2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Miniature Six-channel Range and Bearing System: Algorithm, Analysis and Experimental Validation

Nicholas Farrow, John Klingner, Dustin Reishus and Nikolaus Correll

***Abstract*— We present an algorithm, analysis, and implementation of a six-channel range and bearing system for swarm robot systems with sizes in the order of centimeters. The proposed approach relies on a custom sensor and receiver model, and collection of intensity signals from all possible sensor/emitter pairs. This allows us to improve range calculation by accounting for orientation-dependent variations in the transmitted intensity, as well as to determine the orientation of the emitting robot. We show how the algorithm and analysis generalize to other range and bearing systems, and evaluate its performance experimentally using two ping-pong ball-sized "Droplets" mounted on a precise gantry system.**

I. Introduction

Range and bearing (R&B) detection is a critical operation in multi-robot systems and swarm robotics aggregation [1], dispersion [2], pattern recognition [3], flocking [4], and navigation [5], [6]. There exist multiple systems based on infrared (IR) [7], ultra-sound/radio [8] and ultra-wide-band radio [9] for medium-scale multi-robot systems, but precise R&B on centimeter-scale robots such as Alice [10], Jasmine [11] or Kilobot [12] remains a challenge due to the size of emitter, receiver and signal processing hardware. Robots at this size scale therefore either focus on range-only sensors [12], or implement only crude bearing measurements [10], [11]. Although range and bearing information can be derived from each other [13], [14], these methods are not practical for dynamic applications.

Range and bearing requires measuring both the intensity and the identity of other robots' emitters. This can be achieved by modulating the intensity signal with data packages, which usually requires dedicated signal processing hardware to measure the intensity of the carrier wave before decoding information [7], [15], [16], or during channel allocation, i.e., coordination among robots to prevent simultaneous transmissions, as in this paper. We propose a novel approach to range and bearing, as well as a theoretical framework that is general to the class of circular robots with signal emitters/sensors arranged around the robots' perimeter. This class includes popular robots like Khepera III and E-puck. In our platform, known as Droplets, each robot is circular with a 4.4 cm diameter and has six IR emitters and receivers evenly distributed on the perimeter. This is fewer than used on larger platforms, which use up to 12 channels [2], [7], [16]–[18]. As a result, we have less information available and techniques like interpolating between sensors [16] are less reliable. Previous work also treats the emitting robot as a point source of light located at the center of the robot, which ignores variations in received intensity caused by the orientation of the transmitting robot. The error introduced by this assumption is reduced as the number of emitters increases, and as the distance from the transmitting robot increases. For near-distance applications involving smaller robots, a different approach is needed.

Department of Computer Science, University of Colorado at Boulder, Boulder, CO 80309, firstname.lastname@colorado.edu

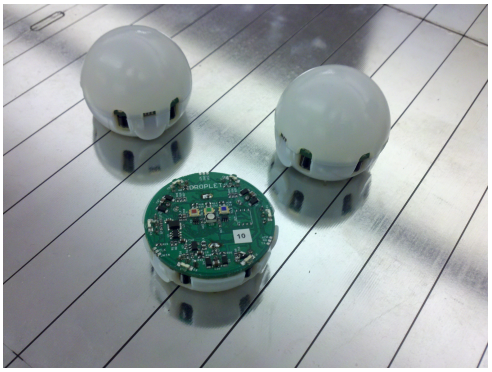

Fig. 1: The Droplet swarm robotics platform. Three Droplets are shown, one with the cover removed. The background is a floor of alternating power and ground strips from which the Droplets draw power (for scale the strip width is 22.8 mm).

Our range and bearing algorithm operates by taking measurements for each distinct sensor-emitter pair. For a platform with six emitters and sensors, this results in 36 pairwise intensities. This allows us to not only extract information about the transmitting robot's orientation relative to the receiving robot, but also allows to further improve accuracy by taking orientation-dependent variations in the transmitted intensity into account.

Our algorithm generalizes to other platforms by abstracting the equations that vary from platform to platform from the algorithm itself. Existing work in this area tends to be hardware specific. While the large number of discrete measurements required by our method increases the time spent collecting data, it is also direction neutral: with a single transmission, all robots in range can get

range and bearing data to the transmitting robot.

II. Mathematical Model for Range and Bearing Estimation

This model assumes a collection of identical, cylindrically symmetric robots with radius ρ. Each robot is equipped with n infrared emitters and n infrared sensors evenly distributed around the perimeter of the robot. Each robot has a well-defined forward direction. Without loss of generality, we will assume that each robot has its own local coordinate system with origin at its center, and x-axis aligned with the forward direction, so that forward is identified as 0 radians for bearing. Given an arbitrary pair of robots from this collection, we denote the robot using its sensors and running this algorithm as *RX*, and the robot using its emitters as *TX*.

We propose an algorithm that simultaneously computes: the distance from *RX* to *TX* (denoted R); the direction *TX* lies in, or bearing, with respect to *RX* (denoted θ); and the relative orientation, or heading, of *TX* to *RX* (denoted ϕ). Throughout this paper, range will mean the center to center distance between robots, and all angles are expressed in radians. Given some vector $\boldsymbol{X}$ (denoted with a bold symbol), the scalar X is defined as $\|\boldsymbol{X}\|$.

In order for a range and bearing measurement to be successful, the two robots (*RX* and *TX*) must coordinate a measurement event using a communication channel. This coordination is initiated by the robot *RX*, who requests R&B information from a specific robot *TX*. During this time, other robots within communication range remain silent and *TX* turns on its emitters in a predefined sequence. *RX* then takes the appropriate measurements based on the predetermined duration and sequence of *TX*'s emissions. Given that the two robots are asynchronous this coordination is necessary. Since the clock speed variations between robots is small compared to the timing of the range and bearing process, coordination is only needed at the beginning of a measurement cycle.

The sensors' sensitivity depends on the incoming light's angle of incidence, α. The sensor model $S(\alpha)$ describes this dependence. Similarly, the intensity of emitted light is dependent on the emitted light's deviation from 'straight on', β, and this dependence is described by the emitter model $E(\beta)$. So $\alpha = 0$ indicates that the emitter is directly in front of the sensor, and $\beta = 0$ indicates that the emitter is pointing directly at the sensor.

$S(\alpha)$ and $E(\beta)$ will necessarily be hardware dependent and are typically taken from the component datasheet. The point-source approximation that is commonly used, for example, is equivalent to the constant emitter model $E(\theta) = 1$. We fully characterize the sensor and emitter models for the Droplet platform in Section IV.

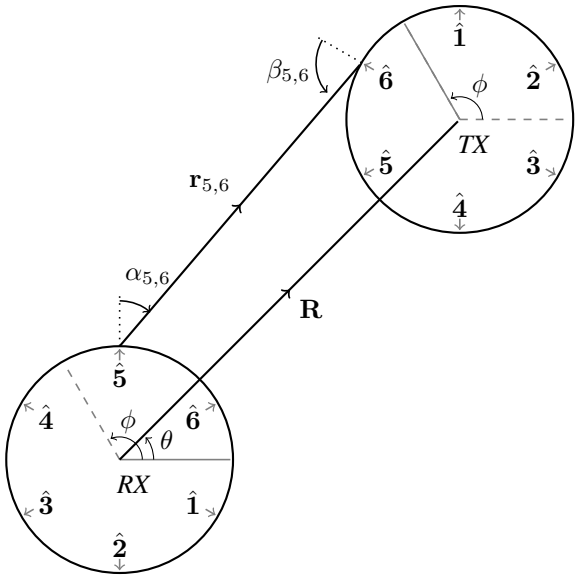

Fig. 2: Figure of two robots illustrating our terms. See Table I for symbol meanings. Note the vector $\mathbf{r}_{i,j}$, connecting sensor i with emitter j, and $\mathbf{R}$, connecting the center of the two robots.

Symbol	Meaning
RX	Sensing robot
TX	Transmitting robot
ρ	Robot radius
n	Number of sensors/emitters on each robot
θ	Angle from *RX* to *TX*, in *RX*'s coordinate frame
ϕ	Relative orientation of *RX* and *TX*
$\hat{\boldsymbol{\imath}}$	Vector to sensor/emitter i
S	Sensor model
E	Emitter model
A	Amplitude model
$\boldsymbol{R}$	Vector from *RX* to *TX*
$\boldsymbol{r}_{i,j}$	Vector from sensor i on *RX* to emitter j on *TX*
$\alpha_{i,j}$	Angle from sensor i's direction on *RX* to emitter j on *TX*
$\beta_{i,j}$	Angle from emitter j's direction on *TX* to sensor i on *RX*
$\lambda_{i,j}$	Light intensity received by sensor i on *RX* that was emitted by emitter j on *TX*
Λ	$n \times n$ matrix of $\lambda_{i,j}$
$\boldsymbol{\mathcal{S}}$	Vector of total light received by each sensor on *RX*
$\boldsymbol{\mathcal{T}}$	Vector of total light transmitted by each emitter on *TX* that was received by some sensor on *RX*

TABLE I

Next, we consider the light attenuation with distance. We call this the amplitude model $A(r)$ with r the distance between the sensor and the light source. We assume that amplitude fall-off is independent of the angle that the light exited the emitter, e.g., that the light is not focused or collimated by the emitter. This assumption permits us to treat the amplitude model as a function of r alone. We also assume that $A(r)$ is monotonic and invertible over the interval $(0, \infty)$. We fully characterize the amplitude model for our platform in Section IV.

We index the sensors by i and emitters by j: $i, j \in \{1, 2, ..., n\}$. The vector $\boldsymbol{r}_{i,j}$ connects sensor i to emitter j, and $\alpha_{i,j}$ is the angle that light enters sensor i from emitter j. Let $\widehat{\boldsymbol{i}}$ be a unit length vector from the center of the *RX* robot to its i^{th} sensor. Similarly, let $\widehat{\boldsymbol{j}}$ be a unit length vector from the center of the *TX* robot to its j^{th} emitter. Geometrically, the vector equation

$$\mathbf{r}_{i,j} = \mathbf{R} - \rho\hat{\mathbf{i}} + \rho\hat{\mathbf{j}} \tag{1}$$

relates the vector $\mathbf{r}_{i,j}$ between the sensors to the global vector linking the center of the two robots along the axis of transmission $\mathbf{R}$. The cosine dot-product identity gives us the following relationship between $\boldsymbol{r}_{i,j}$, $\alpha_{i,j}$, $\beta_{i,j}$.

$$\alpha_{i,j} = \cos^{-1}\left(\widehat{\boldsymbol{i}} \cdot \frac{\boldsymbol{r}_{i,j}}{\|\boldsymbol{r}_{i,j}\|}\right) \tag{2}$$

$$\beta_{i,j} = \cos^{-1}\left(\widehat{\boldsymbol{j}} \cdot \frac{-\boldsymbol{r}_{i,j}}{\|\boldsymbol{r}_{i,j}\|}\right) \tag{3}$$

The intensity that sensor i on *RX* perceives from emitter j on *TX* is denoted $\lambda_{i,j}$. Note that $\lambda_{i,j}$ depends entirely on the spatial configuration of the robots.

$$\lambda_{i,j} = S(\alpha_{i,j})E(\beta_{i,j})A(r_{i,j}) \tag{4}$$

The $\lambda_{i,j}$ values are stored in an $n \times n$ matrix Λ, which we call a 'brightness matrix'. From here, *RX* now has enough information to compute $\overline{R}$, $\overline{\theta}$, and $\overline{\phi}$, as we demonstrate. $\boldsymbol{\mathcal{S}}$ is a vector of length n whose i^{th} component is the total light received by sensor i from all emitters on *TX*, and $\boldsymbol{\mathcal{T}}$ is a vector of length n whose j^{th} component is the total amount of light received from emitter j by all sensors on *RX*.

$$\boldsymbol{\mathcal{S}} = \langle \mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_n \rangle \quad \text{s.t.} \quad \mathcal{S}_i = \sum_{j=1}^{n} \lambda_{i,j} \tag{5}$$

$$\boldsymbol{\mathcal{T}} = \langle \mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_n \rangle \quad \text{s.t.} \quad \mathcal{T}_j = \sum_{i=1}^{n} \lambda_{i,j} \tag{6}$$

Note that $\boldsymbol{\mathcal{S}}$ is the result of summing the rows of the brightness matrix Λ, and $\boldsymbol{\mathcal{T}}$ is the result of summing the columns of Λ (or, equivalently, summing the rows of Λ^T).

Let $\boldsymbol{\sigma} = \langle \widehat{\mathbf{1}}, \widehat{\mathbf{2}}, \ldots, \widehat{\boldsymbol{n}} \rangle$ be the vector of all $\widehat{\boldsymbol{i}}$. By taking the sum over the $\widehat{\boldsymbol{i}}$ vectors weighted by the sensor readings $\mathcal{S}_i$, we obtain a vector $\boldsymbol{\Theta}$ whose direction points approximately in the direction towards *TX*, $\overline{\theta}$. Mathematically,

$$\boldsymbol{\Theta} = \boldsymbol{\mathcal{S}} \cdot \boldsymbol{\sigma} = \sum_{i=1}^{n} \mathcal{S}_i \, \widehat{\boldsymbol{i}} \tag{7}$$

$$\overline{\theta} = \arg(\boldsymbol{\Theta}) \tag{8}$$

By measuring $\lambda_{i,j}$ pairwise we can immediately extend this formalism to the TX robot to find the orientation ϕ of TX in the coordinate frame of RX. By taking the sum over the $\widehat{\boldsymbol{j}}$ vectors weighted by the $\mathcal{T}_j$ sensor readings, we obtain a vector $\boldsymbol{\Phi}$ pointing in the approximate direction of *RX* from the coordinate frame of *TX*. By using a coordinate transform back to the coordinate frame of *RX*, we obtain ϕ. Using $\overline{\theta}$ as an estimate of θ yields an estimate $\overline{\phi}$ of the relative angle of *RX* and *TX*, using only local information available to the robot *RX* through its sensors. Mathematically,

$$\boldsymbol{\Phi} = \boldsymbol{\mathcal{T}} \cdot \boldsymbol{\sigma} = \sum_{j=1}^{n} \mathcal{T}_j \, \widehat{\boldsymbol{j}} \tag{9}$$

$$\overline{\phi} = \arg(\boldsymbol{\Phi}) + \overline{\theta} + \pi \tag{10}$$

We now have an axis connecting the robots (axis of transmission) and have constrained the system from three variables (θ, ϕ, and R) to one: R, the distance between *RX* and *TX*. To obtain an estimate of R, it would suffice to invert Equation 4, solving for $r_{i,j}$ for some sensor i and emitter j and use the estimates $\overline{\theta}$ and $\overline{\phi}$ to compute $\overline{R}$. However, inverting Equation 4 requires knowledge of $\boldsymbol{R}$ since both $\alpha_{i,j}$ and $\beta_{i,j}$ depend on $\boldsymbol{R}$ for each i and j. This relationship is:

$$\alpha_{i,j} = \tan^{-1}\left(\rho(\widehat{\boldsymbol{j}} - \widehat{\boldsymbol{i}}) + \overline{R}\right) - \arg(\widehat{\boldsymbol{i}}) \tag{11}$$

$$\beta_{i,j} = \tan^{-1}\left(\rho(\widehat{\boldsymbol{i}} - \widehat{\boldsymbol{j}}) - \overline{R}\right) - \arg(\widehat{\boldsymbol{j}}) - \arg(\overline{R}) \tag{12}$$

To avoid this circular logic, we make the initial assumption that $R \to \infty$ in order to approximate $\alpha_{i,j}$ and $\beta_{i,j}$. Under this initial approximation, $\boldsymbol{r}_{i,j}$ is parallel with $\boldsymbol{R}$ for all i and j, and hence $\alpha_{i,j}$ depends only on θ and $\beta_{i,j}$ depends only on θ and ϕ. Specifically, for every sensor-emitter pair i, j we initially estimate the values $\alpha_{i,j}$ and $\beta_{i,j}$ with

$$\alpha_{i,j} = \overline{\theta} - \arg(\widehat{\boldsymbol{i}}) \tag{13}$$

$$\beta_{i,j} = \overline{\theta} - \arg(\widehat{\boldsymbol{j}}) - \overline{\phi} - \pi \tag{14}$$

Since only one estimate $\overline{r}_{i,j}$ is needed to improve the estimate R', it suffices to choose one i, j pair. While several different pairs may work, we find the most reliable results are obtained from choosing the i and j from the maximal $\lambda_{i,j}$. An initial estimate $r'_{i,j} + 2\rho$ then provides an upper bound for $\overline{R}$.

$$\overline{r}_{i,j} = A^{-1}\left(\frac{\lambda_{i,j}}{S(\alpha_{i,j})E(\beta_{i,j})}\right) \tag{15}$$

$$R \leq \overline{R} = r'_{i,j} + 2\rho \tag{16}$$

Now, having an initial estimate $\overline{R}$, we can drop the assumption that $R \to \infty$ and use the R' value instead. This allows for more accurate estimates of $\alpha_{i,j}$, $\beta_{i,j}$ using Equations 11 and 12. Using these, we can calculate $\boldsymbol{r}_{i,j}$ for each (i, j) pair:

$$r_{i,j} = A^{-1}\left(\frac{\lambda_{i,j}}{S(\alpha_{i,j})E(\beta_{i,j})}\right) \tag{17}$$

Finally, recalling Equation 1 we can use the improved estimates of $\alpha_{i,j}$, $\beta_{i,j}$, and $r_{i,j}$ to calculate an improved estimate of the range $\overline{R}$ for each (i, j) pair:

$$\overline{\boldsymbol{R}} = \overline{\boldsymbol{r}}_{i,j} + \rho\widehat{\boldsymbol{i}} - \rho\widehat{\boldsymbol{j}} \tag{18}$$

A number of techniques could be employed to pick a single $\overline{R}$ from the set produced, but we expect them to vary based on the idiosyncracies of individual hardware and leave that implementation to a case-by-case basis. In our experience with the Droplet platform, choosing $\overline{R}$ from the maximal $\lambda_{i,j}$ works well in practice.

III. The Droplet Swarm Robotic Platform

The Droplets are an open-source swarm robotic platform, with source code and manufacturing information available online[1]. There are $n = 6$ infrared emitters and receivers located symmetrically around the circumference oriented horizontally. The Droplets use an Atmel Xmega128A3U microcontroller to interface emitters (Kingbright APA3010F3C-GX) and sensors (Kingbright APECVA3010P3BT IR phototransistor) and perform R&B signal processing. Current through the infrared LED is controlled using a digital potentiometer (Microchip Technology MCP4331-503E/ML). The output from the infrared sensors is read using the microcontroller's built-in 8-bit analog-to-digital converter. Digital communication between Droplets is accomplished using the same infrared emitters and the microcontroller's built-in serial communications peripherals (UARTs). Output from the UART is modulated on a 38 kHz carrier wave, which is demodulated with a dedicated remote receiver (Vishay TSOP39438). Note that range and bearing communication occurs using unmodulated DC signals.

IV. Model Characterization

In the state-of-the-practice experiments such as those presented in [16], [17], it is common to use cosine or its square to represent the sensor model, and it is common to assume inverse relationship with r, or exponential decay e^{-r} to represent the amplitude model. While these approximations work fairly well, they ignore the complexities of real, physical sensors.

We experimentally obtained to high accuracy the sensor model, emitter model, and amplitude model for the Droplet platform by using an automated system capable of repeatably positioning θ and ϕ by $0.9°$ increments and of adjusting R to an accuracy of 0.5 mm. The apparatus, shown in Figure 3, consists of two Droplets attached to stepper motors ($400\frac{\text{steps}}{\text{rev}}$). One of the stepper motors is connected to a Firgelli linear actuator with built-in encoder. The apparatus was controlled using custom

[1]https://code.google.com/p/cu-droplet/

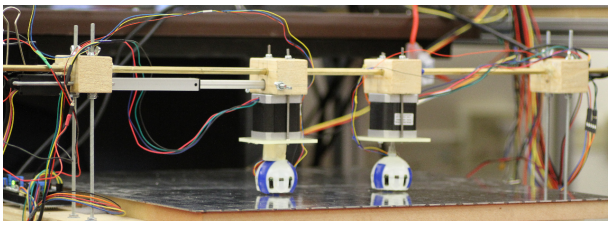

Fig. 3: The apparatus used to characterize the sensor, emitter, amplitude, and noise models of the mathematical system for use in simulations.

software on an Arduino module with motor shield and stepper motor drivers. We experimentally characterized the accuracy of this actuator and its controller in [19]. To find the sensor model $S(\alpha)$, we held ϕ fixed so that an emitter on *TX* was pointed directly at *RX*, held R fixed at 13 cm, 18 cm, 23 cm, or 28 cm, and varied θ from $-180°$ to $180°$ in $0.9°$ increments. To find the emitter model $E(\beta)$ a similar process was used, with the roles of the *RX* and *TX* Droplets interchanged, and the roles of ϕ and θ interchanged. The resulting models, obtained by fitting the experimental data (see Figure 4), are given below:

$$S(\alpha) = \begin{cases} -\alpha^4 + 1 & \text{if } |\alpha| < 0.6186 \\ \frac{1}{8\alpha^4} & \text{else} \end{cases} \tag{19}$$

$$E(\beta) = \begin{cases} -\beta^4 + \frac{\beta^2}{2} + 0.9375 & \text{if } |\beta| < 0.7205 \\ \frac{1}{4\beta^4} & \text{else} \end{cases} \tag{20}$$

To find the amplitude model $A(r)$, we held θ and ϕ fixed such that an emitter on *TX* was aligned with a sensor on *RX* and varied R from 8 cm to 30 cm in increments of 2 mm. This was repeated 36 times at each distance: once for each emitter on *TX* and sensor on *RX*. We found that the sensors became saturated for $R < 12$cm, and so cut off smaller data points before fitting. Our resulting Amplitude Model is

$$A(r) = \frac{14000}{(r-3)^2} - 1 \tag{21}$$

with corresponding inverse

$$A^{-1}(\lambda) = \frac{118}{\sqrt{\lambda + 1}} + 3 \tag{22}$$

For closer ranges, a different amplitude model was similarly computed, with the IR emitters on *TX* set to a lower power. Figure 4 shows the experimentally determined (a) sensor model, (b) emitter model, and (c) amplitude model, together with the data used to construct the models. We also used the data described above to determine the noise model for our hardware. We find that after calibration, the hardware error of our model is approximately Gaussian with $\mu = 0$.

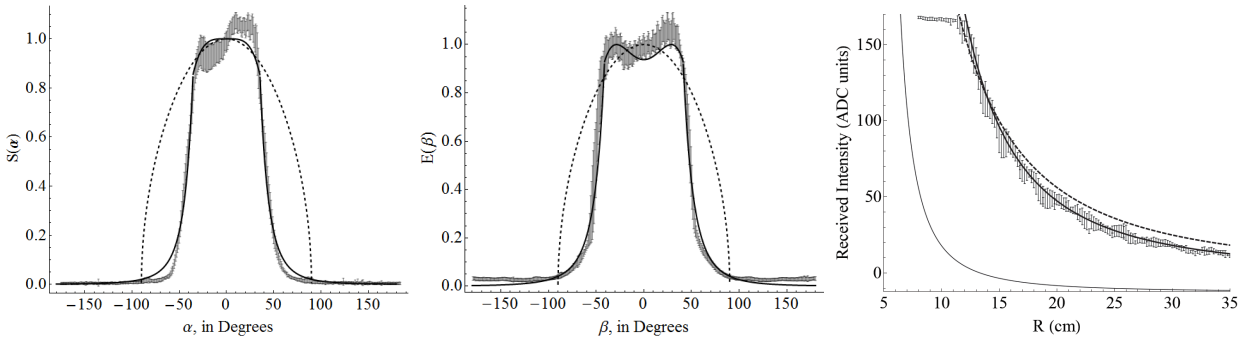

Fig. 4: Experimental derivations of the sensor model, emitter model and amplitude model. In all figures the solid line indicates our model (eqs. 19-21). Error bars indicate one standard deviation, centered on the mean. In (a) and (b) the dashed line is $\sqrt{\cos}$ which has been used as a stand-in in other works. In (c) we also show a thin solid line as the near-range low power amplitude model we use to cover the range where sensors saturate. Also, a dashed line is used to demonstrate a $\frac{1}{R^2}$ model as has been used as stand-ins in other works.

V. Experimental Analysis

To demonstrate our algorithm on real hardware, we used the same apparatus as for parameter characterization, pictured in Figure 3. With this apparatus, two Droplets were swept through a variety of orientations and distances, with the pair making a range and bearing measurement at each position. The results of these measurements are shown in Figures 5–7. We are able to compare these results to the ground truth given by the testing apparatus, which has negligible uncertainty compared to our measurements. In these experiments, each set of R&B measurements took approximately 500 ms; we have verified, however, that reliable R&B measurements can be accomplished in less than 100 ms on the Droplet platform, giving an update rate of 10 Hz.

The number of measurements represented by each position is approximate because the Droplet does not always successfully collect intensity data. This is due to occasional dropped packets in the communication subsystem—the *RX* robot needs to ask the *TX* robot to transmit, and if that request is unheard, range and bearing calculation is not performed. The *RX* robot is able to identify when this occurs by noticing that the largest intensity received is comparable to background noise. In the implementation used for these experiments, the Droplet will make another communication attempt if it detects that the previous attempt failed, repeating this for up to five attempts. After five failed attempts, it reports that a problem occurred and the test moves on to the next position. As this represents an error in the communication rather than the range and bearing, these events are not included in the data. In the rare event that communication fails but the *RX* Droplet does not notice due to particularly high background noise, a spurious data point is generated for that position. These data points are included in the data.

We observe a near constant range error at distances close to the robot, and the range error is always bound within 1 robot diameter. The error does not seem to change significantly with distance up to the 4.5 robot diameters that we checked. We were able to use our algorithm down to inter-robot distances of one robot radius, which is much smaller than other related works. Although the relative error is high for distances shorter than this, we hypothesize that this could be compensated by running the emitter/receivers in reflective distance sensor mode, which allows for millimeter accuracy in comparable robots [10], [11].

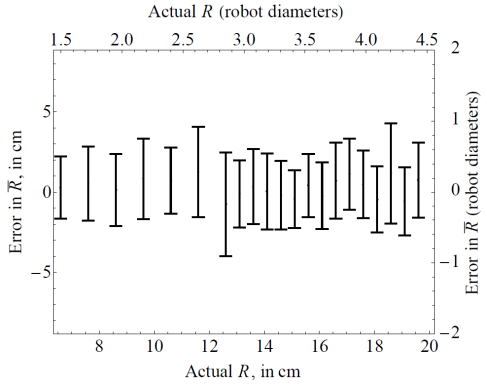

Fig. 5: Absolute error in range estimation with respect to the actual range. Error bars represent the standard deviation. At least 250 measurements were taken for each position.

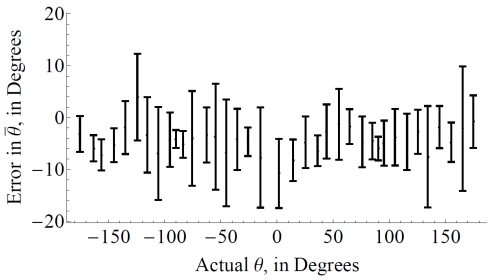

Fig. 6: Absolute error in bearing estimation with respect to the actual bearing. Error bars indicate one standard deviation. Around 150 measurements are represented at each position.

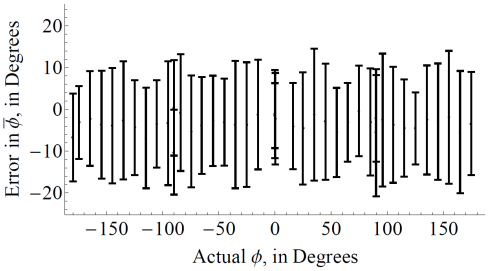

Fig. 7: Absolute error in heading estimation with respect to the actual heading. Error bars indicate one standard deviation. Around 125 measurements are represented at each position.

The error shown in Figure 6 for the bearing estimation seems to be uniformly below $0°$, however we attribute this to the asymmetric real sensor values shown in Figure 4(a). It is possible that the sensors may have some asymmetry in their sensitivity and our model does not account for this. The error shown in Figure 7 for the heading estimation is somewhat larger, however this calculation (Equation 9) also includes $\overline{\theta}$, therefore any error in $\overline{\theta}$ is contributed to the error in $\overline{\phi}$. This error exhibits periodicity consistent with the 6 sensor/emitter pairs on the Droplets, as the two robots rotate in and out of alignment with one another.

VI. Conclusions & Future Work

We present a range and bearing system for low-cost, miniature robotic platforms. We develop a mathematical model that allows us to take variations in the emitter angle into account. We then demonstrate a practical algorithmic implementation of this framework on the Droplet swarm robotics platform. Though computationally more complex than existing state-of-the-art algorithms, carefully modeling emitter and receiver characteristics should increase the robustness and accuracy for platforms with a small number of sensors and especially for small distances. Further, our approach is able to calculate the orientation of the transmitting robot directly.

Though our theoretical framework is able to capture the particular characteristics of sensor/emitter pairs, we do use the simplifying assumption that the sensor and emitter models are consistent across different sensors and emitters, which tends not to be the case. We plan in future work to capture the sensor variation, perhaps with an auto-calibration step on robot initialization.

Acknowledgments

This work has been supported by NSF awards #1153158, #1150223 and a CI fellowship to D. Reishus.

References

[1] N. Correll and A. Martinoli, "Modeling and designing self-organized aggregation in a swarm of miniature robots," *The International Journal of Robotics Research*, vol. 30, no. 5, pp. 615–626, 2011.

[2] J. McLurkin and J. Smith, "Distributed algorithms for dispersion in indoor environments using a swarm of autonomous mobile robots," *Distributed Autonomous Robotic Systems 6*, pp. 399–408, 2007.

[3] L. Liu, B. Fine, D. Shell, and A. Klappenecker, "Approximate characterization of multi-robot swarm "shapes" in sublinear-time," in *Robotics and Automation (ICRA), IEEE Int. Conf. on*, pp. 2886–2891, IEEE, 2011.

[4] I. Navarro and F. Matía, "Distributed orientation agreement in a group of robots," *Autonomous Robots*, pp. 1–21, 2012.

[5] F. Ducatelle, G. A. Di Caro, C. Pinciroli, F. Mondada, and L. M. Gambardella, "Communication assisted navigation in robotic swarms: self-organization and cooperation," in *Intelligent Robots and Systems (IROS), IEEE/RSJ Int. Conf. on*, pp. 4981–4988, IEEE, 2011.

[6] Á. Gutiérrez, A. Campo, F. Monasterio-Huelin, L. Magdalena, and M. Dorigo, "Collective decision-making based on social odometry," *Neural Computing and Applications*, vol. 19, no. 6, pp. 807–823, 2010.

[7] J. Pugh and A. Martinoli, "Relative localization and communication module for small-scale multi-robot systems," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pp. 188–193, IEEE, 2006.

[8] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," in *Proceedings of the 6th annual international conference on Mobile computing and networking*, pp. 32–43, ACM, 2000.

[9] A. Prorok, P. Tomé, and A. Martinoli, "Accommodation of nlos for ultra-wideband tdoa localization in single-and multi-robot systems," in *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on*, pp. 1–9, IEEE, 2011.

[10] G. Caprari and R. Siegwart, "Mobile micro-robots ready to use: Alice," in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pp. 3295–3300, IEEE, 2005.

[11] S. Kornienko, O. Kornienko, and P. Levi, "Minimalistic approach towards communication and perception in microrobotic swarms," in *Intelligent Robots and Systems (IROS), IEEE/RSJ Int. Conf. on*, pp. 2228–2234, IEEE, 2005.

[12] M. Rubenstein, C. Ahler, and R. Nagpal, "Kilobot: A low cost scalable robot system for collective behaviors," in *Robotics and Automation (ICRA), IEEE Int. Conf. on*, pp. 3293–3298, IEEE, 2012.

[13] R. Nagpal, H. Shrobe, and J. Bachrach, "Organizing a global coordinate system from local information on an ad hoc sensor network," in *Information Processing in Sensor Networks*, pp. 333–348, Springer, 2003.

[14] A. Cornejo, A. J. Lynch, E. Fudge, S. Bilstein, M. Khabbazian, and J. McLurkin, "Scale-free coordinates for multi-robot systems with bearing-only sensors," in *Algorithmic Foundations of Robotics X*, pp. 397–414, Springer, 2013.

[15] I. Kelly and A. Martinoli, "A scalable, on-board localisation and communication system for indoor multi-robot experiments," *Sensor Review*, vol. 24, no. 2, pp. 167–180, 2004.

[16] J. Pugh, X. Raemy, C. Favre, R. Falconi, and A. Martinoli, "A fast onboard relative positioning module for multirobot systems," *Mechatronics, IEEE/ASME Transactions on*, vol. 14, no. 2, pp. 151–162, 2009.

[17] A. Gutiérrez, A. Campo, M. Dorigo, J. Donate, F. Monasterio-Huelin, and L. Magdalena, "Open e-puck range & bearing miniaturized board for local communication in swarm robotics," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pp. 3111–3116, IEEE, 2009.

[18] S. Gowal, A. Prorok, and A. Martinoli, "Two-phase online calibration for infrared-based inter-robot positioning modules," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 3313–3319, IEEE, 2011.

[19] E. Komendera, D. Reishus, J. T. Dorsey, W. R. Doggett, and N. Correll, "Precise truss assembly using commodity parts and low precision welding," in *IEEE Int. Conf. on Technologies for Practical Robot Applications*, April 2013.