ARREST: A RSSI Based Approach for Mobile Sensing and Tracking of a Moving Object

Pradipta Ghosh
University of Southern California
Email: pradiptg@usc.edu

Jason A. Tran
University of Southern California
Email: jasontra@usc.edu

Bhaskar Krishnamachari
University of Southern California
Email: bkrishna@usc.edu

***Abstract*—We present Autonomous Rssi based RElative poSitioning and Tracking (ARREST), a new robotic sensing system for tracking and following a moving, RF-emitting object, which we refer to as the Leader, solely based on signal strength information. Our proposed tracking agent, which we refer to as the TrackBot, uses a single rotating, off-the-shelf, directional antenna, novel angle and relative speed estimation algorithms, and Kalman filtering to continually estimate the relative position of the Leader with decimeter level accuracy (which is comparable to a state-of-the-art multiple access point based RF-localization system) and the relative speed of the Leader with accuracy on the order of 1 m/s. The TrackBot feeds the relative position and speed estimates into a Linear Quadratic Gaussian (LQG) controller to generate a set of control outputs to control the orientation and the movement of the TrackBot. We perform an extensive set of real world experiments with a full-fledged prototype to demonstrate that the TrackBot is able to stay within 5m of the Leader with: (1) more than 99% probability in line of sight scenarios, and (2) more than 75% probability in no line of sight scenarios, when it moves 1.8X faster than the Leader.**

I. Introduction

Sensing and tracking of a moving object/human by a robot is an important topic of research in the field of robotics and automation for enabling collaborative work environments [1], including applications such as fire fighting and exploration of unknown terrains [2]. For example, in disaster managements, robots can assist by tracking and following first-responders while the team explores an unknown environment. To achieve this, staying in proximity to the first-responders is key. In this paper, we focus on this class of tracking problems where the term **"tracking"** refers to the relative position sensing and control of a robot that is required to stay in proximity to an uncontrolled moving target such as a Leader robot or human.

Our Contribution: We propose the Autonomous Rssi based RElative poSitioning and Tracking (ARREST), *a purely Radio Signal Strength Information (RSSI) based single node RF sensing system for joint location, angle, and speed estimation and bounded distance tracking of a target moving arbitrarily in 2-D that can be implemented using commodity hardware.* In our proposed system, the target, which we refer to as the *Leader*, carries an RF-emitting device that sends out periodic beacons. The tracking robot, which we refer to as the *TrackBot*, employs an off-the-shelf directional antenna, novel relative position and speed estimation algorithms, and a Linear Quadratic Gaussian (LQG) controller to measure the RSSI of the beacons and control its maneuvers.

Related Works: The most popular class of existing tracking architectures employs vision and laser range finder systems [3, 4] which crumbles in effectiveness when visibility deteriorates or direct line of sight does not exist. Also, the processing of respective data, namely image processing, increases the form factor and power consumption of power constrained robots. As alternatives, the RF Localization related works in wireless sensor networks [5] where robots are employed for localizing static nodes, are relevant. Graefenstein *et al.* [6] employed a rotating antenna on a mobile robot to map the RSSI of a region and exploit the map to localize the static nodes. Similar works have been proposed in the context of locating static radio sources such as radio tagged fish or wild animals [7, 8]. The works of Zickler and Veloso [9], and Oliveira *et al.* [10] on RF-based relative localization are also relevant as we also employ a RF based relative localization in our system. Vasisht, Kumar, and Katabi [11] have applied a MIMO-based system to relatively localize a single node. Simulation of a RSSI based constant distance following technique is demonstrated in [12] where the leader movement path is predetermined and known to the Follower. *However, unlike these works, the TrackBot in the ARREST system relies solely on RSSI data not only for the localization of the mobile Leader with unknown movement pattern, but also for autonomous motion control with the goal of maintaining a bounded distance.* The closest state-of-the-art related to our work is presented in [13]. In this work, the authors developed a system that follows the bearing of a directional antenna for effective communication. However, to our knowledge, the maintenance of guaranteed close proximity to the Leader was not discussed in [13], which is the most important goal in our work. Moreover, this work employs both RSSI and sonar to determine the orientation of the transmitter antenna along with comparatively costly and high power consuming processing hardware with larger form factor. On LQG related works, Bertsekas [14] has demonstrated that a LQG controller can provide the optimal control of a robot along a known/pre-calculated path, when the uncertainty in the motion as well as the noise in observations are Gaussian. Extending this concept, LQG based robotic path planning solutions to deal with uncertainties and imperfect state observations are presented in [15, 16]. *To the best of our knowledge, we are the first to combine RSSI-based relative position, angle, and speed estimation with the LQG controller for localizing and tracking a moving RF-emitting object.*

II. Problem Formulation

Let the location of the *Leader* at time t be represented as $\mathbf{X}_L(t) = (x_L(t), y_L(t))$ in a 2D global frame of reference, $\mathcal{R}_G$. The *Leader* follows an unknown path, $\mathcal{P}_L$. Similarly, let the position of the TrackBot at any time instant t be denoted by $\mathbf{X}_F(t) = (x_F(t), y_F(t))$. The maximum speeds of the Leader and the TrackBot are v_L^{max} and v_F^{max}, respectively. For simplicity, we discretize the time with steps of $\delta t > 0$ and use the notation n to refer to the n^{th} time step i.e., $t = n \cdot \delta t$. Let $d[n] = ||\mathbf{X}_L[n] - \mathbf{X}_F[n]||_2$ be the distance between the TrackBot and the Leader at time slot n, where $||.||_2$ denotes the L_2 norm. Then, with D_{th} denoting the max distance allowed between the Leader (L) and the TrackBot (F), the objective of tracking is to plan the TrackBot's path, $\mathcal{P}_F$, such that $\mathbb{P}(d[n] \leq D_{th}) \approx 1 \;\; \forall n$ where $\mathbb{P}(.)$ denotes the probability. However, in real systems all measurements at any time n by the TrackBot are in the TrackBot's 2D local frame of reference, $\mathcal{R}_F[n]$, with the origin representing the location of the TrackBot, $\mathbf{X}_F[n]$. Let the robot's forward and backward movements at any time instant n be aligned with the X-axis of $\mathcal{R}_F[n]$. Let the position of the Leader in $\mathcal{R}_F[n]$ be $\mathbf{X}_L^{rel}[n] = (x_L^{rel}[n], y_L^{rel}[n])$. Now, to restate the objective of tracking in terms of the local coordinates: $\mathbb{P}(d[n] \leq D_{th}) \approx 1 \;\; \forall n$ where $d[n] = ||\mathbf{X}_L^{rel}[n]||_2 = (x_L^{rel}[n]^2 + y_L^{rel}[n]^2)^{1/2}$.

III. The ARREST System

In our proposed ARREST system, the Leader is a robot or a human carrying a device that periodically transmits RF beacons, and the TrackBot is a robot carrying a directional, off-the-shelf RF receiver. As shown in Fig. 1b, the ARREST architecture consists of three layers: Communication ANd Estimation (CANE), Control And STate update (CAST), and Physical RobotIc ControllEr (PRICE). In order to track the Leader, the TrackBot needs sufficiently accurate estimations of both the Leader's relative position ($\mathbf{X}_L^{rel}$) and relative speed (v_{rel}). Thus, at any time instant $[n]$, we define the state of the TrackBot as a 3-tuple: $\mathcal{S}[n] = \begin{bmatrix} d^e[n], & v_{rel}^e[n], & \theta_{rel}^e[n] \end{bmatrix}^T$ *where the superscript* e *refers to the estimated values,* $d^e[n] = ||\mathbf{X}_L^{rel}[n]||_2$ refers to the estimated distance at time-slot n, $v_{rel}^e[n]$ refers to the relative speed of the TrackBot along the X-axis of $\mathcal{R}_F[n]$ with respect to the Leader, and $\theta_{rel}^e[n]$ refers to the angular orientation (in radians) of the Leader in $\mathcal{R}_F[n]$.

CANE: The function of the CANE layer is to measure RSSI values from the beacons and approximate the Leader's position relative to the TrackBot, (i.e., $d^e[n]$ and $\theta_{rel}^e[n]$). The CANE layer is broken down into three modules: Wireless Communication and Sensing, Rotating Platform Assembly, and Relative Position Estimation. At the beginning of each time slot n, the Wireless Communication and Sensing module and the Rotating Platform Assembly perform a 360° RSSI sweep by physically rotating the directional antenna while storing RSSI measurements of successful beacon receptions into the vector $\mathbf{r_v}[n]$. The Relative Position Estimation module uses $\mathbf{r_v}[n]$ to approximate the relative position of the Leader by leveraging pre-estimated directional gains of the antenna, detailed in Section IV.

CAST: The functions of the CAST layer is to maintain the 3-tuple state estimates and to generate control commands based on current and past observations to send to the PRICE layer. The CAST layer consists of two different modules: the Linear Quadratic Gaussian (LQG) Controller and the Strategic Speed Controller. The *Strategic* Speed Controller estimates the relative speed of the Leader by exploiting past and current state information and generates the speed control signal in conjunction with the LQG controller according to two different strategies, Optimistic and Pragmatic (detailed in Section IV-C). The LQG controller incorporates past state information, past control information, and relative position and speed approximations (discussed further in Section III-A) to: (1) generate the system's instantaneous state, (2) determine how much to rotate the TrackBot itself, and (3) determine what should be the TrackBot's relative speed. The state information generated by the LQG controller is directly sent to the Strategic Speed Controller to calculate the absolute speed of the TrackBot.

PRICE: The goal of the PRICE layer is to convert the control signals from the CAST layer into actual translational and rotational motions of the TrackBot. It consists of two modules: Movement Translator and Robot Chassis. The Movement Translator maps the control signals from the CAST layer to a series of platform-specific Robot Chassis motor control signals (detailed in Section V).

A. Proposed LQG Formulation

In our proposed solution, we formulate the movement control problem of the TrackBot as a discrete time Linear Quadratic Gaussian (LQG) control problem [17]. The linear system equations for a discrete LQG problem can be written as:

$$\mathcal{S}[n+1] = A_n \mathcal{S}[n] + B_n \mathbf{U}[n] + \mathbf{Z}[n] \text{ and } \mathbf{O}[n] = C_n \mathcal{S}[n] + \mathbf{W}[n] \tag{1}$$

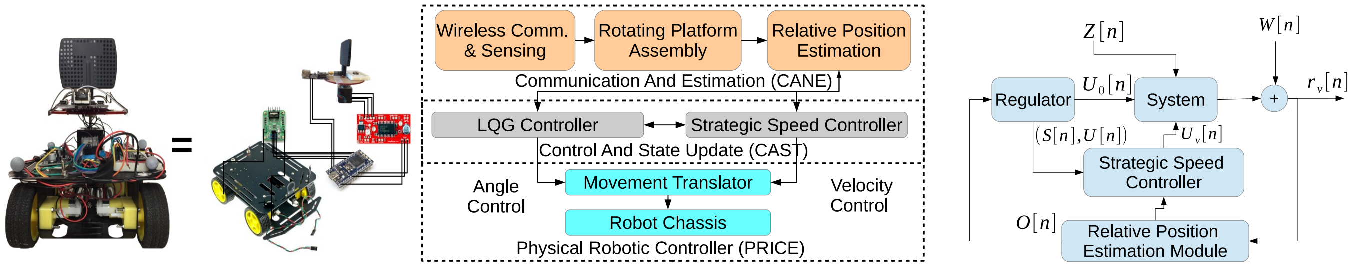
where A_n and B_n are the state transition matrices, $\mathbf{U}[n]$ is the LQG control vector, $\mathbf{Z}[n]$ is the system noise, $\mathbf{O}[n]$ is the LQG system's observation vector, C_n is the state-to-observation transformation matrix, and $\mathbf{W}[n]$ is the observation noise at time-slot n. In our case, $\mathbf{O}[n] = \begin{bmatrix} d^m[n], & v_{rel}^m[n], & \theta_{rel}^m[n] \end{bmatrix}^T$ (*the superscript* m *refers to measured values*), the state transition matrices $A_n = A$, $B_n = B$, $C_n = C$ are time invariant, and the time horizon is infinite as we do not have any control over the Leader's movements. For a infinite time horizon LQG problem [14], the cost function can be written as:

$$J = \lim_{N \to \infty} \frac{1}{N} \mathbb{E} \left(\sum_{n=0}^{N} \mathcal{S}[n]^T \mathbf{Q} \mathcal{S}[n] + \mathbf{U}[n]^T \mathbf{H} \mathbf{U}[n] \right) \tag{2}$$

where $\mathbf{Q} \geq 0, \mathbf{H} > 0$ are the weighting matrices. In our system, the state transition matrix values are as follows:

$$\mathbf{A} = \begin{bmatrix} 1 & -\delta t & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{B} = \begin{bmatrix} 0 & -\delta t & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

where δt *is the time granularity for the state update.* We assume the system noise, $Z[n]$, to be Gaussian and the measurement noise, $W[n]$, to be approximated as Gaussian. Furthermore, we tweak the LQG controller to send out a rotational control signal after a state update and before generating the LQR control signals, $\mathbf{U}[n]$, to align the robot toward



(a) (b) (c)

Fig. 1. (a) The TrackBot Prototype (b) The ARREST Architecture (c) Proposed LQG Controller System

the estimated direction of the Leader before calculating the movement speed. A block diagram of our LQG control system model is presented in Fig. 1c.

IV. RSSI Based Relative Position and Speed Observations

In this section, we discuss our methodologies to map the observed RSSI vector, $\mathbf{r_v}[n]$, into the observation vector, $\mathbf{O}[n]$.

A. Distance Observations

To calculate distance, we employ one the standard equations used in calculating the received power in a RF channel [18]:

$$\begin{aligned} P_{r,dBm} &= P_{t,dBm} + G_{dB} - \mathcal{L}_{ref} - 10\eta \log_{10} \frac{d^m[n]}{d_{ref}} + \psi \\ P^{ref}_{r,dBm} &= P_{t,dBm} + G_{dB} - \mathcal{L}_{ref} + \psi \\ \implies \frac{d^m[n]}{d_{ref}} &\approx 10^{\frac{\left(P^{ref}_{r,dBm} - P_{r,dBm}\right)}{10\cdot\eta}} \end{aligned} \tag{4}$$

where $P_{r,dBm} = avg\ \{\mathbf{r_v}[n]\}$ is the average received power in our system, $P_{t,dBm}$ is the transmitter power in dBm, G_{dB} is the average antenna gain in dB, $\mathcal{L}_{ref}$ is the path loss at the known reference distance d_{ref} in dB, η is the path loss exponent, $d^m[n]$ is the distance between the transmitter and receiver, ψ is the random shadowing and multipath fading noise in dB, and $P^{ref}_{r,dBm}$ is the pre-estimated received power at the reference distance (d_{ref}). More comprehensive details can be found in [19].

B. Angle Observations

The core of our Angle of Arrival (AoA) observation methods, called *pattern correlation*, is to correlate the normalized vector of RSSI measurements, $\mathbf{g_m} = \mathbf{r_v}[n] - \max(\mathbf{r_v}[n])$, with another vector representing the different θ shifted version of the antenna's known, normalized gain pattern, $\mathbf{g_{abs}}(\theta)$:

$$\begin{aligned} &\mathbf{r_v}[n] = [r_{-180}, r_{-178.2}, \cdots, r_{178.2}] \rightarrow \mathbf{g_m} = [r'_{-180}, \cdots, r'_{178.2}] \\ &\mathbf{g_{abs}}(\theta) = [g_{(-180+\theta)}, \cdots, g_{(0+\theta)}, \cdots, g_{(178.2+\theta)}] \end{aligned} \tag{5}$$

where r_ϕ refers to the RSSI measurement, g_ϕ refers to the antenna gain, and $r'_\phi = r_\phi - \max\{\mathbf{r_v}\}$ refers to the observed gain for the antenna orientation of $\phi°$ with respect to the X-axis of $\mathcal{R}_F[n]$. Due to our hardware restriction on angular step size of $1.8°$, the possible antenna orientations (ϕ) are limited to $\Theta = \{-180, \cdots, -1.8, 0, \cdots, 178.2\}$. Below, we describe three AoA observation methods in increasing order of complexity.

1) Basic Correlation Method: The first method (originally demonstrated by [6]) of determining AoA correlates $\mathbf{g_m}$ with $\mathbf{g_{abs}}(\theta)\ \forall\theta \in \Theta$ and calculates the respective L_2 distances. The observed AoA is the θ at which the L_2 distance is the smallest:

$$\theta^m_{rel} = \underset{\theta\in\Theta}{\arg\min} \sum_{k\in\Theta} \omega_k \cdot ||r'_k - g_{(k+\theta)}||_2 \cdot \mathbb{I}_{r'_k} \tag{6}$$

where the indicator function $\mathbb{I}_{r'_k}$ indicates whether the sample r'_k exists or not to account for missing samples in real experiments, and $\omega_k = 1$ is a constant.

2) Clustering Method: While the first method works well if enough uniformly distributed samples ($\geq$ 100 in our implementation) are collected within the $360°$ scan, it fails in scenarios of sparse, non-uniform sampling due to packet loss. In real experiments (mainly indoors), the collected RSSI samples can be uniformly sparse or sometimes batched sparse (samples form clusters with large gaps ($\approx 30°$) between them).

Definition 1. *An angular cluster (Λ) is a set of valid samples for a contiguous set of angles:* $\Lambda = \{k|\mathbb{I}_{r'_k} = 1 \forall k \in \{\phi_f, \phi_f + 1.8, \cdots, \phi_l\} \subset \Theta\}$.

To prevent undue bias from large cardinality clusters that can cause errors in estimating the correlation, we assign a weight (ω_k) to each sample (k) and use the pattern correlation method as in Eqn. (6). In our weighting scheme, we assign $\omega_k = \frac{1}{|\Lambda|}$ where $k \in \Lambda$.

3) Weighted Average Method: Based on real world experiments, we find that the angle observation based on the basic correlation method, say θ^1_m, gives reasonable error performance if the average cluster size, denoted by λ_a, is greater than the average gap size between clusters, μ_a. Conversely, the angle observation based on the clustering method, say θ^2_m, is better if $\lambda_a << \mu_a$. Thus, as a trade-off between both the basic correlation method and the clustering method, we propose a weighted averaging method described below.

$$\theta^m_{rel} = \begin{cases} \frac{\lambda_a}{\mu_a} \cdot \theta^1_m + (1 - \frac{\lambda_a}{\mu_a}) \cdot \theta^2_m & \text{if } \lambda_a \leq \mu_a \\ \theta^1_m & \text{if } \lambda_a > \mu_a \end{cases} \tag{7}$$

In the rest of the paper, we use the weighted average method for angle observations.

C. Speed Observations

In our ARREST architecture, the Strategic Speed Controller uses the relative position observations $(d^m[n], \theta^m_{rel}[n])$ from the CANE layer and the past LQG state estimates to determine the current relative speed, $v^m_{rel}[n]$, as well as the Leader's speed, $v^m_L[n]$. In this context, we employ two different observation strategies. The first strategy, which we refer to as the *Optimistic strategy*, assumes that the Leader will be static for the next time slot and determines the relative speed as follows:

$$v^m_{rel}[n] = v^e_{rel}[n] - \frac{(d^m[n] - d^e[n] \cdot \cos\theta^m_{rel}[n])}{\delta t} \text{ and } v^e_L[n+1] = 0 \tag{8}$$

On the other hand, the *Pragmatic Strategy* assumes that the Leader will continue traveling at the observed speed, $v^m_L[n]$. The strategy determines the relative speed as follows (illustrated in Fig. 2):

$$v_L[n] = \frac{\left(\left(d^e[n] - d^m[n] \cdot \cos\theta^m_{rel}[n]\right)^2 + \left(d^m[n] \cdot \sin\theta^m_{rel}[n]\right)^2\right)^{1/2}}{\delta t}$$
$$\theta_v[n] = \arctan\frac{d^m[n] \cdot \sin\theta^m_{rel}[n]}{d^m[n] \cdot \cos\theta^m_{rel}[n] - d^e[n]} - \theta^m_{rel}[n]$$
$$v^e_L[n+1] = v^m_L[n] = v_L[n] \cdot \cos(\theta_v[n]) \text{ and } v^m_{rel}[n] = v_F[n] - v^m_L[n] \tag{9}$$

Next, the LQG controller uses the observation vector $\mathbf{O}[n]$ to decide the next state's relative speed, $v^e_{rel}[n+1]$ which is used by the Speed Controller to generate the TrackBot's actual speed for next time step, $v_F[n+1] = v^e_L[n+1] + v^e_{rel}[n+1]$. In addition to the different assumptions about the Leader's speed, the Optimistic Strategy assumes that the noise in speed observations are uncorrelated with the noise in distance observations, whereas the Pragmatic strategy assumes strong correlation between distance and speed estimation noise.

V. TrackBot Prototype

A. Hardware

We implemented a TrackBot with our ARREST architecture inside a real, low-cost robot prototype presented in Fig. 1a. For a concise description of our prototype, we list the hardware used for implementation of each of the ARREST components in Table I. In the TrackBot prototype, the directional antenna and the OpenMote are mounted on top of a stepper motor using a plate. While we use two microprocessors (the OpenMote and the mbed), the system can be implemented using one microprocessor. We choose to use two in this prototype to work around wiring issues and work around the lack of sufficient GPIO pins on the OpenMote. The OpenMote is only used for RF sensing. The mbed, programmed using the mbed Operating System [20], sends control signals to rotate the stepper motor in precise steps of 1.8°. *Each consecutive 360° antenna rotations alternate between clockwise and anti-clockwise because this: (1) prevents any wire twisting between the mbed and OpenMote and (2) compensates for the stepper motor's movement errors.*

In the current prototype, The TrackBot first performs a 360° RSSI scan in $2s$ while collecting 200 samples and then moves the chassis with a maximum speed of $30cm/s$. To keep the movement simple, the TrackBot first rotates to the desired direction and then moves straight with the desired speed. To control the wheels, PWM signals are sent from the mbed

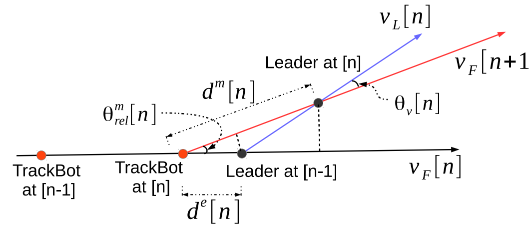


Fig. 2. Illustration of the Relative Speed Observation.

TABLE I
ARREST Hardware Implementation

	Module	Hardware
CANE	Wireless Communication and Sensing	OpenMote[21]; Rosewill Directional Antenna (Model RNX-AD7D with Max Gain: 5dBi, HPBW: 70°)
CANE	Rotating Platform Assembly	Nema 17 (4-wire bipolar Stepper Motor with Step size: 1.8 degrees (200 steps/rev)); EasyDriver - Stepper Motor Driver; mbed NXP LPC1768 [22]
CANE	Relative Position Estimation	mbed NXP LPC1768 [22]
CAST		mbed NXP LPC1768 [22]
PRICE	Movement Translator	mbed NXP LPC1768 [22]
PRICE	Robot Chassis	Baron-4WD Mobile Platform, L298N Stepper Motor Driver Controller Board, HC-SR04 Ultrasonic Sensor

to the motors with a period of $2s$. We choose a $2s$ period here because one $2s$ pulse width equates to a chassis rotation amount of $\approx 180°$ in this setup. We also choose the same period length ($2s$) for forward movement which caps the speed of the robot at $60/6 = 10cm/s$ (taking into account the $2s$ RSSI scan). The whole system is powered by five AA batteries which can run for a total of $\approx 3-4$ hours.

The Leader node is currently implemented as an OpenMote transmitting beacons with the standard omnidirectional antenna and a transmit power of $7dBm$. For programming of the OpenMotes, we use the RIOT operating system [23]. The Leader implementation is capable of transmitting 200 packets/second.

B. ARREST System Parameter Setup

1) Cost Parameters Setup: In the cost function of our LQG formulation, the matrix $\mathbf{Q}$ is a 3×3 positive definite diagonal matrix: $\mathbf{Q} = diag\{Q_d, Q_v, Q_\theta\}$. Our main goal is to keep the distance as well as the relative angle to be as low as possible while keeping emphasis on the distance. From this perspective, we perform a set of experiments to find a good trade-off between Q_v, Q_θ and Q_d where we vary one parameter while keeping the rest of them fixed. Based on these experiments, we opt for the following settings: $Q_v = 0.1$, $Q_\theta = 1$ and $Q_d = 10 \cdot v^{max}_L$ where v^{max}_L is the maximum speed of the Leader. With these settings, our system performs better than any other explored settings. Furthermore, $\mathbf{H}$ is chosen to be a 3×3 Identity matrix.

2) Noise Covariance Matrix Parameters Setup: In our implementation, the system noises are assumed to be i.i.d normal random variables with Σ_{ZZ} being a 3×3 identity matrix. On the other hand, the observation noise covariance matrix requires separate settings for the different strategies. For the Optimistic strategy, we assume that the observation noises are uncorrelated, whereas, for the Pragmatic strategy, the distance estimation errors and the relative speed estimation errors are highly correlated with variances proportional to v_L^{max}. A set of empirically determined values of Σ_{WW} for the Optimistic and the Pragmatic strategies are as follows.

$$\Sigma_{WW}^{Op} = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \Sigma_{WW}^{Pg} = \begin{bmatrix} 1 & v_L^{max} & 0 \\ v_L^{max} & (v_F^{max})^2 & 0 \\ 0 & 0 & 0.1 \end{bmatrix} \quad (10)$$

where Op and Pg refers to the Optimistic and the Pragmatic strategies, respectively.

VI. Experiment Results

A. Baseline Analysis via Emulation

We employ our hardware prototypes to collect an extensive set of RSSI data in cluttered indoor and outdoor environments and use the collected samples to perform a set of emulation experiments as a baseline for our real-world experiments (for details, refer to [19]). For these experiments, we choose a value of $\delta t = 1s$ in (3) to match the maximum achievable speed of our stepper motor. In these experiments, we compare the performance among the two proposed strategies, Optimistic and Pragmatic, and a Baseline algorithm. In the **Baseline algorithm**, the TrackBot estimates the relative position via the basic correlation method (discussed in IV-B1). Once the direction is determined, the TrackBot rotates to align itself toward the estimated direction and then moves with a speed of $\min\{v_F^{max}, \frac{d^e[n]}{\delta t}\}$. In these experiments, we vary the max speed of the TrackBot, v_F^{max}, while keeping the max speed of the Leader, v_L^{max}, fixed and vice versa. We found that Pragmatic Strategy performs best for $1.6 \cdot v_L^{max} < v_F^{max} < 3 \cdot v_L^{max}$ due to adaptability and accuracy of the speed information. On the other hand, the Optimistic Strategy performs best for $v_F^{max} \geq 3 \cdot v_L^{max}$. The reason is that the Leader may constantly change movement directions while the TrackBot travels in straight lines to catch up with the Leader. Therefore, the TrackBot does not directly follow the Leader's path. This results in oscillations in the movement pattern for the Pragmatic strategy while the Optimistic strategy avoids oscillations since it assumes the Leader to be static. The Baseline algorithm performs the worst due to the lack of leveraging past observations to adapt its speed. We also discovered that the ARREST system fails to stay with $5m$ of the Leader if $v_L^{max} > 3m/s$ under all the explored settings (detailed in [19]). However, for $v_L^{max} \leq 3m/s$ and $v_F^{max} = 1.8 \cdot v_L^{max}$, the TrackBot with Pragmatic policy stays within $5m$ of the Leader with probability $\approx 100\%$. We opt for $v_F^{max} = 1.8 \cdot v_L^{max}$ in the real system due to lesser speed constraints. Furthermore, we choose the Pragmatic strategy as it performs best among all three strategies for $v_F^{max} = 1.8 \cdot v_L^{max}$. With these settings, the emulation based error statistics in the the estimation of distance, angle, and speed can be summarized as follows: □ Absolute distance estimation errors are $< 100cm$ with probability $\approx 90\%$ and $< 150cm$ with probability $\approx 100\%$ which is comparable to standard RF based localization methods [5]. □ Absolute angle estimation errors are $< 40°$ with probability $\approx 80\%$. This is justified as the HPBW specification for the antenna is approx $70°$. □ Absolute speed estimation errors are less than $1m/s$ with probability $\approx 90\%$.

B. Real Experiment Results

In this section, we analyze the performance of the ARREST system via a set of real world experiments using the TrackBot prototype.

1) Method: Based on the valuable insights from the emulation results, we choose TrackBot's speed to be at least 1.8X the Leader's speed. The TrackBot makes a decision every $6s$. Between each decision, the TrackBot takes $2s$ for both the antenna rotation and RSSI scan, $2s$ for the chassis rotation, and $2s$ for the chassis translation. However, in the state update equations, $\delta t = 4s$ because the actual chassis movement takes place for only $4s$. With this setup, we perform a set of real tracking experiments in three different environments for months with individual run lasting for 30 minutes during different times of the day: (1) A cluttered office space ($\approx 10m \times 6m$), with a lot of office desks, chairs, cabinets, and reflecting surfaces (2) A hallway ($\approx 18m$ long and $3m$ wide), with pillars as well as sharp corners, and (3) A VICON camera localization based robot experiment facility ($\approx 6m \times 6m$). For the first two environments, we use manual markings on the floor to localize both the Leader and the TrackBot. For the last environment, the VICON facility provides us with camera-based localization at millimeter scale accuracy.

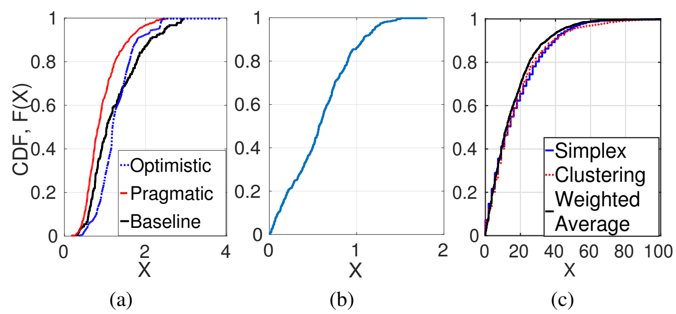


Fig. 3. Real Experiment Based Performance: (a) Absolute Distance in Meters, (b) Absolute Distance Estimation Error in Meters, and (c) Absolute Angle Estimation Error in Degrees

2) The Optimistic Strategy vs. The Pragmatic Strategy: Similar to our emulation based analysis, we perform a real system based comparison of the proposed speed adaptation strategies as well as the **Baseline Algorithm** (introduced in Section VI-A). However, in this set of experiments we do not vary the maximum speed of the TrackBot or the Leader due to prototype hardware limitations. To compare the proximity maintenance performance, we compare the absolute distance CDF statistics of these three strategies in Fig. 3a for $v_L^{max} = 10cm/s$ and $v_F^{max} = 1.8 \cdot v_L^{max}$. Figure 3a validates that Pragmatic strategy performs best among all three strategies when $v_F^{max} = 1.8 \cdot v_L^{max}$. Moreover, the baseline strategy performs the worst due to lack of speed adaptation as well as lack of history incorporation. In summary, our real experiment based results concur with the emulation results.

3) Estimation Errors: To analyze the state estimation errors in our ARREST architecture, we perform a range of prototype based experiments, where the $v_F^{max} = 1.8 \cdot v_L^{max}$ and the Leader follows a set of random paths. The empirical CDF of the absolute errors presented in Fig. 3b clearly illustrates that the instantaneous absolute errors in our distance estimates are $\leq 100cm$ with very high probability ($\approx 90\%$) and are bounded by $1.5m$. These statistics are also reasonable for pure RSSI based estimation systems and concur with the emulation results. Next, in Fig. 3c, we compare the angle estimation error performance of the TrackBot for all three AoA observation methods introduced in Section IV-B where we intentionally introduce random sparsity in the RSSI measurements. Figure 3c illustrates that our proposed *clustering method* and *weighted average method* perform significantly better than the *basic correlation method* which is expected since the first two take into account the clustered sparsity (Detailed in Section IV-B). The instantaneous absolute angle errors are less than $40°$ with high probability ($\approx 90\%$) for all three methods which is justified because the HPBW specification for the antenna is approx $70°$. Figure 3c also illustrates that the weighted angle observation method slightly outperforms the clustering method for AoA observation. The apparent similarity between the performance of the clustering method and the weighted average method is attributed to the consistent lower cluster sizes compared to the gap sizes ($\lambda_a << \mu_a$) in our experiments.

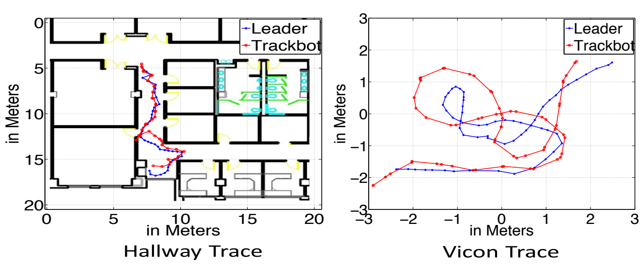


Fig. 4. Full Path Traces from Real World Experiments

4) Tracking Performance: We also perform a set of tracking experiments using our prototype in all three scenarios: indoor, hallway, and VICON system. Our experiments show that our system performs quite well in the respective scenarios and stays within $\approx 2m$ from the Leader for the duration of the experiments. In Fig. 4, we present representative traces from our real experiments in the Hallway and the VICON system. These results suggest that our system works equally well in different environments: cluttered and uncluttered. To verify that further, we perform a set of experiments with a static Leader not in the line of sight of the TrackBot for $\geq 50\%$ of the TrackBot's path. *Our TrackBot was able to find the Leader in* 75% *of such experiments.* The main reason behind this success lies in the TrackBot's ability to leverage a good multipath signal (if exists). In absence of direct line of sight, the TrackBot first follows the most promising multipath component and by doing so it eventually comes in line of sight with the Leader and follows the direct path from that point on. *In most of these experiments* ($\geq 90\%$)*, the TrackBot travels a total distance of less than 2X the distance traveled by the Leader. This implies that our system is efficient in terms of energy consumption due to robotic maneuvers.*

VII. Conclusion

In this paper, we propose ARREST, a solely RF based relative localization and tracking system, for autonomously following a RF-emitting object, and demonstrate the performance and error statistics of our system. However, there are a lot of research questions that need to be addressed in our future work. First, we would like to develop a strategy with a proper trade-off between Optimism and Pragmatism, which will potentially improve the performance. Second, we want to perform a thorough evaluation of our system via a set of large scale experiments. Lastly, we would like to explore the domains of game theory to see if better or more robust predictions of the Leader's motion could improve the performance.

References

[1] M. McClure et al. The darpa landroids program. In *SPIE Defense, Security, and Sensing, 2009*.

[2] R. R. Murphy. Trial by fire [rescue robots]. *Robotics & Automation Magazine, IEEE*, 11(3):50–61, 2004.

[3] B. Jung and G. S. Sukhatme. Detecting moving objects using a single camera on a mobile robot in an outdoor environment. In *IAS, 2004*.

[4] M. Kleinehagenbrock et al. Person tracking with a mobile robot based on multi-modal anchoring. In *IEEE RO-MAN, 2002*.

[5] G. Han et al. Localization algorithms of wireless sensor networks: a survey. *Telecommunication Systems*, 52(4):2419–2436, 2013.

[6] J. Graefenstein et al. Wireless node localization based on rssi using a rotating antenna on a mobile robot. In *IEEE WPNC, 2009*.

[7] P. Tokekar et al. Active target localization for bearing based robotic telemetry. In *IEEE/RSJ IROS, 2011*.

[8] J. N. Twigg et al. Rss gradient-assisted frontier exploration and radio source localization. In *IEEE ICRA, 2012*.

[9] S. Zickler and M. Veloso. Rss-based relative localization and tethering for moving robots in unknown environments. In *IEEE ICRA, 2010*.

[10] L. Oliveira et al. Rssi-based relative localisation for mobile robots. *Ad Hoc Networks*, 13:321–335, 2014.

[11] D. Vasisht et al. Decimeter-level localization with a single wifi access point. In *USENIX NSDI, 2016*.

[12] J. Elson. Simulation of constant-distance robot following using radio proximity beacons. http://www.circlemud.org/jelson/writings/constant-following/constant-following.html, 1999.

[13] B. Min and E. T. Matson. Robotic follower system using bearing-only tracking with directional antennas. In *Robot Intelligence Technology and Applications 2*, pages 37–58. Springer, 2014.

[14] D. P. Bertsekas. *Dynamic programming and optimal control*, volume 1 & 2. Athena Scientific Belmont, MA, 1995.

[15] J. Van Den Berg et al. Lqg-obstacles: Feedback control with collision avoidance for mobile robots with motion and sensing uncertainty. In *IEEE ICRA, 2012*.

[16] J. Tornero et al. Multirate lqg controller applied to self-location and path-tracking in mobile robots. In *IEEE/RSJ IROS, 2001*.

[17] M. Athans. The role and use of the stochastic linear-quadratic-gaussian problem in control system design. *IEEE TAC*, 16(6):529–552, 1971.

[18] T. S Rappaport. *Wireless communications: principles and practice*. prentice hall PTR New Jersey, 1996.

[19] P. Ghosh et al. Arrest: A rssi based approach for mobile sensing and tracking of a moving object. *CsRo*, arXiv:1707.05493, 2017.

[20] MBED-OS. https://www.mbed.com/en/platform/mbed-os/, 2017.

[21] Openmote. http://www.openmote.com/, 2017.

[22] mbed LPC1768. https://developer.mbed.org/platforms/, 2017.

[23] E. Baccelli et al. Riot os: Towards an os for the internet of things. In *IEEE INFOCOM Workshops, 2013*.