

Cooperative and Coordinated Localization of Swarm Robots using Adaptive Boids Rules

Trevor Smith, Eduardo Gutierrez,
Jonas Amoama Bredu, Yu Gu,
Jason Gross, *West Virginia University*

ABSTRACT

This study seeks to improve the localization estimate of swarm robotics through cooperation, agents sharing information, and coordination, multiple robots altering their movements to work together, by utilizing boids rules. Since swarm robots are often constructed of simple and low-quality sensors, due to the increase in cost with swarm size, complex perception tasks, such as localization, become difficult challenges to solve. However, robot swarms have the advantage of large numbers. Therefore, the robots can coordinate their movements, to perform more localization measurements with each other to improve their localization estimates. Boids rules model complex biological flocking behavior, that allows the swarm to coordinate their movements. Thus, by incorporating flocking behavior, into the localization strategy, the robots can group together and perform more localization updates at necessary. In addition, this study expands boids rules with the inclusion of homing and task rules, along with adapting the gains on each rule based on the local stimuli of neighboring robots and the current localization estimate. As a result, the robots are encouraged to perform more favorable emergent behaviors such as clustering when the agents' localization estimates are poor and diffusion behaviors when the estimate is sufficient. The proposed algorithm is decentralized and independent of the localization strategy, thus it can be easily incorporated into existing swarm robotic frameworks. This method was tested with multiple localization frameworks such as decentralized EKF, Covariance Intersection, and Dead Reckoning in a randomly generated way-point bounded environment. As a result of this study, at the cost of increasing task completion time, the proposed algorithm was able to improve the average localization estimate of the swarm, with the greatest improvement occurring at low-density swarms with poor estimators.

I. INTRODUCTION

Swarm robotics is an emerging field that provides the advantages of high adaptability and robustness, over conventional single-agent systems. Swarm robotics is highly adaptable since the robots only use local information, thus allowing them to adapt to the local conditions, without knowing global information. This makes swarm robotics suitable for missions in GPS-denied environments, such as underwater or extra-planetary, where this global information source is not available, and the system must rely on local information. Also, swarms are robust (having the ability to achieve partial completion of a task, in the presence of failures), due to the intrinsic inclusion of non-single-point failure. Therefore, if in a swarm of 100 robots, ten robots fail, the task could be partially completed. However, with single-agent robots, if one part of a robot fails, the entire system is compromised.

Robotic swarms are often utilized in highly parallelizable tasks, that are time-consuming for single-agent robots, such as building complex structures from multi-cellular organisms to flocks of birds as in Rubenstein et al. (2014). In addition, they have also been used in the field of foraging, where groups of robots can be programmed for exploration and finding the shortest path. Both of which are essential in search rescue applications as well as the collection of terrain sample missions, according to Gautam and Moh (2012) and Jevtić and Andina de la Fuente (2007).

Many of these applications require accurate localization to be completed, as mentioned in Wu et al. (2014). However, swarm robots are often made with cheaper lesser-quality sensors, when compared to conventional single-agent systems, due to the cost of a robotic swarm linearly increasing with the size of the swarm. Therefore, complex perception tasks that require high-quality sensors, such as localization, are difficult challenges to solve for robotic swarms.

There are two general classes of position estimation methods for localization: relative positioning and absolute positioning. Relative positioning is often based on inertial and odometry sensors in addition to robot-to-robot interactions such as range and bearing measurements. Inertial and odometry measurements are readily available for each robot in the swarm and integrated to determine the current position estimation. However, Cho et al. (2012) describes that these estimates are only accurate for short periods of time, due to the integration of noise and bias causing random walk and sensor drift. Absolute positioning is based on sensors that interact with environmental elements such as GPS satellites or known landmarks. These measurements do not drift with time but are not always readily available.

To limit the scope of swarm localization, each robot in the swarm is tasked with accurately navigating to randomly selected points, with no environmental confirmation of arrival. In addition, we assume the robots are in a GPS-denied 2D environment,

but have a deployment home base landmark, with a known location, for limited absolute positioning, similar to the environment specified in Gu et al. (2018). The robots also can perform relative positioning using wheel encoders and a gyroscope, while also performing range and bearing measurements between robots using a 2D LiDAR. Furthermore, all communication/detection is limited to a small detection range.

This environment presents multiple difficulties for localization since the robots have to navigate outside the absolute positioning range of the home base to reach the desired goals. This means that each robot's localization estimate will have to rely on relative positioning which drifts over time. In addition, at lower robot densities, the robot will experience infrequent robot-to-robot measurement updates, due to the limited detection range. Consequently, each robot has a limited navigation range before its localization estimate becomes too poor to navigate and it must return home for an absolute positioning update. Thus, if a goal is placed outside the navigation range the robot will be unable to complete the task.

This study addresses the gap in active swarm localization of using coordination, multiple robots altering their movements to work together, and cooperation, sharing of information between robots, to improve the localization estimate of the swarm by performing more robot-to-robot measurement updates whenever needed. Therefore, increasing the navigation range of the swarm and allowing more tasks to be completed. The proposed overall framework is shown in Figure 1 and our main contributions are:

- A novel method to increase robot-to-robot measurements by utilizing boids rules, which model the flocking behavior of swarming organisms.
- Adapting boids rules to local stimuli, to induce a low-level decision-making scheme, allowing the robots to decide how to improve their localization estimate, while completing their task.

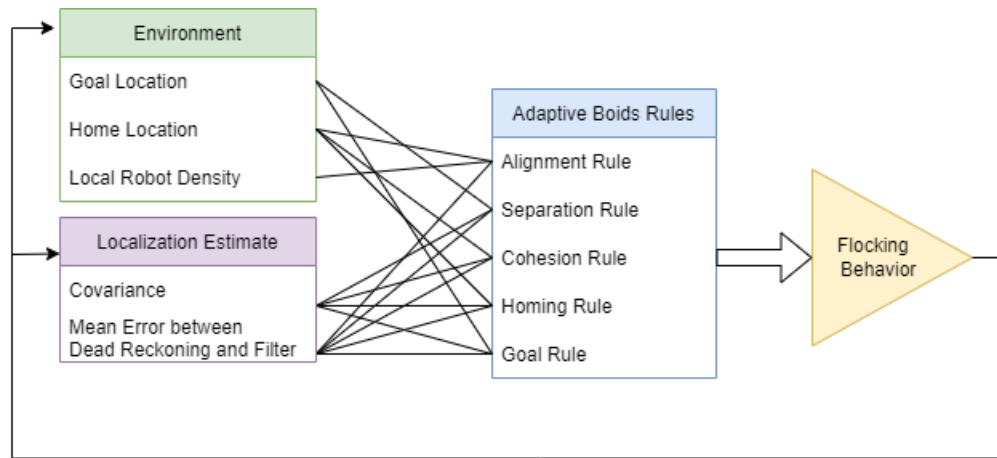


Figure 1: The local environmental information and localization estimate are utilized to adapt the flocking behavior of the swarm to improve the average localization estimate, while also progressing towards the goal.

The rest of this paper is organized as follows. Section II describes the previous work that this study improves upon. Section III details and explains the implementation of the proposed algorithm. Section IV provides insights into the results of simulation experiments. Finally, Section V provides contributions and insights for future works to improve the system.

II. LITERATURE REVIEW

The problem of swarm localization requires knowledge of the current localization techniques and swarm behavior modeling. Since the robots must be able to fuse the sensor information from themselves and other robots, while also cooperating to navigate to goals outside their individual navigation range.

1. Localization Techniques

Cooperative localization can be obtained using decentralized or centralized filter estimators. These filter estimators are used to fuse the information from the agents to achieve a better state estimation. Centralized versions often outperform Decentralized versions according to Luft et al. (2016). However, Bailey et al. (2011) describes the main disadvantages of centralized versions are the exponentially increasing computational cost per agent and continuous high data rate communication needs. Therefore it is only often used in small robotics teams of 2-3 as shown in Gross et al. (2019) and Hardy et al. (2016). In addition, there is a

need for an agent or fusion center to receive the information from all the agents to make the computations. On the other hand, in decentralized methods, each agent has its own filter to make its estimations. This decreases the computational requirements for each agent and eliminates the need for continuous high speed communication, but comes at the cost of degraded performance.

Due to the nature of swarm robotics, decentralized localization techniques are more favorable than centralized techniques for multiple reasons. The first is the computational requirements quickly become intractable due to the large number of agents. Also, in low-density swarms, it is not guaranteed that every robot will be within communication range of the fusion center and thus preventing an update. In addition, if the central agent fails the rest of the swarm fails, thus creating a single-point-of-failure. Therefore, two decentralized localization methods are used in this study: Covariance Intersection (CI) and Decentralized Extended Kalman Filter (DEKF). In addition, Dead Reckoning (DR) is also utilized since it does not require a centralized filter.

Dead Reckoning evaluates the information from proprioceptive sensors such as wheel encoders or Inertial Measurement Units (IMU) to evaluate the robots positioning with respect to its movement and initial position, as described by Bong-Su et al. (2011). The main benefit that DR offers is simplicity and that it doesn't require any external information. However, Bong-Su et al. (2011) also states that it usually produces estimation drift over time, which makes it unreliable for long distances.

Covariance Intersection, as detailed in Carrillo-Arce et al. (2013), Franken and Hupper (2005), and Yang et al. (2021), fuses the position estimates of each robot i from each other robot j 's perspective, in the local neighborhood consisting of L robots. This is done using Equations 1 and 2, via a convex combination of the inverse of the covariance matrices (i.e. the information matrices).

$$\hat{x}_i = \sum_{j=1}^L P_i^{-1} (\alpha_j [P_i^j]^{-1} \hat{x}_i^j) \quad (1)$$

where

$$P_i = \left[\sum_{j=1}^L \alpha_j [P_i^j]^{-1} \right]^{-1} \quad (2)$$

Where, $\hat{x}_i$ and P_i are the position estimate and covariance matrix, respectively, of the robot the neighborhood is centered around. $\hat{x}_i^j$ and P_i^j are the position estimate and covariance matrix, respectively of the neighborhood centered robot's j^{th} neighbor, and α is the set of weighted coefficients that sum to one and are all positive. Finding the set of weighting coefficients α that minimize the determinate or trace of the covariance matrix, is a nonlinear optimization problem, and thus computationally difficult. However, Franken and Hupper (2005) and Yang et al. (2021) present a fast CI method that solves for α using Equations 3 and 4. Where S is the sum of the information matrices of robot i from each neighbor q .

$$\alpha_j = \frac{\det(S) - \det(S - P_i^j) + \det(P_i^j)}{L \det(S) + \sum_{q=1}^L [\det(P_i^q) - \det(S - P_i^q)]} \quad (3)$$

$$S = \sum_{q=1}^L [P_i^q]^{-1} \quad (4)$$

Lastly, the Decentralized EKF is the most complex method. During its mission, each robot is capable of performing three different algorithms to estimate its position depending on the type of measurements it has available as described in Luft et al. (2016, 2018). In the first algorithm, the motion update, each robot has access to a velocity command or odometry information to predict its next location. The state for each robot along with its covariance is predicted with Equations 5 and 6. Where X_i is the state for robot i , f is the non-linear prediction model, u is the input in the model, F is the Jacobian of the non-linear prediction model and Q is the noise in the prediction model. Moreover, the cross-correlation with the rest of the robots in the swarm is also calculated with Equation 7. Where p_{ij} represents the cross-correlation with all the rest of the neighbors j for robot i .

$$\hat{X}_i^{k+1} = f(\hat{x}_i^k, u) \quad (5)$$

$$P_{ii}^{k+1} = F P_{ii}^k F^T + Q \quad (6)$$

$$p_{ij}^{k+1} = F p_{ij}^k \quad (7)$$

The second algorithm, the private update, as named in Luft et al. (2016), is performed individually by each robot whenever they are able to receive measurements from different landmarks or receive GPS measurements that are not shared with the group. This algorithm might or might not be performed by all the robots since it depends on the individual relations within the environment. The update of the state, covariance and cross-correlation terms is done using Equations 8, 9, 10 and 11. Where K is the Kalman gain, H is the Jacobian of the measurement model, R is the noise in the measurement, z is the measurement, h is the non-linear measurement model and I represents the identity matrix.

$$K_i = P_{ii}^k H^T (H P_{ii}^k H^T + R)^{-1} \quad (8)$$

$$\hat{X}_i^{k+1} = \hat{X}_i^k + K_i [z - h(\hat{X}_i^k)] \quad (9)$$

$$P_{ii}^{k+1} = (I - K_i H) P_{ii}^k \quad (10)$$

$$p_{ij}^{k+1} = (I - K_i H) p_{ij}^k \quad (11)$$

Lastly, the third algorithm Luft et al. (2016) describes, is the relative update, which is performed when two robots can share their states and covariances with pairwise communication and can obtain a ranging/bearing measurement from each other. In this algorithm, one robot (i) couples its states, covariance and cross-correlated values with the ones received from the other robot (j) using Equations 12, 13 and 14. The update of the state and the covariance of the different robots in this algorithms is then performed using Equations 15, 16, and 17, which are similar to the ones used in the previous algorithm. After the update, the states and covariances are decoupled with Equations 18, 19 and 20 and sent to their respective robot. Where X_a represents the coupled states, P_a represents the coupled covariances, K_a is the Kalman gain, G is the Jacobian of the measurement model, R is the noise in the measurement, r is the measurement, g is the non-linear measurement model and I represents the identity matrix and p_{ih} represents the cross-correlation terms with the rest of the robots not present in the relative update.

$$X_a^k = \begin{pmatrix} \hat{X}_i^k \\ \hat{X}_j^k \end{pmatrix} \quad (12)$$

$$P_{ij}^k = p_{ij}^k (p_{ji}^k)^T \quad (13)$$

$$P_{aa} = \begin{pmatrix} P_{ii}^k & P_{ij}^k \\ P_{ji}^k & P_{jj}^k \end{pmatrix} \quad (14)$$

$$K_a = P_a^k G^T (G P_a^k G^T + R)^{-1} \quad (15)$$

$$\hat{X}_a^{k+1} = \hat{X}_a^k + K_a [r - g(\hat{X}_i^k, \hat{X}_j^k)] \quad (16)$$

$$P_{aa}^{k+1} = (I - K_a G) P_{aa}^k \quad (17)$$

$$p_{ij}^{k+1} = P_{ij}^{k+1} \quad (18)$$

$$p_{ji}^{k+1} = I \quad (19)$$

$$p_{ih}^{k+1} = P_{ii}^{k+1} (P_{ii}^k)^{-1} p_{ih}^k \quad (20)$$

In this study, each of these three methods will be utilized to display the benefits of the proposed algorithm independent of the localization technique utilized.

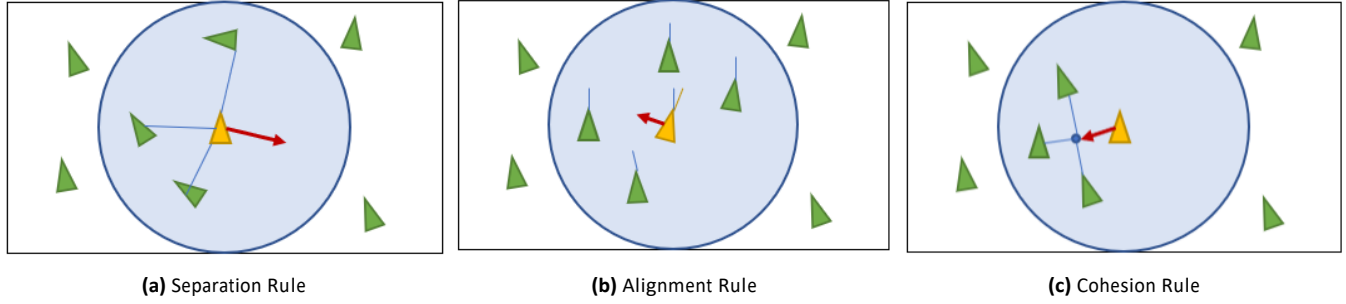


Figure 2: Three rules of the original boids model. Figure based on Alaliyat et al. (2014)

2. Boids Rules

In 1987, Craig Reynolds developed the first computer animation of flocking and proposed three behavior rules for displaying successful flocking: alignment, cohesion and separation. Separation describes the desire of the robots to avoid collision, cohesion the gathering of the robots and alignment the matching and coordinating of one robot's moves with others, as described in Reynolds (1987). These three rules are enacted using simulated steering vectors that are linearly summed to generate a resultant acceleration on the agent. In addition, the magnitude of each of the forces is defined by a gain, K_a , K_c , K_s for alignment, cohesion, and separation, respectively, while the direction is specific for each rule. For alignment, the direction is defined to minimize the heading error between the agent and the average heading of the local neighbors. Cohesion's direction points from the robot to the centroid of the neighboring robots. While separation points opposite the resultant unit vector to all other robots. Figure 2 and Equations 21 - 23 display these three rules, while Equation 24 displays the resultant acceleration of the agent (A_i). Where F_{a_i} , F_{c_i} , F_{s_i} are the steering vectors for alignment, cohesion, and separation, respectively, of agent i . P_i and V_i are the estimated position and velocity vectors of agent i , and L is the number of robots in the local neighborhood.

$$F_{a_i} = K_a \frac{\left(\sum_{j=1}^L \frac{V_j}{L} \right) - V_i}{\left\| \sum_{j=1}^L \left(\frac{V_j}{L} \right) - V_i \right\|} \quad (21)$$

$$F_{c_i} = K_c \frac{\left(\sum_{j=1}^L \frac{P_j}{L} \right) - P_i}{\left\| \left(\sum_{j=1}^L \frac{P_j}{L} \right) - P_i \right\|} \quad (22)$$

$$F_{s_i} = K_s \frac{\sum_{j=1}^L (P_i - P_j)}{\left\| \sum_{j=1}^L (P_i - P_j) \right\|} \quad (23)$$

$$A_i = F_{a_i} + F_{c_i} + F_{s_i} \quad (24)$$

These three simple rules that emerge into highly organic flocking behavior have lead other researchers to utilize and modify the rules for a variety of scenarios. Hartman and Benes (2006), for instance, included an additional rule for leader changing as a complement to the alignment rule, similar to how separation and cohesion are complements. This rule was defined as a probability of steering away from the flock if it is on the edge of the group. This made the model more accurate to live birds and prevented the regular visual patterns, that were seen with the original model.

In addition, since swarming organisms are typically prey, or primary consumers, Delgado-Mata et al. (2007) also added a rule to the three original rules, called "Escape". This rule implements fear of danger, such as predators, into the model, therefore it points opposite the direction of the fear source. Delgado-Mata et al. (2007) also utilizes a locally measured fear stimuli to modify the gains on the new escape vector and the original cohesive vector, causing "avoid danger" and "stay with the flock" behaviors to emerge, respectively. The addition of this rule, introduced a global goal aspect to the model, survive, and can be seen as a step away from recreating a flocking pattern and a step towards a simple robot, that is tasked with surviving.

The proposed algorithm seeks to utilize the Reynolds (1987) original boids model to cause the swarm robots to flock together to perform an increased number localization measurement updates between agents. While also utilizing Hartman and Benes (2006) and Delgado-Mata et al. (2007)'s work by adding additional rules for a homing, to provide a absolute measurement of a known location, and a task rule, to have the robots achieve their objectives.

III. METHODOLOGY

Building upon this previous work in swarm behavior and localization techniques, the original boids model is augmented with the inclusion of homing and task rules, to promote the agents to seek out a base landmark with a known location, and achieve their assigned task, respectively. The homing rule points directly from the agent to the known home location P_h , while the task rule points from the robot to the specified location of the goal P_g . Both of these new rules' magnitude is specified by their gains, K and K_h for home and goal rules, respectively. Furthermore, these rules are displayed in Equations 25 and 26, with Equation 27 displaying the augmentation of Equation 24 to include the two new rules.

$$\vec{P}_{h_i} = K_h \frac{\vec{P}_h - \vec{P}_i}{||\vec{P}_h - \vec{P}_i||} \quad (25)$$

$$\vec{P}_{g_i} = K_g \frac{\vec{P}_g - \vec{P}_i}{||\vec{P}_g - \vec{P}_i||} \quad (26)$$

$$\vec{A}_i = \vec{P}_{a_i} + \vec{P}_{c_i} + \vec{P}_{s_i} + \vec{P}_{h_i} + \vec{P}_{g_i} \quad (27)$$

In addition to adding these two new rules, this work also adapts the gains of each rule based on the local stimuli of the neighboring robots and current localization estimates. This in turn creates a low-level decision framework to seek localization improvement as necessary. A simulation environment was developed to demonstrate the effectiveness of the proposed adaptive boids rules against, goal following, which only utilizes the goal rule, and Monte Carlo optimized static gains, for non-coordinated and non-adaptive, respectively. In addition to being used with the three localization estimators: Dead Reckoning (non-cooperative), Covariance Intersection, and Decentralized Extended Kalman Filter.

1. Adaptive Boids Rules

To accommodate for the dynamic environmental and sensory conditions each of the gains for each boids Rule: alignment (K), cohesion (K_c), separation (K_s), home (K_h), and goal (K_g), were adapted based on the local stimuli observed by each robot. Five local stimuli were utilized to modify the gains of the boids rules: local robot density (ρ) within the robot's circular detection range d , spectral norm of the estimated covariance matrix ($||\sigma||$), mean error between the individual robot's DR position and position from the estimator used (e), the distance to the robot's goal (d_g), and the distance the robot is from its home location (d_h). Each of the stimuli were then normalized, local robot density, mean error, and covariance were normalized by the maximum allowable value, ρ_{max} , e_{max} , σ_{max} . While the distance to home and goal were normalized by the length of the world, L_w . Equations 28-32 display the normalization of the local stimuli, where R is total number of deployed robots.

$$\rho_{norm} = \frac{\rho}{\rho_{max}} = \frac{\frac{L}{\pi d^2}}{\frac{R}{\pi d^2}} = \frac{L}{R} \quad (28)$$

$$||\sigma_{norm}|| = \frac{||\sigma||}{||\sigma_{max}||} \quad (29)$$

$$e_{norm} = \frac{e}{e_{max}} \quad (30)$$

$$d_{g_{norm}} = \frac{d_g}{L_w} \quad (31)$$

$$d_{h_{norm}} = \frac{d_h}{L_w} \quad (32)$$

Now that each local stimuli are normalized, their effects on the boids rule gains were specified as positive linear relationships or negatively linear relationships with a y-intercept of one, ensuring positive gains. These relationships were chosen for simplicity and to elicit behaviors of the swarm by inducing dominance of a specific boids rule.

Starting with the alignment gain (K), whose main focus in this study is robot avoidance, the three stimuli that affect this gain are local robot density, distance to home, and mean error between DR's position estimate and the filtered position estimate. The affect of local robot density on the alignment gain is positive, which leads to the swarm behavior of moving in the same direction when tightly packed, thus, lowering the relative velocity between agents and lowering the probability of collision.

Next, as the distance to home decreases the robots should align to ease traffic flow near the home base and thus has a negative relation. Lastly, for the alignment gain, if the mean error between the robot's individual dead reckoning localization estimate and its filtered localization estimate is large, this could be indicative of a false filtered localization estimate. Therefore, the robot would need to seek the known absolute information from the home base to reset its estimates, by following a group of robots until the home base is reached, and thus positively related. Each of these three effects are linearly summed together to provide equal potential for the dominating behavior to emerge and is displayed in Equation 33.

$$K_a = \rho_{norm} + (1 - d_{h_{norm}}) + e_{norm} \quad (33)$$

Note that the covariance and distance to goal stimuli do not influence this gain. This is due to the fact that covariance can be improved by receiving estimates from other robots and is thus used to affect the cohesion gain to keep the robots together. While the distance to goal stimuli is not included, since the goals are in random locations the robots should not follow each other to each other's goal.

Next for the cohesion gain, whose main focus is to keep robots together to perform more localization updates with other robots, the three stimuli used are estimated spectral norm of the covariance matrix, mean error between DR and the filter used, and distance to home. Since the covariance is decreased by performing relative measurement updates with other robots the cohesion gain should increase as the covariance increase, thus a positive effect is used. Next, as the robot's DR position estimate diverges from its filtered position estimate, this means that it has not recently received a global update, and thus it must rely on other robots that have recently received a global update to improve its estimated position. Therefore, cohesion should increase as the error increases. Also, as the distance to home increases the cohesive gain should increase, to promote robots providing measurement updates with each other when far from the global information source. Utilizing these three effects the cohesion gain equation is displayed in Equation 34.

$$K_c = ||\sigma_{norm}|| + e_{norm} + d_{h_{norm}} \quad (34)$$

Note that the local robot density stimulus is not included in this gain, due to the nature of this gain promoting the increase of density. This would then create either a positive feedback loop, which would cause a permanent dominance of cohesion, or a negative feedback loop, which would prevent cohesion from becoming dominant. In addition, the distance to the goal is not included, since the goals are randomly placed in the environment, the robots should separate when near their goal and thus its effect is used in the separation gain.

The third gain, separation, whose main focus is to promote diffusive behaviors allowing robots to achieve different tasks, is influenced by the spectral norm of the covariance matrix, mean error between DR and filter used, and distance to goal stimuli. Note that again local robot density does not influence this gain for similar reasons as the cohesive gain. The effect of the covariance on the separation gain negative, thus when the covariance is low, diffusive behaviors should occur to allow the robots to more directly achieve their tasks. Similarly, the effects of the mean error are negative to allow separation to dominate when the mean error is low. Next, the distance to goal stimuli also has a negative effect on separation, so when the robot is close to the goal location it can easily break from the group to reach the goal. Each of these effects are linearly combined and displayed in Equation 35.

$$K_s = (1 - e_{norm}) + (1 - d_{g_{norm}}) + (1 - ||\sigma_{norm}||) \quad (35)$$

The distance to home stimulus is not included in this gain, since the robots should not separate when close to home and risk not receiving a global update. Nor should the robots separate when far from home since it is unknown where the random goals are located.

The home gain is the first of the two newly introduced boids rules, and its primary purpose is to promote the robot to head back to the base to receive an absolute measurement. The three stimuli that affect the home gain are the spectral norm of the covariance matrix, mean error between DR and the filter used, and the distance to home. An increase in covariance has a positive effect on increasing the home gain. The reasoning for this is, that covariance can be reduced with measurement updates from other robots or from a global measurement of the home base. So, if the robot is unable to find any neighbors, it is more likely to find them near the home base or receive an absolute measurement from the home base. Similarly, as the mean error increases the robot must receive an absolute measurement from the base to improve the estimate, and therefore has a positive effect. Next, if the robot is near the home base it should deviate from its path to perform a global measurement, improving its localization estimate. Therefore, an increase in the distance to home stimulus has a negative effect on the home gain.

$$K_h = ||\sigma_{norm}|| + e_{norm} + (1 - d_{h_{norm}}) \quad (36)$$

Note that the local robot density and distance to the goal stimuli are not used for this gain. The local density is not used similarly for cohesion or separation gains, it can cause a positive or negative feedback loop that will trap the swarm at the home base or prevent them from reaching it. While the distance to the goal stimulus is not included since the goals are randomly placed, it is unknown if it is near the home base or not.

The last gain is the goal gain. This rule's focus is to promote the robot to achieve its task, which in this study is to navigate to a way-point. The three stimuli that affect the goal gain are the spectral norm of the covariance matrix, the mean error between the DR estimate and the filtered estimate, and the distance to the goal. Furthermore, it serves as the inverse rule for the home rule and thus its effects are the inverses of the home gain and the distance to the goal is used instead of the distance to home. Furthermore, the local robot density and distance to goal stimuli are not used for similar reasons as the home gain.

$$K_g = (1 - ||\sigma_{norm}||) + (1 - e_{norm}) + (1 - d_{g_{norm}}) \quad (37)$$

Thus, by adapting the gains of the boids rules based on the local stimuli of: the local robot density, the spectral norm of the covariance matrix, the mean error between the dead reckoning estimate and the filtered estimate, the distance to home, and the distance to the goal, the swarm will exhibit explicit behavior via the dominance of a boids rule. A summary of the effects of each stimulus on each gain is displayed in Table 1

Table 1: Summary of effects of local stimuli on the boids rule gains

	Robot Density Increase	Mean Error Increase	Covariance Increase	Distance to Home Increase	Distance to Goal Increase
K_a	increases	increases	N/A	decreases	N/A
K_c	N/A	increases	increases	increases	N/A
K_s	N/A	decreases	decreases	N/A	decreases
K_h	N/A	increases	increases	decreases	N/A
K_g	N/A	decreases	decreases	N/A	decreases

2. Simulation Setup

Each of the experiments were conducted in a 10m x 10m square world, with the a known home location placed at the center of the world. The robots are deployed from the home location, thus allowing for an immediate global home update, and are tasked with reaching as many random way-point goals as possible in the simulation time of 100 seconds.

a) Robot

The simulated robots were based on models of the I-Robot Create which are used as a common test-bed. The robots are outfitted with a range of sensors, but not all the sensors were required for the implementation of the algorithms. The sensors that were modeled for the simulations are:

1. Triaxial Inertial Measurement Unit (IMU): triaxial gyroscope, triaxial accelerometer triaxial magnetometer
2. 2D LiDAR
3. Monocular Camera
4. Wheel encoders
5. Bluetooth communication

The triaxial gyroscope and wheel encoders were used to perform angular and linear velocity measurements. The 2D LiDAR was used to identify and perform relative measurement updates with other robots. While the camera was used to detect the home base, for a global measurement when in range. The model for each robot sensor was based on the manufacturers noise models as well as empirical data collected through testing. Table 2 displays the standard deviation for the noise (σ_{Noise}) and the turn on bias standard deviation of the sensors (σ_B). Note that for simplicity all noise models were modeled as Gaussian distributions.

Furthermore, to account for differences in noise between devices, each individual robot's sensor's noise standard deviation was randomized $\pm 1\%$ in addition to their normally distributed random individual turn on bias based on the standard deviations found in Table 2. Therefore, the noise model for each measurement follows Equation 38.



Figure 3: I-Robot Create test platform the simulated robot is based on.

Table 2: Noise and bias models for robot sensors based on manufacturer data sheets and empirical testing.

	Velocity (m/s)	Yaw Rate (rad/s)	Range (m)	Bearing (rad)	Home (m)
σ_{β}	.005	.05	0	0	0
$\sigma_{N_{noise}}$	.05	.02	.01	.02	.001

$$\text{Measurement} = \text{Truth} + N(\sigma_{N_{noise}}) + \beta \quad (38)$$

Where $N(\sigma_{N_{noise}})$ is the normally distributed random noise, with a standard deviation of $\sigma_{N_{noise}}$ and β is the turn on bias that is instantiated at startup time using Equation 39.

$$\beta = N(\sigma_{Bias}) \quad (39)$$

In addition, to focus on local interactions the detection and communication radius for all sensors was limited to 1m. Furthermore, e_{max} and $||\sigma_{max}||$ were set at 1 m and 5 m, respectively.

IV. RESULTS & DISCUSSION

Utilizing the methodology outlined in Section III, 36 experiments were conducted with three different localization techniques (DR, CI, DEKF), three boids rules (goal following, static rules, adaptive rules), and four robot densities (0.05, 0.1, 0.25, and 0.5 robots/m²). For the localization techniques, DR serves as a non-cooperative baseline since it does not share information with other robots, while CI and DEKF serve as test cases for common decentralized localization filters. For boids rules, the goal following rules serve as a non-coordinated baseline, since only the goal rule is used. The static rules evaluate the benefit of using boids rules to improve localization. Furthermore, the proposed adaptive rules, demonstrate the benefits of augmenting the rules based on local stimuli. Lastly, multiple robot densities were used to demonstrate how the adaptive rules perform under low to high density robotic swarms. Each of these experiments was run 100 times and the spectral norm of the covariance matrix, the mean error between the truth and estimated position, goals reached, and goal success rate were recorded to evaluate the performance of the experiment.

1. Covariance

The first recorded result from each experiment was the spectral norm of the covariance matrix. The spectral norm was utilized to compress the 3x3 covariance matrix to a scalar that can be easily plotted. Figure 4 displays the average spectral norm of the covariance matrix across each time step and each robot used in each experiment and each trial, as shown in Equation 40. Where,

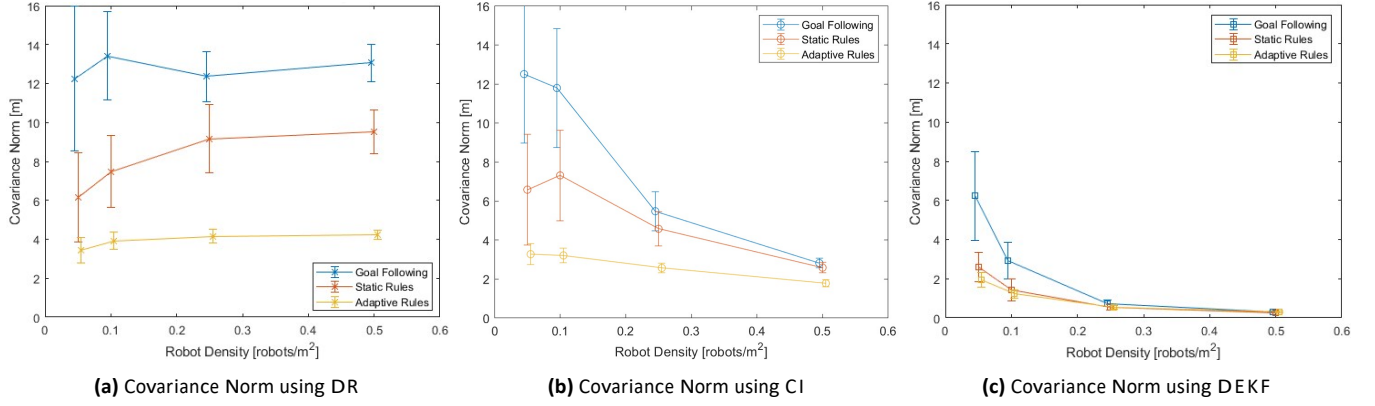


Figure 4: Average spectral norm of the covariance matrix for each estimator and boids rule scheme. Each data point represents the mean, and the error bars denote ± 1 standard deviation.

R is the total number of experimental trials, T is the number of time steps used in each trial, L is the number of robots used in each trial, and σ_{tir} is the spectral norm of the covariance matrix at that specific trial's time-step's robot.

$$||\sigma_{avg}|| = \frac{\prod_{r=1}^R \prod_{t=1}^T \prod_{i=1}^L ||\sigma_{tir}||}{T L R} \quad (40)$$

Furthermore, since covariance is a measure of the uncertainty in the robots location, the smaller the spectral norm of the covariance matrix the more certain the measurement is. Therefore a smaller covariance norm is desirable.

The first observation from Figures 4a-4c, is the overall performance of each estimator. DR (Figure 4a) performed noticeably worse than the other two estimators, as expected, while DEKF performed the best (Figure 4c). This trend is expected from the previous work, since DR does not cooperate with the other agents it is unable to share sensor information to improve its estimate. In addition, DEKF performed better than covariance intersection as expected from previous work since DEKF takes into account the cross correlation values of the covariance matrix between robots.

The second observation from Figures 4a-4c, is the overall performance of each boids rule scheme. Direct goal following rule scheme performed the worst, while the adaptive rules performed the best, with the static rules performing in-between. This result is expected since direct goal following does not coordinate the robot's movements with the other robots to perform more relative measurements, nor does it seek out the home base for a global measurement. Also, as expected the adaptive rules performed better than the static rules since they were able to augment their decisions based on the quality of the current localization estimate, to seek out other robots or the home base when necessary. Note that the DR covariance was able to be improved with the static and adaptive rules, due to the inclusion of the homing rule, encouraging the robot to return home for a global measurement of the home base.

In addition to these observations, as the robot density increased, the covariance began to converge and become independent of the boids rules for Covariance Intersection and DEKF. This is due to an increase in relative measurements due to random chance since the robots were closer together. Therefore, they did not need to search for other robots as often to improve their localization estimates. This is further confirmed by Dead Reckoning which performed relatively the same despite an increase in robot density, since it does not utilize information provided by neighboring robots.

Furthermore, it is observed that the inclusion of the boids rules, both static and adaptive, resulted in the largest improvement of the covariance, at low robot densities and the lesser performing localization methods. This is again due to fewer random relative measurements taking place, so the boids rules flock the swarm to increase relative measurements and global measurements as needed.

2. Mean Error

The second recorded result from each experiment was the mean error between the true location of the robot and the estimated location (Note that this is not the same mean error that is used to adapt the boids rule gains, which is between the DR estimate and the filter estimate). Figure 5 displays the average mean error across each time step and robot used in each experiment and trial, similar to covariance. In addition, since mean error is the measurement of how far the estimated position is from the truth, a smaller mean error is more desirable.

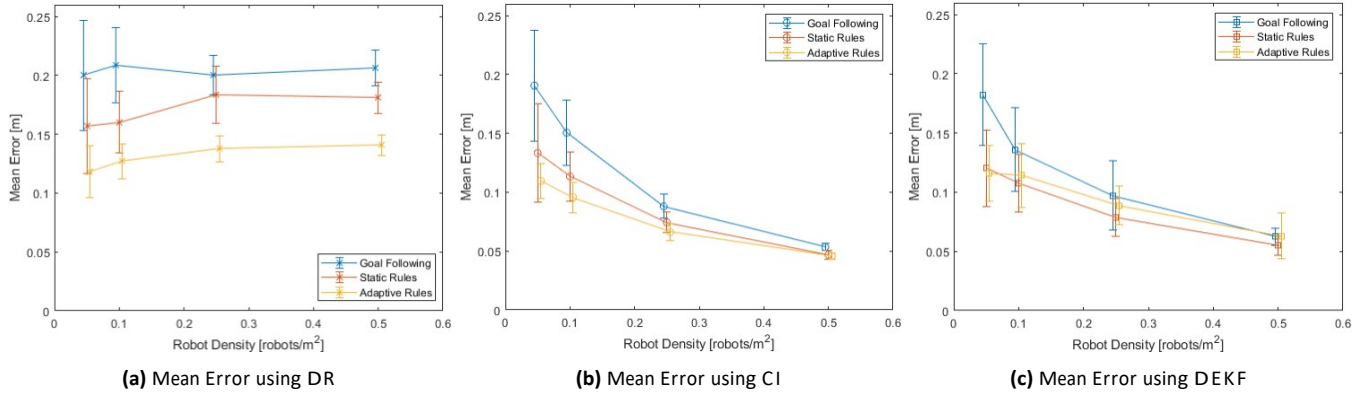


Figure 5: Mean error for each estimator and boids rule scheme. Each data point represents the mean, and the error bars denote ± 1 standard deviation.

The first observation from Figures 5a-5c is the overall performance of each estimator. Dead reckoning performed the worst (5a), while Covariance Intersection (5b) and DEKF (5c) performed approximately the same. Similarly to covariance, it was expected that Dead Reckoning performed the worst. However, the reason why Covariance Intersection and DEKF performed approximately the same is that the mean error is very low, approximately 1% of the length of the world, and thus improvement becomes more difficult. This was likely due to the simplification of the noise models and will be addressed in the future work of this study.

The next observation from Figures 5a-5c is the overall performance of each boids rule scheme. For both Dead Reckoning and Covariance Intersection, as expected, goal following performed the worst, static rules performed better, and the adaptive rules performed the best. However, for DEKF goal following performed the worst, but the static and adaptive rules performed similarly (within one standard deviation from each other). This similarity can be explained for the same reason Covariance Intersection performed similarly to DEKF. Since the mean error was low, it was more difficult for the adaptive rules to further improve the localization estimate from the static rules. Note again that Dead Reckoning was able to be improved with the static and adaptive rules due to the inclusion of the homing rule.

In addition, as the robot density increased, the mean error began to converge and become independent of the boids rule scheme. Further reinforcing that adaptive boids results in a larger improvement of the localization estimate at low robot densities with poor estimators.

3. Goals Reached

In addition to the performance of the localization estimate, the average number of goals achieved per agent was also recorded. Goals were defined as being reached if the true position of the robot was within the goal tolerance (.2m). However, the robots did not have environmental queues to verify if the goal was reached, so the robots had to rely on their estimated position only, to believe its goal was reached. Figure 6 displays the average goals reached per agent for each experiment, with the error bars denoting ± 1 standard deviation. Since the goals represent task completions, a larger number of goals reached is desirable.

When comparing the overall performance of each estimator, from Figures 6a-6c, Dead Reckoning performed the worst (6a) while Covariance Intersection (6b) and DEKF (6c) performed similarly. This is due to the ability to reach goals being dependent on the mean error of the robots. Therefore, since Dead Reckoning had the worst mean error and Covariance Intersection and DEKF had similar mean errors (Figure 5) the goals reached follow the same trend.

Next, when observing the overall performance of each Boid rule scheme in Figures 6a-6c, goal following performed the best, with adaptive next and static as the worst. This result is expected since, goal following does not need to deviate its path to coordinate with other agents, it can proceed directly to its goal, and thus reach more destinations in the given time. In addition, it is expected that the adaptive rules performed better than the static rules since the robots will only need to deviate from their direct goal path when necessary to improve their localization.

Therefore, including the adaptive boids rules comes at the cost of not achieving as many goals in the given time.

4. Goal Success Rate

Additionally, the average success rate of the goal was also recorded throughout each experiment. The average success rate of the goal was defined as the number of goals achieved accurately (within the 0.2 m goal tolerance), divided by the average total

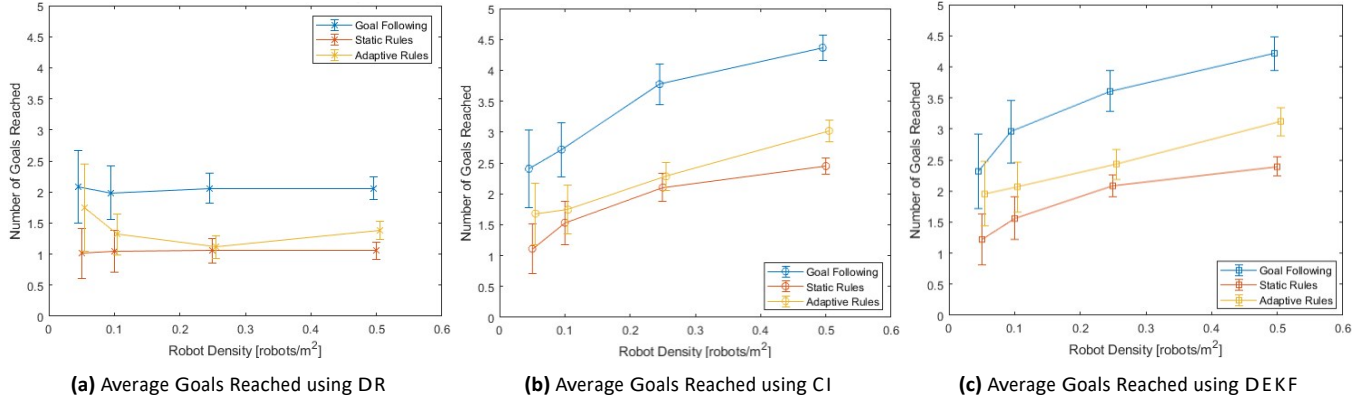


Figure 6: Average goals reached for each estimator and boids rule scheme. Each data point represents the mean, and the error bars denote ± 1 standard deviation.

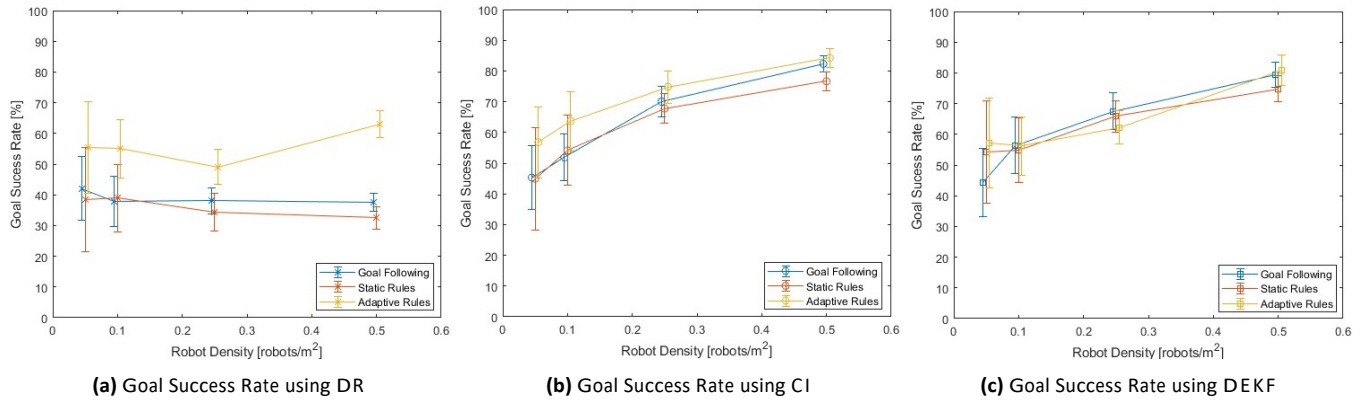


Figure 7: Goal success rate for each estimator and boids rule scheme. Each data point represents the mean, and the error bars denote ± 1 standard deviation.

number of goals each robot believed it reached. Figure 7 displays the average goal success rate for each experiment, where each data point represents the mean goal success rate and the error bars denote ± 1 standard deviation, on the distribution of the 100 trials.

The overall performance of the estimators for the goal success rate, from Figures 7a-7c, is Dead Reckoning (7a) performing the worst, and Covariance Intersection (7b) performing slightly better than DEKF (7c). Also, by observing Figures 7a-7c, the overall performance of each boids rule scheme is shown. Goal following and static rules performed similarly across all estimators. While the adaptive rules performed much better in Dead Reckoning and Covariance Intersection. Furthermore, all three boids rule schemes performed similar in DEKF. The differences in performance for each estimator and rule can be explained by the mean error which governs if the goals are achieved or failed. Since the mean error using the adaptive rules was much lower in DR and CI the goal success rate was much higher. Also for DEKF, where the mean errors for each boids rule scheme are similar, the goal success rates are similar.

Furthermore, since the goal following rules generally had a lower goal success rate than the other boids rules but achieved the most goals (Figure 6), this means that goal following was able to achieve more goals simply because more attempts were made in the time period of the experiments. Therefore, the adaptive rules promoted a higher accuracy of reaching a goal at the cost of reaching goals as fast as possible.

V. CONCLUSION & FUTURE WORK

This paper presents the introduction of swarm behavior as a form of coordination to improve the average localization estimate of the swarm. In addition to adapting the swarm behavior based on local stimuli to create a low-level decision-making framework. Specifically, utilizing adaptive boids rules improves the average localization estimate of the robot swarm. However, this comes at the cost of increasing the task completion time, since the robot has to divert its time to improve its localization estimate.

Furthermore, this algorithm improves the average localization estimate more as the quality of the estimators degrades and as swarm density decreases. This is because there are fewer random-occurring measurements taking place, so the robots must increase their coordination to conduct more relative measurements.

The future work of this study is to optimize the adaptive gain functions, allow some of the robots to stop and become landmarks, and demonstrate the algorithm on physical robots. In this work the adaptive gain functions were controlled by local stimuli that were combined by positive or negative linear relationships for simplicity. Therefore, future work will be spent to determine optimal relationships based on each stimuli, to further improve the localization estimate and increase the number of goals reached. Next, by allowing some of the robots in the swarm to stop and become landmarks far from the home base, as more robots perform relative measurements, the stopped robot's localization estimate will become very accurate. Therefore, the stopped robot will then behave like a home base, providing a global measurement to the other agents of the swarm. Furthermore, according to Kilic et al. (2019, 2021), Skog et al. (2010), and Zhang et al. (2017), by stopping the swarm robots, they will be able to perform zero-velocity updates (ZUPT) to calibrate IMU biases and improve their dead reckoning estimates. Lastly, since this algorithm demonstrates a greater improvement of the localization estimate at low robot densities, the future work of this study will be to apply this algorithm on a small set of physical robots. Thus, confirming the results of the simulation.

ACKNOWLEDGEMENTS

This study was supported in part by the National Science Foundation, under NSF, Award #1851815. The authors would also like to thank Dr. Guilherme A. S. Pereira, Dr. Ali Baheri, and Dr. Xi Yu for providing guidance throughout the research process. We would like to also thank all of the other participating members of WVU's REU 2021 program for assisting with the formation of ideas and implementation of this research.

REFERENCES

- Alaliyat, S., Yndestad, H., and Sanfilippo, F. (2014). Optimisation of boids swarm model based on genetic algorithm and particle swarm optimisation algorithm (comparative study). In *ECMS*, pages 643–650. Citeseer.
- Bailey, T., Bryson, M., Mu, H., Vial, J., McCalman, L., and Durrant-Whyte, H. (2011). Decentralised cooperative localisation for heterogeneous teams of mobile robots. In *2011 IEEE International Conference on Robotics and Automation*, pages 2859–2865. IEEE.
- Bong-Su, C., Woo-sung, M., Woo-Jin, S., and Kwang-Ryul, B. (2011). A dead reckoning localization system for mobile robots using inertial sensors and wheel revolution encoding. *Journal of Mechanical Science and Technology*, 25(11):2907–2917.
- Carrillo-Arce, L. C., Nerurkar, E. D., Gordillo, J. L., and Roumeliotis, S. I. (2013). Decentralized multi-robot cooperative localization using covariance intersection. pages 1412–1417.
- Cho, B. S., Moon, W. S., Seo, W. J., and Baek, K. R. (2012). A dead reckoning localization system for mobile robots using inertial sensors and wheel revolution encoding.
- Delgado-Mata, C., Martinez, J. I., Bee, S., Ruiz-Rodarte, R., and Aylett, R. (2007). On the use of virtual animals with artificial fear in virtual environments. *New Generation Computing*, 25(2):145–169.
- Franken, D. and Hupper, A. (2005). Improved fast covariance intersection for distributed data fusion. In *2005 7th International Conference on Information Fusion*, volume 1, pages 7–pp. IEEE.
- Gautam, A. and Moh, S. (2012). A review of research in multi-robot systems.
- Gross, J., De Petrillo, M., Beard, J., Nichols, H., Swiger, T., Watson, R., Kirk, C., Kilic, C., Hikes, J., Upton, E., et al. (2019). Field-testing of a uav-ugv team for gnss-denied navigation in subterranean environments. In *Proceedings of the 32nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2019)*, pages 2112–2124.
- Gu, Y., Ohi, N., Lassak, K., Strader, J., Kogan, L., Hypes, A., Harper, S., Hu, B., Gramlich, M., Kavi, R., et al. (2018). Cataglyphis: An autonomous sample return rover. *Journal of Field Robotics*, 35(2):248–274.
- Hardy, J., Strader, J., Gross, J. N., Gu, Y., Keck, M., Douglas, J., and Taylor, C. N. (2016). Unmanned aerial vehicle relative navigation ingps denied environments. In *2016 IEEE/ION Position, Location and Navigation Symposium*. IEEE/ION.
- Hartman, C. and Benes, B. (2006). Autonomous boids. *Computer Animation and Virtual Worlds*, 17(3-4):199–206.
- Jevtić, A. and Andina de la Fuente, D. (2007). Swarm intelligence and its applications in swarm robotics.
- Kilic, C., Gross, J. N., Ohi, N., Watson, R., Strader, J., Swiger, T., Harper, S., and Gu, Y. (2019). Improved planetary rover inertial

- navigation and wheel odometry performance through periodic use of zero-type constraints. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 552–559.
- Kilic, C., Ohi, N., , Gu, Y., and Gross, J. N. (2021). Slip-based autonomous zupt through gaussian process to improve planetary rover localization. *IEEE Robotics and Automation Letters*, 6(3):4782–4789.
- Luft, L., Schubert, T., Roumeliotis, S. I., and Burgard, W. (2016). Recursive decentralized collaborative localization for sparsely communicating robots. In *Robotics: Science and Systems*. New York.
- Luft, L., Schubert, T., Roumeliotis, S. I., and Burgard, W. (2018). Recursive decentralized localization for multi-robot systems with asynchronous pairwise communication. *The International Journal of Robotics Research*, 37(10):1152–1167.
- Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 25–34.
- Rubenstein, M., Cornejo, A., and Nagpal, R. (2014). Programmable self-assembly in a thousand-robot swarm. *Science*, 345(6198):795–799.
- Skog, I., Handel, P., Nilsson, J.-O., and Rantakokko, J. (2010). Zero-velocity detection—an algorithm evaluation. *IEEE transactions on biomedical engineering*, 57(11):2657–2666.
- Wu, H., Qu, S., Xu, D., and Chen, C. (2014). Precise localization and formation control of swarm robots via wireless sensor networks.
- Yang, C., Strader, J., and Gu, Y. (2021). A scalable framework for map matching based cooperative localization. *Sensors*, 21(19):6400.
- Zhang, R., Yang, H., Höflinger, F., and Reindl, L. M. (2017). Adaptive zero velocity update based on velocity classification for pedestrian tracking. *IEEE Sensors journal*, 17(7):2137–2145.