. To quantify the quality of a landmark $j$, the so-called *Mahalanobis distance* [12]

$$M(\tilde{\underline{x}}_k^{(i)}, \tilde{\mathbf{C}}_k^{(i)}, \tilde{\underline{x}}_k^{(j)}) = (\tilde{\underline{x}}_k^{(i)} - \tilde{\underline{x}}_k^{(j)})^{\mathrm{T}} \left( \tilde{\mathbf{C}}_k^{(i)} \right)^{-1} (\tilde{\underline{x}}_k^{(i)} - \tilde{\underline{x}}_k^{(j)})$$

can be applied, where the position $\tilde{\underline{x}}_k^{(j)}$ of the landmark is supposed to be known with certainty. Minimizing the Mahalanobis distance leads to favoring landmarks along the longest axis of the robots' covariance ellipses.

In the presented setup, two important factors do not allow for using the Mahalanobis distance directly:

i) The precision of the employed simulated distance measurements is not affected by the distance between the landmark and the robot.
ii) Other robots act as landmarks. Thus, the position of a landmark $\tilde{\underline{x}}_k^{(j)}$ is uncertain to some degree.

Thus, we introduce the *normalized mutual Mahalanobis distance*

$$M^*(\tilde{\underline{x}}_k^{(i)}, \tilde{\mathbf{C}}_k^{(i)}, \tilde{\underline{x}}_k^{(j)}, \tilde{\mathbf{C}}_k^{(j)}) = \frac{(\tilde{\underline{x}}_k^{(i)} - \tilde{\underline{x}}_k^{(j)})^{\mathrm{T}} \left( \tilde{\mathbf{C}}_k^{(i)} \right)^{-1} \tilde{\mathbf{C}}_k^{(j)} (\tilde{\underline{x}}_k^{(i)} - \tilde{\underline{x}}_k^{(j)})}{(\tilde{\underline{x}}_k^{(i)} - \tilde{\underline{x}}_k^{(j)})^{\mathrm{T}} (\tilde{\underline{x}}_k^{(i)} - \tilde{\underline{x}}_k^{(j)})} \ , \quad (3)$$

which has the following important advantages:

i) $M^*(\tilde{\underline{x}}_k^{(i)}, \tilde{\mathbf{C}}_k^{(i)}, \tilde{\underline{x}}_k^{(j)}, \tilde{\mathbf{C}}_k^{(j)})$ is *normalized* by the denominator in (3) and thus is distance invariant.
ii) By inclusion of the uncertainty of the landmark robot $\tilde{\mathbf{C}}_k^{(j)}$, only landmarks with high-precision position estimates in the direction of robot $i$ are used *(mutual)*.

### C. Path Planning

Using the data from the pose estimation submodule, the path planning submodule employs a very basic, but in our case well-working force-based artificial potential field algorithm (for details see, e.g. [13], [14]). Here, each robot is intended to lower its potential energy and thus moves directed by a force corresponding to the gradient of the cumulative potential. Obstacles, i.e., other robots and the borders of the test arena, are modeled with high potentials and the goal region of each robot with low potential. It is well known that especially in existence of concave obstacles this class of approaches leads to local minima, which keep the robots from reaching their goal regions. In our case this is of minor relevance as no concave obstacles are employed and the considered scenarios are typically highly dynamic.

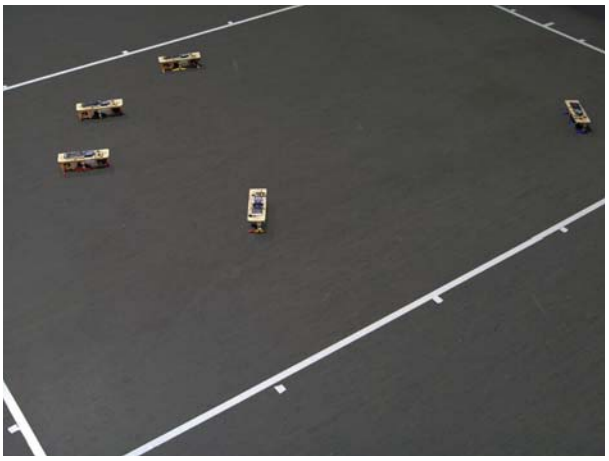

Fig. 5. Team of five miniature 6-DOF walking robots at their initial positions of the experiment.

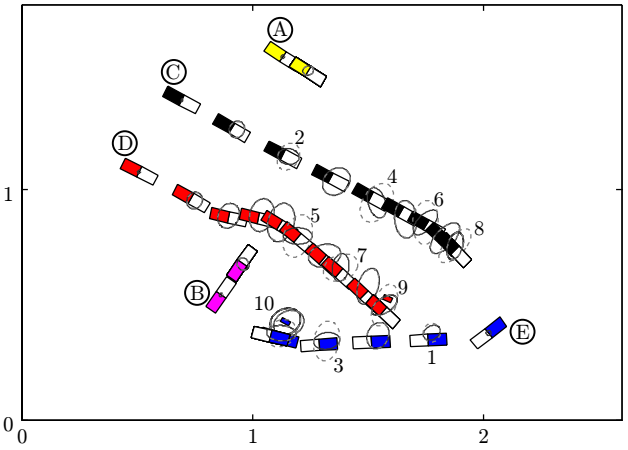


Fig. 6. GUI output of the experimental run. The rectangles indicate the true poses of the robots (as determined with the camera system). All robots are plotted whenever a distance measure between two robots is taken. The estimated positions of the robots with $3\sigma$-bounds are illustrated by the ellipses, where the light dashed gray ones indicate the position before the filtering, the dark solid ones after the incorporation of a distance measurement. The added robot letters are close to the initial positions of the robots, the added numbers indicate the measurement steps of the non-landmark robots that are selected for the distance measurement.

## V. Experiments

The interactions and abilities of the individual modules of the test-environment, especially the graphical user interface, the measurement scheduling submodule, the pose estimation submodule, and the path planning submodule, are illustrated in the following based on an experimental run with five of the miniature walking robots. Initially, the robots are randomly distributed in the test arena as depicted in Fig. 5. The target regions are chosen to generate the following scenario: Robot A (yellow) and robot B (pink) are intended to be

TABLE I

MEASUREMENT SEQUENCE.

| measurement # | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| robot $i$ (uncertain position) | E | C | E | C | D | C | D | C | D | E |
| robot $j$ (landmark) | A | A | A | B | A | A | B | B | A | A |

Robot labeling (letters) and measurement numbers as in Fig. 6.

landmarks and thus, their target regions are close to their initial positions. Robots C (black), D (red), and E (blue) are assigned target regions distant to their initial positions.

In Table I, the measurement sequence of the robots, determined by the measurement scheduling module employing the normalized mutual Mahalanobis distance (Section IV-B) is given, where a measurement is taken between two robots every 20 seconds. It can be seen that the preselected robot $i$ with highest uncertainty is always one from the three robots moving a longer distance. The corresponding landmark robot $j$ is always either robot A or B, whose position estimates are fairly precise and much more certain than the position estimates of the other three robots. Additionally, in Fig. 6 it can be seen that the position uncertainties of the robots (indicated by the ellipses) that are involved in the distance measurement decrease significantly and that always the robots with the most uncertain positions are chosen (e.g. Robot C in the 4th measurement and robot D in the 5th).

The position estimates of the robots are quite precise, the centers of each robot lie within the $3\sigma$ ellipses. However, especially the robots that travel a longer distance do not precisely reach their goal region. This can be explained by the linearization within the extended Kalman filter used for pose estimation. It is expected that by employing more advanced filters this can be resolved. Due to the modular structure, exchanging the employed filter is straightforward. The quality of the pose estimation submodule directly influences the path planning submodule, which operates on the basis of the estimated robot poses. The effectiveness of the employed path planning submodule can be seen especially well for e.g. robot D that avoids robot B that is in its direct path.

## VI. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a test-environment for the evaluation of algorithms from the field of collaborative robot control. The novel 6-DOF miniature walking robots that are employed in the test-environment offer a wide variety of different motion patterns. Thus, the high-level control algorithms are not limited to the typical two-wheel or differential drive robot kinematics. The modular structure of the introduced test-environment, comprising high-level control modules as well as infrastructure models, allows easy interchangeability of individual modules. A cooperative localization experiment with the focus on measurement scheduling, where a novel mutual normalized Mahalanobis distance is employed, illustrates the interaction of the individual modules of the test-environment.

Besides technical advances in the test-environment infrastructure as, e.g. integration of sensors into the robots, future research will be aimed at developing more advanced high-level control submodules. It is expected that the quality of the pose estimation submodule can be significantly increased if more advanced nonlinear filters [15] are incorporated and that the path planning and the measurement scheduling can be improved if non-myopic algorithms, e.g. employing [16], are used.

## ACKNOWLEDGEMENT

The authors want to gratefully acknowledge the support of their students Christof Chlebek, Susanne Dinkler, Christian Klaß, Andreas Motz, Farhad Omid, Dominik Perpeet, and Martin Schadow in building up the test-environment.

## REFERENCES

[1] L. Parker, *Current State of the Art in Distributed Autonomous Mobile Robotics*. Springer-Verlag Tokyo 2000, 2000, pp. 3–12.

[2] J. H. Reif and H. Wang, "Social potential fields: A distributed behavioral control for autonomous robots," *Robotics and Autonomous Systems*, vol. 27, pp. 171–194, 1999.

[3] A. I. Mourikis and S. I. Roumeliotis, "Optimal Sensor Scheduling for Resource-Constrained Localization of Mobile Robot Formations," *IEEE Transactions on Robotics*, vol. 22, no. 5, pp. 917–931, 2006.

[4] ——, "Performance Analysis of Multirobot Cooperative Localization," *IEEE Transactions on Robotics*, vol. 22, no. 4, pp. 666–681, 2006.

[5] S. Bergbreiter and K. Pister, "CotsBots: An Off-the-Shelf Platform for Distributed Robotics," in *Proc. of 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*, vol. 2, October 2003, pp. 1632–1637.

[6] K-Team, "Khepera III," http://www.k-team.com.

[7] F. Sahin, "Groundscouts: Architecture for a Modular Micro Robotic Platform for Swarm Intelligence and Cooperative Robotics," in *Proc. of 2004 IEEE International Conference on Systems, Man and Cybernetics*, vol. 1, October 2004, pp. 929–934.

[8] L. Navarro-Serment, R. Grabowski, C. Paredis, and P. Khosla, "Millibots," *Robotics & Automation Magazine, IEEE*, vol. 9, no. 4, pp. 31–40, Dec. 2002.

[9] F. Weissel and U. D. Hanebeck, "A Test-Environment for Control Schemes in the Field of Collaborative Robots and Swarm Intelligence," in *Proc. of the 7th Intl. Workshop on Computer Science and Information Technologies*, vol. 1, September 2005.

[10] D. Simon, *Optimal State Estimation: Kalman, H-Infinity, and Nonlinear Approaches*, 1st ed. Wiley & Sons, 2006.

[11] R. E. Kalman, "A new Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME, Journal of Basic Engineering*, no. 82, pp. 35–45, 1960.

[12] F. Zhao, J. Shin, and J. Reich, "Information-Driven Dynamic Sensor Collaboration," *IEEE Signal Processing Magazine*, vol. 19, pp. 61–72, March 2002.

[13] F. Miyazaki and S. Arimoto, "Sensory Feedback based on the Artifical Potential for Robots," in *Proceedings of the 9th IFAC World Congress*, 1984, pp. 2381–2386.

[14] V. V. Pavlov and A. N. Voronin, "The Method of Potential Functions for Coding Constraints of the External Space in an Intelligent Mobile Robot," in *Soviet Automatic Control*, vol. 17, no. 6, 1984, pp. 45–51.

[15] M. F. Huber and U. D. Hanebeck, "The Hybrid Density Filter for Nonlinear Estimation based on Hybrid Conditional Density Approximation," in *10th International Conference on Information Fusion (Fusion 2007)*, Quebec, Canada, July 2007.

[16] F. Weissel, M. F. Huber, and U. D. Hanebeck, "Efficient Control of Nonlinear Noise-Corrupted Systems Using a Novel Model Predictive Control Framework," in *Proc. of the 2007 American Control Conference*, 2007.