GLAC: High-Precision Tracking of Mobile Objects With COTS RFID Systems

Wei Gong, *Senior Member, IEEE*, Haoyu Wang, Siyi Li, and Si Chen, *Student Member, IEEE*

***Abstract*— This paper presents GLAC, the first 3D localization system that enables millimeter-level object manipulation for robotics using only COTS RFID devices. The key insight of GLAC is that mobility reduces ambiguity (One-to-many mapping relationship between phase and distance) and thus improves accuracy. Unlike state-of-the-art systems that require extra time or hardware to boost performance, it draws the power of modeling mobility in a delicate way. In particular, we build a novel framework for real-time tracking using the Hidden Markov Model (HMM). In our framework, multiple Kalman filters are designed to take a single phase observation for updating mobility states, and a fast inference algorithm is proposed to efficiently process an exponentially large number of candidate trajectories. We prototype GLAC with only UHF tags and a commercial reader of four antennas. Comprehensive experiments show that the median position accuracies of x/y/z dimensions are within 1 cm for both LoS and NLoS cases. The median position accuracy for slow-moving targets is 0.41 cm, which is 2.2×, 17.3×, and 14.9× better than TurboTrack, Tagoram, and RF-IDraw, respectively. Also, its median velocity accuracy is at least 20× better than all three competitors for fast-moving targets. Besides accuracy, it achieves more than 4× localization time gains over state-of-the-art systems.**

***Index Terms*— RFID, real-time tracking, high speed.**

I. Introduction

ROBOTICS has been evolving significantly over the past decades, becoming more capable, agile, and precise [1], [2], [3], [4]. This robot precision is the key enabler to future applications. Different from earlier industrial automation where many repetitive tasks, e.g., car-door painting, do not require high precision, modern robots nowadays are integrating motion control, actuator, and other advanced technologies to complete a wide range of tasks [1], [5], [6], [7], from welding to surgery. For example, recent studies show that delicate eye surgery can be done with a robot that is ten times more precise than a human surgeon [8].

There is a lot of interest in robotics academia and industry in the hope that robots can perform more flexible and complex tasks, such as assembling products like a decorator [9], providing intelligent pickup and delivery services in the kitchen [10] or somewhere else, and multiple robots working together. For instance, we hope to enable robots to automatically assemble parts of finished devices, so that robots can competently perform in-home assembly operations in retail stores.

Laser and vision-based solutions have been the mainstream to deliver robot precision [11], [12], [13]. The most widely used clinical robotic surgical system, da Vinci, has mechanical arms with surgical instruments and a camera arm that provides a high-definition, magnified, 3D view of the surgical site [14]. Such systems, however, have two main drawbacks: high cost that prohibits small and medium enterprises adoption, e.g., an infrared VICON motion capture system is priced at hundreds of thousands of dollars [15]; and poor handling of occlusion that is the fundamental weakness for visions. As a result, robotics has been looking for real-time localization solutions that are cost-effective, highly accurate, and capable of dealing with occlusion.

To address aforementioned issues, the wireless community has made much effort in RFID localization systems, which offer an appealing alternative as it can achieve high-precision, auto-identification, and working with Non-Line-of-Sight (NLoS) cases [15], [16], [17], [18], [19], [20]. Table I lists the key features of most state-of-the-art systems. Through a careful study, we find that prior arts focus on putting in either more money, e.g., expensive and powerful equipment, or more time by requiring heavy computation or restrictive mobility. For example, RFind [17] is quite resource-intensive; it requires four N210 USRPs (amounting to $ 8000), more than 66 packets to derive a position estimate (translating to a reading delay of seconds), and a bandwidth of 200 MHz wide (10 times of a typical WiFi channel). Tagoram [20], which needs only a single COTS reader, takes several seconds computation time to localize once, making it unfit for real-time tracking. RF-IDraw [18], which has moderate money and time cost, achieves an unsatisfactory median position accuracy, 19 cm; thus it is only suitable for applications that only require shape fidelity, like drawing, as opposed to precise location control, such as robot-assisted surgery. In short, how to build a real-time localization system for precise robot control with COTS RFID equipment remains a big challenge.

In this paper, we present GLAC, the first 3D localization system that enables millimeter-level object manipulation for robotics using only COTS devices. Compared to prior arts, it achieves the most accurate localization while consuming minimal resources in terms of cost-effectiveness and

Manuscript received 28 August 2021; revised 4 November 2022 and 18 October 2023; accepted 27 December 2023; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor C. Petrioli. This work was supported by NSFC under Grant 61971390 and Grant 61932017. *(Corresponding author: Si Chen.)*

Wei Gong and Siyi Li are with the School of Computer Science and Technology, University of Science and Technology of China, Hefei 230026, China (e-mail: weigong@ustc.edu.cn; kd19030926@mail.ustc.edu.cn).

Haoyu Wang is with the School of Software, Tsinghua University, Beijing 100190, China (e-mail: wanghy20@mails.tsinghua.edu.cn).

Si Chen is with the School of Computing Science, Simon Fraser University, Burnaby, BC V5A 1S6, Canada (e-mail: sca228@sfu.ca).

Digital Object Identifier 10.1109/TNET.2023.3348950

TABLE I
COMPARISON OF STATE-OF-THE-ART RFID LOCALIZATION SYSTEMS

	How much we pay				How well we do	
	Cost ($)	Computation	Reading [1]	Bandwidth	Position Accuracy [2]	Dim.
GLAC	$\sim$ 1,000 (1 COTS reader)	$\sim \mu s$	1 pkt	0.25 MHz	sub-centimeter	3D
TurboTrack [16]	$\sim$ 20,000 (3×X310 & 1×N210) [3]	$\sim \mu s$	4 pkts	100 MHz	sub-centimeter	3D
RFind [17]	$\sim$ 8000 (4× N210)	$\sim \mu s$	$>$ 66 pkts	200 MHz	1.1 cm	3D
RF-IDraw [18]	$\sim$ 2000 (2 COTS reader)	$\sim \mu s$	8 pkts	0.25 MHz	19 cm	2D
RF-Compass [15]	$\sim$ 2000 (1× N210)	$\sim \mu s$	60 pkts	0.25 MHz	2.8 cm	3D
PinIt [19]	$\sim$ 2000 (1× N210)	$\sim \mu s$	480 pkts	0.25 MHz	11.2 cm	2D
MobiTagBot [21]	$\sim$ 1000 (1 COTS reader)	$\sim \mu s$	10 pkts	2.5 MHz [4]	2.8 cm	1D
Tagoram [20]	$\sim$ 1000 (1 COTS reader)	$\sim s$	4 pkts	0.25 MHz	12.3 cm	2D

[1] The minimal no. of packets needed to derive a location estimate.
[2] The median accuracy when a tag moves at a velocity of less than 10 cm/s.
[3] The extra cost associated with SDRs involves formal FCC certifications, which is necessary to turn prototypes into commercial RFID devices.
[4] The original paper recommends hopping more than 10 channels for better performance.

time-efficiency. Unlike previous systems that mainly employ more physical resources or assumptions, we take a completely distinct approach, which is to draw power from mobility itself. Our key insight is that ambiguity may hurt once, but not forever. In this context, ambiguity refers mainly to the effect of phase periodicity on actual distance measurements, which is the main source of error composition. In other words, while differentiating ambiguity at a time may be difficult due to insufficient information, it becomes more tractable when we build a network of candidate trajectories containing all the possible ambiguities. We observe that there is only one candidate trajectory that most closely matches the ground truth if the object's mobility is reasonably modeled. This scenario is just like when we put a great number of gladiators into a fight arena, the final survivor is the strongest one. That's why we call it GLAC (**GL**adiator tr**AC**king).

We prototype GLAC with only UHF tags and a commercial reader with four antennas. The main results of our extensive experiments are summarized as follows.

- The median position accuracies of x/y/z dimensions are within 1 cm for both LoS and NLoS cases.
- In slow-moving scenarios, the median position accuracy is 0.41 cm, which is 2.2×, 17.3×, and 14.9× better than TurboTrack, Tagoram, and RF-IDraw, respectively, and those gaps enlarge to 21.8×, 21.7×, and 32.6× for fast-moving scenarios.
- The median velocity accuracy is 2.26 cm/s for slow-moving cases, which is 5.3×, 7.6×, and 7.2× better than TurboTrack, Tagoram, and RF-IDraw, respectively, and those differences grow to more than 20× for all three competitors when the target is fast-moving.
- As to localization time-efficiency, it achieves overall time gains of 4× over TurboTrack, 72× over Tagoram, and 4× over RF-IDraw.
- GLAC-Plus's combined percentage error of position and velocity is 4.7× better than GLAC for high-acceleration cases.

Contributions: We make the following contributions:

- GLAC is the first mobile 3D localization system that supports robot object manipulation using only COTS devices. It enables robots to locate an object with an accuracy of millimeter-level and a timescale of tens of milliseconds. By doing so, it provides a practical tracking solution for robotics, which is cost-effective, highly accurate, and real-time even with NLoS.
- GLAC presents a novel RF-tracking framework formulated using the HMM. It makes three key differences: 1) it leverages the object's mobility for disambiguation; 2) it can estimate the object's velocity in addition to its location using only a single phase measurement; 3) it employs a fast inference algorithm that uses nearest neighbor pruning and solid geometry.
- A real-time prototype of GLAC is implemented, and extensive real-world experiments show its capability of millimeter-level tracking in real-time. As GLAC requires only COTS gadgets, billions of deployed commercial tags and readers are ready to benefit a wide range of agile robot applications.
- To further contribute to the community and support reproducibility, we have opened source the code and experimental data of our implementation on GitHub [22].

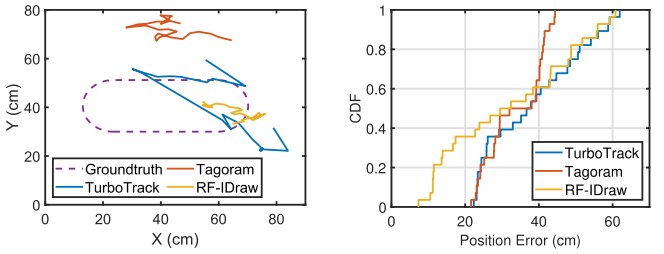
II. MOTIVATION

To begin with, we investigate how prior systems perform under a standard test, which can help us gain deep insights about what the essentials are for high-precision tracking. It goes as follows. We employ a COTS reader with 4 antennas to continuously localize a tag moving at 40 cm/s along a predefined track. The systems are reproduced faithfully,[1] include RF-IDraw [18], Tagoram [20], and TurboTrack [16].

Results are shown in Figure 1a. We observe that all three state-of-the-art systems perform poorly as the resulted trajectories are far from the ground truth. After diving deeper into tracking processes, we find two main factors are contributing to this.

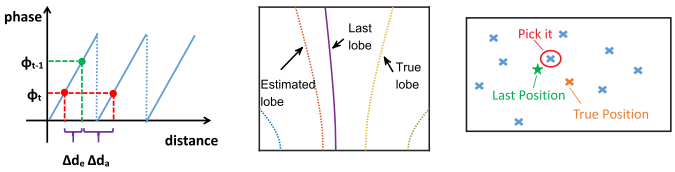
Oversimplified mobility models. Even though all three systems aim for tracking, they do not draw sufficient attention to mobility. For example, Tagoram fits the trajectory using the estimated distance difference that is always less than $\frac{\lambda}{4}$. This strategy may lead to significant errors when the tag moves

[1]For details including settings, parameters and detailed results, please refer to Section V.



(a) Trajectories at 40 cm/s (b) CDF comparison

Fig. 1. Comparison using 1 COTS reader of 4 antennas. (a) When the tag's velocity is 40 cm/s, the trajectories of all three state-of-the-art systems are far from the ground truth. (b) shows they are all unable to deliver high-precision tracking with COTS devices.



(a) Tagoram (b) RF-IDraw (c) TurboTrack

Fig. 2. Oversimplified mobility assumptions cause tracking failures for all three systems.

fast. As shown in Figure 2a, the estimated distance difference is Δd_e as it is less than $\frac{\lambda}{4}$, which is wrong since the actual distance difference is Δd_a. Similar things happen to RF-IDraw and TurboTrack as well, as shown in Figure 2b and 2c. As RF-IDraw tracks the grating lobe that is the closest to the last lobe and TurboTrack looks for the intersection point that is the closest to the particle position of the previous round, both could result in wrong picks due to the tag's fast movement. In short, mobility has to be properly modeled to deal with fast-moving objects.

Low-rate asynchronous reading. One may wonder why TurboTrack looks so bad in our test, far from that stated in the original paper [16]. The answer is COTS readers interrogate tags at a median rate of 30 frames/s [20] whereas SDR readers of [16] work at 300 frames/s. Such a low-rate reading further necessitates mobility modeling. In addition to that, differing from SDR readers that can receive backscattered signals simultaneously across multiple antennas, COTS readers only support asynchronous reading for multiple antennas, i.e., one antenna at a time. Suppose that the delay of two successive asynchronous readings is 30 ms,[2] and the antenna reading order is 1-2-3-4. Then the delay between the 1st antenna and the 4th would translate to a position error of 3.6 cm when the tag moves at 40 cm/s, which may lead to even greater accumulated errors. Simply put, all three state-of-the-art systems are not well prepared for low-rate asynchronous readings.

III. GLAC Design

Turning the **GL**adiator tr**AC**king idea into a system, however, faces three critical challenges.

1) How to reduce ambiguities with constrained COTS devices? The basic principle of resolving ambiguities is to acquire more information. Nevertheless, the intrinsic low reading rate of COTS RFID devices cannot meet the demand for high-precision. As a result, many prior systems attempt to overcome this barrier from different perspectives. TurboTrack employs extra localization helpers (wideband USRPs) that can achieve reading rates of upto 300 frames/s [16]. RF-Compass [15], PinIt [19], and MobiTagBot [21] require to collect a number of packets on the scale of several seconds. RF-IDraw draws information from a deliberately designed antenna placement with 8 antennas using plane geometry. On the contrary, we resort to mobility modeling since our goal is to drive the development of COTS RFID localization with no strings attached. Specifically, we build a novel framework of localization using the HMM where spatial states at a time are discrete and those states over time are continuous. This way, RFID localization is translated into an HMM inference problem. As more observations come in, the likelihood of identifying the optimal trajectory (hidden states) increases, i.e., mobility over time reduces ambiguities. Further, to solve the low-rate COTS reading problem, we design Kalman filters that take a single phase for updating as opposed to multi-phase feeding in all prior arts.

2) How to achieve real-time localization efficiency? The localization time involves reading time and computation time. While our HMM framework is designed to accommodate single-phase readings, which achieves minimal communication overhead, its computation overhead is prohibitive because the number of candidate trajectories increases exponentially over time. To address this, we employ a fast inference scheme that includes nearest neighbor pruning and initial states pruning techniques. Such an expedited scheme ensures that the number of candidate trajectories does not grow over time. Given fully optimized reading and computation time, GLAC can deliver accurate localization with high time-efficiency.

3) How to design suitable mobility model? We'll see that in more details later, considering the varieties of motion patterns, GLAC and its supplement, GLAC-Plus, will adopt different degrees of approximation to simplify the movements and, meanwhile, maintain high preciseness.

Next, we will first formally formulate the problem and present the solution framework, then propose a fast inference that achieves real-time efficiency.

A. HMM Tracking Framework

Almost all COTS readers can output backscatter phase readings, which is a measure of the range between the reader and tag expressed in units of cycles of the carrier frequency. While this phase measurement can be made with very high-precision, the whole number of cycles is not measurable. Given the distance between the reader and tag, z, the measured phase, ϕ, follows,

$$\phi = \frac{2\pi z}{\lambda} + \delta \mod 2\pi, \tag{1}$$

where λ is the wave length, δ is the noise. Since phase is periodic with a period of 2π radians, the phase values will repeat at distances separated by integer multiples of

[2]The median value measured from ImpinJ R420 and Thingmagic M6e.

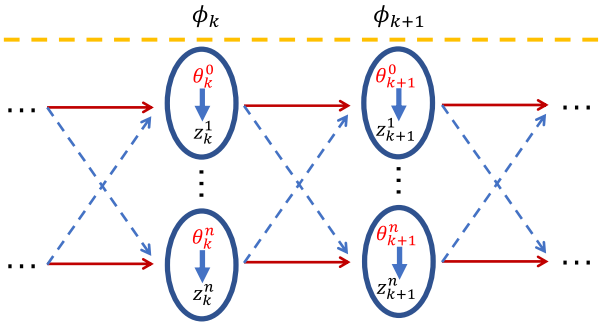


Fig. 3. HMM framework where multiple Kalman filters are used to estimate hidden states from phase observations.

one-half wavelength,[3] $z^n = \phi + \frac{n\lambda}{2}, n = 0, 1, 2 \ldots$.[4] We can imagine phase as a measuring tape extending from the reader to the tag that has numbered markers every one millimeter. Unfortunately, however, the numbering scheme returns to zero with every one-half wavelength. To remove such ambiguities, many previous systems make various attempts including moving antennas [19], [21], virtual antennas [20], and dedicated localization helpers [16], [17]. Instead, we choose to model mobility into an HMM based on a key observation; phases are accurate and all we need to do is to reduce ambiguities.

The basic RFID localization problem is that given a time series of phase measurements (ϕ_0,ϕ_2,...,ϕ_κ), find the most probable location series (P_0, P_2,...,P_κ). We translate this process into an HMM, as shown in Figure 3. First, we model the mobility status of each location as $\theta_k = (P_{x,k}, P_{y,k}, V_{x,k}, V_{y,k})^T$, where $P_{x,k}$ and $P_{y,k}$ are x-y coordinates of the location at time k, and $V_{x,k}$ and $V_{y,k}$ are corresponding velocities. Second, we extend phase observation ϕ_k to separating distance observations (ambiguities) $Z_k = (z_k^0, \ldots, z_k^n)$. Accordingly, state θ_k is discretized into $\theta_k^0, \ldots, \theta_k^n$, each of which is related to the corresponding distance observation. Hence, the original problem becomes given a set of distance observations (Z_0, Z_1, ..., Z_κ), find the most probable states ($\hat{\theta_0}$, $\hat{\theta_1}$, ..., $\hat{\theta_\kappa}$). Simply put,

$$\hat{\theta_{0:\kappa}} = \arg\max_{\theta_{0:\kappa}} p(\theta_{0:\kappa}|Z_{0:\kappa}). \quad (2)$$

Apparently, there are n^κ possible trajectories with n^κ observation sequences. To find the optimal trajectory, the simplest way is to compute the likelihood of the observation sequence for each trajectory. Then we choose the trajectory with the maximum observation likelihood among all possibilities.

To do so, we need to estimate all the hidden states and this is where Kalman filters come in [23]. From a high level, we put a Kalman filter for each trajectory and estimate its mobility statuses continuously. Before state estimation, we find there are two important parts missing, state transition probabilities and observation likelihoods.

[3]For many readers that introduce additional π ambiguity for phase values, separating distances become $\frac{n\lambda}{4}$.

[4]n can be set based on the maximal reading range.

Transition equation. As a general time-varying system, we approximate state transition as a Gaussian process,

$$\theta_{\mathbf{k+1}} = \mathbf{A}\theta_{\mathbf{k}} + \mathbf{s_k} = \begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \theta_{\mathbf{k}} + \mathbf{s_k} \quad (3)$$

where Δt is the time difference, $\mathbf{s_k}$ is the system-state Gaussian noise, $\mathbf{A}$ is the system matrix depicting mobility. Both $\theta_{\mathbf{k}}$ and $\mathbf{s_k}$ follow Gaussian distribution, i.e., $\theta_{\mathbf{k}} \sim \mathcal{N}(\mathbf{m_k}, \mathbf{\Sigma_k})$, $\mathbf{s_k} \sim \mathcal{N}(0, \mathbf{Q_k})$, where $\mathbf{Q_k}$ is the variance of the Gaussian distribution, which can be used to control the upper bound of predictions.

Observation equation. The observed distance, $\mathbf{z_k}$, can be modeled as two points in Euclidean space as follows,[5]

$$\mathbf{z_k} = (\sqrt{(P_{x,k} - A_{x,i})^2 + (P_{y,k} - A_{y,i})^2}) + \mathbf{u_k}. \quad (4)$$

where $A_{x,i}$ and $A_{y,i}$ are the positions of the antenna i that reports ϕ_k, and $\mathbf{u_k} \sim \mathcal{N}(\mathbf{0}, \mathbf{R_k})$ is Gaussian noise.

State estimation. When transition and observation equations are in place, Kalman filters can be invoked iteratively to do state estimation. Note that due to the nonlinearity of our observation equation, the Extended Kalman Filter [24] (EKF) is needed. Then the observation equation becomes

$$\mathbf{z_k} = \mathbf{C}\theta_{\mathbf{k}} + \mathbf{u_k}, \quad (5)$$

where $\mathbf{C_k} = (\frac{\partial \mathbf{z_k}}{\partial P_{x,k}}, \frac{\partial \mathbf{z_k}}{\partial P_{y,k}}, \frac{\partial \mathbf{z_k}}{\partial V_{x,k}}, \frac{\partial \mathbf{z_k}}{\partial V_{y,k}})$, is the observation matrix. Then the Kalman filter is able to predict the state at time $k+1$ using the transition equation,

$$\theta'_{\mathbf{k+1}} = \mathbf{A}\theta_{\mathbf{k}} \quad (6)$$

$$\mathbf{\Sigma'_{k+1}} = \mathbf{A}\mathbf{\Sigma_k}\mathbf{A^T} + \mathbf{Q} \quad (7)$$

Afterwards, an update has to be done by combing observation equation and predicted $\theta'_{\mathbf{k+1}}$ as

$$\theta_{\mathbf{k+1}} = \theta'_{\mathbf{k+1}} + \mathbf{K_{k+1}}(\mathbf{z_{k+1}} - \mathbf{C_{k+1}}\theta'_{\mathbf{k+1}}) \quad (8)$$

$$\mathbf{\Sigma_{k+1}} = (\mathbf{I} - \mathbf{K_{k+1}}\mathbf{C_{k+1}})\mathbf{\Sigma'_{k+1}}, \quad (9)$$

where $\mathbf{K_{k+1}} = \mathbf{\Sigma'_{k+1}}\mathbf{C^T_{k+1}}(\mathbf{C_{k+1}}\Sigma'_k\mathbf{C^T_{k+1}} + \mathbf{R_{k+1}})^{-1}$ is the Kalman gain that balances transitions and observations.

The above four equations are for one round. Iterative executions for a candidate trajectory from the beginning to the end and then for all the candidate trajectories make all the hidden states properly estimated. There are several distinct features of our HMM framework with multiple Kalman filters worth mentioning.

- As velocities are estimated together with each position, position estimates that do not match velocity history will be devalued significantly by the Kalman gain, reducing ambiguities.
- To estimate the next state, we only need a single phase measurement, which accommodates the asynchronous reading mode and differs from all previous approaches [15], [16], [17], [18], [19], [20].
- The Gaussian mixture model used by previous systems [16], [20] is no longer needed as we already

[5]We describe it in 2D for simplicity.

discretized it into multiple separate Gaussian models for Z_k.

B. Fast Inference Using Nearest Neighbor

While the aforementioned HMM framework seems settled, it has a big drawback, high complexity. Specifically, as more phase readings come in, the candidate trajectories increase exponentially, which conflicts the goal of real-time tracking. As the reading time is already minimized by using only single-phase updates, we design two techniques to achieve computation efficiency, nearest neighbor pruning and initial state pruning.

1) Nearest Neighbor Pruning: The basic idea of nearest neighbor pruning is to reduce the number of candidate trajectories by only connecting the nearest distance observation. It works as follows. Suppose the state at time k of one candidate trajectory is $\theta^{\mathbf{i}}_{\mathbf{k}}$, then at time $k+1$, this state is only transit to state $\theta^{*}_{\mathbf{k+1}}$ whose observation distance $\mathbf{z}^{*}_{\mathbf{k+1}}$ is the nearest to the predicted state $\theta'_{\mathbf{k+1}}$. In particular, the pick of $\mathbf{z}^{*}_{\mathbf{k+1}}$ follows

$$\mathbf{z_{k+1}} = \frac{\phi_{k+1}}{2\pi}\lambda + \frac{\lambda}{2}\arg\min_{i\in Z}|\frac{\phi_{k+1}+i\pi}{2\pi}\lambda - d_{k+1}|, \quad (10)$$

where d_{k+1} is the distance between $(P'_{x,k+1}, P'_{y,k+1})$ and antenna i that reports phase at time $k+1$.

Apparently, our nearest neighbor pruning is a greedy algorithm, which may achieve local optimum. But such a sacrifice is worth it because it ensures that the number of candidate trajectories is not growing over time, achieving great time efficiency. While one may think our nearest neighbor strategy is similar to the pitfalls of previous systems shown in Section II, it has at least two major differences. First, it does not pose any limit on location and velocity estimation, which is different from that prior systems explicitly or implicitly set mobility constraints through location estimates. Second, even if sometimes the nearest neighbor strategy chooses the wrong candidate, it could be offset somehow by our observation and system mobility matrices through Kalman gains, whereas prior systems do not have such correction.

2) Optimal Trajectory: After initial state pruning shown in Appendix B, we finally come to fast compute the likelihood of pruned trajectories by iteratively propagating beliefs.

Likelihood at initial time. Assume we have interpolated phases $\hat{\phi_{i,0}}, 1 \le i \le M$. The likelihood at initial time is

$$\mathbf{L}_0 = \prod_{i=1}^{M} P(\theta_{\mathbf{0}}|\hat{\phi_{i,0}}) \propto \prod_{i=1}^{M} P(\hat{\phi_{i,0}}|\theta_{\mathbf{0}}) = \prod_{i=1}^{M} \mathcal{F}(d_0^i; z_0^*, R_0), \quad (11)$$

where d_0^i is the distance between antenna i and initial position $(P_{x,0}, P_{y,0})$, z_0^* is the candidate distance closest to $(P_{x,0}, P_{y,0})$, and $\mathcal{F}(x;\mu,R)$ is a Gaussian probability density function with a mean of μ and a variance of R. If we assign weights equally across all interpolated phases, $\mathbf{L}_0$ can be approximated as $\prod_{i=1}^{M}\mathcal{F}(d_0^i; z_0^*, R_0)$ because we normalize all the likelihoods of trajectories at every time snapshot.

Likelihood update at time $k+1$. Following a trajectory, if we have the trajectory's likelihood at time k, its likelihood the time $k+1$ can be derived from the transition and observe equations as

$$\mathbf{l}_{k+1} = \mathbf{l}_k p(\mathbf{z}^{*}_{\mathbf{k+1}}|\theta_{\mathbf{k+1}}) p(\theta_{\mathbf{k+1}}|\theta_{\mathbf{k}}), \quad (12)$$

$$p(\mathbf{z}^{*}_{\mathbf{k+1}}|\theta_{\mathbf{k+1}}) = \mathcal{F}(d_{k+1}; \mathbf{z}^{*}_{\mathbf{k+1}}, \mathbf{R_{k+1}}), \quad (13)$$

$$p(\theta_{\mathbf{k+1}}|\theta_{\mathbf{k}}) = \mathcal{F}(\theta_{\mathbf{k+1}}; \theta_{\mathbf{k}}, \mathbf{Q_{k+1}}), \quad (14)$$

where $\mathcal{F}$ is a multidimensional Gaussian probability density function. Similarly, all the likelihood of trajectories are normalized for time $k+1$.

C. Putting It All Together

We include an illustrative figure in Appendix A and summarize the major steps of GLAC as follows.

1) Initially, we interpolate phases and use initial state pruning to obtain a handful of quality initial positions and velocities.
2) For each candidate trajectory during an update, we use nearest neighbor pruning to select the next distance observation and employ an EKF to estimate hidden mobility states and the likelihood of the trajectory.
3) We normalize likelihoods among trajectories at every time snapshot and the trajectory of the highest likelihood is chosen as the optimal one.

The algorithm for running multiple EKF instances in parallel has also been presented in [25] for mobile robot tracking. When selecting the final estimated trajectory, we have a distinction where [25] will take as total estimate at time t the average of all the EKF instances, while GLAC choose the trajectory of the highest likelihood as the optimal one.

D. GLAC-Plus

When dealing with the **accelerating objects**, GLAC still has distinct advantages. However, we suspect that the weakness will show up at low data frame rates on account of two main reasons which can be summarized as **low-rate loss**. In terms of this issue, acceleration modeling is proposed to accommodate scenes with both accelerating motions and restricted rates.

1) Low Rate Loss:

- **Data Poor.** Obviously, a discrete sequence of kinestates implies infinite trajectories. The lower the sampling rate is, the greater the ambiguity will be between each data point, which is an inherent flaw in this system.
- **Simplified Motion Model.** We assume that the displacement is continuously differentiable as a function of time. The velocity modeling essentially represents a first-order approximation of Taylor series as follows,[6]

$$\mathbf{x(t)} = \frac{\mathbf{x(t_0)}}{\mathbf{0!}} + \frac{\mathbf{x'(t_0)}}{\mathbf{1!}}(\mathbf{t}-\mathbf{t_0}) + \frac{\mathbf{x''(t_0)}}{\mathbf{2!}}(\mathbf{t}-\mathbf{t_0})^{\mathbf{2}} + \dots \quad (15)$$

As sampling rates reduce, time interval, $\mathbf{t}-\mathbf{t_0}$, turns larger, hence the need to employ higher order estimates. In acceleration modeling, we're introducing the second order.

[6]We illustrate it in 1D.

2) Acceleration Modeling & Differences: On the basis of the original mobility status as $\theta_k = (P_{x,k}, P_{y,k}, V_{x,k}, V_{y,k})^T$, we add two extra dimensions like

$$\Theta_k = (P_{x,k}, P_{y,k}, V_{x,k}, V_{y,k}, A_{x,k}, A_{y,k})^T$$

to acquire more precise assessment. With more specific details, EKF iterations could improve the overall fitting of the motion state involving location and speed. The modules update accordingly as follow:

1) **Transition equation.** The second order term of the motion equation is introduced. $\mathbf{S_k}$ has merely expanded on the dimensionality of $\mathbf{s_k}$ as previously declared.

$$\mathbf{\Theta_{k+1}} = \mathbf{A_{6,6}\Theta_k} + \mathbf{S_k} \tag{16}$$

$$= \begin{pmatrix} 1 & 0 & \Delta t & 0 & \frac{\Delta t^2}{2} & 0 \\ 0 & 1 & 0 & \Delta t & 0 & \frac{\Delta t^2}{2} \\ 0 & 0 & 1 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 1 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \mathbf{\Theta_k} + \mathbf{S_k} \tag{17}$$

2) **Observation equation.** Equation 5 indicates that $\mathbf{C_k}$ should be updated to

$$\mathbf{C_k} = (\frac{\partial \mathbf{z_k}}{\partial P_{x,k}}, \frac{\partial \mathbf{z_k}}{\partial P_{y,k}}, \frac{\partial \mathbf{z_k}}{\partial V_{x,k}}, \frac{\partial \mathbf{z_k}}{\partial V_{y,k}}, \frac{\partial \mathbf{z_k}}{\partial A_{x,k}}, \frac{\partial \mathbf{z_k}}{\partial A_{y,k}}),$$

3) **Initial acceleration.** Since we have calculated the initial position and velocity with two data points each antenna, just one more data point each is adequate to measure the initial acceleration.

4) **Likelihood computation.** Similar to Equation 14, $\mathcal{F}$ is a multidimensional Gaussian probability density function which includes the dimensions of acceleration. In addition, the distribution of state quantity can be set freely according to the specific motion mode.

IV. Implementation

Equipment. We build a prototype using a Thingmagic M6e reader [26] with 4 omnidirectional antennas, which are placed on a plane at (0,0), (0,30), (0,80) and (80,80).[7] The antenna belongs to the monopole antennas, which usually have a Gain of up to 5 dBi and a Half Power Beam Width distribution of $45° \times 360°$ respectively in the vertical and horizontal planes. Target objects are attached with multiple kinds of tags, including Alien 9640 [27], ImpinJ Monza 4D [28], and SMARTRAC DogBone [29]. The whole system contains a 800-dollar reader, 4 antennas priced at $4 each and several tags with a price of about 10 cents. For 2D-tracking, we use a C1-intelligent car, which is remotely controlled through wireless in such a way that it can handle straight-line tracking at different speeds (10 cm/s to 40 cm/s), whereas for 3D-tracking, a VANBOT 4DoF robotic arm [30] are tested, as shown in Figure 8. The VANBOT digital servo [31] driving the robot arm has an adjustable torque of 12-20 kg·cm and a rotation speed of up to 0.16 sec/60°.

[7]This is not mandatory as our solution supports arbitrary placements and the implicit coordinate system is in centimeters.

The localization software is programmed using Java and Mercury APIs [33], which collect phase readings from the reader to a laptop. Two threads are involved; one for GUI and the other for trajectory updates when a new phase reading comes in. The whole system works in real-time.

Handling Phase Offset. The phase model (Equation 1) omits an important contribution to the phase data which is the circuitry and tag offset, which differs from tag to tag and from reading antenna to reading antenna. Some calibrations mentioned in [34] and [35] on the phase seem inevitable. However, in our previous work [36], we have measured the phase offset of the same antennas used in this paper.

The specific procedure is like this: Firstly, our experiment confirms the linear relationship of Equation 18, where ϕ_0 is the phase offset introduced by hardware interface. We fix the position of the antenna, move the tag from 10 cm to 100 cm away, and measurement is performed every 5 cm. Every measurement lasts 30 seconds and the mean phase is recorded. Then, we unwrap every phase by subtracting an integer multiple of π. The unwrapped phase has good linearity, so we linearly fit it. The fitting line is very close to the theoretical line, signifying that phase is a good measure of distance.

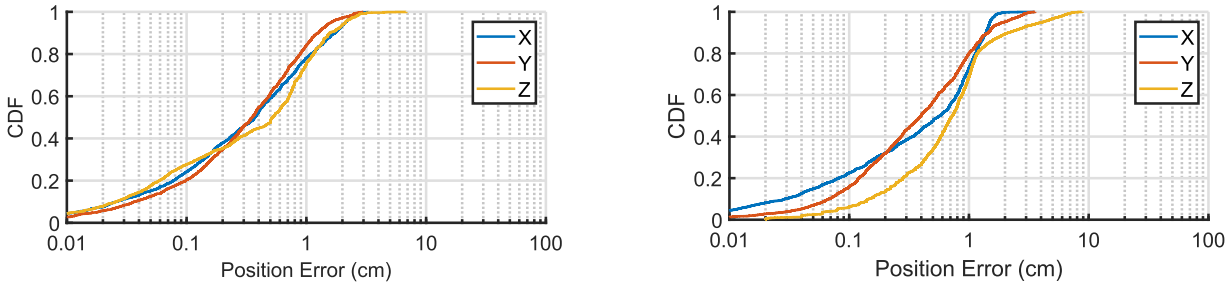
Secondly, a static calibration can be used to solve this problem. There is a linear relationship between phase and distance, and the slope is known. Therefore, only one equation is needed to solve the line. Put the tag stationarily at a known place and measure the mean phase for every antenna, we can easily get the phase offsets of all antennas. Fortunately, if we don't change the antenna setup, calibration should only be done once.

$$\phi = \frac{2\pi z}{\lambda} + \phi_0 + \delta \mod 2\pi, \tag{18}$$

Experimental Environment. To obtain ground truth, we use an OptiTrack system [37], which costs $200,000. It is a vision-based camera-array system that can capture 3D-motions with 0.1 mm accuracy. It also has 360 fps and 2.8 ms latency, which qualifies for position ground truth. The experiments are conducted in various indoor environments, including classrooms with tables and chairs, library with benches and bookshelves, and lab environments full of work-counters and computers. Both LoS and NLoS tests are performed. Similar to past works [16], [18], the NLoS scenarios are built with putting separators to block the LoS between the antenna and the RFID tag, and since OptiTrack can only work in LoS, we let the tracking marker remain in LoS for cameras while keeping the tag in NLoS for antennas.

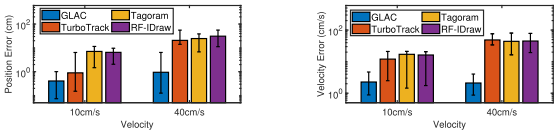
Competition. State-of-the-art competitors includes:

- **TurboTrack** [16] represents an high-precision tracking system achieving sub-centimeter accuracy for 3D. For fair-comparison purposes, TurboTrack here is reproduced with COTS devices, instead of SDRs. Also, one-shot wideband estimation is not implemented as it is not compatible with COTS readers. All the rest modules are faithfully reproduced.



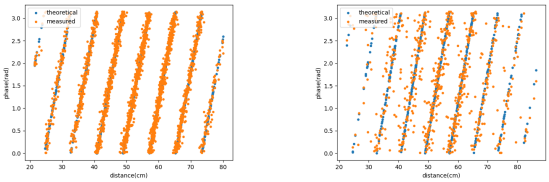
(a) LoS (b) NLoS

Fig. 4. CDFs of GLAC's position errors in x/y/z dimensions for LoS and NLoS scenarios, where the cumulative distribution function (CDF) of a real-valued random variable X, or just distribution function of X, evaluated at x, is the probability (replaced by frequency) that X will take a value less than or equal to x. [32].



(a) Position error (b) Velocity error

Fig. 5. Position and velocity errors of slow-moving (10 cm/s) and fast-moving (40 cm/s) motions for GLAC, TurboTrack, Tagoram, and RF-IDraw. Error bars indicate the 10th and 90th percentiles.



(a) velocity = 10cm/s (b) velocity = 40cm/s

Fig. 6. The relationship between the phase values of the tag and the distance from the tag to the antennae during motion at different speeds, comparing theoretical values with measured values.

- **RF-IDraw** [18] is a representative of accurate shape-tracking system. But the original proposal requires 8 antennas and 2 readers, which is far more than other competitors. So for fair competition, we adopt a 4-antenna version [38] that keeps all of the essential components and only differs in antenna placement.
- **Tagoram** [20] stands for decimeter-level tracking using COTS RFIDs. It can track only in 2D as its computation overhead is overwhelming for 3D. Our implementation set the resolution to 1000×1000. Note that the known-track version in the original paper is not suitable for robotic tracking and thus not implemented.
- **RFind [17], RF-Compass [15] and PinIt [19]** are not presented as benchmarks mainly because their USRP implementations are difficult to migrate to COTS devices.
- **MobiTagBot** [21] is not adopted since it is only implemented in one dimension.

V. Evaluation

A. Performance

We first examine GLAC's performance of 3D localization in detail, and then compare it against three state-of-the-art systems with respect to position accuracy, velocity accuracy, and time-efficiency.

1) 3D Accuracy: We evaluate GLAC's 3D tracking accuracy in both LoS and NLoS scenarios. More than 50 experimental trials are conducted, which amounts to over 10,000 phase readings. For each trial, we program the robotic arm to move the target object along both regular and arbitrary tracks. The position error is calculated as the Euclid distance between the estimated location and the ground truth.

Figure 4 shows CDFs of position errors for both LoS and NLoS cases. We observe that the median accuracies of all x/y/z dimensions are within one centimeter for all scenarios. In particular, the median accuracies of x/y/z dimensions are 0.35 cm, 0.35 cm and 0.52 cm in LoS settings, while counterparts are 0.57 cm, 0.40 cm and 0.73 cm in NLoS settings, which degrade slightly due to lower SNRs. Moreover, the 90th percentiles are within 2 cm in both scenarios for each dimension. Those results fully manifest GLAC's ability for high-precision tracking. This is mainly attributed to our HMM framework that removes ambiguities, which lets the optimal trajectory stand out by gaining more and more likelihoods through mobility.

2) Mobility Comparison: Next, we intend to investigate how GLAC and prior arts perform in different mobilities. The intelligent car is programmed to move along the track at various velocities from 10 cm/s to 50 cm/s. Our experimental trials are conducted in 2D for fair comparison, as RF-IDraw and Tagoram only support 2D-tracking.

Figure 5a shows detailed comparisons of position errors under two representative settings: 10 cm/s for slow movement, and 40 cm/s for fast movement. The results show that GLAC is significantly better than all three state-of-the-art systems in both cases. Specifically, at a velocity of 10 cm/s, the median position accuracies of GLAC, TurboTrack, Tagoram and RF-IDraw are 0.41 cm, 0.90 cm, 7.09 cm and 6.09 cm, respectively. When the velocity increases to 40 cm/s, GLAC's median position accuracy degrades slightly to 0.95 cm. In contrast, those of TurboTrack, Tagoram and RF-IDraw surge to 20.7 cm, 20.6 cm and 31.0 cm, respectively. In other words, in the slow-moving case, GLAC's median position accuracy is 2.2×, 17.3×, and 14.9× better than TurboTrack, Tagoram, and RF-IDraw, and when it comes to the fast-moving case, those gaps enlarge to 21.8×, 21.7×, and 32.6×. As supplementary information, we provide the actual measured and theoretical distance-phase relationship curves at 10 cm/s and 40 cm/s,

as shown in Figure 6a and Figure 6b. When RFID tags operate at higher velocities, the phase measurement error becomes more pronounced; however, our model effectively mitigates this ambiguity.

Besides absolute position, another important metric of mobility is velocity. We compare GLAC's velocity error against three competitors, as shown in Figure 5b. While three competitors do not provide explicit velocity computation, we approximate that as the quotient of the position and time differences between two successive states. Again, the velocity estimation results demonstrate that GLAC is consistently better than prior arts for both dynamic scenarios. In particular, for the slow-moving case, the median velocity accuracy of GLAC is 2.26 cm/s, which is 5.3×, 7.6×, and 7.2× better than TurboTrack, Tagoram, and RF-IDraw, respectively; with the velocity increased to 40 cm/s, the gaps grow to 23.1×, 21×, and 21.4×. Note that although TurboTrack achieves sub-centimeter median position accuracy in the slow-moving case, the corresponding velocity error is as large as 12 cm/s, which is far from 2.26 cm/s, the counterpart of GLAC. This reaffirms that accurate position estimates alone do not guarantee the quality of velocity estimates. Two main factors are contributing to the above performance gaps.

- Prior arts listed above do not take care of asynchronous phase readings, although a few studies like [39] have presented this problem. When fast-moving tags meet low-rate phase readings, the position errors will inevitably increase dramatically. In contrast to that, GLAC novelly synchronizes phases from different antennas for initial state estimation and customizes Kalman filters that can take a single phase for each update.
- In addition, those state-of-the-art systems barely pay attention to mobility models and thus can only derive velocity from position estimates that already contain non-negligible errors. As a result, when the tag moves at a high velocity, position errors inevitable cascade over velocity estimates and future position states. Instead, GLAC models the velocity and position together as the mobility status and update it in an accurate way.

3) Time-Efficiency Comparison: Other than high precision, real-time is one of our design goals. We evaluate the time efficiency of GLAC together with prior systems. We decouple the total time of deriving one position estimate into communication time for reading phases and computation time for localization.

Table II shows the average time cost of over 100 experiment trials. The results clearly show that GLAC is the most time-efficient system. We break these gains down and find that communication is the dominating factor, except for Tagoram. In particular, GLAC is the winner for communication overhead as it only requires a single phase to predict the next location, whereas the other three systems need at least 4 packets.[8] As to computation overhead, GLAC is 5.2× and 24300× better than TurboTrack and Tagoram. This is because TurboTrack uses Particle Filter, which requires a large number of particles to represent the Gaussian distribution, while GLAC employs EKFs that efficiently uses a covariance matrix for Gaussian. Tagoram's huge delay stems from its inefficient exhaustive search. Note that RF-IDraw is the best computation-efficient system, but after counting communication time in, it is not the winner for the overall time cost. In short, GLAC achieves the best overall time-efficiency among all thanks to shortest reading time brought by single-phase update and decent computation time resulted from our fast inference scheme.

[8]Other work [40] has reported faster reading rates through a set of rate adaption techniques, which we plan to investigate in the future for further performance optimization.

TABLE II
LOCALIZATION TIME COMPARISON

	Reading	Computation	Total
GLAC	28ms	78us	28ms
TurboTrack	111ms/4.0×	408us/5.2×	111ms/4.0×
Tagoram	111ms/4.0×	1896ms/24300×	2007ms/72×
RF-IDraw	111ms/4.0×	3us/0.038×	111ms/4.0×

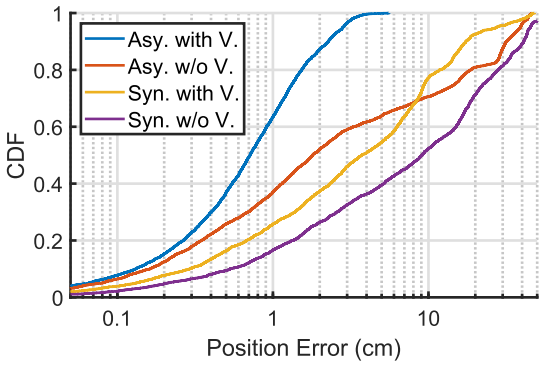


Fig. 7. Decomposing mobility gains. This figure plots the CDFs of position errors for different partial implementations of GLAC.

B. Micro Benchmarks

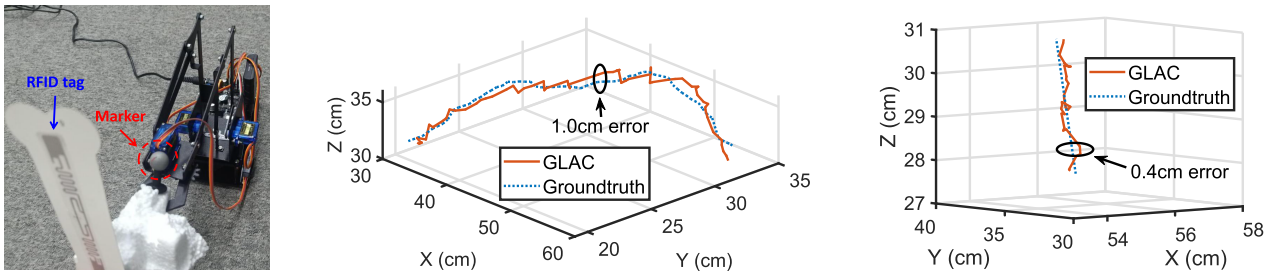
After evaluating the overall performance, we now focus on the performance of constituent modules, which helps understand how GLAC works in detail.

1) Gains of Mobility Modeling: Next, we would like to quantify the effectiveness of our mobility model, namely velocity modelling and ability to dealing with asynchronous phase readings. In particular, we make three variants with partial implementation of the standard GLAC process. The four competitors are as follows.

1) **Asynchronous handling & velocity modeling.** This is the standard GLAC, which is used as the baseline.
2) **Asynchronous handling without velocity modeling.** The first variant scheme is to remove velocity modeling, which changes Equation 3 as follows:

$$\theta_{\mathbf{k+1}} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \theta_{\mathbf{k}} + \mathbf{s_k}.$$

This is built to examine how much the velocity modeling helps localization.
3) **Synchronous handling with velocity modeling.** The second variant scheme is to investigate the impact of asynchronous handling. So we treat every 4 successive



(a) Robotic arm and target tag (b) Rotating (c) Lifting

Fig. 8. Robotic arm tracking: (a) shows the robotic arm carrying an item attached with an RFID tag and an OptiTrack Marker. (b) and (c) show the trajectories of GLAC and the ground truth for rotating and lifting operations.

phase readings from 4 antennas as four synchronous phase readings, which is the way widely adopted in most prior arts [16], [18], [20]. Specifically, we change Equation 4 to:

$$\mathbf{z_k} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \theta_\mathbf{k} + \mathbf{u_k}.$$

Note that $\mathbf{z_k}$ here is the position estimates obtained by the trilateration similar in initial position estimation, and we pick the one nearest to the predicted position as $\mathbf{z_k^*}$ for fast inference. As to likelihood computation, $p(\mathbf{z_{k+1}^*}|\theta_\mathbf{k})$ is replaced by $\prod_{i=1}^{4} p(\mathbf{z_{i,k+1}^*}|\theta_\mathbf{k})$, where $\mathbf{z_{i,k+1}^*}$ is the candidate distance nearest to the predicted state $\theta_\mathbf{k+1}{}'$, as we assume the phase readings across 4 antennas are independent.

4) **Synchronous handling without velocity modeling.** The last scheme is the combination of (2) and (3).

The above four schemes are fed with the same input including various mobilities (from 10 cm/s to 50 cm/s) and output 2D coordinates for simplicity. The CDF results are plotted in Figure 7, and we have the following observations:

- Unsurprisingly, the standard GLAC is the best. In particular, the median position errors of the above four schemes are 0.71 cm, 1.80 cm, 3.73 cm and 9.12 cm.
- Velocity modeling indeed effectively boosts the system performance. In particular, the median position accuracy of scheme 1 is 2.5× better than that of scheme 2. Likewise, scheme 3's median position accuracy is 2.4× better than scheme 4's. The main reason is that without proper velocity modeling, the transition equation's expressiveness becomes quite limited because it only describes a stationary model where all motions are considered as noise, resulting in huge mobile localization errors.
- Asynchronous handling also constructively improves localization accuracy. Specifically, the median accuracies of asynchronous handling (schemes 1&2) are at least 5× better than synchronous handling counterparts (schemes 3&4). The root cause is that synchronous handling may be a good approximation when frame rates are high, like in [16]; however, it inevitably brings more errors when the reading rate (e.g., 30 reads/s) and high-mobility (e.g., 30 cm/s) happen at the same time.

2) Gains of Fast Inference: In addition, we intend to evaluate the computation-efficiency brought by our fast inference design. We compare two schemes, GLAC with fast inference and that with naive inference. So, we run two schemes at the same time with the same data, and then check how well they perform at the same time snapshots.

TABLE III
TIME AND ACCURACY FOR FAST INFERENCE AND NAIVE INFERENCE

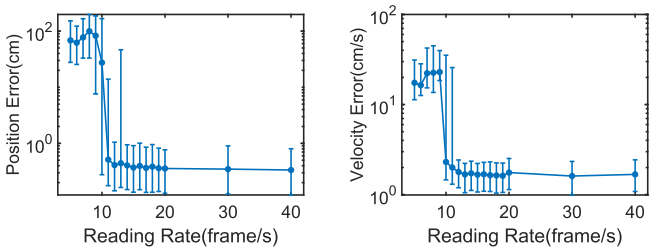
Time	Fast inference/naive inference			
	Position error (cm)	Velocity error (cm/s)	# of EKFs	Iteration depth
10ms	41.1/41.1	4.21/4.21	413/413	1/1
15ms	25.0/41.1	3.90/4.21	145/413	8/1
20ms	0.37/32.4	1.52/4.16	52/6036	38/2
500ms	0.36/27.5	1.49/3.88	1/9.8E4	7.8E4/3
10s	0.37/24.1	1.47/3.74	1/1.4E6	4.2E6/4

Results are listed in Table III. We observe that fast inference is significantly better than naive inference in terms of time-efficiency and accuracy. This is mainly due to that fast inference guarantees the number of trajectories (no. of EKFs) does not increase with time. In consequence, it can go further for iteration depth, which means mobility effectively reduces ambiguities. In particular, with 500 ms, fast inference is left with a trajectory of depth 7,8000 (clearly the optimal one), whereas naive inference's iteration depth is only 3, which makes tons of EKFs (98,000) inefficient. Even with 10s, a large number of EKFs with shallow depths only make a median accuracy of 24.1 cm. Hence, deep iteration depth brought by fast inference is the key to real-time high-precision.

Another observation is with the help of quick stop and fast inference, the optimal trajectory converges even faster. Recall that without fast inference, the number of candidate trajectories increases exponentially as there are n candidate distance observations with a single phase. And each candidate trajectory requires an EKF for updating. Specifically, the number of trajectories for naive inference is $\mathcal{O}(n^\kappa)$, where κ the number of time snapshots, and that of fast inference is $\mathcal{O}(n)$. After combined with quick stop, the number of trajectories for fast inference at 500 ms plummets to only 1, which is 400× lower than the number of trajectories at 10 ms.

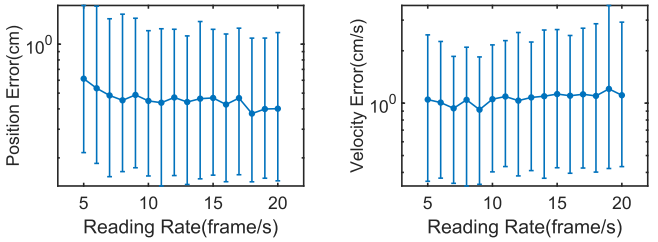
C. Precise Manipulation Showcase

Finally, we show the qualitative performance of GLAC's precise 3D manipulation in Figure 8. During rotating and



(a) Position Error (b) Velocity Error

Fig. 9. Low-rate loss on GLAC. (a) shows the impact of both sampling rate and accelerating motions on the trajectory's median position error. (b) is for the Velocity Error. Error bars indicate the 10th and 90th percentiles.



(a) Position Error (b) Velocity Error

Fig. 10. Performances with GLAC-Plus compared to Figure 9.The additional conditions are the same as above.

lifting operations by the robotic arm, GLAC is able to track 3D-movements at sub-centimeter accuracy, demonstrating GLAC's great potential to enable a range of high-precision robotic applications, such as furniture assembly through multi-robot cooperation, and even robot-assist precise surgeries.

D. GLAC-Plus Optimization

We have implemented a simulation experiment which collects the trajectory's estimation error as changing the frequency of input data. The stimulative tag is moving in a straight line at an acceleration of 4 centimeters per second squared. Figure 9 indicates the results that the performance of GLAC drops sharply below 10 frames per second.

After we upgrade the model and repeat the experiment, Figure 10 demonstrates the advantage of the acceleration modeling. It still maintains excellent performance in the case of limited sampling rates.

In addition to that, GLAC-Plus's combined estimate of location and speed is assumed to be superior to GLAC's. In order to measure the combined error of position and velocity, we define another new quantity given by this expression,

$$\delta = \frac{|\Delta_p|}{\sigma_p} + \frac{|\Delta_v|}{\sigma_v},$$

where Δ_p and Δ_v are the errors between predicted results and the ground truth, σ_p and σ_v are the standard errors of the equipment. As Figure 11 shows, two variable speed motion modes are simulated so as to measure δ. Note that the simulations are conducted at the reading rate of 80 frames per second. Therefore, sampling rate is not an influencing factor in this experiment. In fact, GLAC-Plus reduces the combined error especially for the objects with high acceleration, but

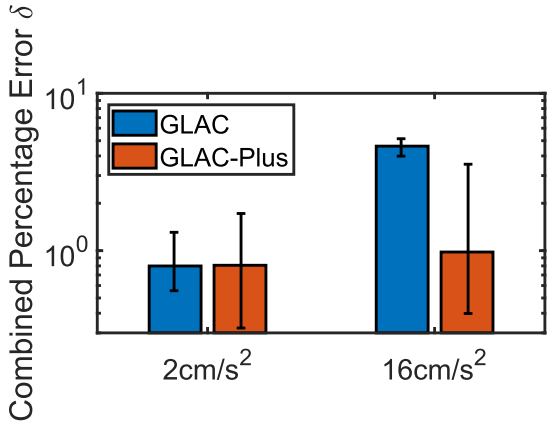


Fig. 11. Combined percentage error, δ, for GLAC and GLAC-Plus. Error bars indicate the 10th and 90th percentiles.

introduces more ambiguity rendering the gaps between error bars larger.

VI. Related Work

RF-Localization. There has been much work on RF localization using COTS and SDR devices, including LTE [41], [42], Bluetooth [43], [44], RFID [16], [18], [20], [36] and WiFi [45], [46], [47], [48], [49]. Among these, RFID-localization can be broadly classified into two categories: fingerprint and trigonometry. The fingerprint-based methods are built based on the assumption that each distinct location has its own unique RF-signature [19], [50], while the trigonometry-based methods are generally more accurate as both triangulation [51], [52] and trilateration [16], [17] are derived from fine-grained phase measurements.

While GLAC falls into the trigonometry category and shares similarity with prior arts from a high level, the crucial difference is that we aim to realize real-time 3D localization using only COTS RFID devices. Bearing this in mind, we novelly model mobility using the HMM and employ it to reduce ambiguities without any other extra hardware or assumptions. This is distinct new thinking and hope it will fuel more community interests along this line.

Unlike GLAC's focus on moving tag tracking, some prototypes assume that the antenna is in motion in static tag scenarios, where the SAR principle has a wide range of applications [53], [54], [55], and some analyses of this principle has been discussed separately [56].

Linear Gaussian Models. It is known that factor analysis [57], mixtures of Gaussian clusters [58], Kalman filter [59], and HMM [60] can be generalized to linear Gaussian modeling [61], [62], which has extensive applications, such as inertial navigation, speech recognition, and stock market forecasting. Based on existing fruitful literature of linear Gaussian models, our work is custom-built. In particular, as the standard type of HMM usually considers discrete states, we carefully examine our problem and build our own HMM where ambiguous states at a time are discrete and do not transit, whereas temporal states are continuous and estimated through Kalman filters. We also design a fast inference scheme to choose the optimal trajectory in real-time.

Mobility Model. Mobility is an important metric and has been widely used in wireless networks from many aspects, including network capacity [63], [64], [65], rate adaption [40], [66], [67], [68], throughput control [69], and wireless sensing [70]. Nevertheless, those works are mainly focused on coarse-grained mobility. For example, Blink [40] only detects the tag is moving or not. Tagwatch [69] requires a couple of seconds to derive mobility states. On the contrary, this work models and derives fine-grained mobility, which is complementary to the above works and thus can be integrated to improve wireless networks, e.g., better network coverage for mobile nodes and higher throughput by being aware of mobility.

VII. Discussion and Conclusion

Finally, a few points are worth elaborating on,

- **Requiring Motion.** One limitation of GLAC stems from fact we need mobile targets to distinguish ambiguities and pick the right trajectory. For static objects, our initial position estimation can be invoked as many as needed. As shown in evaluations, the 90^{th} percentile error for such is less than 2 cm, which is sufficiently accurate for many purposes. Then we must acknowledge that for the accuracy of subsequent tracking, the initial motion speed cannot be too fast, otherwise it will bring more loss of initial estimation, which in turn cascades to affect the overall tracking accuracy. Of course, we can also design a learning algorithm to identify whether the object is moving or not by going through a training process, which is worth further investigation as it would require re-training for various objects.
- **Convergence rate.** The readers may wonder how many steps GLAC takes to make the estimated trajectory and groundtruth match, and what is the percentage of success for GLAC converging to groundtruth each time. Our results have not been quantified, but we currently provide a Recovery mechanism to prevent the results from varying too much, and this does not avoid the possibility of not converging. At least for now, it doesn't seem to be a problem with significant impact. We plan to explore the performance of the average convergence steps of GLAC in future work.
- **Working Range.** As our system is built upon COTS RFID devices, it inherits the relatively short working range, which is about 7-10 meters. The effective range of our prototype is also limited to 80 cm $\times$ 80 cm, which means that the signal-to-noise ratio is relatively high and the phase aliasing problem is not particularly pronounced. Yet, such a limitation is not fundamental to our design. To break it, we intend to explore beamforming [71], which uses multiple antennas to boost signal strengths arriving at target tags. Other possibilities include semi-passive/active tags [72], and novel backscatter paradigms [73], which can typically work at longer distances, 50-200 meters.
- **Manipulatee Size.** Currently, the size of objects that robots manipulate is limited to commercial tags' sizes. While a typical RFID chip has about a $500\mu m \times 500\mu m$ die size, which is indeed tiny, the dimensions of tag inlays are much larger, e.g., an ImpinJ H47 wet inlay is of $47mm \times 47\ mm$, and an Alien 9640 inlay is $95mm \times 8\ mm$. To push such limits, we can either adopt antenna reduction techniques, e.g., Circular Loop Antenna [74], or try RFIDs at much higher frequencies [75] that have much shorter wavelengths.
- **Transferability.** Our key idea that mobility improves accuracy is general and can be extended to other RF systems, such as Bluetooth, WiFi, and LTE. Nevertheless, such adaptations have to take care of RF receivers' form factors, which are usually larger than RFIDs, and identification, which is not natively supported. Other constraints, assumptions, and requirements of those RF systems also pose new challenges, which we are going to investigate in future work.

Overall, we believe that GLAC marks an important step towards precise robot control using COTS RFID systems. With the novel HMM framework and fast inference technology, it pushes the envelope of real-time RFID localization and tracking to the millimeter-level accuracy without requiring any extra hardware and restrictive mobility for tags and readers. By doing so, it paves the way for fast and wide adoption of cheap and readily available commercial RFIDs in robotic applications demanding high-precision, e.g., welding, assembly, and surgeries.

Acknowledgment

The authors would like to thank the Information Science Laboratory, USTC, for the hardware/software services, Prof. Weiwei. Shang and Hao Ji for sharing experimental experience using the OptiTrack system.

References

[1] P. K. Allen, A. Timcenko, B. Yoshimi, and P. Michelman, "Automated tracking and grasping of a moving object with a robotic hand-eye system," *IEEE Trans. Robot. Autom.*, vol. 9, no. 2, pp. 152–165, Apr. 1993.
[2] P. Michelman, "Precision object manipulation with a multifingered robot hand," *IEEE Trans. Robot. Autom.*, vol. 14, no. 1, pp. 105–113, Feb. 1998.
[3] Z. Yaniv and L. Joskowicz, "Precise robot-assisted guide positioning for distal locking of intramedullary nails," *IEEE Trans. Med. Imag.*, vol. 24, no. 5, pp. 624–635, May 2005.
[4] M. P. Murphy and M. Sitti, "Waalbot: An agile small-scale wall-climbing robot utilizing dry elastomer adhesives," *IEEE/ASME Trans. Mechatronics*, vol. 12, no. 3, pp. 330–338, Jun. 2007.
[5] N. Kamamichi, M. Yamakita, K. Asaka, and Z.-W. Luo, "A snake-like swimming robot using IPMC actuator/sensor," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2006, pp. 1812–1817.
[6] J. Rosen et al., "Roboscope: A flexible and bendable surgical robot for single portal minimally invasive surgery," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 2364–2370.
[7] (2019). *Mimioteach Interactive Whiteboard*. [Online]. Available: https://mimio.boxlight.com
[8] H. Meenink et al., "Robot-assisted vitreoretinal surgery," in *Medical Robotics*. Amsterdam, The Netherlands: Elsevier, 2012, pp. 185–209.
[9] (2020). *Okibo*. [Online]. Available: https://www.bimplus.co.uk/worlds-first-autonomous-robot-painter-makes-its-de/
[10] (2022). *Moley*. [Online]. Available: https://moley.com/
[11] K. Kamali, A. Joubair, I. A. Bonev, and P. Bigras, "Elasto-geometrical calibration of an industrial robot under multidirectional external loads using a laser tracker," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 4320–4327.

[12] T. Probst, K.-K. Maninis, A. Chhatkuli, M. Ourak, E. V. Poorten, and L. Van Gool, "Automatic tool landmark detection for stereo vision in robot-assisted retinal surgery," *IEEE Robot. Autom. Lett.*, vol. 3, no. 1, pp. 612–619, Jan. 2018.

[13] G. N. Desouza and A. C. Kak, "Vision for mobile robot navigation: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 2, pp. 237–267, Feb. 2002.

[14] J. C. Byrn et al., "Three-dimensional imaging improves surgical performance for both novice and experienced operators using the da Vinci robot system," *Amer. J. Surg.*, vol. 193, no. 4, pp. 519–522, Apr. 2007.

[15] J. Wang, F. Adib, R. Knepper, D. Katabi, and D. Rus, "RF-compass: Robot object manipulation using RFIDs," in *Proc. 19th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, 2013, pp. 3–14.

[16] Z. Luo, Q. Zhang, Y. Ma, M. Singh, and F. Adib, "3d backscatter localization for fine-grained robotics," in *Proc. USENIX (NSDI)*, 2019, pp. 765–781.

[17] Y. Ma, N. Selby, and F. Adib, "Minding the billions: Ultra-wideband localization for deployed RFID tags," in *Proc. 23rd Annu. Int. Conf. Mobile Comput. Netw.*, Oct. 2017, pp. 248–260.

[18] J. Wang, D. Vasisht, and D. Katabi, "RF-IDraw: Virtual touch screen in the air using RF signals," in *Proc. ACM Conf. SIGCOMM*, Aug. 2014, pp. 235–246.

[19] J. Wang and D. Katabi, "Dude, where's my card: RFID positioning that works with multipath and non-line of sight," in *Proc. ACM SIGCOMM Conf. (SIGCOMM)*, Aug. 2013, pp. 51–62.

[20] L. Yang, Y. Chen, X.-Y. Li, C. Xiao, M. Li, and Y. Liu, "Tagoram: Real-time tracking of mobile RFID tags to high precision using COTS devices," in *Proc. 20th Annu. Int. Conf. Mobile Comput. Netw.*, Sep. 2014, pp. 237–248.

[21] L. Shangguan and K. Jamieson, "The design and implementation of a mobile RFID tag sorting robot," in *Proc. 14th Annu. Int. Conf. Mobile Syst., Appl., Services*, Jun. 2016, pp. 31–42.

[22] (2021). *Glac*. [Online]. Available: https://github.com/Sea-USTC/GLAC

[23] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. Basic Eng.*, vol. 82, no. 1, pp. 35–45, Mar. 1960.

[24] A. H. Jazwinski, *Stochastic Processes Filtering Theory*. Chelmsford, MA, USA: Courier Corporation, 2007.

[25] E. DiGiampaolo and F. Martinelli, "Mobile robot localization using the phase of passive UHF RFID signals," *IEEE Trans. Ind. Electron.*, vol. 61, no. 1, pp. 365–376, Jan. 2014.

[26] (2020). *Thingmagic M6E*. [Online]. Available: https://www.jadaktech.com

[27] (2020). *Alien 9640 UHF RFID Tag*. [Online]. Available: https://www.alientechnology.com

[28] (2020). *Monza 4D Rain RFID Tag*. [Online]. Available: https://www.impinj.com/platform/endpoints/monza-4d

[29] (2020). *Dogbone With Nxp Ucode G2Il*. [Online]. Available: https://www.smartrac-group.com/files/content/Products_Solutions/PDF/0053_SMARTRAC_DOGBONE.pdf

[30] (2020). *Vanbot*. [Online]. Available: https://cheng-long.taobao.com

[31] (2022). *Servo*. [Online]. Available: https://m.tb.cn/h.UUY6Yo8?tk=tYI6db70nBY

[32] (2022). *CDFF*. [Online]. Available: https://statproofbook.github.io/D/cdf.html

[33] (2020). *Thingmagic Mercury Api*. [Online]. Available: https://www.jadaktech.com/products/thingmagic-rfid/thingmagic-mercury-api/

[34] P. V. Nikitin, R. Martinez, S. Ramamurthy, H. Leland, G. Spiess, and K. V. S. Rao, "Phase based spatial identification of UHF RFID tags," in *Proc. IEEE Int. Conf. RFID (IEEE RFID)*, Apr. 2010, pp. 102–109.

[35] C. Li et al., "ReLoc: Hybrid RSSI- and phase-based relative UHF-RFID tag localization with COTS devices," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 10, pp. 8613–8627, Oct. 2020.

[36] H. Wang and W. Gong, "RF-pen: Practical real-time RFID tracking in the air," *IEEE Trans. Mobile Comput.*, vol. 20, no. 11, pp. 3227–3238, Nov. 2021.

[37] (2020). *Optitrack*. [Online]. Available: https://www.optitrack.com

[38] L. Shangguan and K. Jamieson, "Leveraging electromagnetic polarization in a two-antenna whiteboard in the air," in *Proc. 12th Int. Conf. Emerg. Netw. Exp. Technol.*, Dec. 2016, pp. 443–456.

[39] A. Motroni, A. Buffi, P. Nepa, and B. Tellini, "Sensor-fusion and tracking method for indoor vehicles with low-density UHF-RFID tags," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–14, 2021.

[40] P. Zhang, J. Gummeson, and D. Ganesan, "BLINK: A high throughput link layer for backscatter communication," in *Proc. 10th Int. Conf. Mobile Syst., Appl., Services*, Jun. 2012, pp. 99–112.

[41] J. A. del Peral-Rosado, J. A. López-Salcedo, G. Seco-Granados, F. Zanier, and M. Crisci, "Achievable localization accuracy of the positioning reference signal of 3GPP LTE," in *Proc. Int. Conf. Localization (GNSS)*, Jun. 2012, pp. 1–6.

[42] T. Wigren, "LTE fingerprinting localization with altitude," in *Proc. IEEE Veh. Technol. Conf. (VTC Fall)*, Sep. 2012, pp. 1–5.

[43] S. S. Chawathe, "Beacon placement for indoor localization using Bluetooth," in *Proc. 11th Int. IEEE Conf. Intell. Transp. Syst.*, Oct. 2008.

[44] A. N. Raghavan, H. Ananthapadmanaban, M. S. Sivamurugan, and B. Ravindran, "Accurate mobile robot localization in indoor environments using Bluetooth," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 4391–4396.

[45] W. Gong and J. Liu, "Robust indoor wireless localization using sparse recovery," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2017, pp. 847–856.

[46] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, "SpotFi: Decimeter level localization using WiFi," in *Proc. ACM Conf. Special Interest Group Data Commun.*, Aug. 2015, pp. 269–282.

[47] W. Gong and J. Liu, "SiFi: Pushing the limit of time-based WiFi localization using a single commodity access point," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 2, no. 1, pp. 1–21, Mar. 2018.

[48] Z. Yang and W. Gong, "Decimeter-level WiFi tracking in real-time," in *Proc. IEEE/ACM 28th Int. Symp. Quality Service (IWQoS)*, Jun. 2020, pp. 1–10.

[49] W. Gong and J. Liu, "RoArray: Towards more robust indoor localization using sparse recovery with commodity WiFi," *IEEE Trans. Mobile Comput.*, vol. 18, no. 6, pp. 1380–1392, Jun. 2019.

[50] L. M. Ni, Y. Liu, Y. Cho Lau, and A. P. Patil, "LANDMARC: Indoor location sensing using active RFID," in *Proc. 1st IEEE Int. Conf. Pervasive Comput. Commun., (PerCom)*, 2003.

[51] J. Wang, J. Xiong, H. Jiang, X. Chen, and D. Fang, "D-watch: Embracing," in *Proc. ACM CoNEXT*, 2016, pp. 253–266.

[52] S. Azzouzi, M. Cremer, U. Dettmar, R. Kronberger, and T. Knie, "New measurement results for the localization of UHF RFID transponders using an angle of arrival (AoA) approach," in *Proc. IEEE Int. Conf. RFID*, Apr. 2011, pp. 91–97.

[53] P. Tripicchio et al., "A synthetic aperture UHF RFID localization method by phase unwrapping and hyperbolic intersection," *IEEE Trans. Autom. Sci. Eng.*, vol. 19, no. 2, pp. 933–945, Apr. 2022.

[54] A. Tzitzis et al., "Localization of RFID tags by a moving robot, via phase unwrapping and non-linear optimization," *IEEE J. Radio Freq. Identificat.*, vol. 3, no. 4, pp. 216–226, Dec. 2019.

[55] A. Motroni et al., "SAR-based indoor localization of UHF-RFID tags via mobile robot," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Sep. 2018, pp. 1–8.

[56] Y. Ma, H. Liu, Y. Zhang, and Y. Jiang, "The influence of the nonideal phase offset on SAR-based localization in passive UHF RFID," *IEEE Trans. Antennas Propag.*, vol. 68, no. 8, pp. 6346–6354, Aug. 2020.

[57] H. H. Harman, *Modern Factor Analysis*. Chicago, IL, USA: Univ. Chicago Press, 1976.

[58] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: MIT Press, 2012.

[59] G. Welch and G. Bishop, "An introduction to the Kalman filter," Dept. Comput. Sci., Univ. North Carolina Chapel Hill, Tech. Rep., 1995.

[60] O. Cappé, E. Moulines, and T. Rydén, *Inference Hidden Markov Models*. Cham, Switzerland: Springer, 2006.

[61] S. Roweis and Z. Ghahramani, "A unifying review of linear Gaussian models," *Neural Comput.*, vol. 11, no. 2, pp. 305–345, Feb. 1999.

[62] A.-V. I. Rosti and M. Gales, "Generalised linear Gaussian models," Dept. Eng., Univ. Cambridge, Tech. Rep., 2001.

[63] S. Toumpis, "Large wireless networks under fading, mobility, and delay constraints," in *Proc. IEEE INFOCOM*, Mar. 2004.

[64] M. Grossglauser and D. N. C. Tse, "Mobility increases the capacity of ad hoc wireless networks," *IEEE/ACM Trans. Netw.*, vol. 10, no. 4, pp. 477–486, Aug. 2002.

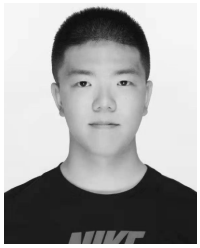
[65] J. Zhao, W. Gong, and J. Liu, "X-tandem: Towards multi-hop backscatter communication with commodity WiFi," in *Proc. 24th Annu. Int. Conf. Mobile Comput. Netw.*, Oct. 2018, pp. 497–511.

[66] L. Sun, S. Sen, and D. Koutsonikolas, "Bringing mobility-awareness to WLANs using PHY layer information," in *Proc. 10th ACM Int. Conf. Emerg. Netw. Exp. Technol.*, Dec. 2014, pp. 53–66.

[67] W. Gong, S. Chen, and J. Liu, "Towards higher throughput rate adaptation for backscatter networks," in *Proc. IEEE 25th Int. Conf. Netw. Protocols (ICNP)*, Oct. 2017, pp. 1–10.
[68] W. Gong, S. Chen, J. Liu, and Z. Wang, "MobiRate: Mobility-aware rate adaptation using PHY information for backscatter networks," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Apr. 2018, pp. 1259–1267.
[69] Q. Lin, L. Yang, H. Jia, C. Duan, and Y. Liu, "Revisiting reading rate with mobility: Rate-adaptive reading in COTS RFID systems," in *Proc. 13th Int. Conf. Emerg. Netw. Exp. Technol.*, Nov. 2017, pp. 199–211.
[70] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, "Understanding and modeling of WiFi signal based human activity recognition," in *Proc. 21st Annu. Int. Conf. Mobile Comput. Netw.*, Sep. 2015, pp. 65–76.
[71] H. Cox, R. Zeskind, and M. Owen, "Robust adaptive beamforming," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-35, no. 10, pp. 1365–1376, Oct. 1987.
[72] (2020). *Confidex Viking Product Family*. [Online]. Available: https://www.confidex.com/smart-industries/confidex-viking/
[73] P. Zhang, D. Bharadia, K. Joshi, and S. Katti, "HitchHike: Practical backscatter using commodity WiFi," in *Proc. 14th ACM Conf. Embedded Netw. Sensor Syst. (CD-ROM)*, Nov. 2016, pp. 259–271.
[74] H.-K. Ryu and J.-M. Woo, "Size reduction in UHF band RFID tag antenna based on circular loop antenna," in *Proc. 18th Int. Conf. Appl. Electromagn. Commun.*, Feb. 2005.
[75] A. Attaran, R. Rashidzadeh, and R. Muscedere, "Rotman lens combined with wide bandwidth antenna array for 60 GHz RFID applications," *Int. J. Microw. Wireless Technol.*, vol. 9, no. 1, pp. 219–225, Feb. 2017.

Wei Gong (Senior Member, IEEE) received the B.S. degree from the Department of Computer Science and Technology, Huazhong University of Science and Technology, and the M.S. and Ph.D. degrees from the Department of Computer Science and Technology and the School of Software, Tsinghua University. He is currently a Professor with the School of Computer Science and Technology, University of Science and Technology of China. His research interests include backscatter networks, edge systems, and the IoT applications.

Haoyu Wang received the B.S. degree from the University of Science and Technology of China and the master's degree from Tsinghua University. He is currently a System Development Engineer working on ECS stability with Alibaba Cloud. His research interests include time-series data and wireless networks.



Siyi Li is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, University of Science and Technology of China. His research interests include wireless networks and the Internet of Things applications.

Si Chen (Student Member, IEEE) received the bachelor's degree from the China University of Geosciences and the master's degree from Simon Fraser University, where she is currently pursuing the Ph.D. degree with the School of Computing Science. Her research interests include wireless networks and big data.