SENSING AND ESTIMATION ON A MODULAR TESTBED FOR SWARM ROBOTICS

Gregory K. Fricke, Devendra P. Garg*
Robotics and Manufacturing Automation Laboratory
Mechanical Engineering and Materials Science Department
Duke University
{gkf4, dpgarg}@duke.edu

Dejan Milutinović
Applied Mathematics and Statistics Department
University of California at Santa Cruz
dejan@soe.ucsc.edu

ABSTRACT

Collective robotics offers the promise of enhanced performance and robustness relative to that of individual robots, with decreased cost or time-to-completion for certain tasks. Having many degrees of freedom, the multi-robot control and estimation problems are challenging, specifically when the solutions require a great amount of communication among the robots. While numerical simulation is a critical tool in swarm robotics research, verification of obtained results under a physical realization of the swarm is far from routine. Therefore, we have developed and used a sensor-integrated testbed for the validation of cooperative-robotics algorithms, observation of swarm behavior, and measurement of system performance.

INTRODUCTION

Conducting experiments in swarm robotics is challenging, time-consuming, and expensive, leading many investigators in the field to extensively rely on computer simulations. While simulation is indispensable for algorithm development and performance estimation, the physical realization of a swarm system is important for full understanding and validation of swarm behavior. Also, physical experiments shed light on the nature of uncertainty sources in the sensors, actuators, and communication pathways specific to mobile robotic systems while illustrating the logistical difficulties involved in implementing swarm behavior.

Robot pose estimation is critical for feedback control of robot trajectory. Methods for pose estimation and robot localization have been extensively studied, utilizing methods such as dead-reckoning [1–6], on-board and off-board computer vision [7,8], environmental sensing [9,10], collective or distributed localization [11, 12], and simultaneous localization and mapping (SLAM) [13] using laser-scanning systems [14, 15], on-board cameras [16], RFID tags and readers [17,18], or ultrasonic rangefinders [10].

Pose estimation and tracking of individual robots are critical to the metrology required for complete evaluation of experimental results. In the case of multi-robot systems, the scalability of the metrology system is a critical concern. With multiple robots, reliance on advanced vision-based techniques quickly surpasses all but the most advanced computational systems, imposing the restriction that performance data must be evaluated off-line without real-time constraints.

Figure 1. Khepera-II arena of the RAMA Lab.

*Address all correspondence to this author.

The modular testbed described in this paper has been developed at the Robotics and Manufacturing Automation (RAMA) Laboratory at Duke University. The testbed, shown in Fig. 1, utilizes multiple Khepera-II robots operating within a defined swarm arena equipped with two SICK LMS400 LADAR sensors and an overhead Cognex Insight 5400 industrial machine-vision system. Researchers at other institutions have recently described similar multi-robot testbeds, e.g., see [15, 19, 20]. Our focus has been to maximize utility, maintain scalability, and avoid overly specialized and cost-prohibitive equipment.

The paper is organized as follows. First, the swarm members, LADAR units, and vision sensor are described. Next, the estimation methods developed to utilize the data are discussed, and the communication methods employed are presented. The final sections are devoted to the results of simulated experiments using real raw data, including a discussion of plans for future development.

SWARM MEMBERS

The swarm members in use are Khepera-II robots built by K-Team, shown in Fig. 2. The robots are non-holonomic, featuring two-wheeled differential-drive. The wheels are independently controllable via calls to supplied position- or velocity-control functions, and their positions are measured by wheel encoders with $\frac{1}{12}mm$ resolution. According to Canudas de Wit's wheeled-robot definitions in [21], this is a Type (0,2) robot. The Khepera-II robot bases are nominally $69mm$ in diameter, and $34mm$ tall (from bottom of wheels to top of base circuit board). The wheels are $15mm$ in diameter, and the wheel separation is $52.5mm$ from center to center.

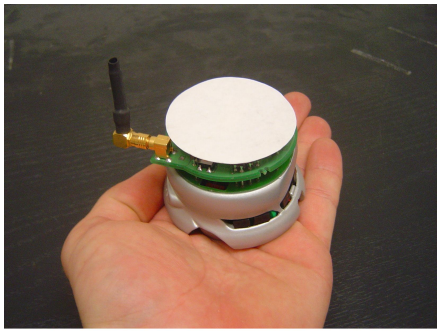

Figure 2. Khepera-II with top turret covered by white paper disc.

The robots utilize a Motorola 32-bit processor and Flash memory for storing user-created programs. Each robot has eight infrared proximity sensors, a modular expansion system allowing additional functionality, and a rechargeable battery system providing roughly one hour of operation on full charge.

In the following subsections, we provide a brief description of the robot components, including the difficulties in using them based on our experience.

Programming

Implementing custom operational programs for the robots is straightforward when utilizing the KTProject library of functions (included with the purchase of Khepera-II robots or via download from the K-Team website). The KTProject IDE includes several sample C source code files that can be quickly compiled and downloaded to the robots via the wired COM link. The robots provide a user-flash capability, allowing the user to burn an image into memory after download.

Motion Control

The library of functions provided in the KTProject IDE includes closed-loop motor control functions. The inclusion of such functions facilitates simple development and quick integration of user functions, albeit with some shortcomings.

While the supplied wheel velocity PID controller delivers fairly repeatable results, there is an apparent lack of an integrator anti-windup mechanism. Thus, if a robot encounters an object that impedes its commanded motion, the torque commands to the wheels continue to increase until either the wheel-surface friction is overcome or the robot pushes past the obstacle. In the latter case, the robot accelerates to velocities much greater than the maximum values set in the controller. This issue must be overcome with custom programming.

The position control is quite useful for commanding open-loop trajectories with simple geometry. It uses a time-optimal velocity profile, commanding the maximum acceleration until maximum velocity is achieved, followed by a coast phase, and concluding with maximum deceleration to stop at the commanded position. This time-optimal servo-control works well for single-degree-of freedom actuators. This 1-DOF control method fails for motion of a body whose kinematics couple the actuators. Since the target position is expressed as a specific number of encoder counts for each wheel, it is easy to see that the final position and orientation of the robot may vary greatly from the target position and orientation if the rate of each wheel is not monitored closely. Small errors in the control loops cause significant variations in robot angular rate, i.e., changing the orientation or heading.

Sensing Ability

Each robot is equipped with eight infrared proximity sensors. When used to detect non-absorptive and non-dissipative obstacles, the sensors perform well with a usable range of roughly 10-50mm. The minimum detection distance does not adversely affect the utility of the data, as the sensors are set back in the body of the robot roughly $10mm$. The steep decay of the negative-exponential sensing function may induce large errors when using the proximity sensors for distance measurement. Additionally, the maximum detection distance may vary widely depending on

the incident surface and ambient light level. In particular, flat-black surfaces prove nearly undetectable by the sensors. As such, these sensors are most effectively used to detect the *presence of* an obstacle rather than to measure the *distance to* an obstacle.

The sensor data is easily accessed through the included software library. The difficulty lies in using the data when the objects in the environment vary in reflectivity.

Communication modes

The modular expansion system of the Khepera-II robots allows the capabilities to be extended via modules called turrets, including the General Purpose I/O turret (GPIO), the Gripper Turret, the Radio Turret (RT) and the High Speed Radio Turret (HSRT). Without the radio turrets, only simple, wired two-way communication with typical PC hardware is possible. The communication method to be used is dictated largely by the specific experiment being conducted. In this research, we consider both robot-to-computer and robot-to-robot communication.

For *robot-to-robot* communication, the RT must be used, as the HSRT is incapable of direct communication between robots. This limitation of the HSRT arises due to the host-client paradigm of the Bluetooth protocol. In a decentralized control environment, a centralized communication "switchboard" would be required in order to utilize the HSRT. Additionally, systems using the HSRT are limited to seven members (again due to the Bluetooth protocol).

For *robot-to-computer* communication, there are three off-the-shelf choices. The first, cheapest, easiest, and most reliable choice is to use the direct-wired serial port. This mode of communication has proved to be reliable both for executing the built-in functions of the robots as well as for initiating a custom executable and receiving status information from the robots. For development, the wired communication is the best choice. However, the physical constraint of the wires eliminates this option in a multiple robot or high-mobility setting. The second choice is the HSRT, which allows relatively high data transmission rates (up to 115200 *baud*). The final choice is the RT, which has proven to be somewhat less than optimal. The data rate is limited to only 9600 *baud*, and the connection is very susceptible to radio frequency interference. For simple commanding of the robots, the RT has proven to be adequate, but establishing reliable communication under a custom executable has not been achieved. The RT is currently in use for all of the experiments underway at the RAMA Lab, but in the future either the HSRT or a custom alternative (such as the custom IR communication turrets developed in [19]) must be employed.

SENSORS

External position measurements are available to the central supervisor from a Cognex Insight 5400 vision system and a pair of SICK LMS400 LADAR rangefinders. The vision system is configured to return arbitrarily-ordered pairs of x–y position data for a pre-specified number of detected robots. To maximize the sampling frequency, we use simple blob detection based on a background contrast comparison and an area threshold. In order to provide ample contrast with the black-painted arena, identical discs of white paper are affixed to the uppermost turret of each Khepera-II robot (see Fig. 2). Doing so, however, nullifies the vision system's ability to measure the orientation of the blobs, as there is no asymmetry. Additionally, in a multi-robot setting the robots are indistinguishable from one another, requiring a limited-information identification method. The Cognex vision system provides robust and accurate positional data, but if multiple robots are present in the field of view, the system is not able to uniquely distinguish them from one another. The issue of maintaining track of individual robots in order to properly and consistently apply the measurements is addressed below.

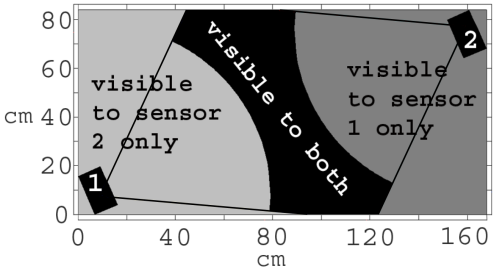

Figure 3. LADAR Coverage of the RAMA Lab Swarm Arena

Range measurements across the arena are provided by two SICK LMS400-1000 industrial LADAR Measurement Systems, communicating with the central host via a 10MBit, half-duplex Ethernet (TCP/IP) connection. The units utilize diode lasers and a rotating mirror to sweep across 70°.

The LMS400 LADAR unit allows the user to configure several parameters of the scan. Scan frequency may be set from $360Hz$-$500Hz$, with angular resolution ranging from 0.1333°-1.0°. Measured distances (at each angular point) are reported from $500mm$-$3000mm$. Range resolution is $1mm$, although the reported systematic measurement standard error is $4mm$ for typical remission values of 40-100%, reaching as high as $10mm$ for lower remission values. Remission values below 6.5% lead to invalid measurements, reported explicitly as $0mm$. To facilitate optimal data collection, range, edge, median, and mean filters may be set. In the experiments conducted, these capabilites have not been exploited; all such parameters were left at factory default values.

Two LADAR units are utilized in the RAMA Lab swarm arena for three basic reasons. The first two are the minimum range and limited detection angle of the LADAR units. Utilizing two units at opposite ends of the arena ensures complete coverage of the entire arena by at least one sensor (see Figure 3). The third reason is the susceptibility to line-of-sight obstruction; that is, one robot may occlude one (or more) other robot(s) lying within the wedge defined by the tangents from the first robot to

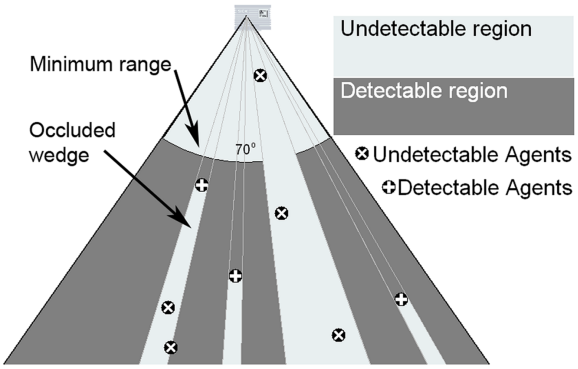

Figure 4. Partial and full occlusion of distant robots by a proximal robot. Image is approximately to scale.

the center-point of the LADAR unit (see Figure 4).

Each of the two SICK LMS400 LADAR sensors returns a set of range data covering the 70° "wedge", discretized into 0.25° samples. At each angular point, a range measurement is returned. This data is simply a sweep over the field of view of the sensor, so the positions of the robots within the field of view must be extracted. This is discussed later in the *Arc Detection* subsection of the *ESTIMATION METHODS* section.

ESTIMATION METHODS

Due to the nearly uniform cylindrical shape of the Khepera-II robots, it is not possible to directly extract the orientation from the range data returned by the LADAR sensors. The vision system has the capability of returning orientation data provided there are distinguishable features on the robot when viewed from above; however, the orientation measurements require significant additional processing time in a scalable problem such as swarm or cooperative robotics, and are thus disabled.

Reliance on visual pattern recognition for extracting orientation and identity information is used by other research groups, e.g., see [19, 22], as well as in earlier experiments by our group [23]. Robot orientation or pose is also estimated under many SLAM implementations, e.g., see [10, 15]. Our approach eliminates the need for complex machine-vision software and processing capabilities.

Algorithms have been developed to allow data collected by the vision and LADAR sensors to be used for experimental data collection and performance validation (metrology), as well as for feedback to the swarm.

Arc Detection

The range data from the SICK LADAR units is a raw set of points, so the data must be processed to extract the locations of the robots. Traditional methods for extracting features from this sort of 1-D data include wavelet transforms [24] and the Circular Hough Transform [25, 26]. Wavelets can be used quite successfully for locating, within a set of data over an independent variable, features of interest. For perceiving specific shapes, appropriate wavelets must be used.

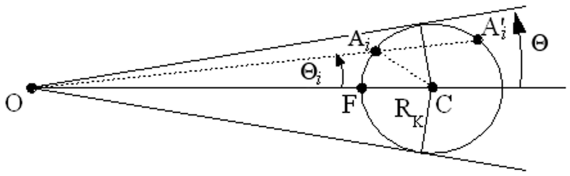

Figure 5. Geometric relationships for arc detection in LADAR data. R_K is the radius of the Khepera-II robot. The angular extent of the robot at range $\overline{OF}$ is 2Θ.

The authors have developed another method for arc detection, akin to a correlation function. With knowledge of the geometry of the Khepera-II robots, it is possible to predict the signal returned by the LADAR sensors if a robot is in its field of view. Simple trigonometry defines the angular extent (2Θ) of the robot at a given range ρ, as illustrated in Fig. 5. The formulae and algorithm for calculating the predicted arc are given in Eq.(1). The predicted arc is stored to a vector $S = \{S_{-\frac{N-1}{2}}, ..., S_0, ..., S_{\frac{N-1}{2}}\}$ of length N. The incremental angle used in the algorithm is $\Delta\Theta$, and the estimated range to the robot at angle Θ_i is S_i. Relating Fig. 5 to Eq.(1): $\rho = \overline{OF}$, $\psi_i = \angle OA_iC$, $\phi_i = \angle OCA_i$, $R_K = \overline{CA}_i = \overline{CF}$, $S_i = \overline{OA}_i$.

$$
\begin{aligned}
&\Theta = \arcsin\left(\frac{R_K}{\rho + R_K}\right) \\
&N = \text{floor}\left(\frac{2\Theta}{\Delta\Theta}\right) \\
&S_0 = \rho \\
&\text{if } N \text{ even, } N = N - 1 \\
&\text{for } i \in \left\{-\frac{N-1}{2}, \ldots, -1, 1, \ldots, \frac{N-1}{2}\right\} \\
&\quad \Theta_i = |i\Delta\Theta| \\
&\quad \psi_i = \arcsin\left(\frac{\rho + R_k}{R_k}\sin\Theta_i\right) \\
&\quad \text{if } \psi_i < \pi/2, \ \psi_i = \pi - \psi_i \\
&\quad \phi_i = \pi - \psi_i - \Theta_i \\
&\quad S_i = R_K\frac{\sin\phi_i}{\sin\Theta_i}
\end{aligned}
\tag{1}
$$

At short ranges just larger than sensor minimum range, the robots extend over approximately 5.46°, which corresponds to 21 sample points at 0.25° angular resolution. At the maximum extent of the arena, the robots exhibit an angular extent of approximately 1.8°, extending over only 7 sample points.

Using these equations, a *predicted arc* is created for every point in the raw data. The raw data surrounding the current

point is truncated to the same length as the predicted arc, i.e., N in Eq.(1). These two vectors are unit-normalized, and their inner-product is evaluated. The resulting scalar is then multiplied by a confidence factor; this final product is the *score* for that range point. Empirically, we have found that N is a very good measure of confidence, giving strong confidence at short ranges (with many points to fit to the arc) and weak confidence at longer ranges (with few points to fit).

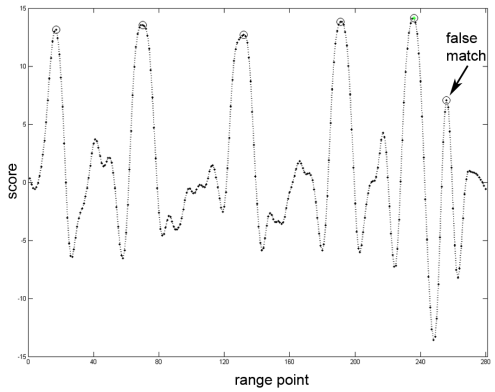

Figure 6. A sample score vector with five robots in the field of view. The right-most peak is incorrectly marked as a match since the algorithm was configured to find six robots.

Upon completion of the point-by-point scoring described above, the top scores are selected. The score data is clearly of the same length as the range data, in our case 280 points (a sample set of score data is shown in Fig. 6). The arc of a single robot clearly extends over N points (N appropriate for that range). Thus the maximum values in the score data must be separated by the appropriate number of points. When combined with a smoothing algorithm this method has proved to be, although not perfect, quite robust at identifying robots within the LADAR field of view.

This algorithm requires the number of robots to be known, so in cases where there are fewer robots in the field of view (due to range limits or occlusion as discussed above) false matches are guaranteed. The handling of these situations is discussed in the *Tracking* subsection below.

2nd Order KF

The central supervisor utilizes a 2nd-order Kalman Filter [27, 28] to maintain the four estimated states of each robot. In addition to estimating the x-y position, the filter estimates the velocity and orientation angle of each robot.

The robots are non-holonomic, thus the kinematic model can be exploited to extract the latent variable θ, the orientation angle, without knowledge of the control input. A novel addition to the typical extended Kalman Filter is the variation of the process noise intensity of state θ (the robot orientation) as a function of another state v (robot linear velocity). The state and measurement models are briefly shown below in Eqs.(2)-(3). The discrete model we use in the 2nd-order Kalman filter is defined in greater detail, and results are provided and discussed in [29].

$$\dot{X}(t) = \underbrace{\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{v}(t) \\ \dot{\theta}(t) \end{bmatrix}}_{X(t)=[x_1\ x_2\ x_3\ x_4]^T} = \underbrace{\begin{bmatrix} v(t)\cos\theta(t) \\ v(t)\sin\theta(t) \\ 0 \\ 0 \end{bmatrix}}_{\mathbf{F}(X(t))=[f_1\ f_2\ f_3\ f_4]^T} + \begin{bmatrix} 0 \\ 0 \\ \xi_v(t) \\ \xi_\theta(t) \end{bmatrix} \tag{2}$$

$$Z(t) = \begin{bmatrix} x_m(t) \\ y_m(t) \end{bmatrix} = \underbrace{\begin{bmatrix} 1\,0\,0\,0 \\ 0\,1\,0\,0 \end{bmatrix}}_{C} X(t) + \begin{bmatrix} w_x \\ w_y \end{bmatrix} \tag{3}$$

In the constant-noise model, the control variables $\dot{v}$ and $\dot{\theta}$ are modeled with zero-mean white noises $\xi_v(t)$ and $\xi_\theta(t)$, respectively. In the varying-noise model, $\xi_\theta(t)$ is treated as a function of state v, the linear velocity. This varying-noise model is discussed in depth in [23,29]. Measurements provided by the vision system and/or LADAR arc-detection algorithm are absolute position measurements (x_m and y_m) assumed to include zero-mean Gaussian-distributed noise $w_x \sim N(0, W_x)$ and $w_y \sim N(0, W_y)$. This measurement model is defined by Eq. (3).

The *prediction* is performed at each time step, initially updating states before applying the correction. These estimates are then propagated and updated in typical form. Without measurements, the system propagates only via the nonlinear state transition function. In update steps where measurements are available and valid, the *innovation* is performed, utilizing the 2nd order Taylor expansion of the dynamic model in Eq.(2). (The interested reader is directed to references [27–29] for further details.)

Tracking

When multiple robots are present in the field-of-regard for the LADAR and vision systems, the order of measurements may be rearranged from sample to sample. Additionally, there may be occasional excessive error in the measurements due to a sensor glitch or communication error, or false identifications returned either by the arc-detection described above or by the vision system. A matching and validation algorithm has been developed to detect and correct these occurrences to the extent possible.

The matching portion of the algorithm is based on the *Hungarian Algorithm*, first identified by Kuhn [30] and further developed by Munkres [31]. The algorithm solves the minimum-cost assignment problem. We treat the error between a measurement and an estimate as the cost, and the algorithm selects the order that minimizes the error.

If one or more of the measurements exhibits error that is too large due to background clutter or other error, those measure-

ments would still be applied erroneously in the estimation algorithm. The method for data validation is adapted from Leonard's geometric beacon tracking method of localization [9].

At each prediction step k of the central supervisor's EKF, a *validation gate* g is established around the estimated output of the i-th robot, $\hat{Z}^i_{k+1} = C\hat{X}^i_k$. The range of the validation gate is dependent on the confidence in the prediction, i.e., the predicted error covariance of the estimate, S. At time-step k, the predicted state for the i-th robot is compared to the j-th measurement, Z^j_k.

$$\begin{aligned} V_{ij} &= Z^j_k - \hat{Z}^i_k \\ g^2 &\geq V_{ij} S^{-1}_{ij} V^T_{ij} \end{aligned} \tag{4}$$

Leonard refers to the size of gate g as a "number of standard deviations" of acceptable error. Bar-Shalom [32] notes instead that the confidence region defined by g is χ^2-distributed, and thus the dimension of the measurement vector must be considered. For our 2-dimensional measurement vector, a one-sided confidence region of 68.2% (roughly equivalent to one standard deviation) is given by

$$\begin{aligned} g^2 &= \chi^2_2[0.318] \approx 2.29 \\ g &\approx 1.51 \end{aligned}$$

For the matching to be successful, the Kalman Filter described above must maintain estimates within a certain error bound. Experiments and simulations at the RAMA Lab continue in the search for the appropriate relationships between acceptable error within the KF estimate and the appropriate value for g that is not too restrictive. The lowest value found to work with the real data, described below, which provides rejection of bad data but allows tracking is $g \approx 2$, corresponding to a confidence region of 86.5%.

Metrology

In a swarm experiment, external observation of the entire swarm is desirable to quantify the performance of the algorithm as well as to verify the completion of the task. For purposes such as this, it is especially beneficial to use a sensor (or set of sensors) whose observational range extends over the entire region of interest such that all robots may be observed simultaneously.

Under these considerations, the vision system implemented here is ideal. Its field of view extends beyond the bounds of the swarm arena, and none of the robots is ever occluded. Additionally, the camera provides accurate position measurement ($\sigma \approx 0.03mm$) at reasonable sampling rate ($\approx$ 1.5Hz for 6 robots).

The major limitation of the vision system is in the scalability: the more robots are present in the field, the longer the sample period becomes as the system processes the images. The use of blob detection ensures that the sample period is minimized.

The LADAR system offers a trade-off, providing a higher sample rate that is independent of the number of robots in the field as long as the communication path is not a bottleneck. As this data is processed off-sensor, the sensor sample rate is maximized. The sacrifices to achieve this data rate are the limited and/or occluded field of view, requiring two (or more) sensors to be used for full arena coverage, and the increased sensor noise ($4mm$ standard error versus $0.03mm$ for the vision system).

Feedback

In the feedback setting, the timeliness of data samples and associated processing time becomes critical. We are currently performing experiments in which position data from the external sensors is provided to the robots at a reduced data rate. Using an odometry calibration method based on Larsen's and Martinoli's augmented Kalman Filter method [3, 5], simulations have shown that the robots require very low update rates to maintain localization accuracy. Experiments are underway to verify the performance in our real environment.

RESULTS

Data was collected from the Cognex InSight 5400 vision system with a single robot moving under the Braitenberg obstacle avoidance algorithm built in to the Khepera-II standard library. The implementation is a variant of Braitenberg's Vehicle 3b, called *Like* [33]. The resulting trajectories are semi-deterministic and depend strongly on the initial pose relative to the arena geometry as well as the variation among the onboard IR proximity sensors. These data sets were gathered using a distinct identifier disc on the robot displaying a spatial pattern such that the vision system can make direct measurements of orientation. This was desired to have a *truth* measurement for comparison with the estimated angle from the modified 2nd order Kalman Filter. The trajectories are plotted in Figure 7. The figures exclude measurements that are invalid due to misidentification or failed pattern-matching. Those bad points are identified in the algorithm and are treated as 'skipped' measurements.

The orientation of the robots in these trajectories was estimated under the model described above using both a state-varying noise model and a constant noise model. The results are given in Table 1.

	constant	varying	%better	(worse)
Set 1	0.1382	0.1475		(6.75)
Set 2	0.2212	0.2030	8.23	
Set 4	0.1276	0.1389		(8.85)
Set 5	0.2155	0.2005	7.00	

Table 1. Estimation scores (RMS errors) for the constant-noise and varying-noise models, expressed in radians.

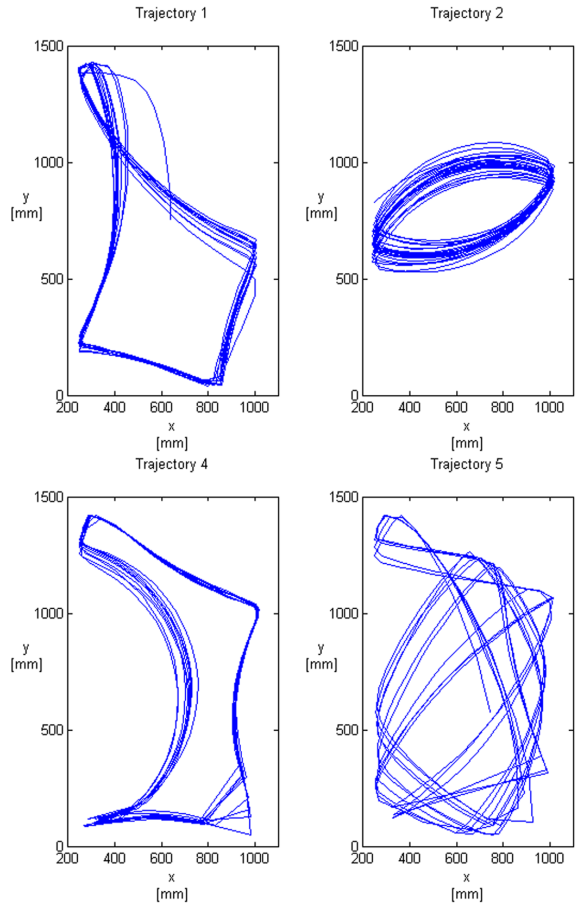

Figure 7. Four of the five measured trajectories.

The relative performance of the varying-noise model compared to the constant-noise model was evaluated by comparing the mean squared error between the measured and estimated angle, i.e., $\sqrt{\sum_k (\mathrm{mod}(\theta_k - \hat{\theta}_k, 2\pi))^2}$. The modulo function is critical to account for angle-wrapping, i.e., the fact that $\hat{\theta} = \hat{\theta} + 2\pi$.

CONCLUSION

A functional arena for experiments in swarm robotics has been established. Sensing and estimation capabilities have been implemented, providing good metrology results as well as the ability to provide online feedback to swarm members. Preliminary experiments have shown that this testbed will provide a good basis for future swarm research and experiments at the RAMA Lab.

Future development plans include improved communication ability. Under consideration is the use of both RTs and HSRTs, for robot-robot and robot-supervisor communication respectively. Alternatively, GumStix small-form-factor computers may be used alongside the HSRT (using Bluetooth communication) to allow robot-to-robot communication. This option allows the inclusion in our facility of other robots with Bluetooth communication ability, such as the iRobot Create.

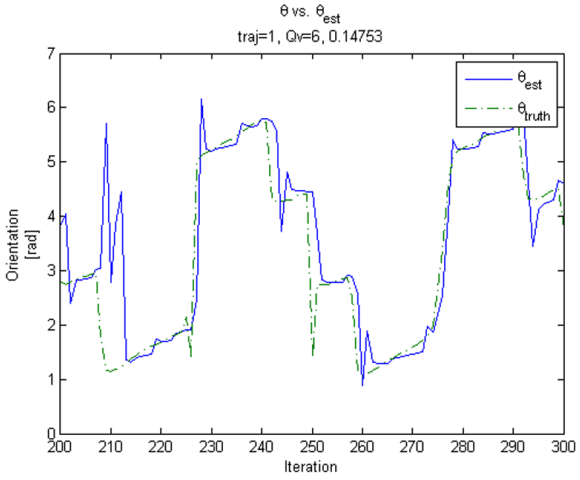

Figure 8. A sub-sample of trajectory set 1 comparing θ_{truth} to $\hat{\theta}$ for the varying-noise model.

This testbed and the estimation methods we have developed present us with opportunities for real-world validation. The estimation algorithms have been developed to be independent of the specific sensors. The arc-detection method may be applied to other sets of range-data and other shapes to be detected. The orientation estimation may be applied to any sensor that provides 2D position data (e.g., vision, GPS, RFID tag/reader, ultrasonic, etc.), maximizing our choices.

ACKNOWLEDGMENT

This work was supported by the Army Research Office under grant number W911NF-08-1-0106.

REFERENCES

[1] Borenstein, J., and Feng, L., 1996. "Measurement and correction of systematic odometry errors in mobile robots". *Robotics and Automation, IEEE Transactions on,* **12**(6), Dec, pp. 869–880.

[2] Chong, K. S., and Kleeman, L., 1997. "Accurate odometry and error modelling for a mobile robot". *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on,* **4**, Apr, pp. 2783–2788.

[3] Larsen, T. D., Bak, M., Andersen, N. A., and Ravn, O., 1998. "Location estimation for an autonomously guided vehicle using an augmented Kalman filter to autocalibrate the odometry". *FUSION98 SPIE Conference, Las Vegas*, Jul.

[4] Von der Hardt, H.-J., Husson, R., and Wolf, D., 1998. "An automatic calibration method for a multisensor system: application to a mobile robot localization system". *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on,* **4**, May, pp. 3141–3146.

[5] Martinelli, A., Tomatis, N., Tapus, A., and Siegwart, R., 2003. "Simultaneous localization and odometry calibration for mobile robot". *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on,* **2**, Oct., pp. 1499–1504.

[6] Antonelli, G., and Chiaverini, S., 2006. "Linear estimation of the odometric parameters for differential-drive mobile robots". *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, Oct., pp. 3287–3292.

[7] Murata, S., and Hirose, T., 1993. "Onboard locating system using real-time image processing for a self-navigating vehicle". *Industrial Electronics, IEEE Transactions on,* **40**(1), Feb, pp. 145–154.

[8] Kato, K., Ishiguro, H., and Barth, M., 1999. "Identifying and localizing robots in a multi-robot system environment". *Intelligent Robots and Systems, 1999. IROS '99. Proceedings. 1999 IEEE/RSJ International Conference on,* **2**, pp. 966–971.

[9] Leonard, J., and Durrant-Whyte, H., 1991. "Mobile robot localization by tracking geometric beacons". *Robotics and Automation, IEEE Transactions on,* **7**(3), Jun, pp. 376–382.

[10] Lin, H.-H., Tsai, C.-C., and Hsu, J.-C., 2008. "Ultrasonic localization and pose tracking of an autonomous mobile robot via fuzzy adaptive extended information filtering". *Instrumentation and Measurement, IEEE Transactions on,* **57**(9), Sept., pp. 2024–2034.

[11] Rekleitis, I., Dudek, G., and Milios, E., 1997. "Multi-robot exploration of an unknown environment, efficiently reducing the odometry error". *International Joint Conference in Artificial Intelligence (IJCAI)*.

[12] Roumeliotis, S., and Bekey, G., 2002. "Distributed multi-robot localization". *Robotics and Automation, IEEE Transactions on,* **18**(5), Oct, pp. 781–795.

[13] Thrun, S., 2008. *Simultaneous Localization and Mapping*. Springer Berlin / Heidelberg, pp. 13–41.

[14] Lu, F., and Milios, E., 1997. "Robot pose estimation in unknown environments by matching 2D range scans". *Journal of Intelligent and Robotic Systems,* **18**(3), pp. 249–275.

[15] Stachniss, C., 2006. "Exploration and mapping with mobile robots". PhD thesis, University of Freiburg, Department of Computer Science, April.

[16] Se, S., Lowe, D., and Little, J., 2005. "Vision-based global localization and mapping for mobile robots". *Robotics, IEEE Transactions on,* **21**(3), June, pp. 364–375.

[17] Hahnel, D., Burgard, W., Fox, D., Fishkin, K., and Philipose, M., 2004. "Mapping and localization with RFID technology". In Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on, Vol. 1, pp. 1015–1020.

[18] Deyle, T., Kemp, C., and Reynolds, M., 2008. "Probabilistic UHF RFID tag pose estimation with multiple antennas and a multipath RF propagation model". *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, Sept., pp. 1379–1384.

[19] Hoyt, S., McKennoch, S., and Bushnell, L. G., 2005. "An autonomous multi-agent testbed using infrared wireless communication and localization". *UWEE Technical Report Number UWEETR-2005-0005*.

[20] Holland, O., Woods, J., De Nardi, R., and Clark, A., 2005. "Beyond swarm intelligence: the UltraSwarm". *Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE*, June, pp. 217–224.

[21] Canudas de Wit, C., 1996. *Theory of Robot Control*. Spring-Verlag.

[22] Jung, D., Heinzmann, J., and Zelinksy, A., 1998. "Range and pose estimation for visual servoing of a mobile robot". *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on,* **2**, May, pp. 1226–1231.

[23] Fricke, G. K., 2009. "Localization, tracking, and odometry calibration of a multi-agent swarm system". Master's thesis, Duke University, Department of Mechanical Engineering and Materials Science, Durham, NC, April.

[24] Daubechies, I., 1990. "The wavelet transform, time-frequency localization and signal analysis". *Information Theory, IEEE Transactions on,* **36**(5), Sep, pp. 961–1005.

[25] Hough, P. V. C., December 18, 1962. Method and means for recognising complex patterns. U.S. Patent 3069654.

[26] Duda, R. O., and Hart, P. E., 1972. "Use of the Hough transformation to detect lines and curves in pictures". *Commun. ACM,* **15**(1), pp. 11–15.

[27] Kalman, R. E., 1960. "A new approach to linear filtering and prediction problems". *Journal of Basic Engineering, ASME,* **82**, pp. 35–45.

[28] Athans, M., Wishner, R., and Bertolini, A., 1968. "Suboptimal state estimation for continuous-time nonlinear systems from discrete noisy measurements". *Automatic Control, IEEE Transactions on,* **13**(5), Oct, pp. 504–514.

[29] Fricke, G. K., Milutinović, D., and Garg, D. P., 2009. Robotic pose estimation via an adaptive Kalman filter using state-varying noise. Submitted to *IASTED Robotics and Applications Conference*, Cambridge, MA, Nov 4-6, 2009.

[30] Kuhn, H. W., 1955. "The Hungarian method for the assignment problem". *Naval Research Logistics Quarterly,* **2**(1-2), pp. 83–97.

[31] Munkres, J., 1957. "Algorithms for the assignment and transportation problems". *Journal of the Society for Industrial and Applied Mathematics,* **5**(1), pp. 32–38.

[32] Y. Bar-Shalom, T. E. F., 1988. *Tracking and data association*. Academic Press Professional, Inc., San Diego, CA, USA.

[33] Braitenberg, V., 1984. *Vehicles: Experiments in Synthetic Psychology*. MIT Press, Cambridge, Massachusetts.